

Компьютерные Т_EXнологии

Е. М. Балдин*

КОМПЬЮТЕРНАЯ ТИПОГРАФИЯ L^AT_EX



Новосибирск 2008, 2012, 2013

*e-mail: E.M.Baldin@inp.nsk.su

«Компьютерная типография L^AT_EX» была выпущена издательством «БХВ-Петербург» в 2008 году за номером ISBN 978-5-9775-0230-6. По договору с издательством права на текст возвратились ко мне, и я решил выложить его под свободной лицензией Creative Commons Attribution-ShareAlike 3.0 Unported (CC-BY-SA 3.0). Если для вашего проекта необходима смена лицензии на другую, то свяжитесь со мной по этому поводу.

Книга является руководством по использованию текстового процессора L^AT_EX. Описаны все базовые элементы процесса набора и вёрстки, упомянуто свыше 1300 команд L^AT_EX и около 750 пакетов, стилей и классов, которые можно использовать при наборе любых текстовых документов. Рассмотрены создание презентаций, вёрстка таблиц, вставка графических объектов, формирование библиографии, алфавитного указателя и многие другие задачи. На конкретных примерах показано, что компьютерной типографии под управлением L^AT_EX доступны проекты любой сложности и направленности. Значительная часть книги посвящена главному преимуществу L^AT_EX перед другими текстовыми процессорами — набору математики.

Эта книга была создана с нуля целиком и полностью исключительно средствами L^AT_EX. Я ответственен и за текст, и за вёрстку, поэтому буду благодарен любым замечаниям и конструктивным советам по поводу улучшения качества электронной версии книги.

Для широкого круга пользователей

The book “Computer Typesetting Using L^AT_EX” was published by the “BHV-Petersburg” (Russia) publishing house in 2008 under ISBN 978-5-9775-0230-6. Under the contract with the publishing house, I have now regained exclusive rights to the content and decided to publish it under a Creative Commons Attribution-ShareAlike 3.0 Unported License (CC-BY-SA 3.0). Please contact me if you require a different type of copyright license for your purposes.

This book is a guide to using the L^AT_EX document preparation system. It describes the complete fundamentals of the typesetting and layout processes and covers at least 1300 L^AT_EX commands and about 750 packages, styles, and classes that can be used in the typesetting of any documents. It also deals with creating presentations, table layouts, inserting graphics, bibliography and index, and many other tasks. Specific examples demonstrate that publishing systems using L^AT_EX can handle projects of any scope and complexity. A major part of the book is devoted to the typesetting of mathematical text, which is the key differentiator of L^AT_EX compared to other word processors.

The book was written completely and exclusively by means of L^AT_EX tools. As the author of both the content and the typesetting, I will greatly appreciate any comments and useful advice on improving the quality of the electronic version of the book.

Краткое оглавление

1. Про $\text{\LaTeX}$, эту и другие книги	11
2. Запускаем $\text{\LaTeX}$	21
3. Базовые элементы	29
4. Начала математики	45
5. Вставка графики	57
6. Начала программирования	69
7. Вёрстка	79
8. Путеводитель по классам $\text{\LaTeX}$	97
9. Делаем презентации	107
10. Базовые навыки	121
11. Всё о таблицах	149
12. Справочно-поисковый аппарат издания	163
13. Математика	185
14. Документация и программный код	207
15. Точные науки	221
16. Гуманитарные направления	235
17. Каталог пакетов $\text{\TeX}$ Live	247

Оглавление

1. Про ЛАТЭХ, эту и другие книги	11
1.1. Немного истории	11
1.1.1. Доисторический период	11
1.1.2. Роль Человека в истории	12
1.1.3. ТРХническое отступление	13
1.1.4. Появление ЛАТЭХ	13
1.2. Дистрибутивы ЛАТЭХ	14
1.3. Об этой книге	15
1.3.1. Оформление ключевых слов и примеров	16
1.3.2. Как читать эту книгу	16
1.4. Литература	17
1.4.1. Классика	17
1.4.2. Учебники и справочники	18
1.4.3. ЛАТЭХ в России	18
2. Запускаем ЛАТЭХ	21
2.1. Проблемы с компиляцией	23
2.2. ЛАТЭХ-конвейер	24
3. Базовые элементы	29
3.1. «Командная логика»	29
3.2. Логика документа	31
3.2.1. Структура ЛАТЭХ-файла	31
3.2.2. Класс документа	33
3.2.3. Стили	34
3.2.4. Тело документа	35
3.3. Логика набора	35
3.3.1. Печатаем текст	35
3.3.2. Пунктуация	37
3.4. Структурная логика	40
3.4.1. Титульный лист	40
3.4.2. Секционирование	41
3.4.3. Перекрёстные ссылки	42

4. Начала математики	45
4.1. Набор формул	46
4.2. Кириллица в формулах	47
4.3. Школьная математика	48
4.3.1. Отделяем целую часть от дробной	49
4.3.2. Индексы	49
4.3.3. Математические символы	50
4.3.4. Корни	51
4.3.5. Дроби	52
4.3.6. Квадратное уравнение	52
4.3.7. Производная и интеграл	52
4.3.8. Функции	54
4.3.9. Скобки	54
4.4. Перенос формул	55
5. Вставка графики	57
5.1. Encapsulated PostScript	57
5.2. Как из растра сделать EPS	59
5.3. graphicx	60
5.3.1. Определение своих правил	61
5.4. Плавающие объекты	62
5.4.1. Управление плавающими объектами	63
5.4.2. «Упаковка» картинок в один float	64
5.4.3. Картинки «в оборку»	65
5.4.4. Подписи к рисункам	66
6. Начала программирования	69
6.1. Определённые размеры и переменные длины	69
6.2. Счётчики	71
6.3. Создаём свои ...	73
6.3.1. Инструменты	75
6.3.2. Вычисления с calc	75
6.3.3. Условные операторы и циклы	76
7. Вёрстка	79
7.1. Скелет страницы	79
7.1.1. Выбор размера бумаги	81
7.1.2. Ориентация	82
7.2. Меняем макет	83
7.2.1. Двигаем размеры	83
7.2.2. Стили страницы	86
7.2.3. Буклеты	86
7.3. «Причёсываем» текст	88

7.3.1.	Строка	88
7.3.2.	Горизонтальные пробелы	89
7.3.3.	Форматирование параграфа	89
7.3.4.	Страница	90
7.3.5.	Висячая строка	90
7.3.6.	Вертикальные просветы	91
7.3.7.	Печать через две строки	91
7.4.	Многоколоночная вёрстка	91
8.	Путеводитель по классам ЛТХ	97
8.1.	Зачем нужны эти классы?	97
8.2.	Классовая база	98
8.3.	Классификация	99
8.3.1.	Модификации и улучшения базы	99
8.3.2.	Пишем письма	100
8.3.3.	Поддерживаем стандарты	101
8.3.4.	Верстаем книги	102
8.3.5.	Создаём отчёты	102
8.3.6.	Делаем презентации	103
8.3.7.	Журнальные и конференционные классы	104
8.3.8.	Организуем резюме	104
8.3.9.	Защищаем диссертации	105
8.3.10.	Всякая всячина	106
9.	Делаем презентации	107
9.1.	slides	108
9.2.	Немного о PDF	108
9.2.1.	Простота создания	109
9.2.2.	Переносимость	110
9.2.3.	Интерактивность	110
9.3.	beamer	111
10.	Базовые навыки	121
10.1.	Интернационализация и локализация	121
10.2.	Символы	123
10.2.1.	Валютные символы	125
10.2.2.	Копирайт и копилефт	125
10.2.3.	Акценты	126
10.3.	К вопросу о шрифтах	126
10.4.	Работа с текстом	127
10.4.1.	Выделение текста	127
10.4.2.	Боксы	129
10.4.3.	Цитаты	130

10.4.4. Перечни	131
10.5. Рубрикация	133
10.6. Ссылки, сноски и примечания	137
10.6.1. Механизм ссылок	137
10.6.2. Подстрочные примечания	138
10.6.3. Затекстовые примечания	140
10.6.4. Заметки на полях	141
10.6.5. WWW-элементы	141
10.7. Процесс подготовки документа	143
10.7.1. Авторские метки с FiXme	143
10.7.2. Нумерация строк с lineno	143
10.7.3. Водяные знаки	144
10.7.4. Сложные документы	145
11. Всё о таблицах	149
11.1. Немного теории	149
11.2. tabbing	150
11.3. tabular и array	151
11.3.1. Подписи к таблицам	154
11.3.2. К вопросу о разделительных линиях	155
11.3.3. Клетки	155
11.3.4. Выравнивание чисел	157
11.3.5. Доступ к данным	158
11.3.6. Клоны tabular	158
11.4. Многополосные таблицы	159
11.5. Вывод	161
11.6. И это тоже таблицы?	162
12. Справочно-поисковый аппарат издания	163
12.1. Оглавление	164
12.2. Колонтитулы	166
12.3. Библиография	168
12.4. Работаем с ВивТ _Е X	170
12.5. Предметный указатель	174
12.6. xindy	178
12.7. Глоссарий	180
13. Математика	185
13.1. Математические символы	185
13.1.1. Типы символов	185
13.1.2. Список символов	186
13.1.3. Греческие символы	187
13.1.4. Акценты	188

13.1.5. Многоточия	189
13.1.6. Математика в текстовой моде	190
13.1.7. Бинарные операторы	190
13.1.8. Символы отношений	191
13.1.9. Стрелки	193
13.1.10. Разделители	194
13.1.11. Операторы переменного размера	195
13.2. Производные и интегралы	195
13.3. Матрицы, тензоры и диаграммы	197
13.4. Пробелы в формулах	199
13.5. Многострочные формулы	200
13.6. Нумерация формул	204
13.7. Теоремы	205
14. Документация и программный код	207
14.1. Форматирование кода	207
14.2. LCD-дисплей	211
14.3. Битовые поля	212
14.4. Представление алгоритмов	214
14.5. История изменений	216
14.6. Исходники L ^A T _E X и контроль версий	218
15. Точные науки	221
15.1. Физика	221
15.1.1. Системы единиц	222
15.1.2. Физика высоких энергий	223
15.1.3. Электронные схемы	226
15.1.4. Лабораторные работы	228
15.2. Химия	229
15.2.1. Лабораторные работы	229
15.2.2. Химические формулы	229
15.2.3. Структурные формулы	231
15.3. Биология	233
16. Гуманитарные направления	235
16.1. Стихи и пьесы	235
16.2. Музыка	238
16.3. Языкознание	243
17. Каталог пакетов T_EX Live	247
17.1. Набор текста	247
17.1.1. Интернационализация и локализация	247
17.1.2. Русификация	248
17.1.3. Пунктуация	248

17.1.4. Выделение текста	249
17.1.5. Разделительные линии	249
17.1.6. Форматирование параграфа	249
17.1.7. Перечни	249
17.1.8. Неформатированный текст	250
17.1.9. Цитирование	250
17.1.10. Рубрикация	250
17.1.11. Титульный лист	251
17.1.12. Перекрёстные ссылки	251
17.1.13. Сноски	251
17.1.14. Затекстовые примечания	252
17.1.15. Заметки на полях	252
17.1.16. WWW-элементы	252
17.1.17. PDF	252
17.1.18. Боксы	252
17.1.19. Процесс подготовки документа	253
17.1.20. Составные документы	253
17.1.21. Всякая всячина	254
17.2. Точные науки	254
17.2.1. Математика	254
17.2.2. Информатика	257
17.2.3. Физика	257
17.2.4. Химия	258
17.2.5. Биология	259
17.3. Школа и университет	259
17.4. Гуманитарные направления	259
17.4.1. Поэзия, пьесы и критика	260
17.4.2. Языкознание	260
17.4.3. Музыка	260
17.5. Делу время — потехе час	261
17.5.1. Рецепты	261
17.5.2. Шахматы	261
17.5.3. Игры, в которые играют люди	261
17.5.4. Время	262
17.5.5. Карточки и визитки	262
17.6. Плавающие объекты	262
17.6.1. Создание плавающих объектов	262
17.6.2. Типы плавающих объектов	262
17.6.3. Подписи к плавающим объектам	263
17.6.4. Размещение плавающих объектов	263
17.7. Графика	264
17.7.1. Добавление иллюстраций	264
17.7.2. Управление цветом	264

17.7.3. Графика средствами LaTeX	264
17.8. Всё о таблицах	266
17.8.1. К вопросу о разделительных линиях	266
17.8.2. Клетки	266
17.8.3. Выравнивание чисел	266
17.8.4. Клоны tabular	266
17.8.5. Многостраничные таблицы	267
17.8.6. Вывод	267
17.8.7. Доступ к данным	267
17.9. Вёрстка	267
17.9.1. Ориентация страницы	267
17.9.2. Макет	268
17.9.3. Вертикальные расстояния между элементами текста	268
17.9.4. Многоколоночная вёрстка	268
17.10. Путеводитель по классам LaTeX	269
17.10.1. Поддерживаем стандарты	270
17.10.2. Пишем письма и отсылаем факсы	270
17.10.3. Верстаем книги	270
17.10.4. Создаём отчёты	271
17.10.5. Делаем презентации	271
17.10.6. Защищаем диссертации	271
17.10.7. Организуем резюме	272
17.10.8. Журнальные и конференционные классы	272
17.10.9. Делаем газеты и буклеты	273
17.11. Справочно-поисковый аппарат издания	274
17.11.1. Оглавление	274
17.11.2. Колонтитулы	274
17.11.3. Библиография	274
17.11.4. Указатель	275
17.12. Программирование в среде LaTeX	276
17.12.1. Счётчики и другие переменные	276
17.12.2. Вычисления с использованием LaTeX	276
17.12.3. Условные выражения и циклы	276
17.12.4. Создание новых команд	277
17.12.5. Разбор параметров	277
17.12.6. Работа с файлами	277
17.12.7. Пакеты LaTeX	278
Установка TeX Live	279

Глава 1

Про L^AT_EX, эту и другие книги

Если кто-то другой набирает ваше произведение, то у Вас нет возможности контролировать появление ошибок; если же Вы выполняете эту работу самостоятельно, то можете винить только себя.

Дональд Э. Кнут

Человеческая цивилизация зависит от книг. Передача знаний от поколения к поколению — это то, что делает человека разумным. Написание книги всегда было/есть/будет одним из самых сложных видов деятельности. L^AT_EX берёт на себя техническую часть по подготовке рукописи, оставляя человеку больше времени на творчество и, в то же время, позволяя ему контролировать весь процесс создания от начала и до конца.

1.1. Немного истории

Есть популярная идея, что «история учит лишь тому, что ничему не учит». Возможно, это так. Но чтобы понять логику текущих событий и явлений, всё-таки необходимо знать, как «оно» когда-то зарождалось и почему «оно» до сих пор существует.

1.1.1. Доисторический период

Сначала вообще не было компьютеров, и люди всё писали вручную. Но прогресс неумолим, и вслед за возникновением печатной машины появилась значительная коммерческая выгода от создания книг. Время шло, процесс печати удешевлялся, число книг возрастало — все были довольны, пока не случилось. . .

1.1.2. Роль Человека в истории

Говорят, живёт на свете Дональд Кнут.
 Доктор Кнут, поверьте, дети, страшно крут.
 И неважно, что он пишет не о том,
 Главное, чтоб всё же вышел пятый том.

Виктор «Витус» Вагнер

Дональд Эрвин Кнут (Donald Ervin Knuth) является одним из немногих людей, благодаря которым информатика заслуженно носит звание научной дисциплины. Произведение, которое принесло ему широкую известность, знают как «пятитомник» «Искусство программирования»¹. После того как в 1975 году был издан третий том «пятитомника», издатель окончательно избавился от печатной машины с металлическим набором типа «монотип» и заменил его на фотонаборное устройство. Результат превзошёл все ожидания: получив оттиски, сделанные по новой технологи, Д. Э. Кнут, который как раз подготовил второе издание второго тома, сильно загрустил. Сама мысль, что книги, написание которых он потратил свыше пятнадцати лет, будут так плохо выглядеть, не давала Кнуту покоя.

Новые машины были не аналоговыми, а дискретными. Буквы составлялись из точек. «Это объект для компьютерной науки», — подумал Д. Э. Кнут и решил научить компьютер делать буквы из точек такими, как надо, то есть красивыми. Поначалу задача казалась несложной, и в планах было решить проблему полностью в течение летних месяцев 1977 года. Д. Э. Кнут потом признавал, что это был его личный рекорд по недооценке сложности проекта.

Примерно через десять лет после начала работы над проектом, системы METAFONT (создание шрифтов) и $\text{\TeX}$ (лучшая программа для разбиения абзацев на строки) были стабилизированы (версия 2.7 для METAFONT и 3.1 для $\text{\TeX}$). Кнут отошёл от активной разработки. В дальнейшем METAFONT и $\text{\TeX}$ модифицировались только для целей исправления ошибок. На текущий момент рабочая версия METAFONT — 2.71828, а $\text{\TeX}$ — 3.141592. Кнут завещал, что после его смерти версии будут заморожены и равны числу e и числу π , соответственно, а все оставшиеся неисправленные ошибки будут считаться особенностями реализации.

Сегодня $\text{\TeX}$ — это самый «безошибочный» программный пакет. За обнаружение ошибки в своей программе Кнут выплачивает вознаграждение, не очень большое, но невероятно ценное. Код программы $\text{\TeX}$ выпускался отдельной книгой « $\text{\TeX}$: The Program» (ISBN: 0201134373). $\text{\TeX}$ является примером свободной программы, которая возникла в академической среде задолго до наступления эпохи GPL.

¹На сайте мастера <http://www-cs-faculty.stanford.edu/~knuth/> можно найти предварительные версии некоторых глав будущего четвёртого тома.

Сегодня мастер на пенсии и всё своё время посвящает написанию «пятитомника». На его домашней страничке можно заметить, что дело явно движется. Ждём результатов с нетерпением.

1.1.3. Т_ЕХ_ническое отступление

В основу Т_ЕХ была заложена относительно простая идея. Т_ЕХ работает только с боксами (box) и клеем (glue). Элементарные боксы — это буквы, которые объединяются в боксы-слова, которые в свою очередь сливаются в боксы-строчки, боксы-абзацы и так далее. Между боксами «разлит» клей, который имеет некоторую ширину по умолчанию и степени увеличения/уменьшения этой ширины. Объединяясь в бокс более высокого порядка, элементарные боксы могут шевелиться, но после того как найдено оптимальное решение, это состояние замораживается, и полученный бокс выступает как единое целое. Оптимальное решение находится с помощью системы штрафов за то, что клея больше или меньше чем некое оптимальное значение, определённое Кнудом эмпирически, а также за разрывы абзаца в неподходящем месте. Чем меньше штрафа было получено, тем размещение «красивее». В зависимости от выбранной системы штрафов меняется и форматирование.

Когда Д. Э. Кнут создавал Т_ЕХ, он много думал. Причём думал не только об алгоритмах и их программной реализации. В частности, он нашёл время подумать о том, как назвать своё произведение. Т_ЕХ читается как «тех». Последняя буква — вовсе не английская буква «икс» (x), а греческая «хи» (χ). Также он продумал и правила изображения этого названия. С тех пор в Т_ЕХ-сообществе возникла мода на создание Т_ЕХ-лого.

1.1.4. Появление Л_АТ_ЕХ

Первоначально Кнут думал, что у Т_ЕХ будет множество модификаций. Предполагалось, что каждая уважающая себя типография будет держать мастера «Т_ЕХника» для создания своей уникальной версии Т_ЕХ под свои не менее уникальные нужды. Позже, когда стало очевидно, что развитие «Т_ЕХнологий» пошло совсем по другому пути, в него были добавлены управляющие конструкции, чтобы Т_ЕХ стал в полном смысле языком программирования.

А началось всё с Лесли Лэмпорта, который в начале 80-х годов XX века начал разработку издательской системы Л_АТ_ЕХ на основе Т_ЕХ. Л_АТ_ЕХ представляет из себя набор макросов на языке Т_ЕХ, позволяющих решить ту или иную задачу. Иными словами, это сборник рецептов. Чтобы выбрать сценарий стирки на автоматической стиральной машине, нет необходимости думать в терминах скорости оборотов, уровня воды и количества порошка — достаточно просто выбрать готовое решение. Чтобы пользоваться системой Л_АТ_ЕХ, не надо быть Т_ЕХником — достаточно выбрать готовый стиль и использовать несколько простых команд в зависимости от того, что нужно сделать.

1.2. Дистрибутивы ЛАТЭХ

ЛАТЭХ не является монолитной программой. ЛАТЭХ состоит из набора пакетов, причём набор пакетов не фиксирован, что позволяет создавать дистрибутивы, преследующие ту или иную цель.

На сегодня все дистрибутивы ЛАТЭХ имеют общий корень, и этот корень носит название CTAN или The Comprehensive TeX Archive Network. Подробнее об этом можно прочитать на страничке проекта <http://www.ctan.org>.

CTAN — это международный файловый архив. Цель CTAN — собрать всё, что относится к ТЭХ и его производным, в одном месте. Основные сайты, представляющие CTAN:

- <ftp://ftp.dante.de/tex-archive/>
- <ftp://ftp.tex.ac.uk/tex-archive/>
- <ftp://ctan.tug.org/tex-archive/>

Зеркала CTAN разбросаны по всему свету, в частности, в России находятся:

- <ftp://ftp.chg.ru/pub/TeX/CTAN/>
- <ftp://ftp.nsu.ru/mirrors/ftp.dante.de/tex-archive/>

CTAN — это репозиторий, в который стекаются все сколько-нибудь стоящие наработки в области ТЭХ-строения. Модель была настолько успешна, что её на вооружение взяло perl-сообщество, организовав CPAN (The Comprehensive Perl Archive Network). Затем этим же путём последовало R-сообщество², создав архив CRAN (The Comprehensive R Archive Network), Python-сообщество с архивом PyPI (Python Package Index), в какой-то мере Ruby-сообщество со своим RubyGems и Java-сообщество с CJAN и JSAN.

Флагманом ТЭХ-сообщества, или ТЭХ User Groups (<http://www.tug.org/>), сегодня является дистрибутив ТЭХ Live (<http://www.tug.org/texlive/>).

ТЭХ Live создавался как дистрибутив, который можно было бы запускать прямо с CD. Базой для него стал ранее самый популярный в среде GNU/Linux дистрибутив teTeX. Первая версия ТЭХ Live была выпущена в 1996 году. Со временем дистрибутив рос и к 2003 году ТЭХ Live перерос размеры CD и стал теперь полностью «влезать» только на DVD. Сейчас примерно раз в один-два года выпускается новая версия дистрибутива. Текущая нумерация дистрибутива ведётся по номеру года. Дистрибутив идёт с подробной документацией по установке, в том числе и на русском языке.

ТЭХ Live — это свободный софт. Разработчики дистрибутива используют определение понятия свободы, принятое Фондом свободного программного обеспечения (<http://www.gnu.org/philosophy/free-sw.html>) или сообществом Debian

²R — язык программирования для статистической обработки данных.

(http://www.debian.org/social_contract#guidelines). В случае конфликтов этих определений, как правило, принимается сторона FSF.

Все пакеты, рассматриваемые в данной книге, за очень редким исключением присутствуют в дистрибутиве T_EX Live 2007. Но нет никакой необходимости использовать именно этот дистрибутив, хотя это и желательно. У пользователей MS Windows особой популярностью пользуется созданный именно для этой системы дистрибутив MiK_TE_X (<http://www.miktex.org/>), а пользователей Mac OS X может больше привлечь дистрибутив MacT_EX (<http://www.tug.org/mactex/>). Пользователям же Unix-подобных систем настоятельно рекомендуется при первом удобном случае перейти на T_EX Live, благо это не сложно.

1.3. Об этой книге

► *Утверждение:* Это свободная книга про свободные T_EXнологии!

В качестве основы для книги использовался цикл из одиннадцати статей по L_AT_EX, написанный мной для журнала Linux Format (<http://www.linuxformat.ru>) в 2006–2007 гг. Сами статьи можно найти в открытом доступе по адресу <http://www.inp.nsk.su/~baldin/LaTeX/>.

«Компьютерная типография L_AT_EX» в мягкой обложке была выпущена издательством «БХВ-Петербург» в 2008 году за номером ISBN 978-5-9775-0230-6. Когда по договору с издательством права на текст возвратились ко мне, я выложить его под свободной лицензией Creative Commons Attribution-ShareAlike 3.0 Unported (CC-BY-SA 3.0) на CTAN: http://ctan.org/tex-archive/info/russian/Computer_Typesetting_Using_LaTeX.

После этого «года не прошло», как у меня дошли руки и я выложил исходники книги в удобном для представления виде на Google Code под лицензией GPL v.3 or later (подробности в файле License): <https://code.google.com/p/ctex-ru/> или <https://ctex-ru.googlecode.com/> (проект ctex-ru).

Для копирования исходников на свой компьютер в целях их изучения и модификации достаточно выполнить команду:

```
> git clone https://code.google.com/p/ctex-ru/
```

Для этого у вас на компьютере должна быть установлена система контроля версий **git**. Подробнее про эту систему можно узнать в замечательной книге, которая носит весьма незатейливое название «Pro Git» <http://git-scm.com/book/ru>.

По любым вопросам связанным с книгой и исходниками следует обращаться ко мне через e-mail: E.M.Baldin@inp.nsk.su или E.M.Baldin@gmail.com. Все конструктивные предложения, критика и исправления принимаются с благодарностью!

1.3.1. Оформление ключевых слов и примеров

Названия пакетов, программ и пунктов меню выделяются жирным шрифтом, например, так: пакет **babel**.

Опции, имена и расширения файлов выделяются машинописным шрифтом, например, так: `lshort.pdf`. Если какой-то файл лежит на архиве CTAN, то перед его полным именем добавляется {CTAN}.

Значительные фрагменты кода выносятся из основного текста, например, так:

```
\begin{equation*}
\int\limits_{-\infty}^{\infty}
e^{-x^2/2\sigma^2}dx=\sqrt{2\pi}\sigma
\end{equation*}
```

Код может переноситься на другую страницу, разрывая рамку. Небольшие фрагменты могут выводиться и без рамки.

Если код соседствует с результатом, то он выводится машинописным шрифтом, а результат формируется справа от него, например, так:

<pre>\begin{equation*} \int\limits_{-\infty}^{\infty} e^{-x^2/2\sigma^2}dx=\sqrt{2\pi}\sigma \end{equation*}</pre>	$\int_{-\infty}^{\infty} e^{-x^2/2\sigma^2} dx = \sqrt{2\pi}\sigma$
--	---

1.3.2. Как читать эту книгу

Книга разбита на небольшие главы, в каждой из которых рассматривается определённый аспект Л^AT_EX. Текст можно читать просто подряд, так как по возможности каждая глава опирается на знания, полученные из предыдущего материала, или обращаться к конкретным главам, как к справочнику.

Если хочется просто найти готовое решение, то, возможно, следует обратиться к краткому (меньше никак не получается) каталогу пакетов T_EX Live на странице 247. В этом разделе пакеты распределены по категориям и те, что рассмотрены за пределами каталога подробнее, отмечены значком ✓.

Чтобы книгой было удобнее пользоваться в качестве именно справочной литературы, она содержит указатель, который разбит на две части:

- указатель команд и окружений,
- список пакетов.

► В списке пакетов курсивом выделены номера страниц, которые ссылаются на раздел «Каталог пакетов T_EX Live».

Одной из основных целей книги является демонстрация максимального числа возможностей $\text{\LaTeX}$, поэтому может показаться, что некоторые аспекты рассмотрены весьма поверхностно. Подробности в этом случае следует искать в документации представленных пакетов.

Знаком ► в начале параграфа отмечаются советы, выпадающие из общей канвы рассказа, но от этого не менее важные.

► Эта книга не является истиной в последней инстанции, и необходимость чтения документации она отнюдь не отменяет.

1.4. Литература

$\text{\LaTeX}$ ’у уже свыше двадцати лет. За время своего существования многие из его частей были существенно переделаны и усовершенствованы. База же, в виде $\text{\TeX}$, до сих пор остаётся стабильной основой. Видимо, поэтому документация к $\text{\LaTeX}$ устаревает с очень большой неохотой.

Доступных в России книг по $\text{\LaTeX}$ относительно немного. С другой стороны, если удастся добыть хоть одну из перечисленных в этом разделе книг, то для обычного набора её, скорее всего, будет достаточно. Логичная организация позволяет $\text{\LaTeX}$ расширяться, не сильно ломая совместимость с уже наработанными навыками.

► Выходные данные упомянутой в этом разделе литературы представлены в заключительной части книги.

Для более подробного ознакомления с конкретными пакетами следует обратиться к документации, поставляемой с пакетом. Чего-чего, а описаний в дистрибутиве $\text{\LaTeX}$ хватает. Также для поиска того или иного решения можно обратиться к сайтам CTAN (<http://www.ctan.org>) или TUG (<http://www.tug.org>).

1.4.1. Классика

Д. Э. Кнут «Всё про $\text{\TeX}$ » [1] — библия $\text{\TeX}$. Для обычного набора текстов в $\text{\LaTeX}$ информация, собранная в этой книге, не обязательна. $\text{\TeX}$ пертам и тем, кто таковыми себя считает, следует читать по несколько раз в обязательном порядке. Просто необходима для написания пакетов $\text{\LaTeX}$. Книга содержит сквозной разноуровневый по сложности материал. При прочтении можно выбирать свой уровень.

Д. Э. Кнут «Всё про METAFONT» [2] — всё, что сказано про библию $\text{\TeX}$, относится и к этой библии METAFONT. Перед созданием иллюстраций с использованием MetaPost, эту книгу следует прочитать внимательно или хотя бы просмотреть для общего развития.

Д. Э. Кнут «Компьютерная типография» [3] — сборник статей Д. Э. Кнута, написанных в процессе создания $\text{\TeX}$ и METAFONT. В книге подробно разобраны алгоритмы, которые легли в основу $\text{\TeX}$ и перечислены проблемы, которые необходимо было решить в процессе создания «компьютерной типографии». Книга

интересна и в историческом плане, как рассказ о становлении одного из самых успешных и долгоживущих компьютерных проектов.

1.4.2. Учебники и справочники

Г. Грэтцер «Первые шаги в $\text{\LaTeX}$ » [4] — новичкам посвящается. Компактный и элементарный учебник, позволяющий быстро освоить базовые приёмы. Упор на математику.

М. Гуссенс, Ф. Миттельбах и А. Самарин «Путеводитель по пакету $\text{\LaTeX}$ и его расширению $\text{\LaTeX 2}_\epsilon$ » [5] — действительно исчерпывающее полное справочное руководство по основным пакетам и приёмам $\text{\LaTeX}$. Эта книга является эталонным справочником по $\text{\LaTeX}$ во всём мире. На английском недавно вышло второе издание. Вероятно, со временем эта книга доберётся и до нас.

М. Гуссенс, С. Ратц и Ф. Миттельбах «Путеводитель по пакету $\text{\LaTeX}$ и его графическим расширениям» [6] — иллюстрирование документов при помощи $\text{\LaTeX}$, MetaPost и PostScript. При подготовке основного путеводителя авторы обнаружили, что описание систем для создания рисунков с использованием $\text{\TeX}$ технологий по объёму начинает превосходить базовый текст. Пришлось выделить для этого отдельную книгу. Здесь есть всё: от шахмат, нот и электронных схем до трюков с PostScript и MetaPost. Очень полезный учебник-справочник для тех, кто самостоятельно делает иллюстрации.

М. Гуссенс, С. Ратц «Путеводитель по пакету $\text{\LaTeX}$ и его Web-приложениям» [7] — использование документов $\text{\LaTeX}$ и его производных на просторах WWW. Лучше бы авторы описали ещё несколько пакетов $\text{\LaTeX}$. Эта книга, скорее всего, просто дань моде. С другой стороны расписано всё, что связано с PDF и что с ним можно сделать. Любителям XML посвящается.

«Не очень краткое введение $\text{\LaTeX 2}_\epsilon$. Или $\text{\LaTeX 2}_\epsilon$ за 94 минуты» в переводе Бориса Тоботраса от 1999 г. Электронная версия и исходники доступны на домашней страничке переводчика: <http://xtalk.msk.su/tex/>. В названии всё сказано. Самый доступный и маленький из имеющихся на сегодня русскоязычных учебников. Из недостатков: отсутствует информация о кириллизации. Полезно для начального изучения.

1.4.3. $\text{\LaTeX}$ в России

С. М. Львовский «Набор и вёрстка в системе $\text{\LaTeX}$ » 3-е издание [8] — классика жанра. Хороший переплёт. Основной упор на математику. Автор ориентируется на нестандартную русификацию, которая имеет свои преимущества, хотя и спорные. Есть авторская электронная версия, которую можно взять, например, здесь: <http://www.mccme.ru/free-books/>.

И. А. Котельников, П. З. Чеботаев « $\text{\LaTeX}$ по-русски» 3-е издание [9] — очень качественный учебник. К сожалению, качество переплёта не очень высокое, что не позволяет активно работать с этой книгой, по крайней мере долго. Третье изда-

ние является последним в этой серии, и четвёртого не будет. Электронная версия книги доступна по адресу: <http://www.tutor.nsu.ru/books/tex/>.

А. И. Рожено «Искусство вёрстки в ЛАТ_ΕX'e» [10] — краткий и достаточно исчерпывающий справочник с упором на разработанные автором макросы $\mathcal{NCS}$. Обсуждаются особенности русского стиля в книгопечатании.

Глава 2

Запускаем L^AT_EX

Лучший способ в чём-то разобраться до конца — попробовать научиться этому компьютер.

Дональд Э. Кнут

Предполагается, что дистрибутив L^AT_EX на компьютере уже установлен и настроен. Если это не так, то следует потревожить администратора или суперпользователя по этому поводу. Обычно проблем с установкой не возникает. Возможные шероховатости с русским языком в современных дистрибутивах возникают по недоразумению.

По традиции для начала скажем: «Здравствуй, мир!!!» Для этого в любом удобном для вас текстовом редакторе¹ нужно создать файл `helloworld.tex`, примерно следующего содержания:

```
% Выбор класса документа
\documentclass{article}
% Чтобы можно было использовать русские буквы в формулах,
% но в случае использования предупреждать об этом
\usepackage[warn]{mathtext}
% Выбор внутренней TEX-кодировки
\usepackage[T2A]{fontenc}
% Выбор кодовой страницы документа
% Так же можно выбрать cp1251 или utf8
\usepackage[koi8-r]{inputenc}
% Выбор языка документа
\usepackage[english,russian]{babel}
```

¹Лучше, чтобы этот текстовый редактор оказался **emacs** — в каждой шутке есть только доля шутки ☺.

```
% Начинать первый параграф раздела следует с красной строки
\usepackage{indentfirst}
% Конец преамбулы и начало текста
\begin{document}
\LARGE Здравствуй, мир!!!
\end{document}
```

Комментарии, которые начинаются со знака %, можно опускать. Всё, что идёт до `\begin{document}`, называется преамбулой или «шапкой». Преамбула определяет вид итогового документа. Нет необходимости каждый раз набивать эти строчки с нуля. Для этого достаточно обучить текстовый редактор вставлять их автоматически при создании нового `tex`-файла. После создания исходного текста его необходимо откомпилировать:

```
> latex helloworld.tex
This is pdfTeX, Version 3.141592-1.30.3-2.2 (Web2C 7.5.5)
 %&-line parsing enabled.
entering extended mode
(./helloworld.tex
LaTeX2e <2003/12/01>
```

...

```
(./helloworld.aux) [1] (./helloworld.aux) )
Output written on helloworld.dvi (1 page, 240 bytes).
Transcript written on helloworld.log.
>
```

В качестве результата L^AT_EX выдаёт файл `helloworld.dvi`. Далее есть выбор:

- посмотреть результат с помощью **xdvi**² или **dviout**³:


```
> xdvi helloworld.dvi
```
- преобразовать `dvi`-файл в `PostScript`⁴ и посмотреть его с помощью **gv** (для Unix) или **ghostview** (для Windows), а потом распечатать на стандартном `PostScript`-принтере (если он есть, естественно), например так:


```
> dvips helloworld.dvi
> gv helloworld.ps
> lpr helloworld.ps
```

²**xdvi** распространён в Unix-подобных операционных системах. В других операционных системах могут быть свои программы просмотра.

³**dviout** является для T_EX Live базовой программой просмотра `dvi` в среде MS Windows.

⁴`PostScript` — язык описания страниц, разработан Джоном Уорноком и Чаком Гешке из Adobe Systems. Интерпретаторы `PostScript` аппаратные или программные (`ghostscript`) широко используется при печати документов.

- сделать PDF⁵ и, естественно, просмотреть его с помощью **xpdf**, **kpdf** или Adobe Reader:

```
> dvips helloworld.dvi
> ps2pdf helloworld.ps helloworld.pdf
> kpdf helloworld.pdf
```

Во всех случаях на экране отобразится одно и то же:

Здравствуй, мир!!!

Рис. 2.1. «Здравствуй, мир!!!» от L^AT_EX

Если используется редактор, ориентированный на набор L^AT_EX-текстов, то, скорее всего, всё, что перечислялось выше, можно сделать из него через клавиатурные сокращения/меню/команды/кнопки.

2.1. Проблемы с компиляцией

Случается, что при наборе допускается ошибка, тогда при компиляции исходника L^AT_EX может затребовать дополнительную информацию.

Если просто запустить **latex** без каких либо инструкций, то на экране появится приглашение:

```
> latex
This is pdfTeX, Version 3.141592-1.30.3-2.2 (Web2C 7.5.5)
%&-line parsing enabled.
**
```

L^AT_EX ждёт ввода имени текстового файла, чтобы начать его обработку. Можно прервать ожидание по ^C (Ctrl+C).

Если же правильно задать файл при запуске **latex**, но при этом ошибиться в коде, то L^AT_EX выдаст сообщение об ошибке с номером строки, где возникла проблема, и предложит сделать выбор:

```
! Undefined control sequence.
l.11 \errorinbody
```

?

⁵PDF — Portable Document Format. Этот формат, как и PostScript, создан фирмой Adobe Systems. Является стандартом для электронной документации.

На запрос (?) можно ввести h, и тогда будет выдано предположение о том, с чем может быть связана ошибка.

? h

The control sequence at the end of the top line of your error message was never \def'ed. If you have misspelled it (e.g., '\hobx'), type 'I' and the correct spelling (e.g., 'I\hbox'). Otherwise just continue, and I'll forget about whatever was undefined.

? x

Для того чтобы прервать выполнение компиляции, нужно набрать x и перевод строки. В случае обычного перевода строки компиляция продолжится до следующей ошибки или до самого конца. Краткую информацию об управляющих командах можно получить, введя знак вопроса (?).

► *Добрый совет:* найденную ошибку следует немедленно ликвидировать. Следующие предупреждения об ошибке могут быть следствием той самой первой.

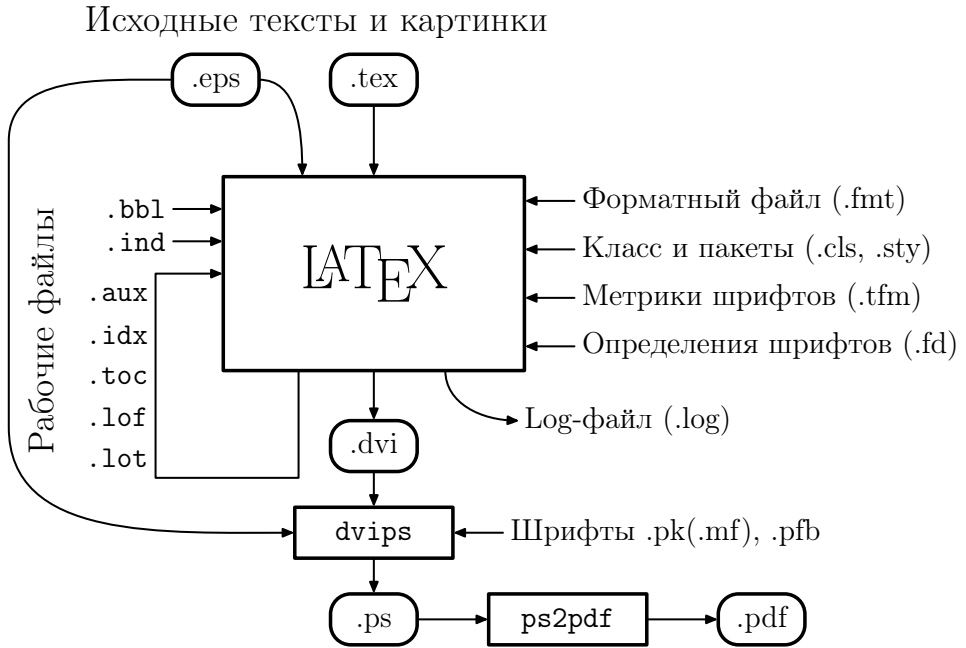
Интерактивный режим для работы с ошибками L^AT_EX — довольно мощный инструмент разбирательства, но на первых порах лучше следовать «Доброму совету». Текстовые редакторы, в которых предусмотрена поддержка редактирования исходников L^AT_EX, обычно на основании сообщения об ошибке позволяют её локализовать.

2.2. L^AT_EX-конвейер

В процессе работы L^AT_EX читает и записывает несколько файлов. Полезно знать, что это за файлы и зачем они нужны.

На вход подаётся текстовый файл с L^AT_EX-разметкой. Традиционно файл имеет расширение **tex**. В качестве результата на выходе получается файл с тем же самым именем, что и на входе, но с расширением **dvi**. Dvi — это device independent (не зависящий от устройства) формат, который хранит информацию о форматировании текста и размещении всех его элементов на странице, но без самих букв и картинок. Программы, преобразующие dvi-файл в другое представление, называются dvi-драйверами.

Программа просмотра **xdvi** преобразует dvi-файл в картинку на экране монитора. Это очень продвинутый драйвер. Он реалистично представляет вид напечатанной страницы, поддерживает гиперссылки и позволяет организовать обратную связь с текстом. Ещё одним популярным dvi-драйвером является **dvips**. **dvips** производит качественный PostScript, который уже можно передать принтеру на печать либо напрямую (если принтер поддерживает PostScript аппаратно), либо через программный интерпретатор Ghostscript. Существуют и другие dvi-

Рис. 2.2. L^AT_EX-конвейер

драйверы, например, **dvi2tty** пытается перевести dvi обратно в текст, **dvilj** переводит dvi в инструкции PCL для принтеров HP, **dvipdf** транслирует dvi в pdf. Обилие dvi-драйверов позволяет рассмотреть/напечатать dvi-файлы практически в любой ситуации.

Свободный программный интерпретатор Ghostscript (**gs**) в свою очередь позволяет преобразовывать PostScript-файлы (**ps**) в другие форматы. Обычно PDF получают именно из PostScript с помощью скрипта **ps2pdf**.

Графика в L^AT_EX, как правило, добавляется через eps-файлы. EPS, или Encapsulated PostScript, — это векторный графический формат, который представляет из себя инструкции на языке PostScript с некоторыми ограничениями. Одно из основных требований заключается в том, что в заголовке eps-файла обязательно должны быть указаны его размеры (BoundingBox). Так выглядит заголовок eps-файла, сделанного из примера «Здравствуй, мир!!!»:

```

%!PS-Adobe-2.0 EPSF-2.0
%%Creator: dvips(k) 5.95b Copyright 2005 Radical Eye Software
%%Title: helloworld.dvi
%%BoundingBox: 148 651 288 668
%%DocumentFonts: SFRM1728
  
```

Перечислим теперь остальные файлы, которые участвуют в этом круговороте:

I. Внешние файлы.

fmt — форматный файл. Содержит, главным образом, все команды L^AT_EX в предварительно откомпилированной форме. Также содержит информацию о переносах. При изменении значений глобальных переменных T_EX требуется пересборка форматного файла:

```
> texconfig init
```

cls, sty — определение макета и структуры документа. Класс (**cls**) документа выбирается с помощью инструкции `\documentclass`. Дополнительные возможности и изменение поведения класса по умолчанию достигаются посредством использования пакетов (**sty**), которые инициализируются командой `\usepackage`.

tfm — метрики шрифтов. Размеры и правила взаимодействия литер друг с другом.

fd — приведение внешних названий шрифтов к виду, принятому в L^AT_EX.

pfb — векторные Type1-шрифты.

pk (mf) — растровые pk-шрифты по мере необходимости создаются из векторных mf-шрифтов (METAFONT) с необходимым разрешением.

II. Файлы, создаваемые в процессе L^AT_EX-конвейера. Обычно для получения итоговой копии документа требуется несколько раз запускать **latex**. При первом проходе записывается некоторое число информационных файлов, которые при последующих проходах используются для нумерации ссылок, составления оглавления и тому подобного.

log — файл протокола. В этот файл «скидывается» вся информация, имеющая отношение к компиляции. Фактически дублирует стандартный вывод на экран.

aux — информация о перекрёстных ссылках.

toc — файл оглавления (table of contents).

lof — список иллюстраций (list of figures).

lot — список таблиц (list of tables).

bbl — список литературы, сформированный с помощью программы BibT_EX. На первых порах можно обойтись без этой T_EXнологии, но для серьёзных проектов управление списком литературы становится не менее серьёзной задачей.

ind — предметный указатель, сформированный программой MakeIndex или **xindy**. В каждой «уважающей себя» и несущей полезную информацию книге есть такой. Для составления используются idx-файлы. Организация указателя — это отдельная весьма нетривиальная задача.

Ранее был описан «классический» L^AT_EX-конвейер. Программа **latex** может быть заменена на **pdflatex**, тогда на выходе сразу будет получаться pdf-файл, а графическая информация должна быть представлена в форматах png⁶ или pdf. Возможны и другие вариации, но в целом структура остаётся той же.

⁶PNG (Portable Network Graphics) — растровый формат хранения графической информации, использующий сжатие без потерь.

Глава 3

Базовые элементы

Не надо пытаться *приблизиться* к лучшим книгам,
надо пройти весь путь до конца и сделать эти книги!

Дональд Э. Кнут

Самое главное — это сам текст, но для его адекватного представления необходимо немного потрудиться. Надо дать правильные команды компьютеру.

3.1. «Командная логика»

Для набора кода в ЛАТ_ЭX знание английского языка приходится очень кстати. Основных команд немного, и их можно запомнить и так, но для совершенствования английский необходим, хотя бы для чтения документации к пакетам. Названия у команд, как правило, вполне осмыслены, что очень помогает при поиске необходимой информации в алфавитном указателе. «Правильный» текстовый редактор тоже не является лишним.

Спецсимволы. Не все символы равноправны. За частью символов в ЛАТ_ЭX зарезервированы специальные значения.

Это: «\», «\$», «%», «_», «{», «}», «&», «#», «^» и «~». В процессе изложения их роль будет со временем раскрыта. Набиралось это так:

Это: <<\textbackslash>>, <<\\$>>, <<\%>>, <<_>>¹, <<\{>>, <<\}>>, <<\&>>, <<\#>>, <<\^{}>> и <<\~{}>>. В процессе изложения их роль будет раскрыта. Набиралось это так:

¹Можно воспользоваться пакетом **underscore** — в этом случае необходимость экранировать знак «_» в текстовой моде отпадает.

Группировка. Группировка множества символов осуществляется с помощью фигурных скобок: { группа }. Фигурные скобки при печати не отображаются.

Сложные конструкции, которые имеют открывающую и закрывающую команды, тоже группируют текст.

Построение команды. Команды в ЛАТ_ΕX начинаются с символа «\» (backslash или обратная косая черта) и продолжаются комбинацией, состоящей только из латинских букв. Команды завершаются пробелом, цифрой или не латинской буквой. Все пробельные символы после команды игнорируются. Для того чтобы пробел после команды не игнорировался, достаточно вставить «пустую группу»: \command{ }. Например, чтобы лого ЛАТ_ΕX (команда \LaTeX) не слилось со следующим за ним словом, следует написать \LaTeX{ }.

Аргументы. Командам ЛАТ_ΕX могут передаваться внешние аргументы:

```
\command[param 1][param 2]{param3}{param4}
```

В квадратные скобки заключаются необязательные параметры (*param1* и *param2*), а в фигурные — обязательные (*param3* и *param4*).

Некоторые команды ЛАТ_ΕX влияют только на свои аргументы. Например, команда \textbf{текст} печатает «текст» жирным шрифтом.

► Отдельные команды ЛАТ_ΕX имеют *перемещаемые параметры*, содержимое которых используется для оформления колонтитулов или заносится во внешние файлы для формирования разного рода списков и поддержания ссылочного механизма. Примерами таких команд являются все команды рубрикации: \chapter, \section, \subsection и так далее.

Некоторые команды ЛАТ_ΕX нельзя использовать в перемещаемых параметрах — такие команды называются *хрупкими*. Большинство команд с необязательными аргументами являются хрупкими. Если же всё-таки использовать такую команду в перемещаемом аргументе необходимо, то непосредственно перед ней при помещении в аргумент нужно добавить команду \protect. Например, так:

```
\subsection{Заголовок\protect\footnote{Сноска}}
```

Декларативные команды. Часть команд ЛАТ_ΕX являются своеобразными переключателями режимов.

Область действия декларативной команды может ограничиваться логической группой или единицей структуры печатного документа, например, страницей. Примеры декларативных команд:

```
% Дальнейший текст будет печататься жирным шрифтом
\bfseries
% Убрать заголовки и нумерацию для текущей страницы
\thispagestyle{empty}
```

Окружения. Сложные конструкции, которые имеют открывающую и закрывающую команды вида `\begin{имя}` и `\end{имя}`, называют окружениями. Вместо слова «имя» подставляется название соответствующего окружения:

```
\begin{center}
Эта строка будет центрирована
\end{center}
```

Окружения могут вкладываться друг в друга, как матрёшки, но их области действия не могут перекрываться.

3.2. Логика документа

Вид документа определяется выбором класса и стилей. Хорошим приёмом является наличие личных готовых шаблонов с уже предопределёнными предпочтениями, которые могут меняться по мере развития документа.

3.2.1. Структура $\text{\LaTeX}$ -файла

Текстовый `tex`-файл состоит из двух частей: преамбулы и, собственно, самого текста. Выглядит он примерно следующим образом:

```
%—начало преамбулы—
% Выбор класса документа, например, article или book
\documentclass[a4paper,12pt,oneside]{scrartcl}
% Минимальная кириллизация — кодовая страница документа
\usepackage[koi8-r]{inputenc}
% Локализация и переносы
\usepackage[english,russian]{babel}
\usepackage[indentfirst}
\usepackage{miscorr}
% Загрузка пакетов по выбору
\usepackage{...}

...
% Определение новых или переопределение старых команд
\newcommand{\mycommand}{...}
\renewcommand{\oldcommand}{...}

...
% Локальные настройки
...
%—конец преамбулы—
\begin{document}
% Тело документа
\end{document}
```

Первым делом с помощью инструкции `\documentclass` осуществляется выбор класса документа. Далее с помощью команд `\usepackage` загружаются стилевые пакеты.

Для того чтобы можно было набирать русский текст, необходимо с помощью пакета **inputenc** указать кодировку текстового файла, например, `koi8-r`², `cp866`³, `cp1251`⁴ или `utf8`⁵ (Unicode).

Далее нужно подключить пакет **babel**, который отвечает за локализацию, в частности за настройку переносов и «национальные особенности» набора. Например, при включении русского языка доопределяется символ номера № (`\No`), символ параграфа § (`\S`) и многое другое.

Для формирования отступа или красной строки у первого параграфа, как это принято в России, необходимо загрузить пакет **indentfirst**. По идее, это должно относиться к «национальным особенностям», но в **babel** по умолчанию не подгружается. Загрузка стиля **misccorr** также исправляет ещё некоторое количество недоработок **babel**.

К вопросу о переносах. Пожалуй, единственное, что может потребовать настройку в свежеставленном дистрибутиве ЛАТЭХ— это включение переносов. Данная проблема заведомо не возникает при установке ТЭХ Live и в большинстве современных дистрибутивах, но всякое бывает.

Для установки переносов необходимо отредактировать файл **language.dat** и регенерировать все имеющиеся форматные файлы (fmt-файлы). Это проще всего сделать через стандартные утилиты настройки, например, в МикТЭХ это делается с помощью программы «MiKTeX Options»⁶, а в среде GNU/Linux можно воспользоваться утилитой **texconfig**⁷. После запуска программы следует выбрать меню **HYPHENATION**, а затем меню **latex**. Далее будет предложено отредактировать⁸ файл переносов **language.dat**. Обычно достаточно таких настроек:

²koi8-r — русская 8-битная кодировка традиционно используемая в UNIX-подобных операционных системах. Имеет свой RFC №1489.

³cp866 или альтернативная кодировка — русская 8-битная кодировка, используемая для набора текстов в DOS и OS/2.

⁴cp1251 или Windows-1251 — стандартная 8-битная кодировка для всех русских версий Microsoft Windows.

⁵UTF-8 (Unicode Transformation Format) — распространённая кодировка, реализующая представление Юникода.

⁶Выбор таблиц переносов осуществляется на закладке **Languages**. После этого на закладке **General** следует нажать кнопку **Update Now** для регенерации форматных файлов.

⁷**texconfig** — это простенькое dialog-подобное консольное приложение. Некоторые настройки могут потребовать привилегии суперпользователя. После изменения настроек через **texconfig** автоматически регенерируются необходимые форматные файлы. В противном случае может потребоваться сделать это вручную, например, с помощью инструкции вида: **texconfig init**.

⁸Редактор можно определить с помощью переменной окружения `$EDITOR`. Если переменная не определена, то вызывается редактор **vi**. В случае отсутствия опыта работы с **vi**, следует выйти из него с помощью последовательности `:q` и настроить переменную окружения на известный редактор или разобраться с **vi**.


```
english hyphen.tex  
russian ruhyphen.tex
```

Всё остальное по желанию можно закомментировать. За переносы отвечает пакет **babel**. По умолчанию, когда включаются переносы для определённого языка, все остальные правила переносов отключаются. Но в случае английского и русского языков это можно обойти, воспользовавшись русско-английской таблицей переносов:

```
ruseng ruenhyph.tex  
=russian  
=english
```

Следует понимать, что подобная настройка с точки зрения философии ЛАТЭХ не совсем корректна. Для гарантированно одинакового результата компиляции независимо от платформы лучше поступиться некоторыми удобствами.

3.2.2. Класс документа

С помощью обязательной инструкции `\documentclass[опции]{класс}` указывается, к какому классу будет относиться рабочий текст.

Класс документа следует выбирать в зависимости от того, что должен из себя представлять результирующий документ. Для начала, в принципе, можно остановиться на стандартном классе **article**. Этот класс разработан специально для статей и небольших отчётов. Для отчётов побольше можно использовать класс **report**, а для книг — класс **book**.

Перечисленные стандартные классы сложились очень давно, и многие производные классы документов основаны на них. Как следствие стандартные базовые классы абсолютно статичны и имеют массу недостатков, которые из-за требований совместимости исправить не получается.

Для специализированных задач используются свои классы. Например, для научных статей популярны различные модификации **revtex4**. Константин Корилов разработал и активно поддерживает класс **eskd** (стандарт ЕСКД), который можно найти на CTAN или в стандартной поставке TeX Live. Те, кого «напрягают» большие поля в стандартных ЛАТЭХ-классах, могут обратить внимание на набор классов **KOMA-Script**: **scrartcl**, **scrreprt** и **scrbook** вместо **article**, **report** и **book** соответственно.

По началу в выбранном классе ничего менять не следует. То, что кажется с непривычки неудобным, на самом деле может улучшить восприятие печатной копии. Например, относительно узкая ширина текста в стандартных классах (следствие больших полей) позволяет при прочтении охватывать взглядом всю строку целиком, что немного увеличивает скорость чтения. А сами большие поля можно использовать для заметок.

Обычно разумные модификации можно выбирать с помощью передачи параметров при выборе класса, например, так:

```
\documentclass[a4paper,12pt,oneside]{scrbook}
```

где **a4paper** — размер листа бумаги (можно выбрать другой стандартный размер, например, **a5paper**), **12pt** — базовый размер шрифта (в стандартных классах доступны размеры в **10pt** и **11pt**), а **oneside** — односторонняя печать (удобнее при просмотре электронной версии).

В заключении этого раздела хотелось бы отметить пакет *ℒCℒ*, который активно разрабатывается А. И. Рожено. Класс **ncc** позиционируется автором как «русскоязычная статья». Класс можно взять на CTAN, он также присутствует в стандартной поставке T_EX Live.

3.2.3. Стили

Стилевой файл (**sty**) или пакет представляет собой набор макросов и определений, созданных для решения какой-то определённой задачи. Для подключения стилового файла используется команда `\usepackage[опции]{стиль}`.

Основное отличие классов от стилей в том, что на документ может быть ровно один класс и сколько угодно стилей. Фактически на любую задачу в T_EX находится ответ в виде соответствующего пакета. В стандартной поставке T_EX Live присутствует свыше двух тысяч **sty**-файлов. Кроме того, ничего не мешает создать свой собственный стиль, предназначенный для решения своих локальных проблем. Хотя всегда лучше воспользоваться *уже* готовыми решениями.

К вопросу о русификации. Чтобы кириллизировать T_EX, необходимы шрифты. Благодаря Ольге Лапко на свете есть шрифты семейства **lh**, которые отлично согласуются с базовыми шрифтами Computer Modern. Мало иметь просто кириллические буквы — надо, чтобы их начертания соответствовали и другим шрифтам, в том числе и математическим. В 2001 году Владимир Волович проделал огромную работу по переводу METAFONT-шрифтов в формат Type1. Благодаря ему теперь можно создавать не только хорошие печатные копии, но и вполне качественные электронные pdf-версии документов.

За перевод из кодировки пользователя во внутреннюю кодировку T_EX отвечает пакет **inputenc**. В качестве опции при загрузке с ним передаётся текущая восьмибитная кодовая страница документа. Для русскоязычной кириллицы могут оказаться интересны следующие варианты: **ko18-r**, **cp866**, **cp1251** и **utf8**. Собственно говоря, всё. Единственное неудобство, возникающее из-за этого, заключается в том, что сообщения об ошибке T_EX выдаёт в своей внутренней T2A-кодировке⁹. Для исправления данного неудобства можно воспользоваться простейшим фильтром **t2filter**. Для начала его надо собрать.

⁹Расположение букв похоже на расположение букв в **cp1251**, но полностью не совпадает — чистая случайность.

Примерный алгоритм поиска и сборки фильтра:

```
> locate t2filter.c
{TEXMF}/texmf-dist/doc/generic/t2/etc/t2filter.c
> cd {TEXMF}/texmf-dist/doc/generic/t2/etc/
> gcc -Wall -O2 -s -o ~/bin/t2filter t2filter.c
> latex {файл}.tex | t2filter
```

3.2.4. Тело документа

Всё, что заключено внутри окружения `document`, является телом документа. Если у вас есть какие-то куски текста, которые печатать не хочется, а выкинуть жалко, то их достаточно вынести в самый конец за инструкцию `\end{document}`.

3.3. Логика набора

Мало открыть файл в текстовом редакторе и начать набирать. Нажимать на клавиши надо осмысленно.

3.3.1. Печатаем текст

При наборе книги/статьи/заметки основное — вовсе не команды, а сам текст. Правила очень просты.

Комментарии. Всё, что следует за знаком «`%`» включительно, является комментарием.

Большие закомментированные сегменты мешают работать с основным текстом, и поэтому их следует исключать из рабочего файла. Но при желании можно воспользоваться окружением `comment` из пакета **verbatim**.

Разделение слов. Пробельные символы используются в $\text{\LaTeX}$ для разделения слов. Пробелы в начале строки игнорируются. Символ перевода строки также воспринимается как пробел. Если в конце строки сразу за последним словом вставить знак комментария:

```
экранировка перевода стр%
оки
```

то разделения слов не происходит — экранировка пробелов. Иногда этот приём может оказаться полезным.

Разделение абзацев. Для того чтобы начать следующий абзац, необходимо оставить пустую строку:

```
текущий абзац закончился

следующий абзац начался\par
```

Число пустых строк между абзацами не имеет значения. Пустая строка эквивалентна команде `\par`:

```
текущий абзац закончился\par
следующий абзац начался
```

Выделение текста. Самый простой способ выделения текста — это смена насыщенности или начертания шрифта. Как правило, для выделения конкретных слов и фраз достаточно команды `\emph`, но возможны и другие варианты:

```
\emph{Внимание!} \textbf{жирный шрифт},      Внимание!: жирный шрифт,
\textit{курсив}, \textsc{Капитель},           курсив, КАПИТЕЛЬ, наклонный
\textsl{наклонный шрифт} и обычный текст.    шрифт и обычный текст.
```

Для каждой команды выделения текста есть декларативные альтернативы: `\bfseries` — **жирный текст**, `\itshape` — *курсив*, `\scshape` — **КАПИТЕЛЬ** и `\slshape` — *наклонный шрифт*.

Для получения жирных *курсива*, **КАПИТЕЛЯ** и **наклонного** текстов инструкции смены начертания и насыщенности можно комбинировать, например:

```
\textbf{\textit{жирный курсив}}\par           жирный курсив
\textbf{\textmd{\textit{просто курсив}}}\par  просто курсив
```

В этом примере команда `\textmd` задаёт среднюю, то есть обычную, насыщенность текста, как бы отменяя действие инструкции `\textbf`.

Результат воздействия команды `\emph` зависит от текущего начертания:

```
{\itshape курсивная среда \emph{выделение}}  курсивная среда выделение
```

Ещё один способ визуального разделения текста — изменение размера шрифта. Для смены размера используются декларативные команды. Базовому размеру шрифта обычного текста соответствует декларация `\normalsize`. В таблице 3.1 представлен список команд переключения размеров.

► По умолчанию для набора текста используется прямой шрифт средней насыщенности семейства Roman. При желании можно заказать рубленый шрифт (Sans Serif) с помощью команды `\textsf` или машинописный (Typewriter) с помощью команды `\texttt`.

Декларация	Строчный образец	Заглавный образец
<code>\tiny</code>	образец	ОБРАЗЕЦ
<code>\scriptsize</code>	образец	ОБРАЗЕЦ
<code>\footnotesize</code>	образец	ОБРАЗЕЦ
<code>\small</code>	образец	ОБРАЗЕЦ
<code>\normalsize</code>	образец	ОБРАЗЕЦ
<code>\large</code>	образец	ОБРАЗЕЦ
<code>\Large</code>	образец	ОБРАЗЕЦ
<code>\LARGE</code>	образец	ОБРАЗЕЦ
<code>\huge</code>	образец	ОБРАЗЕЦ
<code>\Huge</code>	образец	ОБРАЗЕЦ

Таблица 3.1. Команды переключения размеров шрифта

Перечни. Для создания многоабзадных нумерованных перечислений используется окружения `enumerate`, а для ненумерованных `itemize`. Перечни можно вкладывать друг в друга для получения многоуровневых перечислений.

Перечни по своей структуре бывают:

```
\begin{enumerate}
  \item внутриабзачными:
    \begin{inparaenum}
      \item раз, \item два, \item три.
    \end{inparaenum}
  \item многоабзачными. \par
  В свою очередь многоабзачный
  перечень может быть:
  \begin{itemize}
    \item нумерованным,
    \item ненумерованным.
  \end{itemize}
\end{enumerate}
```

Перечни по своей структуре бывают:

- 1) внутриабзачными: а) раз, б) два, в) три.
- 2) многоабзачными.
В свою очередь многоабзачный перечень может быть:
 - нумерованным,
 - ненумерованным.

Для разделения пунктов перечней используется декларация `\item`, которая формирует правильный отступ и ставит соответствующую метку.

Внутриабзачные перечисления реализуются с помощью окружений `inparaenum` и `inparaitem` из пакета **paralist** (подробности на стр. 133).

3.3.2. Пунктуация

Напечатанный текст обезличивается. Нет эмоций — только буквы. Единственное, что остаётся — это знаки пунктуации и, возможно, смайлики ☺.

Запятую, точку, точку с запятой, двоеточие, многоточие, скобки, кавычки, восклицательный и вопросительные знаки следует «прижимать» к словам. Не надо оставлять пробелов, а то $\text{\LaTeX}$ «подумает», что так и надо.

Пробелы. Расстояние между словами $\text{\LaTeX}$ выбирает по своему усмотрению для максимально равномерного заполнения абзаца. Но иногда необходимо указать размер пробела руками:

- $\sim$ — неразрывный пробел, то есть по этому пробелу не производится перенос предложения на другую строку;
- $\backslash,$ — маленький нерастяжимый пробел;
- $\backslash_$ — нормальный пробел.

В основном, указывать размеры пробелов надо в случае набора каких-либо сокращений, например, так следует набирать ФИО: Ф.\,А.\,Миля "----негоже <<отрывать>> ИО от Ф\@. Ещё примеры: т.\,е. (не т.е.), г.\,Новосибирск, рис.\,1 и т.\,п.

В основном, указывать размеры пробелов надо в случае набора каких-либо сокращений, например, так следует набирать ФИО: Ф. А. Миля — негоже «отрывать» ИО от Ф. Ещё примеры: т.е. (не т.е.), г. Новосибирск, рис. 1 и т.д. и т.п.

$\text{\LaTeX}$ считает, что после точки предложение заканчивается, если эта точка стоит не после заглавной буквы. Растяжимость пробелов между предложениями и между словами существенно разная. Поэтому если точка случается в середине предложения, то после неё следует явно вставить пробел « $\backslash_$ » или неразрывный пробел « $\sim$ ». Может случиться так, что точка следует сразу за заглавной буквой (как в примере происходит с буквой Ф), но означает именно конец предложения. Для этого перед такой точкой следует добавить коррекцию в виде команды « $\backslash@$ ».

Если часть слова набрана наклонным шрифтом, а часть прямым, то необходимо вставить специальный корректирующий пробел $\backslash/$

Сравните: $\{\backslash\text{\LARGE}\{\text{\itshape}\Gamma\}\text{рош}\}$ и $\{\backslash\text{\LARGE}\{\text{\itshape}\Gamma\}\backslash/\text{рош}\}$.

Сравните: Γ рош и Γ рош.

Такое ухищрение нужно только в случае, если выделение текста осуществляется с помощью декларативных команд. Если используются команды, в которых текст передаётся в качестве аргумента, то корректирующий пробел вставляется автоматически.

Дефисы, минусы и тире. В издательских системах, основанных на $\text{\TeX}$, различают дефис «-» (hyphen), короткое тире «—» (en-dash), длинное тире «—» (em-dash) и знак минуса «-».

Чтобы получить на печати дефис, короткое или длинное тире, надо набрать один, два или три знака «-» соответственно.

При подключении пакета **babel** с опцией **russian** появляются дополнительные команды, позволяющие более строго следовать русским печатным традициям:

Дефис используют в составных словах (кто-то, где-нибудь), короткое тире рекомендуется для указания диапазона чисел (10—15, 2001—2006), длинное тире означает обычное тире (`\LaTeX{}` "— это круто), минус может существовать только в формулах ($a-b=c$).

Между собственными именами следует использовать команду `"--~`, например, уравнение Клапейрона`"--~`Менделеева. Также тире ставится между словами для обозначения пределов, например, поезд Москва`"--~`Новосибирск или отпуска в июле`"--~`августе. Чтобы длинное тире не отрывалось от предыдущего слова и вокруг него создавались правильные пробелы вместо `—` следует употреблять `"--`, т.е. к трём тире надо добавить двойную кавычку. Прямая речь должна начинаться с команды `"--*`:

`"--* Я сказал.`

Правила могут показаться немного запутанными, но к ним быстро привыкаешь, и они того стоят.

Переносы. Как правило, $\LaTeX$ грамотно переносит слова. Но в случае сложных слов, которые пишутся через дефис, перенос происходит только по дефису. Аналогично, проблемы возникают, когда слово частично состоит из английских букв, а частично из кириллицы.

Прямо в тексте перенос можно указать с помощью команды `\-`, например: дель`\-`та-функ`\-`ция, `\TeX`но`\-`ло`\-`гия.

При наличии русского языка в `\textbf{babel}` вместо дефиса в сложном слове можно поставить команду `"=`, например, дельта`"=`функция. В этом случае переносы будут сделаны корректно без подсказки.

Для часто упоминаемых слов можно задать шаблон переносов с помощью команды `\hyphenation{образ-цы пе-ре-но-са дель-та=-функ-ция}`. Как правило, образцы переноса лучше определять в заголовке документа. Следует понимать, что образцы автоматически не склоняются, поэтому надо предусмотреть всевозможные варианты окончаний.

С помощью команды `\hyphenation` можно запретить перенос слова в нежелательных местах, просто не указав место разрыва. В тексте запрет переноса можно оформить с помощью инструкции `\mbox{эта фраза не переносится}`.

Многоточие. Многоточие печатается с помощью команды `\ldots`. Если многоточие идёт после точки, то необходимо вставить неразрывный пробел `~` (знак тильды).

Ударение. В русском языке длительность ударного гласного примерно в 1.5–2 раза длиннее безударного. Если ударение поставить не в том месте, то слово будет звучать совсем по другому.

В корне `\textbf{зар-}` — `\textbf{зор-}` под ударением пишется гласная в соответствии с произношением, без ударения "—— `\textbf{a}`: `з\textbf{'a}рево`, `з\textbf{'o}рька` "—— `з\textbf{a}рн\textbf{'и}ща`, `оз\textbf{a}р\textbf{'я}ть`

Исключения: `з\textbf{o}р\textbf{'я}нка`, `з\textbf{o}р\textbf{ев\textbf{'а}ть}`.

„Лапки“ и «Ёлочки». В пакете `babel`, кроме всего прочего, определены традиционные русские кавычки, а точнее: „лапки“ (немецкие кавычки) и «ёлочки» (кавычки французские):

Если в начале или в конце текста встречаются внутренние и внешние кавычки, то они должны различаться между собой рисунком.

Он сказал: `<<А пойду—ка я и подпишусь на „Linux Format“>>`.

3.4. Структурная логика

Л^AT_EX ориентирован на логическую разметку документа. Можно, конечно, «сказать», что данный кусок текста следует напечатать размером 20 пунктов, выровнять по левому краю и сделать отступ после него в полтора интервала, но проще указать, что это заголовок раздела.

3.4.1. Титульный лист

Создание титульного листа — это отдельная задача, в которой визуальная составляющая обычно превалирует над структурной. В этом случае следует воспользоваться окружением `titlepage`. При инициализации этого окружения создаётся чистая страница, которой присваивается номер один, а содержание этой

страницы полностью определяется фантазией автора. Но в любом случае это следует делать уже после написания самого текста. Как правило, в начале достаточно стандартного заголовка:

```
\title{\LaTeX, Unix и русский стиль}
\author{Е. \,М.~Балдин\thanks{e-mail: E.M.Baldin@inp.nsk.su}}
\date{2002}
\maketitle
```

```
% Только для производных класса article
```

```
\begin{abstract}
```

В статье говорится про \LaTeX, Unix и русский стиль.

```
\end{abstract}
```

Команда `\maketitle` создаёт стандартный титульный заголовок, используя информацию о названии документа (`\title`), авторе (`\author`) и дате написания текста (`\date`). Команда `\thanks` правильным образом позволяет оформить подстрочное примечание на титульной странице. Если авторов более чем один, то их можно перечислять, разделяя командой `\and` — в этом случае список авторов печатается в виде таблицы.

В статьях (производные от класса **article**) вслед за заголовком следует обязательная аннотация, которая оформляется с помощью окружения `abstract`.

В книгах (производные от класса **book**) определено упомянутое в начале раздела окружение `titlepage`, которое специально предназначено для оформления титульного листа. Всё, что имеет отношение к оформлению титульного листа, следует помещать внутри этого окружения.

3.4.2. Секционирование

Часто бывает полезно сразу за титульной страницей вывести оглавление с помощью команды `\tableofcontents`, при этом не требуется никаких дополнительных действий. Пример такого оглавления можно увидеть в начале этой книги.

Оглавление создаётся автоматически только благодаря тому, что в тексте присутствует логическое разбиение на разделы. Естественно, всё можно сделать руками с помощью визуальной разметки, но тогда придётся забыть об автоматизации, например, при создании того же оглавления.

Так начинается *этот* раздел:

```
\subsection{Секционирование}
\label{sec:base:sec}
```

Часто бывает полезно ...

Команды секционирования образуют строгую иерархию. Самыми старшими в этой иерархии по «званию» являются разделы `\part{Часть}` и `\chapter{Глава}`.

Это очень большие куски текста и соответственно их применение обосновано только в книгах, поэтому они не определены в классах **article** (**scrartcl**) и **report**, зато определены в классе **book** (**scrbook**).

Далее по старшинству следуют:

```
\section{Раздел}
\label{section}

\subsection{Подраздел}
\label{subsection}

\subsubsection[Подподраздел]{Что-то более «мелкое», чем подраздел}
\label{subsubsection}

\paragraph{Параграф}
\label{paragraph}

\subparagraph{Подпараграф}
\label{subparagraph}
```

Если воспользоваться необязательным параметром команды секционирования, то он заменяет основной заголовок при печати оглавления и создании колонтитулов.

Команды секционирования печатают заголовок необходимым шрифтом и нумеруют раздел. Если нет желания, чтобы название раздела попало в оглавление, и не нужна нумерация раздела, то к команде секционирования следует добавить символ `**`, например, `\section*{Приложение}`.

3.4.3. Перекрёстные ссылки

Одной из основных причин, по которой ЛАТЭХ вытеснил обычный ТЭХ из текстовых редакторов и умов ТЭХников, является механизм нумерации и создания ссылок.

Чтобы сослаться на раздел, в нём необходимо оставить метку `\label{метка}`. Метка представляет любую комбинацию латинских букв, цифр и некоторых знаков препинания. В частности при составлении меток удобно использовать двоеточие (`:`) и тире (`-`).

Сразу после определения заголовка этого раздела была поставлена уникальная метка `\label{sec:base:ref}`. Поэтому теперь можно на этот раздел сослаться:

Раздел <<Перекрёстные ссылки>> имеет номер~\ref{sec:base:ref} и находится на странице~\pageref{sec:base:ref}.

Раздел «Перекрёстные ссылки» имеет номер 3.4.3 и находится на странице 42.

Когда ссылки идут через метку, то номер раздела (команда `\ref`) и номер страницы (команда `\pageref`) определяется L^AT_EX автоматически. Причём автоматическая нумерация свойственна не только командам секционирования. Точно так же можно ссылаться на формулы, таблицы, картинки, пункты перечня и листинги программ. Для этого достаточно просто добавить метку `\label` после команды именования соответствующего окружения. В случае таблиц или картинок такой командой является `\caption`, а в случае пунктов перечня — `\item`.

Глава 4

Начала математики

Полиграфисты относят математические работы к каторжным. . .

Дональд Э. Кнут

Иногда от незнакомых с Т_ЕХнологиями людей приходится слышать, что Л^AT_EX годится *только* для набора математики. При знакомстве же с истинными Т_ЕХнологиями возникает понимание, что Л^AT_EX настолько хорош, что с его помощью можно набирать *даже* математику.

Набор математики всегда считался вершиной типографского искусства. Дело в том, что формулы для концентрации информации и дополнительной выразительности в отличие от обычного текста являются многоуровневыми. Д. Э. Кнут к своей программе компьютерной типографии создал язык для описания формул. После короткого периода обучения пользователь будет в состоянии читать и набирать формулы на этом языке фактически независимо от их сложности.

Л^AT_EX не единственная программная среда, использующая Т_ЕХ-нотацию. Эта же нотация рекомендуется при наборе формул на страницах Википедии (статья «Википедия:Формулы» на сайте <http://ru.wikipedia.org>).

Становлению Т_ЕХ как стандарта для набора формул в значительной степени поспособствовало Американское математическое сообщество (The American Mathematical Society или AMS), которое в начале восьмидесятых годов прошлого столетия инициализировало и субсидировало разработку расширения Т_ЕХ, известного как $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_ЕХ. В 1987 году наработки $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_ЕХ были добавлены в Л^AT_EX в виде пакета **amsmath**.

Вместе с **amsmath** в Л^AT_EX было добавлено множество улучшений, позволяющих набирать действительно изощрённую математику. Поэтому при использовании в тексте математики в преамбуле документа следует в обязательном порядке загружать пакет **amsmath**.

В дальнейшем предполагается, что этот пакет уже загружен:

```
\usepackage{amsmath}
```

Полностью описать *все* команды языка описания формул вряд ли реально, потому что математика, как и способы её описания, безграничны. Поэтому основное внимание будет уделено базовым правилам и русскому стилю в формулах.

4.1. Набор формул

При формировании текста формулы подразделяются на *строчные* и *выносные*. Строчные формулы набираются внутри абзаца вместе с текстом. По описанию формулы L^AT_EX создаёт бокс, который обрабатывается наравне с обычными текстовыми боксами. Как правило, строковые формулы — это небольшие вставки, вроде $E = mc^2$. Выносные или *выключенные* формулы выводятся за пределы абзаца.

Строчная формула ограничивается¹ с помощью символа доллара \$«формула»\$ или с помощью команд-скобок \ («формула» \). При наборе предпочтительно использовать второй вариант оформления, так как он позволяет легко определить, где начинается, а где заканчивается формула. «Долларовое» (\$) окружение лучше тем, что оно чуть-чуть короче, кроме этого команда \$ *крепкая*² в отличие от команд-скобок.

Однострочные выносные формулы, как правило, формируются с помощью окружения equation. Так как в этом случае формула вынесена за пределы абзаца, то её можно пронумеровать. Например:

```
\begin{equation}
\label{eq:math:ex1}
\int\limits_{-\infty}^{\infty} e^{-x^2/2} dx = \sqrt{2\pi}
e^{-x^2/2} dx = \sqrt{2\pi}
\end{equation}
```

$$\int_{-\infty}^{\infty} e^{-x^2/2} dx = \sqrt{2\pi} \quad (4.1)$$

Нумерация формул удобна для того, чтобы позже в тексте на неё можно было легко сослаться с помощью команды \eqref{eq:math:1}³. Если же формул

¹Есть более формальное оформление строчной формулы как окружения: \begin{math} «формула» \end{math}. Но в силу понятных причин никто подобное описание не использует в пользу кратких обозначений.

²Когда начинаешь изучать команды L^AT_EX, то довольно быстро сталкиваешься с понятиями «хрупкости»/«крепкости». Крепкие команды в отличие от хрупких можно без опаски использовать в качестве аргументов других команд. С другой стороны, хрупкие команды тоже можно использовать как параметры, защитив их с помощью команды \protect. Эти понятия в большинстве своём пережитки прошлого, и их постепенно изживают, но пока их следует иметь в виду.

³Метка выставляется с помощью команды \label.

немного и не хочется никакой нумерации, то можно воспользоваться окружением `equation*`⁴.

По умолчанию выносная формула центрируется. Если при загрузке пакета **amsmath** указать опцию `fleqn`, то формулы будут выравниваться по левому краю с отступом `\mathindent`. В пакете **nccmath** определена декларативная команда `\fleqn[отступ]`, действие которой аналогично опции `fleqn`. Для центрирования формул в **nccmath** определена команда `\sern`.

При создании выключенной формулы размер шрифта для улучшения читаемости немного увеличивается. ЛАТ_EX имеет несколько стилей для оформления математических формул. При желании для целей улучшения визуального представления формулы можно выбрать один из следующих стилей вручную:

`\displaystyle` — стиль, который используется в выносных формулах,
`\textstyle` — стиль строчных формул,
`\scriptstyle` — в этом стиле набираются индексы,
`\scriptscriptstyle` — индексы второго уровня.

С помощью этих команд можно увеличить размер шрифта для формул внутри абзаца или заставить индексы выглядеть как базовые символы. Для примера можно сравнить:

<code>\begin{equation*}</code>	
<code>\frac{1}{1+}</code>	
<code>\frac{1}{1+}</code>	
<code>\frac{1}{1+}</code>	
<code>\frac{1}{2}}</code>	$\frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}$
<code>\end{equation*}</code>	
<code>\begin{equation*}</code>	
<code>\frac{1}{\displaystyle 1+}</code>	
<code>\frac{1}{\displaystyle 1+}</code>	
<code>\frac{1}{\displaystyle 1+}</code>	
<code>\frac{1}{\displaystyle 1}</code>	
<code>{\displaystyle 2}}</code>	$\frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}$
<code>\end{equation*}</code>	

Пробелы в формулах отмечают только конец команды, а сами по себе смысла не имеют. ЛАТ_EX, как правило, «знает» лучше, как сформировать результат.

4.2. Кириллица в формулах

Всё дело в имеющихся шрифтах — они красивые, разнообразные, но в большинстве своём англоязычные. В настоящее время кириллические математические шрифты в «дикий природе» отсутствуют, поэтому приходится пользоваться их текстовыми версиями.

⁴К `equation` добавляется звёздочка. Подобный приём в создании команд применяется достаточно часто. Команда со звёздочкой (*) обычно не нумеруется и не отображается ни в каких автоматически составляемых списках.

Стиль **mathtext** (пакет **t2**) позволяет использовать кириллицу в формулах без дополнительных ухищрений. Стиль можно применять с опцией **warn**, тогда он сообщает при компиляции обо всех случаях наличия кириллических букв в формулах.

Стиль **mathtext** следует загружать до **babel** и/или **fontenc**:

```
\usepackage [ warn ] { mathtext }
```

```
\[
v_{cp}=\frac{S_{\text{конец}}-S_{\text{начало}}}{\delta t}
\]
```

$$v_{cp} = \frac{S_{\text{конец}} - S_{\text{начало}}}{\delta t}$$

В примере со средней скоростью для создания выключенной формулы используется конструкция вида `\[ «формула» \]` — краткий аналог окружения `equation*`. В отличие от латиницы русские буквы в формулах печатаются прямым шрифтом — это было сделано специально. Чтобы изменить это умолчание, в преамбуле следует добавить команду для переопределения шрифта:

```
\DeclareSymbolFont {T2 A letters } {T2A} {cmr} {m} {it }
```

Стиль **amstext**, загружаемый автоматически при загрузке **amsmath**, определяет команду `\text`, которая позволяет вставлять в формулу обычный текст. В качестве аргумента этой команды можно передавать и русские символы:

```
\[v_{cp}=
\frac{\text{конец пути}-
\text{начало пути}}
{\text{время в пути}}\]
```

$$v_{cp} = \frac{\text{конец пути} - \text{начало пути}}{\text{время в пути}}$$

Преимущество такого подхода заключается в том, что пробелы внутри команды `\text` воспринимаются как нормальные символы и слова не сливаются. Использование `\text` предпочтительнее и для целей переносимости.

4.3. Школьная математика

Математика в школе — это явление, близкое каждому. Именно поэтому фактически любой вменяемый россиянин умеет обращаться с дробями, знает теорему Пифагора, с лёгкостью решает квадратные уравнения и что-то слышал про интеграл и производную. Разберёмся с этим поподробнее.

4.3.1. Отделяем целую часть от дробной

В англоязычной литературе целая часть от дробной отделяется точкой. В России действуют правила, которые подразумевают, что разделителем должна быть запятая. Проблема в том, что запятая в англоязычной литературе используется для разделения многозначных чисел на группы по три цифры (например, 123, 456, 789) и после запятой добавляется пробел. Можно сравнить, как ведут себя разные разделители:

<code>\(3.14159\)</code>	<code>\par</code>	3.14159
<code>\(3,14159\)</code>		3,14159

Однозначного решения этой проблемы нет. Если можно, то следует использовать в качестве разделителя точку. Если чисел в тексте немного, то можно оформлять их руками, заключив запятую в группирующие скобки, как это сделано в следующем примере:

<code>\(3.14159\)</code>	<code>\par</code>	3.14159
<code>\(3\{, \}14159\)</code>		3,14159

Также можно воспользоваться стилем **nccscomma** из пакета **nctools**. При инициализации этого стиля запятая будет считаться простым символом, если после неё идёт цифра, и знаком пунктуации, если после неё есть пробел. Единственное, за чем надо следить, так это за перечислениями вида $k = 1, 2, \dots, n$ с целью не забыть поставить пробел после запятой.

4.3.2. Индексы

Букв в латинском алфавите не так уж и много, а научных понятий не счесть. Один из способов отличать обозначения друг от друга — это индексы:

<code>\[A_{\text{нижний индекс}}\]</code>	<code>\quad</code>	<code>B^{\text{верхний индекс}}</code>	<code>\quad</code>	<code>C_n^k</code>
<code>B_{\text{верхний индекс}}</code>				
<code>C_n^k</code>				

Следует обратить внимание, что если в индексе ровно один знак, то фигурные скобки вокруг него можно и нужно опустить. Теперь несложно записать теорему Пифагора: $\backslash(a^2+b^2=c^2\backslash)$, что эквивалентно $a^2 + b^2 = c^2$.

4.3.3. Математические символы

Кроме символов латиницы и кириллицы математики используют множество самых разнообразных значков. Да и латиница не так уж проста. Если воспользоваться пакетом **amsfonts**, то она может стать:

<code>\begin{itemize}</code>	• $ABCD$ — обычной,
<code>\item{\(ABCD\)} "---</code> обычной,	• $\mathbf{ABCD}$ — жирной,
<code>\item{\(\mathbf{ABCD}\)} "---</code> жирной,	• $\mathcal{ABCD}$ — ажурной,
<code>\item{\(\mathbb{ABCD}\)} "---</code> ажурной,	• $\mathcal{A}BCD$ — прописной,
<code>\item{\(\mathcal{ABCD}\)} "---</code> прописной,	• $\mathfrak{A}BCD$ — готической,
<code>\item{\(\mathfrak{ABCD}\)} "---</code> готической,	• $\mathscr{A}BCD$ — как в <code>mathrsfs</code> .
<code>\item{\(\mathscr{ABCD}\)} "---</code> как в <code>mathrsfs</code> .	
<code>\end{itemize}</code>	• $\mathcal{A}BCD$ — как в <code>mathrsfs</code> .

Это далеко не все возможные шрифтовые стили, которые можно применять в математической моде. Но лучше особо не перегружать формулы всякой «готикой» (намёк на команду `\mathfrak`).

Греческие символы

Не единой латиницей жив математик. Традиционно везде, где только можно, используются греческие буквы:

Греческие символы									
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>	ε	<code>\epsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>	$\omicron$	<code>\omicron</code>
π	<code>\pi</code>	ρ	<code>\rho</code>	σ	<code>\sigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
φ	<code>\phi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>	Γ	<code>\Gamma</code>
Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>	Π	<code>\Pi</code>
Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>

В **Л^AT_EX** присутствует полный набор, и за исключением трёх букв начертание вполне привычное. Для исправления непривычных начертаний эти буквы были переопределены с помощью стиля **amssymb**:

```
% Переопределение \kappa, \epsilon и \phi на русский лад
\renewcommand{\kappa}{\varkappa}
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\phi}{\varphi}
```

Школьные спецсимволы

Спецсимволов в **Л^AT_EX** великое множество. В стандартной поставке **T_EX Live** идёт «Всеобъемлющий список символов **Л^AT_EX**» (The Comprehensive **La**T_EX Sym-

bols List — файл `symbols-a4.pdf`), в котором перечислено около 3300 распространённых символа, доступных пользователями ЛАТ_EX. Почти наверняка любой операнд, который вам нужен, там уже есть. Далее будет перечислена только та часть символов, которая может пригодиться в наборе школьной математики. Стиль `amssymb` для использования обязателен.

«Школьные» символы					
$\hat{a}$	<code>\hat{a}</code>	$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>
$\tilde{a}$	<code>\tilde{a}</code>	$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\vee$	<code>\vee</code>
$\neg$	<code>\neg</code>	$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>
$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\ll$	<code>\ll</code>
$\neq$	<code>\neq</code>	$\equiv$	<code>\equiv</code>	$\sim$	<code>\sim</code>
$\approx$	<code>\approx</code>	$\propto$	<code>\propto</code>	$\parallel$	<code>\parallel</code>
$\angle$	<code>\angle</code>	$\triangle$	<code>\triangle</code>	$\sphericalangle$	<code>\sphericalangle</code>
ℓ	<code>\ell</code>	Σ	<code>\Sigma</code>	$\prod$	<code>\prod</code>

Для соответствия русским традициям два символа ($\leq$ и $\geq$) из представленных в таблице были переопределены:

```
% Переопределение \le и \ge на русский лад
\renewcommand{\le}{\leqslant}
\renewcommand{\ge}{\geqslant}
```

4.3.4. Корни

Для рисования знака корня используется команда:

```
\sqrt [ «степень» ] { «подкоренное выражение» }
```

Степень можно опустить. В этом случае рисуется обычный квадратный корень.

```
\[
\overline{
\underline{\Large
\sqrt[3]{a}+\sqrt[2]{b}+\sqrt[99]{g}
}}
\]
```

$$\overline{\sqrt[3]{a} + \sqrt[2]{b} + \sqrt[99]{g}}$$

Следует обратить внимание, что знак корня размещается в соответствии с размерами подкоренного выражения. Если в выражении присутствует только один корень, то это самое разумное поведение, но в случае нескольких корней, как в приведённом примере, необходимо выравнивание.

Для выравнивания по высоте используется команда `\mathstrut`⁵. В результате её применения вставляется невидимый символ нулевой толщины, в точности равной высоте круглой скобки:

```
\[
  \sqrt[3]{\mathstrut a}+
  \sqrt[2]{\mathstrut b}+
  \sqrt[99]{\mathstrut g}
\]
```

$$\sqrt[3]{a} + \sqrt[2]{b} + \sqrt[99]{g}$$

4.3.5. Дроби

Дроби формируются с помощью команды `\frac`⁶:

```
\[
  дробь=\frac{числитель}{знаменатель}
\]
```

$$\text{дробь} = \frac{\text{числитель}}{\text{знаменатель}}$$

Как и практически вся математика в $\text{\LaTeX}$, дробь записывается точно так же, как читается само выражение.

4.3.6. Квадратное уравнение

И наконец, вершина школьной математики — это решение квадратного уравнения $ax^2 + bx + c = 0$:

```
\[
  x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\]
```

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Теперь можно смело писать методички по школьной математике ☺.

4.3.7. Производная и интеграл

В старших классах в конце обучения чуть-чуть касаются понятий интегрирования и дифференцирования. Возможно, для того, чтобы правильно подсчитать сдачу в магазине, эти знания не являются необходимыми. Но для изучения физики и, как следствие, химии и биологии без интегралов никак. Ведь даже второй

⁵От английского strut — подпорка или страта.

⁶От слова fraction — дробь.

закон Ньютона $\vec{F} = m\vec{a}$ является при более внимательном рассмотрении дифференциальным уравнением второго порядка.

Производная обычно отмечается штрихом. В физике производная по времени выделяется точкой. Можно также честно написать $\frac{d F(x)}{dx}$. Для частной производной вместо буквы d используется спецсимвол ∂ :

$\backslash[f'\quad f''\quad f'''\quad f^n \backslash]$	$f' \quad f'' \quad f''' \quad f^n$
$\backslash[\dot{x}\quad \ddot{x}\quad \ddot{x}\quad \ddot{x}\backslash]$	$\dot{x} \quad \ddot{x} \quad \ddot{x} \quad \ddot{x}$
$\backslash[\frac{d f}{d x}\quad \frac{d^n f}{d x^n}\quad \frac{\partial^2 f}{\partial x \partial y}\backslash]$	$\frac{df}{dx} \quad \frac{d^n f}{dx^n} \quad \frac{\partial^2 f}{\partial x \partial y}$

Дифференцирование есть действие, противоположное интегрированию. В отличие от производной для интеграла есть своя функция $\int$:

$\backslash[\frac{d}{dx}\int F(x) dx=F(x) \backslash]$	$\frac{d}{dx} \int F(x) dx = F(x)$
--	------------------------------------

Приглядевшись к этому примеру, можно отметить, что в отличие от русских математических традиций представленный здесь интеграл не прямой, а наклонный. Проблема решается, например, с помощью загрузки стиля **wasysym** с опцией **integrals**. К сожалению, получающиеся интегралы «не смотрятся». Поэтому пока лучше использовать начертания по умолчанию в надежде, что в будущем ситуация изменится к лучшему.

Неопределённый интеграл — это хорошо, но с определённым тоже надо уметь работать. Качественное оформление пределов критично для восприятия:

$\backslash[\int_0^\infty\quad \int\limits_0^\infty\quad \sum_{i=1}^n\quad \sum\limits_{i=1}^n\quad \backslash]$	$\int_0^\infty \quad \int\limits_0^\infty \quad \sum_{i=1}^n \quad \sum\limits_{i=1}^n$
--	---

По умолчанию пределы размещаются справа от интеграла. Ситуацию можно поправить с помощью команды $\backslashlimits$. Следует отметить, что пределы для знака суммы $\backslashsum$ по умолчанию расположены сверху и снизу. Для изменения этого умолчания следует воспользоваться командой $\backslashnolimits$.

4.3.8. Функции

Все символы в математической моде печатаются курсивом, поэтому названия именованных функций для выделения печатаются прямым шрифтом. Кроме смены шрифта функции с обеих сторон должны правильно «отбиваться» пробелами, иначе будет некрасиво. При загрузке русского языка с помощью пакета **babel** кроме стандартных имён функций доопределяется несколько сокращений, применяемых в русскоязычной литературе. Среди часто употребляемых функций можно упомянуть: `\cos`, `\arccos`, `\sin`, `\arcsin`, `\tg`, `\arctg`, `\ctg`, `\arcctg`, `\sh`, `\ch`, `\th`, `\cth`, `\exp`, `\ln`, `\log`, `\lim`, `\min` и `\max`. В математической моде эти функции можно использовать как обычные команды.

```
\begin{equation*}
\begin{split}
&\log_2 10 = \ln 10 / \ln 2 \simeq 3.32 \\
&\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \\
&(a+b)^n = \sum_{k=1}^n C_n^k a^k b^{n-k}
\end{split}
\end{equation*}
```

$$\log_2 10 = \ln 10 / \ln 2 \simeq 3.32$$

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

$$(a+b)^n = \sum_{k=1}^n C_n^k a^k b^{n-k}$$

Из примера видно, что обработка индексов для функции `\log` (логарифм) и `\lim` (предел) значительно отличается. Для доопределения новых функций правильнее всего в преамбуле будет воспользоваться командой `\DeclareMathOperator`:

```
% В преамбуле — определение новых функций
\DeclareMathOperator{\log-like}{log-like}
\DeclareMathOperator*{\lim-like}{lim-like}
```

В зависимости от варианта команды индексы отображаются как для логарифма (команда без звёздочки) или как для предела (команда со звёздочкой).

4.3.9. Скобки

Для визуальной группировки символов внутри формулы скобки — вещь незаменимая. Для красоты и выразительности размер скобок в формуле варьируется. В принципе, размер скобок можно выставить вручную, тогда перед скобкой следует добавить одну из следующих команд:

```
\[ ( \big( \Big( \bigg( \]
```

(((((

Эстеты в зависимости от ситуации в конце команды могут добавить модификатор позиционирования разделителя как левого — `l` (отбивка как для `\left`), правого — `r` (отбивка как для `\right`) и среднего — `m`:

$$\left[\left| \int \right| \left| \int \right| \left| \int \right| \right]$$

```
\[ \Biggl| \int \Biggl| \Biggm| \int \Biggm|
\Biggr| \int \Biggr| \]
```

Особенно здорово, если скобки автоматически подбирают свой размер под выражение, которое они окружают. Парные команды `\left` и `\right` включают режим подобной подстройки.

```
\[ \left(
  \left[
    \left\langle
      \left\{
        \left\uparrow
        \left\lceil
          \left|
            \left\lfloor
              \text{что-то}^{10}
              \right\rfloor^{9}
              \left\lceil^{8}
                \left\downarrow^{6}
                \right\}^{5}
              \right\rangle^{4}
            \right\}^{3}
          \right]^{2}
        \right\rangle
      \right\}
    \right]
    \right\lceil^{8}
    \right\rceil^{7}
    \right\downarrow^{6}
  \right\}^{5}
\right\rangle^{4}
\right]^{3}
\right)^{2}
```

$$\left(\left[\left\langle \left\{ \left\uparrow \left\lceil \left| \left\lfloor \text{что-то}^{10} \right\rfloor^{9} \left\lceil^{8} \left\downarrow^{6} \right\}^{5} \right\rangle^{4} \right\}^{3} \right]^{2} \right\rangle \right\} \right] \right\lceil^{8} \right\rceil^{7} \right\downarrow^{6} \right\}^{5} \right\rangle^{4} \right]^{3} \right)^{2}$$

4.4. Перенос формул

В русскоязычной литературе принято, что при переносе строчной формулы на другую строку знак, по которому формула разрывается, дублируется на следующей строке. Например так:

$$a + b =$$

$$= c$$

По умолчанию автоматического переноса с дублированием знаков не происходит. Проще всего решить эту проблему вручную с помощью макроса⁷, который необходимо определить в преамбуле:

% перенос формул в тексте
\newcommand*{\hm}[1]{\#1\nobreak\discretionary}{}%
{\hbox{\$\mathsurround=0pt \#1\$}}}

Здесь была определена команда `\hm`, которую следует добавлять в местах потенциального переноса формулы, примерно так: `\(a + b \hm{=} c\)`. Сделать это

⁷Рецепт от Евгения Миньковского из `fido7.ru.tex`.

можно во время окончательной доводки текста. В любом случае для полировки рукописи ручная работа необходима.

Разрыв математических формул при переносе предпочтителен на знаках отношения ($=$, $>$, $<$, $\leq$, $\geq$, $\neq$, $\simeq$); во вторую очередь на отточии, знаках сложения и вычитания; в третью — на знаке умножения в виде крестика. Не рекомендуется разбивать формулу на знаке деления и на каких-либо других знаках, кроме упомянутых ранее.

► Изложенных в этой главе правил и приёмов вполне хватит для набора в рамках школьной математики. Для более изощрённых формул требуются более продвинутые приёмы и конструкции, часть из которых представлена в главе 13.

Глава 5

Вставка графики

Картинка стоит тысячи слов.

Английская пословица

Вероятно, $\text{\TeX}$ на текущий момент лучше других программ вёрстки умеет разбивать абзацы на строки. То есть удачнее всех разливать порции «клея» между «боксами», но подготовка графики выносится за рамки этого процесса. Почти. . .

С точки зрения $\text{\TeX}$ картинка — это просто очень большой прямоугольник, который надо как-то разместить на странице. От пользователя нужны только размеры этого прямоугольника. Отображение же иллюстрации лежит на плечах драйверов. Самым востребованным форматом для представления графики в $\text{\LaTeX}$ до сих пор является Encapsulated PostScript.

5.1. Encapsulated PostScript

Уже больше двадцати лет прошло с тех пор, как никому не известная фирма Adobe Systems получила инвестиции от фирмы Apple на «обучение» лазерных принтеров молодому тогда ещё языку PostScript. Как следствие, этот платформонезависимый язык с полностью открытой спецификацией стал безальтернативным стандартом. Даже сейчас PostScript фактически не имеет конкурентов в области допечатной подготовки. Поэтому почти все «уважающие себя» графические программы умеют экспортировать результаты своей деятельности в виде инструкций PostScript. Особенно это касается векторных графических редакторов, так как PostScript подразумевает векторную графику.

Encapsulated PostScript или кратко EPS — это векторный графический формат. Файлы в этом формате, как правило, имеют расширение **eps**. Это фактически тот

же PostScript, но с некоторыми упрощениями и дополнительными договорённостями. Самая интересная с точки зрения L^AT_EX договорённость — это обязательное наличие в заголовке информации о размере картинki, которая передаётся вместе с комментарием:

```
%!PS-Adobe-2.0 EPSF-2.0
%%Creator: dvips(k) 5.95b Copyright 2005 Radical Eye Software
%%Title: picture.dvi
%%BoundingBox: 127 464 430 667
%%DocumentFonts: SFRM1200 SFRM0800
%%EndComments
```

Первая строка комментария обычно содержит версию PostScript¹. Вслед за комментарием **BoundingBox** идёт информация о размерах. Первые два числа соответствуют координатам левого нижнего угла картинki, а последние соответствуют координатам правого верхнего угла. Единицей измерения является «большой пункт» (bp=1/72 in), который примерно равен 0.351 мм. Для вёрстки текста указанной информации достаточно.

Чтобы из уже имеющегося одностраничного PostScript-файла сделать EPS, необходимо и, как правило, достаточно добавить **BoundingBox**. Для вычисления искомых размеров можно воспользоваться утилитой **ps2eps** из одноимённого пакета. Если же в стандартной поставке эта программа отсутствует, то можно напрямую воспользоваться программой Ghostscript — свободным программным интерпретатором PostScript:

```
> gs -q -dSAFER -dNOPAUSE -dBATCH -sDEVICE=bbbox «имя файла»
```

Размеры выясняются с помощью указания специального драйвера **bbbox**. Ключи **-q**, **-dNOPAUSE** и **-dBATCH** используются для подавления ненужной информации и вопросов со стороны программы. Ключ **-dSAFER** гарантирует, что Ghostscript не будет производить никаких деструктивных действий².

Ещё одной особенностью EPS-формата является возможность добавлять растровое изображение для предварительного просмотра. Это было сделано для случаев, когда программы не понимают PostScript, но что-то на месте дырки для картинki отобразить надо. Такое добавление идёт вразрез с принципиальной кросс-платформенностью PostScript, и по возможности добавления картинki для предварительного просмотра следует избегать. Но в любом случае для операции с этим расширением, в том числе и для добавления/удаления можно воспользоваться утилитой **epstool** из одноимённого программного пакета.

¹Некоторые программы перед комментарием добавляют бинарный мусор. Не будем тыкать пальцем в драйвер вывода для PostScript-принтеров у одной очень распространённой операционной системы. Для полноценной работы с такими файлами этот мусор необходимо удалить.

²Отключается возможность выполнения таких команд, как удаление и переименование, а чтение файлов происходит в режиме read-only. Очень полезный ключ, если Ghostscript используется в качестве фильтра.

В конце рассказа про EPS хотелось бы упомянуть о замечательной утилите **pstoedit** из, естественно, одноимённого же программного пакета. Не все, но многие из более-менее внятно созданных PostScript-файлов она ухитряется перевести в редактируемый векторный графический формат. Это упрощает работу с правкой файлов, которые не имеют исходников.

5.2. Как из растра сделать EPS

Одним из важных вопросов вставки графики в ЛАТ_ЭX является конвертация растровых форматов в EPS. Растр гораздо проще создавать. Кое-где, например, в случае снимков экрана, применение растра оптимальнее векторных форматов. Стандартные подходы, как в случае утилиты **convert** из пакета **ImageMagick**, не всегда дают оптимальные результаты.

Возможным и вполне разумным решением является замена традиционной линейки: **latex**→**dvips**→**ps2pdf** на **pdflatex**, который сразу «из коробки» поддерживает растровые форматы PNG³ и JPEG⁴, которые можно внедрять в формат PDF⁵ напрямую. Массового перехода на эту технологию пока не видно, но заметное движение в эту сторону есть. У неё есть неоспоримые достоинства, но она не лишена недостатков. Рассказ о **pdflatex** выходит за рамки этого раздела.

Конвертацию из формата JPEG проще всего осуществить с помощью программы **jpeg2ps**, которую можно найти в любом CTAN-архиве в директории `{CTAN}/nonfree/support/jpeg2ps`. Утилита не преобразовывает JPEG-файл, а просто добавляет правильный eps-заголовок. Декомпрессия картинки JPEG производится уже PostScript-интерпретатором. Это стало возможным, начиная с версии PostScript Level 2. К недостаткам утилиты можно отнести то, что в силу своей лицензии она не может распространяться со свободными дистрибутивами, а к достоинствам — отсутствие зависимостей от других программных пакетов.

Более комплексными решениями являются утилиты **sam2p** из одноимённого пакета и **bmeps**. Их также можно найти на CTAN в директориях `graphics/sam2p` и `support/bmeps` соответственно. **sam2p** является своеобразным комбайном, который поддерживает множество растровых графических форматов, в то время как **bmeps** фокусируется на PNG и JPEG. Обе эти программы позволяют получить вполне приличную eps-картинку для печати или для просмотра на экране. В обоих случаях необходимо поразбираться в ключах и настройках. Программа **bmeps** является более удобным решением, производящим достаточно маленькие по размеру eps-файлы, но и **sam2p** достаточно хорош.

³Portable Network Graphics — растровый графический формат, использующий сжатие без потерь.

⁴Joint Picture Experts Group — самый популярный графический формат из форматов, использующих сжатие с потерями.

⁵Portable Document Format — платформонезависимый формат электронных документов, созданный Adobe System.

Опять же в архиве CTAN в директории `{CTAN}graphics/a2ping` можно взять довольно «увесистый» perl-скрипт **a2ping.pl**. Этот скрипт является своеобразной надстройкой над **sam2p** и Ghostscript, что позволяет ему более-менее автоматически конвертировать из растра в PostScript и обратно.

5.3. graphicx

Ответственным за создание «бокса» для размещения картинки является пакет **graphicx**⁶, а точнее команда `\includegraphics`:

```
% Эмблемы TeX и METAFONT, созданные
% Дуайном Вибби, взяты со странички Д.Э.Кнута.
```

```
% Пингвина нарисовал Neal Tucker
```

```
\includegraphics[width=\textwidth]{title.eps}
```



`\textwidth` — ширина тела текста. В команде `\includegraphics` есть один обязательный параметр — вставляемая картинка. Необязательные параметры передаются с помощью пар «ключ»=«значение», разделяемых запятой. За подобный способ объявления параметров отвечает пакет **keyval**. Некоторые из поддерживаемых пакетом параметров перечислены далее.

bb — позволяет исправить **BoundingBox** прямо в ЛАТЭХ-коде, не меняя ерф-файл. Значение представляет собой последовательность из четырёх чисел, кодирующую положения левого нижнего и правого верхнего углов рамки, например: `[bb=127 464 430 667]`. Вместо одного **bb** можно воспользоваться четвёркой ключей, каждому из которых передаётся только одно значение: `[bbllx=127,bbllly=464,bbrrx=430,bbrry=667]`.

Кроме перечисленных ключей для модификации **BoundingBox** можно использовать ключ **viewport** — четыре числа-значения описывают границы **BoundingBox**, где в качестве центра координат выбирается левый нижний угол уже существующего описания, и **trim** — четыре числа-значения описывают отступы от левой, нижней, правой и верхней границ.

clip — обрезает вставленную картинку по **BoundingBox**. Это необходимо сделать в случае изменения границ для «выкусывания» части картинки, иначе она будет «вылезать» за пределы выделенного ей бокса. По умолчанию имеет значение **false**. Отсутствие значения у ключа **clip** при его упоминании эквивалентно значению **true**. Подобное поведение верно и для других логических переключателей.

angle — поворачивает картинку на указанный угол в градусах.

⁶**graphicx** пришёл на смену пакету **graphics** — различия в последней букве. Команды из предыдущего пакета также можно использовать, но настоятельно не рекомендуется.

origin — определяет координаты центра, вокруг которого вращается рисунок. Кроме непосредственно координат **origin** принимает и буквенные сокращения: **l**, **b**, **r** и **t** — соответствует центру вращения слева, снизу, справа и сверху. В этом случае выбирается середина указанной стороны. Возможны комбинации, задающие углы картинки: **lt**, **rt**, **rb** и **lb**. **c** — означает центр картинки.

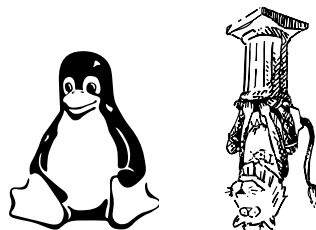
width — ширина вставляемой картинки.

height — высота вставляемой картинки.

scale — масштабный коэффициент.

keepaspectratio — логический переключатель. Модифицирует параметры высоты и ширины картинки в сторону уменьшения с целью сохранения естественных пропорций картинки.

```
\includegraphics[trim=110 0 105 100,clip,
    width=0.49\textwidth]{title.eps}
\hspace{0.5cm}
\includegraphics[viewport=0 0 100 200,clip,
    width=0.49\textwidth,
    height=3cm,keepaspectratio,
    angle=180,origin=c]{title.eps}
```



► Аргументы `\includegraphics` интерпретируются слева направо. Для команд вращения и масштабирования порядок следования *имеет* значение.

5.3.1. Определение своих правил

Пакет **graphicx** предоставляет возможность перед вставкой картинки вызвать внешнюю программу для её обработки. Например, так можно добавить возможность включения в документ png-файлов:

```
\DeclareGraphicsRule{.png}{eps}{.bb}{'bmeps -p3 -c #1}
```

Первый параметр определяет расширение нового формата графики, для которого задаются правила. В представленном примере это png-файл. Второй параметр указывает тип графики, который передаётся непосредственно L^AT_EX. В данном примере гарантируется, что после преобразования это будет полноценный eps или Encapsulated PostScript, что необходимо, так как **dvips** по умолчанию ничего другого и не знает. Третий параметр определяет расширение файла, откуда можно извлечь параметры **BoundingBox**, то есть файл с расширением **bb** должен содержать строку вида:

```
%%BoundingBox: 0 0 848 979
```

До вставки для каждой png-картинки необходимо создать такой файл, например, следующим образом:

```
> bmeprs -b «картинка».png «картинка».bb
```

Последний параметр определяет команду, которую следует выполнить для преобразования картинки. Команда должна выдавать результат на стандартный вывод. #1 соответствует имени обрабатываемого файла. Непосредственное выполнение команды происходит при трансляции dvi-файла.

► Выполнение внешней команды является потенциально *опасной* процедурой, поэтому защита по умолчанию этого не позволяет. Например, для просмотра dvi-файла через **xdvi** следует использовать ключ **-allowshell**. В противном случае будет выдаваться запрос на исполнение команды каждый раз, когда встречается вставка по новым правилам. Для преобразования в PostScript в случае **dvips** также следует отключить защиту с помощью ключа **-R0**. Лучше всё-таки по возможности избегать описанной процедуры и сразу готовить картинки в eps-формате.

Для трактования всех неизвестных драйверу расширений как eps следует применить команду:

```
\DeclareGraphicsRule{*}{eps}{*}{}{}
```

Это полезно в случае вставки картинок MetaPost, которые по умолчанию не имеют расширений. Если третий параметр равен «звёздочке», то это означает, что **BoundingBox** следует искать в том же файле, что и графику.

5.4. Плавающие объекты

Мало просто поместить картинку — её надо разместить красиво. При этом всё должно делаться автоматически с минимальным вмешательством со стороны автора. Просто `\includegraphics` для этого дела не очень подходит, так как размещение регулируется исключительно пользователем. Чтобы автоматизировать размещение рисунка, в ЛАТЭХ имеется специальная сущность: *плавающий объект* (`float`⁷). Если для этого объекта нет места на текущей странице, то он переносится на следующую. Для размещения картинок стандартные классы определяют плавающий объект как окружение `figure`:

```
\begin{figure}[ht]
  \centering % Центрируем картинку
  \includegraphics{«картинка»}
  \caption{«подпись»}\label{fig:metka}
\end{figure}
```

⁷Пакет, который позволяет создавать новые типы плавающих объектов, так и называется **float**. Более современная версия этого пакета написана Ольгой Лапко и называется **floatrow**.

В качестве необязательного параметра окружению `figure` можно передать допустимые способы размещения плавающего объекта:

`h` — разместить по возможности здесь же,

`t` — разместить в верхней части страницы,

`b` — разместить в нижней части страницы,

`p` — разместить на отдельной странице, где нет ничего кроме плавающих объектов.

Приоритет для размещения `figure` определяется порядком следования букв. Если первой следует буква `h`, а второй `t` или `b`, то в случае неудачи $\text{\LaTeX}$ размещает плавающий объект в верхней или нижней части *следующей* страницы соответственно.

Для «красивого» размещения картинок $\text{\LaTeX}$ опирается на некоторые значения по умолчанию, которые не всегда для текущего случая могут быть оптимальными. Поэтому если очень хочется разместить картинку, например, внизу, то пожелание можно усилить с помощью восклицательного знака: `[b!]`.

5.4.1. Управление плавающими объектами

Если плавающих объектов в документе немного, то всё будет хорошо без какого-либо вмешательства человека. Но если их много, то так или иначе надо будет управлять их размещением.

`clearpage`

Если $\text{\LaTeX}$ не справляется с размещением картинок, то он переносит их на следующую страницу. В какой-то момент может накопиться целая «толпа» таких перенесённых картинок и возникнет необходимость в их «насильственном» выводе в каком-то определённом месте. Для этого существует команда `\clearpage`. При вызове этой команды завершается текущая страница, выводятся все отложенные плавающие объекты, и только потом продолжается обычный вывод текста. Единственная проблема этой команды в том, что текущая страница по ней обрывается. Чтобы избежать обрыва, можно воспользоваться пакетом `afterpage`, точнее одноимённой командой из него:

```
\afterpage{\clearpage}
```

Команда `\afterpage` откладывает выполнение указанных в ней инструкций до конца текущей страницы.

`suppressfloats`

Команда `\suppressfloats` полностью подавляет размещение плавающих объектов на текущей странице. В качестве необязательного параметра ей можно передать `t` или `b` — в этом случае запрет распространяется только на размещение плавающих объектов вверх или вниз страницы соответственно.

placeins

Пакет **placeins** не даёт «утекать» плавающим объектам за установленные пределы. Барьер устанавливается с помощью команды `\FloatBarrier`.

Это бывает полезно, когда хочется, чтобы все картинки не выходили за пределы своего раздела. В этом случае следует переопределить нужную команду секционирования для установки перед ней барьеров. В случае команды секционирования раздела (section) достаточно передать опцию `[section]` пакету при загрузке:

```
\usepackage[section]{placeins}
```

endfloat

Часто при подготовке статей требуют их размещения после текста на отдельных страницах, предвояя эту галерею списком иллюстраций. Пакет **endfloat** именно это и делает. Достаточно его загрузить.

5.4.2. «Упаковка» картинок в один float

Для уменьшения «поголовья» плавающих объектов полезно размещать картинки группами. Например, чтобы разместить две картинки рядом, можно применить команду `\parbox` или окружение `minipage`:

```
\parbox[«позиционирование»]{«ширина»}{«текст»}

\begin{minipage}[«позиционирование»]{«ширина»}
  текст
\end{minipage}
```

В обоих случаях есть обязательный параметр «ширина», по которой формируется создаваемый бокс и необязательный «позиционирование» — расположение сформированного бокса относительно базовой линии по вертикали. Позиционирование может проводиться по центру (опция `[c]` — верно по умолчанию), по верхней линии (`[t]`) и по нижней линии бокса (`[b]`). Шаблон для двух рядом стоящих рисунков может иметь примерно следующий вид:

```
\begin{figure}[ht]\centering
  \parbox[b]{0.49\textwidth}{\centering
    \includegraphics{«рисунок-1»}
    \caption{«подпись-1»}\label{fig:metka-1}}
  \hfil\hfil%раздвигаем боксы по горизонтали
  \begin{minipage}[b]{0.49\textwidth}
    \centering
    \includegraphics{«рисунок-2»}
```



```
\caption{«подпись-2»}\label{fig:metka-2}
\end{minipage}
\end{figure}
```

Использование команды `\parbox` или окружения `minipage` зависит исключительно от личных предпочтений. С их помощью можно организовать и более сложные конструкции.

Для целей автоматизации упаковки можно использовать и специализированные пакеты.

subfig — организует группы из множества картинок. Относительно современный пакет.

miniplot — делает то же самое, что и **subfig**, хоть и менее изощрённо.

figsize — специализируется на автоматическом вычислении размеров картинок для размещения их в указанных пределах.

dpfloat — определяет новый тип плавающего окружения, занимающего сразу две страницы. Двойные иллюстрации на развороте.

5.4.3. Картинки «в оборку»

Маленькие иллюстративные рисунки удобно делать в оборку с текстом, то есть текст должен обтекать их. Такие картинки располагаются на внешней стороне страницы, то есть слева для чётных, а справа для нечётных страниц или в случае одностороннего режима печати.

Традиционно разбираются два пакета для создания подобных рисунков — это **floatflt** и **wrapfig**. Заметим, что **floatflt** более автоматизирован для размещения картинок, но он также чаще «ломается» при большом числе плавающих объектов и его нужно самостоятельно устанавливать с CTAN. Возможны даже «потери» картинок. Эти пакеты определяют окружения `floatingfigure` и `wrapfigure` соответственно.

```
\begin{floatingfigure}[«размещение»]{«ширина»}
...
\end{floatingfigure}
```

Необязательный параметр «размещение» окружения `floatingfigure` позволяет изменить алгоритм размещения картинки:

rflt — размещать справа,

lflt — размещать слева,

vflt — слева для чётных и справа для нечётных страниц (по умолчанию).

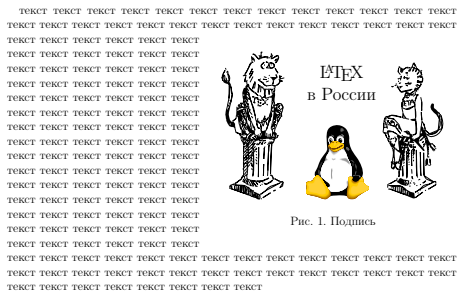


Рис. 1. Подпись

Рис. 5.1. Картинка в оборку.

Окружение `wrapfigure` имеет следующую структуру:

```
\begin{wrapfigure}[«число строк в оборке»]
                  {«размещение»}{«ширина»}
...
\end{wrapfigure}
```

В отличие от `floatingfigure` окружение `wrapfigure` требует определить правила размещения картинки в обязательном порядке. Доступные варианты:

`r` — размещать справа,

`l` — размещать слева,

`i` — размещать с внешней стороны страницы,

`o` — размещать с внутренней стороны страницы.

Если вместо строчных букв передать заглавные, то включается запрет на сдвиг по вертикали — картинка должна быть размещена, начиная с той строки абзаца, в которой она была определена.

Необязательный параметр «число строк в оборке» позволяет указать число строк текста, которые должны быть сбоку от картинки. При этом выносная формула считается за три строки текста. Если параметр не определён, то число строк вычисляется автоматически, к сожалению, не всегда оптимально.

Свою процедуру размещения картинки в оборку с текстом предлагает также и пакет **nccfloats** из коллекции **ncctools**, созданной А. И. Рожено:

```
\sidefig («ширина картинки»)(«ширина текста»)
{\includegraphics{«картинка»}}{«текст»}
```

В этом случае предлагается передавать команде `\sidefig` и саму картинку и текст, помещаемый сбоку. Параметр «ширина текста» можно опустить. Тогда текст занимает всё оставшееся пространство. Подробности в **nccfloats.pdf**.

5.4.4. Подписи к рисункам

Для добавление подписи к рисунку используется команда `\caption`, которую можно применять только внутри плавающих объектов. В качестве обязательного параметра передаётся текст подписи. При выводе подпись центрируется, если она достаточно мала. В противном случае подпись оформляется в виде абзаца. Текст подписи не должен содержать команд разрыва строки. Все хрупкие команды внутри подписи должны быть защищены с помощью команды `\protect`. `\caption` можно передать также необязательный параметр, который должен представлять собой краткую версию подписи, появляющуюся в автоматически создаваемых списках.

Оформление подписи жёстко привязано к стилю документа, и изменить её без переопределения самой команды `\caption` не просто. Для модификации параметров следует воспользоваться пакетами **caption** или **ccaption**. Документация в **caption.pdf** и **ccaption.pdf** соответственно.

При включении поддержки русского языка `\usepackage[russian]{babel}` перед подписью выводится слово «Рис.», за которым идёт автоматически вычисляемый порядковый номер картинки. В качестве разделителя между счётчиком и подписью по умолчанию используется двоеточие. Для замены двоеточия на точку в преамбуле достаточно набрать, например, следующее:

```
\usepackage{scaption}  
% заменяем для рисунков ':' после номера рисунка на '.'  
\captiondelim{. } % после точки стоит пробел!
```

Кроме традиционного размещения подписи под картинками, подпись можно вынести, например, на поля страницы. Недаром же стандартные классы имеют такие широкие поля. Пакет **mcaption** определяет окружение `margincap`:

```
\begin{figure}[ht]  
  \begin{margincap}{«Подпись»}  
    \includegraphics{«картинка»}  
  \end{margincap}  
\end{figure}
```

В качестве обязательного параметра окружению передаётся подпись, а внутри определяется картинка. Подробности в `mcaption.pdf`.

Глава 6

Начала программирования

Когда придумываешь что-то сам, высок шанс ничего не придумать. Но когда живёшь чужим умом, уж точно ничего не придумаешь. Никогда не делай того, что делают другие. Это на 100% обрекает на неудачу.

Герш Ицкович Будкер

Л^AT_EX позволяет не просто набирать текст — он позволяет его программировать, а следовательно, перекладывать часть своей работы на компьютер. Привычка думать — одна из самых необычных особенностей разумного человека. Она позволяет экономить силы и время.

6.1. Определённые размеры и переменные длины

Л^AT_EX поддерживает переменные типа «длина» для определения расстояния. Например, ранее в главе «Вставка графики» на стр. 60 уже упоминалась команда `\textwidth` — это переменная, хранящая значение длины, равной ширине текста.

Для создания переменной типа «длина» необходимо воспользоваться командой `\newlength`. В качестве обязательного параметра передаётся имя переменной. При создании переменной ей по умолчанию присваивается нулевая длина, так что следующим шагом необходимо приравнять её чему-либо с помощью команды `\setlength`:

<code>\newlength{\MyLength}</code>	Длина <code>\MyLength</code> равна
<code>\setlength{\MyLength}{1cm plus 2.5fill minus 5mm}</code>	74.69145pt plus 2.5fill minus
<code>\addtolength{\MyLength}{5em}</code>	14.22636pt.
Длина <code>\lstinline!\MyLength!</code> равна <code>\the\MyLength</code> .	

Длина в $\text{\LaTeX}$ не просто какой-то определённый размер — это более сложная структура с указанием границ возможного сжатия и растяжения. Границы растяжения определяются с помощью инструкции `plus`, а сжатия — `minus`. При формировании абзацев $\text{\TeX}$ использует эту информацию для максимально «красивого» заполнения.

Команда `\setlength` эквивалентна оператору присваивания. В свою очередь команда `\addtolength` позволяет увеличить переменную на указанную величину, которая может быть отрицательной. Макрос `\the` позволяет «развернуть» переменную длины для вывода на печать.

$\text{\TeX}$ «говорит» в терминах англо-американской системы мер. Эта система отживает своё, но её наследие будет ещё долго проявляться и портить жизнь современному «метрическому» миру. Для определённости следует знать, что один дюйм (in) равен 2.54 сантиметра и в нём уместается 72.27 пунктов ($1\text{ pt} \simeq 0.35\text{ mm}$). Метрические величины представлены привычными сантиметрами (cm) и миллиметрами (mm). Кроме упомянутых $\text{\TeX}$ умеет оперировать с размерами в больших пунктах (bp), пунктах Дидо (dd), пиках (pc) и цигеро (cc) — традиционные единицы измерения, используемые в типографиях. Минимальной ненулевой единицей длины в $\text{\TeX}$ является приведённый пункт (sp), который составляет $1/65536$ от одного пункта.

Кроме определённых единиц измерения длины можно задавать также и в относительных: `1ex` соответствует высоте строчной латинской буквы x, а `1em` — ширине прописной латинской буквы M. Эти величины меняются вместе со сменой шрифта, что позволяет задавать автоматически масштабирующиеся горизонтальные и вертикальные промежутки, не привязанные к конкретному размеру и типу шрифта. Например, широкий пробел, задаваемый с помощью команды `\quad`, определяется как `\hspace{1em}`.

```
\setlength{\MyLength}{1ex}
```

```
Высота x равна \the\MyLength\par
```

```
\Large \setlength{\MyLength}{1ex}
```

```
Высота x равна \the\MyLength
```

Высота x равна 3.87405pt

Высота x равна 6.19771pt

Интересной инструкцией является длина `fill` — это *бесконечность*. $\text{\TeX}$ поддерживает операции с бесконечностями, причём оперирует тремя видами бесконечностей: `fil`, `fill` и `filll`, где `fil` $\ll$ `fill` $\ll$ `filll`. С помощью этих сущностей производится центрирование боксов и более сложные выравнивания.

Если хочется узнать ширину текста, то следует воспользоваться командой `\settowidth`:

```
\settowidth{\MyLength}{очень длинная фраза}
```

```
\addtolength{\MyLength}{1em}
```

```
\centering
```

```
\framebox[1.2\MyLength]{очень длинная фраза}\par
```

```
\framebox[1.2\MyLength]{короткая фраза}
```

очень длинная фраза

короткая фраза

Аналогично команда `\settoheight` позволяет выяснить высоту текста над базовой линией, а `\settodepth` — глубину под базовой линией. При использовании длины можно добавить перед ней множитель.

А теперь немного «магии» из английского FAQ по $\text{\LaTeX}$:

```
\makeatletter1
\newcommand{\maxwidth}{%
\ifdim\Gin@nat@width>\linewidth
\linewidth
\else
\Gin@nat@width
\fi
}
\makeatother
```

Эта конструкция определяет переменную длины `\maxwidth` таким образом, что при вставке картинки:

```
\includegraphics [ width=\maxwidth ] { «картинка» }
```

ширина картинки становится равной минимальной из двух возможных значений: «естественной» ширине картинки (размер в `BoundingBox`) или ширине строки. Это позволяет вывести картинку в натуральную величину при условии, что она не вылезает за рамки дозволенного, и «запихать» её в эти рамки, коли она за них вылезает.

6.2. Счётчики

Для нумерации разделов, перечней, плавающих окружений, формул и прочего используются счётчики (`counter`). Как и в случае с длинами в $\text{\LaTeX}$ представлена и целочисленная арифметика с использованием счётчиков в качестве переменных:

```
\newcounter{MyCount}
\setcounter{MyCount}{5}
Значение MyCount равно \arabic{MyCount},
или \alph{MyCount}, или \Asbuk{MyCount}. \par
\addtocounter{MyCount}{1550}
\arabic{MyCount} эквивалентно \Roman{MyCount}.
```

Значение `MyCount` равно 5,
или е, или Д.
1555 эквивалентно MDLV.

Новый счётчик создаётся с помощью команды `\newcounter` и инициализируется нулём. Создание счётчика является глобальной операцией, то есть при компиляции информация о его создании не исчезнет, даже если новый счётчик был

¹Комбинация команд `\makeatletter`/`\makeatother` нужна для того чтобы получить доступ к внутренней $\text{\LaTeX}$ -команде. Такие команды традиционно используют символ `@` в своём названии, что предотвращает случайное использование этих команд новичками.

определён внутри окружения. Для присвоения счётчику другого значения используется команда `\setnewcounter`, а для изменения на какое-то определённое число — `\addtocounter`.

В отличие от длин, основная роль которых помнить размеры какого-то определённого бокса, счётчики используются для отображения какой-либо структурной информации. Поэтому особое внимание уделяется оформлению вывода счётчиков. Чтобы просто отобразить численное значение счётчика с помощью арабских цифр, используется команда `\arabic{счётчик}`. Для римской числовой нотации необходимо воспользоваться командой `\Roman` и `\roman` — заглавные и строчные буквы соответственно. Счётчик может быть представлен также буквой алфавита: `\alph` — латинская строчная, `\asbuk` — кириллическая строчная и `\Asbuk` — кириллическая заглавная.

В стандартных классах уже определён набор счётчиков, в которых хранятся номера страницы (счётчик `page`), раздела (соответственно счётчики `part`, `chapter`, `section`, `subsection`, `subsubsection` и т. д.), подстрочного примечания (`footnote`), плавающих окружений (счётчики `figure` и `table`) и формул (`equation`).

При создании счётчика также автоматически создаётся команда с префиксом `\the` перед именем счётчика. Вызов такой команды выводит номер счётчика. При выводе номера раздела, плавающего объекта, уравнения и тому подобного используются именно такого рода команды, поэтому, переопределив `\the`-команду, можно немного изменить стиль, например, следующая команда предписывает в дальнейшем маркировать все страницы в римском стиле:

```
\renewcommand{\thepage}{\Roman{page}}
```

На базе счётчиков можно организовывать иерархические структуры, то есть можно указывать зависимости:

```
\newcounter{Main}\addtocounter{Main}{10}
\newcounter{Dep}[Main]\addtocounter{Dep}{10}
Было: \theMain.\theDep\par
\stepcounter{Main}
Стало: \theMain.\theDep
```

Было: 10.10
Стало: 11.0

При создании нового счётчика можно создать связь с уже существующим, указав имя существующего счётчика в качестве необязательного параметра. В примере выше счётчик `Dep` зависит от счётчика `Main`. Эта связь проявляется в том, что если увеличить значение базового счётчика (`Main`) на единицу с помощью команды `\stepcounter`, то подчинённый счётчик (`Dep`) обнуляется. Обычно новый счётчик устанавливают в подчинение счётчикам разделов (`section`).

Стиль **chngcntr** из пакета **ltxmisc** позволяет изменять зависимости уже после определения счётчика. Команда `\counterwithin{Dep}{Main}` привязывает счётчик `Dep` к счётчику `Main`. Для обратного действия можно воспользоваться командой `\counterwithout{Dep}{Main}`. Подробности можно найти в конце стилевого файла `chngcntr.sty`.

Команда `\refstepcounter{счётчик}` отличается от `\stepcounter` тем, что помимо обнуления всех зависимых счётчиков, `\refstepcounter` определяет значение, выводимое командой ссылки `\ref`, как текст, создаваемый `\the`-командой:

```
% окружение "Задача"
\newcounter{Problem}[section]
\renewcommand{\theProblem}{\thesection.\arabic{Problem}}
\newenvironment{Problem}[0]{%
  \par\refstepcounter{Problem}%
  \theProblem\,}%
{\par}%
```

Здесь определено окружение `Problem` и одноимённый счётчик. Счётчик `Problem` зависит от счётчика раздела. Вывод счётчика `\theProblem` переопределён как номер раздела, за которым следует уже сам счётчик. Внутри окружения счётчик `Problem` увеличивается на единицу с помощью команды `\refstepcounter{счётчик}`. Результат использования нового окружения представлен в следующем примере:

```
\begin{Problem}\label{ex:1}
  Задача раз
\end{Problem}
\begin{Problem}\label{ex:2}
  Задача два
\end{Problem}
Ссылки на раз~\ref{ex:1} и два~\ref{ex:2}.
```

6.2.1 Задача раз
6.2.2 Задача два
Ссылки на раз 6.2.1 и два 6.2.2.

6.3. Создаём свои ...

...команды, окружения и прочее. Наверняка, возникшая в процессе набора, простенькая надоедливая проблема решена и не один раз. С другой стороны при нарастающей квалификации проще бывает изобрести велосипед заново в удобной на текущий момент форме:

```
\newcommand{\ee}{\ensuremath{e^{+}e^{-}}}\xspace}
```

Часто новые команды создаются для комбинаций, используемых исключительно в математическом окружении. Команда `\ensuremath` обеспечивает это окружение независимо от текущего режима:

`\(J/\psi\to\ee\)` является одним из подвидов `\ee{}`"=рассеяния.

$J/\psi \rightarrow e^+e^-$ является одним из подвидов e^+e^- -рассеяния.

Команда `\xspace` из одноимённого пакета **xspace** добавляет в конце команды пробел в случае, если за командой нет знаков препинания, то есть избавляет от необходимости самому вставлять явный пробел после команды.

Имеются три команды, которые позволяют создавать свои или переименовать уже имеющиеся макросы:

```
\newcommand{«команда»}[N][«зн. по ум.»]{«определение»}
\renewcommand{«команда»}[N][«зн. по ум.»]{«определение»}
\providecommand{«команда»}[N][«зн. по ум.»]{«определение»}
```

`\newcommand` определяет новую команду. Если такая команда уже была, то при компиляции генерируется ошибка. `\renewcommand` напротив переопределяет уже существующую команду. В свою очередь `\providecommand` создаёт новую команду, если на момент описания такой команды не было, и ничего не делает, если она уже существовала.

В каждом из этих макросов есть два обязательных параметра — это имя команды и её описание. Если команде необходимо передать параметр/параметры, то первый необязательный аргумент должен принять значение от одного (1) до девяти (9). В разделе 4.4 обсуждалась команда для дублирования знака в формуле при переносе её на следующую строку (`\(a + b \hm{=} c\)`):

```
\newcommand*{\hm}[1]{#1\nobreak\discretionary}{%
  {\hbox{$\mathsurround=0pt #1$}}}{}}
```

Вместо знака решётки (#) с цифрой после него при компиляции макроса подставляется соответствующий параметр. В данном случае параметр был только один, и можно сказать, что его значение сохраняется в «переменной» #1.

Звёздочка (*) в конце макроса `\newcommand` налагает на передаваемый параметр команды `\hm` дополнительное условие: в передаваемом тексте не должно быть пустых строк и команды `\par`. Иногда это упрощает отладку кода.

Наличие второго необязательного параметра позволяет первый параметр определить как параметр по умолчанию:

```
\newcommand{\exmpl}[1][умолчанию]%
  {<<значение по #1>>}
Сравните \exmpl{} и \exmpl[требованию].
```

Сравните «значение по умолчанию»
и «значение по требованию».

Для определения нового окружения используется команда `\newenvironment`:

```
\newenvironment{outlined}%
  {\hrule\smallskip\begin{center}}%
  {\end{center}\smallskip\hrule}
\begin{outlined}
Выделенный текст.
\end{outlined}
```

Выделенный текст.

Формальное описание этой команды похоже на описание `\newcommand`:

```
\newenvironment{«окружение»}[N][«зн. по ум.»]%
  {«код, открывающий окружение»}%
  {«код, закрывающий окружение»}
```



Рис. 6.1. Определение ширины (width), высоты (height) и глубины (depth)

Точно так же, как и в случае `\newcommand`, созданному окружению можно передавать параметры. Подставлять параметры можно только там, где в описании находится текст «код, открывающий окружение». Кроме команды создания нового окружения можно так же переопределять уже имеющиеся окружения с помощью аналогичной команды `\renewenvironment`.

6.3.1. Инструменты

Л^AT_EX — это надстройка над T_EX, и по-хорошему, прежде чем писать свои сложные макросы, следует проштудировать «Всё про T_EX» [1] от мастера. Но уж очень это непростое занятие, требующее значительных умственных и временных затрат. К счастью, в Л^AT_EX есть набор макропакетов, который облегчает жизнь «Л^AT_EX-программиста-новичка».

6.3.2. Вычисления с calc

В дополнение к стандартным возможностям пакет **calc** из коллекции **tools** расширяет базовые операции с длинами и счётчиками. Фактически **calc** вводит арифметические операции в привычной со школы инфиксной записи:

Было до <code>\theMyCount.\par</code>	Было до 1555.
<code>\setcounter{MyCount}{\value{MyCount}-1000}</code>	Стало после 555.
Стало после <code>\theMyCount.</code>	

При загрузке **calc** `\setcounter`, `\addtocounter`, `\setlength` и `\addtolength` переопределяются так, что в качестве аргумента можно передавать арифметические выражения.

<code>\setlength{\MyLength}{</code> <code>(1em+\widthof{очень длинная фраза})*\real{1.2}}</code> <code>\centering</code> <code>\framebox[\MyLength]{очень длинная фраза}\par</code> <code>\framebox[\MyLength]{короткая фраза}</code>	<table border="1"><tr><td>очень длинная фраза</td></tr><tr><td>короткая фраза</td></tr></table>	очень длинная фраза	короткая фраза
очень длинная фраза			
короткая фраза			

Кроме арифметических операций в **calc** определяются макросы `\widthof{текст}`, `\heightof{текст}` и `\depthof{текст}` — ширина, высота и глубина текста, переданного команде.

Если используется пакет **calc**, то при сложении и вычитании данные должны быть однотипными, а при умножении длины на число длина должна стоять до числа ($4\text{mm} * 2$ — верно, а $2 * 4\text{mm}$ — не верно). Делить и умножать можно только на целые числа. Действительные числа вводятся с помощью макроса `\real` и отношения длин, вычисляемого с помощью команды:

```
\ratio{«длина»}{«длина»}
```

Подробное описание пакета можно найти в его документации **calc.pdf**.

6.3.3. Условные операторы и циклы

Простой условный оператор предоставляется пакетом **xifthen**, который является улучшенной версией стандартного **ifthen**. Обучим считать компьютер от одного до трёх по-русски:

```
\newcommand{\RusCnt}[1]{%
  \ifthenelse{\equal{#1}{1}}{один}{%
    \ifthenelse{\equal{#1}{2}}{два}{%
      \ifthenelse{\equal{#1}{3}}{три}{%
        \ifthenelse{\cnttest{#1}>{3}}{много}{%
          \RusCnt{1}, \RusCnt{2}, \RusCnt{3} и \RusCnt{4}.
        }
      }
    }
  }
}
```

раз, два, три и много.

В качестве первого аргумента макрос `\ifthenelse` принимает тестовое условие, в случае его выполнения, выполняются команды второго аргумента, а в случае не выполнения — третьего.

В примере сравнивается полученная строка текста с цифрами «1», «2» и «3» с помощью макроса `\equal` — это с точки зрения команды строки. Для того чтобы работать с этими строками как с числами, в четвёртом операторе `\ifthenelse` используется тест `\cnttest{число}сравнение{число}`. Этот тест в качестве аргумента принимает строку в нотации пакета **calc**. Результатом выполнения каждого теста является булево значение.

Стандартные тесты, определённые в пакете **ifthen**.

- Сравнение целых чисел (`\value{счётчик}` — тоже целое число) с помощью операторов `=`, `>` и `<`.
- `\isodd{целое число}` — это число нечётное?
- `\isundefinedcommand{имя команды}` — эта команда не определена?
- `\equal{строка 1}{строка 2}` — эти строки равны?
- `\boolean{булево значение}` — это истина?
- `\lengthtest{длина 1 = длина 2}` — эти длины равны? Вместо знака равенства можно вставить знак больше `>` и меньше `<`.

При составлении тестовых условий можно использовать `\(` и `\)` в качестве символов группировки выражений, `\AND` — как логическое И, `\OR` — как ИЛИ и `\NOT` — как логическое отрицание.

В **xifthen** определены дополнительные тесты.

- `\isnamedefined{имя команды}` — эта команда определена?
- `\isemptyаргумент` — аргумент пустой?
- `\isequivalentto{команда 1}{команда 2}` — эти команды эквивалентны?
- `\cnttest{выражение 1}сравнения{выражение 2}` — сравнение двух **calc**-подобных числовых выражений.
- `\dimtest{выражение 1}сравнения{выражение 2}` — сравнение двух **calc**-подобных выражений типа «длина».

Также в пакете добавлена возможность определения своих тестов. Для выяснения подробностей следует посмотреть в документацию **xifthen.pdf**.

Кроме условных выражений в пакете **ifthen** определён цикл `\whiledo`:

```
\def\Source{ Это , короткий , тест }
\whiledo{\not\equal{\Source}{}}
{ \GetTokens{TokenOne}{TokenTwo}{\Source}%
  \fbox{\TokenOne}%
  \let\Source\TokenTwo}
```

Это

короткий

тест

Первый аргумент команды цикла представляет из себя тестовое выражение, а второй — действие, которое выполняется, пока первый аргумент не ложный.

Для создания примера использовался макрос `\GetTokens` из пакета **tokenizer**, который разбирает текстовый список. По умолчанию разделителем такого списка является запятая. Первому аргументу макроса присваивается первый элемент списка, второму — остаток, а третьему передаётся сам список. Подробности в файле **tokenizer.pdf**.

Вёрстка

Автору сравнительно легко потратить немного лишнего времени на какие-то детали, потому что он потратил гораздо больше времени на написание книги.

Дональд Э. Кнут

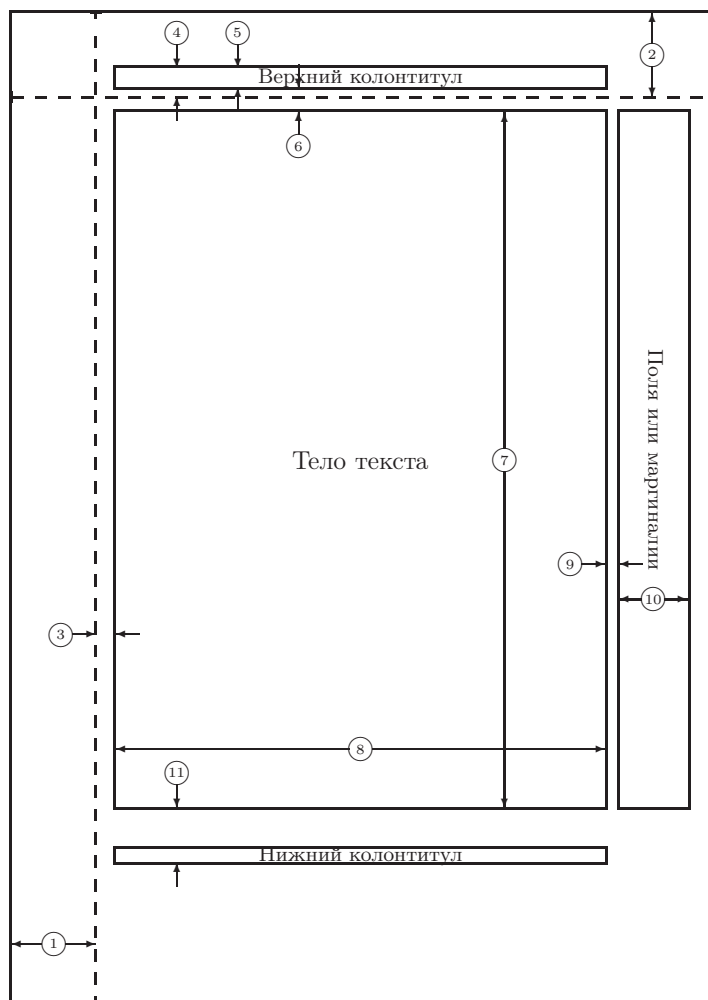
Волшебных текстовых процессоров не существует. Телепатией программы пока не обладают. Они, естественно, делают то, что им сказано, но вкус и чувство прекрасного у них полностью отсутствует. В конце занимательного приключения по созданию текстов частенько приходится брать управление в свои руки, дабы навести лоск на почти готовое произведение.

Вёрстка — составление страниц (полос) газеты, журнала, книги определённого размера из набранных строк, заголовков, иллюстраций и тому подобного в соответствие с разметкой или макетом. В этой главе разберёмся с тем, как задавать размеры, что такое макет полосы набора и как «удерживать» текст в рамках дозволенного.

Здесь присутствует далеко не вся информация, необходимая при вёрстке текста. Несмотря на то, что `LaTeX` позволяет верстать книги любителям без помощи профессионалов, но лучше при любой возможности спрашивать у этих профессионалов совета. Здесь очень важно понимание *что, где и зачем* надо исправлять, потому что способ решения *уже* известной проблемы, как правило, очевиден.

7.1. Скелет страницы

На рис. 7.1 приведён результат выполнения команды `\layout` из одноимённого пакета. Основное место на странице занимает текст — это тело текста.



- | | | | |
|----|-----------------------|----|------------------------------------|
| 1 | один дюйм + \hoffset | 2 | один дюйм + \voffset |
| 3 | \oddsidemargin = 17pt | 4 | \topmargin = -25pt |
| 5 | \headheight = 17pt | 6 | \headsep = 20pt |
| 7 | \textheight = 595pt | 8 | \textwidth = 418pt |
| 9 | \marginparsep = 12pt | 10 | \marginparwidth = 59pt |
| 11 | \footskip = 47pt | | \marginparpush = 6pt (не показано) |
| | \hoffset = 0pt | | \voffset = 0pt |
| | \paperwidth = 597pt | | \paperheight = 845pt |

Рис. 7.1. Макет полосы набора класса `scrartcl` с опцией `a4paper` (результат выполнения команды `\layout` из пакета **layout**)

Справа и слева от текста расположены поля. Поля обычно остаются пустыми, но иногда они используются для заметок (маргиналий или фонариков). В верхней и нижней части страницы расположены соответственно верхний и нижний колонтитулы. Колонтитул представляет собой справочную строку, помогающую ориентироваться в структуре текста.

Совокупность размеров и расположений указанных полей, а также вид и содержание колонтитулов называется *макетом полосы набора*. На рис. 7.1 пунктирной линией изображены *поля драйвера* (① и ②). Относительно этих базовых полей выстраиваются все остальные поля. По договорённости отступы до полей драйвера равны одному дюйму. Переопределив `\hoffset` и `\voffset` (по умолчанию они равны нулю), можно легко сдвинуть полосу набора целиком по горизонтали и вертикали соответственно.

Далее перечислены параметры, которые управляют макетом полосы набора.

- Тело текста характеризуется высотой `\textheight` (⑦) и шириной `\textwidth` (⑧). При многоколоночной вёрстке ширина колонки равна `\columnwidth`. Переменная `\linewidth` принимает значение, равное длине строки текущего текста.
- `\oddsidemargin` (③) добавляется слева в случае односторонней печати. При двухсторонней печати полосы набора для чётных и нечётных страниц различаются. В этом случае для нечётных страниц слева опять же добавляется `\oddsidemargin`, а для чётных — `\evensidemargin`.
- Верхний колонтитул располагается на расстоянии `\topmargin` (④) от поля драйвера, имеет высоту `\headheight` (⑤), а тело текста отступает от колонтитула на расстояние `\headsep` (⑥).
- `\footskip` позиционирует базовую линию нижнего колонтитула относительно последней строки текста.
- Поля для заметок имеют ширину `\marginparwidth` (⑩) и отступают от тела текста на расстояние `\marginparsep` (⑨). Ещё одна опция управляет минимальным расстоянием между заметками: `\marginparpush`.

7.1.1. Выбор размера бумаги

Размер листа бумаги описывается переменными `\paperwidth` и `\paperheight`. Стандартные базовые классы ЛАТ_EX (**article**, **book**, **report** и **letter**) по умолчанию предполагают, что для печати используется бумага формата **letter**. Очевидно это умолчание не годится для России, где стандартом является формат А4 (210 × 297 мм). Обычно установить правильный формат можно с помощью передачи параметра **a4paper** при выборе класса документа:

```
\documentclass[a4paper,12pt,oneside]{scrbook}
```


режима `\rotboxpages` и `\endrotboxpages`, которые указывают границы повернутого вверх ногами текста. Пакет умеет разворачивать не только страницы, но и колонки. Подробности в документации `rotpages-doc.pdf`

7.2. Меняем макет

Причина, по которой неискушённый Т_ЭХник начинает менять макет полосы набора, обычно заключается в том, что он использует стандартные классы Л^АT_ЭX: один из четвёрки **article**, **book**, **report** или **letter**. Это очень древние классы, и на них много чего «завязано», поэтому они прибывают в абсолютно замороженном состоянии. Лучше для начала найти себе другой базовый класс «по душе». Например, **scrartcl**, **scrbook**, **scrreprt** и **scrlettr** — классы из коллекции КОМА-Script, которые, дублируя функциональность стандартной четвёрки, ориентируются на европейских пользователей и размер листа А4. А. И. Роженко в рамках коллекции **ncclatex** (N_CC) предоставляет класс **ncc**, ориентированный на русских Т_ЭХников.

7.2.1. Двигаем размеры

Выбор, естественно, не ограничивается упомянутыми ранее классами — он огромен. Поэтому прежде чем что-то изменять, следует внимательно изучить уже имеющиеся решения. И даже если вы знаете что делаете, лучше не переопределять вручную переменные, управляющие размерами макета полосы набора. Правильным решением будет воспользоваться одним из уже имеющих специализированных пакетов, например: **geometry**, **vmargin** или **typearea**.

► Ещё раз: «Макет лучше не менять!»

geometry

Пакет **geometry** меняет размеры прямо в процессе загрузки стилевого файла, например так:

```
\usepackage[height=25cm,a4paper,hmargin={3cm,2cm}]{geometry}
```

Разбор параметров выполняется с помощью пакета **keyval**, который уже упоминался в главе 5. В качестве параметров можно передавать выражения, если загружен пакет **calc**.

Список воспринимаемых пакетом опций очень обширен. В документации к пакету **geometry.pdf** подробное описание всех имеющихся опций занимает свыше пяти страниц текста. Использование этого пакета для изменения размеров полей предпочтительнее, чем изменение размеров напрямую. Обилие принимаемых параметров позволят выразить любую идею по формату полосы набора в наиболее

естественной форме без ошибок в расчётах. Подробное описание пакета, к сожалению, выходит за все разумные рамки объёма книги. К счастью, имеется очень хорошая и чрезвычайно подробная документация.

vmargin

Пакет **vmargin** управляет размерами макета через выставку полей. Перед выставкой полей следует задать формат листа:

```
\setpapersize [«ориентация»]{«формат листа»}
```

В качестве обязательного параметра команды можно указать один из следующих форматов: A0, A1, ..., A9, B0, B1, ..., B9, C0, ..., C9, USletter, USlegal и USexecutive. Необязательный параметр может принимать значения **landscape** (альбомная ориентация) или **portrait** (портретная ориентация по умолчанию). Для нестандартных форматов листа можно задать размеры с помощью ключевого слова **custom** команды:

```
\setpapersize{custom}{«ширина»}{«высота»}
```

После того как удалось определиться с размером страницы, можно задавать размеры для полосы набора с помощью одной из следующих команд:

```
% Полоса набора с колонтитулами
\setmargins{«ширина поля слева»}{«высота поля сверху»}%
           {«ширина текста»}{«высота текста»}%
           {«высота»}{«отступ»}%верхний колонтитул
           {«высота»}{«отступ»}%нижний колонтитул
% Полоса набора без колонтитулов
\setmargnohf{«ширина поля слева»}{«высота поля сверху»}%
            {«ширина текста»}{«высота текста»}%
% Установка размеров без изменения колонтитулов
\setmarg{«ширина поля слева»}{«высота поля сверху»}%
        {«ширина текста»}{«высота текста»}%
```

Кроме перечисленных команд в **vmargin** определены их аналоги: `\setmarginsrb`, `\setmargnohfrb`, `\setmargrb`. Отличие этих команд от вышеупомянутых заключается в том, что в качестве параметров вместо ширины и высоты тела текста им передаются ширина правого поля и высота нижнего поля. Подробно пакет описан в документации **vmargin.pdf**.

typearea

Стиль **typearea** из всеобъемлющего пакета **KOMA-Script** создавался специально под классы, определённые в этом пакете (**scrartcl**, **scrbook** и т. д.),

но может использоваться и отдельно. Параметры макета по умолчанию вычисляются автоматически по правилам, сформулированным Яном Чихольдом (Jan Tschichold).

typearea используется примерно так же, как **geometry**, то есть параметры макета можно передать при инициализации:

```
\usepackage [BCOR8.25mm] { typearea }
```

BCOR (binding correction) — ширина, идущая на переплёт или выпадающая из полезной площади по какой-то другой причине.

Если в качестве класса документа выбран один из семейства **KOMA-Script**, то опции, передаваемые **typearea**, следует объявлять прямо во время инициализации класса:

```
% Загружается класс из KOMA-Script
\documentclass [ pagesize ] { scrbook }
% Устанавливается размер страницы 70x100M/16
\setlength { \paperwidth } { 170mm }
\setlength { \paperheight } { 240mm }
% Устанавливается размер полосы
\areaset { 130mm } { 194mm }
```

Опция **pagesize** указывает пакету **typearea**, что при компиляции необходимо добавить дополнительные управляющие команды, для того чтобы dvi-драйверы (такие как **dvips**, **xdvi** и так далее) правильно определяли нестандартные размеры бумаги. Иначе размер бумаги, отличный от умолчания, придётся задавать при исполнении этих программ.

Формат издания, установленный ГОСТ 5773-90, приводит к тому, что размер страницы отличается от международных стандартов (например, от A1, A2, A3, A4 и так далее). За размер бумаги отвечают переменные `\paperwidth` и `\paperheight` — ширина и высота страницы соответственно. Для установки размеров полосы текста в пакете **typearea** определена команда `\areaset`, первый обязательный аргумент которой соответствует ширине текста, а второй — высоте.

Подробности можно найти в описании **KOMA-Script** ([scrguien.pdf](#)) в главе «Construction of the Page Layout with typearea». Перед использованием *придётся* прочитать инструкцию.

typogrid

Для визуального контроля при создании макета, возможно, будет удобно воспользоваться пакетом **typogrid**, который наносит на лист типографскую решётку. Решётка наносится только в случае, если в качестве параметра пакету или классу документа передаётся опция **draft**, и соответственно опция **final** отменяет действие пакета.

Пакету можно передать число колонок, которые должна сформировать решётка, с помощью опции `columns=число`. По умолчанию число равно 12.

7.2.2. Стили страницы

Полоса набора — это не только размеры, это ещё и наполнение колонтитулов. Обычно наполнение колонтитулов определяется в классе документа. Лучше без необходимости ничего не менять.

Простейший способ изменить стиль страницы — это воспользоваться командой:

```
\pagestyle{«стиль страницы»}
```

Если стиль надо переопределить только для текущей страницы, то следует воспользоваться командой `\thispagestyle{стиль страницы}`.

Есть три стандартно определённых стиля:

- `empty` — страница выводится без каких-либо колонтитулов — только текст.
- `plain` — выводится только номер страницы в нижнем колонтитуле.
- `headings` — в верхнем колонтитуле выводится номер страницы и информация, определяемая классом документа.

Если и это наполнение не устраивает, то можно определить свой собственный стиль. Пакет **fancyhdr**, подробно описанный в разделе 12.2 на странице 166, специализируется как раз на этом.

7.2.3. Буклеты

Если требуется быстро сделать «буклет-раскладушку» формата A5 и есть готовый PostScript со страницами формата A4, то утилиты из программного пакета **psutils**, скорее всего, позволят получить желаемый результат:

```
> cat «исходный файл».ps | psbook | psnup -2 > «буклет».ps
```

Исполняемый файл **psbook** меняет порядок страниц так, чтобы получился двухсторонний буклет, а **psnup** с ключом `-2` размещает по две страницы на одной. Затем получившийся PostScript можно распечатать. Если хочется чего-то более специального, то следует прочитать документацию, идущую к пакету, и, возможно, особенно внимательно приглядеться к исполняемому файлу **pstops**.

Для простого размещения двух страниц на одной можно просто распечатать результирующий pdf-файл, например, с помощью **Adobe Reader** или **kpdf**, соответствующим образом изменив опции печати. Также при работе с pdf-файлами представляет интерес пакет **pdfpages**, позволяющий манипулировать pdf-страницами внешних pdf-документов.

twoup

Пакет **twoup** позволяет решить проблему создания буклетов без внешних утилит. Он зависит от пакета **booklet**, который специализируется на изменении порядка следования страниц, то есть является аналогом **psbook**.

Для использования **twoup** достаточно в преамбуле при загрузке класса указать размер используемой бумаги и загрузить стили **booklet** и **twoup**:

```
\documentclass[a4paper,12pt]{scrbook}
\usepackage[print]{booklet}
\usepackage{twoup}
```

Опция **print** инициализирует реорганизацию страниц.

После компиляции программе **dvips**, возможно, придётся вручную передать размеры листа:

```
> dvips -T 29.6cm,21.0cm имя dvi-файла
```

Теперь получившийся буклет можно печатать

pgfpages

pgfpages из пакета **pgf** представляет собой набор макросов, позволяющих автору разместить несколько страниц текста на одной без использования сторонних программ. **pgfpages** перехватывает управление у **TeX** в момент окончания формирования страницы и размещает её внутри бокса, не выходя за пределы инструкций **TeX**. Из недостатков **pgfpages** следует упомянуть, что с этим пакетом перестают работать гиперссылки.

Для того чтобы разметить две страницы, можно набрать следующий код:

```
% Преамбула
\usepackage{pgfpages}
\pgfpagesuselayout{2 on 1}[a4paper,landscape,%
                        border shrink=5mm]

\begin{document}
Текст на левой странице
\clearpage
Текст на правой странице
\end{document}
```

Макросу `\pgfpagesuselayout` передаётся команда о размещении двух страниц на одной (2 on 1). Также макросу передаётся список опций:

a4paper — результирующая страница должна быть размера A4,
landscape — ориентация результирующей страницы альбомная,
border shrink=5mm — сформировать отступ от краёв в 5 мм.

Подробности следует искать в документации к **pgf** ([pgfmanual.pdf](#)) в главе «Page Management».

«Коридоры» графически разделяют текст абзаца или полосы на некоторое подобие неожиданных колонок, а они могут осложнять восприятие текста, из-за чего «коридоры» надо устранять как дефект набора.

Рис. 7.4. Дефект набора «коридор» (пример взят из [11])

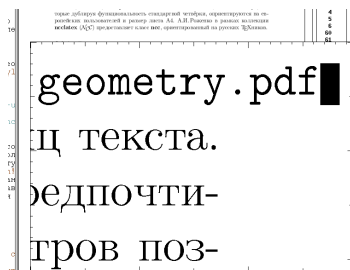


Рис. 7.5. Переполненная строка в `xdvi` при включённой опции `draft`

7.3. «Причёсываем» текст

После того как границы определены, посмотрим, что можно сделать для красивого размещения текста на странице. Часть забот по следованию правилам вёрстки ЛАТЭХ берёт на себя. Например, вовсе не требуется следить за единообразием оформления полос. Но есть дефекты, которые компьютеру заметить не под силу, например, так называемые коридоры, один из которых демонстрируется на рис. 7.4.

7.3.1. Строка

Когда строка с точки зрения Т_ЕX становится слишком разряженной (Underfull) или слишком сжатой (Overfull), то в log-файле появляются предупреждения, начинающиеся со слов вида:

```
Overfull \hbox (26.2pt too wide) in paragraph at lines 47--56
[[ ]]\T2A/cmr/m/n/12 Список вос-при-ни-ма-е-мых па-ке-том
оп-ций очень об-ши-рен. В до-ку-мен-та-ции geometry.pdf
```

Если при выборе класса документа передать ему опцию `draft`:

```
\documentclass [ draft , a4paper , 12pt , oneside ] { scrbook }
```

то такие проблемные места будут отмечаться в результирующем документе. На рисунке 7.5 приведён пример такой отметки.

Обычно подобные переполнения связаны с тем, что ЛАТЭХ не знает, как перенести какое-либо слово. В этом случае следует обучить его, что и где можно переносить, как это было показано в главе 3. В крайнем случае можно насильно разорвать строку с помощью команды `\linebreak` или `\.`. В отличие от `\linebreak` команда `\.` не заставляет остаток строки выравниваться по правому полю.

Если можно редактировать текст, то для исправления дефектов набора лучше переделать предложение так, чтобы в новой инкарнации текст не создавал проблем для чтения.

Note Пакет **microtype** (микротипографика) позволяет улучшить качество плотного набора за счёт изменения межбуквенных расстояний. Пакет работает только совместно с **pdflatex**, так как является интерфейсом к одному из реализованных в этом пакете расширений к стандартному **TeX**. Пакет также не работает со шрифтами в T2A, то есть с кириллицей в частности. Автор осознаёт эту проблему и будет благодарен за любую помощь. Подробности в `microtype.pdf`.

7.3.2. Горизонтальные пробелы

Расстояние между словами можно изменить с помощью горизонтальных промежутков. Горизонтальные промежутки создаются с помощью команды `\hspace`. В качестве параметра команде передаётся длина. Существует также *-вариант команды `\hspace*`, который отличается от основной тем, что создание пробела не игнорируется даже тогда, когда пробел приходится на начало или конец строки.

Существует несколько определённых по умолчанию горизонтальных пробелов: `\quad` — горизонтальный промежуток шириной `1em`. Также есть `\qqquad` — удвоенный `\quad`, и `\endspace` — половина от `\quad`.

`\hfill` — бесконечный горизонтальный промежуток. Два `\hfill` подряд в два раза больше, чем один. Также есть более «маленькая» бесконечность — `\hfil`.

`\hrulefill` — то же, что и `\hfill`, но заполненный промежуток подчёркивается. Есть аналогичная команда `\dotfill`, которая заполняет всё точками.

7.3.3. Форматирование параграфа

Обычно каждый параграф начинается с красной строки. Для её подавления служит команда `\noindent`. Противоположной для неё по смыслу является команда `\indent`.

Для целей позиционирования текста в **LaTeX** предусмотрено три стандартных окружения: `center` (центрирование текста), `flushleft` (выравнивание по левому краю) и `flushright` (выравнивание по правому краю). Каждому из этих окружений соответствуют одноимённые декларации.

```
\begin{center}
```

Выравниваемся по центру.

```
Выравниваемся по центру.
```

```
\end{center}
```

А теперь по левому краю.

```
\flushleft А теперь по левому краю.
```

```
\flushright А теперь по правому.
```

А теперь по правому.

Упомянутые команды центрирования формируют абзац и вставляют вертикальный пробел перед ним. Пробел перед абзацем не всегда нужен, поэтому есть

соответствующие команды, которые также выравнивают текст, но пробел не добавляют: `\centering` (выравнивание по центру), `\raggedright`² (по левому краю) и `\raggedleft` (по правому краю).

Перечисленные стандартные способы выравнивания текста обладают одним неприятным свойством — при использовании этих механизмов отключаются переносы. Пакет **ragged2e** определяет макрос `\RaggedRight`, который обходит эту проблему и позволяет выравнивать по левому краю без отключения переносов:

<code>\begin{minipage}[t]{0.48\textwidth}</code>		
<code>\raggedright</code> Длинное предложение	Длинное предложение	Длинное предложение должно переноситься нормально.
должно переноситься нормально		
<code>\end{minipage}\rule[-1.8cm]{0.4pt}{2cm}</code>	должно	
<code>\begin{minipage}[t]{0.48\textwidth}</code>	переноситься	
<code>\RaggedRight</code> Длинное предложение	нормально	
должно переноситься нормально.		
<code>\end{minipage}</code>		

Из примера видно что там, где была применена процедура `\RaggedRight`, колонка получилась более компактной.

7.3.4. Страница

Проблемы с вёрсткой могут возникнуть и при формировании страниц. В крайнем случае всегда можно воспользоваться командами принудительного завершения страницы `\newpage` или `\pagebreak`. `\newpage` просто завершает страницу, а при выполнении `\pagebreak` после формирования страницы полоса выравнивается по нижней кромке — это может привести к неоправданному растяжению страницы.

Если проблему вёрстки можно решить путём увеличения/уменьшения страницы на одну-две строки, то лучше воспользоваться следующими макросами:

```
\newcommand{\longpage}{\enlargethispage{\baselineskip}}
\newcommand{\shortpage}{\enlargethispage{-\baselineskip}}
```

Команда `\longpage` увеличивает тело текста текущей страницы на одну строку, а `\shortpage`, соответственно, уменьшает. Длина `\baselineskip` служит для определения *интерлиньяжа* или междустрочного пробела.

7.3.5. Висячая строка

Одним из самых неприятных дефектов набора является «висячая строка». Висячая строка — это концевая строка абзаца, стоящая первой на странице, или

²`\raggedright` соответствует `\flushleft` и дословно можно перевести как неровный справа.

начальная строка абзаца, стоящая на странице последней. Таких артефактов следует всячески избегать. Для подавления этого эффекта в заголовке документа следует переопределить две переменные:

```
\clubpenalty=10000
\widowpenalty=10000
```

7.3.6. Вертикальные просветы

По аналогии с командой `\hspace{длина}` вертикальные промежутки создаются с помощью команды `\vspace{длина}`. Модификация команды `\vspace*{длина}` организует вертикальный просвет, который не игнорируется, даже если просвет попадает на начало или конец страницы.

Вертикальные просветы также имеют предопределённые команды:

`\bigskip` — вертикальный промежуток, равный примерно `\baselineskip`. Существуют `\medskip` — половина от `\bigskip` и `\smallskip` — $1/4$ от `\bigskip`.
`\vfill` — бесконечный вертикальный промежуток. Два `\vfill` подряд в два раза больше, чем один. Также есть более «маленькая» бесконечность — `\vfil`.

7.3.7. Печать через две строки

До сих пор временами встречаются требования вида: «Предоставить диплом, напечатанный через два интервала» — пережиток эпохи печатных машинок. Для решения этой проблемы лучше всего воспользоваться пакетом **setspace**. В пакете определена команда `\doublespacing`, которая выполняет искомое действие.

В **setspace** определены макросы `\onehalfspacing` и `\singlespacing` — печать через полтора и один интервал соответственно. Для вертикальной разрядки небольшого фрагмента текста лучше воспользоваться одноимёнными окружениями или окружением `spacing`:

```
\begin{spacing}{2.5}                                «Этот текст напечатан с интервалом
<<Этот текст напечатан с
интервалом в две с
половиной строки>>.                                в две с половиной строки».
\end{spacing}
```

В качестве основного параметра окружению `spacing` передаётся число строк, через которое следует печатать текст.

7.4. Многоколоночная вёрстка

Стандартные классы воспринимают при инициализации опцию `twocolumn`:

```
\documentclass[a4paper,12pt,twocolumn]{scrartcl}
```

На рис. 7.6 приведён пример действия этой опции. К двухколоночной печати можно перейти и по ходу текста с помощью команды `\twocolumn[заголовок]`. При этом совершается переход на следующую страницу, выводится заголовок, если он есть, а затем весь последующий текст формируется в две колонки. Для перехода в одноколоночный режим следует воспользоваться командой `\onecolumn`, при этом опять сначала происходит переход на следующую страницу. Стандартными средствами совместить на одной странице две и одну колонки нельзя. Для этого следует воспользоваться специальными пакетами.

Обычный двухколоночный режим имеет массу недостатков, в частности, как видно из рис. 7.6, по умолчанию на последней странице колонки не сбалансированы. Стиль **balance** из пакета **preprint** исправляет эту ошибку. Результат представлен на рис. 7.7. Для этого необходимо загрузить пакет и активировать балансировку командой `\balance`.

Иногда удобно все сноски со страницы разметить в конце правой колонки, как на рис. 7.7. Для этого достаточно воспользоваться стилем **ftnright** из пакета **tools**. При этом знак сноски помещается сразу в строке, а разделительная линия отсутствует. Подробности можно найти в файле **ftnright.pdf**

► В пакете **fixltx2e** исправлено некоторое количество неприятностей, связанных с двухколоночной вёрсткой. В частности исправлена ошибка формирования колонтитулов (по умолчанию колонтитул определяется второй колонкой, а не первой) и решена проблема нумерации иллюстраций, когда широкая иллюстрация оказывалась позади одноколоночной. Надо только загрузить его.

sttools

Пакет **sttools** специализируется на решении проблем, возникающих при стандартной двухколоночной вёрстке. К сожалению, пакет довольно стар и давно не обновлялся, хотя явно следовало бы.

При загрузке стиля **flushend** включается правильная балансировка колонок, которую при желании можно отключить с помощью команды `\raggedend` и вернуть обратно с помощью команды `\flushend`.

Стиль **cuted** определяет окружение `strip`, которое позволяет перейти к одноколоночной вёрстке, а затем вернуться обратно без перехода на другую страницу. Пример действия окружения `strip` показан на рис. 7.8. При определении окружения обратите внимание, что между предыдущим и последующим параграфами и самим окружением должны присутствовать пустые строки.

Все остальные подробности об этом пакете можно найти в файле **presfull.pdf**.

multicol

Стандартный стиль **multicol** из пакета **tools** в отличие от стандартной двухколоночной вёрстки позволяет совмещать на одной странице многоколоночную

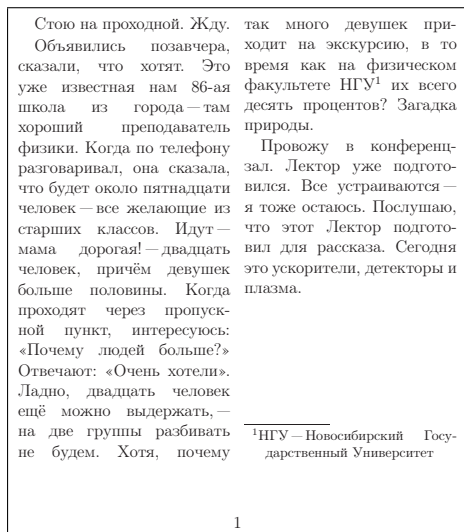


Рис. 7.6. Действие опции **twocolumn**, колонки не сбалансированы

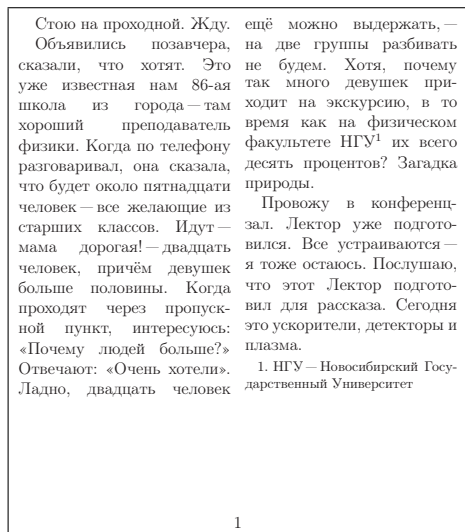


Рис. 7.7. Колонки сбалансированы с помощью пакета **balanced**, а подстрочное примечание сформировано с помощью стиля **ftnright**

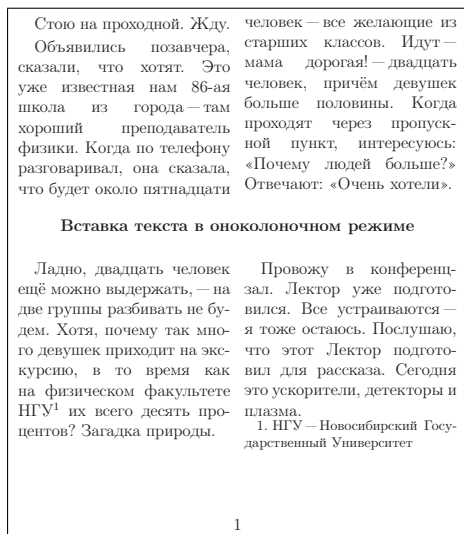


Рис. 7.8. Колонки сбалансированы с помощью стиля **flushend**: вставка в центре осуществляется с помощью окружения **strip**, определённо в стиле **cuted**

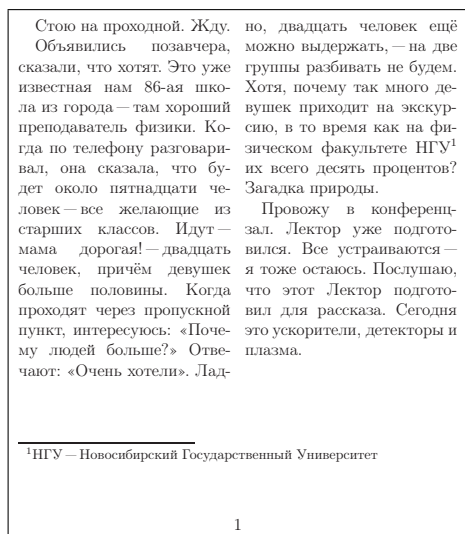


Рис. 7.9. Пакет **multicol**: весь текст заключён в окружение **multicols** с параметром 2

и одноколонную вёрстку. Также балансировка колонок на последней странице производится автоматически и можно задать до десяти текстовых колонок.

В пакете определено окружение `multicols`:

```
\begin{multicols}{3}
  [\section*{Экскурсия}]
\RaggedRight
Стою на проходной. Жду.
```

```
Объявились позавчера, сказали,
что хотят. Это уже известная
нам 86-ая школа из города "---
там хороший преподаватель
физики.
\end{multicols}
```

Экскурсия

Стою на про- ходной. Жду.	хотят. Это уже извест- ная нам 86- ая школа из	хороший пре- подаватель физики.
Объявились позавчера, сказали, что	города — там	

► Если колонки слишком узкие, то имеет смысл выравнивать только по левому краю, а не по обоим, что и делает в этом примере команда `\RaggedRight` (подробности в разделе 7.3.3 на стр. 89).

В качестве обязательного параметра окружению `multicols` передаётся число колонок. Первый необязательный параметр принимает команды, которые исполняются непосредственно перед началом многоколоночной вёрстки. Эти команды исполняются в одноколонном режиме, что позволяет эффективно совмещать одноколонный и многоколоночные режимы.

Если по какой-то причине какую-либо колонку необходимо оборвать и в то же время продолжить следующие, то для этого в пакете **multicol** определена команда `\columnbreak`.

Плавающие объекты внутри окружения `multicols` поддерживаются лишь частично. Они могут возникать либо в верхней, либо в нижней части страницы, иными словами для картинок и таблиц действуют только опции `t` и `b`. При этом они оформляются так же, как и в случае одноколонного режима. Подстрочные примечания формируются как при одноколонной вёрстке. Все особенности использования пакета описаны в файле `multicol.pdf`.

parallel

Пакет **parallel** используется для параллельной печати текстов. В пакете определено окружение `Parallel`:

```
\begin{Parallel}{0.4\textwidth}{0.5\textwidth}
  \ParallelLText{First they ignore you, then they laugh at you,
    then they fight you, then you win.}
  \ParallelRText{Сначала они тебя не замечают, потом смеются
    над тобой, затем борются с тобой. А потом ты побеждаешь.}
\end{Parallel}
```

Результат от исполнения кода представлен ниже:

First they ignore you, then they
laugh at you, then they fight you,
then you win.

Сначала они тебя не замечают, потом
смеются над тобой, затем борются с то-
бой. А потом ты побеждаешь.

Первый обязательный параметр соответствует ширине левой колонки, а второй — правой. Команда `\ParallelLText` формирует левую колонку текста, а комплементарная ей `\ParallelRText`, соответственно, правую. Для выяснения более подробной информации следует обратиться к описанию `parallel.pdf`

Глава 8

Путеводитель по классам L^AT_EX

Однако для многих людей L^AT_EX является более простой системой, в нём автоматизированы те вещи, которые люди всегда хотят автоматизировать.

Дональд Э. Кнут

Класс документа — это первое, что требуется указать при наборе. В то же время *первое* вовсе не значит *важное*. Заключительный выбор класса почти всегда можно отложить до окончания основного набора. С другой стороны, выбирать всё равно придётся.

8.1. Зачем нужны эти классы?

Класс документа выбирается с помощью команды `\documentclass`. Её нужно и можно выполнить ровно один раз в самом начале документа:

```
\documentclass[a4paper,12pt]{scrartcl}
```

В качестве обязательного аргумента указывается имя класса, которому через запятую передаются необязательные параметры. Класс определяется в файле с расширением `cls`. В дистрибутиве T_EX Live 2007 присутствуют 208 уникальных файлов с подобным расширением. Естественно, это далеко не все существующие на белом свете классы. Тем более что никто не мешает создать свой личный класс. Но для начала лучше воспользоваться одним из *уже* имеющихся.

Класс определяет вид и структуру документа. Класс — это база, которую можно править с помощью подключаемых стилевых файлов. В классе задаётся геометрия страницы и определяются команды секционирования. Класс может быть

просто небольшой модификацией уже имеющегося класса, а может стать принципиально новой реализацией представления печатного или электронного T_EXнического слова.

В качестве примера последнего можно привести молодой (2007 год) пока пакет **paper_{tex}** (рис. 8.1 — пример от автора класса Ignacio Llopis), который позволяет применять L^AT_EX в деле вёрстки газет¹, то есть для того, для чего L^AT_EX в проекте вовсе не предназначался.

8.2. Классовая база



Рис. 8.1. **paper_{tex}**

докладов **proc** в свою очередь является небольшой модификацией класса **article**, причём основное отличие состоит в обязательном использовании двухколоночной вёрстки. Для написания писем был создан класс **letter**. Набор команд в этом классе существенно отличается от команд базовых инструкций у уже перечисленных классов. В частности для писем нет нужды в командах секционирования. Класс **slides** — это простой и быстрый способ сделать презентацию. С помощью этого класса не удастся создать пёстрого фона и головокружительных эффектов смены слайдов, зато это позволяет сосредоточиться на главном, а именно непосредственно на тексте.

Особняком от этих классов стоит класс **minimal**, который является болванкой для создания и тестирования новых классов и идей. В классе **minimal** не опреде-

Исторически так сложилось, что L^AT_EX начался с шести (6) классов: **article** (статья), **book** (книга), **report** (отчёт), **proc** (доклад), **letter** (письмо) и **slides** (слайды).

По идее статьи следовало набирать с использованием класса **article**. В этом классе определены команды секционирования вплоть до `\section` (раздел) включительно. Одним из желательных элементов оформления является аннотация (окружение `abstract`). В классе **book** присутствует расширенный набор команд секционирования, в который добавлена команда `\chapter` (глава). Также в **book** присутствует базовый набор команд для оформления титульного листа, предисловия и оглавления. Класс для отчётов **report** является упрощённой версией класса **book**. Отчёты — это те же книги, только читают их по необходимости, а не по желанию. Предназначенный для создания тезисов

¹На текущий момент этот класс не годится для вёрстки чего-нибудь более серьёзного чем школьная газета, но ведь надо начинать с простого.

лено никаких специальных команд — там всё по минимуму. В качестве побочного эффекта документ, в котором в качестве базового выбран этот класс, транслируется ЛАТ_EX значительно быстрее, чем в случае других более специализированных классов.

С этих классов всё начиналось, но не закончилось. Следует понимать, что базовые классы далеко не так хороши, как хотелось бы. После трансляции текста сразу возникает желание взять в руки «электронный надфиль» и пройтись по настройкам класса. Это цена за то, что эти классы являются базой. На них ссылаются и их модифицируют множество других классов и пакетов, поэтому их развитие было заморожено. Для начального набора сгодится и это, но для конечной вёрстки лучше подобрать что-то более подходящее или придётся серьёзно модифицировать значения по умолчанию.

Часто набор необязательных параметров для стандартных классов используется и в других классах, например, в целях совместимости. Некоторые из полезных опций перечислены далее:

10pt, 11pt, 12pt — установка базового размера шрифта. Как правило, этих трёх значений хватает.

a4paper — установка размера листа бумаги. Эту опцию следует использовать всегда, так как по умолчанию стандартные классы ЛАТ_EX используют размер листа **letter**.

draft — режим черновой печати для «отлавливания» проблем вёрстки. В этом режиме не внедряются картинки и отмечаются строчки, где алгоритм разбиения абзаца на строки даёт осечку.

oneside/twoside — форматирование документа для односторонней и двухсторонней печати соответственно.

twocolumn — печать в две колонки.

8.3. Классификация

Число классов постоянно растёт, поэтому не следует думать, что всё исчерпывается классами, перечисленными в этой главе.

8.3.1. Модификации и улучшения базы

Всем не нравятся стандартные классы и всякий старается их улучшить. Кто-то убирает какой-то конкретный недостаток, как это сделано в наборе классов **extsizes** (**extarticle**, **extbook**, **extletter**, **extproc**, **extreport**), которые отличаются от стандартных только возможностью указать базовый размер шрифта отличный от обычного 10-12pt². Есть наборы классов, которые делались с какой-то

²Кроме 10pt, 11pt и 12pt классы из набора **extsizes** поддерживают 8pt, 9pt, 14pt, 17pt и 20pt. Смена размера базового шрифта приводит к принципиально иному дизайну.

определённой целью. Примером такого подхода являются классы из коллекции $\mathcal{AMS}$ (**amsart**, **amsbook**, **amsproc**), которые были предназначены для публикации в журналах Американского математического сообщества. Классы из набора **ntgclass**³ представляют собой «героическую» попытку немецкоговорящих голландцев сделать то же, что и в стандарте, но существенно другим способом.

KOMA-Script

В последнее время всё больше внимания обращает на себя полный набор классов **KOMA-Script**. В этот раз хорошо постарались немцы. Следует учитывать, что европейские традиции полиграфии (в основном французские), всё-таки к нам ближе, чем американские, на которые традиционно ориентировалось L^AT_EX-сообщество. Для статей предполагается использовать **scrartcl**, для книг — **scrbook**, для писем — **scrlettr2**, а для отчётов — **scrreprt**.

В отличие от стандарта классы из **KOMA-Script** позволяют использовать базовые размеры шрифта в 9pt, 14pt и 17pt. Огромные поля, имеющие место в стандартных классах, в классах **KOMA-Script** значительно уменьшены. Претерпели изменения и другие элементы. Если оформление по умолчанию не кажется адекватным, то **KOMA-Script** предоставляет обширный набор высокоуровневых настроек. Подробная документация на более чем двухстах страницах «The KOMA-Script bundle» (**scrguien.pdf**) позволяет подстроить все необходимые параметры.

$\mathcal{NCS}$

Ещё один вариант в качестве замены стандартным классам — это использование пакета $\mathcal{NCS}$. Очень подробно об этом пакете написано в замечательной книге от создателя $\mathcal{NCS}$ Александра И. Рожено «Искусство верстки⁴ в L^AT_EX'e» [10].

Для использования следует загрузить класс **ncc** и передать ему желаемый стиль оформления в качестве параметра: **article** (статья — используется по умолчанию), **preprint** (препринт), **book** (монография) или **report** (отчёт). Дальнейшие подробности об использовании этого класса можно почерпнуть в краткой инструкции к пакету: **ncclatex.pdf**.

8.3.2. Пишем письма

Класс **letter** является стандартным для L^AT_EX, и как следствие, никто им не пользуется. Часто стандарт для написания писем создаётся автором самостоятельно. Так, например, любит делать сам Д. Э. Кнут. Это очень неплохо работает в силу того, что структура письма не очень сложна. Как следствие в L^AT_EX имеется огромное число альтернатив для **letter**.

³В набор **ntgclass** входят классы для набора статей (**artikel1**, **artikel2** и **artikel3**), для набора книг (**boek** и **boek3**), писем (**brief**) и отчётов (**rapport1** и **rapport3**).

⁴Да, да именно «верстки» — букву «ё» опять обидели.

Кроме более-менее общих решений полно и частных. Например, для внутренней переписки университета города Падуа есть специальный пакет **cdpbundl**, содержащий целых три класса.

eskdx — это относительно молодой пакет. Ранее аналогичная попытка была предпринята Вячеславом Фёдоровым. В результате на свет появился пакет **eskd**

Рис. 8.2. Класс `eskdх`

⁵То есть тупая, нудная и не интересная.

(без «х»). В отличие от класса Константина Корикова класс `eskd.cls` требует обязательной установки шрифтов из коллекции **pscyr**⁶.

Стандарт для написания документов, описывающих стандарты. Что может быть ещё более стандартным? Класс `isov2.cls` из пакета **iso** является стандартом для стандартов. Документация к пакету `isoman.pdf` подробно описывает все технические тонкости в деле подготовки документов по стандартам ISO. Для создания документации ISO 10303 есть свой класс **iso10303**.

8.3.4. Верстаем книги

Написание книги — это очень долгий процесс, и первоначальную «набивку» текста можно начать даже со стандартным классом **book**. С другой стороны, структура книги может быть очень сложной, и правильный выбор базового класса позволит несколько облегчить процесс созидания.

В качестве улучшенного стандартного класса **book** можно использовать класс **octavo**. Класс `scrbook` из **KOMA-Script** также является хорошей альтернативой для **book**. Структура и основные команды копируют стандартный класс. Значения же параметров по умолчанию более адекватны для Европейской полиграфии.

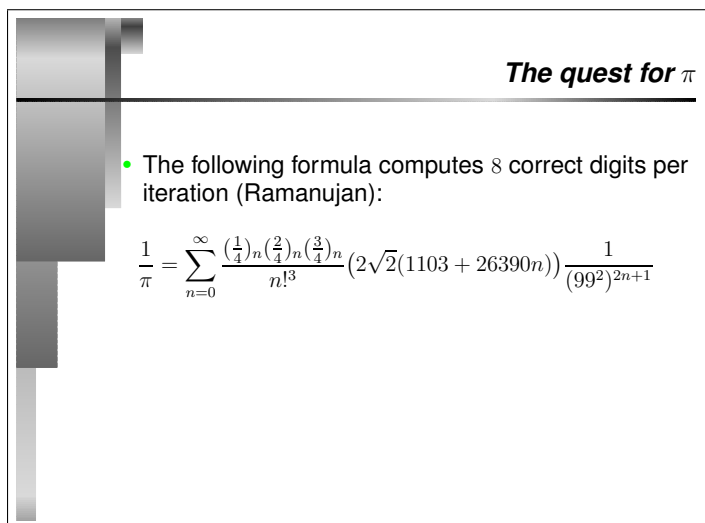
Описание класса **memoir** (`memman.pdf`) представляет собой книгу о создании книги, превышающую по объёму *триста* страниц. Там есть всё, начиная от формальной структуры печатной книги, советов по оформлению электронных копий, заканчивая собственно описанием класса. Все элементы структуры и управляющие размеры показаны в виде рисунков и схем. Класс не является надстройкой над чем-либо — это произведение искусства, созданное с нуля. Документацию следует пролистать хотя бы просто для ознакомления. Класс развивается до сих пор. Новые возможности описываются в дополнении к основной документации (`memmanadd.pdf`), и на текущий момент дополнение почти достигло объёма в сто страниц.

Если не требуется написать книгу, а нужно распечатать мегабайтный текст на дешёвеньком принтере в режиме экономии тонера, то для этого дела вполне может подойти класс **sffms** — простенько и строки через два интервала.

8.3.5. Создаём отчёты

Отчёт — не книга, но и здесь есть свои правила, структура и особенности. Для начала можно воспользоваться **scrreprt** из **KOMA-Script**, как заменой стандартному классу **report**.

⁶Шрифты из коллекции **pscyr** авторами больше не поддерживаются и не развиваются. Основные проблемы этого пакета не технические, а лицензионные. В связи с чем данный пакет отсутствует в дистрибутивах L^AT_EX. В дополнение к абсолютно не решаемым лицензионным там хватает и технических проблем. Последнюю версию этого пакета можно взять по адресу: <ftp://ftp.vsu.ru/pub/tex/font-packs/pscyr/>.

Рис. 8.3. **prosper** в действии

Инженер-электронщик Эли Билауэр (Eli Billauer) сделал $\text{\LaTeX}$ -класс для бумаг в HiTech-стиле и назвал его соответственно **hitec**. Простенько и со вкусом.

Есть, естественно, и частные решения. Хочется заключить контракт с Американским правительством? Стандартная форма 298, обеспечиваемая классом **sffms**, будет весьма кстати. Класс **manual** из пакета **nassflow** даст возможность пообщаться со структурой под названием «Center for the Automation of Weapon and Command Systems, Royal Netherlands Navy».

8.3.6. Делаем презентации

В начале предполагалось, что цель презентации в распространении нужной информации от одного человека ко многим. Поэтому во главу угла ставился текст, а «украшательства» сводились к простой рамке. Стандартные классы **seminar** и **sides** вполне годились для этого.

Но время суровых докладчиков прошло и «рюшечки» вышли на первый план. $\text{\LaTeX}$ может предоставить и «рюшечки», но лучше всё-таки помнить о смысле.

Класс **prosper** (рис. 8.3) создан как улучшенный пакет **seminar** и поддерживает не только оверлеи, гиперссылки и шаблоны оформления, но и «стандартный» набор динамических эффектов, доступных через формат pdf. Класс **ppr-prv** позволяет создать печатную версию электронных слайдов класса **prosper**.

В пакет **texpower** входит класс **powersem**, который по сути дела просто загружает **slides**, а всю работу по созданию презентации выполняет стиль **texpower**. Возможности этого пакета сравнимы с **prosper** — просто он немного другой.

В последнее время в деле создания презентации всё популярнее и популярнее становится относительно молодой, но довольно мощный класс **beamer**. Активная поддержка сообщества при создании этого пакета позволила автору **beamer** Тилу Танталу (Till Tantau) собрать в одном месте немало тем для слайдов. Ключевой особенностью этих тем является разнообразие. Наличие широкого выбора стандартных тем позволяет быстро выбрать обрамление для презентации. Более чем 200 страниц документации ускоряет решение любой возникшей проблемы.

Даже если использовать LyX (<http://www.lyx.org>) в качестве WYSIWYM редактора, то и его вполне можно настроить для создания презентаций. Для этого следует воспользоваться ещё одним достаточно молодым, но уже вполне функциональным классом **powerdot**, в комплекте с которым идут настройки для LyX.

Класс **talk**, в отличие от упомянутых ранее пакетов, позволяет пользователю определить более одного стиля слайдов для презентации. Резкая смена стиля во время доклада — иногда нужно и такое.

Для создания настенного постера в первом приближении можно воспользоваться пакетом **aposter**, который позволяет работать с большими форматами бумаги. Канонического класса, который бы решал все проблемы при изготовлении постера, в L^AT_EX на текущий момент нет. Возможно, ближе всего к идеалу подошёл класс **sciposter** из одноимённого пакета.

8.3.7. Журнальные и конференционные классы

Каждый серьёзный научный журнал и крупная конференция имеет свой L^AT_EX-класс. Обычно этот класс лежит где-то на официальном сайте. Например, журналы, издаваемые издательством МАИК «Наука/Интерпериодика», должны следовать правилам, выложенным здесь: <http://www.maik.ru/pub/tex/>. Но довольно много журнальных классов можно найти и в стандартном дистрибутиве L^AT_EX. Например, класс **asaetr** используется в American Society for Agricultural Engineers (ASAE). Maple Technical Newsletter можно создавать с помощью класса **mntn**. Классом **jpsj2** отметились японцы. Из русскоязычных журналов замечен только «Сибирский журнал вычислительной математики» — класс **sibjnm**.

Следует отметить обязательный для подготовки журнальных публикаций в издательстве Elsevier класс **elsart**. Класс **nature** позволит подготовить pdf-файл для журнала Nature. Часто при создании публикаций для журналов или конференций используются небольшие модификации класса **revtex4**.

Для объединения разных документов в один, например, для оформления трудов конференции, может пригодиться класс **combine**.

8.3.8. Организуем резюме

Написание резюме или curriculum vitae — довольно популярный вид деятельности в современном мире. Для резюме нет общепринятого стандарта, но некоторые указания существуют.

Европейская комиссия рекомендует определённый формат для составления резюме, и этот формат полностью реализуется с помощью специализированного класса **eurorescv**.

Класс **vita** представляет собой конструктор для создания резюме. Документация фактически отсутствует, но разобраться по имеющимся примерам для IT-специалиста и певца не составляет сложности.

Класс **curve**, напротив, обладает качественной документацией. Механизм рубрик позволяет классу **curve** поддерживать несколько резюме разной направленности и легко переключаться между ними.

Современный класс **morderncv** рекомендуется как гибкое и простое средство создания резюме как современного вида, так и классических форм.

Следует отметить, что классы определяют многое, но далеко не всё. Поэтому после выбора класса документа можно и нужно подключить стили, которые могут серьёзно поменять внешний вид. Стиль **currvita** позволяет создавать резюме в окружении стандартных классов.

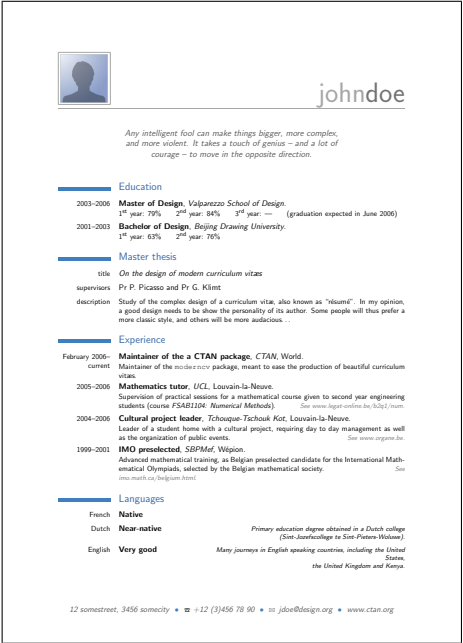


Рис. 8.4. Класс **moderncv**

8.3.9. Защищаем диссертации

Раньше были курсовые и дипломные работы, а теперь куда ни глянь — везде диссертации. Каждый уважающий себя университет имеет свой уникальный стиль оформления диссертации. Если хочется написать свой класс, то в качестве отправной точки можно выбрать класс **ucthesis** от Калифорнийского университета (UC Berkeley).

Станислав Кручинин озадачился судьбами русскоговорящих диссертантов и создал класс **disser**. Пакет отсутствует в составе дистрибутива TeX Live, но его легко можно взять в директории `{CTAN}/macros/latex/contrib/disser` в любом CTAN-архиве. Следует учесть, что какая-либо пользовательская документация на текущий момент полностью отсутствует. С другой стороны диссертанты по идее люди не глупые и разберутся в имеющихся достаточно подробных примерах.

8.3.10. Всякая всячина

Далеко не все классы подчиняются уже перечисленной классификации. За рамки темы вышли классы для составления календарей, обложек для CD (cd), вопросников (qcm), объявлений о занятиях (assignment), концертных программ (ConcProg), программ курса (courseoutline и coursepaper), рабочего журнала для биологов (labbook), пьес (stage), тибетских карточек (pecha — рис. 8.5), карточек для запоминания иностранных слов (flashcards) и для многого другого.

Старые и давно не поддерживаемые классы могут не собраться в новом окружении, но исходники доступны и всегда можно довести их до необходимой кондиции.

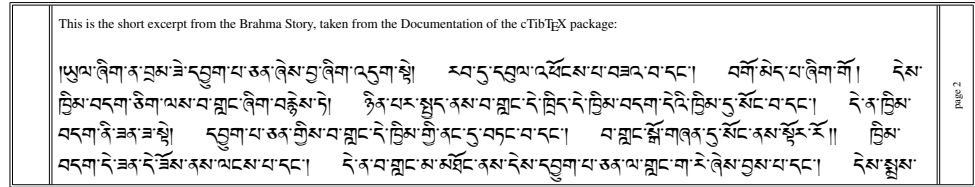


Рис. 8.5. Что-то определённо тибетское — класс pecha

Глава 9

Делаем презентации

Существует три разновидности людей:

- те, кто видит;
- те, кто видит, когда им показывают;
- и те, кто не видит.

Леонардо да Винчи

Хочется стать популярным? Достаточно подкупить слушателей. Хочется донести свою идею? Нужно сделать нормальную презентацию. При этом вовсе не нужно аляповатого фона, мультипликации при смене слайдов, но необходим разборчивый текст и картинки к месту.

Создание презентации — очень тяжёлое занятие, не следует жалеть о потерянных минутах для наведения блеска. Делая же презентацию, нужно не забывать об эмпирических правилах.

- Один слайд требует не меньше одной минуты.
- Один слайд со смыслом требует не менее пяти минут.
- Времени всегда не хватает.
- Не следует «пихать» в презентацию больше слайдов, чем получится рассказать по времени. Перебор по времени только раздражает слушателей.
- Каждый слайд должен иметь свой заголовок (`\frametitle`).
- В один слайд можно поместить около 20–40 слов и заведомо не больше 80.
- Полезно использовать `block`, `theorem`, `proof` и `example`. Эти окружения структурируют текст и помогают выделять основные мысли.
- Для разных аудиторий эти правила могут существенно отличаться.

9.1. slides

Идея обработки

Разрабатываются два независимых отбора событий по трековой системе с эффективностью ε_T и по калориметру с эффективностью ε_C , тогда

$$\varepsilon_T = \varepsilon_T^{(C)} \cdot \varepsilon_C + \varepsilon_T^{(\bar{C})} \cdot (1 - \varepsilon_C), \quad (1)$$

где $\varepsilon_T^{(C)}$ — эффективность отбора T при условии, что отбор C сработал (она определяется экспериментально на данных J/ψ), а $\varepsilon_T^{(\bar{C})}$ — эффективность того же отбора при условии, что отбор C не сработал (Определяется на $J/\psi \rightarrow \pi^+\pi^-(X)$).

Аналогично

$$\varepsilon_C = \varepsilon_C^{(T)} \cdot \varepsilon_T + \varepsilon_C^{(\bar{T})} \cdot (1 - \varepsilon_T). \quad (2)$$

1

Рис. 9.1. **slides** — это просто

не выйдет. Опция класса **landscape** позволяет выбирать альбомную ориентацию для страницы по умолчанию. Слайды создаются с помощью окружения **slide**.

```
\documentclass[a4paper,landscape]{slides}
...
\begin{document}
\begin{slide}
\begin{center}
\Large Идея обработки
\end{center}
...
\end{slide}
\end{document}
```

Класс **seminar** похож на **slides** и лишь чуть-чуть более современен (1993 г.), но не в пример лучше документирован (файл **sem-user.pdf**) и кроме стандартного окружения **slide** имеет простейший набор команд для создания рамок.

Если надо что-то сделать по-быстрому из уже готового текста с целью просто продемонстрировать какую-то идею, то старые, но надёжные классы **slide** и **seminar** вполне для этого подойдут.

9.2. Немного о PDF

PDF (Portable Document Format) — открытый платформенезависимый формат для описания электронных документов был создан фирмой Adobe Systems в 1993 году. В 2006 году была опубликована версия стандарта под номером 1.7. В декабре 2007 года PDF 1.7 стал стандартом ISO 32000. Файл в PDF-формате может представлять собой комбинацию векторной графики, текста и растровых

Динозавр среди классов ЛАТЭХ, специализирующихся на презентациях. Идея очень простая. В качестве класса документа выбирается **slides**. В результате базовый размер шрифта автоматически увеличивается. Это позволяет прочитывать стандартный текст на экране и избавиться от одного из «смертных грехов» докладчика, а именно желания уместить слишком много информации на одной страничке. Здесь по умолчанию ничего с этим

изображений (фотографий, снимков экрана и тому подобное). В стандарте PDF предусмотрена возможность создания гиперссылок, заполняемых форм и интерактивных вставок на JavaScript. Начиная с версии 1.6 декларируется возможность описания 3D интерактивных документов — что бы это ни значило, звучит заманчиво, но к сожалению, пока рано использовать эти возможности.

С точки зрения формата для представления презентации PDF удовлетворяет необходимым условиям, таким как:

- Простота создания. Это сила качественных открытых форматов — рано или поздно их начинают поддерживать все кому не лень.
- Переносимость. Везде найдётся программа просмотра PDF.
- Элементы интерактивности. Документ представляет собой не только плоскую последовательность страниц.

9.2.1. Простота создания

Допустим, что тем или иным способом был получен PostScript-файл презентации. Из него с помощью **ghostscript**, точнее с помощью скрипта **ps2pdf**, можно получить нормальный PDF:

```
> ps2pdf «файл.ps» «файл.pdf»
```

Получить PDF можно и напрямую из tex-исходников с помощью программы **pdflatex**. Эта программа отличается от **latex** в основном только тем, что в качестве выходного формата получается PDF. При использовании **pdflatex** следует учитывать, что все графические файлы должны быть либо в виде pdf (вектор), либо png/jpeg (растр). **pdflatex** не умеет обрабатывать eps-файлы, за исключением картинок, созданных с помощью MetaPost.

Если вдруг по какой-то причине pdf нужно преобразовать в PostScript, то лучше воспользоваться утилитой **pdftops** из пакета **xpdf**.

```
> pdftops [-eps] «pdf-файл»
```

Если необходимо получить картинку в формате EPS, то следует использовать ключ **-eps**.

Немного о шрифтах

В PDF можно внедрять векторные шрифты Type1. Это позволяет отображать готовые документы независимо от набора имеющихся шрифтов. Отображение на экране, особенно при низких разрешениях, зависит исключительно от качества внедрённых шрифтов. Парадокс качества: чем хуже разрешение, тем больший объём работы надо проделать с векторным шрифтом, чтобы он выглядел приемлемо. К счастью, в случае презентаций это не является проблемой, так как для читаемости на большом экране размер шрифта нужно сильно увеличить.

Это эффективно увеличивает разрешение до сравнимого с разрешением лазерного принтера, под который и оптимизированы наиболее популярные векторные шрифты Computer Modern (пакет **cm-super**).

Ни в коем случае для отображения на экране не стоит использовать растровые шрифты в формате True3. Шрифты **cm-super** (в TeX Live есть заведомо) обязательно должны быть установлены.

9.2.2. Переносимость

Везде есть Adobe Reader и Ghostscript. Если этого где-то нет, то оно легко может там появиться. Adobe Reader предоставляется всем желающим самой Adobe Systems. Как следствие в смысле поддержки всех расширений формата PDF эта программа условно «впереди планеты всей». Поэтому презентацию, скорее всего, придётся показывать именно с её помощью.

Одной из раздражающих особенностей Adobe Reader, мешающей использовать эту программу при работе над документом, является то, что в нём отсутствует возможность перезагружать изменённый документ. Эту проблему можно частично решить с помощью сторонних программ **pdfopen** и **pdfclose** (заведомо присутствуют в дистрибутиве TeX Live):

```
> pdfclose —file «файл.pdf»
# обновляем «файл.pdf»
> pdfopen —file «файл.pdf»
```

Ghostscript и программа просмотра с его использованием также есть везде. Ghostscript отображает PDF как обычный «плоский» документ, то есть об интерактивных «эффектах» можно забыть. Зато проблем с обновлением текста нет: нажал «.» (точку) и картинка обновилась.

xpdf (<http://www.foolabs.com/xpdf/>) для просмотра PDF доступен только для систем, где есть X Window. Начиная с версии 3.02, **xpdf** поддерживает структуру PDF вплоть до 1.7. **xpdf** используется как «движок» и для других программ просмотра, например, для **kpdf**. Обновить документ можно с помощью клавиши «г». Очень удобен при просмотре в процессе подготовки документа.

9.2.3. Интерактивность

Зависит исключительно от стиля, который используется для подготовки PDF. Присутствует весь простейший джентльменский набор: гиперссылки, различные виды переходов со слайда на слайд и анимация. Есть ограниченная возможность демонстрировать клипы и внедрять в презентацию звуки.

За активные гиперссылки отвечает пакет **hyperref**. Пакет обладает многочисленными и очень гибкими настройками, которые можно найти в **manual.pdf**.

Для вставки мультимедийных фрагментов можно воспользоваться пакетом **movie15**. Ограничения: этот пакет работает только в связке с **pdflatex**, и просмотр возможен только в Adobe Reader в сборке под MS Windows или MacOS.

При внедрении видеофрагмента следует убедиться, что он может быть просмотрен на платформе, на которой планируется провести презентацию. Для выяснения подробностей нужно обратиться к файлу документации `movie15.pdf`.

► На текущий момент даже в существующей поддержке проигрывания клипов внутри PDF есть проблемы. Одна из самых распространённых — это попытка проиграть клип в презентации, сделанной на Mac OS в MS Windows, и наоборот. Так как клип воспроизводится системным плеером, а кодеки кодирования видеофрагментов на этих системах по умолчанию разные, то с очень большой долей вероятности ничего не получится.

9.3. beamer

Время шло, компьютеры матерели, появились проекторы и захотелось чего-то разноцветного. Так увидело свет новое поколение презентационных классов.

С помощью пакета **beamer** в принципе можно создавать «прозрачки», как это делается посредством **slides**, но основное его предназначение — электронная презентация. Пакету чуть более трёх лет, но он очень активно развивается, и на сегодня это, пожалуй, лучший пакет для презентаций в ЛАТЭХ. Автор Тил Тантау (Till Tantau) оказался очень восприимчивым к предложениям сообщества относительно своего проекта.

У пакета **beamer** есть масса стандартных стилей, исчерпывающее описание на более чем двухстах страницах (`beameruserguide.pdf`) и домашняя страничка по адресу <http://sourceforge.net/projects/latex-beamer>. ТЭХ Live содержит **beamer** по умолчанию.

beamer можно использовать как с **pdflatex**, так и со «стандартной связкой» **latex + dvips + ps2pdf**. Первое, что надо сделать — это выбрать класс:

```
\documentclass[hyperref={unicode=true}]{beamer}
\usepackage[koi8-r]{inputenc}
```

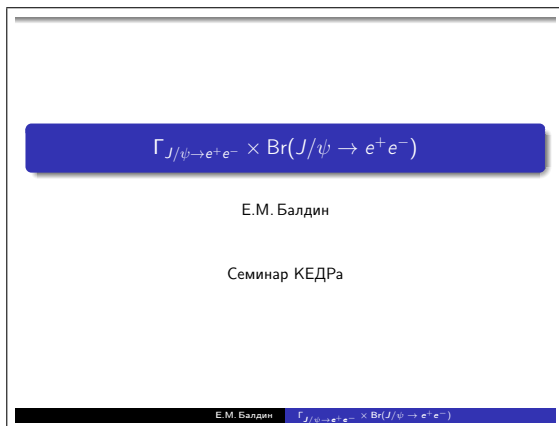
Класс **beamer** по умолчанию загружает пакет **hyperref**. Если в документе планируется использовать этот пакет со значениями, отличными от умолчания (например, добавить опцию `unicode=true` для кириллических закладок), то их следует передать как необязательный параметр команды выбора класса.

Если текст представлен в кодировке UTF-8, то это также необходимо указать при загрузке **beamer**:

```
\documentclass[utf8]{beamer}
\usepackage[utf8]{inputenc}
```

Теперь можно выбрать тему для презентации и определить заголовок для титульного листа. Единицей представления для **beamer** является окружение `frame`.

```
% Выбор темы (преамбула)
\usetheme{Madrid}
% Выбор декорации
\useoutertheme{shadow}
\title{«Заголовок»}
\date{«Дата или место»}
\author{«Автор»}
\begin{document}
% Титульная страница
\begin{frame}
  \titlepage
\end{frame}
```



Окружению frame можно передать необязательный параметр `t`, который «прижимает» текст к верхней части слайда.

Теперь можно приступить к самой презентации. Как и в обычных статьях, в **beamer** можно применять команды структурной разметки типа `section`. Эти команды должны идти за пределами окружения `frame`. Структурная разметка, в частности, полезна для быстрого доступа, например, через навигационное меню или оглавление. Оглавление создаётся с помощью стандартной команды `\tableofcontents`. Чтобы оглавление разворачивалось не сразу, а по ходу дела, этой команде следует передать необязательный параметр `pausesections`.

```
% Тема с навигацией
\usetheme{Montpellier}
% Чёрно-белые цвета
% Хорошо для бумаги
\usecolortheme{dove}

% Структурная разметка
\section{Теория}
\subsection{Формула}
\begin{frame}
  \frametitle{«Заголовок»}
  ...
\end{frame}
```

Теоретическая зависимость (Азимов и др.)

$$\frac{d\sigma^{e^+e^-}}{d\Omega} = \frac{1}{M^2} \left\{ \frac{9}{4} \frac{\Gamma_{e^+e^-}^2}{\Gamma M} \left(1 + \frac{3}{4} \beta \right) (1 + \cos^2 \theta) \operatorname{Im} f - \right. \\ \left. - \frac{3\alpha}{2} \frac{\Gamma_{e^+e^-}}{M} \left(1 + \frac{11}{12} \beta \right) \left[(1 + \cos^2 \theta) - \frac{(1 + \cos^2 \theta)^2}{(1 - \cos \theta)} \right] \operatorname{Re} f + \right. \\ \left. + \frac{\alpha^2}{4} \left(1 + \frac{13}{12} \beta \right) \frac{(3 + \cos^2 \theta)^2}{(1 - \cos \theta)^2} \right\},$$

где

$$f = \left(\frac{\frac{M}{2}}{-W + M - \frac{i\Gamma}{2}} \right)^{1-\beta}, \quad \beta = \frac{4\alpha}{\pi} \left(\ln \frac{W}{m_e} - \frac{1}{2} \right).$$

Для создания заголовка текущего слайда используется команда `\frametitle`. Так же можно определить и подзаголовок с помощью команды `\framesubtitle`.

► Для этого примера была выбрана цветовая схема `dove`, которая всю цветовую палитру приводит к оттенкам серого. Это удобно, если слайды будут печататься

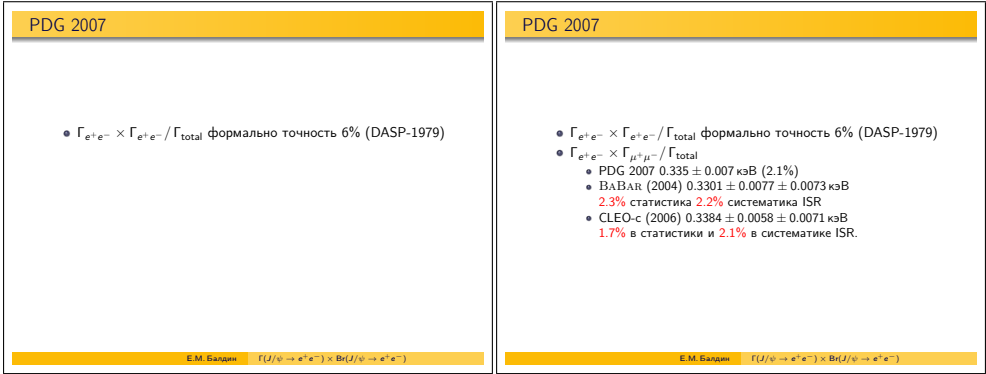


Рис. 9.2. Пример оверлеев, созданных с помощью `\pause` в **beamer**

на бумаге, например, как иллюстрации в обычной книге без цветных вкладок. Для электронных презентаций это не нужно. Примеры разных цветовых схем разобраны в документации к пакету.

Оверлеи

В процессе представления очень полезны *оверлеи* — составные слайды, которые как бы накладываются друг на друга. Для создания простейшего оверлея используется команда `\pause`:

```
\begin{itemize}
\item ...
% Здесь заканчивается первый оверлей и начинается второй
\pause
\item ...
\end{itemize}
```

На рис. 9.2 приведён пример двух слайдов, созданных автоматически из такой простой конструкции. Сначала показывается левый слайд, а затем правый и создаётся впечатление, что второй пункт перечисления как бы добавляется к первому без смены слайда.

► Если приглядеться к цифрам со знаками процента на слайде (например, 1.7% в последней строчке), то можно заметить, что они отличаются от остального текста. В цвете это отличие было бы очевидно, так как эти цифры выделены командой `\alert`. Команда `\alert` — это аналог `\emph`. По умолчанию выделенный фрагмент просто отображается красным цветом, но при желании `\alert` всегда можно переопределить. Выделить короткий фрагмент текста также можно с помощью команды `\beamerbutton{текст}` — в этом случае из текста получается кнопка.

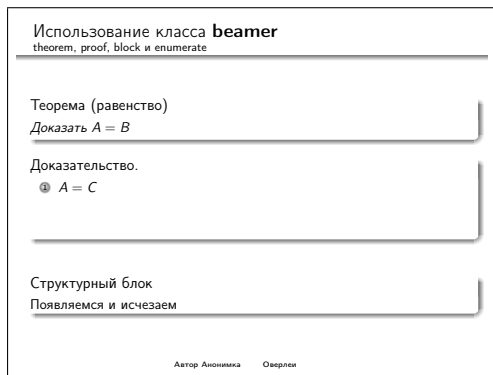


Рис. 9.3. Оверлей №1

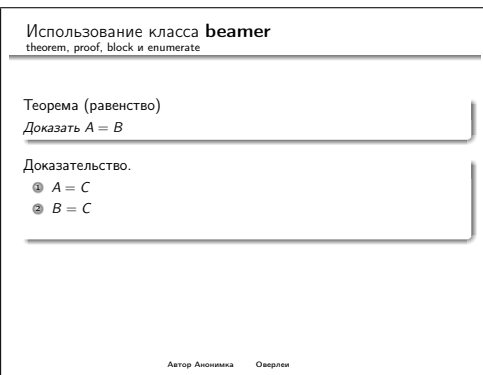


Рис. 9.4. Оверлей №2

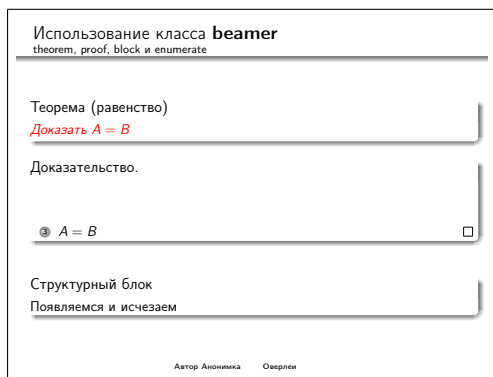


Рис. 9.5. Оверлей №3

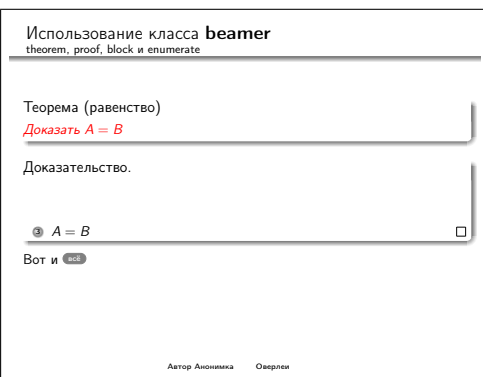


Рис. 9.6. Оверлей №4

Для работы с оверлеями в **beamer** добавлен ещё один способ передачи параметров командам $\langle \rangle$ — меньше/больше. Таким образом команде передаётся список оверлеев, на которых она должна действовать. То есть переопределённая в пакете **beamer** команда `\color<3–4>` раскрашивает текст в указанный цвет с 3 по 4 оверлей. Список можно передавать через запятую или как интервал. Список вида: `-3,5-9,12,17-` означает, что команда действует для оверлеев из интервалов: от начала до 3, от 5 до 9, для 12, от 17 и до конца.

В **beamer** предусмотрена масса способов работы с оверлеями. Рассмотрим некоторые из них:

```
% Создание своей теоремы
\newtheorem{MyTheorem}{Теорема}
\begin{frame}
% Заголовок
\frametitle{Использование класса \textbf{beamer}}
```

```
% Подзаголовок
\framesubtitle{theorem, proof, block и enumerate}
% Использование теоремы
\begin{MyTheorem}[равенство]
% Выделяется красным цветом на 3 и 4 оверлеях
\color<3-4>[rgb]{1,0,0}{Доказать  $(A=B)$ }
\end{MyTheorem}
% Доказательство
\begin{proof}
\begin{enumerate}
% Присутствует вплоть до 2 оверлея
\item<-2>  $(A=C)$ 
% Появляется только на 2 оверлее
\item<2>  $(B=C)$ 
% Есть на 3 и 4 оверлеях
\item<3,4>  $(A=B)$ \qedhere
\end{enumerate}
\end{proof}
% Последняя фраза (от 4 оверлея и далее, если можно)
\uncover<4->{Вот и \beamerbutton{всё}}
% Манипуляция с блоком (есть на 1 и 3 оверлеях)
\begin{block}<1,3>{Структурный блок}
  Появляемся и исчезаем
\end{block}
\end{frame}
```

На рис. 9.3–9.6 представлены четыре оверлея, которые получены при компиляции приведённого ранее кода.

Некоторые стандартные команды в пакете **beamer** переопределены так, что могут воспринимать списки оверлеев. Примерами таких команд являются:

- `\color{текст}` — цвет текста;
- `\includegraphics` — для вставки графики;
- `\item` — определена внутри перечислений, к которым относятся окружения `itemize` и `enumerate`;
- окружение `theorem` (подробности в разделе 13.7 на стр. 205);
- окружение `proof`. Если есть теорема, то должно быть и доказательство. В конце доказательства традиционно добавляется символ $\square$ — знак QED (*quod erat demonstrandum* — что и требовалось доказать). Команда `\qedhere` размещает знак QED в той же строке, где он указан. По умолчанию же QED будет отведена своя собственная строка, что, как правило, нежелательно.

В **beamer** дополнительно определены новые команды, воспринимающие список оверлеев:

- `\alert{текст}` — выделение текста;
- `\only` или `\visible` — добавление текста только для указанного списка оверлеев;
- `\invisible` — команда комплементарная `\only`;
- `\uncover` — то же, что и `\only`, только резервируется место под текст даже на тех слайдах, где он отсутствует;
- `\alt<список>{текст}{альтернативный текст}` — для указанного списка оверлеев выводится «текст» иначе «альтернативный текст»;
- окружение `block` — именованный блок. Похоже на окружение `theorem`.

Гиперссылки

Для создания гиперссылки для начала следует установить метку или «якорь» в нужном месте. Это можно сделать с помощью команды `\label`. После того как метка установлена, с помощью команды `\hyperlink` организуется гиперссылка:

```
\label{metka}
\hyperlink{metka}{«Гиперссылка»}
```

Вместо обычного текста можно использовать фактически любую ЛАТЭХ-структуру, например, команду создания «кнопки» `\beamerbutton`. Более общей командой для установки метки является команда:

```
\hypertarget<«номер оверлея»>{«метка»}{«текст»}
```

С её помощью можно указать не только структурную единицу, но и на какой именно оверлей следует сослаться.

Программный код

Для представления программного кода необходимо использовать окружения типа **verbatim** или **lstlistings**. Для того чтобы код на слайде отображался правильно, окружению `frame` необходимо передать опцию `fragile`. Оформление кода может выглядеть, например, так:

```
\begin{frame}[fragile]
% Определяем более короткие команды для удобства
\newcommand{\un}{\uncover}
\newcommand{\al}{\alert}
\frametitle{An Algorithm For Finding Primes Numbers.}
```

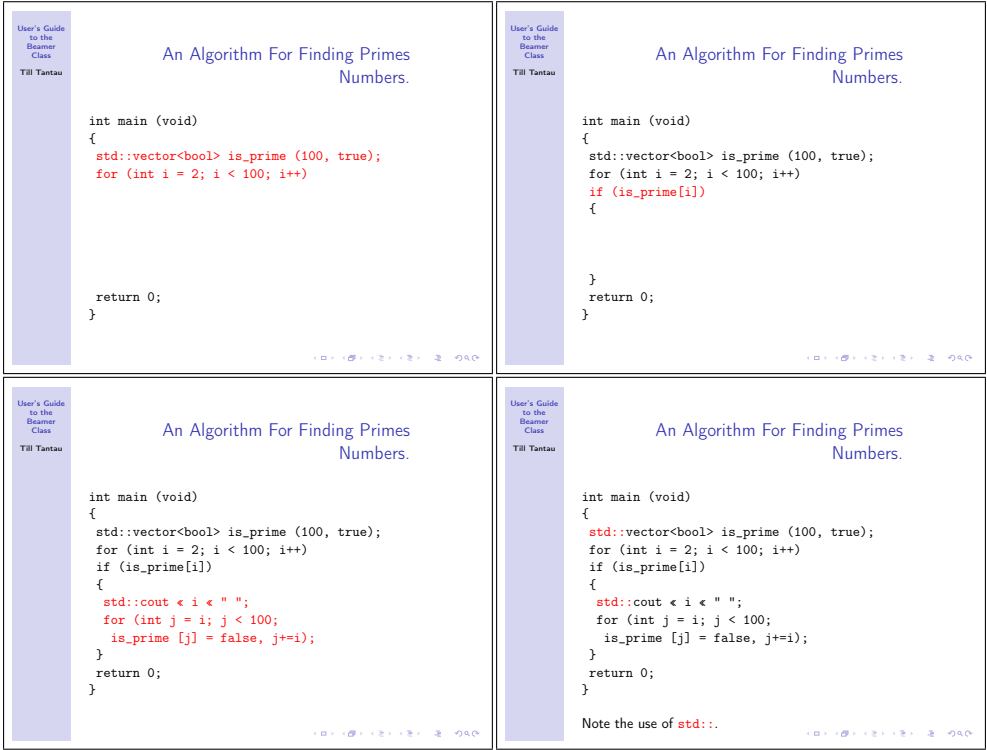


Рис. 9.7. Представление программного кода (тема Hannover, естественно, в цвете)

```
\begin{semiverbatim}
\un<1->\{ \al<0>\{int main (void)\}\}
\un<1->\{ \al<0>\{\}\}
\un<1->\{ \al<1>\{ \al<4>\{std::\} vector<bool>is_prime(100,true);\}\}
\un<1->\{ \al<1>\{ for (int i = 2; i < 100; i++)\}\}
\un<2->\{ \al<2>\{ if (is_prime[i])\}\}
\un<2->\{ \al<0>\{ \}\}
\un<3->\{ \al<3>\{ \al<4>\{std::\} cout << i << " ";\}\}
\un<3->\{ \al<3>\{ for (int j = i; j < 100;\}\}
\un<3->\{ \al<3>\{ is_prime [j] = false , j+=i);\}\}
\un<2->\{ \al<0>\{ \}\}
\un<1->\{ \al<0>\{ return 0;\}\}
\un<1->\{ \al<0>\{\}\}
\end{semiverbatim}
\visible<4->\{Note the use of \alert{\texttt{std::}}.\}
\end{frame}
```

Выбор и настройка темы

Все темы **beamer** разбиваются на пять классов.

- Именные темы — концепция презентации. Для выбора темы используется команда `\usetheme`. Обычно создатель именной темы просто выбирает в ней соответствующие цветовую, шрифтовую и декоративные темы. В **beamer** на начало 2007 года есть следующие именные темы: AnnArbor, Antibes, Bergen, Berkeley, Berlin, Boadilla, CambridgeUS, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged и Warsaw.
- Цветовые темы — палитра презентации. Для выбора темы используется команда `\usecolortheme`. Можно выбрать из следующего набора палитр: albatross, beaver, beetle, crane, dolphin, dove, fly, lily, orchid, rose, seagull, seahorse, sidebartab, structure, whale и wolverine.
- Шрифтовые темы — выбор подмножества шрифтов. Для выбора темы используется команда `\usefonttheme`. Существуют следующие шрифтовые темы: professionalfonts, serif, structurebold, structureitalicserif и structuresmallcaps serif.
- Текстовые и структурные декорации — темы, определяющие, как выглядят перечисления, теоремы и выделения. Для выбора темы используется команда `\useinnertheme`. Можно выбрать следующие варианты декораций: circles, inmargin, rectangles, rounded.
- Внешние декорации — темы, определяющие вид заголовков и обрамления слайда. Для выбора темы используется команда `\useoutertheme`. Существуют следующие типы обрамлений: infolines, miniframes, shadow, sidebar, smoothbars, smoothtree, split и tree.

Никто не мешает создать свою собственную тему и назвать её именем своего города или страны. Подробности, о том как это делается, а также примеры уже существующих тем следует искать в документации к пакету.

Печать слайдов

Размер слайдов всего 128 мм на 98 мм. При растягивании такого маленького слайда на весь экран получаются большие и хорошо читаемые буквы. Для печати слайдов проще всего взять программу Adobe Reader и в диалоге печати установить опцию **Page Scaling** (масштабирование страницы) в состояние **Fit to Printable Area** (растянуть страницу на всё доступное пространство для печати). Также можно воспользоваться стилевым файлом **pgfpages** из пакета **pgf**. В этом случае специально растягивать слайды не потребуется.

```
\usepackage{pgfpages}
\pgfpagesuselayout{resize to}[a4paper,%
                           border shrink=5mm,landscape]
```

Здесь слайд растягивается на страницу A4 в альбомной ориентации с отступом от краёв в 5 мм. Если хочется распечатать по два слайда на страницу, то необходимо передать ещё и следующие настройки:

```
\pgfpagesuselayout{2 on 1}[a4paper,border shrink=5mm]
```

Ускорение компиляции

При подготовке презентации можно использовать опцию **draft** при выборе класса. Это немного ускорит компиляцию. Также можно указывать, какие именно слайды следует включать при компиляции (похоже на `\includeonly`):

```
\includeonlyframes{ex1,ex3}
\frame[label=ex1]
{Этот слайд будет включён при компиляции. }
\begin{frame}[label=ex2]
Аналогично ex2.
\end{frame}
\frame{A вот этого слайда не будет.}
```

Использование меток позволяет выводить уже имеющиеся слайды ещё раз с помощью команды `\againframe`:

```
%ex1 будет выведен ещё раз
\againframe{ex1}
```

Мультимедиа

Пакет **beamer** включает стилевой файл **multimedia**. После загрузки этого файла можно воспользоваться командами `\movie` и `\sound` для включения клипа или звука в презентацию. К сожалению, пока эта возможность ограничена тем, что поддерживает её только Adobe Reader в сборке для MS Windows или MacOS. При этом всегда необходимо проверять результат на работоспособность.

В пакете **beamer** предусмотрена возможность создания анимации на основе созданных слайдов. Команда

```
\animate<«список оверлеев»>
```

позволяет автоматически проигрывать последовательность слайдов. Для того чтобы эта возможность сработала, необходимо Adobe Reader раскрыть на весь экран.

Глава 10

Базовые навыки

Ставим единицу в том месте, где должна быть краска, и ноль в том, где её быть не должно — и можно печатать книжную страницу!

Дональд Э. Кнут

Собранная в этой главе информация может помочь при наборе и формировании структуры текста. Прежде чем приступить к изучению, имеет смысл посмотреть главу 3 «Базовые элементы» на странице 29.

10.1. Интернационализация и локализация

Основным инструментом в деле интернационализации и локализации является пакет **babel**. Он не просто включает поддержку переносов, но и локализует стандартные заголовки, типа «Оглавление», «Список литературы», «Приложение» и тому подобное. Для выбора языков документа их следует указать как опцию при загрузке **babel**:

```
\usepackage[english,russian]{babel}
```

Основным языком станет последний язык из перечисленных в списке.

Пакет поддерживает сорок три языка без учёта диалектов: африкаанс (afrikaans), английский (english, USenglish, american, UKenglish, british, canadian, australian, newzealand), баскский (basque), болгарский (bulgarian), бретонский (breton), венгерский (magyar), (hungarian), верхнелужицкий (upper Sorbian), галисийский (galician), голландский (dutch), греческий (greek, polutonikogreek), датский (danish), еврейский (hebrew), интерлингва (interlingua), ирландский (irish), исландский (icelandic), испанский (spanish), итальянский (italian), каталанский (catalan),

латинский (latin), малайский (bahasa), немецкий (austrian, german, germanb, ngerman, naustrian), нижнелужицкий (lowersorbian), норвежский (norsk), (nynorsk), польский (polish), португальский (portuges, portuguese, brazilian), русский (russian), румынский (romanian), северносаамский (samin), сербский (serbian), словацкий (slovak), словенский (slovene), турецкий (turkish), украинский (ukrainian), уэльский (welsh), финский (finnish), французский (french, francais, canadien, acadian), хорватский (croatian), чешский (czech), шведский (swedish), шотландский (scottish), эсперанто (esperanto) и эстонский (estonian).

Для смены языка в тексте можно воспользоваться декларацией `\selectlanguage`:

```
{\selectlanguage{english}\chaptername{}}
по-русски "--- это \chaptername.
```

Chapter по-русски — это Глава.

При переключении языка меняются не только правила переносов, но и вообще всё, за что отвечает **babel**. В качестве альтернативного способа переключения языка можно использовать окружение `otherlanguage`:

```
% Русские переносы
Наука "--- это то, что известно достаточно
хорошо, чтобы объяснить это компьютеру. Всё
остальное "--- это искусство. Дональд~Кнут\rag
% Переключаемся на английский
\begin{otherlanguage}{english}
% Теперь пошли английские переносы
Science is what we understand well enough to
explain to a computer. Art is everything
else we do. Donald~Knuth
\end{otherlanguage}
```

Наука — это то, что известно достаточно хорошо, чтобы объяснить это компьютеру. Всё остальное — это искусство. Дональд Кнут
Science is what we understand well enough to explain to a computer. Art is everything else we do. Donald Knuth

Для кратких включений текста на другом языке можно воспользоваться командой `\foreignlanguage{язык}{текст}`. Эта команда меняет только таблицу переносов и специальные символы, присущие языку, но не локализацию:

```
\foreignlanguage{english}{\chaptername{}} "---
видно, что локализация не поменялась.
```

Глава — видно, что локализация не поменялась.

Этой команде соответствует окружение `otherlanguage*`.

Следует знать, что при переключении с русского (**russian**) на английский язык (**english**), переключается также и внутренняя раскладка $\TeX$ с T2A, в которой определена кириллица, на T1, в которой кириллицы нет, поэтому при включённом английском языке кириллица не выводится. Для того чтобы обойти эту проблему, следует воспользоваться реализациями $\LaTeX$ -компилятора, которые в качестве внутренней кодировки используют Unicode, например, **xelatex**.

babel — не просто пакет по выбору языков — это конструктор, позволяющий добавить свой собственный язык и правила. В частности, при определении своих команд, зависящих от языка текста, может быть полезна инструкция `\iflanguage`:

```
Здесь\selectlanguage{english}
\iflanguage{russian}{русский}{not russian}.\par
\selectlanguage{russian}
А здесь \iflanguage{russian}{русский}{not russian}.
```

Здесь not russian.
А здесь русский.

► Загрузка русского языка в пакете **babel**, к сожалению, не полностью выстав-
ляет правильные умолчания для русской полиграфии. В частности, для того
чтобы первый абзац начинался с красной строки, необходимо загрузить стиль
indentfirst. Дополнительно следует загрузить и стиль **miscorr** из пакета **t2**,
который исправляет отдельные недочёты **babel**.

10.2. Символы

Набор текста не ограничен только буквами алфавита. Для обозначения раз-
личных понятий человечество постоянно придумывает новые значки. Наиболее
полным источником информации о доступных в **Л^AT_EX** символах является текст
«The Comprehensive L^AT_EX Symbol List». Всё что нужно — это открыть pdf-файл
symbols-a4.pdf и найти нужный символ.

Символы, определённые в текстовой моде			
^	\textasciicircum	~	\textasciitilde
*	\textasteriskcentered	\	\textbackslash
	\textbar	{	\textbraceleft
}	\textbraceright	•	\textbullet
†	\textdagger	‡	\textdaggerdbl
\$	\textdollar	...	\textellipsis
—	\textemdash	—	\textendash
¡	\textexclamdown	>	\textgreater
<	\textless	ª	\textordfeminine
º	\textordmasculine	¶	\textparagraph
·	\textperiodcentered	¿	\textquestiondown
“	\textquotedblleft	”	\textquotedblright
‘	\textquoteleft	’	\textquoteright
®	\textregistered	§	\textsection
£	\textsterling	™	\texttrademark
_	\textunderscore	~	\textvisiblespace

Дополнительные символы (textcomp)			
*	<code>\textasteriskcentered</code>		<code>\textbardbl</code>
○	<code>\textbigcircle</code>	ℬ	<code>\textblank</code>
	<code>\textbrokenbar</code>	●	<code>\textbullet</code>
†	<code>\textdagger</code>	‡	<code>\textdaggerdbl</code>
=	<code>\textdblhyphen</code>	=	<code>\textdblhyphenchar</code>
%	<code>\textdiscount</code>	€	<code>\textestimated</code>
‡	<code>\textinterrobang</code>	‡	<code>\textinterrobangdown</code>
♪	<code>\textmusicalnote</code>	№	<code>\textnumero</code>
○	<code>\textopenbullet</code>	ª	<code>\textordfeminine</code>
♂	<code>\textordmasculine</code>	¶	<code>\textparagraph</code>
·	<code>\textperiodcentered</code>	‰	<code>\textpertenthousand</code>
‰	<code>\textperthousand</code>	‡	<code>\textpilcrow</code>
'	<code>\textquotesingle</code>		<code>\textquoteststraightbase</code>
”	<code>\textquoteststraightdblbase</code>	℞	<code>\textrecipe</code>
※	<code>\textreferencemark</code>	§	<code>\textsection</code>
—	<code>\textthreequartersemdash</code>	~	<code>\texttildelow</code>
—	<code>\texttwelveudash</code>		

Стрелки в текстовой моде (textcomp)			
↓	<code>\textdownarrow</code>	←	<code>\textleftarrow</code>
→	<code>\textrightarrow</code>	↑	<code>\textuparrow</code>

Дополнительные символы (wasy sym)			
✱	<code>\hexstar</code>	✱	<code>\varhexstar</code>
☒	<code>\ataribox</code>	♠	<code>\bell</code>
☹	<code>\blacksmiley</code>	☞	<code>\Bowtie</code>
↓	<code>\brokenvert</code>	✓	<code>\checked</code>
☑	<code>\CheckedBox</code>	⌚	<code>\clock</code>
∅	<code>\diameter</code>	▼	<code>\DOWNarrow</code>
☹	<code>\frownie</code>	⊘	<code>\invdiameter</code>
✝	<code>\kreuz</code>	◀	<code>\LEFTarrow</code>
⚡	<code>\lightning</code>	‰	<code>\permil</code>
☎	<code>\phone</code>	☞	<code>\pointer</code>
Ⓚ	<code>\recorder</code>	▶	<code>\RIGHTarrow</code>
☺	<code>\smiley</code>	□	<code>\Square</code>
☼	<code>\sun</code>	▲	<code>\UParrow</code>
☒	<code>\wasylozenge</code>	☒	<code>\XBox</code>

Символы, определённые и для текстовой и для математической моды					
\$	\\$	}	\}	{	\}
...	\dots	¶	\P	£	\pounds
†	\dag	‡	\ddag		
Дополнительные символы из пакета amssymb					
✓	\checkmark	®	\circledR	✠	\maltese
Не ASCII-символы					
å	\aa	Å	\AA	Æ	\AE
ä	\dj	Ä	\L	İ	\l
Œ	\OE	œ	\oe	ß	\ss
ø	\o	Ø	\O	§	\S
SS	\SS				

10.2.1. Валютные символы

Валютные знаки (textcomp)					
₮	\textbaht	¢	\textcent	¢	\textcentoldstyle
₯	\textcolonmonetary	₱	\textcurrency	\$	\textdollar
\$	\textdollaroldstyle	₫	\textdong	€	\texteuro
f	\textflorin	G	\textguarani	£	\textlira
₦	\textnaira	P	\textpeso	£	\textsterling
₩	\textwon	₹	\textyen		
Дополнительные символы валют (wasysym)					
¢	\cent	₱	\currency		
Знаки евро (eurosym)					
€	\geneuro	€	\geneuronarrow	€	\geneurowide
€	\officiaeuro				

10.2.2. Копирайт и копилефт

Знаки копирайта и копилефта (textcomp)					
©	\textcircledP	©	\textcopyleft	©	\textcopyright
®	\textregistered	SM	\textservicemark	TM	\texttrademark

10.2.3. Акценты

В пакете **textcomp** определена команда `\newtie`, которая добавляет округлую крышечку над буквой ($\text{\AA}$). Чтобы заключить букву в круг, можно воспользоваться командой `\textcircled` ($\text{\textcircled{A}}$).

Акценты в текстовой моде

$\text{\AA}$	<code>\text{"A}\text{"a}</code>	$\text{\AA}$	<code>\text{'A}\text{'a}</code>	$\text{\AA}$	<code>\text{.A}\text{.a}</code>	$\text{\AA}$	<code>\text{=A}\text{=a}</code>
$\text{\AA}$	<code>\text{^A}\text{^a}</code>	$\text{\AA}$	<code>\text{'A}\text{'a}</code>	$\text{\AA}$	<code>\text{~A}\text{~a}</code>	$\text{\AA}$	<code>\text{bA}\text{b a}</code>
$\text{\AA}$	<code>\text{cA}\text{c a}</code>	$\text{\AA}$	<code>\text{dA}\text{d a}</code>	$\text{\AA}$	<code>\text{kA}\text{k a}</code>	$\text{\AA}$	<code>\text{rA}\text{r a}</code>
$\text{\AA}$	<code>\text{tA}\text{t a}</code>	$\text{\AA}$	<code>\text{uA}\text{u a}</code>	$\text{\AA}$	<code>\text{UA}\text{U a}</code>	$\text{\AA}$	<code>\text{vA}\text{v a}</code>

10.3. К вопросу о шрифтах

Говорить, что кириллических шрифтов нет совсем, было бы неправильно. Они есть, но их *очень* мало.

Если не рассматривать коммерческие шрифты, то при использовании $\text{\LaTeX}$ наиболее предпочтительнее шрифты семейства Computer Modern, которые доступны в формате Type1 в рамках пакета **cm-super**. Эти шрифты используются в дистрибутиве $\text{\TeX}$ Live по умолчанию. Кириллица в семейство Computer Modern была добавлена усилиями Ольги Лапко (пакет **lh**).

Шрифты семейства Computer Modern имеют все необходимые начертания (*курсив*, **жирный текст**, КАПИТЕЛЬ и *наклонный шрифт*) и хорошо согласуются с математическими символами (рис. 10.1).

Альтернативой шрифтам Computer Modern может служить семейство Concrete. Это семейство было создано Д. Э. Кнудом из шрифтов Computer Modern и хорошо согласуется с математическими символами Euler Германа Цапфа. Для использования этих шрифтов достаточно загрузить в преамбуле стиль **concrete** (рис. 10.2).

В шрифтах Antykwa Toruńska также есть кириллица. Для инициализации следует загрузить стиль **anttor** (рис. 10.3). Вероятно, эти шрифты будут лучше сочетаться с математическими символами Euler, которые можно загрузить с помощью стиля **euler**.

Выбор не богатый, но это лучше, чем ничего. Перечисленные ранее шрифты доступны в дистрибутиве $\text{\TeX}$ Live «из коробки». При этом никто не ограничивает в подключении сторонних шрифтов. Подключение шрифтов формата Type1 подробно описано в замечательном документе «The Font Installation Guide» (файл `fontinstallationguide.pdf`).

«Здравствуй, мир!!!»

«Здравствуй, мир!!!»

$$\frac{1}{\sqrt{2\pi}\sigma} \int_0^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 \quad \frac{1}{\sqrt{2\pi}\sigma} \int_0^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1$$

Рис. 10.1. Шрифт Computer Modern

Рис. 10.2. Шрифты Concrete и Euler

<<Здравствуй, мир!!!>>

$$\frac{1}{\sqrt{2\pi}\sigma} \int_0^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1$$

Рис. 10.3. Шрифты Antykwa Toruńska и Euler
(в текстовом шрифте отсутствуют «ёлочки»)

10.4. Работа с текстом

Спокойное течение текста часто хочется прервать восклицанием, цитатой авторитета или просто списком покупок.

10.4.1. Выделение текста

А разделе 3.3 «Логика набора» на стр. 36 рассказано, как выделять текст с помощью смены насыщенности и начертания шрифта. Обычно этого хватает с избытком. Более того, классические методы выделения, такие как подчёркивание и ~~разрядка~~ пришли из эпохи печатных машинок, когда ощущалась значительная нехватка средств выразительности. Но в любом случае иногда бывает нужно даже это, и L^AT_EX, естественно, обладает необходимым функционалом.

soul

Пакет **soul** специализируется на различных способах выделения текста:

Вот так выглядит `\so{разрядка}`,
`\ul{подчёркивание}` и `\st{перечёркивание}`.

Вот так выглядит `разрядка`,
подчёркивание и ~~перечёркивание~~.

В пакете **soul** имеется возможность цветового выделения текста. При использовании команд **soul** выделенный текст корректно переносится, внутри команд можно менять шрифт и начертание, но команды **soul** нельзя вкладывать друг в друга. Также имеется ещё некоторый набор ограничений. Для выяснения подробностей лучше обратиться к документации `soul.pdf`.

10.4.2. Боксы

TeX управляется с боксами. Он сам их нарезает и склеивает, но при желании можно создать бокс самому.

Самый простой способ создать бокс — это команда `\mbox{текст}`. В разделе «Пунктуация» на странице 39 эта команда использовалась для запрета переноса в слове. `\mbox` не производит никаких видимых эффектов. `\fbox` же напротив создаёт рамочку вокруг созданного бокса, а в остальном полностью эквивалентен `\mbox`. Обе упомянутые команды автоматически выставляют размеры создаваемых боксов. Для задания своих размеров существуют команды `\makebox` и `\framebox`:

```
\centering
\framebox[\textwidth][l]{АКАДЕМИЯ\ НАУК}
```

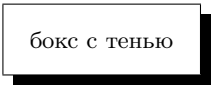



```
\framebox[0.5\width][c]{АКАДЕМИЯ\ НАУК}
```

Первый необязательный параметр — это ширина создаваемого бокса, второй — позиция текста внутри бокса. Позиция текста определяется с помощью одной буквы: `l` — сдвигает текст влево, `r` — вправо, `c` — размещает по центру, `s` — растягивает текст за счёт пробелов на всю заявленную ширину. При указании размеров нового бокса можно использовать стандартные макросы `\width` (ширина текста в аргументе), `\totalheight` (полная высота текста), `\height` (*высота* текста — расстояние от базовой линии до верхней точки) и `\depth` (*глубина* текста — расстояние от базовой линии до нижней точки). После определения бокса L^AT_EX больше не обращает внимание на границы текста, а смотрит только на размеры.

Макрос `\shabox`, определённый в стиле **shadow**, может несколько украсить рамочку бокса:

```
\centering\shabox{бокс с тенью}
```



Есть и другие способы «украшения» и задания боксов. Некоторые из пакетов перечислены на стр. 252.

Для измерения размеров ненапряжённого текста можно воспользоваться командами:

```
\settowidth{длина}{измеряемый текст}
\settoheight{длина}{измеряемый текст}
\settodepth{длина}{измеряемый текст}
```

Здесь *длина* — переменная, которой присваивается измеренное значение ширины, высоты и глубины текста соответственно.

Для создания бокса более чем в одну строку используются окружение `minipage` или команда `parbox`:

```
\parbox{0.3\textwidth}
  {Объявление в газете:}\quad
\begin{minipage}{0.62\textwidth}
  Ищу работу машинистки. Печатаю со
  скоростью 4 тысячи знаков в минуту.
  Правда, такая белиберда получается!
\end{minipage}
```

Объявление
в газете: Ищу работу машинист-
ки. Печатаю со скоро-
стью 4 тысячи знаков в
минуту. Правда, такая
белиберда получается!

Подробнее об этих командах рассказано в разделе «Упаковка картинок в один float» на стр. 64.

10.4.3. Цитаты

Традиционно цитаты текста выделяются отступами с обеих сторон параграфа. Для такого форматирования в $\text{\LaTeX}$ можно использовать окружения `quote` (используется для коротких цитат) и `quotation` (в отличие от `quote` печатается с красной строки).

```
Описывая Гекльберри Финна,  
Марк Твен писал:  
\begin{quotation}  
  Кое-что он преувеличивал, но в  
  основном говорил правду.  
\end{quotation}
```

Описывая Гекльберри Финна, Марк
Твен писал:

Кое-что он преувеличи-
вал, но в основном говорил
правду.

Ещё один способ оформить цитату — добавление к ней эпиграфа. Для этого существует специальный пакет, который, как это не сложно догадаться, так и называется **epigraph**. Для того чтобы добавить эпиграф в своё произведение, можно воспользоваться командой:

```
\epigraph{«текст»}{«источник»}
```

где «текст» — это тело эпиграфа, а «источник» — тот, кто ответственен за создание тела. По умолчанию эпиграф помещается справа. Именно с помощью этой команды сформированы все эпиграфы в этой книге.

Ширину текста в эпиграфе можно установить с помощью изменения переменной `\epigraphwidth` с помощью команды `\setlength`, например, так:

```
\setlength{\epigraphwidth}{0.3\textwidth}
```

Перед началом каждой главы (`\chapter`) делается значительный пропуск, чтобы логически обозначить начало новой части. Это пустое место идеально подходит для вставки эпиграфа. Для этого используется команда `\epigraphhead`:

```
\chapter*{новая глава}  
\epigraphhead[«расстояние»]{\epigraph{«тело»}{«источник»}}
```

Необязательный параметр «расстояние» представляет число, которое характеризует отступ в терминах текущего значения переменной `\unitlength`. Для главы типичный отступ равен 70. Если эпиграф слишком большой, то заголовок главы можно чуть опустить с помощью команды `\dropchapter{длина}`.

Документация вместе с исходниками пакета находится в файле `epigraph.dtx`.

10.4.4. Перечни

Перечни уже были упомянуты в разделе 3.3.1 на стр. 37. Существует три основных окружения для создания перечней:

`enumerate` — нумерованный список,
`itemize` — ненумерованный список,
`description` — список без метки по умолчанию.

Внутри окружения пункты перечня разделяются с помощью команды `\item`:

```
\begin{enumerate}
\item первый пункт,\label{ex:first}      1) первый пункт,
\item второй пункт.                      2) второй пункт.
\end{enumerate}
В пункте \ref{ex:first}\ldots           В пункте 1...
```

В качестве необязательного параметра команде `\item` можно передать ключевое слово. Обычно ключевое слово передают в случае окружения `description`, так как оно не имеет ключевого слова по умолчанию:

```
\begin{description}
\item[enumerate] "--- нумерованный,      enumerate — нумерованный,
\item[itemize]   "--- ненумерованный,    itemize — ненумерованный,
\item[description] "--- без метки.        description — без метки.
\end{description}
```

Перечни можно вкладывать друг в друга, создавая вложенные списки. При этом каждый следующий уровень форматируется по своему. Вот так, например, выглядят разные уровни окружений `enumerate` и `itemize`:

1) первый уровень перечня `enumerate`

а) второй уровень

— третий уровень

А. четвёртый уровень

2) опять первый уровень `enumerate`

• второй уровень окружения `itemize`

— третий уровень

* четвёртый уровень

· пятый уровень

Чтобы заставить окружение `enumerate` нумеровать списки `\item` по-русски, то после инициализации пакета **babel** следует выполнить команду:

```
\renewcommand{\theenumi}{\asbuk{enumi}}
```

Это поменяет формат нумерации на верхнем уровне. На следующих уровнях используются счётчики `enumii`, `enumiii`, `enumiv` по возрастающей. Подробнее про счётчики рассказано в разделе 6.2 на стр. 71.

eqlist

Из примера с окружением `description` видно, что в случае разных по длине меток текст описания не выравнивается. Это можно исправить, загрузив пакет **eqlist**, в котором определяется одноимённое окружение:

```
\begin{eqlist*}
  \item[enumerate] "--- нумерованный,      enumerate — нумерованный,
  \item[itemize]   "--- ненумерованный,     itemize   — ненумерованный,
  \item[description] "--- без метки.         description — без метки.
\end{eqlist*}
```

Окружения `eqlist` и `eqlist*` отличаются друг от друга тем, что окружение `eqlist*` чуть более компактно. Если одна из меток списка слишком длинная для того, чтобы по ней выравниваться, то этот пункт лучше отметить командой `\longitem`. Подробности можно почерпнуть в файле документации **eqlist.pdf**.

desclist

Схожую с пакетом **eqlist** функциональность предоставляет стиль **desclist** из пакета **ncctools**

```
\begin{desclist}{\bfseries}
      { \hfill---}[description]      enumerate — нумерованный,
  \item[enumerate] нумерованный,      itemize   — ненумерованный,
  \item[itemize]   ненумерованный,    description — без метки.
  \item[description] без метки.
\end{desclist}
```

Первый параметр окружения `desclist` — это то, что ставится перед меткой, второй параметр — это то, что ставится после метки, а необязательный параметр представляет собой текст, по которому происходит выравнивание. Подробности можно узнать в файле **desclist.pdf**.

paralist

Кроме многоабзадных перечней, о которых речь шла ранее, перечни могут быть и внутриабзачными, для их автоматического формирования можно воспользоваться пакетом **paralist**:

Перечни формируются с помощью окружений

```
\begin{inparaenum}
  \item enumerate,
  \item itemize и
  \item description.
\end{inparaenum}
```

Перечни формируются с помощью окружений 1) `enumerate`, 2) `itemize` и 3) `description`.

Для симметрии к нумерованному окружению `inparaenum` есть ненумерованное `inparaitem`. Документация доступна в исходнике пакета **paralist.dtx**.

10.5. Рубрикация

«Нужны ли книге оглавление или содержание? Любой скажет: что за вопрос, конечно, нужны. И не только в книге научной и деловой. В любой». А. Э. Мильчин.

Для оформления разделов в основном используются команды секционирования `\section`, `\subsection`, `\subsubsection`, `\paragraph` и `\subparagraph`. Команды перечислены в порядке убывания значимости при рубрикации.

Кроме самих заголовков разделов, созданных с помощью команд секционирования, часто в начале создаётся и оглавление. За создание оглавления отвечает команда `\tableofcontents`. При каждой компиляции информация о разделах собирается в файле с тем же именем, что и у `tex`-файла, но с расширением `toc`. При следующей компиляции эта обновлённая информация о разделах используется при создании оглавления.

Обычно любой сколько-нибудь сложный текст следует начинать с планирования структуры, то есть создать план-шаблон будущего документа. На рис. 10.4 с точностью до размеров листа бумаги представлен результат компиляции следующего за этим абзацем кода.

```
\documentclass[10pt]{scrartcl}
\usepackage{warn}{mathtext}
\usepackage{T2A}{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
\usepackage[indentfirst}
\title{Пример рубрикации}
\author{A.\,B.~Top}
\begin{document}
\maketitle{}
```

Пример рубрикации

А. В. Тор

1 января 2013 г.

Содержание

1. Раздел	1
1.1. Подраздел	1
1.1.1. Подподраздел	2
Закключение	2
А. Приложение	2

1. Раздел

Основной элемент рубрикации.

1.1. Подраздел

Вспомогательный элемент рубрикации.

1.1.1. Что-то более мелкое, чем подраздел

Вспомогательный для вспомогательного. В содержании выводится краткая версия заголовка.

Параграф Важный параграф.

Подпараграф Параграф чуть менее важный.

Раздел, отсутствующий в содержании

Всяко бывает. Иногда и такое нужно.

Закключение

Закключение в отличие от, скажем, раздела 1.1 на странице 1 нумеровать не надо, но в содержании отразить необходимо.

А. Приложение

Рис. 10.4. Пример рубрикации

```
\tableofcontents{}

\section{Раздел}
\label{sec:section}
Основной элемент рубрикации.

\subsection{Подраздел}
\label{sec:subsection}
Вспомогательный элемент рубрикации.

\subsubsection[Подподраздел]{Более мелкое, чем подраздел}
\label{sec:subsubsection}
Вспомогательный для вспомогательного. В содержании выводится
краткая версия заголовка.

\paragraph{Параграф}
\label{sec:paragraph}
Важный параграф.

\subparagraph{Подпараграф}
\label{sec:subparagraph}
Параграф чуть менее важный.

\section*{Раздел, отсутствующий в содержании}

Всяко бывает. Иногда и такое нужно.
```

```
\section*{Заключение}
\label{sec:afterwords}
\addcontentsline{toc}{section}{Заключение}
```

Заключение в отличие от, скажем, раздела `\ref{sec:subsection}` на странице `\pageref{sec:subsection}` нумеровать не надо, но в содержании отразить необходимо.

```
\appendix
```

```
\section{Приложение}
\label{appendix}
```

```
\end{document}
```

Раздел типа `\section` соответствует первому уровню секционирования, раздел `\subsection` — второму и так далее. При желании можно ограничить уровень отображения информации в оглавлении. Для этого следует изменить значение переменной `tocdepth`, например, так:

```
\setcounter{tocdepth}{2}
```

В этом случае будет показана информация о разделах вплоть до второго уровня включительно.

Кроме оглавления также можно вывести список иллюстраций `\listoffigures` и таблиц `\listoftables`. Информация об иллюстрациях и таблицах автоматически собирается в файлах с расширениями `lof` и `lot`.

Для добавления какой-то информации в оглавление в обход команд секционирования можно воспользоваться командой

```
\addcontentsline{toc}{«уровень раздела»}{«строка в оглавлении»}
```

У этой команды три аргумента. Первый аргумент соответствует расширению файла (`toc`, `lof` или `lot`), куда добавляется «строка в оглавлении». Уровень раздела определяется именами команд секционирования, то есть `section`, `subsection` и так далее. Команды ЛАТЭХ при передаче в файлы списков следует защищать командой `\protect`, дабы избежать проблем с «хрупкими» инструкциями.

Команда `\appendix` отделяет приложение от основного текста. После её вызова правила нумерации разделов изменяются. `\appendix` тоже является командой секционирования.

При оформлении книг (производные от **book** классы, например, **scrbook**) к командам рубрикации добавляется ещё два уровня `\chapter` — глава и `\part` — часть. Помимо этого традиционно крупные художественные формы разбиваются на три

части: *вступительную* — открывается командой `\frontmatter` (нумеруется римскими цифрами), *основную* — открывается командой `\mainmatter` (нумеруется арабскими цифрами) и *заключительную* — открывается командой `\backmatter`. При этом главы нумеруются только в основной части. Во вступительной части обычно помещают титульный лист (окружение `titlepage`) и оглавление:

```
\documentclass[a4paper,12pt]{scrbook}
```

```
...
```

```
% Создать указатель
```

```
\makeindex
```

```
\begin{document}
```

```
% Вступительная часть
```

```
\frontmatter
```

```
\begin{titlepage}
```

```
    Титульный лист
```

```
\end{titlepage}
```

```
\tableofcontents{}
```

```
% Основная часть
```

```
\mainmatter
```

```
\chapter{Глава 1}
```

```
\label{cha:chapter-1}
```

```
\chapter{Глава 2}
```

```
\label{cha:chapter-2}
```

```
% Заключительная часть
```

```
\backmatter
```

```
% Библиография
```

```
\bibliographystyle{gost780u}
```

```
\bibliography{bibfile}
```

```
% Вывести указатель
```

```
\printindex
```

```
\end{document}
```


10.6. Ссылки, сноски и примечания

Иногда не хочется разбивать канву повествования и в то же время необходимо как-то вставить пояснение. Это можно сделать, просто сославшись на какой-то другой фрагмент текста с помощью ссылки, или вставить пояснение на этой же странице с помощью подстрочного примечания, или отнести пояснение в конец книги.

10.6.1. Механизм ссылок

В примере на стр. 133, демонстрирующем работу команд секционирования, вслед за каждой командой секционирования ставилась метка с помощью инструкции `\label`. Метка представляет из себя последовательность ASCII-символов. При компиляции документа информация об имеющихся метках добавляется в файл с расширением `aux`. Для того чтобы извлечь эту информацию, то есть номер раздела (команда `\ref`) или номер страницы (`\pageref`), необходимо пропустить текст через **latex** ещё раз.

Для того чтобы можно было сослаться на внешний документ, следует воспользоваться пакетом **xr**. В данном случае в преамбуле необходимо добавить примерно следующие инструкции:

```
\usepackage{xr}
\externaldocument [EXT-]{externaldoc}
```

Это позволяет получить доступ к меткам файла `externaldoc.tex`. Обращение к меткам, как и обычно, осуществляется с помощью команд `\ref`/`\pageref`, только перед именем метки добавляется префикс `EXT-`. Можно обойтись и без префикса, так как данный параметр является опциональным, но в этом случае повышается вероятность конфликта из-за одинаковых меток.

Ссылаться можно не только на разделы. Метки внутри нумерованных окружений, типа `equation` (выключенные математические формулы) или `theorem` (теоремы), принимают их номер. Это также касается рисунков (окружение `figure`) и таблиц (окружение `tabular`). В таком случае `\label` должна следовать сразу за командой `\caption`, формирующей подпись к плавающему объекту.

Обычно ссылку на формулу заключают в скобки. Для того чтобы не путаться, в пакете **amsmath** определена ещё одна команда ссылки `\eqref`. Для правильной ссылки на номера формул вместо команды `\ref` следует *всегда* использовать инструкцию `\eqref`:

```
\begin{equation}
\label{eq:1}
x_{\{1,2\}}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\end{equation}
Обычная ссылка~\ref{eq:1}\par
Ссылка на формулу~\eqref{eq:1}
```

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (10.1)$$

Обычная ссылка 10.1
Ссылка на формулу (10.1)

Из примера видно, что метка состоит из двух частей в виде префикса `ex` и остатка. Обычно для каждого окружения выбирают свой префикс. Этот обычай как нельзя кстати, если воспользоваться возможностями пакета **prettyref**. Данный пакет в зависимости от префикса метки позволяет сформировать внешний вид ссылки.

%связываем префикс с новым видом

`\newrefformat{eq}{формула \No~\ref{#1}}`

Новый вид ссылки: `\prettyref{eq:1}`

Новый вид ссылки: формула № 10.1

При использовании `\prettyref` для формирования ссылки в имени метки не должно быть более одного двоеточия (:), разделяющего префикс и остаток.

► При подготовке документов с большим количеством ссылок можно в них запутаться. Для визуализации меток можно воспользоваться пакетом **showkeys**. При инициализации пакета все метки и ссылки подписываются, что облегчает процесс поиска неправильных ссылок.

10.6.2. Подстрочные примечания

Подстрочное примечание формируется с помощью команды `\footnote`. Правила оформления примечаний прописываются в определении класса. Без особых на то причин менять эти правила не стоит. Примечание, если позволяет место, печатается на той же странице, где помещена ссылка. Текст и примечание отделяются разделительной линией.

Примечание можно добавлять и внутри окружения `minipage`, но тогда оно печатается внутри окружения:

`\begin{minipage}{1.0\linewidth}`

Ссылки `раз\footnote{Сноска.}` и

два `\footnote[26]{Подстрочное примечание.}`.

`\end{minipage}`

Ссылки ^а раз и два^з.

^аСноска.

^зПодстрочное примечание.

Необязательный параметр `\footnote` позволяет присвоить примечанию значение по выбору пользователя.

Для того чтобы можно было сделать сноску внутри заголовка раздела, необходимо защитить инструкцию `\footnote` командой `\protect`, так как команда создания подстрочного примечания является «хрупкой»:

```
\section{Заголовок\protect\footnote{Подстрочное примечание.}}
```

В сложных ситуациях, например, когда требуется сделать подстрочное примечание внутри бокса, для формирования сноски нужно прибегнуть к независимым командам создания ссылки и создания примечания соответственно:

```
\footnotemark[num]
```

```
\footnotetext[num]{«сноска»}
```

Необязательный параметр `num`, как и в случае команды `\footnote`, позволяет формировать свою нумерацию. Для хранения текущего номера ссылки используется счётчик `footnote`.

nccfoots

Стиль **nccfoots** из пакета **ncctools** позволяет организовать ручную нумерацию подстрочных примечаний:

<code>\begin{minipage}{1.0\linewidth}</code>	Ссылки раз ^α и два ^β .
Ссылки раз <code>\Footnote{\${\alpha\$}}{Сноска.}</code> и	
два <code>\Footnotemark{\${\beta\$}}.</code>	^α Сноска.
<code>\Footnotetext{\${\beta\$}}{Подстрочное примечание.}</code>	^β Подстрочное примечание.
<code>\end{minipage}</code>	

У аналогов стандартных команд есть дополнительный обязательный параметр, куда следует поместить метку.

footmisc

Стилевой файл **footmisc** (документация **footmisc.pdf**) позволяет управлять параметрами сносок. При загрузке можно передать следующие опции:

- perpage** — нумерация сносок при переходе на новую страницу опять начинается с единицы;
- para** — подстрочные примечания печатаются подряд как единый параграф без переноса строки. Иногда такое оформление востребовано в критических заметках, где требуется много сносок;
- side** — печать подстрочного примечания на полях;
- symbol*** — знак сноски становится символьным;
- norule*** — перед примечанием не печатается разделяющая линейка.

savefnmark

С помощью пакета **savefnmark** можно переиспользовать старые метки. В нём определяются две команды:

- `\saveFN` — сохраняет последнюю метку в именованную структуру, следующую за командой;
- `\useFN` — восстанавливает сохранённую метку.

Сноска <code>\footnote{примечание}\saveFN\oldfoot\par</code>	Сноска ^a
Переиспользуем сноску <code>\useFN\oldfoot</code>	Переиспользуем сноску ^a
	^a примечание

При желании можно сохранять номер сноски внутри таблицы для переиспользования его в тексте и наоборот.

10.6.3. Затекстовые примечания

Иногда, когда текст в сносках может усложнять прочтение документа, можно вместо подстрочных примечаний организовать затекстовые. По возможности этой структуры следует избегать.

endnotes

Для создания затекстовых примечаний можно воспользоваться стилевым файлом **endnotes** из пакета **ltxmisc**. В пакете определён набор макросов совершенно аналогичных командам для формирования подстрочных примечаний: `\endnote`, `\endnotemark` и `\endnotetext`. Дополнительно определена команда для простого добавления текста без всякой метки `\addtoendnotes{текст}`. Для печати затекстовых комментариев в конце текста следует добавить команды:

```
\renewcommand{\notesname}{Затекстовые комментарии}
\theendnotes
```

Информация об использовании стиля **endnotes** находится в самом стилевом файле **endnotes.sty**.

pagenote

Аналогичную **endnotes** функциональность предоставляет и пакет **pagenote**.

```
% Преамбула
\usepackage[page]{pagenote}
\makepagenote
...
% Тело документа
Текст\pagenote{Примечание}
...
\renewcommand{\notesname}{Затекстовые комментарии}
\printnotes
```

Опция **page** добавляет при формировании списка затекстовых примечаний номера страниц, на которых на них ссылались. Также пакету можно передать опцию **continuous**. В этом случае при окончании главы нумерация затекстовых примечаний не обнуляется.

Команда `\printnotes` печатает все определённые до неё затекстовые примечания и запрещает добавление новых. В отличие от неё команда `\printnotes*` позволяет использовать себя несколько раз, например, в конце каждой главы. `\printnotes*`

печатает уже определённый до неё список примечаний и очищает его для последующих добавлений.

10.6.4. Заметки на полях

Большие поля некоторых классов, например, стандартных типа **article**, позволяют оставлять на них заметки прямо по ходу текста. Заметки на полях можно оставлять с помощью команды `\marginpar{текст}`. У команды есть необязательный аргумент, который если определён, то печатается, если заметка попадает на левое поле. Для того чтобы поменять поле по умолчанию, на котором печатается заметка, на другое, следует воспользоваться командой `\reversemarginpar`, а восстановить старый порядок можно с помощью `\normalmarginpar`.

marginnote

Команда `\marginpar` при формировании заметки создаёт плавающий объект, и поэтому в некоторых случаях её нельзя использовать, например, в подстрочных примечаниях или в других плавающих объектах. Пакет **marginnote** определяет одноимённый макрос для создания заметок, который лишён этого недостатка:

```
\marginnote [ «заметка слева» ] { «заметка справа» } [ «сдвиг» ]
```

Первые два аргумента полностью соответствуют `\marginpar`, а последний необязательный аргумент может применяться для вертикального выравнивания заметки на полях. Положительное значение «сдвига» перемещает заметку вниз, а отрицательное — вверх. Подробности можно найти в файле **marginnote.pdf**.

► Иногда при формировании заметок на полях в начале страницы заметка оказывается не с той стороны, с которой должна быть по умолчанию. Для исправления этого бага ЛАТЭХ был написан стиль **mparhack** — его достаточно просто загрузить.

changebar

На полях можно оставлять не только текстовые метки. Пакет **changebar** написан для индикации изменений в тексте. Изменённый текст следует окружить макросами `\cbstart` и `\cbend`, и он будет помечен как этот параграф. Пакет может работать как **dvips**, так и **pdflatex**. Толщину линии и цвет можно легко настроить. Более подробно об этом пакете можно узнать в файле **changebar.pdf**.

10.6.5. WWW-элементы

Общего рецепта, как делать ссылки на электронные ресурсы, нет. Проще всего использовать команду `\url` из одноимённого пакета.

url

Пакет **url** позволяет печатать **www**-адреса, не особо заботясь об экранировании спецсимволов.

Моя WWW-страничка находится тут:

```
\url{http://www.inp.nsk.su/
~baldin/}.
```

Моя WWW-страничка находится тут:

```
http://www.inp.nsk.su/~baldin/.
```

В тексте ссылки не допускаются символы `%`, `#`, `^`, и она не должна заканчиваться символом `\`. Если есть желание уйти и от этих ограничений, то аналогично команде `\verb` инструкцию можно использовать и так: `\url!http://www.адрес.ru!`.

hyperref

В плане создания гиперссылок особенно интересен пакет **hyperref**. Этот пакет не просто позволяет печатать ссылки на электронные ресурсы — он их реально делает. Достаточно загрузить пакет:

```
% Загрузка пакет hyperref
\usepackage[unicode=true]{hyperref}
```

и в документе появятся гиперссылки. Гиперссылками станут и обычные ссылки, и оглавления. Также они появятся в алфавитных указателях и в библиографии. Пакет ориентирован на использование для pdf-документов, но гиперссылки можно увидеть и с помощью некоторых программ просмотра dvi-файлов, например, с помощью **xdvi**. Опция `unicode=true` нужна, чтобы сформировать русские закладки.

Подробная инструкция об использовании **hyperref** находится в директории документации пакета в файле **manual.pdf**. В сети есть замечательная авторская документация по этому пакету на русском языке, созданная Владимиром Сюткиным: http://www-sbras.nsc.ru/win/docs/TeX/LaTeX2e/docs_koi.html.

► Пакет **hyperref** конфликтует со множеством других пакетов **LaTeX**.

breakurl

При использовании **pdflatex** совместно с **hyperref** **www**-адреса автоматически переносятся на следующую строку, но если используется классическая схема в виде **latex**→**dvips**→**ps2pdf**, то автоматического переноса не происходит. Эту проблему и призван решить пакет **breakurl**.

Пакет **breakurl** следует загружать после **hyperref**. В пакете определена команда `\burl`, которая является идентичной по использованию `\url`.

Моя WWW-страничка:

```
\burl{http://www.inp.nsk.su/~baldin/}.
```

Моя WWW-страничка: <http://www.inp.nsk.su/~baldin/>.

По умолчанию, если пакет не загружается с опцией `preserveurlmacro`, то `\url` переопределяется как синоним `\burl`. Документация к пакету — `breakurl.pdf`.

10.7. Процесс подготовки документа

Подготовка текста — непростое занятие. Непростое и потому, что с первого раза почти никогда ничего не получается.

10.7.1. Авторские метки с FiXme

Далеко не всегда документ создаётся «набело». Постоянно в процессе подготовки текста возникают вопросы, которые не удаётся решить сразу. Это нормально. Проблема заключается лишь в том, что требуется как-то отметить проблемный кусок, чтобы затем его доделать.

Пакет `FiXme` предоставляет четыре команды:

<code>\fxnote</code>	— заметка;
<code>\fxwarning</code>	— предупреждение;
<code>\fxerror</code>	— не фатальная ошибка;
<code>\fixme</code>	— требование для обязательного исправления.

В качестве аргумента каждой из этих команд передаётся строка текста.

При активизации опции `draft` при загрузке стиля документа, например, так:

```
\documentclass [a4paper, 12pt, draft] {scrbook}
```

текст, передаваемый этим командам в качестве аргумента, по умолчанию выводится на полях документа. При отсутствии опции `draft` или наличии опции `final` заметки исчезают. Команда `\listoffixmes` формирует список заметок по подобию оглавления.

Кроме этого команда `\fixme` инспирирует ошибку при компиляции документа, что не позволит при окончательной вёрстке пропустить важные нерешённые проблемы.

Для подключения пакета необходимо загрузить стиль `fixme`. В качестве параметра при загрузке можно передать информацию о размещении предупреждений: `inline` — разместить прямо в тексте, `margin` — разместить на полях (по умолчанию), `footnote` — выводить предупреждения как подстрочные примечания и `index` — разместить в алфавитном указателе. Информация по пакету в `fixme.pdf`.

10.7.2. Нумерация строк с lineno

Пакет `lineno` предназначен исключительно для нумерации строк в документе. Это упрощает навигацию внутри текста для указания конкретной ошибки.

Самый простой способ пронумеровать небольшой фрагмент — это воспользоваться окружением `linenumbers`. Для больших фрагментов лучше использовать

пару деклараций `\linenumbers/\nolinenumbers`, которые, соответственно, включают и выключают нумерацию.

По умолчанию нумерация идёт сквозная и размещается слева от текста. Для переключения режима, когда с каждой новой страницей нумерация начинается с единицы, следует воспользоваться макросом `\setpagewiselinenumbers`. Для того чтобы нумерация всегда располагалась с внешней части документа, используется макрос `\switchlinenumbers`, а при добавлении * (звёздочки) нумерация всегда будет на внутренних полях документа.

Метку для ссылки на строку можно установить так: `\linelabel{ex:lineno}`. А теперь с помощью обычной команды `\ref` можно сослаться на эту строку №9. Эта возможность не корректно работает с пакетом **hyperref** (создаётся неопределённая гиперссылка). Дополнительную информацию для пользователя следует искать в самом конце файла `lineno.pdf`.

10.7.3. Водяные знаки

Л^AT_EX, Linux и русский стиль.

© Е.М.Балдин*



*имя: Е.М.Балдин, email: e.m.bal@yandex.ru
Заголовок: ЛАТЭХ и МИКРОСОФТ, создатель: Драйвон Бабиб, вкладки со страницей Д.Э. Кортис.

Рис. 10.5. **draftwatermark**

Чтобы показать, что копия документа не окончательная, или просто вставить какой-то текст в качестве фона, можно воспользоваться специализированным пакетом **draftwatermark**. С помощью него можно делать «водяные знаки». Этот пакет полностью перекрывает возможности ранее популярного, но устаревшего пакета **draftcopy**. После загрузки пакета через всю страницу по диагонали печатается слово «DRAFT». Пакет совместим как с **dvips**, так и с **pdflatex**.

Стилю можно передать для печати своё сообщение и изменить размер шрифта. На рис. 10.5 изображена страница, на фоне которой идёт надпись «Рабочая версия». Для того чтобы её получить, в преамбуле были добавлены следующие строки:

```
% Загрузка пакета
\usepackage{draftwatermark}
% Масштабирование базового шрифта
\SetWatermarkScale{2.3}
% Изменение насыщенности
\SetWatermarkLightness{0.3}
% Пользовательская строка
\SetWatermarkText{\textbf{Рабочая версия}}
```


Если при загрузке **draftwatermark** ему передаётся опция **firstpage**, то сообщение печатается только на первой странице. Максимальная насыщенность текста соответствует 0, а минимальная — 1. Размер шрифта можно задать и напрямую с помощью макроса `\SetWatermarkFontSize`. Для задания угла наклона фоновой сообщения есть команда `\SetWatermarkAngle{угол}`.

► Стиль **watermark** из пакета **ncctools** позволяет выводить на подложке страницы не только текст, но и графические объекты. Схожую функциональность предоставляют пакеты **bophook**, **eso-pic** и **wallpaper**.

10.7.4. Сложные документы

Всё можно хранить в одном файле — это ничему не противоречит. Более того, само понятие файл для пользователя не так уж и необходимо. Но уж если файл есть, то почему бы не разделить большой текст на несколько частично независимых фрагментов?

С помощью команды `\input{имя файла}` можно вставлять другой `tex`-файл в документ. `ЛАТЭХ` просто добавляет содержимое по месту команды, считывая файл либо до конца, либо до первой встретившейся инструкции `\endinput`. Совершенно не важно, в какой части документа встречается `\input`. Бывает довольно удобно вынести преамбулу в отдельный файл. В имени файла можно опустить расширение `tex`.

Для включения больших фрагментов текста, таких как главы, можно воспользоваться декларацией `\include`. В преамбуле с помощью команды `\includeonly` можно перечислить, какие части надо подключить при текущей сборке. При этом сохраняется правильная нумерация страниц, и можно сослаться на *не* включённые в эту сборку разделы. Сохранение правильной нумерации становится возможным, так как при использовании команды `\include` перед вставкой кода добавляется декларация `\cleartpage`, которая гарантирует, что «плавающие объекты» не выйдут за пределы своей главы.

Пример использования `\include`:

```
% Загрузка стандартной преамбулы
\input{preheader}
% Введение и презентация в эту сборку не включаются
\includeonly{
%  intro ,
  base ,
%  presentation
}
\begin{document}
\include{intro} % Введение
\include{base}  % Базовые элементы
\include{presentation} % Презентация
```

```
\end{document}
```

Команда `\include` позволяет включить только `tex`-файл (при написании имени расширение `tex` опускается). Данный метод особенно актуален в случае больших текстов в процессе их создания, так как значительно ускоряет компиляцию. В приведённом ранее примере подключается только глава `base.tex`.

► При помещении команды `\listfiles` в преамбуле можно узнать, какие реально файлы используются при компиляции текста. Список посылается на стандартный вывод при компиляции.

excludeonly

В дополнение к декларации `\includeonly` пакет **excludeonly** добавляет одноимённую команду, которая позволяет исключать переданные в качестве аргумента главы. Это может быть удобнее и нагляднее, чем ставить комментарии.

Для того чтобы, как и ранее по тексту, подключить только `base.tex`, к инструкции `\includeonly` следует добавить `\excludeonly`:

```
\includeonly{intro , base , presentation}
\excludeonly{intro , presentation}
```

askinclude

Для интерактивного добавления глав в основной документ можно воспользоваться пакетом **askinclude**. В этом случае при подключении пакета в процессе компиляции у пользователя запрашивается разрешение на выполнение команд `\include`. Это полезно, если хочется подготовить разные части книги, не модифицируя исходники.

optional

Пакет **optional** из коллекции **ltxmisc** предоставляет пользователю команду `\opt`, используя которую можно выводить нужный вариант текста или кода:

```
% Файл variant.tex
\usepackage[answer1]{optional}

Стандартная шапка "—— выводится всегда.

\opt{variant1,answer1}{\input{variant1}}
\opt{variant2,answer2}{\input{variant2}}

\opt{}{Этот текст никогда не будет напечатан}
```

```
\opt{answer1}{\input{answer1}}
\opt{answer2}{\input{answer2}}
```

Для интерактивного выбора можно воспользоваться командой `\AskOption`, поместив её в преамбуле. Полезно также в этом случае поместить в преамбулу и описание меток:

```
\newcommand{\ExplainOptions}{variant1, variant2 – problems;
  answer1, answer2 – solutions}
\AskOption
```

При компиляции текста будет задан вопрос, какую метку выбрать:

```
...

variant1, variant2 – problems; answer1, answer2 – solutions
Specify which optional text to process:

\UseOption=
```

Указать правильную опцию можно и из командной строки:

```
> latex "\newcommand\UseOption{answer1}\input{variant}"
```

Это легко можно встроить в какой-либо сценарий или написать соответствующий `Makefile`. Подробности можно найти в комментариях к самому коду пакета `optional.sty`.

Глава 11

Всё о таблицах

Я над ней, может быть, двадцать лет думал,
а вы думаете: сидел и вдруг... готово.

*Комментарий Д. И. Менделеева к легенде
о том, что периодическая таблица
элементов ему якобы приснилась*

Далеко не всякое содержание следует облекать в форму таблицы. Но для малых объёмов однородных значимых данных таблицы бывают лучшим способом отображения.

Таблица — это особая форма передачи содержания. Таблица — один из самых сложных текстовых элементов. Сложных не в том смысле, что её сложно оформлять — сложно добывать для неё подходящие данные. Очень легко сделать бессмысленную таблицу. Слишком много информации не оставит по прочтению никакого следа. Всегда надо думать, как и что представлять, но хорошая таблица стоит тысячи слов.

11.1. Немного теории

Честно говоря, описание таблиц с помощью ЛАТ_ЕX-разметки может показать не очень удобным. Действительно, когда вперемешку идут данные и управляющие структуры, исходник таблицы может выглядеть не очень красиво. Спасает только то, что большие таблицы с однородной структурой можно создавать с помощью скрипта, а маленькие таблицы не являются особой проблемой. Как правило, гораздо больше времени, чем оформление таблицы, занимает поиск и верификация данных.

Содержимое таблицы организуется в колонки (графы) и горизонтальные строки таким образом, что каждый элемент является составной частью и строки, и колонки. Таблица состоит из следующих основных элементов: нумерационного и тематического заголовков (номер таблицы и её название), головки¹ (заголовочная часть таблицы), хвоста (вся остальная часть таблицы без головки), боковика (первая слева графа таблицы) и прографки (хвостовая часть таблицы без боковика).

Боковик		Прографка		Головка
ИЛИ		Истина	Ложь	
Истина		Истина	Истина	Хвост
Ложь		Истина	Ложь	

Таблица 11.1. Структура таблицы

Заполняя таблицу текстовыми или цифровыми данными, полезно следовать правилам.

- Выносить данные, общие для каждого элемента графы, в её заголовок, а общие для каждого элемента строки — в заголовок боковика.
- По возможности употреблять числа не более чем из четырёх значащих цифр. Для этого более многозначные числа следует округлять. Общий множитель следует вынести в заголовок. То же самое нужно сделать и с единицами измерения.
- Всегда перед знаком, отделяющим целую часть числа от дробной, должна быть цифра. То есть правильно писать «0.1», но не «.1».
- Проставлять вместо отсутствующих данных многоточие «...», «Нет свед.». Если данных в принципе быть не может, то лучше отметить это с помощью тире «—».
- Не следует использовать знаки, означающее «то же, что и предыдущее значение». Лучше повторить число.

11.2. tabbing

Если точно известна ширина столбцов, и таблица относительно простая, то можно воспользоваться окружением `tabbing`.

¹В «Издательском словаре-справочнике» А. Э. Мильчина [11] утверждается, что попытка замены этого термина по той причине, что он якобы является жаргонным, не привилась.

```
\begin{tabbing}
MMMM \= MMMM \=      \kill
\textbf{\ ИЛИ} \> Истина \> Ложь  \\\
Истина \> Истина \> Истина \\\
Ложь   \> Истина \> Ложь
\end{tabbing}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Первая строчка устанавливает положение табуляторов с помощью команды \=. Команда \kill, завершающая управляющую строку, даёт понять текстовому процессору, что её не надо печатать. Далее идёт обычный текст, где переход к следующей табуляции осуществляется с помощью команды \>, а перевод строки завершается комбинацией \\. Всё просто — точно так же таблицы печатали с помощью обычной печатной машинки, только табуляция задавалась «железным», а не программным способом.

К использованию окружения tabbing следует подходить с известной долей осторожности. Окружение формирует абзац, состоящий из отдельных строк, в котором нет места переносам и многострочным элементам. Зато это позволяет ЛАТЭХ легко переносить часть таблицы на следующую страницу. Абсолютно всё приходится делать своими руками, но в этом есть своеобразная прелесть. Часто tabbing становится базой для определения других более специализированных окружений.

Некоторые стандартные команды ЛАТЭХ внутри tabbing переопределены. Это в частности касается команд переноса (\-) и акцентирования (\' и \'). Для сохранения/воспроизводства текущей табуляции следует воспользоваться командами \pushtabs/\poptabs.

11.3. tabular и array

Самым популярным окружением для отображения таблиц в ЛАТЭХ является tabular. Окружение array фактически полностью повторяет функционал tabular, но в отличие от последнего работает в математической моде — полезно для создания матриц, которые по своей сути тоже обычные таблицы.

```
\centering
\begin{tabular}[c]{l|ll}
\textbf{\ ИЛИ} & Истина & Ложь \\ \hline
Истина & Истина & Истина \\
Ложь   & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Данные делятся на ячейки с помощью символа «логическое И» или конъюнкции — & . Переход на следующую строку контролируется стандартной командой переноса строки \\. В качестве необязательного параметра команды переноса

строки можно указать дополнительный сдвиг по вертикали. Вертикальный размер каждой из строк автоматически выставляется в соответствии с высотой текста.

tabular создаёт единый объект — таблицу. Как и в случае картинок таблицу удобно заключать в плавающее окружение table:

```
\begin{table}[ht]
\centering% центрируем таблицу
\begin{tabular}[«позиционирование таблицы»]{«формат столбцов»}
«тело таблицы»
\end{tabular}
\caption{«подпись»}\label{tab:metka}
\end{table}
```

Это позволяет автоматически создать нумерационный заголовок таблицы и добавить тематический. Необязательный аргумент окружения tabular даёт возможность указать, как позиционировать всю таблицу по вертикали по отношению к окружающему тексту:

- t — выравнивание по верхней строке, то есть верхняя строка таблицы будет расположена на одном уровне со строкой, где эта таблица размещена;
- c — выравнивание по центру;
- b — выравнивание по нижней строке.

Ширина столбцов вычисляется автоматически по заданному формату, который задаётся через обязательный аргумент окружения. Каждому из столбцов должна соответствовать своя буква:

- l (left) — выравнивание по левому краю;
- c (center) — выравнивание по центру;
- r (right) — выравнивание по правому краю;
- p{«ширина»} — задание колонки определённой ширины. В случае жёстко заданной длины слишком длинный текст может разбиваться на несколько строк.

Если формат столбцов повторяется, то для сокращения записи можно воспользоваться следующей спецификацией:

```
{*{n}}{«формат столбца или столбцов»}
```

Где n — число повторений. Такой вот своеобразный цикл.

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\ \hline \hline
Истина & Истина & Истина \\
Ложь & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Разделительные линии между столбцами задаются с помощью вертикальной черты |. Две вертикальные линии || формируют двойной разграничитель. Горизонтальные линии создаются с помощью команды \hline. По аналогии с двойной вертикальной чертой две команды формируют двойную горизонтальную линию. Инструкция @{} позволяет вставить между столбцами любой символ, указанный в качестве обязательного аргумента. При этом подавляются околостолбцовые промежутки, добавляемые по умолчанию автоматически. Это может быть полезно в случае, если один столбец представляет собой какую-то измеренную величину, а второй — её ошибку, тогда вместо разделительной черты между ними лучше вставить знак ±.

```
\centering
\begin{tabular}{c||p{2cm}@{${\pm}$\ }r|}
\textbf{ИЛИ} & Истина & Ложь & \|[2mm]
\cline{2-3}
Истина & Истина & Истина & 
\cline{1-1}\cline{3-3}
Ложь & Истина & Ложь & \cline{2-2}
\end{tabular}
```

ИЛИ	Истина	±	Ложь
Истина	Истина	±	Истина
Ложь	Истина	±	Ложь

Для того чтобы отчеркнуть только часть столбцов, можно воспользоваться командой \cline{диапазон столбцов}.

Стилевой файл **array** из коллекции **tools** в дополнение к стандартным типам столбцов, используемых в **tabular**, добавляет два новых:

- m{«ширина»}** — то же, что и **p{«ширина»}**, но добавляется вертикальное выравнивание содержимого таблицы по центру этой клетки;
- b{«ширина»}** — то же, что и **m{«ширина»}**, но вертикальное выравнивание содержимого идёт по нижней базовой линии последней строки.

Более подробную информацию о дополнительных настройках, предоставляемых стилем **array**, следует искать в файле документации **array.pdf**.

В окружении **tabular** можно использовать подстрочное примечание. При этом примечание формируется не внизу страницы, а рядом с самой таблицей:

```
\centering
\begin{tabular}{r}\hline\hline
Пример\footnote{подстрочного примечания}.\|
\hline\hline
\end{tabular}
```

Пример ^а .
^а подстрочного примечания

► Подробнее о том, как *должна* выглядеть таблица и как это достигается, можно узнать, например, в статье «Publication quality tables in L^AT_EX» (**booktabs.pdf**), написанной Симоном Фиром (Simon Fear) для пакета **booktabs**. Этот пакет для тех, кто везде любит наводить лоск.

Таблица 1. Логическое **ИЛИ**

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Рис. 11.1. `\caption` размещается до таблицы без поправок

Таблица 1. Логическое **ИЛИ**

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

Рис. 11.2. Используется команда `\topcaption` из пакета **topcapt**

11.3.1. Подписи к таблицам

Создание подписей к таблицам ничем не отличается от создания подписей к рисункам (см. раздел 5.4.4). Но в отличие от рисунка, где подпись традиционно выставляется под ним, подпись для таблицы часто лучше размещать сверху. Это связано, например, с тем, что в тексте могут встречаться многостраничные таблицы.

Для размещения подписи над таблицей проще всего воспользоваться пакетом **topcapt**, в котором определяется команда `\topcaption`, полностью эквивалентная `\caption`:

```
\begin{table}[ht]
\topcaption{«подпись»}\label{tab:metka}
\begin{tabular}
...
\end{tabular}
\end{table}
```

Необходимость использования пакета **topcapt** объясняется тем, что в случае применения `\caption` используются неправильные вертикальные промежутки между подписью и таблицей. Рисунки 11.1 и 11.2 позволяют почувствовать разницу. Того же эффекта можно добиться, просто переопределив значения для вертикальных промежутков:

```
\begin{table}
\setlength{\abovecaptionskip}{0pt}
\setlength{\belowcaptionskip}{10pt}
\caption{«подпись»}\label{tab:metka}
\begin{tabular}
...
\end{tabular}
\end{table}
```

11.3.2. К вопросу о разделительных линиях

По характеру оформления линейками таблицы бывают закрытые (глухие), полужакрытые, открытые. Два настоятельных совета:

- никогда не следует использовать вертикальные линии в таблице;
- двойные линии в оформлении таблицы также лишние.

Если данные в таблице настолько разные, что хочется их поделить вертикальной линией, то лучше сделать две таблицы. Но опять же никто не может запретить использовать разделительные линии.

Стиль **hhline** из коллекции **tools** определяет команду `\hhline`, которая позволяет управлять созданием двойной рамки, не создавая ненужных пересечений. Подробности в документации **hhline.pdf**. Пакет **arydshln** необходим в случае использования пунктирных разделительных линий. Документация к этому пакету **arydshln-man.pdf** более чем исчерпывающая.

cellspace

Иногда при формировании таблицы разделительные линии наезжают на текстовое содержание:

```
\begin{tabular}{c}\hline
 $$e = 1 + \frac{1}{2} + \frac{1}{6} + \dots$ 
 $+ \frac{1}{k!} + \dots$ 
 $\$$ 
\\ \hline
\end{tabular}$ 
```

$$e = 1 + \frac{1}{2} + \frac{1}{6} + \dots + \frac{1}{k!} + \dots$$

Обойти эту проблему поможет пакет **cellspace**. В пакете доопределены дополнительные форматы колонок, при использовании которых гарантируется, что всегда останется небольшой промежуток между линией и текстом:

```
\begin{tabular}{Sc}\hline
 $$e = 1 + \frac{1}{2} + \frac{1}{6} + \dots$ 
 $+ \frac{1}{k!} + \dots$ 
 $\$$ 
\\ \hline
\end{tabular}$ 
```

$$e = 1 + \frac{1}{2} + \frac{1}{6} + \dots + \frac{1}{k!} + \dots$$

Вместо форматов колонок **l**, **c** и **r** следует использовать **Sl**, **Sc** и **Sr** соответственно. Все другие форматы должны определяться внутри формата **S**, например, **S{p{3cm}}**.

11.3.3. Клетки

Для объединения расположенных рядом по горизонтали клеток можно воспользоваться командой `\multicolumn{n}{формат колонки}{текст}`, а для объединения по вертикали: `\multirow{n}{высота}{текст}` из одноимённого пакета.

Пример использования `\multicolumn`:

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\\[2mm]
\hline\hline
Истина & \multicolumn{2}{c|}{Истина} \\\
Ложь & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	
Ложь	Истина	Ложь

Первый обязательный параметр соответствует числу объединённых колонок, второй — формату получившейся объединённой колонки, третий — тексту.

Пример использования `\multirow` из пакета **multirow**:

```
\centering
\begin{tabular}{|*{3}{p{1cm}}|}
\textbf{ИЛИ} & Истина & Ложь \\\[2mm]
\hline\hline
Истина & \multirow{2}{1cm}{Истина} & Истина \\\
Ложь & & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь		Ложь

Подробнее об использовании команд пакета написано в README к нему. В пакете определены две команды с двумя и тремя обязательными аргументами:

```
\multirow{«число строк»}{«ширина»}{«текст»}
\multirow{«число строк»}{*}{«текст»}
```

В качестве первого аргумента передаётся число строк, которые займёт объединённая клетка, далее можно выбрать либо автоматическое вычисление ширины, либо указать её самостоятельно. Число строк может быть отрицательным. В этом случае объединяются ячейки сверху от команды. Подобное может потребоваться, чтобы согласовать свою работу с пакетом **colortbl**:

```
\centering % при цветной печати будет цвет
\begin{tabular}{c>{\columncolor{yellow}}cc}
\backslashbox{два}{один} & Истина & Ложь \\\
\rowcolor{yellow}
Истина & & Истина \\\
Ложь & \multirow{-2}{*}{Истина} & Ложь
\end{tabular}
```

один два	Истина	Ложь
	Истина	Истина
	Ложь	Ложь

Пакет **colortbl** предназначен для раскрашивания таблицы в разные цвета. В документации **colortbl.pdf** подробно излагаются принципы работы пакета.

Команда `\backslashbox{текст}{текст}`, делящая клетку на две части наклонной чертой, определена в пакете **slashbox** (пакет необходимо скачать с CTAN и самостоятельно установить). Иногда так оформляют клетку на перекрестии боковика и головки. С пакетом идёт пример **slashbox.tex**.

Автор кириллических шрифтов **lh** Ольга Лапко для более изощрённой работы с таблицами предлагает пакет **makecell**. В этом пакете определена команда `\makecell`, которая «создаёт окружение одноколоночной таблицы с предопределёнными общими параметрами выключки, интерлиньяжа и вертикальных отбивок вокруг. Её удобно использовать для многострочных ячеек. Дополнительный аргумент команды позволяет изменить выключку таблицы». В пакете документация на английском языке (`makecell.pdf`) дублируется русской документацией (`makecell-rus.tex`).

11.3.4. Выравнивание чисел

Таблица часто строится вокруг чисел. Поэтому не удивительно, что внимание к выравниванию чисел для целей упрощения восприятия данных должно быть повышенным.

Пакет **dcolumn** из коллекции **tools** добавляет ещё одну спецификацию к формату столбцов таблицы:

<code>\centering</code>				
<code>\begin{tabular}{ c l D{.}{,}{5} }</code>				
<code>0.3141 & 0.3141 & 0.3141 \\</code>	<table><tr><td>0.3141</td><td>0.3141</td><td>0,3141</td></tr></table>	0.3141	0.3141	0,3141
0.3141	0.3141	0,3141		
<code>3.141 & 3.141 & 3.141 \\</code>	<table><tr><td>3.141</td><td>3.141</td><td>3,141</td></tr></table>	3.141	3.141	3,141
3.141	3.141	3,141		
<code>31.415 & 31.415 & 31.41 \\</code>	<table><tr><td>31.415</td><td>31.415</td><td>31,41</td></tr></table>	31.415	31.415	31,41
31.415	31.415	31,41		
<code>\end{tabular}</code>				

Новая спецификация имеет формат: `D{delim}{output}{nfrac}`, где `delim` — символ или набор символов, по которому происходит выравнивание (обычно это точка или запятая), `output` — символ, который замещает `delim` при компиляции (например, бывает нужно заменить точку на запятую), `nfrac` — максимальное число позиций в дробной части числа (при отрицательном значении число позиций не фиксируется). Подробности в документации `dcolumn.pdf`.

Пакет **numprint** (документация `numprint.pdf`) можно использовать просто для внятной печати чисел. Одноимённая команда `\numprint` позволяет легко переводить текстовую запись в математическую:

<code>\numprint[\text{м}/\text{с}]{299792458}\par</code>	299 792 458 м/с
<code>\numprint{2.718281828459045}\par</code>	2,718 281 828 459 045
<code>\numprint{6e23}\par</code>	$6 \cdot 10^{23}$
<code>\numprint{+-0.02}</code>	$\pm 0,02$

Следует обратить внимание на то, как формируется запись числа — каждые три следующие цифры целой или дробной части разделяются небольшим пробелом. В пакете **numprint** определён формат `n` с двумя параметрами для выравнивания чисел. Первый параметр указывает, сколько мест следует оставить для цифр целой части, а второй — для цифр дробной части.

Использование n-формата из пакета **numprint**:

```
\centering
\begin{tabular}{|n{3}{4}|n{2}{3}|n{1}{2}|}
0.3141 & 0.3141 & 0.3141 \\
3.141 & 3.141 & 3.141 \\
31.41 & 31.41 & 31.41 \\
314.1 & 314.1 & 314.1
\end{tabular}
```

0,314 1	0,314 1	0,314 1
3,141	3,141	3,141
31,41	31,41	31,41
314,1	314,1	314,1

Пакет **rccol** обладает схожей функциональностью, что и **dcolumn**, но дополнительно позволяет округлять значения. К сожалению, мне не удалось заставить его нормально работать, если в качестве разделителя используется точка. **rccol**, несмотря на информацию в документации **rccol.pdf**, признаёт только запятую. Возможно, это недоразумение.

Пакет **warpcol** предоставляет общую процедуру формирования формата числовых колонок. В документации к пакету **warpcol.pdf** представлены примеры того, как добиться различных эффектов при выравнивании.

11.3.5. Доступ к данным

CSV (от англ. Comma Separated Values — значения, разделённые запятыми) — это текстовый формат, предназначенный для представления табличных данных. Для доступа к этим данным напрямую можно воспользоваться пакетом **csvtools**. Документация к пакету **csvtools.pdf** достаточно подробна, но лучше осуществлять доступ к данным с помощью скрипта, результатом действия которого является сам **tex**-файл.

11.3.6. Клоны tabular

tabularx из коллекции **tools** — расширение **tabular** на предмет автоматического вычисления ширины колонок, имеющих форматный определитель **X**.

```
\centering
\begin{tabularx}{\textwidth}{|D{.}{,}{4}|X|c|}
0.3141 & очень длинная строка & 0.3141 \\
3.141\footnote{В окружение tabularx можно
добавлять подстрочное примечание}
& 3.141 & 3.141
\end{tabularx}
```

0,3141	очень длин- ная строка	0.3141
3,141 ^a	3.141	3.141

^aВ окружение **tabularx** можно добавлять подстрочное примечание

В качестве первого аргумента окружения **tabularx** передаётся ширина таблицы. При компиляции **X** преобразуется в **r{ширина}**, где вместо ширины выставляет-ся автомататически вычисленная величина так, чтобы в целом ширина таблицы

оставалась неизменной. Если в таблице присутствует более одного столбца, имеющих формат X, то для формирования таблицы может потребоваться несколько проходов. Все подробности, как обычно, можно узнать в документации к пакету (`tabularx.pdf`).

Пакет **tabulary** также является модификацией стиля **tabular**, точнее **array**. Пакет был создан для автоматического расчёта ширины колонок таблицы с целью минимизации высоты таблицы. Подробности можно найти в документации к пакету `tabulary.pdf`. Пользоваться одноимённым окружением следует с некоторой долей осторожности.

ctable — альтернативный взгляд на оформление таблицы. Для описания структуры таблицы вместо окружения используется команда `\ctable`, которая объединяет в себе функциональность **tabular** и **table**. В дополнение к этим особенностям в `\ctable` можно вставлять команды для создания подстрочных примечаний для таблицы. Примеры и документацию к пакету можно найти в файле `ctable.pdf`.

11.4. Многополосные таблицы

Окружение **tabular** и его производные всем хороши за исключением того, что они не могут занимать больше одной страницы.

Для вёрстки таблиц на несколько страниц обычно используют одно из двух окружений: `supertabular` и `longtable`, определённых в одноимённых стилях. Оба эти окружения обладают схожей функциональностью, но разными способами оформления. Оба предоставляют возможность создания стандартных шапки и окончания, появляющихся в начале, на каждой новой странице и в конце таблицы, соответственно. Про использование этих пакетов можно прочитать в документации: `supertabular.pdf` для `supertabular` и `longtable.pdf` для `longtable` из коллекции **tools**.

`longtable` в отличие от `supertabular` гарантирует неизменность ширины столбцов на протяжении всей таблицы, что достигается за несколько проходов $\text{\LaTeX}$. Неизменность ширины столбцов в зависимости от ситуации может рассматриваться как преимущество, так и как недостаток.

В пакете **supertabular** кроме окружения `supertabular` определено окружение `mpsupertabular`, которое каждый отдельный кусок таблицы на своей странице заключает в окружение `minipage`, что позволяет использовать подстрочные примечания прямо в таблице.

xtab

Более молодым и, возможно, более правильным является пакет **xtab**. По сути дела это расширение **supertabular** с некоторыми улучшениями и исправлениями. Также в пакете **xtab** определено окружение `mpxtabular`, которое отличается от `xtabular` тем, что каждая страница предварительно заключается в окружение

`minipage`, что позволяет создавать подстрочные примечания, которые выводятся в конце страницы.

Код для создания длинной таблицы с помощью **xtab** выглядит примерно следующим образом:

```
% Основной заголовок — печатается только в начале
\topcaption{Заголовок таблицы} \label{tab:xtab}

% Основная головка — печатается только в начале
\tablefirsthead{\hline Левая колонка & Правая колонка\\\hline}

% Вспомогательный заголовок и головка — печатается на
%каждой новой странице кроме последней
\tablehead{\multicolumn{2}{c}%
  {\tablename\ \thetable{}} "—— продолжение}\\
  \hline Левая колонка & Правая колонка\\ \hline}

% Вспомогательный заголовок и головка — печатается на
%последней странице
\tablelasthead{\multicolumn{2}{c}%
  {\tablename\ \thetable{}} "—— окончание}\\
  \hline Левая колонка & Правая колонка\\ \hline}

% Код завершения страницы при переносе на следующую
\tabletail{\hline \multicolumn{2}{|r|}%
  {Продолжение на следующей странице}\\ \hline}

% Заключительный код
\tablelasttail{\hline}

\begin{center}
\begin{xtabular}{|l|p{0.5\textwidth}}|}
xtabular & Это расширение обычного окружения tabular.
          Все команды, которые могут работать в окружении
          tabular, работают и в xtabular\\

          ...

\\
\end{xtabular}
\end{center}
```


11.5. Вывод

Вывод — таблица без линеек или организованный в колонки и строки материал, который не разделён линейками. Оглавление `\tableofcontents`, списки иллюстраций `\listoffigures` и таблиц `\listoftables` тоже являются таблицами.

```
\centering
\begin{tabular}{p{0.7\textwidth}@{}r}
Глава 1\dotfill & 2 \\
Глава 2\dotfill & 10 \\
Глава 3\dotfill & 124
\end{tabular}
```

Глава 1.....	2
Глава 2.....	10
Глава 3.....	124

Команда `\dotfill` формирует *отточия* (заполнение пространства точками).

listliketab

Окружение `listliketab` из одноимённого пакета специализируется на создании таблиц, оформленных как перечисление. Это окружение будет очень кстати при оформлении вопросников:

```
\storestyleof{enumerate}
\begin{listliketab}
\newcounter{tabenum}\setcounter{tabenum}{0}
\newcommand{\nextnum}{\addtocounter{tabenum}{1}\thetabenum.}
\begin{tabular}{L>\bf}l@{~~или~~}>\bf}l@{~~или~~}>\bf}l}
\nextnum & Красный & зелёный & голубой \\
\nextnum & Низкий & средний & высокий \\
\nextnum & Радостный & грустный & удивлённый \\
\end{tabular}
\end{listliketab}
```

Команда `\nextnum` используется для оформления нумерованных списков.

Результат выполнения кода:

1. **Красный** или **зелёный** или **голубой**
2. **Низкий** или **средний** или **высокий**
3. **Радостный** или **грустный** или **удивлённый**

Документация с примерами находится в файле `listliketab.pdf`

11.6. И это тоже таблицы?

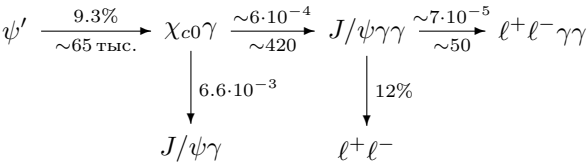
Пакеты **tableaux** и **tabvar** созданы для исследования области определения и поведения функции.

► Пригодится при изучении или написании методички по началам матанализа. Молодцы французы!

x	$-\infty$	$-\sqrt[3]{2}$	0	1	$+\infty$								
$f'(x)$	$-$	$-$	$\parallel$	$-$	0	$+$							
$f(x)$	$+\infty$	$\searrow$	0	$\searrow$	$+\infty$	$\searrow$	$-\infty$	$\searrow$	$+\infty$	$\searrow$	$\frac{3}{2}$	$\searrow$	$+\infty$

tabvar является более современной версией стиля **tableaux**. Поэтому подробности и примеры лучше всего искать в его документации **tabvar.pdf** и **demo.tex**.

Следующий пример оформляется вовсе не как таблица, но что-то «табличное» в нём есть. Пакет **pb-diagram** пригодится, когда нужно нарисовать простенькую диаграмму:



Глава 12

Справочно-поисковый аппарат издания

На этом же этаже располагалось книгохранилище. По поводу его размеров рассказывали, что в глубине, в полукилометре от входа, идёт вдоль стеллажей неплохое шоссе, оснащённое верстовыми столбами.

*«Понедельник начинается в субботу»
Аркадий и Борис Стругацкие*

Книги делятся на те, что читаются один раз, и те, что многократно перечитываются. Наличие информации о структуре книги повышает ценность любого текста. Отсутствие этой информации — прямой намёк, что после прочтения произведения следует забыть и выбросить.

Справочно-поисковый аппарат издания позволяет читателю облегчить и убыстрить поиск имеющихся в книге объектов. В качестве элементов, из которых складывается справочно-поисковая система, можно упомянуть рубрикацию, оглавление, колонтитулы, ссылки, подстрочные примечания, алфавитный указатель и библиографию.

Этот аппарат существует исключительно для читателя, и он достаточно трудоёмок при создании. Но сложности не должны пугать истинных энтузиастов в деле создания текстов, так как их преодоление значительно повышает ценность серьёзного произведения.

12.1. Оглавление

`\tableofcontents` (toc-файл) — стандартная команда ЛАТ_EX для создания оглавления. Кроме оглавления также можно вывести список иллюстраций `\listoffigures` (lof-файл) и таблиц `\listoftables` (lot-файл). На основе чего формируется само оглавление, подробно описано в разделе 10.5.

Для добавления какого-либо текста в оглавление можно воспользоваться командой `\addtocontents{toc}{текст}`. Вместо `toc` (оглавление) можно указать `lof` (список иллюстраций) или `lot` (список таблиц). Текст может содержать команды, но их лучше защитить с помощью макроса `\protect`, на случай если они окажутся хрупкими.

shorttoc

В больших документах полезно перед полным оглавлением иметь его краткую менее подробную версию. Пакет **shorttoc** добавляет команду:

```
\shorttableofcontents{Краткое оглавление}{1}
```

Первый параметр макроса — это заголовок оглавления, а второй параметр — глубина оглавления. В случае этой команды будет напечатано оглавление вплоть до разделов (`\section`), которые имеют уровень 1. Уровень обычного оглавления можно менять, переопределив счётчик `tocdepth`.

tocloft

По тем или иным причинам можно быть недовольным оформлением оглавления. Пакет **tocloft** даёт доступ ко всем «ручкам», которые можно покрутить, чтобы придать оглавлению достойный вид. Многочисленные подробности обретаются в файле `tocloft.pdf`. При чтении документации следует учитывать, что если в имени команды встречается заглавная **Z**, то это сокращение для `toc` or `lof` или `lot`, а если заглавная **X** — сокращение `part`, `chap`, `sec`, `subsec`, `subsubsec`, `fig`, `tab` и тому подобное. Например, так можно увеличить промежуток, который выделяется на печать номера раздела в оглавлении:

```
\addtolength{\cftsecnumwidth}{1ex}
```

а так, чтобы вместо точек расстояние между названием и номером страницы (лидеры) заполняли звёздочки:

```
\renewcommand{\cftdot}{\ensuremath{\ast}}
```

Таким образом можно добавить лидеры для глав:

```
\renewcommand{\cftchappleader}{\cftdotfill{\cftchapdotsep}}
\renewcommand{\cftchapdotsep}{\cftdotsep}
```

tocbibind

По умолчанию в оглавлении нет ссылок на библиографию и предметный указатель¹. Конечно несложно добавить эти элементы с помощью `\addcontentsline`, но, возможно, проще это сделать с помощью специально для этого написанного пакета **tocbibind**. При загрузке пакету **tocbibind** можно передать следующие опции:

notbib — не добавлять ссылку на библиографию;
notindex — не добавлять ссылку на указатель;
nottoc — не добавлять ссылку на оглавление;
notlot — не добавлять ссылку на список иллюстраций;
notlof — не добавлять ссылку на список таблиц;
chapter — если это возможно, ссылки оформлять так же, как главы;
section — если это возможно, ссылки оформлять так же, как разделы;
numbib — библиография тоже нумеруется, как и главы;
numindex — указатель тоже нумеруется, как и главы;
none — полностью отключить всё, что было добавлено при загрузке пакета.

Подробности можно найти в файле документации **tocbibind.pdf**.

minitoc

Если же хочется сделать из оглавление нечто изощрённое, то пакет **minitoc** с его более чем 6 сотнями страниц документации поможет на этом пути. Пакет позволяет сделать для каждой главы своё оглавление:

```
\dominitoc % формирование оглавления на уровне главы
\dominilof % формирование списка иллюстрации на уровне главы
\dominilot % формирование списка таблиц на уровне главы
\tableofcontents % обычное оглавление
\listoffigures % обычный список иллюстраций
\listoftables % обычный список таблиц

\chapter{Новая глава}
\minitoc % оглавление для этой главы
\mtcskip
\minilof % список иллюстраций для этой главы
\mtcskip
\minilot % список таблиц для этой главы
```

Подробности изложены в файле документации **minitoc.pdf**.

¹В классах **KOMA-Script** это решается с помощью дополнительных опций **bibtotoc** и **idxtotoc** для библиографии и указателя соответственно.

12.2. Колонтитулы

Правила формирования колонтитулов целиком зависят от выбранного класса документа. Если же хочется изменить значения по умолчанию, то проще всего выбрать стиль страницы `myheadings` и сформировать колонтитулы:

```
\pagestyle{myheadings}  
\markboth{«левый колонтитул»}{«правый колонтитул»}
```

Если печать односторонняя, то достаточно воспользоваться командой `\markright`, которая имеет только один аргумент. Для полного управления содержимым колонтитулов лучше всего подходит пакет **fancyhdr**.

fancyhdr

Чтобы воспользоваться возможностями пакета, необходимо загрузить стилевой файл и с помощью `\pagestyle` выбрать стиль.

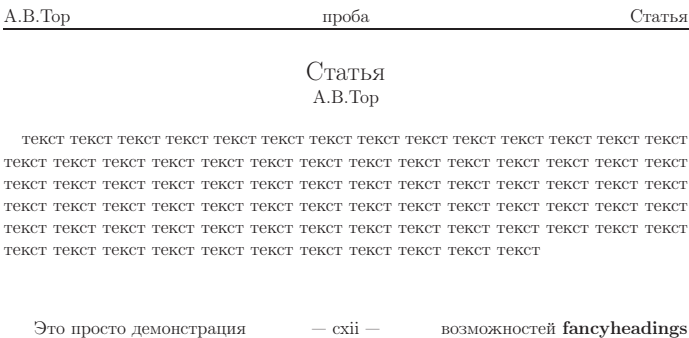


Рис. 12.1. Заполняем колонтитулы с помощью **fancyhdr**

Пакет **fancyhdr** позволяет управлять содержимым колонтитула. Возможности: верхний и нижний колонтитул разбивается на три независимых части, многострочные колонтитулы, колонтитулы, вылезавшие по ширине за `\textwidth`, декоративные линейки, разные колонтитулы для чётных и нечётных страниц, отдельные колонтитулы для специальных полос (начала глав, страницы, отведённые под плавающие объекты). Простейший пример для определения колонтитулов с помощью **fancyhdr**:

```
\usepackage{fancyhdr}  
...  
\begin{document}  
\pagestyle{empty} %очищаем стиль страницы  
\pagestyle{fancy} %включаем пользовательский стиль
```

```
\lhead{A.B.Top}    %верхний колонтитул слева
\chead{---~\arabic{page}~---} % там же по центру
\rhead{Статья}     %верхний колонтитул справа
% аналогично для оформления нижнего колонтитула
\cfoot{---~\roman{page}~---}
\lfoot{\hspace{0.7cm}Это просто демонстрация}
\rfoot{возможностей \textbf{fancyheadings}}
```

Результат компиляции примера представлен на рис. 12.1. Переменная `page` содержит номер страницы.

Для того чтобы задать колонтитулы на чётных/нечётных страницах для двухсторонней печати, необходимо воспользоваться необязательным параметром команд, например:

```
\rhead[чётный колонтитул]{нечётный колонтитул}
```

Если необязательный параметр отсутствует, то для чётных страниц используется тот же колонтитул, что и для нечётных.

Подробности о том, что можно сотворить с колонтитулами, следует искать в документации к пакету **fancyhdr.pdf**.

nccfancyhdr

nccfancyhdr из пакета **ncctools** является более современной и качественной реализацией **fancyhdr**. При загрузке стиля **nccfancyhdr** содержимое колонтитулов можно задавать теми же командами, что и в **fancyhdr**. Подробная и исчерпывающая документация в файле **nccfancyhdr.pdf** позволит узнать, как создавать свои стили страницы, как задавать стили страницы, где размещаются иллюстрации, как создать разделительную линию между текстовым полем и колонтитулом, а также многое другое.

technics

Одним из примеров использования пакета **fancyhdr** является стиль **technics**. Он переопределяет колонтитулы, чтобы получить стандартную рамку для стандартных корпоративных документов. Пример такой рамки можно увидеть на рисунке 12.2.

Без подготовки пользоваться этим стилем нельзя. Чтобы его можно было использовать с кириллицей, необходимо как минимум закомментировать строку с **inputenc**. Этот стиль следует рассматривать исключительно как пример работы с **fancyhdr**, благо файл **technics.sty** относительно небольшой.

 Для внутреннего использования Пингвины и К	
Подготовлено Е. М. Балдин E.M.Baldin@inp.nsk.su	Название Пример использования technics
Завизировано	Дата 30 августа 2007 г.
© 2007 Пингвины и К 1	

Рис. 12.2. Рамочка от **technics**

12.3. Библиография

Хорошая книга представляет собой ценность. Но даже лучшая книга не в состоянии охватить абсолютно все аспекты рассматриваемой в ней темы. Книги существуют в книжном пространстве. Всегда можно найти что-то, на чём книга основывалась, что-то, что развивает основную идею, и что-то, что позволяет взглянуть на главную тему с другой стороны. Список литературы книгу только украшает.

Для ссылок на литературу используется команда

```
\cite [«комментарий»] {«список меток»}
```

Метки формируются с помощью механизма ВивТ_ЕX, рассмотренного в разделе 12.4, либо добавляются внутри окружения thebibliography:

Полезно почитать книгу \cite [Роженко] {rozenko – 2005}.

```
\begin{thebibliography}{9}
\bibitem{rozenko – 2005}Роженко А.И. Искусство верстки в
\LaTeX'e. \newblock — Новосибирск: Изд. ИВМиМГ СО~РАН,
2005. 398~с.
\end{thebibliography}
```

Команда \newblock позволяет логически разделить разные по смыслу элементы. В качестве обязательного аргумента окружения thebibliography требуется передать текст, соответствующий самой широкой метки для выравнивания. То есть, если список литературы содержит меньше 10 записей, достаточно передать однубуквенную фразу, например, «9», а в случае двузначного числа книг в списке, уже потребуется «99» и так далее.

Полезно почитать книгу [1, Роженко].

Список литературы

- [1] Роженко А.И. Искусство вёрстки в $\text{\LaTeX}$ 'е. — Новосибирск: Изд. ИВМиМГ СО РАН, 2005. 398 с.

Рис. 12.3. Пример библиографической ссылки

По умолчанию в тексте не может быть более одного окружения `thebibliography`. В случае необходимости эту проблему можно обойти, загрузив пакет **compactbib**. Нумерация элементов при этом остаётся сквозной. Вертикальные пробелы между элементами несколько уменьшаются по сравнению с умолчанием.

Как и в случае с перекрёстными ссылками, информация о списке литературы заносится в файл с расширением **aux**. То есть для правильного его отображения необходимо два прохода $\text{\LaTeX}$.

В одной команде `\cite` можно перечислить несколько меток через запятую, например, так выглядит ссылка на книги по $\text{\LaTeX}$ русских авторов [8, 9, 10]:

```
\cite{Lvovskij-2003,Kotelnikov-2004,Rozhenko-2005}
```

Временами при большом количестве цитируемой литературы эти ссылки могут разрастаться. Для того чтобы слишком большие списки могли автоматически переноситься на следующую строку, можно воспользоваться стилем **breakcites** из пакета **ltxmisc**.

В дополнение к команде цитирования `\cite` в случае подключения библиографической базы можно использовать инструкцию `\nocite`. Команда `\nocite` не создаёт никакой ссылки в тексте, но упомянутая запись отображается в списке литературы.

► Для того чтобы можно было использовать кириллицу в метках для цитирования литературы, а именно иметь возможность написать что-то вроде:

```
\cite{Котельников-2004}
...
\bibitem{Котельников-2004}
```

следует воспользоваться пакетом **citehack**:

```
\usepackage{citehack}
```

Из названия пакета очевидно, что это «хак» со всеми вытекающими последствиями, то есть не следует им особо злоупотреблять.

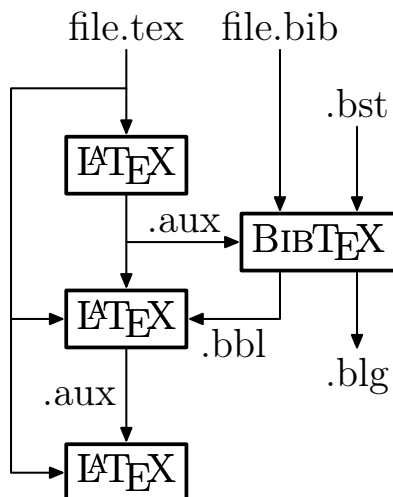


Рис. 12.4. Конвейер $\text{\LaTeX}$ + $\text{\BibTeX}$: `tex` — $\text{\LaTeX}$ -исходник, `bib` — библиографическая база, `bst` — стилевой файл для библиографии, `blg` — `log`-файл $\text{\BibTeX}$, `bbl` — отсортированный список литературы, `aux` — информация о ссылках

12.4. Работаем с $\text{\BibTeX}$

Можно список литературы оформлять руками. Есть какой-никакой стандарт, например, тот же ГОСТ 7.80-00² или ГОСТ 7.1-84³. Берём и просто следуем подробной инструкции. Но далеко не все издательства подчиняются этому стандарту, в котором, например, нет информации, как нужно оформлять `www`-ссылки. Да и вообще список сопутствующей литературы — это нечто большее, чем просто довесок к статье или книге, это вполне самостоятельный фрагмент информации, который очень полезно уметь представлять по-разному.

Для решения этой проблемы Орен Поташник разработал программу $\text{\BibTeX}$, которая сама формирует окружение `thebibliography`, получая информацию из текстовой библиографической базы. Структура библиографической базы $\text{\BibTeX}$ является довольно распространённым форматом, который использует в том числе и Google Scholar (<http://scholar.google.com/>), не говоря уж о том, что основной архив доступных электронных препринтов <http://arxiv.org> предоставляет библиографическую информацию исключительно в виде записей $\text{\BibTeX}$.

► Из программного обеспечения, позволяющего работать с $\text{\BibTeX}$, следует упомянуть встроенный в **Emacs** пакет **RefTeX** и **JabRef**.

²ГОСТ на оформление библиографического указателя, принятый в 2000 году. Правила оформления могут нравиться или не нравиться, но это всё-таки хоть какой-то стандарт.

³Устарел после выхода ГОСТ 7.1-2003. Необходима доработка этого стиля.

Список литературы

- [1] М. Гуссенс, Ф. Миттельбах, А. Самарин. Путеводитель по пакету Л^AT_EX и его расширению Л^AT_EX 2_ε: Пер. с англ. — М. : Мир, 1999. — 606 с. — ISBN: 5-03-003325-4.
- [2] С. М. Львовский. Набор и вёрстка в системе Л^AT_EX. — М. : МЦНМО, 2003. — 448 с. — ISBN: 5-94057-091-7. — 3-е изд., испр. и доп.
- [3] И. А. Котельников, П. З. Чеботаев. Л^AT_EX по-русски. — Новосибирск : Сибирский Хронограф, 2004. — 496 с. — ISBN: 5-87550-195-2. — 3-е изд., перераб. и доп.
- [4] А. И. Роженко. Искусство вёрстки в Л^AT_EX'е. / Под ред. А. С. Алексеева. — Новосибирск : Изд. ИВМиМГ СО РАН, 2005. — 398 с. — ISBN: 5-901548-25-6.

Рис. 12.5. Список литературы, оформленный с помощью стиля **gost2008**

► Тот, кто не освоил **Emacs** и кому не нравится Java, может поискать программные пакеты **gBib** и **KBib** для Gnome и KDE соответственно. Простой конвертер **bibtex2html** позволяет получить список литературы и html-виде. Естественно, и простое редактирование текстового файла руками также никто не отменял.

Как правило, библиографическая база в формате ВивТ_EX хранится в файле с расширением **bib**. Перед тем как с помощью команды `\bibliography` подключить базу к Л^AT_EX-исходнику, нужно выбрать стиль сортировки библиографии:

```
\bibliographystyle{«стиль»}
\bibliography{«имя bib-файла без расширения»}
```

Теперь можно скомпилировать сам текст:

```
> latex mybook.tex
> bibtex mybook
> latex mybook.tex
```

Информация о том, какой именно **bib**-файл используется, сохраняется в aux-файле при первой компиляции с помощью **latex**.

В Л^AT_EX есть четыре стандартных стиля для оформления списка литературы. **plain** — открытый стиль. Библиографические записи помечаются порядковыми номерами и сортируются в алфавитном порядке. Чтобы правильно отсортировать библиографию на русском языке, необходимо вместо **bibtex**

воспользоваться командой **bibtex8**, указав с помощью ключа `--csfile` соответствующее правило сортировки⁴.

unsrt — несортирующий стиль. В отличие от **plain** порядок представления списка литературы определяется порядком цитирования библиографии в тексте.

alpha — алфавитный стиль. Вместо нумерации библиографии используются имена меток. Литература сортируется по меткам.

abbrv — аббревиатурный стиль. Вместо полных имён авторов, названий месяцев и журналов печатаются сокращения. Сортировка и нумерация соответствует стилю **plain**.

Игорь А. Котельников разработал стили для В_IT_EX, соответствующие ГОСТ 7.0.5-2008 (Библиографическая ссылка. Общие требования и правила составления.), ГОСТ 7.1-2003 (Библиографическая запись. Библиографическое описание. Общие требования и правила составления.), ГОСТ 7.80-2000 (Библиографическая запись. Заголовок. Общие требования и правила составления.), ГОСТ 7.11-2004 (Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках) и ГОСТ 7.83-2001 (Электронные издания. Основные виды и выходные сведения.): **gost2003s/gost2008s**, аналогичные **plain**, и **gost2003/gost2008**, аналогичные **unsrt**. Информацию по этим библиографическим стилям можно найти в директории `texmf-dist/doc/bibtex/gost` в файлах `gost.pdf`, `gost2003.pdf` и `gost2008.pdf`.

Многие журналы, принимающие публикации в L^AT_EX, имеют свои собственные В_IT_EX-стили. В стандартной поставке T_EX Live 2012 идёт около 350 различных библиографических стилей.

Сама по себе база состоит из записей вида:

```
@book{Gussens-1999,
  author = {М. Гуссенс and Ф. Миттельбах and А. Самарин},
  title = {Путеводитель по пакету \LaTeX{} и его
                                     расширению \LaTeXe: Пер. с англ.},
  year = {1999},
  isbn = {5-03-003325-4},
  publisher = {Мир},
  address = {М.},
  numpages = {606},
  language = {russian},
  OPTnote = {}
}
```

⁴В стандартной поставке L^AT_EX есть правило сортировки для кодовой страницы `sr866` `sr866rus.csf`. На основе этого файла можно создать правило для другой кодовой страницы.

После знака «коммерческое at» @ идёт тип записи. В фигурных скобках вслед за меткой через запятую перечисляются пары ключ-значение. ВивТ_ЕX поддерживает определённый набор типов записей, каждому из которых соответствуют свои обязательные и необязательные поля⁵. Если не заполнено обязательное поле, то при компиляции ВивТ_ЕX генерирует ошибку. Имеются стандартные типы записей (ВивТ_ЕX не чувствителен к регистру).

- **Aritcle** — статья в журнале. Обязательные поля: author, title, journal, year. Необязательные поля: volume, number, pages, month, note, annote.
- **Book** — книга. Обязательные поля: author или editor, title, publisher, year. Необязательные поля: volume, number, series, address, edition, month, note, annote.
- **Booklet** — брошюра. Обязательное поле: title. Необязательные поля: author, howpublished, address, month, year, note, annote.
- **Conference** или **InProceedings** — статья, опубликованная в трудах конференции. Обязательные поля: author, title. Необязательные поля: crossref, booktitle, pages, year, editor, volume, number, series, address, month, organisation, publisher, note, annote.
- **Proceedings** — труды конференции. Обязательные поля: title, year. Необязательные поля: booktitle, editor, volume, number, series, address, month, organisation, publisher, note, annote.
- **InBook** — ссылка на часть книги, то есть на её главу, раздел или просто на определённый набор страниц. Обязательные поля: author или editor, title, chapter, publisher year. Необязательные поля: volume или number, series, type, address, edition, month, pages, note, annote.
- **InCollection** — часть книги со своим заглавием. Обязательные поля: author, title, booktitle. Необязательные поля: crossref, pages, publisher, year, editor, volume или number, series, type, chapter, address, edition, month, note, annote.
- **Manual** — техническая документация. Обязательное поле: title. Необязательные поля: author, organistaion, address, edition, month, year, note, annote.
- **PhdThesis** — диссертация. Обязательные поля: author, title, school, year. Необязательные поля: address, month, note, annote.
- **MastersThesis** — дипломная работа. Обязательные и необязательные поля копируют **PhdThesis**.

⁵Если к названию необязательного поля добавить ОПТ (note→OPTnote), то такие поля игнорируются, даже если присутствуют в записи.

- **TechReport** — технический отчёт. Обязательные поля: author, title, institution, year. Необязательные поля: type, numer, address, month, note, annotate.
- **Unpublished** — неопубликованный авторский текст. Обязательные поля: author, title, note. Необязательные поля: month, year, annotate.
- **Misc** — то, что не подходит для других типов записей. Обязательные поля отсутствуют. Необязательные поля: author, title, howpublished, month, year, note, annotate.

Значение полей в большинстве случаев понятно из их названия. Исключением, пожалуй, является поле `crossref`, в качестве значения которого можно передать ссылку на другую запись. В этом случае `BIBTeX` при трансляции возьмёт там значения всех недостающих полей записи. Из особенностей следует упомянуть, что авторы в поле `author` разделяются с помощью союза `and`.

Кроме перечисленных стандартных полей при использовании библиографических ГОСТ-стилей `gost2003/gost2008` и `gost2003s/gost2008s` можно использовать поля `numpages` — число страниц и `language` — влияет на сортировку при выборе стиля типа **plain**. Допустимые значения для поля `language` — `russian`, `ukrainian` и `english` по умолчанию.

12.5. Предметный указатель

Для создания «полуфабриката» алфавитного указателя в преамбуле помещается инструкция `\makeindex`. Слова, которые нужно поместить в указатель, отмечаются с помощью команды `\index`:

Указатель `\index{Предметный указатель}` ...

Сама по себе команда `\index` игнорируется, но всё, что в ней отмечается вместе с информацией о положении команды, заносится в файл с расширением `idx`. Имя `idx`-файла по умолчанию соответствует имени основного документа. Полуфабрикатом `idx`-файл является потому, что записи хранятся в нём в неотсортированном виде. С помощью программ сортировки **rumakeindex** получается уже отформатированный правильным образом указатель в файле с расширением `ind`. Этот файл уже можно вставить в документ:

`\input{ind-файл}`

Это же делает и команда `\printindex` из пакета **makeidx**. В дополнение к подключению индекса команда `\printindex` проверяет существование индексного файла и не даёт `LATEX` генерировать ошибку в случае его отсутствия. Пакет **makeidx** содержит ещё несколько полезных команд для создания индекса, поэтому его в любом случае имеет смысл загрузить.

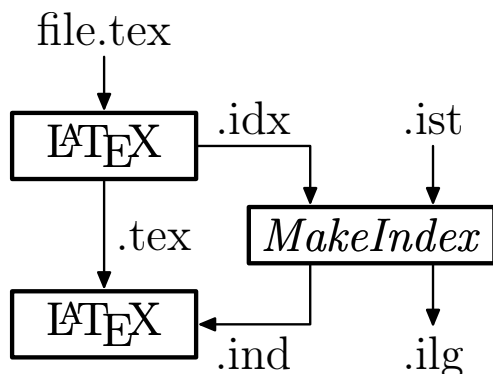


Рис. 12.6. Конвейер $\text{\LaTeX} + \text{MakeIndex}$: `tex` — $\text{\LaTeX}$ -исходник, `idx` — не отсортированный индекс (полуфабрикат), `ist` — стилевой файл для указателя, `ilg` — log-файл *MakeIndex*, `ind` — отсортированный указатель

Программа **rumakeindex** является простейшим скриптом, где с помощью **sed** кириллические буквы из внутреннего представления $\text{\LaTeX}$ переводятся в `ko18-g` и правильным образом сортируются с использованием стандартного механизма **makeindex**. Если необходимо отсортировать индексный файл для включения в текст, использующий другую кодовую страницу, или просто не устраивают правила сортировки, то этот скрипт легко переделывается.

Оригинальная программа сортировки индекса **makeindex** (подробная документация представлена в файле `makeindex.dvi`) была написана довольно давно и, естественно, не учитывала национальных особенностей других языков кроме английского и немецкого. К сожалению, она оказалась достаточно гибка, чтобы полностью отказаться от неё в пользу другого механизма сортировки уже нормально поддерживающего интернационализацию. Наиболее вероятным претендентом на замену уже долгое время является **xindy** (см. раздел 12.6).

Аргумент команды `\index` может содержать любые символы кроме `!`, `"`, `@` и `|`. Их специальное значение проявляется только внутри команды. Чтобы убрать специальное значение этих символов внутри `\index`, необходимо перед ними добавить символ `"` (двойную кавычку). На рис. 12.7 представлен пример простейшего указателя. Далее будут раскрыты методы его создания.

Для формирования многоуровневых иерархических указателей используется разделитель в виде восклицательного знака `«!»:`

```
% на страницах 110 и 111
\index{Указатель!rumakeindex}
```

Команды `\index` с одинаковыми аргументами группируются в одну запись с полным списком страниц.

Предметный указатель

MakeIndex, 110

Указатель, 110–111

makeindex, см. rumakeindex

rumakeindex, 110, 111

xindy, 111

Рис. 12.7. Пример готового предметного указателя

Символ вертикальной черты «|» используется для отделения видимого аргумента от управляющих знаков. Для создания перекрёстных ссылок внутри предметного указателя используется команда **see**:

```
\index{Указатель!makeindex|see{rumakeindex}}
```

makeindex преобразует **see** в команду `\see`, определённую в пакете **makeidx**. Результат действия этой команды можно посмотреть на рис. 12.7 рядом со словом **makeindex**.

Также с помощью вертикальной черты можно сформировать указатель на диапазон страниц:

```
% на странице 110
\index{Указатель|({
  много текста
% на странице 111
\index{Указатель|)}}

```

Иногда номер страницы нужно как-то выделить. Команды выделения текста также должны идти после вертикальной черты:

```
% Выделяем страницу 110 жирным шрифтом
\index{MakeIndex@\textit{MakeIndex}textbf}

```

Конструкция «ключ»@«запись» используется для правильной сортировки внутри `\index`. По «ключу» производится сортировка, а «запись» выводится в предметном указателе.

splitidx

Стандартные средства создания предметных указателей в ЛАТЭХ несколько ограничены. В частности предполагается, что используется ровно один предметный указатель. Это решается с помощью пакета **splitidx**. Пусть в `mybook.tex` необходимо включить обычный предметный указатель и указатель персоналий:

```
% Преамбула
\usepackage[split]{splitidx}
% Команда на создание индексов
\makeindex
% Определение обычного указателя (метка idx)
\newindex[Алфавитный указатель]{idx}
% Определения указателя персоналий (метка aut)
\newindex[Персоналии]{aut}

...

% Тело документа

% Это слово записывается в основной индексный файл
\index{физика}

% Это слово записывается в файл персоналий
\sindex{aut}{Ломоносов}

% Распечатывается указатель персоналий
\printindex{aut}

% Распечатывается основной указатель
\printindex{idx}
```

В пакете определена команда `\newindex`, которой в качестве необязательного параметра передаётся заголовок указателя, а обязательный параметр является меткой. Пользуясь этой меткой, команда `\sindex` направляет слово в конкретный список. Список со стандартным именем `mybook.idx` является основным. К именам файлов вспомогательных списков добавляются их метки. В случае метки `aut` имя списка имеет вид `mybook-aut.idx`. То есть для создания индекса необходимо выполнить примерно такую последовательность команд:

```
> latex mybook.tex
> rumakeindex mybook.idx
> rumakeindex mybook-aut.idx
> latex mybook.tex
```

Если необходимо создать очень большое количество предметных указателей, то можно превысить лимит `TeX` на число записываемых файлов, которых не может быть больше 16. В этом случае необходимо убрать опцию `split` при загрузке пакета и воспользоваться программой разделения индексного файла `splitindex.pl`, которая идёт вместе с пакетом. Подробности об использовании следует искать в документации пакета `splitidx.pdf`.

12.6. xindy

xindy (<http://www.xindy.org/>) — современная альтернатива **makeindex**. При использовании **xindy** не возникает необходимости в «костылях» наподобие скрипта **rumakeindex**. **xindy** из коробки поддерживает десятки языков вплоть до клингонского. **xindy** во всём превосходит **makeindex**. Кроме поддержки множества языков при создании индексов **xindy** автоматически проверяет корректность перекрёстных ссылок (чуть подробнее о перекрёстных ссылках в предметном указателе с помощью инструкции **see** на стр. 176) и имеет модульную структуру, позволяющую относительно легко писать расширения и стили.

Единственное препятствие на пути распространения **xindy** — это фактически полное отсутствие внятной документации для пользователя. С документацией по созданию своих стилей всё нормально, но это немного другой уровень, нежели требуется автору. Также совершенно непонятна причина, по которой **xindy** отсутствует⁶ в основных дистрибутивах `LaTeX`. Поэтому с большой долей вероятности программу придётся скачать с официального сайта. Со временем эти проблемы, очевидно, разрешатся.

► Правила сортировки для русского языка, поставляемые с **xindy**, не разделяют буквы «е» и «ё». Это можно поправить при сборке пакета из исходников.

Для создания указателя с помощью **xindy** проще всего воспользоваться перл-скриптом **texindy**:

```
> texindy -L russian -C koi8-r «файл».idx
```

Полное имя `idx`-файла набирать обязательно. Опция `-L` позволяет передать язык (например, `belarusian` или `ukrainian`), а опция `-C` — кодовую страницу документа (например, `utf-8`, `cp1251` или `koi8-u`).

texindy для формирования указателя вызывает **xindy**, загружая необходимые модули. Сами по себе загружаемые модули представляют текстовые файлы с расширением `xdy`. Правила создания модулей подробно описаны в документации, идущей с **xindy**. При написании своего модуля полезно иметь общее представление о языке `Lisp`.

Дополнительные модули можно загрузить с помощью опции `-M`, которую можно вызывать многократно. Интерес могут представлять следующие стандартные

⁶**xindy** присутствует в некоторых современных дистрибутивах `GNU/Linux`, например, в `Debian (Lenny)`.

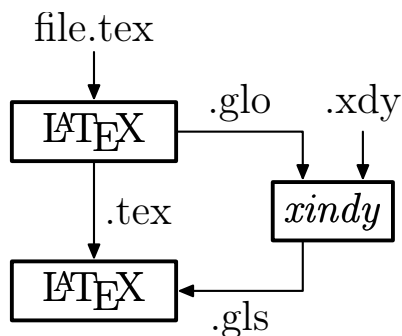


Рис. 12.8. Конвейер $\text{\LaTeX}+xindy$ при создании глоссария: `tex` — $\text{\LaTeX}$ -исходник, `glo` — не отсортированный глоссарий (полуфабрикат), `xdy` — стилевой файл для глоссария, `gls` — отсортированный глоссарий

модули (предполагается, что для создания указателя используется программа `texindy`):

- Сортировка:

word-order — сортировка по словам. Пробельный символ идёт раньше любого знака. Загружается по умолчанию, выключается с помощью опции `-1`;

letter-order — пробельные символы игнорируются. Включается с помощью опции `-1`;

numeric-sort — числа сортируются как числа, то есть «Агент7» появится раньше чем «Агент13». Этот модуль загружается автоматически;

keep-blanks — при создании указателя перестают игнорироваться ведущие и замыкающие пробельные символы;

ignore-hyphen — при сортировке игнорируются тире;

ignore-punctuation — при сортировке игнорируется все символы пунктуации.

- Нумерация страниц:

page-ranges — интервалы страниц. Если элемент указателя присутствует на нескольких страницах подряд (например, на 35, 36, 37 и 38 страницах), то при ссылке формируется интервал страниц (35–38). Загружается по умолчанию и может быть отключён с помощью опции `-r`;

book-order — книжная сортировка страниц. Правильная сортировка страниц для книг: сначала идут римские цифры, затем арабские, а затем всё остальное.

Глоссарий

Глоссарий собрание глосс, предшественник словаря

Указатель упорядоченный перечень объектов текста, благодаря чему можно быстро находить сведения об этих объектах, когда требуется справка или выборочное чтение издания

xindy программа для формирования вспомогательного указателя, может использоваться и для создания глоссария; *см.* Указатель; *см. также* Глоссарий

Рис. 12.9. Пример глоссария

12.7. Глоссарий

Глоссарий (рис. 12.9) — это список терминов вместе с их объяснениями. Это краткий словарь, который позволит читателю быстро разобраться с употребляемой в тексте терминологией. Сам глоссарий, как и указатели, создаётся внешней программой **makeidx** или **xindy**.

makegloss

Для создания «полуфабриката» глоссария по аналогии с созданием указателей существует команда `\makegloss`, которая помещается в преамбулу, элементы глоссария формируются с помощью команды `\glossary`, печать же производится с помощью команды `\makeglossary`. На рис. 12.8 изображена схема взаимодействия **Л^AT_EX** и **xindy** при создании глоссария.

Для подготовки заготовки для глоссария и, собственно, формирования глоссария лучше воспользоваться пакетом **makeglos**. Подробности можно поискать в файле документации `makeglos.pdf`, а примеры создания глоссария можно посмотреть в его исходнике `makeglos.tex`. В этом пакете определяются дополнительные команды, необходимые для формирования глоссария.

```
%Преамбула
\usepackage{makeglos}
% Команда создания заготовки для глоссария glo-файла
\makeglossary
...
% Тело документа
% Определение заголовка глоссария
\renewcommand{\glossaryname}{Глоссарий}
% Добавление ссылки на глоссарий в оглавление
\addcontentsline{toc}{section}{\glossaryname}
```

```
% Печать готового глоссария   gls-файла
\printglossary{}
% Создание заготовки для глоссария
\glossary{Указатель:упорядоченный перечень объектов текста,
  благодаря чему можно быстро находить сведения об этих
  объектах, когда требуется справка или выборочное чтение
  издания}
\glossary{Глоссарий:собрание глосс, предшественник словаря}
\glossary{xindy|is{программа для формирования вспомогательного
  указателя, может использоваться и для создания глоссария}}
\glossary{xindy|see{Указатель}}
\glossary{xindy|seealso{Глоссарий}}
```

Словарное слово отделяется двоеточием от своего определения. Также можно слово и определение разделить вертикальной чертой, но в этом случае необходимо воспользоваться лексемой `is`. Каждое слово можно определять с помощью `\glossary` несколько раз.

Команда `\glossaryname` хранит заголовок глоссария. Переопределив аналогичным образом команду `\glossaryintro`, перед началом глоссария можно добавить введение.

Для создания перекрёстных ссылок можно использовать лексемы `see` и `seealso`. При компиляции `glo`-файла они заменяются на макросы `\gsee` и `\galso`:

```
> texindy «файл».glo -o «файл».gls -M makeglos \
  -L russian -C koi8-r
```

Чтобы сформировать глоссарий, **texindy** необходимо загрузить с помощью опции `-M` специальный стилевой файл **makeglos.xdy**. Он лежит в той же директории, где и документация к пакету **makeglos**. Файл проще всего просто скопировать в рабочую директорию.

gloss

Пакет **gloss** позволяет создать глоссарий с использованием базы ВивТ_ЭX. Для этого отдельно от основного текста необходимо записать словарики в `bib`-файл. Каждый элемент словарика будет иметь примерно следующий вид:

```
@gd{gnu,
word = {ГНУ},
definition = {Дикое животное}
}
```

В данном примере **gnu** — это метка, на которую можно сослаться в тексте с помощью команды `\gloss`. Ключ **word** соответствует словарной статье. Это имя появляется в тексте при ссылке на неё. Если ранее был загружен пакет **hyperref**,

то формируется гиперссылка на соответствующую статью в глоссарии. Определение формируется с помощью ключа **definition**.

Сам текст должен содержать следующие команды:

```
% Преамбула
\usepackage{gloss}
% Команда создания списка в gls.aux-файле
\makegloss
...
% Тело документа
% Ссылка на словарную статью ГНУ
\gloss{gnu}
% Определение заголовка глоссария
\renewcommand{\glossname}{Глоссарий}
% Печать глоссария
\printgloss[|<<имя bib-файла|]
```

Порядок действий при создании глоссария примерно следующий:

- компиляция `tex`-файла,
- формирование глоссария с помощью `ViTeX` на основании записей в `gls.aux`-файле, иными словами следует выполнить команду:

```
> bibtex8 имя bib-файла.gls
```

- компиляция `tex`-файла для вставки глоссария,
- компиляция `tex`-файла для правильного вывода перекрёстных ссылок.

Пакет позволяет поддерживать несколько глоссариев. Об этом и о многом другом можно узнать в файле документации `gloss.dvi`

nomenc1

Пакет **nomenc1** позволяет создать список используемых в тексте обозначений. Этот список по своей структуре похож на обычный глоссарий, только в качестве описываемых терминов здесь предполагаются спецсимволы, хотя это и не обязательно.

```
% Преамбула
\usepackage[russian]{nomenc1}
% Команда создания списка в nlo-файле
\makenomenclature
...
% Тело документа
```

```
\nomenclature [W] {\ Warning } {Предупреждение}  
\nomenclature [E] {\ Error } {Ошибка}  
  
% Печать списка символов  
\printnomenclature
```

Команда `\nomenclature` добавляет элементы в список. Кроме обязательных двух параметров символ и описание, команде в качестве необязательного параметра можно передать префикс, который будет использоваться для сортировки списка. В показанном ранее примере при печати «Ошибка» будет находиться раньше «Предупреждения».

Для создания списка программе **makeindex** необходимо указать стилевой файл `nomenc1.ist`:

```
> makeindex «имя».nlo -s nomenc1.ist -o «имя».nls
```

Примеры и качественное описание пакета представлены в файле `nomenc1.pdf`.

Глава 13

Математика

Математика — это королева наук.

Карл Фридрих Гаусс

Все точные науки говорят на языке математики. Много из входящего в этот язык было реализовано в пакетах $\mathcal{A}\mathcal{M}\mathcal{S}$, поэтому первое, что следует сделать, — это загрузить **amsmath**:

```
\usepackage{amsmath}
```

В дальнейшем по умолчанию считается, что этот стилевой файл загружен.

13.1. Математические символы

В этом разделе предполагается, что кроме **amsmath** также загружены пакеты **amsfonts** и **amssymb**.

13.1.1. Типы символов

$\text{\LaTeX}$ сам расставляет пробелы внутри формул. При этом он руководствуется классом символов. При желании класс символа можно поменять, передав его в качестве обязательного аргумента командам.

- Операторы:

$\backslash\mathbin$ — бинарные операторы. В этот класс входят знаки сложения (+), вычитания (−), умножения ($\times$ или $\cdot$);

$\backslashmathrel$ — операторы отношения. В этот класс входят знаки больше (>), меньше (<), равенства (=);

- `\mathop` — математические операторы. К этому классу относятся такие команды, как `\sin` ($\sin$), `\sum` (Σ), `\lim` ($\lim$).
- Символы:
 - `\mathord` — класс обычных символов. Включает в себя простые символы a (`(a)`), ζ (`\zeta`), $\rightarrow$ (`\to`).
 - Математическая пунктуация:
 - `\mathopen` — открывающие скобки;
 - `\mathclose` — закрывающие скобки;
 - `\mathinner` — внутренние элементы. В этот класс входит многоточие ($\dots$);
 - `\mathpunct` — символы пунктуации. Этот класс включает обычные пунктуационные знаки.

13.1.2. Список символов

Многие символы, которые существуют и используются в L^AT_EX, перечислены в титаническом труде «The Comprehensive L^AT_EX Symbol List». В стандартном дистрибутиве этот достойный восхищения талмуд доступен в виде исходников и pdf-файла `symbols-a4.pdf`.

Математические символы					
$\aleph$	<code>\aleph</code>	$\angle$	<code>\angle</code>	$\backslash$	<code>\backslash</code>
$\Box$	<code>\Box</code>	$\bot$	<code>\bot</code>	$\clubsuit$	<code>\clubsuit</code>
$\Diamond$	<code>\Diamond</code>	$\diamondsuit$	<code>\diamondsuit</code>	ℓ	<code>\ell</code>
$\emptyset$	<code>\emptyset</code>	$\exists$	<code>\exists</code>	$\flat$	<code>\flat</code>
$\forall$	<code>\forall</code>	$\hbar$	<code>\hbar</code>	$\heartsuit$	<code>\heartsuit</code>
$\Im$	<code>\Im</code>	$\imath$	<code>\imath</code>	$\in$	<code>\in</code>
∞	<code>\infty</code>	$\jmath$	<code>\jmath</code>	$\mho$	<code>\mho</code>
∇	<code>\nabla</code>	$\natural$	<code>\natural</code>	$\neg$	<code>\neg</code>
$\ni$	<code>\ni</code>	∂	<code>\partial</code>	$\prime$	<code>\prime</code>
$\Re$	<code>\Re</code>	$\sharp$	<code>\sharp</code>	$\spadesuit$	<code>\spadesuit</code>
$\surd$	<code>\surd</code>	$\top$	<code>\top</code>	$\triangle$	<code>\triangle</code>
$\wp$	<code>\wp</code>				

Текстовые символы в математической моде					
$\S$	<code>\mathdollar</code>	$\dots$	<code>\mathellipsis</code>	$\P$	<code>\mathparagraph</code>
$\S$	<code>\mathsection</code>	$\pounds$	<code>\mathsterling</code>	$_$	<code>\mathunderscore</code>

Дополнительные математические символы (amssymb)					
$\backprime$	<code>\backprime</code>	$\Bbbk$	<code>\Bbbk</code>	$\beth$	<code>\beth</code>
$\bigstar$	<code>\bigstar</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\blacksquare$	<code>\blacksquare</code>
$\blacktriangle$	<code>\blacktriangle</code>	$\blacktriangledown$	<code>\blacktriangledown</code>	$\circledR$	<code>\circledR</code>
$\circledS$	<code>\circledS</code>	$\complement$	<code>\complement</code>	$\daleth$	<code>\daleth</code>
$\diagdown$	<code>\diagdown</code>	$\diagup$	<code>\diagup</code>	$\eth$	<code>\eth</code>

$\lrcorner$	<code>\Finv</code>	$\ni$	<code>\Game</code>	$\beth$	<code>\gimel</code>
$\hslash$	<code>\hslash</code>	$\lozenge$	<code>\lozenge</code>	$\measuredangle$	<code>\measuredangle</code>
$\nexists$	<code>\nexists</code>	$\sphericalangle$	<code>\sphericalangle</code>	$\square$	<code>\square</code>
∇	<code>\triangledown</code>	$\varnothing$	<code>\varnothing</code>	$\triangle$	<code>\vartriangle</code>
Дополнительные математические символы (wasysym)					
$\Box$	<code>\Box</code>	$\Diamond$	<code>\Diamond</code>	$\vartriangleleft$	<code>\vartriangleleft</code>
$\therefore$	<code>\asytherefore</code>				

Комплексная арифметика

По западной традиции действительная (Re) и мнимая (Im) части комплексного числа набираются как $\Re$ (`\Re`) и $\Im$ (`\Im`) соответственно. Для правильного набора следует переопределить стандартные команды:

```
\renewcommand{\Re}{\text{Re}}
\renewcommand{\Im}{\text{Im}}
```

13.1.3. Греческие символы

Латинских букв часто не хватает для обозначения всех понятий, поэтому греческий алфавит и востребован для написания математических формул.

По умолчанию строчные греческие символы имеют наклонное начертание, а прописные в отличие от них прямые. Стилиевой файл **fixmath** из пакета **was** при загрузке меняет прямые начертания заглавных греческих букв на наклонные. В пакете **was** также определён стиль **upgreek**, обеспечивающий доступ к прямым начертаниям строчных греческих букв.

Греческий алфавит					
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
$\omicron$	<code>\omicron</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>
ϱ	<code>\varrho</code>	σ	<code>\sigma</code>	ς	<code>\varsigma</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>
ψ	<code>\psi</code>	ω	<code>\omega</code>	Γ	<code>\Gamma</code>
Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>
Ω	<code>\Omega</code>			Ψ	<code>\Psi</code>
Дополнительные греческие символы (amssymb)					
F	<code>\digamma</code>	κ	<code>\varkappa</code>		

Прямые греческие символы (upgreek)							
α	<code>\upalpha</code>	β	<code>\upbeta</code>	γ	<code>\upgamma</code>	δ	<code>\updelta</code>
ϵ	<code>\upepsilon</code>	ε	<code>\upvarepsilon</code>	ζ	<code>\upzeta</code>	η	<code>\upeta</code>
θ	<code>\uptheta</code>	ϑ	<code>\upvartheta</code>	ι	<code>\upiota</code>	κ	<code>\upkappa</code>
λ	<code>\uplambda</code>	μ	<code>\upmu</code>	ν	<code>\upnu</code>	ξ	<code>\upxi</code>
π	<code>\uppi</code>	ϖ	<code>\upvarpi</code>	ρ	<code>\uprho</code>	ϱ	<code>\upvarrho</code>
σ	<code>\upsigma</code>	ς	<code>\upvarsigma</code>	τ	<code>\uptau</code>	υ	<code>\upupsilon</code>
ϕ	<code>\upphi</code>	φ	<code>\upvarphi</code>	χ	<code>\upchi</code>	ψ	<code>\uppsi</code>
ω	<code>\upomega</code>	Γ	<code>Upgamma</code>	Δ	<code>Updelta</code>	Θ	<code>Uptheta</code>
Λ	<code>Uplambda</code>	Ξ	<code>Upxi</code>	Π	<code>Uppi</code>	Σ	<code>Upsigma</code>
Υ	<code>Upupsilon</code>	Φ	<code>Upphi</code>	Ψ	<code>Upsi</code>	Ω	<code>Upomega</code>

13.1.4. Акценты

Когда буквы кончаются, а понятия ещё остаются, то на помощь приходят разнообразные акценты:

Акценты					
$\acute{a}$	<code>\acute{a}</code>	$\bar{a}$	<code>\bar{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\grave{a}$	<code>\grave{a}</code>
$\hat{a}$	<code>\hat{a}</code>	$\mathring{a}$	<code>\mathring{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
				$\vec{a}$	<code>\vec{a}</code>

При желании с помощью пакета **amsxtra** акценты можно расставить не над символом, а сбоку:

Акценты							
$A^\vee$	<code>A\spcheck</code>	$A^\cdot$	<code>A\spdot</code>	$A^{\cdot\cdot}$	<code>A\spddot</code>	$A^{\cdots}$	<code>A\spdddot</code>
$A^\wedge$	<code>A\sphat</code>	$A^\sim$	<code>A\sptilde</code>				

Если и этого мало, то можно создавать составные акценты, комбинируя стандартные команды: $\check{\check{a}}$, $\check{\grave{a}}$, $\grave{\grave{a}}$ и так далее. Пакет **amsmath** гарантирует правильное выравнивание для составных акцентов.

Пакет **amsfonts** позволяет пометить комбинацию символов или формулу широкой крышечкой или тильдой:

```
[\begin{split}
&\widehat{\text{формула}}\\
&\widetilde{\text{ещё формула}}
\end{split}]
```

$$\widehat{\text{формула}}$$
$$\widetilde{\text{ещё формула}}$$

Кроме тильды и крышечки есть ещё стрелочки, фигурные скобки и просто линии.

Широкие акценты			
$\overleftarrow{abc}$	<code>\overleftarrow{abc}</code>	$\overline{abc}$	<code>\overline{abc}</code>
$\overrightarrow{abc}$	<code>\overrightarrow{abc}</code>	$\overbrace{abc}$	<code>\overbrace{abc}</code>
$\underline{abc}$	<code>\underline{abc}</code>	$\underbrace{abc}$	<code>\underbrace{abc}</code>

В пакете **cancel** определяются макросы `\cancel`, `\bcancel` и `\xcancel`, рисующие линии, перечёркивающие аргумент команды. Таким образом можно просто зачёркивать математические выражения. В этом же пакете переопределена команда `\cancelto{значение}{выражение}`, рисующая через выражение по диагонали стрелку, указывающую на значение:

```
\begin{align*}
&\cancel{2\times 2=5} \quad \xcancel{2+2=3} \\
&\bcancel{2^2=22} \quad \cancelto{5}{2+3}
\end{align*}
```

$2 \times 2 = 5$ $2 + 2 = 3$ $2^2 = 22$ $2 + 3 = 5$

13.1.5. Многоточия

Многоточия бывают разные:

Многоточия					
$\cdotp$	<code>\cdotp</code>	$\cdots$	<code>\cdots</code>	$:$	<code>\colon</code>
$\ddots$	<code>\ddots</code>	$\vdots$	<code>\vdots</code>	$\vdots$	<code>\vdots</code>

В соответствие с правилами отечественного набора [11] во всех случаях, кроме случая перемножения символов, используется многоточие, расположенное на базовой линии `\ldots`. При этом знак операции или препинания повторяется до и после многоточия.

При наборе многоточия в ряде перемножаемых символов многоточие располагается на средней линии `\cdots`. При этом многоточие символов никакими знаками не отделяется.

```
\[ \begin{split}
&a_1+a_2+\ldots+a_n \\
\text{и } &a_1, a_2, \ldots, a_n, \\
\text{но! } &a_1a_2\cdots a_n
\end{split} ]
```

► Пакет **amsmath** переопределяет команду `\dots` в соответствие с зарубежными правилами типографии так, что в зависимости от ситуации правильно выбирается `\ldots` или `\cdots`.

13.1.6. Математика в текстовой моде

Набирая математические символы в текстовой моде, следует проявлять аккуратность. Например, знак минус (−) значительно отличается от короткого тире (−), хотя символ один и тот же, но моды разные.

В пакете **textcomp** определено некоторое количество математических символов специально для текстовой моды:

Математические символы в текстовой моде (textcomp)			
°	<code>\textdegree</code>	÷	<code>\textdiv</code>
/	<code>\textfractionsolidus</code>	¬	<code>\textlnot</code>
−	<code>\textminus</code>	$\frac{1}{2}$	<code>\textonehalf</code>
$\frac{1}{4}$	<code>\textonequarter</code>	1	<code>\textonesuperior</code>
±	<code>\textpm</code>	√	<code>\textsurd</code>
$\frac{3}{4}$	<code>\textthreequarters</code>	3	<code>\textthreesuperior</code>
×	<code>\texttimes</code>	2	<code>\texttwosuperior</code>

13.1.7. Бинарные операторы

Кроме стандартных математических операторов сложения (+) и вычитания (−) в математике определено ещё множество других.

Бинарные операторы			
Π	<code>\amalg</code>	*	<code>\ast</code> или <code>*</code>
▽	<code>\bigtriangledown</code>	△	<code>\bigtriangleup</code>
∩	<code>\cap</code>	·	<code>\cdot</code>
∪	<code>\cup</code>	†	<code>\dagger</code>
◇	<code>\diamond</code>	÷	<code>\div</code>
∓	<code>\mp</code>	⊙	<code>\odot</code>
⊕	<code>\oplus</code>	⊗	<code>\otimes</code>
±	<code>\pm</code>	▷	<code>\rhd</code>
□	<code>\sqcap</code>	◻	<code>\sqcup</code>
×	<code>\times</code>	◁	<code>\triangleleft</code>
⊆	<code>\unlhd</code>	⊇	<code>\unrhd</code>
∨	<code>\vee</code>	∧	<code>\wedge</code>
○	<code>\bigcirc</code>	●	<code>\bullet</code>
		◦	<code>\circ</code>
		‡	<code>\ddagger</code>
		◁	<code>\lhd</code>
		⊖	<code>\ominus</code>
		⊗	<code>\otimes</code>
		\	<code>\setminus</code>
		★	<code>\star</code>
		▷	<code>\triangleright</code>
		⊕	<code>\uplus</code>
		ℓ	<code>\wr</code>

Дополнительные символы бинарных операторов (**amssymb**)

$\bar{\wedge}$	<code>\barwedge</code>	◻	<code>\boxdot</code>	⊖	<code>\boxminus</code>
⊞	<code>\boxplus</code>	⊠	<code>\boxtimes</code>	Ⓜ	<code>\Cap</code>
•	<code>\centerdot</code>	⊗	<code>\circledast</code>	⊙	<code>\circledcirc</code>
⊖	<code>\circleddash</code>	Ⓜ	<code>\Cup</code>	∪	<code>\curlyvee</code>
∪	<code>\curlywedge</code>	*	<code>\divideontimes</code>	+	<code>\dotplus</code>
$\overline{\wedge}$	<code>\doublebarwedge</code>	⌢	<code>\intercal</code>	∧	<code>\leftthreetimes</code>
⋈	<code>\ltimes</code>	⌣	<code>\rightthreetimes</code>	⋈	<code>\rtimes</code>

$\smallsetminus$	<code>\smallsetminus</code>	$\veebar$	<code>\veebar</code>
Дополнительные символы бинарных операторов (wasysym)			
$\triangleleft$	<code>\lhd</code>	$\blacktriangleleft$	<code>\LHD</code>
$\triangleright$	<code>\rhd</code>	$\blacktriangleright$	<code>\RHD</code>
$\trianglelefteq$	<code>\unrhd</code>	$\circ$	<code>\ocircle</code>
		$\trianglelefteq$	<code>\unlhd</code>

13.1.8. Символы отношений

Символы отношений					
$\approx$	<code>\approx</code>	$\asymp$	<code>\asymp</code>	$\bowtie$	<code>\bowtie</code>
$\cong$	<code>\cong</code>	$\dashv$	<code>\dashv</code>	$\doteq$	<code>\doteq</code>
$\equiv$	<code>\equiv</code>	$\frown$	<code>\frown</code>	$\preceq$	<code>\preceq</code>
$\perp$	<code>\perp</code>	$\models$	<code>\models</code>	$\parallel$	<code>\parallel</code>
$\perp$	<code>\perp</code>	$\prec$	<code>\prec</code>	$\propto$	<code>\propto</code>
$\sim$	<code>\sim</code>	$\simeq$	<code>\simeq</code>	$\smile$	<code>\smile</code>
$\succ$	<code>\succ</code>	$\succeq$	<code>\succeq</code>	$\vdash$	<code>\vdash</code>
Символы неравенства					
$\geq$	<code>\geq</code>	$\gg$	<code>\gg</code>	$\leq$	<code>\leq</code>
$\ll$	<code>\ll</code>	$\neq$	<code>\neq</code>		
Символы включения					
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\subset$	<code>\subset</code>
$\supset$	<code>\supset</code>	$\supseteq$	<code>\supseteq</code>	$\subseteq$	<code>\subseteq</code>
Отрицание символов отношений					
$\neq$	<code>\neq</code>	$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>
$\not\approx$	<code>\not\approx</code>	$\not\asymp$	<code>\not\asymp</code>	$\not\cong$	<code>\not\cong</code>
$\not\equiv$	<code>\not\equiv</code>	$\not\geq$	<code>\not\geq</code>	$\not\leq$	<code>\not\leq</code>
$\not\perp$	<code>\not\perp</code>	$\not\preceq$	<code>\not\preceq</code>	$\not\perp$	<code>\not\perp</code>
$\not\prec$	<code>\not\prec</code>	$\not\propto$	<code>\not\propto</code>	$\not\sim$	<code>\not\sim</code>
$\not\simeq$	<code>\not\simeq</code>	$\not\sqsubseteq$	<code>\not\sqsubseteq</code>	$\not\sqsupseteq$	<code>\not\sqsupseteq</code>
$\not\succ$	<code>\not\succ</code>	$\not\succeq$	<code>\not\succeq</code>	$\not\subset$	<code>\not\subset</code>
$\not\supset$	<code>\not\supset</code>	$\not\supseteq$	<code>\not\supseteq</code>	$\not\subseteq$	<code>\not\subseteq</code>
$\notin$	<code>\notin</code>				
Дополнительные символы отношений (amssymb)					
$\approxeq$	<code>\approxeq</code>	$\backepsilon$	<code>\backepsilon</code>	$\backsimeq$	<code>\backsimeq</code>
$\backsimeq$	<code>\backsimeq</code>	$\because$	<code>\because</code>	$\between$	<code>\between</code>
$\bumpeq$	<code>\bumpeq</code>	$\bumpeq$	<code>\bumpeq</code>	$\circeq$	<code>\circeq</code>
$\curlyeqprec$	<code>\curlyeqprec</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>	$\doteqdot$	<code>\doteqdot</code>
$\eqcirc$	<code>\eqcirc</code>	$\fallingdotseq$	<code>\fallingdotseq</code>	$\multimap$	<code>\multimap</code>
$\pitchfork$	<code>\pitchfork</code>	$\precapprox$	<code>\precapprox</code>	$\preccurlyeq$	<code>\preccurlyeq</code>

$\precsim$	<code>\precsim</code>	$\dot{=}$	<code>\risingdotseq</code>	$\shortmid$	<code>\shortmid</code>
$\shortparallel$	<code>\shortparallel</code>	$\smallfrown$	<code>\smallfrown</code>	$\smallsmile$	<code>\smallsmile</code>
$\succapprox$	<code>\succapprox</code>	$\succcurlyeq$	<code>\succcurlyeq</code>	$\succsim$	<code>\succsim</code>
$\therefore$	<code>\therefore</code>	$\thickapprox$	<code>\thickapprox</code>	$\thicksim$	<code>\thicksim</code>
$\varpropto$	<code>\varpropto</code>	$\Vdash$	<code>\Vdash</code>	$\vdash$	<code>\vdash</code>
$\Vdash$	<code>\Vdash</code>				

Дополнительные символы неравенства (**amssymb**)

$\gtrless$	<code>\gtrless</code>	$\lesseqgtr$	<code>\lesseqgtr</code>	$\geqq$	<code>\geqq</code>
$\gtrsim$	<code>\gtrsim</code>	$\ggg$	<code>\ggg</code>	$\gtrapprox$	<code>\gtrapprox</code>
$\gneq$	<code>\gneq</code>	$\gtrless$	<code>\gtrless</code>	$\gtrsim$	<code>\gtrsim</code>
$\gtrapprox$	<code>\gtrapprox</code>	$\gtrdot$	<code>\gtrdot</code>	$\gtreqless$	<code>\gtreqless</code>
$\gtreqless$	<code>\gtreqless</code>	$\gtrless$	<code>\gtrless</code>	$\gtrsim$	<code>\gtrsim</code>
$\gtrsim$	<code>\gtrsim</code>	$\leqq$	<code>\leqq</code>	$\leqslant$	<code>\leqslant</code>
$\gvertneqq$	<code>\gvertneqq</code>	$\lessdot$	<code>\lessdot</code>	$\lesseqgtr$	<code>\lesseqgtr</code>
$\lessapprox$	<code>\lessapprox</code>	$\lessgtr$	<code>\lessgtr</code>	$\lesssim$	<code>\lesssim</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\lnapprox$	<code>\lnapprox</code>	$\lneq$	<code>\lneq</code>
$\lll$	<code>\lll</code>	$\lnsim$	<code>\lnsim</code>	$\lvertneqq$	<code>\lvertneqq</code>
$\lneq$	<code>\lneq</code>	$\ngeqq$	<code>\ngeqq</code>	$\ngeqslant$	<code>\ngeqslant</code>
$\ngeq$	<code>\ngeq</code>	$\nleq$	<code>\nleq</code>	$\nleqq$	<code>\nleqq</code>
$\ngtr$	<code>\ngtr</code>	$\nless$	<code>\nless</code>		
$\nleqslant$	<code>\nleqslant</code>				

Дополнительные символы включения (**amssymb**)

$\nsubseteq$	<code>\nsubseteq</code>	$\nsupseteq$	<code>\nsupseteq</code>	$\nsupseteqq$	<code>\nsupseteqq</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\Subset$	<code>\Subset</code>
$\subsetneqq$	<code>\subsetneqq</code>	$\subsetneq$	<code>\subsetneq</code>	$\subsetneqq$	<code>\subsetneqq</code>
$\Supset$	<code>\Supset</code>	$\supseteqq$	<code>\supseteqq</code>	$\supsetneq$	<code>\supsetneq</code>
$\supsetneqq$	<code>\supsetneqq</code>	$\varsubsetneq$	<code>\varsubsetneq</code>	$\varsubsetneqq$	<code>\varsubsetneqq</code>
$\varsupsetneq$	<code>\varsupsetneq</code>	$\varsupsetneqq$	<code>\varsupsetneqq</code>		

Дополнительные символы отрицания (**amssymb**)

$\ncong$	<code>\ncong</code>	$\nmid$	<code>\nmid</code>	$\nparallel$	<code>\nparallel</code>
$\nprec$	<code>\nprec</code>	$\npreceq$	<code>\npreceq</code>	$\nshortmid$	<code>\nshortmid</code>
$\nshortparallel$	<code>\nshortparallel</code>	$\nsim$	<code>\nsim</code>	$\nsucc$	<code>\nsucc</code>
$\nsucceq$	<code>\nsucceq</code>	$\nvDash$	<code>\nvDash</code>	$\nvdash$	<code>\nvdash</code>
$\nVDash$	<code>\nVDash</code>	$\precnapprox$	<code>\precnapprox</code>	$\precnsim$	<code>\precnsim</code>
$\succnapprox$	<code>\succnapprox</code>	$\succnsim$	<code>\succnsim</code>		

Дополнительные треугольные символы (**amssymb**)

$\blacktriangleleft$	<code>\blacktriangleleft</code>	$\blacktriangleright$	<code>\blacktriangleright</code>
$\triangleleft$	<code>\triangleleft</code>	$\trianglelefteq$	<code>\trianglelefteq</code>
$\triangleright$	<code>\triangleright</code>	$\trianglerighteq$	<code>\trianglerighteq</code>

$\triangleleft$	<code>\trianglelefteq</code>	$\trianglelefteq$	<code>\trianglelefteq</code>
$\triangleright$	<code>\trianglerighteq</code>	$\vartriangleleft$	<code>\vartriangleleft</code>
$\vartriangleright$	<code>\vartrianglerighteq</code>		

Дополнительные символы отношений (**wasysym**)

$\gtrsim$	<code>\apprge</code>	$\lesssim$	<code>\apprl</code>	$\neg$	<code>\invneg</code>
$\Join$	<code>\Join</code>	$\leadsto$	<code>\leadsto</code>	$\otimes$	<code>\logof</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</code>	$\propto$	<code>\wasympto</code>

13.1.9. Стрелки

Стрелки			
$\Downarrow$	<code>\Downarrow</code>	$\downarrow$	<code>\downarrow</code>
$\hookleftarrow$	<code>\hookleftarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\leadsto$	<code>\leadsto</code>	$\leftarrow$	<code>\leftarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\leftharpoonup$	<code>\leftharpoonup</code>
$\lleftarrow$	<code>\lleftarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\Longleftarrow$	<code>\Longleftarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>
$\Longleftrightarrow$	<code>\Longleftrightarrow</code>	$\mapsto$	<code>\mapsto</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\nearrow$	<code>\nearrow</code>
$\mapsto$	<code>\mapsto</code>	$\Rightarrow$	<code>\Rightarrow</code>
$\nwarrow$	<code>\nwarrow</code>	$\rightarrowtail$	<code>\rightarrowtail</code>
$\rightarrow$	<code>\rightarrow</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\rightharpoonup$	<code>\rightharpoonup</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>
$\searrow$	<code>\searrow</code>	$\swarrow$	<code>\swarrow</code>
$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>

Дополнительные стрелки (**amssymb**)

$\circlearrowleft$	<code>\circlearrowleft</code>	$\circlearrowright$	<code>\circlearrowright</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>
$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>
$\downdownarrows$	<code>\downdownarrows</code>	$\downharpoonleft$	<code>\downharpoonleft</code>
$\downharpoonright$	<code>\downharpoonright</code>	$\leftarrowtail$	<code>\leftarrowtail</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\leftrightarrows$	<code>\leftrightarrows</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>
$\looparrowleft$	<code>\looparrowleft</code>	$\looparrowright$	<code>\looparrowright</code>
$\looparrowright$	<code>\looparrowright</code>	$\Lsh$	<code>\Lsh</code>
$\rightarrowtail$	<code>\rightarrowtail</code>	$\rightleftarrows$	<code>\rightleftarrows</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\rightrightarrows$	<code>\rightrightarrows</code>
$\rightsquigarrow$	<code>\rightsquigarrow</code>	$\Rsh$	<code>\Rsh</code>

13.1.10. Разделители

Скобки переменного размера					
$(($	$($	$)$	$)$	$[$	$[$
$]]$	$)$	$\{ \{$	$\backslash \{$	$\} \}$	$\backslash \}$
$ $	$ $	$\backslash \backslash$	<code>\backslash slash</code>	$\downarrow \downarrow$	<code>\downarrow arrow</code>
$\Downarrow \Downarrow$	<code>\Downarrow</code>	$\langle \langle$	<code>\langle</code>	$\lceil \lceil$	<code>\lceil</code>
$\lfloor \lfloor$	<code>\lfloor floor</code>	$\rangle \rangle$	<code>\rangle angle</code>	$\rceil \rceil$	<code>\rceil</code>
$\rfloor \rfloor$	<code>\rfloor floor</code>	$\uparrow \uparrow$	<code>\uparrow arrow</code>	$\Uparrow \Uparrow$	<code>\Upward arrow</code>
$\updownarrow \updownarrow$	<code>\updownarrow arrow</code>	$\Updownarrow \Updownarrow$	<code>\Updownarrow arrow</code>		
Дополнительные разделители (amssymb)					
$ $	<code>\lvert</code>	$\ \ $	<code>\lVert</code>	$ $	<code>\rvert</code>
$\ \ $	<code>\rVert</code>	$\llcorner \llcorner$	<code>\llcorner</code>	$\lrcorner \lrcorner$	<code>\lrcorner</code>
$\ulcorner \ulcorner$	<code>\ulcorner</code>	$\urcorner \urcorner$	<code>\urcorner</code>		

13.1.11. Операторы переменного размера

Операторы переменного размера			
$\bigcap$	<code>\bigcap</code>	$\bigcup$	<code>\bigcup</code>
$\bigodot$	<code>\bigodot</code>	$\bigoplus$	<code>\bigoplus</code>
$\bigotimes$	<code>\bigotimes</code>	$\bigsqcup$	<code>\bigsqcup</code>
$\biguplus$	<code>\biguplus</code>	$\bigvee$	<code>\bigvee</code>
$\bigwedge$	<code>\bigwedge</code>	$\coprod$	<code>\coprod</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>
$\prod$	<code>\prod</code>	$\sum$	<code>\sum</code>
Дополнительные операторы переменного размера (amsmath)			
$\iint$	<code>\iint</code>	$\iiint$	<code>\iiint</code>
$\iiint$	<code>\iiint</code>	$\int \cdots \int$	<code>\int \cdots \int</code>

13.2. Производные и интегралы

Фактически всё, что нужно знать про производные и интегралы, уже было изложено в разделе 4.3.7 на стр. 52, где рассказывалось про школьную математику. В разделе 13.1.11 перечислены дополнительные символы, которые можно использовать для представления сложных интегралов.

commath

Пакет **commath** (документация `commath.pdf`.) специализируется на отображении дифференциалов, производных и пределов.

<code>\[</code>	<code>\begin{split}</code>		
<code>&\dif{x}</code>	<code>\quad \mathrm{d}x</code>	<code>\quad \mathrm{d}x</code>	
<code>&\od{f}{x}</code>	<code>\quad \frac{\mathrm{d}f}{\mathrm{d}x}</code>	<code>\quad \frac{\mathrm{d}^2 f}{\mathrm{d}x^2}</code>	
<code>&\pd{f}{x}</code>	<code>\quad \frac{\partial f}{\partial x}</code>	<code>\quad \frac{\partial^2 f}{\partial x^2}</code>	<code>\quad \frac{\partial^5 f}{\partial x^2 \partial y^3}</code>
<code>&\md{f}{5}{x}{2}{y}{3}</code>			
<code>&\eval{0}{f(x)}</code>	<code>\quad f(x) _{x=0}</code>	<code>\quad f(x) _{x=0}</code>	<code>\quad f(x) _{x=0}</code>
<code>&\eval{2}{f(x)}</code>			
<code>&\eval{f(x)}</code>			
<code>&\fullfunction{f}</code>	<code>\quad f</code>	<code>\quad \mathbb{R} \longrightarrow \mathbb{R}</code>	
<code>\quad \mathbb{R} \longrightarrow \mathbb{R}</code>		<code>\quad x \longmapsto \sqrt{x}</code>	
<code>\end{split}</code>	<code>\]</code>		

esdiff

Надоело постоянно писать $\frac{\partial^3 f}{\partial x \partial y^2}$? Тогда, возможно, немного помогут макросы, определённые в пакете **esdiff**:

```
\[ \begin{split}
&\&\diff{f}{x}\quad\diff[n]{f}{x}\quad\frac{df}{dx}\quad\frac{d^n f}{dx^n}\quad\left(\frac{df}{dx}\right)_{x_0}\quad\left(\frac{d^n f}{dx^n}\right)_{x_0} \\
&\diff*{f}{x}{x_0}\quad\diff*[n]{f}{x}{x_0} \\
&\diffp{f}{x}\quad\diffp[3]{f}{x}\quad\frac{\partial f}{\partial x}\quad\frac{\partial^3 f}{\partial x^3}\quad\frac{\partial^5 f}{\partial x^3 \partial y^2}\quad\left(\frac{\partial P}{\partial V}\right)_T \\
&\diffp{f}{x^3 y^2}\quad\diffp*{P}{V}{T}
\end{split} \]
```

Документацию к пакету можно получить из файла исходников **esdiff.dtx**.

Стилю **esdiff** при инициализации можно передавать опции для изменения значений форматирования производных по умолчанию:

text — возвращает стиль `\textstyle` для отображения производной в текстовой моде. По умолчанию используется стиль `\displaystyle`;

italic — знак *d* в производной отображается курсивом (*d*). По умолчанию используется опция **roman**;

thin, **med** и **big** — регулирует расстояние между знаком производной *d* и обозначением функции или переменной. По умолчанию пробел отсутствует совсем;

thinp, **medp** и **bigp** — то же, что и предыдущий пункт, но для знака ∂ ;

thinc, **medc** и **bigc** — задаёт расстояние между ∂x и ∂y в частной производной вида $\diffp{f}{x}{y}$. По умолчанию пробел отсутствует.

При использовании **esdiff** рекомендуется загружать опцию **thinc**:

```
\usepackage[thinc]{esdiff}
```

Прямые интегралы

Традиционно в российской математической типографии использовались прямые символы интегрирования. С другой стороны основной язык науки сейчас английский и к наклонным символам тоже можно привыкнуть.

Прямые интегралы можно получить с помощью пакета **wasysym**, загруженного с опцией **integrals**. Также прямые интегралы появляются при использовании математических шрифтов **euler**, но в этом случае для согласования текста и математики необходимо использовать шрифты Concrete (пакет **concrete**, чуть более подробная информация в разделе 10.3 на стр. 126).

`\[% Интегралы по умолчанию`

`\int\quad\iint\quad\iiint\quad\oint\ dx`

`\]`

$$\int \quad \iint \quad \iiint \quad \oint dx$$

```
\[%\usepackage[integrals]{wasysym}
\int\quad\iint\quad\iiint\quad\oint\ dx
\]
\[%\usepackage{concrete}
\int\quad\iint\quad\iiint\quad\oint\ dx
\]
```

$$\int \iint \iiint \oint dx$$

$$\int \iint \iiint \oint dx$$

13.3. Матрицы, тензоры и диаграммы

Для набора матриц можно использовать стандартное окружение для создания таблиц в математической моде агау (см. раздел 11.3 на стр. 151).

```
\[\left(\begin{array}{ccc}\!
a_{11}&a_{12}&a_{13}\\
a_{21}&a_{22}&a_{23}\\
a_{31}&a_{32}&a_{33}\end{array}\right)
\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=
\begin{pmatrix}b_1\\b_2\\b_3\end{pmatrix}\]
```

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

При описании формата окружения агау комбинация символов @{} используется для подавления пробелов перед первым и после последнего столбцов. В отличие от окружения агау окружения из семейства matrix не требуют описания формата столбцов. Это проще при наборе, но теряется возможность выравнивания элементов матрицы.

```
\begin{gather*}
\begin{matrix}1&2&3&4\end{matrix}\quad\quad
\begin{pmatrix}5&6&7&8\end{pmatrix}\quad\quad
\begin{bmatrix}9&A\\B&C\end{bmatrix}
\begin{vmatrix}D&E\\F&10\end{vmatrix}\quad\quad
\begin{Vmatrix}11&12\\13&14\end{Vmatrix}
\end{gather*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \begin{bmatrix} 9 & A \\ B & C \end{bmatrix} \begin{vmatrix} D & E \\ F & 10 \end{vmatrix} \begin{Vmatrix} 11 & 12 \\ 13 & 14 \end{Vmatrix}$$

Для многоточия, простирающегося на несколько ячеек, используется команда \hdotsfor[множитель для разрядки точек]{число столбцов}:

```
\[\left(\begin{array}{cccc}\!
a_{11}&a_{12}&\ldots&a_{1n}\\
a_{21}&a_{22}&\ldots&a_{2n}\\
\hdotsfor[2.5]{4}\\
a_{n1}&a_{n2}&\ldots&a_{nn}\end{array}\right)\]
```

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

tensor

Для представления тензоров со смешанными индексами можно воспользоваться пакетом **tensor**:

```
\[ \begin{split}
&\&M\indices{^a_b^{cd}_e}\quad \\
&\&\text{tensor}[^a_b^{cd}\{M\}\{^a_b^{cd}\}\quad \\
&\&\text{tensor}[^*_*]{\prod\limits\{^*_*\}\quad \\
&\&\text{tensor}\{M\}\{^*\{i_1\}_{m_1}\{i_2\}_{m_2}\quad \\
&\&\quad \quad \quad \{i_3\}_{m_3}\{i_4\}_{m_4}\}\quad \\
&\&\text{tensor}\{^{[14]}_6\}\{\mathrm{C}\}\quad \\
&\&\nuclide[14][6]{C} \\
&\end{split} \]
```

$$M_{b\ e}^{a\ cd} \quad {}^a_b\ c_d M_{b\ d}^{a\ c}$$

$$\prod_{*}^{*} M_{m_1 m_2 m_3 m_4}^{i_1 i_2 i_3 i_4}$$

$${}^{14}_6 C \quad {}^{14}_6 C$$

Можно просто добавлять индексы к символу с помощью команды `\indeces`, а можно воспользоваться командой `\tensor`, у которой есть необязательный аргумент для набора индексов перед символами. Эти же команды, но со звёздочкой `«*»` на конце, убирают пробелы между индексами.

Этот же пакет можно использовать для набора названий химических элементов. Для удобства определена команда `\nulide`. Дополнительную информацию следует искать в документации **tensor.pdf**.

amscd

Для создания простых коммутативных диаграмм можно воспользоваться окружением **array**, но окружение **CD**, определённое в стилевом файле **amscd**¹, подходит для этого лучше.

```
\[ \begin{CD}
F(X) @>{F(f)}>> F(Y) \\
@V{\eta_X}VV @VV{\eta_Y}V \\
G(X) @<<{G(f)}<< G(Y) \\
\end{CD} \]
```

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ \eta_X \downarrow & & \downarrow \eta_Y \\ G(X) & \xleftarrow{G(f)} & G(Y) \end{array}$$

Команды `@>>>`, `@<<<`, `@VVV`, `@AAA` определяют стрелки вправо, влево, вниз и вверх соответственно. Три одинаковых знака в каждой команде необходимы для набора индексов. Если выражение находится между первым и вторым символом в командах для определения горизонтальных стрелок, то это выражение печатается сверху стрелки, а если между вторым и третьим, то снизу. Аналогично и для команд, определяющих вертикальные стрелки: между первым и вторым символом — печать слева от стрелки, а между вторым и третьим — справа.

¹Стиль **amscd** является частью пакета $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$, но при инициализации **amsmath** автоматически не загружается.

► Пакет **amscd** позволяет создавать простые коммутативные диаграммы, для более сложных случаев следует обратиться к более общим или специализированным пакетам, например, к пакету **xypic**.

Биномиальные коэффициенты

В **amsmath** определена команда `\binom`, которая легко позволяет записать «Бином Ньютона»:

$$\backslash[(x+y)^n=\sum_{k=0}^n\binom{n}{k}x^ky^{n-k}\backslash]$$

$$(x+y)^n=\sum_{k=0}^n\binom{n}{k}x^ky^{n-k}$$

13.4. Пробелы в формулах

Л^AT_EX хорошо расставляет пробелы между математическими символами, но временами формулы требуют ручной доработки. Например, знак приращения, дифференциала или частной производной следует отделять от предыдущей комбинации символов небольшим пробелом (`\partial x\`, `\partial y`). Иногда могут потребоваться пробелы больших размеров или даже отрицательные.

Пробелы в математических формулах			
Положительные пробелы		Отрицательные пробелы	
$\boxed{A}\boxed{B}$	<code>\(AB\)</code>		
$\boxed{A}\ \boxed{B}$	<code>\(A\ B\)</code>		
$\boxed{A}\ \boxed{B}$	<code>\(A\,,\ B\)</code>	$\boxed{A}\!\!\boxed{B}$	<code>\(A\!B\)</code>
$\boxed{A}\ \boxed{B}$	<code>\(A\colon B\)</code>	$\boxed{A}\!\!\boxed{B}$	<code>\(A\!\negmedspace B\)</code>
$\boxed{A}\ \boxed{B}$	<code>\(A\; B\)</code>	$\boxed{A}\!\!\boxed{B}$	<code>\(A\!\negthickspace B\)</code>
$\boxed{A}\ \boxed{B}$	<code>\(A\ B\)</code>		
$\boxed{A}\ \boxed{B}$	<code>\(A\quad B\)</code>		
$\boxed{A}\ \boxed{B}$	<code>\(A\qqquad B\)</code>		
$\boxed{A}\ \boxed{B}$	<code>\(A\hspace{5mm}B\)</code>	$\boxed{B}$	<code>\(A\hspace{-5mm}B\)</code>
$\boxed{A}\ \boxed{B}$	<code>\(A\hphantom{xyz}B\)</code>		
$\boxed{A}\,xyz\boxed{B}$	<code>\(AxyzB\)</code>		

В таблице пробелов для лучшего их представления буквы A и B заключены в рамку с помощью команды `\fbox`.

Следует обратить внимание на команду `\hphantom`. Эта команда позволяет сформировать горизонтальный пробел, в точности равный ширине текста, передаваемого её аргументу. Кроме команды `\hphantom` существуют аналогичные по функционалу команды `\vphantom` и `\phantom`.

Команда `\phantom{аргумент}` образует пустой бокс, полностью совпадающий по размеру с её аргументом, а команда `\vphantom` создаёт «подпорку», равную по высоте аргументу.

13.5. Многострочные формулы

В главе 4, где описывались начала математики, все формулы однострочные. В реальности математические выражения постоянно норовят выйти за пределы не только строки, но и страницы.

По умолчанию разрыв страницы не может появиться внутри выключенной многострочной формулы, созданной средствами пакета **amsmath**. Причина этого поведения заключается в том, что такие проблемные места должны контролироваться автором лично. Команда `\allowdisplaybreaks` меняет это умолчание, позволяя L^AT_EX добавлять разрыв страницы внутри выносной формулы автоматически. Также автор может указать место разрыва внутри многострочной формулы с помощью декларации `\displaybreak`, разместив её непосредственно перед командой переноса строки `\\`.

► Если нумерация формул в тексте не нужна, то следует заменить окружение для создания однострочных формул `equation` на `equation*`:

<code>\begin{equation}\label{eq:ex1}</code>		
<code>\fbox{\text{Однострочная формула}}</code>	Однострочная формула	(13.1)
<code>\end{equation}</code>		
<code>\begin{equation*}%без номера</code>		
<code>\fbox{\text{Однострочная формула}}</code>	Однострочная формула	
<code>\end{equation*}</code>		

Для объединения нескольких однострочных формул в одну общую группу используются окружения `align`/`align*`:

<code>\begin{align}</code>		
<code>&\fbox{\text{Несколько формул}}\\</code>	Несколько формул	(13.2)
<code>&\fbox{\text{внутри одного окружения}}\\</code>	внутри одного окружения	(13.3)
<code>&\fbox{\text{с позиционированием}}</code>	с позиционированием	(13.4)
<code>\end{align}</code>		

Внутри окружения `align` строки разделяются с помощью стандартной команды переноса `\\`. Выравнивание выполняется по символу `&`. Часто выравнивание производится по знаку равенства или по какому-либо бинарному оператору. Формулы можно выравнивать по нескольким точкам.

<pre>\begin{align} x'&=\gamma(x-\beta ct) \quad &y'&=y\\ ct'&=\gamma(ct-\beta x) \quad &z'&=z \end{align}</pre>	$x' = \gamma(x - \beta ct) \quad y' = y \quad (13.5)$ $ct' = \gamma(ct - \beta x) \quad z' = z \quad (13.6)$
---	--

Существует два расширения для align: flalign и alignat. У alignat имеется обязательный параметр, указывающий число формул в одной строке. Если это число, например, равно 2, то число столбцов в окружении равно $2 \times 2 = 4$. Эти расширения внутри имеют тот же синтаксис, что и align/align*, но в отличие от них flalign/flalign* максимизируют расстояния между группами формул, а использование alignat/alignat даёт возможность задавать расстояния вручную:

<pre>\begin{align*} x'&=\gamma(x-\beta ct) \quad &y'&=y\\ ct'&=\gamma(ct-\beta x) \quad &z'&=z \end{align*} \begin{flalign*} x'&=\gamma(x-\beta ct) \quad &y'&=y\\ ct'&=\gamma(ct-\beta x) \quad &z'&=z \end{flalign*} \begin{alignat*}{2} x'&=\gamma(x-\beta ct) \quad &y'&=y\\ ct'&=\gamma(ct-\beta x) \quad &z'&=z \end{alignat*}</pre>	$x' = \gamma(x - \beta ct) \quad y' = y$ $ct' = \gamma(ct - \beta x) \quad z' = z$ $x' = \gamma(x - \beta ct) \quad y' = y$ $ct' = \gamma(ct - \beta x) \quad z' = z$ $x' = \gamma(x - \beta ct) \quad y' = y$ $ct' = \gamma(ct - \beta x) \quad z' = z$
--	--

Если выравнивание формул не нужно, то сгодятся окружения gather/gather*:

<pre>\begin{gather} \fbbox{\text{Несколько формул}}\\ \fbbox{\text{без}}\\ \fbbox{\text{выравнивания}} \end{gather}</pre>	Несколько формул (13.7) без (13.8) выравнивания (13.9)
---	---

Для формирования многострочной формулы существует окружение multiline:

<pre>\begin{multiline} \fbbox{\text{Ооочень}}\\ \fbbox{\text{длинная}}\\ \shoveleft{\fbbox{\text{многострочная}}}\fbbox{\text{формула}} \end{multiline}</pre>	Ооочень длинная многострочная формула (13.10)
---	--

Все строки окружения `multline`, кроме первой и последней, по умолчанию центрируются по ширине. Первая строка выравнивается по левому краю, а последняя — по правому. Команда `\shoveleft` позволяет выравнивать указанную строку по левому краю, а `\shoveright`, соответственно, по правому.

Окружение `split` также позволяет разбить длинную формулу:

<pre>\begin{equation} \begin{split} &\fbox{\text{Разбиение формулы}}\backslash &\fbox{\text{на строки с выравниванием}} \end{split} \end{equation}</pre>	Разбиение формулы на строки с выравниванием	(13.11)
--	---	---------

Внутри `split` разные части формулы можно выравнивать с помощью символа `&` по примеру окружения `align`.

► Следует обратить внимание, что если номер формулы не помещается в строке, то нумерация переносится ниже.

Окружение `split` можно использовать только внутри окружений, формирующих выключенную формулу. `split` не умеет расставлять нумерацию, поэтому целиком полагается в этом на внешнее окружение. По умолчанию номер формулы центрируется по вертикали (опция `centertags` пакета **amsmath**), но если при загрузке пакета **amsmath** указать опцию `tbtags`, то номер формулы размещается в последней строке, если номера ставятся справа, и в первой строке, если умолчание изменено, и номера выводятся слева (опция `leqno` пакет **amsmath**).

Окружения `aligned`, `gathered` и `alignedat`, как и `split`, могут использоваться только внутри выключенной формулы:

<pre>\begin{equation} \begin{aligned}[b] x'&=\gamma(x-\beta ct) \backslash ct'&=\gamma(ct-\beta x) \end{aligned}\quad \begin{gathered}[t] y'=y\backslash z'=z \end{gathered} \end{equation}</pre>	$ \begin{aligned} x' &= \gamma(x - \beta ct) \\ ct' &= \gamma(ct - \beta x) \quad y' = y \quad z' = z \end{aligned} $	(13.12)
---	---	---------

Синтаксис этих окружений соответствует синтаксису схожих по звучанию с ними окружений, за исключением необязательного параметра, который позволяет выравнивать конструкции по вертикали. Параметр может принимать значения:

- [c] — выравнивание по центральной линии (используется по умолчанию),
- [b] — выравнивание по последней строке,
- [t] — выравнивание по первой строке.

Вставка текста

Для вставки небольших фрагментов текста между формулами можно воспользоваться командой `\intertext`.

```
\begin{align}
\begin{cases}
\dot{x} = -R\omega \sin \omega t \\
\dot{y} = \hphantom{-}R\omega \cos \omega t
\end{cases}
\end{cases} \quad \text{следовательно,}
```

$$\begin{cases} \dot{x} = -R\omega \sin \omega t \\ \dot{y} = R\omega \cos \omega t \end{cases} \quad (13.13)$$

```
\intertext{следовательно,}
a_{\tau}=0 \quad (13.14)
\end{align}
```

Использование команды `\intertext` позволяет не нарушать выравнивание формул внутри окружения. Пакет **nccmath** переопределяет команду `\intertext`, добавляя необязательный параметр, задающий вертикальный промежуток между текстом и окружающими его формулами: `\intertext[расстояние]{текст}`.

► Окружение `cases` из примера можно использовать для оформления систем уравнений.

cases

В дополнение к стандартному окружению `cases` одноимённый стилевой файл **cases** из коллекции **ltxmisc** определяет простое окружение `numcases` для отображения простых условных конструкций:

```
\begin{numcases}{|x|=}
x, & \text{для } \$x \geq 0\$ \\
-x, & \text{для } \$x < 0\$
\end{numcases}
```

$$|x| = \begin{cases} x, & \text{для } x \geq 0 \\ -x, & \text{для } x < 0 \end{cases} \quad (13.15)$$

$$|x| = \begin{cases} x, & \text{для } x \geq 0 \\ -x, & \text{для } x < 0 \end{cases} \quad (13.16)$$

empheq

Пакет **mh** содержит множество улучшений и дополнений для **amsmath**. В частности туда входит стиль **empheq**, предоставляющий разнообразные средства для «украшения» формул, например вот так:

```
\begin{empheq}[box=\shadowbox*]{align}
E&=mc^2 \tag{*} \\
Y&= \sum_{n=1}^{\infty} \frac{1}{n^2} \notag
\end{empheq}
```

$$\begin{aligned} E &= mc^2 \\ Y &= \sum_{n=1}^{\infty} \frac{1}{n^2} \end{aligned} \quad (*)$$

Подробнейшая документация **empheq.pdf** позволит сделать даже самый унылый математический трактат «нарядным».

13.6. Нумерация формул

Как правило, если для оформления выключенной формулы используется окружение без символа `*` на конце, то формула нумеруется автоматически. Если номер не помещается в той же строке, что и формула, то он выводится чуть ниже.

► Следует обратить внимание на то, что традиционно при формировании ссылки на номер формулы вокруг неё выставляются скобки. Команда `\ref` это не делает, зато это умеет делать `\eqref`:

На формулу можно сослаться
так `\ref{eq:num1}` и так `\eqref{eq:num1}`.

На формулу можно сослаться
так **13.17** и так **(13.17)**.

По умолчанию номер формулы выводится справа от неё с выравниванием по правой границе текста. Умолчание можно изменить, например, с помощью опции `leqno`, передаваемой при загрузке пакета **amsmath**. В этом случае номера формул будут выводиться слева от них.

Если автоматическая нумерация формул не устраивает, то всегда с помощью команд `\tag` и `\tag*` можно поставить свою метку. Аналогично можно пропустить номер формулы с помощью команды `\notag`.

```
\newcommand{\mft}[1]{\fbox{\text{#1}}}\begin{gather}\mft{{Формула \No 1}}\label{eq:num1}\\ \mft{{Формула \No 2}}\notag\\ \mft{{Формула \No 3}}\tag{$*$}\label{eq:num2}\\ \mft{{Формула \No 4}}\tag*{$*}$\\ \mft{{Формула \No 5}}\tag{\ref{eq:num1}$'$}\\ \mft{{Формула \No 6}}\tag{\ref{eq:num2}$'$}\\ \end{gather}
```

Формула №1 (13.17)

Формула №2

Формула №3 (*)

Формула №4 *

Формула №5 (13.17')

Формула №6 (*')

С помощью `\tag \ref` можно организовать подчинённую нумерацию формул.

Подчинённая нумерация формул

Стандартные механизмы формирования подчинённой нумерации охватывают далеко не все возможные варианты создания иерархии формул. Если необходимо что-то нестандартное, то, как было показано в предыдущем примере, подчинённую нумерацию достаточно легко сделать руками.

Существует также некоторое количество пакетов, специализирующихся на решении этой проблемы, например, пакет **deleq**. За подробностями следует обратиться к файлу документации `deleq.dvi`.

Пакет **amsmath** определяет окружение `subequations`, которое формирует подчинённую нумерацию автоматически.

```
\begin{equation}
  \fbox{\text{Формула \No 1}}
```

```
\end{equation}
```

```
\begin{subequations}
```

```
\label{eq:num3}
```

```
\begin{equation}
```

```
\label{eq:num3-1}
```

```
\fbox{\text{Формула \No 2}}
```

```
\end{equation}
```

```
\begin{equation}
```

```
\fbox{\text{Формула \No 3}}
```

```
\end{equation}
```

```
\end{subequations}\par
```

Можно сослаться на группу~\eqref{eq:num3},

а можно на конкретную

формулу~\eqref{eq:num3-1}.

Формула №1

 (13.18)

Формула №2

 (13.19a)

Формула №3

 (13.19b)

Можно сослаться на группу (13.19), а можно на конкретную формулу (13.19a).

Вид и формат подчинённой нумерации можно изменить, переопределив команду \theequation, как показано далее:

```
\renewcommand{\theequation}
      {\theparentequation \asbuk{equation}}
```

Теперь подчинённая нумерация будет выводиться с помощью русских букв.

► Естественный механизм автоматической нумерации формул в ЛАТЭХ был одной из основных причин, по которой ТЭХ-сообщество предпочло ЛАТЭХ чистому ТЭХ.

13.7. Теоремы

В разделе 9.3 на стр. 111, посвящённом описанию презентационного класса **beamer**, уже упоминалось о возможности создавать новые именованные окружения с помощью команды \newtheorem:

```
\newtheorem{Thexmpl}{Пример}
```

```
\begin{Thexmpl}[Теорема Пифагора]
```

```
\label{th:Pythagor}
```

Пифагоровы штаны во все стороны равны.

```
\end{Thexmpl}
```

```
\begin{Thexmpl}
```

```
\label{th:WisdomLaw}
```

Мудрость ограничена, а глупость
бесконечна.

```
\end{Thexmpl}
```

Можем сослаться на первую теорему:

Теорема~\ref{th:Pythagor}, а можно

и на вторую: Теорема~\ref{th:WisdomLaw}.

Пример 1 (Теорема Пифагора).

Пифагоровы штаны во все стороны равны.

Пример 2. *Мудрость ограничена, а глупость бесконечна.*

Можем сослаться на первую теорему: Теорема 1, а можно и на вторую: Теорема 2.

Команда `\newtheorem` имеет несколько форм:

```
\newtheorem{«теорема»}{«заголовок»}
\newtheorem{«теорема»}[«существующая теорема»]{«заголовок»}
\newtheorem{«теорема»}{«заголовок»}[«имя счётчика»]
```

В первом случае создаётся «теорема» с указанным «заголовком» и своей собственной нумерацией. Во втором случае созданная «теорема» будет пользоваться тем же счётчиком, что и уже «существующая теорема». В третьем случае «теорема» для формирования своей нумерации будет использовать указанный счётчик (например, `section`). При этом нумерация «теоремы» по отношению к этому счётчику будет подчинённой.

Определив с помощью `\newtheorem` новое окружение, его можно использовать для формирования математических утверждений:

```
\begin{«теорема»}[«подзаголовок»]
    «текст утверждения»
\end{«теорема»}
```

► Пакет **nccthm** из коллекции **ncctools** содержит множество улучшений по сравнению со стандартными механизмами формирования математических утверждений, а также имеет русскую локализацию. Исчерпывающую документацию по этому замечательному пакету можно найти в файле **nccthm.pdf** или в [10].

Глава 14

Документация и программный код

+++ Ошибка Деления На Огурец.

Переустановите Вселенную И Перезагрузитесь +++

Так зависит Гекс

«Санта-Хрякус» Терри Пратчетт

Если вспомнить историю, то Д. Э. Кнут создал $\text{\TeX}$ именно для целей представления кода и алгоритмов в своём глобальном пятитомнике «Искусство программирования».

Написание документации — это неотъемлемая часть процесса создания качественного программного продукта. $\text{\LaTeX}$ сам по себе качественный инструмент и достоин быть включённым в эту технологическую цепочку.

14.1. Форматирование кода

$\text{\LaTeX}$ может использоваться не только для набора математики. Хотя набор математики безусловно вершина типографского искусства, но есть масса задач, где сложность форматирования сравнима. Представление исходников программ — это весьма не простое занятие, особенно если хочется сделать код читаемым.

Традиционно в книгах из-за избыточной сложности код программы делают одним шрифтом фиксированной ширины. И все настолько привыкли к такому порядку, что любые попытки «раскрасить» код натываются на иррациональное отторжение. Возможно, со временем ситуация поменяется в том числе и из-за возможностей $\text{\LaTeX}$ в деле формирования читаемой распечатки программы.

verbatim

Простейший способ включения кода в текст — это заключить его в стандартное окружение `verbatim`, которое просто выводит этот текст на печать с отступами пользователя.

Небольшие вставки можно делать с помощью команды `\verb!текст!`. Сразу после `\verb` должен идти группирующий символ (в данном случае «!»), который обязан указать окончание действия команды. Группирующий символ может быть любым, кроме пробела или звёздочки «*».

Пакет **verbatim** из коллекции **tools** переопределяет стандартную команду так, что внутри окружения можно вставлять тексты неограниченного размера. Кроме этого пакет предоставляет команду `\verbatiminput`, которой в качестве основного аргумента можно передать имя внешнего файла.

Кроме упомянутых макросов и окружения определены такие же, но со звёздочкой в конце имени, то есть команды `\verb*/\verbatiminput*` и окружение `verbatim*`. *-форма отличается от базовой тем, что все пробелы визуализируются с помощью символа «_» (`\textvisiblespace`).

```
\begin{verbatim*}
\textbf{verbatim} в      \LaTeX{}      \textbf{verbatim}_BBBBBBBB\LaTeX{}
\end{verbatim*}
```

altt

Стандартный пакет **altt** — это почти то же самое, что и **verbatim**, но позволяет использовать команды $\TeX$ внутри окружения, правда, шрифт в любом случае остаётся фиксированной ширины, как для печатной машинки.

```
\begin{alltt}
\textbf{altt} в          \LaTeX{}          altt в           $\TeX$ 
\end{alltt}
```

Гораздо более разнообразные средства управления выводом неформатированного текста предоставляет пакет **fancyvrb**. За подробной информацией следует обратиться к документации пакета `fancyvrb.pdf`.

listings

Пакет **listings** специализируется на оформлении программного кода. К пакету прилагается подробнейшая документация¹, которую имеет смысл изучить, если выводить код на печать приходится часто. С помощью команд пакета можно включить как небольшие кусочки кода, так и целые его сегменты, ну и, естественно, файлы.

¹Следует поискать файл `listings.pdf`.

Для загрузки пакета **listings** необходимо добавить в заголовок следующие инструкции:

Listing 14.1. Заголовок listings

```
\usepackage{listings}
% подгружаемые языки — подробнее в документации listings
\lstloadlanguages {[LaTeX]TeX, bash, MetaPost, Perl, C++, make}
% включаем кириллицу и добавляем кое-какие опции
\lstset{language=[LaTeX]TeX, % выбираем язык по умолчанию
        extendedchars=true, % включаем не латиницу
        escapechar=|, % |«выпадаем» в LATEX|
        frame=tb, % рамка сверху и снизу
        commentstyle=\itshape, % шрифт для комментариев
        stringstyle=\bfseries} % шрифт для строк
```

Сразу после загрузки пакета рекомендуется «подгрузить» используемые в тексте языки программирования² с помощью макроса `\lstloadlanguages`. В квадратных скобках перед названием языка можно указать желательный диалект.

Команда `\lstset` позволяет устанавливать значения по умолчанию, которые всегда можно переопределить для конкретного сегмента кода. Некоторые из полезных умолчаний перечислены далее.

- Для того чтобы можно было печатать кириллицу, например в комментариях, следует определить переменную `extendedchars=true`³.
- Опция `escapechar` позволяет при наборе кода пользоваться услугами L^AT_EX напрямую. Всё, что находится между выбранными символами, обрабатывается средствами L^AT_EX. Естественно, если выбранный символ (в данном случае «|») используется в отображаемом языке, то могут возникнуть проблемы при компиляции. Для того чтобы обнулить `escapechar`, достаточно ничего не писать за знаком равно при переопределении `escapechar` по месту формирования кода.

²Версия пакета **listings** 1.3с поддерживает следующие языки (в скобках указаны диалекты): ABAP, ACSL, Ada (83, 95), Algol (60, 68), Ant, Assembler (x86masm), Awk (gnu, POSIX), bash, Basic (Visual), C (ANSI, Handel, Objective, Sharp), C++ (ANSI, GNU, ISO, Visual), Caml (light, Objective), Clean, Cobol (1974, 1985, ibm), Comal 80, csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran (77, 90, 95), GCL, Gnuplot, Haskell, HTML, IDL (empty, CORBA), inform, Java (empty, AspectJ), JVMIS, ksh, Lisp (empty, Auto), Logo, make (empty, gnu), Mathematica (1.0, 3.0), Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modula-2, MuPAD, NASTRAN, Oberon-2, OCL (decorative, OMG), Octave, Oz, Pascal (Borland6, Standard, XSC), Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S (empty, PLUS), SAS, Scilab, sh, SHELXL, Simula (67, CII, DEC, IBM), SQL, tcl (empty, tk), TeX (AllLaTeX, common, LaTeX, plain, primitive), VBScript, Verilog, VHDL (empty, AMS), VRML (97), XML, XSLT.

³Если это не работает, то необходимо обновить пакет до последней версии или сменить дистрибутив L^AT_EX на более подходящий.

- Инструкция **frame**=<POSITION> позволяет рисовать рамку вокруг сегмента кода. На вход принимаются буквы **t** — обрамление сверху, **b** — снизу, **l** и **r** — слева и справа соответственно. В случае **frame=trbl** будет нарисована простейшая одинарная рамка. Опция **frame**= эквивалентна отказу от обрамления. Если вместо прописных букв указать заглавные **frame=TRBL**, то рамка будет двойная. В пакете есть возможность сделать рамки посложнее.

Все команды, определённые в пакете **listings**, начинаются с префикса **lst**. Команда для включения небольших кусочков кода `\lstinline!код!` аналогична по действию команде `\verb!текст!`.

Сегмент кода оформляется с помощью окружения `lstlisting`:

```
\begin{lstlisting}[language=Perl,
  frame=tlBR,
  caption={Включение сегмента кода}]
# Проверка для перезаписи
if (open(CHECK,"<$file")) {
  $cmd=$term->readline
    ("Overwrite (yes/NO): ");
  if (lc($cmd) ne "yes") {die;}
  close(CHECK);}
\end{lstlisting}
```

Listing 14.2. Включение сегмента кода

```
# Проверка для перезаписи
if (open(CHECK,"<$file")) {
  $cmd=$term->readline
    ("Overwrite_(yes/NO):_");
  if (lc($cmd) ne "yes") {die;}
  close(CHECK);}
```

Необязательный параметр может принять опции, специфичные для оформления этого куска кода. Например, опция **language** позволяет установить язык программирования, отличный от выбранного по умолчанию, **frame** определяет рамку вокруг фрагмента, а **caption** создаёт подпись к фрагменту кода.

Файлы можно включать с помощью команды `\lstinputlisting`:

```
% Установка значений по умолчанию
\lstset{numbers=left , language=MetaPost ,
% Для цветных принтеров
%   backgroundcolor=\color{yellow} ,
%   frame=shadowbox , rulesepcolor=\color{black}}
% Вставка файла
\lstinputlisting[firstline=16, lastline=24,
  emph={forsuffixes , text , bpath} ,
  emphstyle={\bfseries\scshape} ,
  emph={ [2] fill , unfill} ,
  emphstyle={ [2]\bfseries\underbar }]{intro.mp}
```

```
1 vardef drawshadowed(expr dx,dy)(TEXT t) =
2   fixsize(t);
3   FORSUFFIXES s=t:
```

```

4   fill ВРАТН.s shifted (dx,dy);
5   unfill ВРАТН.s;
6   drawboxed(s);
7   %   draw pic(s) withcolor red; %цвет текста
8   endfor;
9   enddef;

```

С помощью опций `firstline` и `secondline` можно указать строки, которые следует вывести. В зависимости от выбора языка форматирование существенно меняется. Инструкция `numbers=left` нумерует строки слева.

Для работы с цветами можно воспользоваться командой `\color`, определённой в одноимённом пакете. Цвета хороши для выделения каких-то ключевых слов и подложки, за которую отвечает опция `backgroundcolor`. Возможности для определения своих «словариков» предоставляет опция `emph=<список ключевых слов>`. В начале списка может идти его метка в квадратных скобках, таким образом можно поддерживать одновременно несколько списков. С помощью `emphstyle` можно определить способ выделения ключевых слов.

Обычно код располагается прямо по месту основного текста, так как обсуждение исходников можно не прерывать в самом коде, благо есть комментарии. Но при желании можно воспользоваться опцией `float`, чтобы из фрагмента кода получился полноценный «плавающий» объект.

Пакет с учётом диалектов поддерживает свыше сотни распространённых языков программирования и разметки. Так что, скорее всего, вам не придётся определять свой язык с помощью инструкции `\lstdefinelanguage`. Но если очень хочется, то и это возможно.

14.2. LCD-дисплей

LCD-дисплеи сейчас встроены даже в кофемолки. Они легко узнаваемы, поэтому нет необходимости копировать их вид в документацию с помощью фотографий — достаточно нарисовать что-то похожее. Изобразить вид дисплея можно с помощью пакета `LATEX lcd`.

```

\definecolor{darkgreen}{rgb}{0.22,0.26,0.19}
\definecolor{lightgreen}{rgb}{0.05,0.97,0.55}
\LCDcolors{darkgreen}{lightgreen}
\centering
\LARGE\textLCD{12}|Linux Format|\\[2mm]
\LCDcolors{white}{darkgreen}
\small\textLCD{12}|Linux Format|

```

Linux Format

Linux Format

Для определения цветов используется макрос `\definecolor` из пакета `color`. Команда `\LCDcolors` формирует цвет букв и фона, а макрос `\textLCD` выводит LCD-подобный текст на экран. `\textLCD` понимает стандартные команды изме-

нения размера шрифта, поэтому его можно использовать совместно с обычным текстом внутри абзаца.

По умолчанию определены только латинские буквы, цифры и некоторые из стандартных символов. Для определения других символов можно воспользоваться макросом `\DefineLCDchar`. Макросу передаётся имя символа и битовая маска, определяющая картинку 5×7 точек. Имя символа может быть однобуквенным, тогда соответствующая буква замещается новым рисунком, или многобуквенным, тогда созданный рисунок кодируется указанным словом в фигурных скобках. Другие размеры матрицы в пакете отсутствуют, но при желании его вполне можно доработать.

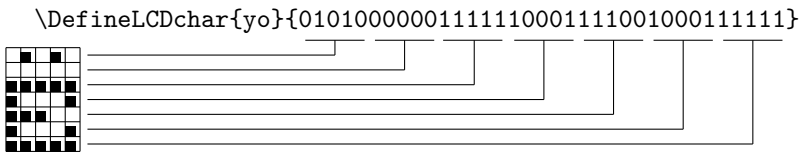
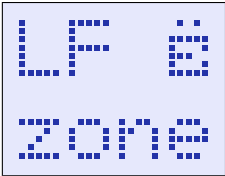


Рис. 14.1. Определяем букву «Ё» для LCD

Для эмуляции дисплея используется команда `\LCD`, в качестве обязательных параметров ей передаётся число строк и число столбцов, за которыми следует содержание строк, разделённых каким-то разделителем. В приведённом примере в качестве разделителя используется вертикальная черта, но вместо неё может быть любой символ.

```
\DefineLCDchar{yo}{01010000001111110001111001000111111}
\definecolor{lightblue}{rgb}{0.9,0.91,0.99}
\definecolor{darkblue}{rgb}{0.14,0.2,0.66}
\LCDcolors{darkblue}{lightblue}
\LCDframe
\setlength{\LCDunitlength}{1.1mm}
\LCD{2}{4}|LF {yo} |
|zone |
```



14.3. Битовые поля

Для описания сетевых протоколов, а также для бинарных форматов данных удобнее всего представить последовательность битов графически, то есть в виде таблицы. Это специализация пакета **bytefield**. В пакете определено одноимённое окружение `bytefield`:

```
\begin{bytefield}{«битовая ширина поля»}
  «битовые поля»
\end{bytefield}
```



Таблица 14.1. Формат пакета UDP

В окружении `bytefield` работают команды `\wordbox` и `\bitbox`, которые формируют поля, занимающие ширину таблицы или только часть её соответственно:

```
\wordbox [ «рамка» ] { «число строк» } { «текст» }
\bitbox [ «рамка» ] { «число занимаемых битов» } { «текст» }
```

Не обязательный параметр «рамка» позволяет сформировать обрамление для текущего битового поля. Значение по умолчанию `[lrtb]` означает, что рамка рисуется со всех сторон поля: `l` — слева, `r` — справа, `t` — сверху и `b` — снизу. Строки разделяются двойной обратной чертой `\\`.

Вот так описывается формат пакета сетевого протокола UDP⁴:

```
\begin{bytefield}{32}
\bitheader{0,15,16,31}\\
\begin{rightwordgroup}{Заголовок}
\bitbox{16}{Порт отправителя}\bitbox{16}{Порт получателя}\\
\bitbox{16}{Размер}\bitbox{16}{Контрольная сумма}
\end{rightwordgroup}\\
\wordbox[lrt]{1}{Данные}\\
\skippedwords\\
\wordbox[lrb]{1}{до 65{.}527 байт}
\end{bytefield}
```

Кроме уже упомянутых команд создания полей при описании формата UDP использовалась команда нумерации столбцов `\bitheader`, конструкция для создания группы `\wordgroup` и макрос `\skippedwords` для формирования «разрыва».

В качестве обязательного аргумента команде `\bitheader` передаётся список нумеруемых битов, при этом можно передавать диапазоны чисел, например, `{0-31}`. В пакете определены два окружения для группировки битовых полей `rightwordgroup`

⁴User Datagram Protocol — это сетевой протокол для передачи данных в сетях IP.

и `leftwordgroup` — отличие этих команд в том, что для первой заголовок группы вводится справа, а для второй — слева. За более подробной информацией следует обратиться к документации пакета.

14.4. Представление алгоритмов

Собственно говоря, именно то, ради чего Д. Э. Кнут и создал `TeX` — комбинация математики и сложного форматирования. Поэтому пакеты для облегчения записи алгоритмов в `LaTeX` были с самого его рождения. На текущий момент число даже стандартных пакетов, попадающих под эту тематику, больше десятка. Здесь рассмотрена только малая их часть.

`algorithms`

Пакет `algorithms` ориентирован на написание алгоритмов, а не на представление кода. Это позволяет отрешиться от форматирования и сосредоточиться на основной задаче. Пакет определяет окружение `algorithmic`. Для использования в преамбуле следует загрузить одноимённый стиль.

```
\begin{algorithmic}[1]
\IF{\(i \leqslant 0\)} \STATE {\(i \gets 1\)} \ELSE
\IF{\(i \geqslant 0\)} \STATE {\(i \gets 0\)}
\COMMENT{смысла в этом алгоритме не ищите}
\ENDIF
\ENDIF
\ENSURE {\(i \geqslant 0\)}
\FORALL{\(x \in \mathcal{A}\)}
\STATE {\(\mathcal{B} \gets x^2\)}
\ENDFOR
\RETURN {\(\mathcal{B}\)}
\end{algorithmic}
```

```
1: if  $i \leq 0$  then
2:    $i \leftarrow 1$ 
3: else
4:   if  $i \geq 0$  then
5:      $i \leftarrow 0$  {смысла
      в этом алгоритме не
      ищите}
6:   end if
7: end if
Ensure:  $i \geq 0$ 
8: for all  $x \in \mathcal{A}$  do
9:    $\mathcal{B} \leftarrow x^2$ 
10: end for
11: return  $\mathcal{B}$ 
```

Если необязательный аргумент определён, то осуществляется нумерация строк. Если аргумент равен 1, то нумеруются все строки, если 2 — то каждая вторая, а далее по индукции.

Команда `\STATE` определяет простое утверждение. Условный оператор представлен командами `\IF{<условие>}`, `\ELSIF{<условие>}`, `\ELSE` и `\ENDIF`.

Циклы представлены операторами `\FOR` и `\FORALL`, которые закрываются командой `\ENDFOR`. Аналогично присутствуют пары `\WHILE{<условие>}` — `\ENDWHILE`, `\REPEAT` — `\UNTILL{<условие>}` и бесконечный цикл `\LOOP` — `\ENDLOOP`. Кроме уже перечисленных конструкций определены предварительное условие для корректного выполнения алгоритма `\REQUIRE`, постусловие,

которое должно выполняться при корректной работе алгоритма, `\ENSURE`, возвращение результата формируется с помощью `\RETURN`, промежуточная печать `\PRINT` и комментарий `\COMMENT`.

Собственно говоря, всё. Псевдокод автоматически разбивается на строки и форматируется в соответствии с общепринятыми представлениям. Очевидно также, что навыки набора математики будут здесь очень кстати. Подробности по настройке пакета следует выяснять в документации к нему: `algorithms.pdf`.

Для того чтобы из объекта `algorithmic` сделать «плавающий объект», можно воспользоваться окружением `algorithm`, для этого следует в преамбуле загрузить одноимённый стиль. Внутри `algorithm` можно использовать команды `\caption` и `\label`.

Клоны algorithms

С помощью имеющихся наработок пакета **algorithms** был создан **algorithmicx**. Данный пакет предоставляет более расширенный набор команд. Кроме этого пользователю предлагаются команды, которые позволяют сформировать свои алгоритмические конструкции. Автор также предоставил вариант форматирования отступов, принятый в Pascal, что позволяет относительно легко переводить программы на этом языке к виду, годному для красивой распечатки. Пакет не совместим с **algorithms**. Решение схожей функциональности предлагает пакет **algorithm2e**. Форматирование C-подобно. Предоставлен избыточный набор конструкций и возможность самостоятельного создания новых структур. Есть зачатки локализации. Пакет не совместим с **algorithms**.

clrscode

Пакет **clrscode** даёт возможность набирать псевдокод, как это делали авторы книги⁵ «Алгоритмы: построение и анализ» Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест и Клиффорд Штайн. Для работы с пакетом необходимо загрузить одноимённый стиль.

```
\begin{codebox}
\Procname{
    $\proc{Сортировка методом вставок}$}
\li \For $j \gets 2$ \To $\id{length}[A]$
\li \Do $\id{key} \gets A[j]$
\li $\i \gets j-1$
\li \While $\i > 0$ and $A[i] > \id{key}$
\li \Do $A[i+1] \gets A[i]$
\li $\i \gets i-1$ \End
\li $A[i+1] \gets \id{key}$ \End
\end{codebox}
```

СОРТИРОВКА МЕТОДОМ ВСТАВОК

```
1 for  $j \leftarrow 2$  to  $length[A]$ 
2   do  $key \leftarrow A[j]$ 
3      $i \leftarrow j - 1$ 
4     while  $i > 0$  and  $A[i] > key$ 
5       do  $A[i + 1] \leftarrow A[i]$ 
6          $i \leftarrow i - 1$ 
7      $A[i + 1] \leftarrow key$ 
```

⁵ *Introduction to algorithms*, Second Edition Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.

pseudocode

Профессор Дональд Л. Крехер (Donald L. Kreher) и профессор Дуглас Р. Стинсон (Douglas R. Stinson) написали книгу «Combinatorial Algorithms: Generation, Enumeration and Search». Специально для этой книги в целях написания псевдокода они создали пакет, который так и назвали: **pseudocode**. Дональд Л. Крехер использовал одноимённое окружение и в своей следующей книге по алгоритмам, выпущенной уже 2005 году. Пакет развивается и поддерживается.

```
\begin{pseudocode}{C2F\_таблица}
      {\text{от}, \text{до}}
\PROCEDURE{C2F}{c}
\COMMENT{Преобразование
  $\circ C$ to $\circ F$}
f \GETS {9c/5} + 32\
\RETURN{f}
\ENDPROCEDURE
\MAIN
x \GETS \text{от} \
\WHILE x \leqslant \text{до} \DO
\BEGIN
\OUTPUT{x, \CALL{C2F}{x}}\
x \GETS x+1
\END
\ENDMAIN
\end{pseudocode}
```

Algorithm 0.1.1: C2F_таблица(от, до)

```
procedure C2F(c)
  comment: Преобразование  $^{\circ}\text{C} \rightarrow ^{\circ}\text{F}$ 
   $f \leftarrow 9c/5 + 32$ 
  return (f)
```

```
main
   $x \leftarrow \text{от}$ 
  while  $x \leq \text{до}$ 
    do  $\begin{cases} \text{output } (x, \text{C2F}(x)) \\ x \leftarrow x + 1 \end{cases}$ 
```

14.5. История изменений

В процессе создания программных продуктов возникает множество сопутствующей документации, в которой в частности описываются подробные спецификации и технические задания.

Как правило, техническое задание меняется в процессе его реализации, так как невозможно предусмотреть всё заранее, поэтому документация имеет привычку также изменяться. Для реконструкции внесённых изменений в документацию в неё добавляется журнал изменений или change log.

Журнал изменений можно отобразить в виде таблицы:

- номер версии;
- время внесения изменений;
- инициалы авторов изменений для их идентификации;
- краткое описание сделанных изменений.

Пример использования пакета vhistory

Евгений М. Балдин

Jochen Wertenauer

Версия 1.2 от 29/09/07

Журнал изменений

Версия	Дата	Автор(ы)	Изменения
1.0	22/01/04	JW	Создание
1.1	21/07/07	JW	Исправление бара с own captions
1.2	29/09/07	JW, EMB	Проверка работоспособности

Рис. 14.2. Пример использования пакета **vhistory**

История изменений формируется внутри окружения `versionhistory`. Каждая строка журнала создаётся с помощью команды `\vhEntry`:

```
\vhEntry{«Версия»}{«Дата»}{«Автор(ы)»}{«резюме»}
```

«Автор(ы)» обозначаются с помощью сокращений, которые допускается делать только латиницей. Разделителем между сокращениями служит символ вертикальной черты «|». Пример из документации к пакету:

```
\vhEntry{1.1}{13.05.04}{JW|AK|KL}{Typos corrected.}
```

Далее следует пример использования **vhistory**, результат действия которого отображается на рис. 14.2.

```
% Препамбула
% Опция own captions необходима для локализация заголовков
\usepackage[own captions]{vhistory}

|\ldots|

% Тело документа
% Расшифровка сокращений для обозначения авторов
\newcommand{\JW}{Jochen Wertenauer}
\newcommand{\EMB}{Евгений М. Балдин}
```

```

{\renewcommand{\setseparator}{\and}}
\title{Пример использования пакета \textbf{vhistory}}
\author{\vhListAllAuthorsLong}
\date{Версия \vhCurrentVersion} от \vhCurrentDate}
\maketitle
}

% Локализация заголовков.
\renewcommand{\vhhistoryname}{Журнал изменений}
\renewcommand{\vhversionname}{Версия}
\renewcommand{\vhdatename}{Дата}
\renewcommand{\vhauthorname}{Автор(ы)}
\renewcommand{\vhchangenamename}{Изменения}

% Создание журнала изменений
\begin{versionhistory}
  \vhEntry{1.0}{22/01/04}{JW}{Создание}
  \vhEntry{1.1}{21/07/07}{JW}{Исправление бага с owncaptions}
  \vhEntry{1.2}{29/09/07}{JW|EMB}{Проверка работоспособности}
\end{versionhistory}

```

Следует обратить внимание на команды `\vhCurrentVersion` и `\vhCurrentDate`. В них хранится последняя версия и дата последнего изменения. Эти данные полезно вынести в колонтитул (см. раздел 12.2).

Команда `\vhListAllAuthorsLong` печатает список авторов, если предварительно расшифровать их акронимы с помощью `\newcommand`.

► Версия пакета, идущая в дистрибутиве **TeX Live 2007**, не позволяет из-за бага включить опцию `owncaptions`, что не позволяет локализовать заголовки. Автор исправил этот баг 21/07/2007 в версии 1.3 и сейчас поправленная версия пакета доступна на CTAN (`{CTAN}/macros/latex/contrib/vhistory.zip`).

Подробности о возможностях пакета **vhistory** можно найти в его документации `vh_sets_en.pdf`.

14.6. Исходники **LaTeX** и контроль версий

LaTeX-исходник тоже представляет собой код. И как всякий код он достоин включения в систему контроля версий. Часто бывает любопытно узнать версию текущего документа и последний момент его обновления. Если в качестве системы контроля версий используется **Subversion** или **svn**, то для начала следует загрузить пакет **svn**⁶.

⁶Если же в вашем проекте используется **cvs** (Concurrent Versions System), то следует воспользоваться пакетами **rcs** или **rscinfo**.

```
\usepackage{svn}
\SVN $Date$
\SVN $Rev$
```

При этом в текст следует добавить метки, предваряемые командой `\SVN`. Для интерполяции меток в системе Subversion при обновлении файла следует выполнить команды вида:

```
> svn propset svn:keywords "Date_Rev" «имя файла»
> svn commit -m "интерполяция_меток"
```

При этом **svn** передаётся информация, какие именно метки требуется обновлять при выполнении **commit**. В данном случае это метки **Date** и **Rev** — дата и версия соответственно. Более подробную информацию можно получить с помощью команды

```
> svn help propset
```

Команда `\SVN $Date$` определяет команды `\SVNDate` и `\SVNTime`, ответственные за календарную дату и время. Все остальные составные команды вида `\SVN $Keyword$`, где **Keyword** — одна из интерполируемых меток **svn**, определяют команды вида: `\SVNKeyword`.

После интерполяции метки будут выглядеть примерно следующим образом:

<code>\SVN \$Date:</code>	2006-11-25 21:02:20 +0600	<code>\$</code>	
<code>\SVN \$Rev:</code>	265	<code>\$</code>	
Документ обновлён	<code>\SVNDate\ \SVNTime</code>		Документ обновлён 25 ноября 2006 г.
			21:02:20
			Текущая версия 265
Текущая версия	<code>\SVNRev</code>		

Схожую функциональность предоставляет пакет **svninfo**.

► К сожалению, в книгах по L^AT_EX редко рассматриваются пакеты, полезные для представления программных текстов или псевдокода. Здесь была предпринята попытка восполнить этот зияющий пробел. Тема настолько обширна, что разрабатывать её можно почти бесконечно. L^AT_EX — сам по себе код, поэтому программистам, по идее, должно быть уютно в его окружении.

Глава 15

Точные науки

Единственное, чему научила меня моя долгая жизнь: что вся наша наука перед лицом реальности выглядит примитивно и по-детски наивно — и всё же это самое ценное, что у нас есть.

Альберт Эйнштейн

L^AT_EX замечательно подходит для издания книг по точным наукам, так как в основе всех точных наук лежит математика — одновременно наука и язык.

15.1. Физика

Физики всего мира знают и общаются на едином языке — языке математики. Иногда встречаются математические структуры, специфичные только для физиков, например, к таковым можно попробовать отнести бра-кет нотацию¹ Дирака:

$$\begin{aligned} & \backslash[\mathbf{p}]\psi(\mathbf{x}) \\ & \stackrel{\mathrm{def}}{=} \langle \mathbf{p} | \psi \rangle = -i \hbar \nabla \psi(x) \end{aligned}$$

Но, по большому счёту, для всего, что нужно физикам, математики имеют и используют самые адекватные способы представления. Так что везде применимы те же правила, что и при создании математических текстов.

¹Команды `\bra` и `\ket` определены в стилевом файле **braket** из коллекции **ltxmisc**. Краткое описание команд можно найти в самом файле **braket.sty**.

15.1.1. Системы единиц

Правила написания обозначений единиц в России регулирует ГОСТ 8.417-2002. В этом своде правил предписывают обязательное использование единиц СИ, что в профессиональной физике не всегда оправдано, но в повседневной жизни предпочтительно. Также там перечислены единицы физических величин, разрешённые к применению, приведены их международные и русские обозначения, а также установлены правила их использования.

- Обозначения единиц печатают прямым шрифтом, точку как знак сокращения после обозначения не ставят. Поэтому в формуле всегда следует экранировать обозначения единиц с помощью команды `\text` (см. раздел 4.2).
- Обозначения помещают за числовыми значениями величин через пробел, перенос на другую строку не допускается. Исключения составляют обозначения в виде знака над строкой, перед ними пробел не ставится. Примеры: `\(25\,\text{m}/\text{c}\)` (25 м/с), `\(36\{,}6^\circ\)` (36,6°).
- Если числовое значение представляет собой дробь с косой чертой, его заключают в скобки, например: `\((1/60)\,\text{c}^{-1}\)`, что даёт $(1/60) \text{ c}^{-1}$.
- При указании значений величин с предельными отклонениями их заключают в скобки `\((100.0\text{ pm } 0.1)\,\text{kg}\)`, что даёт $(100.0 \pm 0.1) \text{ кг}$ или проставляют обозначение единицы за числовым значением величины и за её предельным отклонением: `\(50\,\text{g} \pm 1\text{ g}\)` (50 г ± 1 г).
- Обозначения единиц, входящие в произведение, отделяют точками на средней линии: `\(\text{H}\cdot\text{m}\)` (Н · м).
- В качестве знака деления в обозначениях можно использовать горизонтальную черту или косую черту (только одну). При применении косой черты, если в знаменателе стоит произведение единиц, его заключают в скобки. Правильно: Вт/(м · К), неправильно: Вт/м/К, Вт/м · К.
- Допускается применять обозначения единиц в виде произведения обозначений единиц, возведённых в степени (положительные и отрицательные): Вт · м⁻² · К⁻¹, А · м². При использовании отрицательных степеней не разрешается использовать горизонтальную или косую черту (знак деления).
- Допускается применять сочетания специальных знаков с буквенными обозначениями, например: °/с (градус в секунду).
- Не допускается комбинировать обозначения и полные наименования единиц. Неправильно: км/час, правильно: км/ч.

SIstyle

Если в случае кириллицы приходится фактически руками формировать вывод размерных единиц, то для англоязычного сообщества эта проблема частично автоматизирована. Обязанность оформления берёт на себя пакет **SIstyle**, для использования которого следует загрузить стилевой файл **sistyle.sty**. В пакете определены всего три команды: `\SI` — формирование размерных единиц по правилам СИ, `\num` — вывод чисел (можно определить локаль) и `\ang` — формирование угловых единиц.

`\SI{\m.kg/(\text{kr}^3.A)}\quad`

`\SI{\{(MPa)^{0\pnt 5}}\quad`

`\SI{\{(MPa)^{\num{0.5}}\}\par`

`$v=\SI{10}{m.s^{-1}}$\quad`

`$v=\SI{10}{m/s} $\quad`

`$v=\SI{10}{\tfrac{m}{s}} $\par`

`\ang{45}\quad \ang{10; 12; 4}`

$$\begin{array}{lll} \text{m.kg}/(\text{kr}^3\cdot\text{A}) & (\text{MPa})^{0.5} & (\text{MPa})^{0.5} \\ v = 10 \text{ m}\cdot\text{s}^{-1} & v = 10 \text{ m/s} & v = 10 \frac{\text{m}}{\text{s}} \\ 45^\circ & 10^\circ 12' 4'' \end{array}$$

При использовании кириллического текста его необходимо экранировать с помощью команды `\text`, то есть в этом смысле пакет требует доработки.

Документация к пакету **SIstyle-2.3.pdf** (вместо 2.3 может быть другая версия) чрезвычайно подробная и касается не только вопросов использования, но самих правил представления единиц СИ.

► Схожую функциональность обеспечивает пакет **SIunits** (**SIunits.pdf**). Пакет **unitsdef** (**unitsdef.pdf**) предоставляет обширный набор команд для набора физических единиц. Также при написании своей системы представления единиц может заинтересовать пакет **units** (**units.pdf**).

15.1.2. Физика высоких энергий

Традиционно в физике высоких энергий для написания статей используют $\LaTeX$. Считается хорошим тоном выложить свою статью перед публикаций, да и просто без всякой публикации, в журнал на <http://arxiv.org>².

hepparticles

Правила формирования имён элементарных частиц довольно простые, но ситуация осложняется тем, что кроме действительно элементарных частиц (кварков, лептонов и нейтрино) существует множество адронов как стабильных, так и резонансов, а также сугубо, возможно пока, гипотетических частиц. Пакет

²Самое большое хранилище электронных публикаций в открытом доступе, содержащее на 02/02/2008 461,639 статей по физике, математике, компьютерным наукам, биологии и статистике. Для помещения публикации в архив настоятельно рекомендуется использовать $\LaTeX$ -исходники. В этом случае в частности автоматически учитываются библиографические ссылки в статье. Допускается, но не рекомендуется из-за отсутствия доступа к исходному тексту размещение электронной публикации в виде pdf- или PostScript-файла.

hepparticles предоставляет набор макросов, позволяющий упорядочить создание имён:

```
\begin{itemize}
\item Элементарные частицы:
\HepParticle{B}{d}{0},
\HepAntiParticle{B}{d}{0}
\item Действительно элементарные частицы:
\HepGenParticle{q}{d}{-},
\HepGenAntiParticle{ell}{mu}{-}
\item SUSY=частицы:
\HepSusyParticle{chi}{1}{-},
\HepGenSusyParticle{q}{2}{-}
\item Резонансы:
\HepParticleResonance{J\!/psi}{1S}{-}{\star}
\item Процессы:
\HepProcess{\HepParticle{B}{d}{0} \HepTo
\HepParticle{K}{-}{-}\HepParticle{pi}{+}{+}}.
\end{itemize}
```

- Элементарные частицы: $B_d^0, \bar{B}_d^0$
- Действительно элементарные частицы: $q_d, \bar{\ell}_\mu$
- SUSY-частицы: $\tilde{\chi}_1, \tilde{\varphi}_2$
- Резонансы: $J/\psi(1S)^*$
- Процессы: $B_d^0 \rightarrow K^- \pi^+$.

Кроме макросов, позволяющих сконструировать названия частиц, пакет предоставляет команду для записи формул процессов `\HepProcess` и немного удлинённую по сравнению со стандартной стрелку `\HepTo`. Подробную документацию следует искать в файле `hepparticles.pdf`.

Фейнмановские диаграммы

Говорят, что Фейнману активно не нравилось то, что пингвинообразные диаграммы не похожи на пингвинов, но он вынужден был смириться с этим фактом, так как диаграммы Фейнмана самого Фейнмана тоже никак не напоминают.

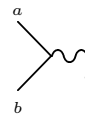
Для создания относительно простых диаграмм можно воспользоваться пакетом **feyn**. Правила создания даже простых диаграмм достаточно сложны, но в документации пакета `feyn.pdf` присутствует множество внятных примеров.

Пример использования пакета **feyn**:

Простая диаграмма:

```
\(\Feyn{fs f glu f fs}\)
и чуть сложнее:
\[\Diagram{\vertexlabel^a \
fd \
& g\vertexlabel_{\mu,c} \
\vertexlabel_b fu\
} = ig\gamma_{\mu} (T^c)_{ab}\]
```

Простая диаграмма:  и чуть сложнее:



$$= ig\gamma_\mu (T^c)_{ab}$$

Для создания действительно сложных диаграмм традиционно используется стилевой файл **feynmp** из пакета **feynmf**. Этот пакет был представлен в 1995 г. Торстеном Охлом (Torsten Ohl), и ему уже более десяти лет, но он (пакет) совершенно адекватно справляется со своей работой. Сам **LaTeX** не обладает необхо-

димой функциональностью для столь специфичной задачи, поэтому при формировании диаграмм используется мощь MetaPost³.

Например, интересующий меня с целью извлечения $\Gamma_{e^+e^-} \times \text{Br}(J/\psi \rightarrow e^+e^-)$ процесс в первом приближении имеет следующий вид:

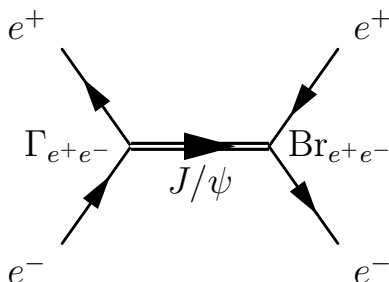


Рис. 15.1. Простейшая фейнмановская диаграмма (пакет **feynmp**)

Это не сложная диаграмма, и если не считать метки, то для её описания требуется всего пять операторов. Пусть следующий код набран в файле `eeepsiee.tex`:

```
% Файл eeepsiee.tex
\usepackage{feynmp} % В преамбуле
...
\begin{fmffile}{ee-psi-ee} % Имя mp-файла
\begin{fmfgraph*}(110,62) % Размер диаграммы
  \fmfleft{ei,pi} % Что было (источники)
  \fmfright{eo,po} % Что стало (исходящие)
  \fmflabel{$e^-$}{ei} % Метка источника  $e^-$ 
  \fmflabel{$e^+$}{pi} % Метка источника  $e^+$ 
  \fmflabel{$e^+$}{po} % Метка исходящей вершины
  \fmflabel{$e^-$}{eo} % Метка исходящей вершины
  % Линия, соединяющая источники
  \fmf{fermion}{ei, Ji, pi}
  % Линия, соединяющие исходящие вершины
  \fmf{fermion}{po, Jo, eo}
  % Метка для начальной вершины промежуточной частицы
  \fmflabel{$\Gamma_{ee}$}{Ji}
  % Метка для конечной вершины промежуточной частицы
  \fmflabel{$Br_{ee}$}{Jo}
  % Соединительная линия между источниками и исходящими
  \fmf{heavy, label=$J/\psi$}{Ji, Jo}
```

³В серии статей для Linux Format я описал, как можно использовать MetaPost для создания иллюстраций. Эти тексты сейчас доступны под открытой лицензией CC-BY-SA 3.0 на страничке <http://www.inp.nsk.su/~baldin/mpost/>.

```
\end{fmfgraph*}
\end{fmffile}
```

В коде используются символные метки вида `ei` и `Jo` для обозначения точек и вершин. Местоположение нефиксированных вершин выбирается автоматически.

Окружение `fmffile` в качестве параметра требует имя `mp`-файла, в который будут записаны команды языка `META`. В обсуждаемом примере имя файла определено как `ee-psi-ee.mp`. Для того чтобы получить диаграмму, описанную в файле `eepsiee.tex`, были проделаны следующие действия:

```
# Создание ee-psi-ee.mp
> latex eepsiee.tex
# Создание eps-файла из ee-psi-ee.mp
> mpost ee-psi-ee.mp
# Сборка итогового документа
> latex eepsiee.tex
```

mpost — это компилятор MetaPost. Обычно **mpost** присутствует в любом дистрибутиве `ЛATEX`. После выполнения этих команд результат можно посмотреть с помощью программы `xdvi` или преобразовать `dvi`-файл в `PostScript` или `pdf`.

Исчерпывающую документацию к пакету можно обнаружить в директории, где лежит документация к **feynmf** в файле `manual.ps.gz`.

isotope

Ядерная физика — это, конечно, не физика высоких энергий, но она всегда где-то рядом. В пакете **isotope** (документацию можно извлечь из исходников пакета `isotope.dtx`) определена одна команда `\isotope`, позволяющая отображать имена изотопов химических элементов:

<code>\isotope{Ra}\quad \isotope[228]{Ra}\quad</code>	$\text{Ra} \quad {}^{228}\text{Ra} \quad {}^{228}_{88}\text{Ra}$
<code>\isotope[228][88]{Ra}</code>	
<code>\[\isotope[A][Z]{X}\to</code>	${}_Z^AX \rightarrow {}_{Z-2}^{A-4}Y + {}_2^4\alpha$
<code>\isotope[A-4][Z-2]{Y}+\isotope[4][2]{\alpha}\]</code>	

15.1.3. Электронные схемы

Вообще-то не дело текстовому процессору рисовать электронные схемы, даже если они простые, но `ЛATEX` может и это.

Пакет **circ** по сути представляет собой обвязку над специальными шрифтами, позволяющими изобразить простые элементы вида резисторов, конденсаторов, транзисторов и тому подобное. Ничего кроме `ЛATEX` для создания электронных схем с помощью **circ** не требуется. Точнее ничего, кроме умения писать программы. В преамбуле следует загрузить стиль **circ**.

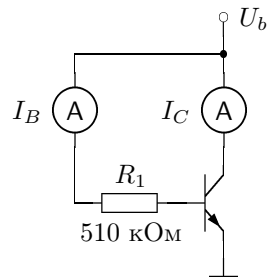
```
\usepackage[ basic , box , gate , oldgate , ic , optics , physics ]{ circ }
```

Опции соответствуют загружаемым наборам символов:

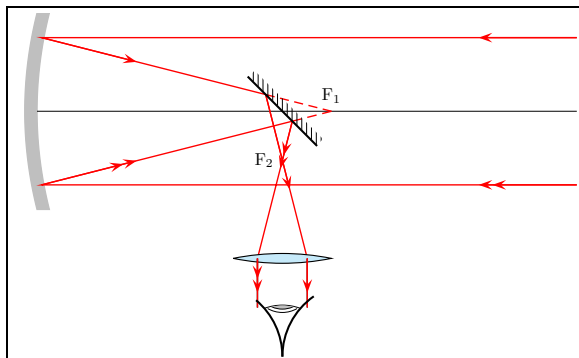
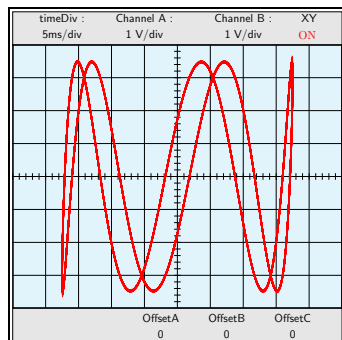
- basic — набор стандартных символов: резисторы, конденсаторы, переключатели, диоды, транзисторы и тому подобное;
- box — символы «чёрного ящика», осциллографа, генератора и усилителя;
- gate — набор логических схем;
- oldgate — набор логических схем, выполненных в «старом стиле»;
- ic — интегрированные логические схемы (триггеры и так далее);
- optics — оптика;
- physics — чуть-чуть механики.

Схема описывается внутри окружения `circuit`. Сразу после `\begin{circuit}` следует указать размер символов с помощью цифры от 0 до 4. Самый маленький размер соответствует 0. При создании презентаций размер символов полезно увеличить.

```
\begin{circuit}0
\npn1 {?} B 1 % Транзистор.
\frompin npn1C % Рисуем от коллектора
\ - 1 u % проводок вверх.
\nl\A1 {$I_C$} u % Амперметр, измеряющий ток коллектора A1.
\atpin npn1B % Рисуем от базы транзистора
\ - 1 l % проводок влево.
\R1 {510 кОм} 1 % Сопротивление R1.
\ - 1 l % Кусочек провода влево.
\centerto A1 % Выравниваем элемент по A1.
\nl\A2 {$I_B$} u % Второй амперметр A2.
\frompin A2b % Рисуем линию от A2 (снизу)
\vtopin R1l % до сопротивления R1 (слева).
\frompin A1t % Рисуем линию от A2 (сверху)
\ - 1 u % вверх.
\ .1 % Создаём соединение (жирная точка).
\frompin A2t % Проводок от A2
\vtopin .1 % до соединения.
\htopin .1 % От соединения
\ - 1 u % проводок вверх для подвода
\cc\connection1 {$U_b$} с u % напряжения.
\frompin npn1E % Рисуем от эмиттера
\ - 1 d % проводок вниз
\GND1 % и заземляем его.
\end{circuit}
```



Подробная инструкция представлена в файле `circ.dvi`. Там же можно найти пример для простой оптической демонстрации лабораторной работы по физике.

Рис. 15.2. Схема телескопа (пакет **pst-optic**)Рис. 15.3. Фигура Лиссажу (пакет **pst-osci**)

Немного по-другому, но по сути примерно так же работает пакет **pst-circ**. Документация **pst-circ-doc.pdf** к тому пакету замечательно оформлена и легко читается.

Для создания рисунков с элементами электронных схем может также пригодится пакет для MetaPost **makecirc**. Все подробности изложены в документации к пакету **MakeCirc-en.pdf**

15.1.4. Лабораторные работы

Классикой для лабораторных работ по физике являются работы, связанные с оптикой. С описанием схемы установки вполне может справиться упомянутый ранее пакет **circ**, но сама геометрическая оптика наиболее полно отражена в макросах **PSTricks**, собранных в пакете **pst-optic** (документация **pst-optic.pdf**, примеры **pst-optic-examples.pdf**). Например, рис. 15.2 был получен с помощью команды:

```
\telescope [ mirrorFocus=6,posMirrorTwo=5,yBottom=-5]
```

Не менее интересным представляется пакет **pst-osci**. На рис. 15.3 представлен результат выполнения макроса:

```
\Oscillo [ amplitude1=3.5,phase1=90,amplitude2=3.5,
            period1=25,period2=5,phase2=60,Lissajous=true ]
```

Те, кому хоть раз в жизни приходилось оформлять лабораторные работы с использованием осциллографа, по достоинству оценят этот пакет. Документация, как обычно, в файле **pst-osci.pdf**

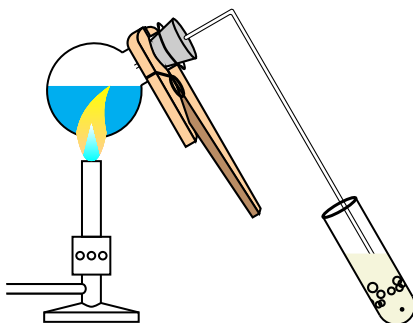
15.2. Химия

Великий Л. Д. Ландау говорил: «Всё, что в химии научного — это физика, а остальное — кухня». Естественно, он был не прав, но то, что химики должны владеть и математикой, и физикой — это так.

15.2.1. Лабораторные работы

При взгляде со стороны создаётся впечатление, что студенты-химики только и делают, что лабораторные работы. Оформление внятных методических материалов для этих занятий — весьма не простой вид деятельности. Набор макросов **PSTricks**, собранный в пакете **pst-labo**, аналогичный наборам из раздела 15.1.4, может значительно помочь в этом благородном деле.

```
\psset{glassType=ballon,
      unit=0.5cm,
      becBunsen,
      pince=true}
\pstChauffageTube[becBunsen,
                  barbotage]
```



Информацию о том, как рисовать колбы, горелки, различные цвета и консистенцию для смесей, трубки и многое другое, можно найти в файле документации **pst-labo-docEN.pdf**.

15.2.2. Химические формулы

Первое, с чем сталкивается интересующийся при знакомстве с химией, — это огромное количество названий и сокращений для элементов и их соединений. Даже физика высоких энергий с её зоопарком элементарных частиц не идёт с этим разнообразием ни в какое сравнение.

Для эпизодического набора химических формул достаточно стандартного математического окружения ЛАТ_ЭХ. Нужно только установить прямое начертание для символов:

```
\(\frac{1}{2}\mathrm{H}_2\mathrm{O}\) или \(\frac{1}{2}\mathrm{H}_2\mathrm{O}\) или 3\mathrm{Cr}_2\mathrm{O}_7^{2-}
```

Не очень удобно, но вполне терпимо в случае небольших фрагментов.

mhchem

Пакет **mhchem** состоит из двух стилевых файлов: **mhchem.sty** и **rsphrase.sty**. В стиле **mhchem** определены команды `\ce` и `\bond`, предназначенные для создания химических молекулярных формул и выражений. Инициализация пакета:

```
\usepackage[version=3]{mhchem}
```

Номер версии необходим для гарантии обратной совместимости. Так как автор уже как минимум дважды серьёзно изменял действие своих команд, то нет никакой гарантии, что он не сделает это в будущем.

Набор молекулярных формул производится внутри команды `\ce` более-менее естественным для химиков способом:

Простые соединения			
H_2O	<code>\ce{H2O}</code>	Sb_2O_3	<code>\ce{Sb2O3}</code>
H^+	<code>\ce{H+}</code>	CrO_4^{2-}	<code>\ce{CrO4^2-}</code>
AgCl_2^-	<code>\ce{AgCl2-}</code>	$[\text{AgCl}_2]^-$	<code>\ce{[AgCl2]-}</code>
Y^{99+}	<code>\ce{Y^{99}+}</code>	Y^{99+}	<code>\ce{Y^{99+}}</code>
$\text{H}_{2(\text{aq})}$	<code>\ce{H2_{(aq)}}</code>	NO_3^-	<code>\ce{NO3-}</code>
$(\text{NH}_4)_2\text{S}$	<code>\ce{(NH4)2S}</code>		
Спецсимволы			
$\text{KCr}(\text{SO}_4)_2 \cdot 12 \text{H}_2\text{O}$	<code>\ce{KCr(SO4)2*12H2O}</code>		
$\text{KCr}(\text{SO}_4)_2 \cdot 12 \text{H}_2\text{O}$	<code>\ce{KCr(SO4)2.12H2O}</code>		
RNO_2^-	<code>\ce{RNO2^{-}}</code>		
$\mu\text{-Cl}$	<code>\ce{\$\mu\$hyphen\$Cl}</code>		
Множители			
$2 \text{H}_2\text{O}$	<code>\ce{2H2O}</code>	$\frac{1}{2} \text{H}_2\text{O}$	<code>\ce{1/2H2O}</code>
Изотопы			
${}^{227}_{90}\text{Th}^+$	<code>\ce{^{227}_{90}Th+}</code>		
Связи			
$\text{C}_6\text{H}_5\text{—CHO}$	<code>\ce{C6H5-CHO}</code>	$\text{X=Y}\equiv\text{Z}$	<code>\ce{X=Y#Z}</code>

Связи внутри химической формулы можно указать и другими способами с помощью макроса `\bond`:

<code>\ce{A\sbond B\dbond C\tbond D}\</code>	$\text{A—B=C}\equiv\text{D}$
<code>\ce{A\bond{-}B\bond{=}C\bond{#}D}\</code>	$\text{A—B=C}\equiv\text{D}$
<code>\ce{A\bond{~}B\bond{~}C}\</code>	$\text{A}\cdots\text{B}=\text{C}$
<code>\ce{A\bond{~}B\bond{~-}C\bond{~-}D}\</code>	$\text{A}\equiv\text{B}\equiv\text{C}\equiv\text{D}$
<code>\ce{A\bond{...}B\bond{...}C}\</code>	$\text{A}\cdots\text{B}\cdots\text{C}$
<code>\ce{A\bond{->}B\bond{<-}C}</code>	$\text{A}\rightarrow\text{B}\leftarrow\text{C}$

► Используемый для создания прерывистой линии макрос приводит к некорректному отображению символов при просмотре dvi-файла.

Внутри команды `\ce` допускается использование математики, а также самой команды `\ce`. Дополнительно внутри неё определены простые естественные сокращения для стрелок, используемых для написания химических уравнений.

<code>\centering</code>	$\text{CO}_2 + \text{C} \longrightarrow 2 \text{CO}$
<code>\ce{CO2 + C -> 2CO}\</code>	$\text{CO}_2 + \text{C} \longleftarrow 2 \text{CO}$
<code>\ce{CO2 + C <- 2CO}\</code>	$\text{CO}_2 + \text{C} \rightleftharpoons 2 \text{CO}$
<code>\ce{CO2 + C <=> 2CO}\</code>	$\text{H}^+ + \text{OH}^- \rightleftharpoons \text{H}_2\text{O}$
<code>\ce{H+ + OH- <=> H2O}\</code>	$\text{A} \longleftrightarrow \text{A}'$
<code>\ce{\\$A\\$ <-> \\$A'\\$}\</code>	$\text{CO}_2 + \text{C} \xrightarrow{\alpha} 2 \text{CO}$
<code>\ce{CO2 + C ->[\alpha] 2CO}\</code>	$\text{CO}_2 + \text{C} \xrightarrow[\beta]{} 2 \text{CO}$
<code>\ce{CO2 + C ->[\alpha][\beta] 2CO}\</code>	$\text{A} \xrightarrow{+\text{H}_2\text{O}} \text{B}$
<code>\ce{\\$A\\$ ->[\ce{+H2O}] \\$B\\$}</code>	

Стиль **rsphrase** содержит маркировку и расшифровку (Risk and Safety Phrases) для опасных химических соединений на разных языках. Русский среди них отсутствует, но локализация возможна. Документация находится в файле `mhchem.pdf`.

15.2.3. Структурные формулы

Одна из самых первых попыток расширить возможности $\text{T}_{\text{E}}\text{X}$ за пределы разбиения строк на хорошие абзацы была связана именно с изображением структурных формул. Так появился **chemtex**. Но это было давно и на смену ему пришёл **xymtex**.

xymtex — пакет для отображения средствами $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ множества химических структурных формул. Качество получающихся формул по отзывам очень высокое и кое в чём превосходит традиционные коммерческие аналоги. Другое дело, что набор структурных химических формул достаточно сложен. Возможно, создание промежуточной GUI-программы, которая использовала бы этот пакет, было бы оптимальным решением данной проблемы. К сожалению, **xymtex** отсутствует в дистрибутиве $\text{T}_{\text{E}}\text{X}$ Live и последнюю версию пакета следует брать по адресу <http://homepage3.nifty.com/xymtex/fujitas3/xymtex/indexe.html>.

Со странички автора следует скачать архив `xymtex403b.lzh`. Для его распаковки потребуется популярный в Японии (автор родом из Японии) архиватор **lha**. После распаковки директорию **xymtex** следует скопировать в локальную **texmf**-директорию и регенерировать индексные файлы:

```
# Распаковка архива
> lha x xymtex403b.lzh
# Копирование директории xymtex
> cp -r xymtex «texmf-local»
# Обновление индексов
> texhash
```

Теперь можно использовать пакет по прямому назначению. Документацию `xymtx402403.pdf` можно найти в поддиректории `doc402403`. Она написана несколько тяжеловато, и способ задания формул может показаться непривычным:

```
%\usepackage{xymtexp} % Преамбула
%\usepackage{chmst-ps}
% Тело документа
% Формула Адонитоксина
\begin{XyMcompd}(2100,1800)(200,0){}{
  \pyranose{1Sa==H;2Sb==H;2Sa==OH;3Sb==H;3Sa==OH;
    4Sb==HO;4Sa==H;5Sb==H;5Sa==CH$_{3}$;%
    1Sb==%
  \ryl(8==O){3==%
    \steroid{3==(y1);5A==H;8B==H;9A==H;%
      {{10}B}==\lmoiety{OHC};{{14}A}==OH;%
      {{13}B}==\lmoiety{H$_{3}$C};{{16}B}==OH;%
      {{17}B}==\fiveheterov[e]{3==O}{4D==O;1==(y1)}
    }
  }
}
\end{XyMcompd}
```

Зато в результате получается, примерно, следующее:

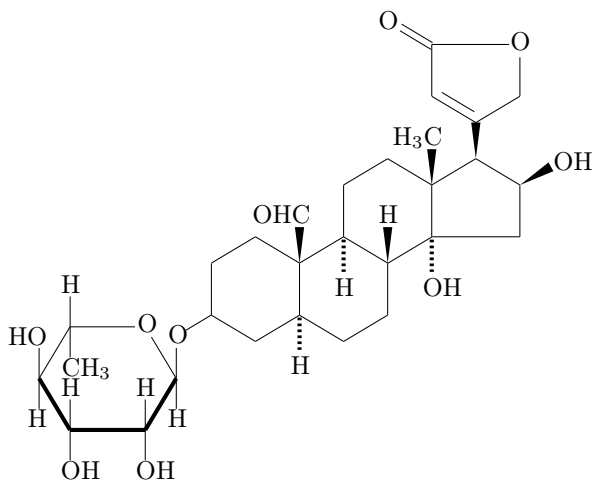
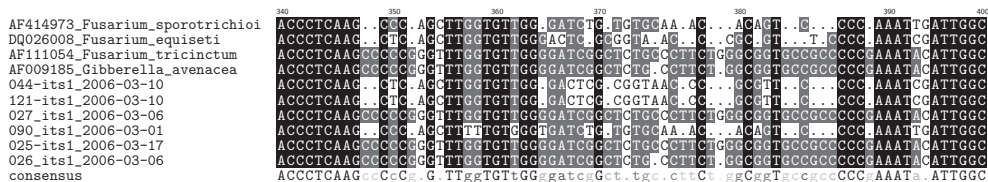


Рис. 15.4. Структурная формула адонитоксина

► Надстройка над этим пакетом в виде GUI-программы была бы весьма кстати.

15.3. Биология

Биологу приходится пользоваться математикой, разбираться в физике и знать химию. Возможно, поэтому специализированных биологических пакетов $\text{\LaTeX}$ относительно немного. Хотя среди них встречаются весьма изощрённые, например, пакет **texshade**, предназначенный для выделения одинаковых последовательностей нуклеотидов и пептидов.



```

AF414973_Fusarium_sporotrichoi ACCCTCAAG...CCC.ACC.TTGGTGTGG.GATCTC.TGCTGCAA.AC...ACAGT...CCC.AAATGATTGGC
DQ026008_Fusarium_equiseti ACCCTCAAG...ACTC.ACC.TTGGTGTGGGACTC.GGGGA.A.AC...CCGCGT...T.CCCC.AAATCGATTGGC
AF111054_Fusarium_tricinctum ACCCTCAAGCCCGGGTTTGGTGTGGGGATCGGGTCTGCCCTTCGGCGGTGCCGCCGCCGAAATTCATTGGC
AF009185_Gibberella_avenacea ACCCTCAAGCCCGGGTTTGGTGTGGGGATCGGGTCTGCCCTTCGGCGGTGCCGCCGCCGAAATTCATTGGC
044-its1_2006-03-10 ACCCTCAAG...CTC.ACC.TTGGTGTGGGACTCC.CGGTAA.GC...CCCTT...CCC.AAATCGATTGGC
121-its1_2006-03-10 ACCCTCAAG...CTC.ACC.TTGGTGTGGGACTCC.CGGTAA.GC...CCCTT...CCC.AAATCGATTGGC
027-its1_2006-03-06 ACCCTCAAGCCCGGGTTTGGTGTGGGGATCGGGTCTGCCCTTCGGCGGTGCCGCCGCCGAAATTCATTGGC
090-its1_2006-03-01 ACCCTCAAG...CCC.ACC.TTTTGTGGTGGATCTGCTGCTGCAA.AC...ACAGT...CCC.AAATGATTGGC
025-its1_2006-03-17 ACCCTCAAGCCCGGGTTTGGTGTGGGGATCGGGTCTGCCCTTCGGCGGTGCCGCCGCCGAAATTCATTGGC
026-its1_2006-03-06 ACCCTCAAGCCCGGGTTTGGTGTGGGGATCGGGTCTGCCCTTCGGCGGTGCCGCCGCCGAAATTCATTGGC
consensus ACCCTCAAG...CcC.G.TTggTGtTGGgGatcGgGt.tgc.cttCt.ggCggTg-cgcccCCCgAAATa.ATTGGC

```

Рис. 15.5. Выравнивание (alignment) генов рибосомальных РНК нескольких видов грибов из рода *Fusarium*/*Gibberella* с помощью пакета **texshade**

Пример с **texshade** был предоставлен Алексеем Б. Шипуновым. По WWW-адресу <http://herba.msu.ru/shipunov/software/tex/tex-ru.htm> можно найти его страничку « $\text{\LaTeX}$ для ботаников и зоологов».

Глава 16

Гуманитарные направления

Музыка есть таинственная арифметика души; она вычисляет, сама того не сознавая.

Готфрид Лейбниц

Гуманитарные дисциплины отнюдь не меньше точных нуждаются в качественном инструменте для создания книг. Та же нотная нотация по своей изощрённости вполне сравнивается с математической, а аппарат для создания комментариев к текстам архисложен в своём исполнении, если его не автоматизировать.

16.1. Стихи и пьесы

Поэзия, наверное, всегда сопровождала человечество. Сложно понять, зачем это нужно, но людям нравится.

Небольшие стихотворные фрагменты можно оформить с помощью окружения `verse`:

```
\begin{verse}
```

```
Пусть от бед тебя хранит\\*  
Сгрызенный тобой гранит.
```

```
\emph{Студенту в помощь}  
\end{verse}
```

```
Пусть от бед тебя хранит  
Сгрызенный тобой гранит.
```

```
Студенту в помощь
```

Строфы разделяются пустыми строками, а сами бьются на строки с помощью команды переноса строки `\\`. Если после этой команды поставить звёздочку (`\\*`), то это будет эквивалентно запрету на разрыв строфы в этом месте при переносе на новую страницу.

verse

Пакет **verse** значительно расширяет возможности одноимённого окружения. В документации **verse.pdf** кроме самого пакета описаны и простые альтернативы и приведены очень подробные примеры формирования стихотворных фраз.

```
\renewcommand{\poemtoc}{subsection}
\poemtitle{Анчар}
\settowidth{\versewidth}{Стоит "---
    один во всей вселенной.}
```

```
\begin{verse}[\versewidth]
В пустыне чахлой и скупой,\,
На почве, зноем раскаленной,\,
Анчар, как грозный часовой,\,
Стоит "--- один во всей вселенной.
```

```
\vin Природа жаждущих степей\,
\vin Его в день гнева породила\,
\vin И зелень мертвую ветвей,\,
\vin И корни ядом напоила.
```

```
Яд каплет сквозь его кору,\,
К полудню растопясь от зною,\,
И застывает ввечеру\,
Густой, прозрачною смолою.\,
```

```
\ldots
```

```
\emph{A.\,C. Пушкин}
\end{verse}
```

Команда `\poemtoc` определяет уровень, которому будет соответствовать название стихотворения в оглавлении. `\poemtitle` формирует название стихотворения. `\versewidth` — просто предопределённая переменная, в которой можно хранить длину строки. Окружение `verse` принимает в качестве необязательного параметра примерную типичную длину строки. Используя информацию о средней длине строки, окружение пытается максимально симметрично расположить стихотворение на странице. Команда `\vin` формирует отступ.

В пакете также определены команды для автоматического форматирования строф, оформления заголовков и формирования изошрённых форм у стихотворения.

stage

Для написания пьес, можно воспользоваться, например, классом **stage**. При этом стилевой файл **stage.sty** лучше скопировать себе в рабочую директорию и ки-

Анчар

В пустыне чахлой и скупой,
На почве, зноем раскаленной,
Анчар, как грозный часовой,
Стоит — один во всей вселенной.

Природа жаждущих степей
Его в день гнева породила
И зелень мертвую ветвей,
И корни ядом напоила.

Яд каплет сквозь его кору,
К полудню растопясь от зною,
И застывает ввечеру
Густой, прозрачною смолою.

...

А. С. Пушкин

Зилов	Главный герой
Мальчик	Просто мальчик
Акт I	
(Pause.)	
SCENE 1	
(В дверях появляется венок. Это большой, дешёвый, с крупными бумажными цветами и длинной чёрной лентой сосновый венок. Вслед за ним появляется несущий его мальчик лет двенадцати. Он всерьёз озабочен исполнением возложенной на него миссии.)	
Мальчик	
Здравствуйте. Скажите, вы Зилов?	
Зилов	
Ну я.	
Мальчик	
(поставил венок у стены). Вам.	
Зилов	
Мне?... Зачем?	
(Мальчик молчит.)	
THE END	

Рис. 16.1. Простой класс **stage**

рилизовать его (заменить английские названия разделов на русские), так как по умолчанию предполагается английский язык.

Пример использования класса **stage**:

```
% Класс stage
\documentclass{stage}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
\usepackage[indentfirst]{%first paragraph indent}

\author{Александр Вампилов.}
\title{Утиная охота}
\begin{document}
```

```
% Описание действующих лиц
\addcharacter{Зилов}{Главный герой}
\addcharacter{Мальчик}{Просто мальчик}
% Действия
\act
% Пауза
\pause
% Картина
\scene

% Введение
\opensd{В дверях появляется венок. Это большой, дешёвый,
  с крупными бумажными цветами и длинной чёрной лентой
  сосновый венок. Вслед за ним появляется несущий его
  мальчик лет двенадцати. Он всерьёз озабочен исполнением
  возложенной на него миссии.}

% Диалоги
\dialog{Мальчик}{Здравствуйте. Скажите, вы Зилов?}

\dialog{Зилов}{Ну я.}

\dialog{Мальчик}{(поставил венок у стены). Вам.}

\dialog{Зилов}{Мне?.. Зачем?}

\stage{Мальчик молчит.}

\end{document}
```

Подробности можно найти в документации к пакету `stage-documentation.pdf`.

16.2. Музыка

Отображение музыки с помощью нот или других знаков в каком-то смысле по сложности и своей логической стройности сравнимо с математической нотацией. Очень жаль, что использование $\text{\LaTeX}$ в качестве нотного процессора не так популярно, ведь всё для этого есть.

guitar

Для записи текста песен и гитарных аккордов много не требуется. Стиливой файл **guitar** это немного и предоставляет.

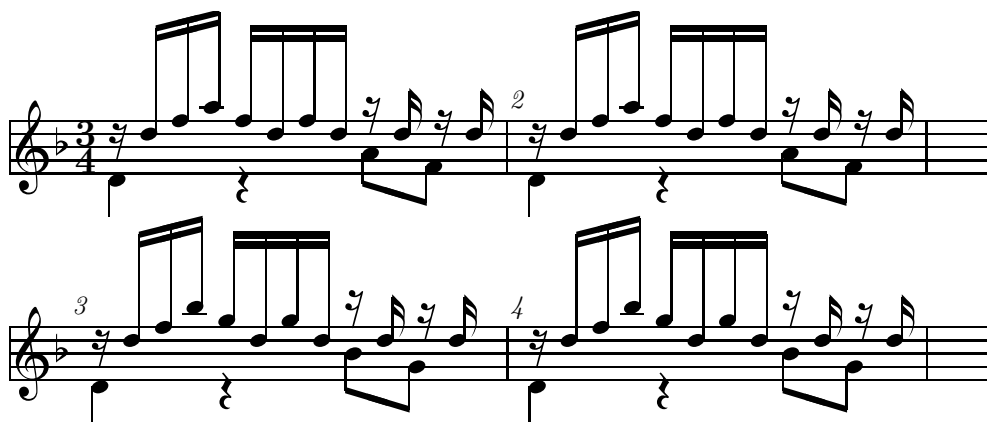


Рис. 16.2. Фрагмент Прелюдии И. С. Баха в переложении Андреса Сеговии, набранный с использованием пакета **musixtex**

На рис. 16.2 изображены первые две строки Прелюдии И. С. Баха в переложении Андреса Сеговии для гитары. Вот так эти ноты выглядят при наборе:

```
\begin{music} % Фрагмент Прелюдии И.С.Баха
\parindent=0pt% Запрещаем отступ
\generalmeter{\meterfrac34}% Длительность такта 3/4
\generalsignature{-1}% Устанавливаем тональность
% Определяем команду для вертикального сдвига аргумента
\newcommand*{\lift}[2]{\raise#1\Interligne#2}
\startpiece% Начинаем набор
\notes% Первый такт
\zql{d}\lift{1}{\qs}% Набор ноты ре и паузы друг над другом
\ibbu0s5\qb0{km}\tbu0\qb0{o}% Перебор
\ibbu0s0\zqb0{m}% Начало перебора
\lift{-3}{\qr}% Набор паузы под первой нотой перебора
\qb0{km}\tbu0\qb0{k}% Конец перебора
\ibl0j{-6}\zqb0{h}%
\lift{3}{\qs}\ssu{k}%
\tbl0\zqb0{f}\lift{2}{\qs}\ssu{k}%
\enotes% Первый такт окончен
\bar% Вертикальная черта
    Ещё три примерно таких же такта.
\endpiece% Окончание набора
\end{music}
```


musixtex позволяет набирать музыку фактически любой степени сложности, но простым в использовании его назвать нельзя. С другой стороны основные правила набора осваиваются за вполне обозримый промежуток времени и формат MusiX_{TEX} можно использовать как вывод какой-либо визуальной программы для последующей качественной печати нотных записей.

musixlyr

Пакет **musixlyr** — расширение MusiX_{TEX}, которое можно использовать для наборов текстов песен совместно с мелодией. На рис. 16.3 на примере первого куплета песни Вадима Егорова «Друзья уходят как-то невзначай...» показан вывод кода, представленного далее:

```
% Преамбула
% необходимо загрузить до musixlyr
\usepackage{musixtex}
% У пакета musixlyr отсутствует стиль для ЛATEX
\input{musixlyr}

% Тело документа

\begin{music}
\parindent=0pt% Запрещаем отступ
\generalmeter{\meterfrac{4}{4}}% Длительность такта 4/4
\generalsignature{-1}% Устанавливаем тональность
\nobarnumbers % Выключаем автоматическую нумерацию тактов
% Куплет из песни Вадима Егорова
\setlyrics{Egorov}{% Присваиваем текст переменной Egorov
    Дру-зья у-хо-дят как то не-взна-чай,
    Дру-зья у-хо-дят {в про}-шло-е, как {в за}-мать.
    А мы сме-ём-ся {с но}-вы-ми друзь-я-ми,
    О ста-рых вспо-ми-на-ем по но-чам_
    О ста-рых вспо-ми-на-ем по но-чам
}
\assignlyrics1{Egorov} % Добавляем текст к нотам
\startpiece% Начало нотного набора
% Немного опускаем текст, чтобы не цеплялся за ноты
\lyrraise{1}{b-1.5mm}
\Notes\qp\ds\ca{acf}\en\bar
\Notesp\ca{hg}\qp\ca{fe}\ds\cca{de}\en\alaline
% Возвращаем текст в позицию по умолчанию
\lyrraise{1}{b0mm}
\Notesp\ha{f}\ds\ca{def}\en\bar
\Notesp\ca{ji}\qp\ca{hgfe}\en\alaline
```

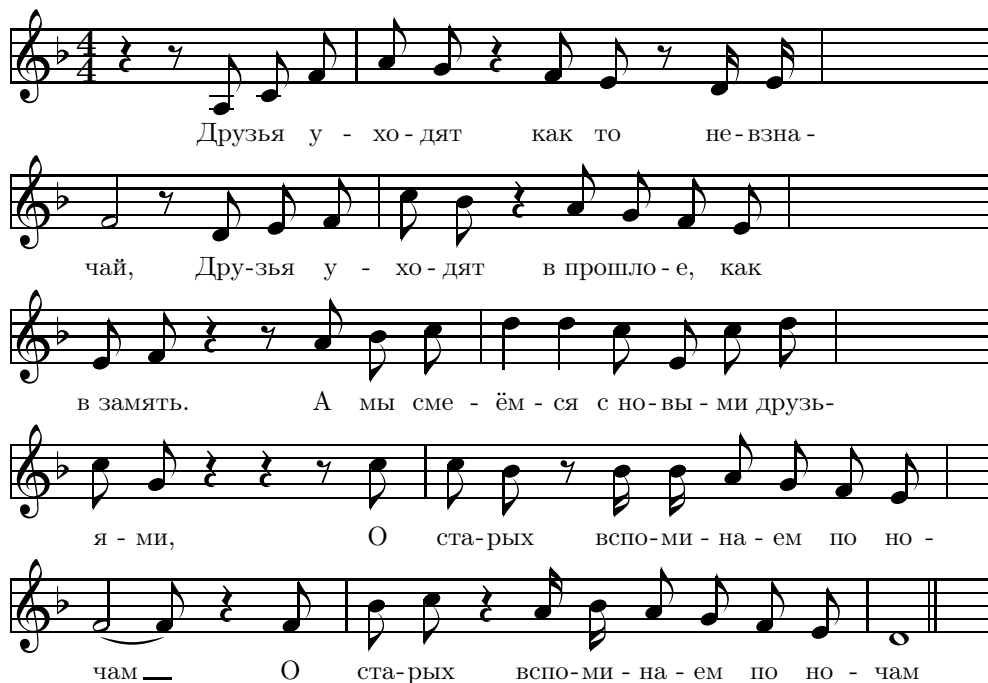


Рис. 16.3. Первый куплет песни Вадима Егорова, набранный с помощью пакетов **mysixtex** и **mysixlyr**

```
\Notesp\ca{ef}\qp\ds\ca{hij}\en\bar
\Notesp\qa{kk}\ca{jejk}\en\alaligne
\Notesp\ca{jg}\qp\qp\ds\ca{j}\en\bar
\Notesp\ca{ji}\ds\cca{ii}\ca{hgfe}\en\alaligne
\NOtes\islurd0f\ha{f}\tslur0f\ca{f}\qp\ca{f}\en\bar
\Notesp\ca{ij}\qp\cca{hi}\ca{hgfe}\en\bar
\notesp\wh{d}\en
\setdoublebar
\endpiece% Конец нотного набора
\end{music}
```

Текст добавляется к нотам автоматически в соответствии с разбивкой, указанной в команде `\setlyrics`. Знак минуса (-) позволяет разбить слова на слоги. С помощью фигурных скобок ({}), символы группируются. Знак подчёркивания (_) позволяет продолжить слог на более чем одну ноту. Документация к пакету **mxlyrdoc.dvi** достаточно подробна.

16.3. Языкознание

Для лингвиста необходимо уметь выводить на лист бумаги множество разнообразных символов: как обычных и не очень алфавитов, так и спецсимволов. $\text{\LaTeX}$ предоставляет лингвисту впечатляющий набор разнообразных шрифтов и отдельных символов. Для просмотра имеющихся вариантов можно воспользоваться уже не раз упомянутым в этой книге документом The Comprehensive $\text{\LaTeX}$ Symbol List (`symbols-a4.pdf`).

tipa

Для записи транскрипции Международной фонетической ассоциацией на основе латинского алфавита был разработан Международный фонетический алфавит (International Phonetic Alphabet или IPA).

Пакет **tipa** позволяет использовать символы IPA при наборе текста. Для доступа к символам можно использовать:

- команду с параметром `\textipa{текст}`,
- декларативную команду `\tipaencoding`,
- окружение IPA.

Для того чтобы напечатать символ, можно и просто его вызвать по имени, но можно и воспользоваться упомянутыми командами и окружениями:

Можно просто набирать имена символов:

```
[\textepsilon ksp1\textschwa%
\textprimstress ne%
\textsci\textesh\textschwa n]
```

Можно просто набирать имена символов: `[eksp1ə'neɪən]`

А можно воспользоваться сокращениями: `[eksp1ə'neɪən]`

А можно воспользоваться сокращениями:

```
\textipa{[Eksp1@"neɪS@n]}
```

Всегда можно выбрать из двух вариантов в зависимости от того, что удобнее. Все подробности и все символы подробно описаны в документации к этому пакету `tipaman.pdf`.

HieroTeX

Пакет HieroTeX отсутствует в дистрибутиве $\text{\TeX}$ Live. Домашняя страничка проекта <http://www.iut.univ-paris8.fr/~rosmond/archives/> является и страничкой автора. На ней можно найти дистрибутив `HieroTeX-3.5.tgz` и Type1-шрифты `HieroType1-3.1.4.tgz`. Всё это вместе предназначено для целей египтологии. HieroTeX позволяет писать иероглифами.

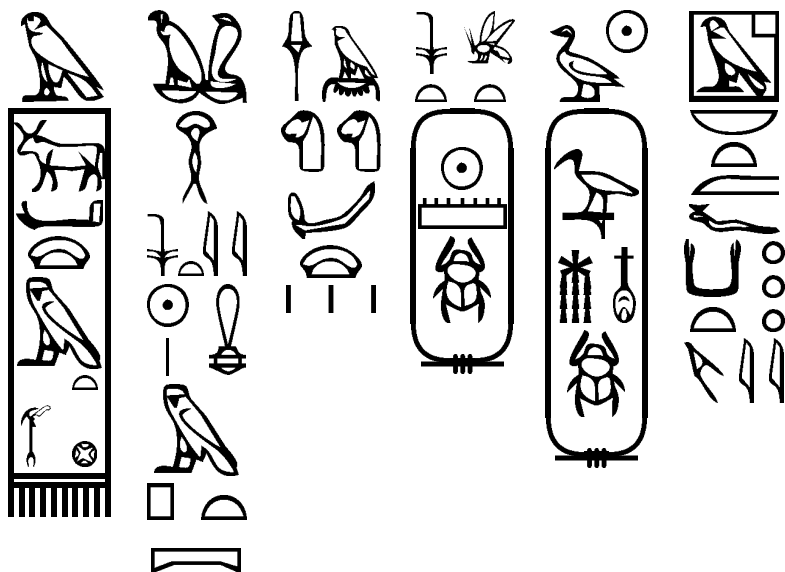


Рис. 16.4. Египетские иероглифы (HieroTeX)

Установка пакета представляет собой сборку программы-транслятора **sesh**, копирование стилей и шрифтов в локальное **texmf**-дерево и обновление индексов с помощью программы **texhash**. Подробнее об установке можно почитать в документации пакета **EGypt**.pdf. Далее создаётся файл (пусть будет **hierotex.htx**) вида:

```
% файл hierotex.htx
% Пример взят из EGypt.tex пакета hierotex
\documentclass{article}
\usepackage{egypt}
\usepackage{hiero}
\begin{document}
\EnColonne[1.2\Htm]{
\begin{hieroglyph}
G5 <S E1 D40 xa m R19*(t\s1:niwt\s1) >-!
nbtY wAH sw*!\t\s1*!\i*i (ra:Z1)*mi m (p*t:pt)-!
sxm*G8 F9*F9 Dsr-xa-Z2-!
(sw:t)*(bit:t) <ra-mn-xpr>-! zA*!\!\!(ra:.)
<-G26-ms*nfr-xpr->-! O10 nb-t M-f (kA:t)*(N33:N33:N33)
mr*i*i
\end{hieroglyph}}
\end{document}
```

Этот файл преобразуется с помощью команды **sesh** в ЛАТ_EX-файл (пусть это будет **hierotex.tex**):

```
> sesh < hierotex.htx > hierotex.tex
```

Меняются символы только внутри окружения **hieroglyph**. Можно откомпилировать результат с помощью **pdflatex** или **latex** и получить то, что изображено на рис. 16.4. Возможно, выглядит немного запутанно, но сами иероглифы не менее изощрённы, а прекрасная документация будет путеводной нитью для египтолога. По отзывам это очень качественный инструмент для специалиста.

arcs

Иногда лингвисту нужно просто подчеркнуть какую-то часть слова, и в этом ему поможет пакет **arcs**:

Можно рисовать дуги `\underarc{под}` и `\overarc{над}` текстом.

Можно рисовать дуги  под и  над текстом.

► В заключение хотелось бы указать на специальный для лингвистов ЛАТ_EX-ресурс: <http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/>, который так и называется «The LaTeX for Linguists».

Глава 17

Каталог пакетов T_EX Live

Наиболее эффективный способ решения возникшей проблемы после её осознания — это поиск уже готового решения. Прежде чем решать проблему самому — лучше потратить некоторое время на просмотр каталога и найти готовое или похожее решение и довести его до нужного состояния. *Нет* необходимости всё делать с нуля.

T_EX Live — довольно большой дистрибутив и содержит значительную часть CTAN-архива¹.

В этой главе предпринята попытка кратко упомянуть почти все пакеты L^AT_EX, которые входят в T_EX Live и не вызывают проблем при использовании.

Часть пакетов описаны в книге более подробно, о чём даёт знать специальный значок ✓ в начале абзаца.

17.1. Набор текста

Текст может походить на плавный поток равнинной реки, но может быть похожим и на горную речку с её неповторимыми особенностями. Для этого тексту нужны разнообразные средства выражения.

17.1.1. Интернационализация и локализация

✓ **fontenc** — стандартный пакет для выбора внутренней кодировки L^AT_EX.

✓ **inputenc** — стандартный пакет для указания, в какой кодировке набран текст.

¹Лучшим кратким путеводителем по CTAN является созданный Грэхамом Вильямсом (Graham Williams) «TeX Catalogue», который постоянно обновляется и доступен по адресу <http://texcatalogue.sarovar.org> в html-формате.

✓ **babel** — стандартный пакет локализации или выбора языка документа. Пакет поддерживает сорок три языка без учёта диалектов. Список поддерживаемых языков приведён в разделе 10.1 на стр. 121.

Кроме языков, поддерживаемых пакетом **babel**, отдельно существуют пакеты для набора в L^AT_EX (не полный список) на арабском (**arabtex**), армянском (**armtex**), бенгальском (**arosgn**, **bangtex**), бирманском (**burmese**), вьетнамском (**vn_{tex}**), гурмукхи (**gurmukhi**), деванагари (**devanagari**), дравидийском (**mala_{yalam}**), индийском (**itrans**), инуитском (**oinuit**), ирокезском языке племени чероки (**ocherokee**), китайском (**cjk**), корейском (**cjk**), маньчжурском (**manjutex**), монгольском (**montex**), санскритском (**sanskrit**), сингальском (**sinhala**), тамильском (**adami**), тибетском (**ctib4tex**, **pecha**), эфиопском (**enthiop**), сомали (**os_{manian}**), телугу (**telugu**) и японском (**cjk**) языках.

ucs — поддержка Unicode.

braille — поддержка набора с помощью шрифта Брайля (для слепых).

17.1.2. Русификация

✓ **anttor** — инициализация шрифтов Antykwa Torúnska. В шрифтах есть кириллица.

✓ **concrete** — инициализация шрифтов **concrete** (для текста) и **euler** (для математики). Шрифт использовался Д. Э. Кнудом для набора книги Конкретная математика. Полностью русифицирован.

cyrillic — поддержка кириллицы в L^AT_EX. Включает набор кириллических кодировок для **in_{putenc}**.

✓ **lh** — набор кириллических шрифтов базового семейства Computer Modern. Доступны в векторном формате Type1 в составе пакета **cm_{super}**.

✓ **indentfirst** — после инициализации этого стиля первый параграф раздела начинается с красной строки.

✓ **t2** — набор макросов и вспомогательных программ для поддержки набора кириллицы. В пакет входят стили **misc_{corr}** (при загрузке исправляет некоторые несоответствия с правилами отечественной полиграфии), **mat_{text}** (позволяет набирать русский текст в математической моде) и **cite_{hack}** (кириллические метки для Bib_{T_EX}).

✓ **es_{kdx}** — реализация стандарта ЕСКД.

✓ **gost780u** и **gost71u** из пакета **gost** — поддержка ГОСТ 7.80-00 и ГОСТ 7.1-84 для Bib_{T_EX}.

17.1.3. Пунктуация

extdash из пакета **ncctools** — простые псевдонимы для команд дефисов, тире и пробелов.

✓ **ncccomma** из пакета **ncctools** — позволяет использовать запятую вместо точки для отделения целой части числа от дробной. Схожей функциональностью обладает стиль **icomma** из пакета **was**.

✓ **underscore** — позволяет набирать знак подчёркивания («_») в текстовой моде.

17.1.4. Выделение текста

contour — возможность добавление цветного контура вокруг букв. Эмуляция контурных шрифтов.

nccstretch из пакета **ncctools** — разрядка текста.

✓ **soul** — печать в разрядку, подчёркивание, перечёркивание текста и многое другое.

truncate — урезает текст до необходимой длины.

ulem — разные способы подчёркивания и перечёркивания текста.

umoline — определяет макросы для подчёркивания текста с возможностью переноса.

17.1.5. Разделительные линии

dashrule — пунктирные разделительные линии.

✓ **mboxfill** из пакета **ncctools** — заполнение пространства указанными символами.

✓ **nccrules** из пакета **ncctools** — различные типы разделительных линий.

17.1.6. Форматирование параграфа

✓ **lineno** — добавляет номер строки на полях.

✓ **ragged2e** — выравнивание абзаца по правому или левому краю без выключения механизма переносов.

hanging — управление отступами внутри параграфа.

✓ **comment** — определяет одноимённое окружение, позволяющее закомментировать текст внутри него.

shapepar — печать параграфа любой заданной формы. Полезен, если нужно напечатать что-то в форме сердечка или тыквы.

17.1.7. Перечни

✓ **desclist** из пакета **ncctools** — улучшенный аналог окружения `description`.

engrec — нумерация перечня производится с помощью заглавных и строчных греческих букв.

enumerate — добавляет необязательный аргумент к окружению `enumerate` для определения стиля нумерации перечня.

enumitem — задание макета перечня.

✓ **eqlist** — определяет одноимённое окружение, похожее на `description`, но с одинаковым отступом для текста перечня.

etaremunum — определяет нумерованный перечень с обратной нумерацией.

expdlist — расширение окружения `description`.

multenum — многоколоночный перечень.

✓ **paralist** — формирование внутриабзачного перечня.

17.1.8. Неформатированный текст

✓ **alltt** — вывод неформатированного текста с возможностью исполнения команд L^AT_EX внутри одноимённого окружения.

fancyvrb — определяет множество изопрённых способов по работе с неформатированным текстом.

moreverb — расширение возможностей пакета **verbatim**.

✓ **verbatim** — вывод неформатированного текста.

17.1.9. Цитирование

balancedquotes — заменяет симметричные "кавычки" на сбалансированные, например, «ёлочки».

csquotes — пакет для создания цитат. Локализован для некоторых европейских языков. Русский язык в наличии.

✓ **epigraph** — формирование эпитафий.

quotchap — добавление эпитафий к главам.

17.1.10. Рубрикация

alnumsec — буквенная нумерация разделов.

chappg — страницы для каждой новой главы опять начинают нумероваться с единицы.

fncychap — шесть дополнительных нестандартных способов оформления заголовков глав. В этой книге используется определённый в данном пакете стиль Lenny.

nccsect из пакета **ncctools** — альтернативный механизм рубрикации.

sectionbox — помещает разделы в причудливо оформленные боксы. Возможно, сгодится для создания постера.

titlesec — альтернативный способ формирования заголовков разделов, сильно отличающийся от стандартной схемы. Для работы с оглавлением в том же стиле следует использовать пакет **titletoc**.

appendix — дополнительные возможности по работе с приложением.

17.1.11. Титульный лист

authblk из пакета **preprint** — переопределяет команду `\author`, позволяя вводить имя автора и организацию, к которой он принадлежит, более естественным образом.

coverpage — включает стиль **CoverPage**, который определяет шаблон для создания обложки для научной статьи.

titling — улучшение стандартной команды `\maketitle`.

abstract — доступ к параметрам, отвечающим за оформление аннотации.

17.1.12. Перекрёстные ссылки

crossreference — создание специального списка перекрёстных ссылок.

lastpage — создаёт метку, при ссылке на которую получаешь последнюю страницу документа.

✓ **prettyref** — позволяет форматировать ссылки в зависимости от префикса метки.

refcheck — печатает имена метки на полях, выделяя те из них, на которые ссылок нет.

✓ **showkeys** — визуализирует имена меток. Это может оказаться полезным в процессе подготовки документа.

typedref — заменяет команду `\ref` командами `\figureref`, `\sectionref` и `\eqref`. Команда `\label` запоминает тип метки и не позволяет неправильно на себя ссылаться.

varioref из пакета **tools** — ссылки на страницы с учётом их положения относительно текущей.

✓ **xr** — возможность ссылок на внешний документ.

17.1.13. Сноски

fixfoot — позволяет использовать одно и то же подстрочное примечание несколько раз в пределах одной страницы.

✓ **footmisc** — предоставляет множество управляемых параметров для влияния на вид и формат сносок.

footnpag — при окончании каждой страницы номер сноски обнуляется.

✓ **ftnright** из пакета **tools** — размещение подстрочных примечаний при двухколоночной вёрстке в конце правой колонки.

manyfoot из пакета **ncctools** — многоуровневые сноски.

✓ **nccfoots** из пакета **ncctools** — ручная нумерация сносок.

✓ **savefnmark** — позволяет сохранять и использовать заново знаки сноски. Это работает и при переходе от обычного текста к таблице и обратно.

yafoot — содержит стилевые файлы **pfnote** (на новой странице нумерация сносок опять начинается с начала), **fnpos** (позволяет контролировать вертикальное положение сноски) и **dblfnote** (печать подстрочных примечаний в две колонки).

17.1.14. Затекстовые примечания

endheads — обеспечивает в разделе затекстовых примечаний «бегущий заголовок» для быстрой ориентации при поиске.

✓ **endnotes** — размещает все сноски в конце текста.

✓ **pagenote** — позволяет организовать затекстовые комментарии в конце каждой главы.

17.1.15. Заметки на полях

✓ **marginnote** — позволяет создавать заметки на полях без использования плавающего объекта.

✓ **mparhack** — исправление ошибки L^AT_EX с возможно неправильным размещением заметок на полях.

✓ **changebar** — метки в виде полосы на полях документа. Используется для индикации изменений в тексте.

17.1.16. WWW-элементы

✓ **breakurl** — перенос гиперссылки на другую строку.

✓ **hyperref** — поддержка гиперссылок.

typethtml из пакета **carlisle** — печатает HTML прямо из L^AT_EX. Поддерживает почти все теги HTML2 и большинство математических фрагментов из HTML3.

✓ **url** — ввод гиперссылки без необходимости экранирования специальных символов.

cooltooltips — позволяет создавать всплывающие подсказки над гиперссылками. Работает с **Adobe Reader** начиная с 7 версии.

17.1.17. PDF

smap — добавляет возможность поиска и копирования текста в pdf-документ.

pdfcrypt из пакета **oberdiek** — позволяет зашифровать pdf-файл и выставить ограничения на печать документа.

17.1.18. Боксы

boites — определяет окружения, формирующие боксы, которые могут частично переноситься на другую страницу.

boxedminipage — определяет одноимённое окружение, создающее мини-страницу с рамкой вокруг неё.

boxhandler — именованные боксы.

dashbox — рисует пунктирную рамку вокруг бокса.

eqparbox — создаёт группу боксов одинаковой ширины, равной максимальной ширине среди ширин аргументов.

fancybox — создание боксов с разнообразными рамочками и фоном.

nccboxes из пакета **ncctools** — дополнительные способы создания боксов. Пакет ориентирован на работу с таблицами. В частности предоставляет набор команд для создания невидимых подпорок определённой высоты.

pst-fr3d — 3D-боксы.

✓ **shadow** — макрос `\shabox`, создающий бокс с рамкой и тенью.

17.1.19. Процесс подготовки документа

✓ **draftwatermark** — печатает по диагонали через всю страницу слово «DRAFT» или какое-либо ещё выражение. Этот пакет полностью перекрывает возможности более старого пакета **draftcopy**.

✓ **fixme** — пояснительные комментарии и заметки о ходе выполнения работы над документом в теле текста.

prelim2e — при инициализации внизу каждой страницы появляется временная метка и текстовая строка, которая может задаваться пользователем.

progress — предоставляет одноимённую команду для индикации степени завершенности документа и создаёт HTML-файл, показывающий этапы подготовки документа. Пакет может быть полезен при командной работе над большим документом.

pdfsync — синхронизация кода и pdf-файла. Позволяет попасть в соответствующее место кода на основании pdf-фрагмента и обратно. Поддерживает текстовые редакторы **iTeXMac**, **iTeXMac2**, **TeXShop** и **emacs (AucTeX)** и программы просмотра pdf **iTeXMac**, **iTeXMac2**, **TeXShop**, **TeXnscope** и **PDFView**.

srcltx — синхронизация кода и dvi-файла. Поддерживается программами просмотра DVI **xdvi** и **yap**.

snapshot — позволяет получить версии всех пакетов, от которых зависит документ, и гарантировать, что документ будет собираться только с пакетами именно этих версий. Это позволяет «заморозить» результат компиляции.

✓ Пакеты **svn** и **svninfo** облегчают контроль версий исходных текстов L^AT_EX при использовании **svn**. В случае если применяется система контроля версий **cvs**, следует воспользоваться аналогичными пакетами **rcs** и **rcsinfo**.

svnk или **svn-multi** — ещё одна поддержка **svn**.

✓ **vhistory** — пакет для ведения журнала изменений документа.

vpe — аналог **srcltx** для PDF. Требуется наличия исполняемого файла **vpe** в системе.

17.1.20. Составные документы

Для комбинации нескольких документов в один следует обратить внимание на пакеты **combine** и **pdfpages**.

✓ **askinclude** — интерактивное подключение частей документа с помощью инструкции `\include`.

✓ **excludeonly** — определяет одноимённый макрос, который позволяет указывать, какие файлы не следует включать с помощью инструкции `\include`.

chapterfolder — предоставляет набор макросов для описания директорий, в которых лежат исходные тексты глав и разделов, и подключения их без указания полного пути.

import — аналог `\input`, позволяющий загруженному файлу использовать свою директорию как базовую при подключении других фрагментов текста и картинок.

volumes — усовершенствование механизма, предоставляемого стандартной командой `\includeonly`.

17.1.21. Всякая всячина

crop — создаёт угловые метки (cropmark) для обрезки страницы или вклейки фотографий. Схожую функциональность предоставляет **nccscropmark** из пакета **ncctools**.

fixltx2e — исправление некоторых ошибок L^AT_EX 2_ε.

mylatex из пакета **carlisle** — содержит файл `mylatex.ltx`, позволяющий сгенерировать свой форматный файл, в который встроены часто используемые пакеты. Полезно для ускорения компиляции, хотя для современных компьютеров не особо актуально.

morse — представление текста с помощью азбуки Морзе (английская версия).

pagesel из пакета **oberdiek** — предоставляет возможность выбирать, какие страницы следует вывести. Позволяет указывать конкретные страницы, диапазоны и чётные/нечётные страницы.

totpages — предоставляет информацию о числе страниц, получившихся в результате компиляции.

wordlike — формирует структуру заголовков, разметку страницы и подставляет шрифты, для того чтобы результат выглядел «как в ворде». Пакет на любителя. С кириллицей, скорее всего, заставить работать не получится.

17.2. Точные науки

L^AT_EX создавался под математику, а так как математика — язык всех точных наук, то любая из них легко может быть изложена с помощью L^AT_EX.

17.2.1. Математика

✓ **amsmath** — базовый стиль $\mathcal{AMS}$ -L^AT_EX. Определяет дополнительные окружения для выключенных формул. При загрузке автоматически подгружает пакеты **ambsy** (жирные математические символы), **amsopn** (создание новых математических операторов) и **amstext** (определяет команду `\text`, которая позволяет

набирать обычный текст в математическом режиме). Если в документе есть формулы, то этот стиль обязателен для использования.

✓ **amssymb** — даёт имена всем символам из шрифтов семейства $\mathcal{AMS}$. Автоматически загружает пакет **amsfonts** (дополнительные математические символы и шрифты). Входит в коллекцию $\mathcal{AMS}$.

✓ **amscd** — коммутативные диаграммы. Входит в коллекцию $\mathcal{AMS}$.

amsthm — улучшенные теоремы. Определяет окружение `proof`. Входит в коллекцию $\mathcal{AMS}$.

✓ **mh** — множество улучшений и дополнений для пакета **amsmath**. Пакет содержит стили **mathtools** — улучшение вывода формул, **nththeorem** — улучшение для окружения `theorem` и **empheq** — дополнительные средства визуализации для выключенных формул.

mhqqu — печать формул в несколько колонок.

Пакет **bezos** состоит из стилей **accents** (определение новых акцентов и создание их сложных комбинаций), **tensind** (улучшенные тензорные индексы), **dotlessi** (макросы `\dotlessi` и `\dotlessj`), **esindex**, **checkend** (более подробное предупреждение об ошибке при наличии не закрытого окружения) и **arabicfront** (нумерация начинается с первой страницы).

breqn — автоматический перенос в многострочных выключенных формулах. Одно из решений проблемы открывающей и закрывающей скобок при переносах формулы. При взаимодействии с другими пакетами случаются коллизии.

✓ **cancel** — определяет макросы `\cancel`, `\bcancel` и `\xcancel`, рисующие линии, перечёркивающие аргумент команды.

✓ **cases** — определяет окружения `numcases` и `subnumcases` для отображения условных конструкций.

✓ **commath** — улучшенное отображение дифференциалов, частных производных, пределов.

complexity — позволяет набирать тексты для теории сложности вычислений.

dcpic — коммутативные диаграммы средствами $\text{\LaTeX}$.

deleq — предоставляет гибкий способ нумерации выражений.

easy — коллекция «простых» математических макросов. Состоит из стилей **easy** (загружает все **easy**-расширения), **easyeqn** (определяет окружение для выключенных уравнений), **easymat** (описание матриц), **easybmat** (описание блочных матриц), **easytabl** (набор таблиц), **easyvector** (C-подобный синтаксис для отображения векторов и матриц), **easybib** (дополнительные настройки для отображения библиографии).

egameps — позволяет рисовать средствами **pstricks** развёрнутую форму игры (extensive games).

eqname — определяет команду `\eqname` для замены числовой метки уравнения символьной.

✓ **esdiff** — макросы для представления производных.

esvect — разнообразные и более чёткие, чем используемые по умолчанию, знаки вектора над символами.

eulervm — загрузка математических шрифтов Euler.

extarrows и **extpfeil** — дополнительные типы стрелок.

faktor — определяет одноимённый макрос `\faktor`, который формирует дробь вида A/B . Требуется **amsmath**.

flagderiv — логические доказательства с использованием флаг-нотации (flag notation).

gauss — визуализация решения системы линейных уравнений методом Гаусса.

hhtensor — дополнительные типы представления имён векторов, матриц и тензоров.

karnaugh — отрисовка «Karnaugh Maps», также известных как «Veitch diagram».

maybemath — определяет макросы `\maybebm` и `\maybeit`. В зависимости от контекста текст становится жирным или наклонным.

nath — акроним от *natural math notation*. Переопределяет многие математические команды для целей автоматизации процесса вёрстки. В частности размер скобок в уравнении выбирается автоматически в зависимости от контекста, а размер квадратного корня не зависит от наличия индекса у подкоренного выражения.

nccthm из пакета **ncctools** — улучшение стандартного набора команд генерации математических утверждений (теорем).

onlyamsmath — при инициализации этого пакета пользователю остаётся доступна только математическая нотация, принятая в L^AT_EX в ущерб низкоуровневой T_EX-нотации. Полезно для избавления от дурных привычек.

✓ **pb-diagram** — коммутативные диаграммы.

pst-3dplot — графики трёхмерных математических функций. Возможна загрузка данных из внешнего файла для печати трёхмерных поверхностей.

pst-bar — рисование гистограмм.

pst-eucl — построение простых и не очень геометрических чертежей.

pst-func — построение математических функций. В пакете предусмотрены полиномы, суммы Фурье, функции Бесселя, гауссовское распределение и многое другое.

qobitree — простые древесные диаграммы.

sobolev — операции с пространствами Соболева и Гильберта.

sseq — позволяет рисовать диаграммы спектральных последовательностей, или spectral sequence charts.

subeqn — определяет окружение subequations для создания ещё одного уровня нумерации формул.

t-angles — рисования диаграмм для алгебры Хопфа (Hopf algebra).

✓ Пакеты **tableaux**, **tabvar** и **variations** созданы для исследования области определения и поведения функции. Пригодится при изучении или написании методички по началам матанализа.

✓ **tensor** — представление тензоров.

vector — представление векторов.

- venn** — создание диаграмм Эйлера—Венна (MetaPost).
- ✓ **was** — состоит из стилевых файлов **fixmath** (меняет прямые начертания заглавных греческих букв на наклонные), **gensymb** (делает доступными в текстовой моде некоторые из часто используемых символов, например, знак градуса), **icomma** (обеспечивает правильные пробелы в текстовой моде для чисел с дробной частью), **upgreek** (обеспечивает доступ к прямым начертаниям греческих символов через приставку **up** в их именах).
- ✓ **wasysym** — дополнительные символы. Загрузка пакета с опцией **integrals** переопределяет символы интегралов с наклонных на прямые.

17.2.2. Информатика

- ascii** — шрифты IBM ASCII.
- ✓ Стандартные пакеты **algorithms**, **algorithmic**, **algorithmicx**, **algorithm2e**, **clrscode** и **pseudocode** специализируются на представлении алгоритмов.
- basix** — интерпретатор языка BASIC, написанный на $\TeX$. Пакет доступен по адресу: `{CTAN}/macros/generic/basix/`. В дистрибутиве $\TeX$ Live отсутствует.
- ✓ **bytefield** — диаграммы для описания бинарных протоколов передачи данных.
- cursor** — рисует L-образный курсор в математической моде.
- examplep** — печать примеров кода на $\LaTeX$ и на MetaPost. Пакет состоит из стилей **codep** (определяет окружение `code`, позволяющее выводить код слева, а результат компиляции справа) и **examplep** (определяет окружения и команды для печати и вывода в файл неформатированного текста). Значительная часть функциональности, обещанной в документации, в коде отсутствует.
- keystroke** — изображение клавиш клавиатуры. PostScript-изображения создаются не очень корректно и при преобразовании в pdf возникают «артефакты».
- ✓ **lcd** — эмуляция вывода текста на маленьком LCD-дисплее.
- ✓ **listings** — оформление программного кода.
- method** — формальное описание функции.
- pst-dbicons** — описание и прототипирование баз данных с помощью ER-диаграмм.
- pst-uml** — создание не сильно сложных UML-диаграмм.
- semantic** — нотация для описания семантического разбора и компиляции программного кода, включая T-диаграммы.

17.2.3. Физика

- Пакеты шрифтов **astro** и **cmastro** предоставляют доступ к астрономическим символам, включая знаки зодиака.
- braket** — бра-кет нотация Дирака.
- ✓ **circ** — создание не сильно сложных электрических схем средствами $\LaTeX$ и METAFONT.
- ✓ **feyn** — простые фейнмановские диаграммы.

✓ **feynmf** — набор макросов для создания сложных фейнмановских диаграмм средствами L^AT_EX и METAFONT/MetaPost.

✓ **hepparticles** — набор макросов для печати названий частиц для Физики высоких энергий.

✓ **isotope** — макрос для печати названий изотопов химических элементов.

pst-circ — простые электрические схемы.

✓ **pst-optic** — создание иллюстрации для геометрической оптики.

✓ **pst-osci** — иллюстрации осциллограмм.

✓ **S_Istyle** — единицы физических величин по правилам СИ. Схожую функциональность предоставляет пакет **S_Iunits**.

slashed из пакета **carlisle** — слэш-нотация Фейнмана. Пакет определяет макрос `\slashed`, который перечёркивает переданный ему символ в математической моде, например: $\not{A} \stackrel{\text{def}}{=} \gamma^\mu A_\mu$.

timing — рисование простых временных диаграмм средствами L^AT_EX и METAFONT.

units — печать единиц физических величин. Есть альтернативный способ представления дроби, а также проблемы со шрифтами.

unitsdef — печать единиц физических величин в том числе и по правилам СИ.

17.2.4. Химия

bpchem — печать химических элементов, названий, формул и нумерация химических соединений.

chemarr из пакета **oberdiek** — дополнительный набор стрелок для химиков.

chemarrow — ещё один дополнительный набор стрелок.

chemcompounds — простая последовательная нумерация химических соединений. Возможно присвоение соединению своего уникального имени.

chemcono — нумерация химических соединений по принципу организации библиографических ссылок.

cryst — набор символов для описания типа симметрии в кристаллографии.

✓ **mhchem** — продвинутый пакет для печати названий химических элементов, формул и значков для маркировки опасных химических соединений (возможна локализация).

✓ **pst-labo** — создание иллюстраций для лабораторных работ с помощью пакета **pstricks**.

r-und-s — маркировка опасных химических соединений (возможна локализация).

✓ **xymtex** — пакет для отображения средствами L^AT_EX множества химических структурных формул. Превосходит устаревший пакет **chemtex** со схожей функциональностью по всем параметрам. Последнюю версию пакета следует брать по адресу <http://homepage3.nifty.com/xymtex/fujitas3/xymtex/indexe.html>. В дистрибутиве T_EX Live-2007 отсутствует.

17.2.5. Биология

biocon — создание коротких имён для биологических видов.

dichokey — создание дихотомических идентификационных ключей для целей классификации вида.

dnaseq — вывод простых ДНК-последовательностей.

pst-pdgr — родословное дерево для нужд медиков.

✓ **texshade** — выделение одинаковых последовательностей нуклеотидов и пептидов (nucleotide and peptide alignments).

textopo — расцветка и подписи для рисунков, отображающих геометрическую структуру структурных белков (membrane protein topology plots).

labbook — класс для создания рабочего журнала для биологов с хронологическими метками о проведении эксперимента.

17.3. Школа и университет

answers — совместный набор простых задач и их решений с возможностью их разнесения в разные места текста.

assignment — для создания объявлений о занятиях в университете.

courseoutline и **coursepaper** — создание программы курса предположительно для университета.

eCards — электронные интерактивные карточки с подсказкой для запоминания каких-либо фактов. Для просмотра pdf необходим **Adobe Reader**.

✓ **exam** — предоставляет одноимённый класс, позволяющий подготовить экзаменные билеты и формы для проведения тестов.

examdesign — подготовка тестов.

exercise — набор макросов для добавления упражнений по пройденному материалу.

flashcards — двусторонние карточки для запоминания иностранных слов.

interactiveworkbook — подготовка интерактивного pdf-вопросника. Для просмотра pdf необходим **Adobe Reader**.

probsoln — создание банка заданий с возможностью вывода всех упражнений или какого-то подмножества, в том числе и в случайном порядке.

qcm — класс для создания вопросников с вариантами выборов ответов.

17.4. Гуманитарные направления

Не только точными науками живёт человечество. Гуманитарии тоже используют L^AT_EX.

17.4.1. Поэзия, пьесы и критика

dramatist — вывод пьесы в стихах или прозе.

ednotes — печать комментариев.

ledmac — набор макросов для формирования издания с комментариями. Базируется на T_EX-пакете **edmac**. Имеется расширение в виде пакета **ledpar** для печати параллельного текста.

play — простой набор окружений и макросов для набора пьес.

parrun — пакет, позволяющий печатать текст и его перевод один над другим.

poemscol — пакет, ориентированный на создание сборников стихов и критических заметок.

sides — класс с небольшим набором специализированных макросов для оформления текста пьесы. Базируется на более простом классе **plari**.

✓ **stage** — класс для создания пьес.

verse — сочиняем стихи.

17.4.2. Языкознание

Для рисования различного вида деревьев имеет смысл взглянуть на **pst-tree** — набор макросов для **pstricks**.

arcs — позволяет рисовать дуги под (`\underarc`) и над (`\overarc`) текстом.

cbcoptic — набор текстов на коптском.

covington — набор макросов, позволяющих создавать сложные множественные акценты, трансляцию текста слово-в-слово, нумерацию текстовых примеров, описывать структуру предложения и множество других мелких, но полезных для лингвиста вещей.

engpron — набор макросов, позволяющих набирать транскрипцию слов, как это сделано в словаре Даниэля Джонса (Daniel Jones «English Pronouncing Dictionary»)

lexikon — набор простых макросов для создания двуязычного словаря.

linguex — оформление примеров.

phonetic — дополнительные фонетические символы.

rst — анализ структуры речи.

synttree — синтаксическое дерево.

✓ **tipa** — шрифты и набор макросов для отображения международной фонетической транскрипции (International Phonetic Alphabet).

teubner — расширение **babel**, загруженного с опцией **greek** для набора на классическом греческом.

xytree — формирование лингвистического синтаксического дерева.

17.4.3. Музыка

abc — поддержка L^AT_EX ABC-нотации (запись музыки ASCII-символами). Требуется установки сторонней программы **abcm2ps**.

ConcProg — класс для написания концертных программ. Класс входит в состав пакета **concprog**.

cd и **cd-cover** — классы для создания обложек CD-дисков.

✓ **guitar** — подпись гитарных аккордов над текстом песен.

musictex или **MusicTeX** — очень мощный пакет, ориентированный на набор нот средствами **LaTeX**.

✓ **musixtex** или **MusiXTeX** — современная замена **MusicTeX** со множеством улучшений.

✓ **musixlyr** — расширение **musixtex** для добавления текстов песен.

songbook — формирование песенных сборников с возможностью добавления аккордов над текстом.

17.5. Делу время — потехе час

Для кого-то игры — это хобби, а для кого-то — профессия.

17.5.1. Рецепты

Первый набор **TeX**-макросов для формирования книги рецептов написал сам Дональд Э. Кнут по просьбе своей жены. Подробнее об этих макросах можно прочитать в статье «Макро, написанные для Джил», опубликованной в сборнике «Компьютерная типография».

cooking — несколько простых макросов для оформления сборника рецептов. Ингредиенты описываются рядом с действием по их использованию.

cuisine — чуть более современный аналог **cooking**.

17.5.2. Шахматы

cchess — китайские шахматы.

chess — шрифты для оформления шахматных диаграмм.

skak — пакет позволяет печатать шахматные партии с использованием PGN-нотации (Portable Game Notation) и визуализировать шахматные позиции. Пакет **skaknew** содержит векторные шрифты, полученные из шрифтов **skak**.

texmate — печать партий с возможностью аннотаций и отображением шахматных позиций.

17.5.3. Игры, в которые играют люди

backgammon — визуализация позиции при игре в нарды.

crosswrd — простой кроссворд. Схожую функциональность предоставляет пакет **wpuzzle**.

dice — изображение игральных костей, в том числе и трёхмерные. Если нужны векторные изображения граней костей, то следует использовать пакет **epsdice**.

othello — реверси.

psgo — доски и позиции для игры в Го.

sudoku — головоломка sudoku.

17.5.4. Время

advdate из пакета **ltxmisc** — набор макросов, позволяющий добавить несколько дней к текущему времени, что влияет, например, на вывод команды `\today`.

calendar — пакет с примерами создания не сильно сложных расписаний.

calxxxx — шаблон для печати календарика за любой год. Годится для примера.

china2e — доступ к символам китайского лунного календаря.

clock — рисует часовой циферблат и показывает заказанное время.

datetime — управление форматом вывода даты и времени. Аналогичный сервис предоставляет пакет **isodate**.

weekday — вычисление дня недели по дате.

17.5.5. Карточки и визитки

bizcard — визитки.

labels — множество простых карточек-заметок на одном листе.

ticket — создание визиток, стандартных заметок и записок, которые могут пригодиться при организации большого собрания или конференции.

17.6. Плавающие объекты

Текст — сущность, как правило, непрерывная, но иллюстрации вполне можно отпустить в «свободное плавание» по книге. Надо только за ними присматривать.

17.6.1. Создание плавающих объектов

float — создание новых типов плавающих объектов. Схожую функциональность имеет пакет **floatraw**.

rotfloat — расширяет возможности пакета **float**, добавляя возможность вращения объектов.

17.6.2. Типы плавающих объектов

boxhandler — альтернатива стандартным окружениям **figure** и **table**.

dpfloat — определяет новый тип плавающего окружения, занимающего сразу две страницы. Двойные иллюстрации на развороте.

✓ **floatflt** — полноценный плавающий объект, обтекаемый текстом.

nccfloats из пакета **ncctools** — набор макросов для формирования плавающих объектов.

sidecap — определяет ещё один тип плавающих объектов, позволяющий размещать подпись к нему сбоку.

✓ **wrapfig** — создаёт прямоугольник для размещения рисунка, который обтекается текстом.

✓ **miniplot** — делает то же, что и пакет **subfig**, хоть и менее изощрённо.

photo — плавающее окружение для вставки фотографий.

picinpar — ещё один способ разметить объект внутри абзаца. В отличие от пакета **wrapfig** позволяет разместить объект по центру. Документация на немецком, но разобрать примеры можно и без знания немецкого.

plates — определяет плавающее окружение для создания картинок, которые печатаются отдельно от документа, а затем вклеиваются в него (наклейки). Полезно, если иллюстрации в отличие от текста цветные.

✓ **subfig** — организует группы из множества картинок. Относительно современный пакет.

17.6.3. Подписи к плавающим объектам

capt-of — простой способ сформировать подпись для плавающих объектов вне плавающих окружений.

captcont — замораживает номер иллюстрации по желанию.

✓ **caption** — управление подписями к плавающим объектам, возможность формирования подписей за пределами плавающих окружений. Схожую функциональность предоставляет пакет **ccaption**.

✓ **mcaption** — размещение подписи к плавающим объектам на полях.

subfloat — добавляет дополнительный уровень нумерации для плавающих объектов, позволяя объединять несколько объектов в одну группу.

✓ **topcapt** — определяет макрос `\topcaption`, полностью аналогичный `\caption`, но правильно размещающий подпись над плавающим объектом.

17.6.4. Размещение плавающих объектов

✓ **endfloat** — размещение плавающих объектов в конце документа.

figcaps из пакета **preprint** — при инициализации этого пакета картинки и таблицы из печатного документа убираются, а их названия печатаются в конце на отдельной странице. Полезно при подготовке статьи к печати, где иллюстрации идут отдельно.

hvfloat — позволяет вращать и размещать плавающий объект и подпись к нему как угодно относительно друг к другу.

✓ **placeins** — позволяет выставлять «барьеры», за пределы которых плавающие объекты не перемещаются.

17.7. Графика

Есть много книг без картинок. Но с картинками, чаще всего, они были бы гораздо интереснее.

17.7.1. Добавление иллюстраций

bmpsize из пакета **oberdiek** — стиль для анализа и вычисления размера растрового изображения. При использовании этого пакета нет необходимости в отдельном файле с размерами при подключении растрового изображения.

figsize — специализируется на автоматическом вычислении размеров картинок для размещения их в указанных пределах.

✓ **graphicx** — добавление иллюстраций в документ.

epstopdf из пакета **oberdiek** — позволяет подключать eps-файлы при компиляции с помощью **pdflatex**, вызывая внешнюю программу **epstopdf**.

nccpic из пакета **ncctools** — расширение возможности пакета **graphicx** при работе с растровыми изображениями.

hilowres — позволяет определить при вставке два файла для одного и того же рисунка, например, с низким и высоким разрешением. В зависимости от выбора опций при компиляции подключается либо один, либо другой файл.

rotating — позволяет поворачивать любые боксы.

watermark из пакета **ncctools** — создание «водяных знаков», иначе говоря, вывод графических объектов или текста на подложке страницы. Схожую функциональность предоставляют пакеты **bophook**, **eso-pic** и **wallpaper**.

17.7.2. Управление цветом

aurora — заголовочные файлы для **dvips** для целей цветоотделения. Лицензия позволяет производить цветоотделение только для некоммерческого использования.

✓ **color** — управление цветом текстовых элементов.

colorinfo — извлечение информации об используемой цветовой модели и значений цветовой переменной.

xcolor — интерфейс к определению цветов.

17.7.3. Графика средствами LaTeX

bardiag — создание простых диаграмм на основе пакета PSTricks.

bez123 — кривые Безье.

coordsys — позволяет рисовать и подписывать оси координат и решётки, в том числе и логарифмические. Расширение для окружения `picture`.

curve2e — расширение **pict2e**.

curves — рисование парабол и других кривых. Расширение для окружения `picture`.

ebezier — кривые Безье. Расширение для окружения `picture`.

ecltree — рекурсивные деревья. Расширение для окружения `picture`.

epic — расширение и улучшение команд из окружения `picture`. Доступен также расширенный набор команд для **epic** в виде пакета **eepic**.

histogr — простые гистограммы. Расширение для окружения `picture`.

pdftricks — поддержка возможностей **pstricks** при компиляции файла с помощью **pdflatex**.

pict2e — улучшенная версия окружения `picture`, лишённая многих ограничений.

pgf — «A Portable Graphic Format for TeX». Мощный инструмент подготовки встроенной графики в L^AT_EX, поддерживающий все наиболее популярные `tex`-компиляторы.

pstricks — доступ к инструкциям языка Postscript из L^AT_EX. В базовый пакет входят стилевые файлы **pst-node** (размещение и соединение вершин), **pst-plot** (построение диаграмм), **pst-tree** (построения деревьев).

pst-3d — псевдо-3D, тени, трёхмерный текст и тому подобное.

pst-barcode — печать штрихового кода.

pst-blur — создание размытых теней.

pst-coil — витые и зигзагообразные объекты.

pst-eps — экспорт объектов непосредственно в `eps`-файл.

pst-fill — заливка областей.

pst-geo — отрисовка картографической информации в различных проекциях.

pst-gr3d — 3D-решётка.

pst-grad — заливка области RGB- и HSB-градиентами.

pst-infixplot — позволяет набирать формулы в привычной инфиксной нотации.

pst-lens — эффект линзы на любой картинке или тексте.

pst-light3d — 3D-тени для текста, линий и кривых.

pst-math — расширение стандартных математических функций, встроенных в PostScript.

pst-pdf — экспорт объектов в `pdf`-файл.

pst-poly — изображение многоугольников.

pst-text — набор текста вдоль траектории.

pst-slpe — разные градиенты.

pst-vue3d — простые 3D-объекты в перспективе.

pst-xkey из пакета **xkey** — разбор пар «ключ-значение» для пакета **pstricks**.

scalebar — рисует масштабные линейки.

xypic — графики и диаграммы средствами TeX.

17.8. Всё о таблицах

Случается, что данные проще передавать в виде таблицы. Таблица — это особая форма передачи содержания, и для её правильного оформления следует многое учитывать.

✓ **booktabs** — набор макросов для тонкой настройки параметров таблиц. Цель пакета — помочь автору напечатать идеальную таблицу.

17.8.1. К вопросу о разделительных линиях

✓ **arydshln** — пунктирные разделительные линии.

delarray — разделители (скобки и тому подобное) вокруг окружения `array`.

✓ **hhline** — набор разделительных линий для таблицы.

17.8.2. Клетки

✓ **cellspace** — гарантирует, что между текстом и горизонтальной разделительной линией всегда будет промежуток.

✓ **colortbl** — раскраска строк, колонок и клеток таблицы.

✓ **makecell** — пакет для тонкой настройки параметров и структуры таблицы. Для создания клеток, занимающих несколько строк, в пакете определена команда `\makecell`.

✓ **multirow** — клетки, занимающие несколько строк. В пакет также входят стилевые файлы **bigdelim** (разделители высотой в несколько строк) и **bigstrut** (высокие подпорки).

✓ **slashbox** — формирование клетки на перекрестии заголовка и боковика, разделённой косой чертой.

17.8.3. Выравнивание чисел

✓ **dcolumn** из пакета **tools** — выравнивание по разделителю целой и дробной частей. Схожую функциональность предоставляют пакеты **rccol** и **warpcol**.

✓ **numprint** — позволяет вставлять разделитель через каждые три цифры или менять само написание цифр, чтобы не сбиться при вычислении порядков.

17.8.4. Клоны tabular

array — расширение **tabular** и `array` от Франка Миттельбаха. Тонкая настройка параметров таблицы и разделителей.

blkarray из пакета **carlsle** — определяет окружение **blockarray**, которое в математической моде ведёт себя как `array`, а в текстовой как **tabular**.

tabularht из пакета **oberdiek** — окружение типа **tabular** с параметром, определяющим высоту таблицы.

✓ **tabularx** из коллекции **tools** — расширение **tabular** с ограничением ширины таблицы.

✓ **tabulary** — расширение **tabular** с автоматическим вычислением ширины колонок.

✓ **ctable** — легко определяемые выровненные по центру таблицы.

17.8.5. Многостраничные таблицы

✓ **longtable** — многостраничная таблица с заголовками по умолчанию. Не работает при многоколоночной вёрстке.

ltxtable из пакета **carlisle** — «смесь» **longtable** и **tabularx**.

✓ **supertabular** — чуть более сложный в использовании и менее гибкий аналог **longtable**, зато можно применять в случае многоколоночной вёрстки.

✓ **xtab** — современное расширение возможностей пакета **supertabular** с оглядкой на **longtable**.

17.8.6. Вывод

✓ **listliketab** — печатает перечисление с оформлением в виде таблицы.

17.8.7. Доступ к данным

✓ **csvtools** — чтение CSV-данных.

17.9. Вёрстка

Вёрстка — составление страниц (полос) газеты, журнала, книги определённого размера из набранных строк, заголовков, иллюстраций и тому подобного в соответствие с разметкой или макетом. **Л^AT_EX** умеет многое, но ему нужно сказать, что именно следует сделать.

✓ **afterpage** — откладывание исполнения команды до перехода на следующую страницу.

everypage — исполнение команд в начале каждой страницы.

microtype — интерфейс к микротипографическому расширению **pdftext**. Мощный механизм улучшения общего впечатления от текста. На текущий момент не работает с кириллицей (кодировка T2A). Автор это осознаёт и всячески приветствует исправление данного недостатка.

17.9.1. Ориентация страницы

✓ **lscap** — определяет окружение **landscape**, позволяющее часть помещённого в него текста разместить в альбомной ориентации.

✓ **portland** — смена портретной ориентации на альбомную и наоборот.

✓ **pdflscape** из пакета **oberdiek** — аналог **lscape**. При просмотре pdf-файла страница, повернутая с помощью **landscape**, показывается в альбомной ориентации независимо от ориентации остальных страниц документа.

✓ **rotpages** — позволяет поворачивать указанные страницы вверх тормашками и переставляет их так, чтобы эти страницы можно было нормально читать, перевернув книгу. Полезно в случае, если печатается сборник задач, где необходимо немного затруднить чтение ответов.

17.9.2. Макет

chngepage — смена макета страницы по месту в тексте.

✓ **geometry** — мощный и гибкий интерфейс для установки размеров макета документа.

✓ **layout** — информация о текущем макете.

pdfscreen — создание pdf-документа для чтения текста с экрана монитора.

✓ **pgfpages** из пакета **pgf** — определяет набор макросов, позволяющих автору разместить несколько страниц текста на одной без использования сторонних программ.

pdfpages — вставка и манипуляции с pdf-страницами из внешних pdf-документов.

✓ **twoup** — создание буклета. Для корректной реорганизации страниц требуется пакет **booklet**.

rmpage — тонкая настройка макета.

typearea из пакета **KOMA-Script** — управление полями.

fullpage из пакета **preprint** — все поля выставляются равными 1.5 см.

savetrees — максимально плотная упаковка текста на страницу. Использование этого пакета спасает деревья, но что он делает с глазами?

stdpage — гарантирует, что на странице будет определённое число строк, а в каждой строке не больше определённого числа символов.

✓ **typogrid** — наносит типографскую решётку. Может быть удобно при формировании макета.

✓ **vmargin** — определяет размер полей.

17.9.3. Вертикальные расстояния между элементами текста

✓ **setspace** — устанавливает вертикальное расстояние между строк.

nccparskip из пакета **ncctools** — управляет вертикальными промежутками между абзацами и устраняет избыточные вертикальные отступы в списках.

17.9.4. Многоколоночная вёрстка

✓ **balance** из пакета **preprint** — выравнивает текст в колонках на последней странице при двухколоночной печати.

dblfloatfix — исправляет ошибку с размещением широкой иллюстрации (аналог **midfloat** из пакета **sttools**) и ошибку с нумерацией иллюстраций (исправлено в пакете **fixltx2e**).

✓ **multicol** из пакета **tools** — позволяет совмещать на одной странице многоколоночную и одноколоночную вёрстку.

✓ **parallel** — параллельная печать двух фрагментов текста. Обычно используется для представления оригинального текста и его перевода.

parcolumns из пакета **sauerj** — параллельная печать текста во многих колонках.

✓ **sttools** — пакет, специализирующийся на решении проблем, возникающих при двухколоночной вёрстке. Пакет содержит стилевые файлы **stfloats** (управление размещением плавающих объектов), **floatpag** (переопределение стиля страницы, на которой размещены только плавающие объекты), **flushend** (балансировка колонок текста на последней странице), **cuted** (переключение между одноколоночным/двухколоночным стилями посередине страницы), **marginal** (дополнительные возможности для работы с заметками на полях) и **midfloat** (размещение широкой иллюстрации при двухколоночной вёрстке посередине страницы).

17.10. Путеводитель по классам L^AT_EX

Стандартные классы **article** (статья), **book** (книга), **report** (отчёт), **proc** (доклад), **letter** (письмо) и **slides** (слайды для презентации) прибывают в замороженном состоянии по причине того, что на них базируется и от их настроек зависит множество других пакетов. Поэтому использовать следует отличные от стандартных классы документов. Благо превосходные замены в наличии имеются.

✓ Набор классов **KOMA-Script** является отличной заменой стандартным классам. Базовая функциональность пакета обеспечивается классами **scrartcl** (статья), **scrreprt** (отчёт), **scrbook** (книга) и **scrlltr2** (письмо).

✓ Пакет **ncclatex**, автором которого является Александр Роженко, предоставляет класс **ncc**. В зависимости от опций, передаваемых классу, его можно использовать для написания статей, препринтов, монографий и отчётов. *NC* [10] создавался с оглядкой на русские традиции вёрстки.

Набор классов **hc** основывается на **KOMA-Script** и содержит классы **hcart** (статья), **hcreport** (отчёт), **hcletter** (письмо). Кроме перечисленного в набор входит простой класс для создания презентационных слайдов **hcsides**.

Набор **ntgclass** состоит из переписанных с нуля немецкоговорящими голландцами классов для набора статей (**artikel1**, **artikel2** и **artikel3**), книг (**boek** и **boek3**), писем (**brief**) и отчётов (**rapport1** и **rapport3**). Классы создавались с оглядкой на европейские традиции вёрстки.

Набор **extsizes** — небольшая модификация стандартных классов, добавляющая к стандартным 10pt, 11pt и 12pt размерам шрифта 8pt, 9pt, 14pt, 17pt и 20pt. Состоит из классов **extarticle**, **extbook**, **extletter**, **extproc**, **extreport**.
 ✓ **minimal** — этот класс используется в случае, если критична скорость инициализации, а оформление не нужно совсем.

17.10.1. Поддерживаем стандарты

✓ **eskdex** — реализация стандарта ЕСКД от Константина Корикова. Основу коллекции составляют классы: **eskdtext** (для текстовой документации), **eskdatab** (для чертежей и схем) и **eskdgraph** (для документов, разбитых на графы).

Пакет **isostds** содержит набор макросов для создания документации по ISO (**iso**) и реализацию ISO 10303 (**iso10303**).

17.10.2. Пишем письма и отсылаем факсы

Базовый класс для написания писем **letter** можно заменить **scrletter2** из набора классов **KOMA-Script** или из других аналогичных наборов.

adrconv и **directory** — наборы стилей, позволяющих держать адреса в базе данных BibT_EX и оперировать с ними.

adrlist — позволяет организовать простой цикл по специально сформированному списку адресов.

✓ **akletter** — полноценная превосходная замена стандартного класса **letter**.

beletter — бельгийские письма.

✓ **cdpbundl** — деловые письма в итальянском стиле.

envlab — разметка конвертов по правилам US Postal Service.

dinbrief и **g-brief** — немецкие письма. Специально для класса **dinbrief** была написана программа **dinbrief-gui** на Tcl/Tk.

facsimile и **fax** — оформление сообщений для отсылки через факс.

formlett и **mailing** — рассылка стандартных писем по множеству адресов.

lettre — французские письма.

✓ **newlfn** — изопрённый класс, позволяющий легко создавать свои стандарты писем. Интегрирует в себя пакеты **fancyhdr** и **geometry**.

17.10.3. Верстаем книги

Вместо стандартного класса **book** в обязательном порядке необходимо использовать более современную замену, например, **scrbook** из набора **KOMA-Script**.

✓ **memoir** — исключительно полно документированный класс для создания книг, в том числе и сильно ориентированных на математику.

✓ **octavo** — абсолютно всё для создания книг: от гибкого структурирования, до форматирования геометрии страницы и форматирования обложки.

✓ **sffms** — быстрый способ сделать распечатку текста через два интервала на дешёвом принтере.

17.10.4. Создаём отчёты

Базовый класс отчётов **report**, как правило, заменяется более современным из стандартных наборов, например, **scrreprt** из KOMA-Script.

✓ **hitec** — отчёт в стиле HiTech.

Пакет **refman** содержит классы **refart** и **refrep**, предназначенные для создания отчётов и документации с широкими полями для заметок по примеру документации, традиционно распространяемой Adobe.

17.10.5. Делаем презентации

✓ Базовый класс для создания слайдов для презентаций **slides** до сих пор может оказаться востребован в силу своей исключительной простоты.

a0poster — создание текста на очень большом листе бумаги.

✓ **beamer** — современный пакет для подготовки компьютерных презентаций с большим количеством стандартных презентационных стилей.

powerdot — современный презентационный класс с долгой историей.

prosper — более современный аналог **seminar** и соответственно **slides**. Вспомогательный класс **ppr-prv** позволяет получить печатную версию слайдов, подготовленных с помощью класса **prosper**.

sciposter — наиболее функциональный на текущий момент класс для подготовки постеров.

✓ **seminar** — чуть более современный аналог **slides** с оверлеями и разными эффектами.

talk — альтернатива **beamer** с возможностью смены оформления слайдов. На сегодня есть только один предопределённый тип слайда.

texpower — презентационный пакет, ориентированный на визуальные эффекты. Содержит класс **powersem**.

Кроме перечисленных классов можно обратить внимание на совместимый с классическим **seminar** пакет **ifmslide** — презентации с использованием **pdflatex**. Схожую функциональность обеспечивает **pdfslide**.

Для создания эффектов перехода между слайдами может оказаться полезным и пакет **ppower4**, который является постпроцессором для создания эффектов в PDF-презентациях.

17.10.6. Защищаем диссертации

Классов для написания диссертаций великое множество. Не зачем создавать ещё один подобный класс с нуля — есть куча примеров, на которые можно опереться.

✓ **disser** — в помощь русскоязычным студентам и аспирантам для написания работ для защиты звания бакалавра, магистра, кандидата и доктора наук. Спасибо Станиславу Кручинину. К сожалению, пакет отсутствует в составе дистрибутива

T_EX Live-2007, поэтому его следует взять на любом CTAN-архиве в директории {CTAN}/macros/latex/contrib/dissert.

ebsthesis — оформление диссертации в соответствии с правилами European Business School.

hepthesis — пакет общего назначения. Начинался как класс для PhD в области физики высоких энергий (HEP).

pittetd — University of Pittsburgh.

gatech-thesis — Georgia Institute of Technology.

muthesis — University of Man.

uaclasses — University of Arizona.

ucthesis — UC Berkeley.

uiucthesis — University of Illinois.

umich-thesis — University of Michigan.

uwthesis — University of Washington.

york-thesis — York University (Канада).

17.10.7. Организуем резюме

✓ **curve** — класс CurVe позволяет поддерживать несколько разных резюме в одном основном файле.

europescv — неофициальный класс для подготовки резюме по стандартам, рекомендованным Европейской комиссией.

moderncv — для создания резюме с современным дизайном.

В той или иной степени резюме можно создавать с помощью классов **vita**, **currvita** или **cv**.

17.10.8. Журнальные и конференционные классы

У многих конференций и журналов свои правила оформления статей, но в основном конференционные и журнальные классы — это небольшие модификации базовых.

aaai — конференции AAAI (American Association for Artificial Intelligence).

aastex или AAS_TEX — стиль для представления публикаций в журналах American Astronomical Society.

abstbook — быстрый способ «сложить» сборник из конференционных «абстракт-тов».

acmconf — конференционный класс для ACM (Association for Computing Machinery). Для подготовки списка литературы следует использовать стиль **acm**.

active-conf — конференции ACTIVE (Австралия).

aguplus — класс AGU_TEX для AGU (American Geophysical Union).

aiaa-tc — класс для конференций, проводимых AIAA (American Institute of Aeronautics and Astronautics). Имеет также свой библиографический стиль **aaai-named**.

✓ *AMS*-классы включают в себя **amsart** (статья), **amsbook** (книга) и **amsproc** (доклад).

apa — American Psychological Association. Для приведения в соответствие с требованиями ассоциации оформления ссылок на литературу используется пакет **apacite** и стили **apa** или **apalike**.

asaetr — American Society for Agricultural Engineers (ASAE).

ascelike — American Society of Civil Engineers. Одноимённый библиографический стиль прилагается.

cc — класс из пакета **computational-complexity** для журнала Computational Complexity.

✓ **combine** — подготовка трудов конференции. Объединение множества докладов в один документ.

✓ **elsart** — класс, обязательный при подготовке журнальных публикаций в издательстве Elsevier.

jpsj2 — Journal of the Physical Society of Japan.

iagproc — IAG (International Association of Geodesy).

IEEEconf и **IEEEtran** — классы для IEEE (Institute of Electrical and Electronics Engineers).

mtn — Maple Technical Newsletter.

nature — класс, позволяющий подготовить pdf-файл для редакции журнала Nature. К нему прилагается одноимённый стиль для подготовки библиографии.

✓ **nccproc** — труды конференций, проводящихся под эгидой Института вычислительной математики и математической геофизики.

nrc1 и **nrc2** — классы из пакета **nrc** для National Research Council (Канада).

✓ **revtex4** — один из самых популярных журнальных классов. В частности, он используется при оформлении статей для журналов МАИК «Наука/Интерпериодика» (<http://www.maik.ru/pub/tex/>).

ptptex — PTPeX (Progress of Theoretical Physics), Япония.

✓ **sibjnm** — «Сибирский журнал вычислительной математики».

siggraph — конференции Siggraph.

smfart и **smfbook** — классы французского математического общества.

spie — конференции SPIE.

17.10.9. Делаем газеты и буклеты

Создание газет не являются сильной стороной L^AT_EX, но ничто не запрещает попробовать и это.

✓ **papertex** — верстаем газету с помощью L^AT_EX.

leaflet — подготовка буклетов-раскладушек в шесть страниц на одном листе.

17.11. Справочно-поисковый аппарат издания

Написать любую книгу безумно тяжело. Сделать её полезной почти невозможно. Справочно-поисковый аппарат издания — это хорошее подспорье на этом пути.

17.11.1. Оглавление

- ✓ **minitoc** — позволяет создавать своё оглавление в каждой главе.
- multitoc** — позволяет сформировать оглавление в несколько колонок.
- ✓ **shorttoc** — позволяет создать ещё одно оглавление другой глубины.
- ✓ **tocbibind** — добавляет ссылки на библиографию, алфавитный указатель, списки таблиц и рисунков в оглавление.
- ✓ **tocloft** — позволяет управлять параметрами, отвечающими за форматирование оглавления, списков таблиц, рисунков и тому подобного.
- tocvsec2** — предоставляет дополнительные возможности по формированию записей оглавления и нумерации разделов.

17.11.2. Колонтитулы

- ✓ **fancyhdr** — полный контроль над колонтитулами.
- ✓ **nccfancyhdr** из пакета **ncctools** — улучшенная реализация пакета **fancyhdr**.
- ✓ **technics** — пример того, как с помощью **fancyhdr** нарисовать стандартную табличку на каждой странице сверху. Сам по себе стиль не рабочий — нужен напильник.

17.11.3. Библиография

- amsrefs** — альтернатива BibT_EX от A_MS.
- authorindex** — помогает создать список авторов всех цитируемых документов.
- backref** из пакета **hyperref** — создаёт интерактивную гиперссылку, позволяющую попасть из библиографического списка к месту ссылки.
- babelbib** — позволяет создавать и поддерживать библиографические ссылки на множестве языков с использованием пакета **babel**. Русский язык пока не поддерживается.
- bibcheck** из пакета **ltxmisc** — проверяет, на все ли элементы стандартного окружения thebibliography есть ссылки в документе.
- bibtopic** — включение нескольких файлов с библиографическими данными в один документ.
- bibunits** — создание нескольких библиографических списков для различных структурных единиц документа, например, для частей \part.
- ✓ **breakcites** из пакета **ltxmisc** — позволяет цитате \cite переноситься на другую строку. Полезно при громоздких множественных ссылках.
- camel** — ещё одна альтернатива BibT_EX в управлении библиографией.

cite — набор улучшающих стилей, состоящий из **cite** (автоматически группирует ссылки), **overcite** (автоматически группирует ссылки и формирует их как верхние индексы), **drftcite** (вместо номеров выводит метки — удобно при подготовке) и **chapterbib** (подключение нескольких файлов с библиографической информацией).

✓ **citehack** из пакета **t2** — позволяет использовать кириллицу в метках для команды цитирования `\cite`.

citeref — формирует список ссылок на библиографию.

✓ **compactbib** — позволяет использовать окружение `thebibliography` несколько раз.

custom-bib — облегчает создание своего библиографического стиля.

doipubmed — дополнительные полезные команды (`\doi`, `\pubmed` и `\citeurl`) для формирования библиографического списка.

footbib — размещение библиографической информации в виде подстрочных примечаний.

harvard — семейство библиографических стилей Harvard.

multibbl — создание нескольких параллельных библиографических списков.

multibib — помогает оформить несколько библиографических списков.

natbib — дополнительные библиографические стили. В основном для формирования ссылок вида «автор–год».

notoccite — исправление стандартного механизма цитирования библиографии при использовании цитат, появляющихся в оглавлениях или списках таблиц/рисунков при использовании В_IT_EX-стиля **unsrt**.

17.11.4. Указатель

✓ **gloss** — создание глоссария с использованием В_IT_EX.

index — расширение стандартного механизма создания предметных указателей.

✓ **makeidx** — стандартный пакет Л_AT_EX для создания предметных указателей.

✓ **makeglos** — создание глоссария.

✓ **nomenc1** — создание списка используемых символов. Иными словами — создание номенклатуры или списка обозначений.

robustindex — автоматическое переопределение номеров страниц в алфавитном указателе при изменении числа страниц документа. Позволяет в случае, если не добавляется дополнительной индексной ссылки, не пересоздавать указатель.

✓ **splitindex** — ещё один пакет для возможного увеличения числа указателей до бесконечности.

✓ **varindex** — задание нескольких индексных элементов одной командой. Полезно, когда хочется, чтобы одновременно были пункты и «Теорема Пикара», и «Пикар — теорема».

17.12. Программирование в среде L^AT_EX

Лучше всего по возможности найти уже готовое решение, но иногда приходится делать что-то новое. Для создания своих пакетов полезно воспользоваться уже имеющимися инструментами, чтобы не изобретать велосипед заново.

perltext — PerlT_EX. Позволяет использовать perl в среде L^AT_EX.

17.12.1. Счётчики и другие переменные

arrayjob — добавляет массивы в L^AT_EX и операции с ними.

binhex — перевод из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

✓ **chngcntr** — позволяет добавить или убрать зависимость после определения счётчика.

clefval — позволяет создавать связанные пары «ключ/значение» с доступом к значению по ключу. В какой-то мере это аналог хэша.

engord из пакета **oberdiek** — добавляет правильные английские суффиксы к числительным.

dcounter из пакета **ncctools** — счётчики с понятием стиля нумерации, или динамические счётчики.

fntcount — представляет различные форматы (двоичный, восьмеричный, шестнадцатеричный и так далее) отображения счётчиков.

sublabel из пакета **preprint** — позволяет счётчикам иметь дополнительную поднумерацию, то есть 4a, 4b, 4c и тому подобное.

zahl2string из пакета **sauerj** — при передаче числа формирует соответствующую строку текста на немецком.

17.12.2. Вычисления с использованием L^AT_EX

✓ **calc** — позволяет выполнять арифметические операции с длинами и счётчиками.

fltpoint — вычисления с плавающей точкой.

refcount из пакета **oberdiek** — предоставляет набор макросов, позволяющих выполнять операцию присваивания между счётчиками.

fp — набор функций, включающий тригонометрические, и операторов для вычислений с фиксированной запятой с высокой точностью.

17.12.3. Условные выражения и циклы

ifdraft из пакета **oberdiek** — предоставляет набор условных выражений для определения моды draft/final.

ifmtarg из пакета **ltxmisc** — условный оператор для определения существования аргумента. Аргумент отсутствует, если представляет собой ноль или более пробелов.

ifpdf из пакета **oberdiek** — предоставляет условное выражение `\ifpdf` для определения факта использования при компиляции **latex** или **pdflatex**.

✓ **ifthen** — макропакет, в котором определены команды `\ifthenelse` (условный переход) и `\whiledo` (цикл). Имеет смысл присмотреться к усовершенствованной версии этого пакета **xifthen**.

labelcas — проверка существования метки в документе.

multido — определяет оператор цикла `\multido`.

stdclsdv — набор условных выражений для определения используемого класса.

17.12.4. Создание новых команд

acronym — простой способ создания сокращений или акронимов.

makecmds — определяет новые макросы `\makecommand`, `\makeenvironment` и `\provideenvironment`, позволяющие задать новую команду, если она до этого отсутствовала, или переопределить, если уже ранее она была определена.

optparams из пакета **sauerj** — создание новой команды с необязательными параметрами. Число необязательных параметров может достигать до девяти.

robustcommand — создание «не хрупких» команд.

toolbox — пакет для создания новых типов предметных указателей, глоссариев и макросов.

twoopt из пакета **oberdiek** — позволяет создавать новые команды с двумя необязательными параметрами, которые имеют значение по умолчанию.

17.12.5. Разбор параметров

coolstr — работа со строками. Под строкой понимается набор ASCII-символов, то есть кириллица не работает. В пакете определена команда `\substr`, извлекающая подстроку от и до указанных позиций символов из строки, переданной в качестве параметра.

keyval — разбор списков вида «ключ=значение».

processkv из пакета **sauerj** — разбор списка пар «ключ=значение». При этом на каждой паре при разборе может выполняться определённый пользователем макрос.

✓ **tokenizer** — позволяет разбивать текстовые списки на элементы.

xkeyval — улучшенная версия пакета **keyval**. Пакет позволяет передавать и принимать в качестве параметра список пар «ключ=значение».

17.12.6. Работа с файлами

attachfile2 из пакета **oberdiek** — прикрепить файл к документу PDF.

fink — доступ к имени файла, который компилируется на момент исполнения команды `\finkfile`.

embedfile из пакета **oberdiek** — внедрить файл в документ PDF.

extract — в пакете определены окружения, позволяющие записать при компиляции указанный текст в файл.

filecontents — запись L^AT_EX-кода в файл с возможностью включить этот файл с помощью команды `\include`.

newfile — пользовательский интерфейс к чтению/записи файлов.

✓ **optional** — предоставляется возможность интерактивно указать при компиляции, какая часть документа будет использована, а какая пропущена.

versions — набор макросов, позволяющих пометить часть текста и выкинуть её при компиляции.

17.12.7. Пакеты L^AT_EX

afterpackage из пакета **ncctools** — добавляет команды к пакету, после того как он уже загружен.

✓ **doc** — пакет для создания новых пакетов. Включение кода и документации в один файл.

✓ **docstrip** — подготовка к использованию кода и документации, созданных с помощью пакета **doc**.

blindtext — предоставляет для целей тестирования искусственно сгенерированный текст на английском или немецком языках. Схожую функциональность имеет пакет **lipsum**.

diagnose — набор макросов для диагностирования ошибок.

dialogl — набор макросов для создания интерактивных L^AT_EX-скриптов.

export — экспорт/импорт переменных L^AT_EX во внешний файл.

makedtx — автоматизация создания пакетных dtx-файлов.

msg — пакет для локализации сообщений других пакетов. Иногда пользователю проще понять сообщение об ошибке, если оно выдаётся на его родном языке.

xdoc — расширение стандартного, но постепенно устаревающего, пакета **doc**.

Установка T_EX Live

На прилагаемом к книге CD записан дистрибутив T_EX Live-2007. Этот дистрибутив L^AT_EX всегда можно взять по адресу <http://www.tug.org/texlive/>.

Для того чтобы упаковать максимально возможный объём дистрибутива на один CD, число поддерживаемых дистрибутивом T_EX Live архитектур урезано с пятнадцати до трёх:

i386-linux	— GNU/Linux на x86;
powerpc-darwin	— MacOSX на PowerPC ¹ ;
win32	— 32-битные версии Windows, то есть начиная с 9x и далее.

На диске в директории `texmf-doc/doc/russian/texlive-ru/` находится подробная документация на русском языке об установке и поддержке дистрибутива в html- и pdf-форматах: `live.html` и `live.pdf` соответственно. Если установка этого дистрибутива производится в первый раз, то настоятельно рекомендуется изучить этот текст.

GNU/Linux

В современных версиях GNU/Linux, например, в Debian (Lenny), T_EX Live может идти как дистрибутив L^AT_EX по умолчанию. В этом случае нет необходимости заниматься установкой, хотя это ещё может зависеть от версии T_EX Live.

Для распаковки и установки дистрибутива в системах GNU/Linux или Mac OS X следует из корневой директории CD запустить скрипт:

```
> sh ./install-tl.sh
```

Программа установки работает в текстовом режиме. В процессе установки имеет смысл обратить особое внимание на директорию, куда будет произведена установка. По умолчанию предлагается установить на диск абсолютно всё. При этом будет занято около 1 Гб дискового пространства. Не следует на этом экономить, хотя всегда после установки можно будет добавить недостающие пакеты.

Для того чтобы в дальнейшем гарантировано использовать исполняемые файлы из установленного дистрибутива, следует добавить путь до них в начало списка директорий переменной PATH, например, так:

```
# для bash лучше в .bash_profile
> TEXTLIVE=/usr/local/texlive/2007/
> PATH=$TEXTLIVE/bin/i386-linux:$PATH
> export PATH
```

¹Несмотря на то, что T_EX Live можно использовать под Mac OS X, по возможности рекомендуется воспользоваться дистрибутивом MacT_EX (<http://www.tug.org/mactex/>). Дистрибутив T_EX Live включён в состав этого специализированного дистрибутива.

При редактировании tex-файла лучше воспользоваться любимым текстовым редактором. Все более-менее распространённые текстовые редакторы имеют поддержку L^AT_EX. Если на момент установки «любимый редактор» отсутствует, то следует присмотреться к родному из двух основных: **emacs**² или **vim**. И тот и другой заслуживают изучения, так как это больше, чем просто текстовые редакторы. Также может оказаться, что для целей редактирования удобнее будет воспользоваться специализированным программным пакетом Kile (KDE Integrated L^AT_EX Environment) или даже LyX (WYSIWYM текстовый процессор).

Windows³

Под операционной системой Windows наиболее популярен дистрибутив MiK_TE_X (<http://www.miktex.org/>). Так как все дистрибутивы L^AT_EX «растут» от одного корня CTAN, то обзор пакетов T_EX Live вполне применим и к обзору пакетов MiK_TE_X.

Для установки дистрибутива T_EX Live в среде Windows нужно воспользоваться программой установки **tlpmgui.exe**, которая находится в директории **setup-win32** установочного диска. Она также запускается при вставке CD, если в системе разрешён автозапуск.

Процесс установки не сложен. При запуске **tlpmgui.exe** следует выбрать полную установку (**scheme-full**) или, так как по умолчанию поддержка русского языка отсутствует, выбрать русский язык вручную с помощью кнопки «Language Collection». После того как будет нажата кнопка **Install**, возникнет запрос на установку пакета **perl** — это необходимо, если поддержки данного языка в системе нет. После положительного ответа на этот вопрос можно уйти попить чая часа на полтора (время зависит от мощности компьютера). В конце установки программа поинтересуется, надо ли устанавливать GhostScript для работы с PostScript и программу просмотра dvi-файлов — **dviout**. Если эти пакеты в системе отсутствуют, то их надо установить.

После полной установки имеет смысл перезагрузиться. Для более подробной информации следует обратиться к документации по установке, упомянутой в начале приложения.

Кроме непосредственно установки дистрибутива L^AT_EX в случае Windows необходимо добавить некоторое количество сторонних программ для просмотра и редактирования.

- **gsview** (<http://www.ghostgum.com.au/>). Это бесплатная программа для просмотра файлов в формате PostScript. Исходники программы есть в открытом доступе, но бинарная сборка с авторского сайта включает всплы-

²Поддержка L^AT_EX реализована в стандартных расширениях AucT_EX и RefT_EX. Визуализация формул и картинок реализована в пакете **preview-latex**.

³Если вы планируете использовать L^AT_EX, то, возможно, имеет смысл приглядеться к GNU/Linux в качестве основной платформы. Но это уже совсем другая история ©.

вающее окно при каждом запуске программы с настоятельной просьбой зарегистрироваться.

- **Adobe Reader** (<http://www.adobe.com/products/reader/>) — кроссплатформенная бесплатная программа просмотра pdf-файлов.
- Редактор для редактирования tex-исходников. Варианты:
 - **TeXnicCenter** (<http://www.toolscenter.org/>) — свободный редактор с частично открытыми исходниками. После установки придётся настроить как минимум просмотр dvi-файлов;
 - **WinEdt** (<http://www.winedt.com/>) — популярный ShareWare текстовый редактор. Начиная с версии 5.5, сконфигурирован в том числе и для работы с T_EX Live-2007;
 - **Notepad++** (<http://notepad-plus.sourceforge.net/ru/site.htm>) — простейший тестовый редактор, который только редактирует текст;
 - можно установить тот же **emacs** или **vim**, как в GNU/Linux.

В дистрибутиве T_EX Live-2007 за год его существования было обнаружено некоторое количество ошибок. Известные проблемы перечислены на страничке дистрибутива <http://tug.org/texlive/bugs.html>.

► К сожалению, одна из этих проблем напрямую связана с использованием кириллицы. Автоматически не исполняется **mktexmf** при запуске T_EX-программ. При этом в частности возникают проблемы с использованием кириллических шрифтов **lh**.

В качестве обходного решения предлагается изменить настройки по умолчанию в файле **texmf.cnf**. Нужно установить следующие значения для переменных:

```

MKTEXPK = 1
MKTEXMF = 1
MKTEXTFM = 1
MKTEXFMT = 1

```

Этот конфигурационный файл по умолчанию располагается в директории **/TeXLive2007/texmf-var/web2c/**. Его также можно найти с помощью команды

```
> kpsewhich texmf.cnf
```

Ошибки проявляются всё равно. Это происходит при первой компиляции текста с помощью **latex**, когда автоматически генерируются ещё ни разу не использованные шрифты. Но если эти ошибки игнорировать⁴, то dvi-файл получается нормальным. При последующих компиляциях, когда шрифты уже в наличии, проблем не возникает.

⁴Если компиляция происходит в командной строке, то на запрос после информации об ошибке следует нажать клавишу Q и перевод строки (to run quietly). Если компиляция происходит средствами редактора, то ничего этого делать не надо.

Указатель команд и окружений

- A**
- `\AA`, 125
 - `\aa`, 125
 - `abstract`, 41, 98
 - `\act`, 237
 - `\acute`, 188
 - `\addcharacter`, 237
 - `\addcontentsline`, 135, 165
 - `\addtocontents`, 164
 - `\addtocounter`, 72, 75
 - `\addtoendnotes`, 140
 - `\addtolength`, 70, 75
 - `\AE`, 125
 - `\ae`, 125
 - `\afterpage`, 63
 - `\againframe`, 119
 - `\aleph`, 186
 - `\alert`, 113, 116
 - `algorithm`, 215
 - `algorithmic`, 214, 215
 - `align`, 200–202
 - `align*`, 200, 201
 - `alignat`, 201
 - `aligned`, 202
 - `alignedat`, 202
 - `\allowdisplaybreaks`, 200
 - `\alph`, 72
 - `\alpha`, 50, 187
 - `\alt`, 116
 - `\amalg`, 190
 - `\AND`, 77
 - `\and`, 41
 - `\ang`, 223
 - `\angle`, 51, 186
 - `\appendix`, 135
 - `\apprge`, 193
 - `\apple`, 193
 - `\approx`, 51, 191
 - `\approxeq`, 191
 - `\arabic`, 72
 - `\arccos`, 54
 - `\arctg`, 54
 - `\arcsin`, 54
 - `\arctg`, 54
 - `\areaset`, 85
 - `array`, 151, 197, 266
 - `\Asbuk`, 72
 - `\asbuk`, 72
 - `\AskOption`, 147
 - `\ast`, 190
 - `\asympt`, 191
 - `\ataribox`, 124
 - `\author`, 41, 251
- B**
- `\backepsilon`, 191
 - `\backmatter`, 136
 - `\backprime`, 186
 - `\backsim`, 191
 - `\backsimeq`, 191
 - `\backslash`, 186, 194
 - `\backslashbox`, 156
 - `\balance`, 92
 - `\bar`, 51, 188, 240, 241
 - `\barwedge`, 190
 - `\baselineskip`, 90, 91
 - `\Bbbk`, 186
 - `\bcancel`, 189, 255
 - `\beamerbutton`, 113, 115, 116
 - `\because`, 191
 - `\begin`, 31

\bell, 124
 \beta, 50, 187
 \beth, 186
 \between, 191
 \bfseries, 36
 \bibitem, 168
 \bibliography, 171

\Big, 54
 \big, 54
 \bigcap, 195
 \bigcirc, 190
 \bigcup, 195
 \Bigg, 54
 \bigg, 54

\bigodot, 195
 \bigoplus, 195
 \bigotimes, 195
 \bigskip, 91
 \bigsqcup, 195
 \bigstar, 186
 \bigtriangledown, 190

\bigtriangleup, 190
 \biguplus, 195
 \bigvee, 195
 \bigwedge, 195
 \binom, 199
 \bitbox, 213
 \bitheader, 213

\blacklozenge, 186
 \blacksmiley, 124
 \blacksquare, 186
 \blacktriangle, 186
 \blacktriangledown, 186
 \blacktriangleleft, 192
 \blacktriangleright, 192

block, 107, 115, 116
 blockarray, 266
 bmatrix, 197
 \bond, 230
 \boolean, 76
 \bot, 186
 \Bowtie, 124
 \bowtie, 191

\Box, 186, 187
 \boxdot, 190
 \boxminus, 190
 \boxplus, 190
 \boxtimes, 190
 \bra, 221
 \breve, 188
 \brokenvert, 124
 \bullet, 190
 \Bumpeq, 191
 \bumpeq, 191
 \burl, 142, 143
 bytfield, 212, 213

C

\cancel, 189, 255
 \cancelto, 189
 \Cap, 190
 \cap, 190
 \caption, 43, 66, 137, 154, 215, 263
 cases, 203

\cbend, 141
 \cbstart, 141
 CD, 198
 \cdot, 51, 190
 \cdotp, 189
 \cdots, 189
 \ce, 230, 231

\cent, 125
 center, 89
 \centerdot, 190
 \centering, 90

\ceqn, 47
 \cftchapidotsep, 164
 \cftchapidleader, 164
 \cftdot, 164
 \cftdotfill, 164
 \cftdotsep, 164
 \cftsecnumwidth, 164
 \ch, 54
 \chapter, 30, 41, 98, 130, 135
 \check, 188
 \checked, 124

`\CheckBox`, 124
`\checkmark`, 125
`\chi`, 50, 187
`\circ`, 190
`\circeq`, 191
`\circlearrowleft`, 193
`\circlearrowright`, 193
`\circledast`, 190
`\circledcirc`, 190
`\circleddash`, 190
`\circledR`, 125, 186
`\circledS`, 186
`circuit`, 227
`\cite`, 168, 169, 274, 275
`\citeurl`, 275
`\clearpage`, 63, 145
`\clock`, 124
`\clubsuit`, 186
`\cnttest`, 76, 77
`code`, 257
`\colon`, 189
`\color`, 114, 115, 211
`\columnbreak`, 94
`\columncolor`, 156
`\columnwidth`, 81
`\COMMENT`, 215
`comment`, 35
`\complement`, 186
`\cong`, 191
`\coprod`, 195
`\cos`, 54
`\counterwithin`, 72
`\counterwithout`, 72
`\ctable`, 159
`\ctg`, 54
`\cth`, 54
`\Cup`, 190
`\cup`, 190
`\curlyeqprec`, 191
`\curlyeqsucc`, 191
`\curlyvee`, 190
`\curlywedge`, 190
`\currency`, 125

`\curvearrowleft`, 193
`\curvearrowright`, 193

D

`\dag`, 125
`\dagger`, 190
`\daleth`, 186
`\dashleftarrow`, 193
`\dashrightarrow`, 193
`\dashv`, 191
`\date`, 41
`\ddag`, 125
`\ddagger`, 190
`\dddots`, 53, 188
`\ddot`, 53, 188
`\ddots`, 189
`\DeclareMathOperator`, 54
`\definecolor`, 211
`\DefineLCDchar`, 212
`\Delta`, 50, 187
`\delta`, 50, 187
`\depth`, 129
`\depthof`, 75
`desclist`, 132
`description`, 131, 132, 249, 250
`\diagdown`, 186
`\Diagram`, 224
`\diagup`, 186
`\dialog`, 237
`\diameter`, 124
`\Diamond`, 186, 187
`\diamond`, 190
`\diamondsuit`, 186
`\Dif`, 195
`\dif`, 195
`\diff`, 196
`\diffp`, 196
`\digamma`, 187
`\dimtest`, 77
`\displaybreak`, 200
`\displaystyle`, 47, 196
`\div`, 51, 190
`\divideontimes`, 190

$\backslash$ DJ, 125
 $\backslash$ dj, 125
 document, 35
 $\backslash$ documentclass, 26, 32, 33, 97
 $\backslash$ doi, 275
 $\backslash$ dominilof, 165
 $\backslash$ dominilot, 165
 $\backslash$ dominitoc, 165
 $\backslash$ dot, 51, 53, 188
 $\backslash$ doteq, 191
 $\backslash$ doteqdot, 191
 $\backslash$ dotfill, 89, 161
 $\backslash$ dotlessi, 255
 $\backslash$ dotlessj, 255
 $\backslash$ dotplus, 190
 $\backslash$ dots, 125, 189
 $\backslash$ doublebarwedge, 190
 $\backslash$ doublespacing, 91
 $\backslash$ DOWNarrow, 124
 $\backslash$ Downarrow, 193, 194
 $\backslash$ downarrow, 193, 194
 $\backslash$ downarrowarrows, 193
 $\backslash$ downharpoonleft, 193
 $\backslash$ downharpoonright, 193
 $\backslash$ dropchapter, 131

E

$\backslash$ ell, 51, 186
 $\backslash$ ELSE, 214
 $\backslash$ ELSIF, 214
 $\backslash$ emph, 36, 113
 empheq, 203
 $\backslash$ emptyset, 186
 $\backslash$ EnColon, 244
 $\backslash$ end, 31, 35
 $\backslash$ endextract, 239
 $\backslash$ ENDFOR, 214
 $\backslash$ ENDIF, 214
 $\backslash$ endinput, 145
 $\backslash$ ENDLOOP, 214
 $\backslash$ endnote, 140
 $\backslash$ endnotemark, 140
 $\backslash$ endnotetext, 140

$\backslash$ endpiece, 239
 $\backslash$ endrotboxpages, 83
 $\backslash$ endspace, 89
 $\backslash$ ENDWHILE, 214
 $\backslash$ enotes, 239–241
 $\backslash$ ENSURE, 215
 $\backslash$ ensuremath, 73
 enumerate, 37, 115, 131, 132, 249
 $\backslash$ epigraph, 130
 $\backslash$ epigraphhead, 130
 $\backslash$ epigraphwidth, 130
 $\backslash$ epsilon, 187
 $\backslash$ eqcirc, 191
 eqlist, 132
 eqlist*, 132
 $\backslash$ eqname, 255
 $\backslash$ eqref, 46, 137, 204, 251
 $\backslash$ eqslantgtr, 192
 $\backslash$ eqslantless, 192
 $\backslash$ equal, 76
 equation, 46, 47, 137, 200
 equation*, 47, 48, 200
 $\backslash$ equiv, 51, 191
 $\backslash$ eta, 50, 187
 $\backslash$ eth, 186
 $\backslash$ eval, 195
 $\backslash$ evensidemargin, 81
 example, 107
 $\backslash$ excludeonly, 146
 $\backslash$ exists, 51, 186
 $\backslash$ exp, 54

F

$\backslash$ faktor, 256
 $\backslash$ fallingdotseq, 191
 $\backslash$ fbox, 129, 199
 $\backslash$ Feyn, 224
 $\backslash$ feyn, 224
 figure, 62, 63, 137, 262
 $\backslash$ figureref, 251
 $\backslash$ finkfile, 277
 $\backslash$ Finv, 187
 $\backslash$ fixme, 143

`flalign`, 201
`flalign*`, 201
`\flat`, 186
`\fleqn`, 47
`\FloatBarrier`, 64
`floatingfigure`, 65, 66
`\flushend`, 92
`flushleft`, 89
`\flushleft`, 89, 90
`flushright`, 89
`\flushright`, 89
`\fmf`, 225
`fmffile`, 226
`fmfgraph`, 225
`\fmflabel`, 225
`\fmfleft`, 225
`\fmfright`, 225
`\footnote`, 138, 139
`\footnotemark`, 138
`\footnotesize`, 37
`\footnotetext`, 138
`\footskip`, 81
`\FOR`, 214
`\FORALL`, 214
`\forall`, 51, 186
`\foreignlanguage`, 122
`\frac`, 52, 53, 196
`frame`, 111, 112, 116
`\framebox`, 129
`\framesubtitle`, 112, 114
`\frametitle`, 107, 112, 114
`\frontmatter`, 136
`\frown`, 191
`\frownie`, 124
`\fullfunction`, 195
`\fxerror`, 143
`\fxnote`, 143
`\fxwarning`, 143

G

`\galso`, 181
`\Game`, 187
`\Gamma`, 50, 187

`\gamma`, 50, 187
`gather`, 201
`gather*`, 201
`gathered`, 202
`\ge`, 51, 191
`\generalmeter`, 240, 241
`\generalsignature`, 240, 241
`\geneuro`, 125
`\geneuronarrow`, 125
`\geneurowide`, 125
`\geq`, 191
`\geqq`, 192
`\geqslant`, 51, 192
`\GetTokens`, 77
`\gg`, 51, 191
`\ggg`, 192
`\gimel`, 187
`\gloss`, 181
`\glossary`, 180, 181
`\glossaryintro`, 181
`\glossaryname`, 180, 181
`\gnapprox`, 192
`\gneq`, 192
`\gneqq`, 192
`\gnsim`, 192
`\grave`, 188
`\gsee`, 181
`\gtrapprox`, 192
`\gtrdot`, 192
`\gtreqless`, 192
`\gtreqqlless`, 192
`\gtrless`, 192
`\gtrsim`, 192
`guitar`, 239
`\gvertneqq`, 192

H

`\hat`, 51, 188
`\hbar`, 186
`\hdotsfor`, 197
`\headheight`, 81
`\headsep`, 81
`\heartsuit`, 186

$\backslash$ height, 129
 $\backslash$ heightof, 75
 $\backslash$ HepAntiParticle, 224
 $\backslash$ HepGenAntiParticle, 224
 $\backslash$ HepGenParticle, 224
 $\backslash$ HepGenSusyParticle, 224
 $\backslash$ HepParticle, 224
 $\backslash$ HepParticleResonance, 224
 $\backslash$ HepProcess, 224
 $\backslash$ HepSusyParticle, 224
 $\backslash$ HepTo, 224
 $\backslash$ hexstar, 124
 $\backslash$ hfil, 89
 $\backslash$ hfill, 89
 $\backslash$ hhline, 155
hieroglyph, 244, 245
 $\backslash$ hline, 153
 $\backslash$ hm, 55
 $\backslash$ hoffset, 81
 $\backslash$ hookleftarrow, 193
 $\backslash$ hookrightarrow, 193
 $\backslash$ hphantom, 199, 200
 $\backslash$ hrule, 128
 $\backslash$ hrulefill, 89, 128
 $\backslash$ hslash, 187
 $\backslash$ hspace, 89, 91, 199
 $\backslash$ hspace*, 89
 $\backslash$ Huge, 37
 $\backslash$ huge, 37
 $\backslash$ hyperlink, 116
 $\backslash$ hyphenation, 39, 40

I

$\backslash$ idotsint, 195
 $\backslash$ IF, 214
 $\backslash$ iflanguage, 123
 $\backslash$ ifpdf, 277
 $\backslash$ ifthenelse, 76, 277
 $\backslash$ iiiint, 195
 $\backslash$ iiint, 195, 196
 $\backslash$ iint, 195, 196
 $\backslash$ Im, 186, 187
 $\backslash$ imath, 186

$\backslash$ in, 51, 186
 $\backslash$ include, 145, 146, 253, 254, 278
 $\backslash$ includegraphics, 60–62, 115
 $\backslash$ includeonly, 119, 145, 146, 254
 $\backslash$ indeces, 198
 $\backslash$ indent, 89
 $\backslash$ index, 174–176
 $\backslash$ infy, 51, 186
inparaenum, 133
inparaitem, 37, 133
 $\backslash$ input, 145, 174, 254
 $\backslash$ int, 53, 195, 196
 $\backslash$ intercal, 190
 $\backslash$ intertext, 203
 $\backslash$ invdiameter, 124
 $\backslash$ invisible, 116
 $\backslash$ invneg, 193
 $\backslash$ iota, 50, 187
IPA, 243
 $\backslash$ isempty, 77
 $\backslash$ isequivalentto, 77
 $\backslash$ isnamedefined, 77
 $\backslash$ isodd, 76
 $\backslash$ isotope, 226
 $\backslash$ isundefinedcommand, 76
 $\backslash$ item, 37, 43, 115, 131, 132
itemize, 37, 115, 131
 $\backslash$ itshape, 36

J

$\backslash$ jmath, 186
 $\backslash$ Join, 193

K

$\backslash$ kappa, 187
 $\backslash$ ket, 221
 $\backslash$ kill, 151
 $\backslash$ kreuz, 124

L

$\backslash$ L, 125
 $\backslash$ l, 125
 $\backslash$ label, 42, 43, 46, 116, 137, 215, 251
 $\backslash$ Lambda, 50, 187

- `\lambda`, 50, 187
- `\land`, 51
- `\landscape`, 82, 267, 268
- `\landscape`, 82
- `\langle`, 194
- `\LARGE`, 37
- `\Large`, 37
- `\large`, 37
- `\LaTeX`, 30
- `\layout`, 79, 80
- `\LCD`, 212
- `\LCDcolors`, 211
- `\lceil`, 194
- `\ldotp`, 189
- `\ldots`, 40, 189
- `\le`, 51, 191
- `\leadsto`, 193
- `\left`, 54, 55
- `\LEFTarrow`, 124
- `\Leftarrow`, 193
- `\leftarrow`, 193
- `\leftarrowtail`, 193
- `\leftharpoondown`, 193
- `\leftharpoonup`, 193
- `\leftleftarrows`, 193
- `\Leftrightarrow`, 193
- `\leftrightarrow`, 193
- `\leftrightarrows`, 193
- `\leftrightharpoons`, 193
- `\leftrightsquigarrow`, 193
- `\leftthreetimes`, 190
- `\leftwordgroup`, 214
- `\lengthtest`, 76
- `\leq`, 191
- `\leqq`, 192
- `\leqslant`, 51, 192
- `\lessapprox`, 192
- `\lessdot`, 192
- `\lesseqgtr`, 192
- `\lesseqqgtr`, 192
- `\lessgtr`, 192
- `\lesssim`, 192
- `\lfloor`, 194
- `\LHD`, 191
- `\lhd`, 190, 191
- `\lightning`, 124
- `\lim`, 54, 186
- `\limits`, 53
- `\linebreak`, 88
- `\linelabel`, 144
- `linenumbers`, 143
- `\linenumbers`, 144
- `\linewidth`, 81
- `\listfiles`, 146
- `listliketab`, 161
- `\listoffigures`, 135, 161, 164
- `\listoffixmes`, 143
- `\listoftables`, 135, 161, 164
- `\ll`, 51, 191
- `\llcorner`, 194
- `\Lleftarrow`, 193
- `\lll`, 192
- `\ln`, 54
- `\lnapprox`, 192
- `\lneq`, 192
- `\lneqq`, 192
- `\lnsim`, 192
- `\log`, 54
- `\logof`, 193
- `\longitem`, 132
- `\Longleftarrow`, 193
- `\longleftarrow`, 193
- `\Longleftrightarrow`, 193
- `\longleftrightarrow`, 193
- `\longmapsto`, 193
- `\longpage`, 90
- `\Longrightarrow`, 193
- `\longrightarrow`, 193
- `longtable`, 159
- `\LOOP`, 214
- `\looparrowleft`, 193
- `\looparrowright`, 193
- `\lor`, 51
- `\lozenge`, 187
- `\lrcorner`, 194
- `\Lsh`, 193

`\lstdefineline`, 211
`\lstineline`, 210
`\lstinputlisting`, 210
`\lstlisting`, 210
`\lstloadlanguages`, 209
`\lstset`, 209
`\ltimes`, 190
`\lVert`, 194
`\lvert`, 194
`\lvertneqq`, 192
`\lyrraise`, 241

M

`\mainmatter`, 136
`\makeatletter`, 71
`\makeatother`, 71
`\makebox`, 129
`\makecell`, 157, 266
`\makecommand`, 277
`\makeenvironment`, 277
`\makegloss`, 180
`\makeglossary`, 180
`\makeindex`, 174
`\makenomenclature`, 182
`\makepagenote`, 140
`\maketitle`, 41, 251
`\maltese`, 125
`\mapsto`, 193
`margincap`, 67
`\marginnote`, 141
`\marginpar`, 141
`\marginparpush`, 81
`\marginparsep`, 81
`\marginparwidth`, 81
`\markright`, 166
`\mathbb`, 50
`\mathbcal`, 50
`\mathbf`, 50
`\mathbin`, 185
`\mathclose`, 186
`\mathdollar`, 186
`\mathellipsis`, 186
`\mathfrak`, 50

`\mathindent`, 47
`\mathinner`, 186
`\mathop`, 186
`\mathopen`, 186
`\mathord`, 186
`\mathparagraph`, 186
`\mathpunct`, 186
`\mathrel`, 185
`\mathring`, 188
`\mathsection`, 186
`\mathsterling`, 186
`\mathstrut`, 52
`\mathunderscore`, 186
`matrix`, 197
`\max`, 54
`\maxwidth`, 71
`\maybebm`, 256
`\maybeit`, 256
`\mbox`, 40, 129
`\mboxfill`, 128
`\md`, 195
`\measuredangle`, 187
`\medskip`, 91
`\medspace`, 199
`\meterfrac`, 240, 241
`\mho`, 186
`\mid`, 191
`\min`, 54
`\minilof`, 165
`\minilot`, 165
`minipage`, 64, 65, 130, 138, 159, 160
`\minitoc`, 165
`\models`, 191
`\movie`, 119
`\mp`, 51, 190
`mpsupertabular`, 159
`mpxtabular`, 159
`\mtcskip`, 165
`\mu`, 50, 187
`multicols`, 93, 94
`\multicolumn`, 155, 156
`\multido`, 277
`\multimap`, 191

`\multirow`, 155, 156
`multline`, 201, 202
`music`, 239–241

N

`\nabla`, 186
`\natural`, 186
`\ncong`, 192
`\ne`, 191
`\nearrow`, 193
`\neg`, 51, 186
`\negmedspace`, 199
`\negthickspace`, 199
`\neq`, 51, 191
`\newblock`, 168
`\newcommand`, 74, 75, 218
`\newcounter`, 71
`\newenvironment`, 74
`\newindex`, 177
`\newlength`, 69
`\newpage`, 90
`\newtheorem`, 114, 205, 206
`\newtie`, 126
`\nexists`, 187
`\nextnum`, 161
`\ngeq`, 192
`\ngeqq`, 192
`\ngeqslant`, 192
`\ngtr`, 192
`\ni`, 186
`\nLeftarrow`, 194
`\nleftarrow`, 194
`\nLeftrightarrow`, 194
`\nleftrightarrow`, 194
`\nleq`, 192
`\nleqq`, 192
`\nleqslant`, 192
`\nless`, 192
`\nmid`, 192
`\No`, 32
`\nocite`, 169
`\noindent`, 89
`\nolimits`, 53

`\nolinenumbers`, 144
`\nomenclature`, 183
`\normalmarginpar`, 141
`\normalsize`, 36, 37
`\NOT`, 77
`\not`, 191
`\notag`, 204
`\notes`, 239–241
`\notesname`, 140
`\notin`, 191
`\nparallel`, 192
`\nprec`, 192
`\npreceq`, 192
`\nrightarrow`, 194
`\nrightharpoonup`, 194
`\nshortmid`, 192
`\nshortparallel`, 192
`\nsim`, 192
`\nsubseteq`, 192
`\nsucc`, 192
`\nsucceq`, 192
`\nsupseteq`, 192
`\nsupseteqq`, 192
`\ntriangleleft`, 192
`\ntrianglelefteq`, 192
`\ntriangleright`, 192
`\ntrianglerighteq`, 192
`\nu`, 50, 187
`\nulide`, 198
`\num`, 223
`numcases`, 203, 255
`\numprint`, 157
`\nVDash`, 192
`\nvDash`, 192
`\nvdash`, 192
`\nwarrow`, 193

O

`\O`, 125
`\o`, 125
`\ocircle`, 191
`\od`, 195
`\oddsidemargin`, 81

`\odot`, 190
`\OE`, 125
`\oe`, 125
`\officialeuro`, 125
`\oint`, 195
`\Omega`, 50, 187
`\omega`, 50, 187
`\ominus`, 190
`\onecolumn`, 92
`onehalfspacing`, 91
`\onehalfspacing`, 91
`\only`, 116
`\opensd`, 237
`\oplus`, 190
`\opt`, 146
`\OR`, 77
`\oslash`, 190
`otherlanguage`, 122
`otherlanguage*`, 122
`\otimes`, 190
`\overarc`, 245, 260
`\overbrace`, 189
`\overleftarrow`, 189
`\overleftrightarrow`, 194
`\overline`, 189
`\overrightarrow`, 189

P

`\P`, 125
`\pagebreak`, 90
`\pagenote`, 140
`\pageref`, 43, 137
`\pagestyle`, 166
`\paperheight`, 81, 85
`\paperwidth`, 81, 85
`\par`, 36, 74
`\paragraph`, 133
`Parallel`, 94
`\parallel`, 51, 191
`\ParallelText`, 95
`\ParallelRText`, 95
`parbox`, 130
`\parbox`, 64, 65

`\part`, 41, 135, 274
`\partial`, 53, 186
`\pause`, 113, 237
`\pd`, 195
`\permil`, 124
`\perp`, 51, 191
`\pgfpagesuselayout`, 87
`\phantom`, 200
`\Phi`, 50, 187
`\phi`, 187
`\phone`, 124
`\Pi`, 50, 187
`\pi`, 50, 187
`picture`, 264, 265
`\pitchfork`, 191
`\pm`, 51, 190
`pmatrix`, 197
`\poemtitle`, 236
`\poemtoc`, 236
`\pointer`, 124
`\poptabs`, 151
`portrait`, 82
`\portrait`, 82
`\pounds`, 125
`\prec`, 191
`\precapprox`, 191
`\preccurlyeq`, 191
`\preceq`, 191
`\precnapprox`, 192
`\precnsim`, 192
`\precsim`, 192
`\prettyref`, 138
`\prime`, 186
`\PRINT`, 215
`\printglossary`, 180
`\printindex`, 174
`\printnomenclature`, 182
`\printnotes`, 140
`\printnotes*`, 140
`Problem`, 73
`\prod`, 51, 195
`proof`, 107, 115, 255
`\propto`, 51, 191

`\protect`, 30, 46, 66, 135, 138, 164
`\providecommand`, 74
`\provideenvironment`, 277
 pseudocode, 216
`\Psi`, 50, 187
`\psi`, 50, 187
`\pubmed`, 275
`\pushtabs`, 151

Q

`\qedhere`, 115
`\qqquad`, 89, 199
`\quad`, 70, 89, 199
 quotation, 130
 quote, 130

R

`\raggedend`, 92
`\raggedleft`, 90
`\RaggedRight`, 90, 94
`\raggedright`, 90
`\rangle`, 194
`\rceil`, 194
`\Re`, 186, 187
`\real`, 76
`\recorder`, 124
`\ref`, 43, 73, 137, 144, 204, 251
`\refstepcounter`, 73
`\renewcommand`, 74
`\renewenvironment`, 75
`\REPEAT`, 214
`\REQUIRE`, 214
`\RETURN`, 215
`\reversemarginpar`, 141
`\rfloor`, 194
`\RHD`, 191
`\rhd`, 190, 191
`\rho`, 50, 187
`\right`, 54, 55
`\RIGHTarrow`, 124
`\Rightarrow`, 193
`\rightarrow`, 193
`\rightarrowtail`, 193

`\rightharpoondown`, 193
`\rightharpoonup`, 193
`\rightleftarrows`, 193
`\rightleftharpoons`, 193
`\rightrightarrows`, 193
`\rightsquigarrow`, 193
`\rightthreetimes`, 190
`\rightwordgroup`, 213
`\risingdotseq`, 192
`\Roman`, 72
`\roman`, 72
`\rotboxpages`, 83
`\rowcolor`, 156
`\Rsh`, 193
`\rtimes`, 190
`\rule`, 128
`\rVert`, 194
`\rvert`, 194

S

`\S`, 32, 125
`\saveFN`, 139
`\scene`, 237
`\scriptscriptstyle`, 47
`\scriptsize`, 37
`\scriptstyle`, 47
`\scshape`, 36
`\searrow`, 193
 section, 112
`\section`, 30, 98, 133, 135, 164
`\section*`, 42
`\sectionref`, 251
`\see`, 176
`\selectlanguage`, 122
 semiverbatim, 116
`\setcounter`, 75, 135
`\setlength`, 69, 70, 75, 130
`\setlyrics`, 241, 242
`\setmarginrb`, 84
`\setmarginohfrb`, 84
`\setmargrb`, 84
`\setminus`, 190
`\setnewcounter`, 72

$\setpagewiselinenumbers$, 144
 $\settodepth$, 71, 129
 $\settoheight$, 71, 129
 $\settowidth$, 70, 129
 $\SetWatermarkAngle$, 145
 $\SetWatermarkFontSize$, 145
 $\SetWatermarkLightness$, 144
 $\SetWatermarkScale$, 144
 $\SetWatermarkText$, 144
 $\sh$, 54
 $\shabox$, 129, 253
 $\sharp$, 186
 $\shortmid$, 192
 $\shortpage$, 90
 $\shortparallel$, 192
 $\shorttableofcontents$, 164
 $\shoveleft$, 202
 $\shoveright$, 202
 $\SI$, 223
 $\sidefig$, 66
 Σ , 50, 187
 σ , 50, 187
 $\sim$, 51, 191
 $\simeq$, 51, 191
 $\sin$, 54, 186
 $\sindex$, 177
 singespacing , 91
 singespacing , 91
 skippedwords , 213
 slashed , 258
 slide , 108
 slshape , 36
 small , 37
 smallfrown , 192
 smallsetminus , 191
 smallskip , 91
 smallsmile , 192
 smile , 191
 smiley , 124
 so , 127, 128
 sound , 119
 spacing , 91
 spadesuit , 186
 spcheck , 188
 spdddot , 188
 spddot , 188
 spdot , 188
 sphat , 188
 sphericalangle , 51, 187
 split , 202
 sptilde , 188
 sqcap , 190
 sqcup , 190
 sqrt , 51, 52
 sqsubset , 192, 193
 sqsubseteq , 191
 sqsupset , 192, 193
 sqsupseteq , 191
 Square , 124
 square , 187
 SS , 125
 ss , 125
 st , 127
 stage , 237
 star , 190
 startextract , 239
 startpiece , 239–241
 STATE , 214
 stepcounter , 72, 73
 stretchwith , 128
 strip , 92, 93
 subequations , 204, 256
 subnumcases , 255
 subparagraph , 133
 subsection , 30, 133, 135
 Subset , 192
 subset , 191
 subseq , 191
 subseqq , 192
 subsetneq , 192
 subsetneqq , 192
 substr , 277
 subsubsection , 133
 succ , 191
 succapprox , 192
 succcurlyeq , 192

`\succeq`, 191
`\succapprox`, 192
`\succnsim`, 192
`\succsim`, 192
`\sum`, 51, 53, 186, 195
`\sun`, 124
`supertabular`, 159
`\suppressfloats`, 63
`\Supset`, 192
`\supset`, 191
`\supseteq`, 191
`\supseteqq`, 192
`\supsetneq`, 192
`\supsetneqq`, 192
`\surd`, 186
`\SVN`, 219
`\SVNDate`, 219
`\SVNKeyword`, 219
`\SVNTime`, 219
`\swarrow`, 193
`\switchlinenumbers`, 144

T

`tabbing`, 150, 151
`table`, 152, 262
`\tableofcontents`, 41, 112, 133, 161, 164
`tabular`, 137, 151–153, 158, 266, 267
`tabularx`, 158
`\tag`, 204
`\tag*`, 204
`\tau`, 50, 187
`\tensor`, 198
`\text`, 48, 222, 223, 254
`\textasciicircum`, 123
`\textasciitilde`, 123
`\textasteriskcentered`, 123, 124
`\textbackslash`, 123
`\textbaht`, 125
`\textbar`, 123
`\textbardbl`, 124
`\textbf`, 30, 36
`\textbigcircle`, 124
`\textblank`, 124

`\textbraceleft`, 123
`\textbraceright`, 123
`\textbrokenbar`, 124
`\textbullet`, 123, 124
`\textcent`, 125
`\textcentoldstyle`, 125
`\textcircled`, 126
`\textcircledP`, 125
`\textcolonmonetary`, 125
`\textcopyleft`, 125
`\textcopyright`, 125
`\textcurrency`, 125
`\textdagger`, 123, 124
`\textdaggerdbl`, 123, 124
`\textdblhyphen`, 124
`\textdblhyphenchar`, 124
`\textdegree`, 190
`\textdiscount`, 124
`\textdiv`, 190
`\textdollar`, 123, 125
`\textdollaroldstyle`, 125
`\textdong`, 125
`\textdownarrow`, 124
`\textellipsis`, 123
`\textemdash`, 123
`\textendash`, 123
`\textepsilon`, 243
`\textesh`, 243
`\textestimated`, 124
`\texteuro`, 125
`\textexclamdown`, 123
`\textflorin`, 125
`\textfractionsolidus`, 190
`\textgreater`, 123
`\textguarani`, 125
`\texthight`, 81
`\textinterrobang`, 124
`\textinterrobangdown`, 124
`\textipa`, 243
`\textit`, 36
`\textLCD`, 211
`\textleftarrow`, 124
`\textless`, 123

- `\textlira`, 125
- `\textlnot`, 190
- `\textmd`, 36
- `\textminus`, 190
- `\textmusicalnote`, 124
- `\textnaira`, 125
- `\textnumero`, 124
- `\textonehalf`, 190
- `\textonequarter`, 190
- `\textonesuperior`, 190
- `\textopenbullet`, 124
- `\textordfeminine`, 123, 124
- `\textordmasculine`, 123, 124
- `\textparagraph`, 123, 124
- `\textperiodcentered`, 123, 124
- `\textpertenthousand`, 124
- `\textperthousand`, 124
- `\textpeso`, 125
- `\textpilcrow`, 124
- `\textpm`, 190
- `\textprimstress`, 243
- `\textquestiondown`, 123
- `\textquotedblleft`, 123
- `\textquotedblright`, 123
- `\textquoteleft`, 123
- `\textquoteright`, 123
- `\textquotesingle`, 124
- `\textquotestraightbase`, 124
- `\textquotestraightdblbase`, 124
- `\textrecipe`, 124
- `\textreferencemark`, 124
- `\textregistered`, 123, 125
- `\textrightarrow`, 124
- `\textsc`, 36
- `\textschwa`, 243
- `\textsci`, 243
- `\textsection`, 123, 124
- `\textservicemark`, 125
- `\textsf`, 36
- `\textsl`, 36
- `\textsterling`, 123, 125
- `\textstyle`, 47, 196
- `\textsurd`, 190
- `\textthreequarters`, 190
- `\textthrequartersemdash`, 124
- `\textthreesuperior`, 190
- `\texttildelow`, 124
- `\texttimes`, 190
- `\texttrademark`, 123, 125
- `\texttt`, 36
- `\texttwelveudash`, 124
- `\texttwosuperior`, 190
- `\textunderscore`, 123
- `\textuparrow`, 124
- `\textvisiblespace`, 123, 208
- `\textwidth`, 60, 69, 81, 166
- `\textwon`, 125
- `\textyen`, 125
- `\tg`, 54
- `\th`, 54
- `\thanks`, 41
- `\the`, 70, 72, 73
- `thebibliography`, 168–170, 274, 275
- `\theendnotes`, 140
- `\theequation`, 205
- `theorem`, 107, 115, 116, 137, 255
- `\therefore`, 192
- `\Theta`, 50, 187
- `\theta`, 50, 187
- `\thickapprox`, 192
- `\thicksim`, 192
- `\thickspace`, 199
- `\thispagestyle`, 86
- `\tilde`, 51, 188
- `\times`, 51, 190
- `\tiny`, 37
- `\tipaencoding`, 243
- `\title`, 41
- `titlepage`, 40, 41, 136
- `\to`, 186
- `\today`, 262
- `\top`, 186
- `\topcaption`, 154, 263
- `\topmargin`, 81
- `\totalheight`, 129
- `\triangle`, 51, 186

$\backslash$ triangledown, 187
 $\backslash$ triangleleft, 190
 $\backslash$ trianglelefteq, 193
 $\backslash$ triangleq, 193
 $\backslash$ triangleright, 190
 $\backslash$ trianglerighteq, 193
 $\backslash$ twocolumn, 92
 $\backslash$ twoheadleftarrow, 194
 $\backslash$ twoheadrightarrow, 194

U

$\backslash$ ul, 127
 $\backslash$ ulcorner, 194
 $\backslash$ uncover, 115, 116
 $\backslash$ underarc, 245, 260
 $\backslash$ underbrace, 189
 $\backslash$ underleftarrow, 194
 $\backslash$ underleftrightharpoon, 194
 $\backslash$ underline, 189
 $\backslash$ underrightharpoon, 194
 $\backslash$ unitlength, 131
 $\backslash$ unlhd, 190, 191
 $\backslash$ unrhd, 190, 191
 $\backslash$ UNTILL, 214
 $\backslash$ upalpha, 188
 $\backslash$ UParrow, 124
 $\backslash$ Uparrow, 193, 194
 $\backslash$ uparrow, 193, 194
 $\backslash$ upbeta, 188
 $\backslash$ upchi, 188
 $\backslash$ Updelta, 188
 $\backslash$ updelta, 188
 $\backslash$ Updownarrow, 193, 194
 $\backslash$ updownarrow, 193, 194
 $\backslash$ upepsilon, 188
 $\backslash$ upeta, 188
 $\backslash$ Ugamma, 188
 $\backslash$ upgamma, 188
 $\backslash$ upharpoonleft, 194
 $\backslash$ upharpoonright, 194
 $\backslash$ upiota, 188
 $\backslash$ upkappa, 188
 $\backslash$ Uplambda, 188

$\backslash$ uplambda, 188
 $\backslash$ uplus, 190
 $\backslash$ upmu, 188
 $\backslash$ upnu, 188
 $\backslash$ Upomega, 188
 $\backslash$ upomega, 188
 $\backslash$ Upphi, 188
 $\backslash$ upphi, 188
 $\backslash$ Uppi, 188
 $\backslash$ uppi, 188
 $\backslash$ Upsi, 188
 $\backslash$ uppsi, 188
 $\backslash$ uprho, 188
 $\backslash$ Upsigma, 188
 $\backslash$ upsigma, 188
 $\backslash$ Upsilon, 50, 187
 $\backslash$ upsilon, 50, 187
 $\backslash$ uptau, 188
 $\backslash$ Uptheta, 188
 $\backslash$ uptheta, 188
 $\backslash$ upuparrows, 194
 $\backslash$ Upupsilon, 188
 $\backslash$ upupsilon, 188
 $\backslash$ upvarepsilon, 188
 $\backslash$ upvarphi, 188
 $\backslash$ upvarpi, 188
 $\backslash$ upvarrho, 188
 $\backslash$ upvarsigma, 188
 $\backslash$ upvartheta, 188
 $\backslash$ Upxi, 188
 $\backslash$ upxi, 188
 $\backslash$ upzeta, 188
 $\backslash$ urcorner, 194
 $\backslash$ url, 141–143
 $\backslash$ usecolortheme, 118
 $\backslash$ useFN, 139
 $\backslash$ usefonttheme, 118
 $\backslash$ useinnertheme, 118
 $\backslash$ useoutertheme, 118
 $\backslash$ usepackage, 26, 32, 34
 $\backslash$ usetheme, 118

V

`\value`, 76
`\varangle`, 187
`\varepsilon`, 50, 187
`\varhexstar`, 124
`\varkappa`, 50, 187
`\varnothing`, 51, 187
`\varphi`, 50, 187
`\varpi`, 187
`\varpropto`, 192
`\varrho`, 187
`\varsigma`, 187
`\varsubsetneq`, 192
`\varsubsetneqq`, 192
`\varsupsetneq`, 192
`\varsupsetneqq`, 192
`\vartheta`, 187
`\vartriangle`, 187
`\vartriangleleft`, 193
`\vartriangleright`, 193
`\Vdash`, 192
`\vDash`, 192
`\vdash`, 191
`\vdots`, 189
`\vec`, 51, 188
`\vee`, 190
`\veebar`, 191
`\verb`, 142, 208, 210
`\verb*`, 208
`verbatim`, 208
`verbatim*`, 208
`\verbatiminput`, 208
`\verbatiminput*`, 208
`verse`, 235, 236
`\versewidth`, 236
`versionhistory`, 217
`\vfil`, 91
`\vfill`, 91
`\vphantom`, 200
`\vhCurrentDate`, 218
`\vhCurrentVersion`, 218
`\vhEntry`, 217
`\vhListAllAuthorsLong`, 218

`\vin`, 236
`\visible`, 116
`Vmatrix`, 197
`vmatrix`, 197
`\voffset`, 81
`\vphantom`, 200
`\vspace`, 91
`\vspace*`, 91
`\Vdash`, 192

W

`\wasylounge`, 124
`\wasyppto`, 193
`\wasythefore`, 187
`\wedge`, 190
`\WHILE`, 214
`\whiledo`, 77, 277
`\widehat`, 188
`\widetilde`, 188
`\width`, 128, 129
`\widthof`, 75
`\wordbox`, 213
`\wordgroup`, 213
`\wp`, 186
`\wr`, 190
`wrapfigure`, 65, 66

X

`\XBox`, 124
`\xcancel`, 189, 255
`\Xi`, 50, 187
`\xi`, 50, 187
`\xleftarrow`, 194
`\xrightarrow`, 194
`\xspace`, 73
`xtab`, 160
`xtabular`, 159

Z

`\zeta`, 50, 186, 187

Список пакетов

A

a0poster, 104, 271

aaai, 272

aaai-named, 272

aastex, 272

abc, 260

abstbook, 272

abstract, 251

accents, 255

acm, 272

acmconf, 272

acronym, 277

active-conf, 272

adami, 248

adrconv, 270

adrlist, 270

advdate, 262

afterpackage, 278

afterpage, 63, 267

aguplus, 272

aiaa-tc, 272

akletter, 101, 270

algorithm2e, 215, 257

algorithmic, 214, 257

algorithmicx, 215, 257

algorithms, 214, 215, 257

alltt, 250

alnumsec, 250

altn, 208

amsart, 100, 273

amsbook, 100, 273

amsbsy, 254

amscd, 198, 199, 255

amsmath, 45, 47, 48, 137, 185, 188, 189, 195, 198–200, 202–204, 254–256

amsmath, 45, 47, 48, 137, 185, 188, 189, 195, 198–200, 202–204, 254–256

amsopn, 254

amsproc, 100, 273

amsrefs, 274

amssymb, 50, 51, 125, 185–187, 190–194, 255

amstext, 48, 254

amsthm, 255

amsxtra, 188

answers, 259

anttor, 126, 248

apa, 273

apacite, 273

apalike, 273

appendix, 250

arabicfront, 255

arabtex, 248

arcs, 245, 260

armtex, 248

arosgn, 248

array, 153, 159, 198, 266

arrayjob, 276

article, 33, 41, 42, 81, 83, 98, 141, 269

artikel1, 100, 269

artikel2, 100, 269

artikel3, 100, 269

arydshln, 155, 266

asaetr, 104, 273

ascelike, 273

ascii, 257

askinclud, 146, 253

assignment, 106, 259

astro, 257

attachfile2, 277

aurora, 264

authblk, 251

authorindex, 274

B

babel, 32, 33, 39, 40, 48, 54, 121–123, 132, 248, 260, 274

babelbib, 274

backgammon, 261

backref, 274

balance, 92, 268

balanced, 93

balancedquotes, 250

bangtex, 248

bardiag, 264

basix, 257

beamer, 104, 111–116, 118, 119, 205, 271

beletter, 270

bez123, 264

bezos, 255

bibcheck, 274

bibtopic, 274

bibunits, 274

bigdelim, 266

bigstrut, 266

binhex, 276

biocon, 259

bizcard, 262

blindtext, 278

blkarray, 266

bmpsized, 264

boek, 100, 269

boek3, 100, 269

boites, 252

book, 33, 41, 42, 81, 83, 98, 102, 135, 269, 270

booklet, 87, 268

booktabs, 153, 266

bophook, 145, 264

boxedminipage, 252

boxhandler, 252, 262

bpchem, 258

braille, 248

braket, 221, 257

breakcites, 169, 274

breakurl, 142, 252

breqn, 255

brief, 100, 101, 269

burmese, 248

bytefield, 212, 257

C

calc, 75–77, 83, 276

calendar, 262

calxxxx, 262

camel, 274

cancel, 189, 255

capt-of, 263

captcont, 263

caption, 66, 263

carlisle, 252, 254, 258, 266, 267

cases, 203, 255

cbcoptic, 260

ccaption, 66, 263

cchess, 261

cd, 106, 261

cd-cover, 261

cdpbundl, 101, 270

cellspace, 155, 266

changebar, 141, 252

chappg, 250

chapterbib, 275

chapterfolder, 254

checkend, 255

chemarr, 258

chemarrow, 258

chemcompounds, 258

chemcono, 258

chemtex, 231, 258

chess, 261

china2e, 262

chngcntr, 72, 276

chngpage, 268

circ, 226, 228, 257

cite, 275

citehack, 169, 248, 275
 citeref, 275
 cjk, 248
 clefval, 276
 clock, 262
 clrcode, 215, 257
 cm-super, 110, 126, 248
 cmap, 252
 cmastro, 257
 codep, 257
 color, 211, 264
 colorinfo, 264
 colortbl, 156, 266
 combine, 104, 253, 273
 commath, 195, 255
 comment, 249
 compactbib, 169, 275
 complexity, 255
 computational-complexity, 273
 ConcProg, 106, 261
 concrete, 126, 196, 248
 contour, 249
 cooking, 261
 coolstr, 277
 cooltooltips, 252
 coordsys, 264
 courseoutline, 106, 259
 coursepaper, 106, 259
 CoverPage, 251
 covington, 260
 crop, 254
 crossreference, 251
 crosswrd, 261
 cryst, 258
 csquotes, 250
 csvtools, 158, 267
 ctable, 159, 267
 ctib4tex, 248
 cuisine, 261
 currvita, 105, 272
 cursor, 257
 curve, 105, 272
 curve2e, 264

curves, 264
 custom-bib, 275
 cuted, 92, 93, 269
 cv, 272
 cwpuzzle, 261
 cyrillic, 248

D

dashbox, 252
 dashrule, 128, 249
 datetime, 262
 dblfloatfix, 269
 dblfnote, 251
 dcolumn, 157, 158, 266
 dcounter, 276
 dcpic, 255
 delarray, 266
 deleg, 204, 255
 desclist, 132, 249
 devanagari, 248
 diagnose, 278
 dialogl, 278
 dice, 261
 dichokey, 259
 dinbrief, 270
 directory, 270
 disser, 105, 271
 dnaseq, 259
 doc, 278
 docstrip, 278
 doipubmed, 275
 dotlessi, 255
 dpfloat, 65, 262
 draftcopy, 144, 253
 draftwatermark, 144, 145, 253
 dramatist, 260
 drftcite, 275

E

easy, 255
 easybib, 255
 easybmat, 255
 easyeqn, 255

easymat, 255
 easytabl, 255
 easyvector, 255
 ebezier, 265
 ebsthesis, 272
 eCards, 259
 ecltree, 265
 edmac, 260
 ednotes, 260
 eepic, 265
 egameps, 255
 elsart, 104, 273
 embedfile, 277
 empheq, 203, 255
 endfloat, 64, 263
 endheads, 252
 endnotes, 140, 252
 engord, 276
 engpron, 260
 engrec, 249
 enthiop, 248
 enumerate, 249
 enumitem, 249
 envlab, 270
 epic, 265
 epigraph, 130, 250
 epsdice, 261
 epstopdf, 264
 eqlist, 132, 250
 eqname, 255
 eqparbox, 252
 esdiff, 196, 255
 esindex, 255
 eskd, 101
 eskdbtab, 101, 270
 eskdgraph, 101, 270
 eskdtext, 101, 270
 eskdx, 33, 101, 248, 270
 eso-pic, 145, 264
 esvect, 255
 etaremun, 250
 euler, 126, 196, 248
 eulervm, 256

europecv, 105, 272
 eurosym, 125
 everypage, 267
 exam, 259
 examdesign, 259
 examplep, 257
 excludeonly, 146, 254
 exercise, 259
 expdlist, 250
 export, 278
 extarrows, 256
 extarticle, 99, 270
 extbook, 99, 270
 extdash, 248
 extletter, 99, 270
 extpfeil, 256
 extproc, 99, 270
 extract, 278
 extreport, 99, 270
 extsizes, 99, 270

F

facsimile, 270
 faktor, 256
 fancybox, 253
 fancyhdr, 86, 166, 167, 270, 274
 fancyvrb, 208, 250
 fax, 270
 feyn, 224, 257
 feynmf, 224, 226, 258
 feynmp, 224, 225
 figcaps, 263
 figsize, 65, 264
 filecontents, 278
 fink, 277
 fixfoot, 251
 fixltx2e, 92, 254, 269
 fixmath, 187, 257
 fixme, 143, 253
 flagderiv, 256
 flashcards, 106, 259
 float, 62, 262
 floatflt, 65, 262

floatpag, 269
 floatraw, 62, 262
 fltpoint, 276
 flushend, 92, 93, 269
 fmtcount, 276
 fncychap, 250
 fnpos, 251
 fontenc, 48, 247
 footbib, 275
 footmisc, 139, 251
 footnpag, 251
 formlett, 270
 fp, 276
 ftnright, 92, 93, 251
 fullpage, 268

G

g-brief, 270
 gatech-thesis, 272
 gauss, 256
 gensymb, 257
 geometry, 83, 85, 268, 270
 gloss, 181, 275
 gost, 172, 248
 gost2003, 174
 gost2003s, 174
 gost2008, 171, 174
 gost2008s, 174
 gost71u, 248
 gost780u, 248
 graphics, 60
 graphicx, 60, 61, 264
 guitar, 238, 239, 261
 gurmukhi, 248

H

hanging, 249
 harvard, 275
 hc, 269
 hcart, 269
 hcletter, 269
 hcreport, 269
 hcslices, 269

hepparticles, 224, 258
 hepthesis, 272
 hhline, 155, 266
 hhline.pdf, 155
 hhtensor, 256
 hierotex, 243
 hilowres, 264
 histogr, 265
 hitec, 103, 271
 hvfloat, 263
 hyperref, 110, 111, 142, 144, 181, 252, 274

I

iagproc, 273
 icomma, 249, 257
 IEEEtran, 273
 ifdraft, 276
 ifmslide, 271
 ifmtarg, 276
 ifpdf, 277
 ifthen, 76, 77, 277
 import, 254
 indentfirst, 32, 123, 248
 index, 275
 inputenc, 32, 34, 167, 247, 248
 interactiveworkbook, 259
 iso, 102, 270
 iso10303, 102, 270
 isodate, 262
 isostds, 270
 isotope, 226, 258
 itrans, 248

J

jpsj2, 104, 273

K

karnaugh, 256
 keystroke, 257
 keyval, 60, 83, 277
 KOMA-Script, 33, 83–85, 100–102, 165, 268–271

L

labbook, 106, 259
 labelcas, 277
 labels, 262
 lastpage, 251
 layout, 80, 268
 lcd, 211, 257
 leaflet, 273
 ledmac, 260
 ledpar, 260
 letter, 81, 83, 98, 100, 269, 270
 lettre, 270
 lexikon, 260
 lh, 34, 126, 157, 248, 281
 lineno, 143, 249
 linguex, 260
 lipsum, 278
 listings, 208–210, 257
 listliketab, 161, 267
 longtable, 159, 267
 lscape, 82, 267, 268
 lstlistings, 116
 ltxmisc, 72, 140, 146, 169, 203, 221,
 262, 274, 276
 ltxtable, 267

M

mailing, 270
 makecell, 157, 266
 makecirc, 228
 makecmds, 277
 makedtx, 278
 makeglos, 180, 181, 275
 makeidx, 174, 176, 275
 malayalam, 248
 manjutex, 248
 manual, 103
 manyfoot, 251
 marginal, 269
 marginnote, 141, 252
 mathtext, 48, 248
 mathtools, 255
 maybemath, 256

mboxfill, 128, 249
 mcaption, 67, 263
 memoir, 102, 270
 method, 257
 mh, 203, 255
 mhchem, 230, 258
 mhequ, 255
 microtype, 89, 267
 midfloat, 269
 minimal, 98, 270
 miniplot, 65, 263
 minitoc, 165, 274
 miscorr, 32, 123, 248
 moderncv, 105, 272
 montex, 248
 morderncv, 105
 moreverb, 250
 morse, 254
 movie15, 110
 mparhack, 141, 252
 msg, 278
 mtn, 104, 273
 multenum, 250
 multibbl, 275
 multibib, 275
 multicol, 92–94, 269
 multido, 277
 multimedia, 119
 multirow, 155, 156, 266
 multitoc, 274
 musictex, 239, 261
 musixlyr, 241, 261
 musixtex, 239–241, 261
 muthesis, 272
 mylatex, 254
 mysixlyr, 242
 mysixtex, 242

N

nassflow, 103
 natbib, 275
 nath, 256
 nature, 104, 273

ncc, [34](#), [83](#), [100](#), [269](#)
 nccboxes, [253](#)
 ncccomma, [49](#), [249](#)
 ncccropmark, [254](#)
 nccfancyhdr, [167](#), [274](#)
 nccfloats, [66](#), [262](#)
 nccfoots, [139](#), [251](#)
 ncclatex, [83](#), [269](#)
 nccmath, [47](#), [203](#)
 nccparskip, [268](#)
 nccpic, [264](#)
 nccproc, [273](#)
 nccrules, [128](#), [249](#)
 nccsect, [250](#)
 nccstretch, [128](#), [249](#)
 nccthm, [206](#), [256](#)
 ncctools, [66](#), [128](#), [132](#), [139](#), [145](#), [167](#),
 [206](#), [248–251](#), [253](#), [254](#), [256](#),
 [262](#), [264](#), [268](#), [274](#), [276](#), [278](#)
 nctools, [49](#), [249](#)
 newfile, [278](#)
 newlfn, [101](#), [270](#)
 nomencl, [182](#), [275](#)
 notoccite, [275](#)
 nrc, [273](#)
 nrc1, [273](#)
 nrc2, [273](#)
 ntgclass, [100](#), [101](#), [269](#)
 ntheorem, [255](#)
 numprint, [157](#), [158](#), [266](#)

O

oberdiek, [252](#), [254](#), [258](#), [264](#), [266](#), [268](#),
 [276](#), [277](#)
 ocherokee, [248](#)
 octavo, [102](#), [270](#)
 oinuit, [248](#)
 onlyamsmath, [256](#)
 optional, [146](#), [147](#), [278](#)
 optparams, [277](#)
 osmanian, [248](#)
 othello, [262](#)
 overcite, [275](#)

P

pagenote, [140](#), [252](#)
 pagesel, [254](#)
 papertex, [98](#), [273](#)
 paralist, [37](#), [133](#), [250](#)
 parallel, [94](#), [269](#)
 parcolumns, [269](#)
 parrun, [260](#)
 pb-diagram, [162](#), [256](#)
 pdfcrypt, [252](#)
 pdflatex, [264](#)
 pdfscape, [82](#), [268](#)
 pdfpages, [86](#), [253](#), [268](#)
 pdfscreen, [268](#)
 pdfslide, [271](#)
 pdfsync, [253](#)
 pdftricks, [265](#)
 pecha, [106](#), [248](#)
 perltex, [276](#)
 pfnote, [251](#)
 pgf, [87](#), [118](#), [265](#), [268](#)
 pgfpages, [87](#), [118](#), [268](#)
 phonetic, [260](#)
 photo, [263](#)
 picinpar, [263](#)
 pict2e, [264](#), [265](#)
 pittetd, [272](#)
 placeins, [64](#), [263](#)
 plain, [174](#)
 plari, [260](#)
 plates, [263](#)
 play, [260](#)
 poemscol, [260](#)
 portland, [82](#), [267](#)
 powerdot, [104](#), [271](#)
 powersem, [103](#), [271](#)
 ppower4, [271](#)
 ppr-prv, [103](#), [271](#)
 prelim2e, [253](#)
 preprint, [92](#), [251](#), [263](#), [268](#), [276](#)
 prettyref, [138](#), [251](#)
 probsoln, [259](#)
 proc, [98](#), [269](#)

processkv, 277
 progress, 253
 prosper, 103, 271
 pscyr, 102
 pseudocode, 216, 257
 psgo, 262
 pst-3d, 265
 pst-3dplot, 256
 pst-bar, 256
 pst-barcode, 265
 pst-blur, 265
 pst-circ, 228, 258
 pst-coil, 265
 pst-dbicons, 257
 pst-eps, 265
 pst-eucl, 256
 pst-fill, 265
 pst-fr3d, 253
 pst-func, 256
 pst-geo, 265
 pst-gr3d, 265
 pst-grad, 265
 pst-infixplot, 265
 pst-labo, 229, 258
 pst-lens, 265
 pst-light3d, 265
 pst-math, 265
 pst-node, 265
 pst-optic, 228, 258
 pst-osci, 228, 258
 pst-pdf, 265
 pst-pdgr, 259
 pst-plot, 265
 pst-poly, 265
 pst-slpe, 265
 pst-text, 265
 pst-tree, 260, 265
 pst-uml, 257
 pst-vue3d, 265
 pst-xkey, 265
 PSTricks, 228, 229
 pstricks, 255, 258, 260, 265
 ptptex, 273

Q

qcm, 106, 259
 qobitree, 256
 quotchap, 250

R

r-und-s, 258
 ragged2e, 90, 249
 rapport1, 100, 269
 rapport3, 100, 269
 rccol, 158, 266
 rcs, 218, 253
 rcsinfo, 218, 253
 refart, 271
 refcheck, 251
 refcount, 276
 refman, 271
 refrep, 271
 report, 33, 42, 81, 83, 98, 102, 269, 271
 revtex4, 33, 104, 273
 rmpage, 268
 robustcommand, 277
 robustindex, 275
 rotating, 264
 rotfloat, 262
 rotpages, 82, 268
 rsphrase, 231
 rst, 260

S

sanskrit, 248
 sauerj, 269, 276, 277
 savefnmark, 139, 251
 savetrees, 268
 scalebar, 265
 sciposter, 104, 271
 scrartcl, 33, 42, 83, 84, 100, 269
 scrbook, 33, 42, 83, 84, 100, 102, 135, 269, 270
 scrlettr, 83
 scrlettr2, 100, 101, 269, 270
 scrreprt, 33, 83, 100, 102, 269, 271
 sectionbox, 250

semantic, 257
 seminar, 103, 108, 271
 setspace, 91, 268
 sffms, 102, 103, 270
 shadow, 129, 253
 shapepar, 249
 shorttoc, 164, 274
 showkeys, 138, 251
 sibjnm, 104, 273
 sidecap, 263
 sides, 103, 260
 siggraph, 273
 sinhala, 248
 SIstyle, 223, 258
 SIunits, 223, 258
 skak, 261
 skaknew, 261
 slashbox, 156, 266
 slashed, 258
 slides, 82, 98, 103, 108, 111, 269, 271
 smfart, 273
 smfbook, 273
 snapshot, 253
 sobolev, 256
 songbook, 261
 soul, 127, 249
 spie, 273
 splitidx, 177
 splitindex, 275
 srcltx, 253
 sseq, 256
 stage, 106, 236, 237, 260
 stdclsdv, 277
 stdpage, 268
 stfloats, 269
 sttools, 92, 269
 subeqn, 256
 subfig, 65, 263
 subfloat, 263
 sublabel, 276
 sudoku, 262
 supertabular, 159, 267
 svn, 218, 253

svn-multi, 253
 svninfo, 219, 253
 svnkw, 253
 synttree, 260

T

t-angles, 256
 t2, 48, 123, 248, 275
 table, 159
 tableaux, 162, 256
 tabular, 153, 159, 266
 tabularht, 266
 tabularx, 158, 267
 tabulary, 159, 267
 tabvar, 162, 256
 talk, 104, 271
 technics, 167, 168, 274
 telugu, 248
 tensind, 255
 tensor, 198, 256
 teubner, 260
 texmate, 261
 texpower, 103, 271
 texshade, 233, 259
 textcomp, 124–126, 190
 texttopo, 259
 ticket, 262
 timing, 258
 tipa, 243, 260
 titlesec, 250
 titletoc, 250
 titling, 251
 tocibind, 165, 274
 tocloft, 164, 274
 tocvsec2, 274
 tokenizer, 77, 277
 toolbox, 277
 tools, 75, 92, 153, 155, 157–159, 208,
 251, 266, 267, 269
 topcapt, 154, 263
 totpages, 254
 truncate, 249
 twoopt, 277

twoup, 87, 268
 typearea, 83–85, 268
 typedref, 251
 typehtml, 252
 typogrid, 85, 268

U

uaclases, 272
 ucs, 248
 uthesis, 105, 272
 uiuthesis, 272
 ulem, 249
 umich-thesis, 272
 umoline, 249
 underscore, 249
 units, 223, 258
 unitsdef, 223, 258
 unsrt, 275
 upgreek, 187, 188, 257
 url, 142, 252
 uwthesis, 272

V

variations, 256
 varindex, 275
 varioref, 251
 vector, 256
 venn, 257
 verbatim, 35, 116, 208, 250
 verse, 236, 260
 versions, 278
 vhistory, 217, 218, 253
 vita, 105, 272
 vmargin, 83, 84, 268
 vntex, 248
 volumes, 254
 vpe, 253

W

wallpaper, 145, 264
 warpcol, 158, 266
 was, 187, 249, 257
 wasysym, 53, 124, 125, 187, 191, 193,
 196, 257

watermark, 145, 264
 weekday, 262
 wordlike, 254
 wrapfig, 65, 263

X

xcolor, 264
 xdoc, 278
 xifthen, 76, 77, 277
 xkey, 265
 xkeyval, 277
 xr, 137, 251
 xspace, 73
 xtab, 159, 160, 267
 xyntex, 231, 258
 xypic, 199, 265
 xytree, 260

Y

yafoot, 251
 york-thesis, 272

Z

zahl2string, 276

Литература

- [1] Д. Э. Кнут. Всё про $\text{T}_{\text{E}}\text{X}$. — М. : Вильямс, 2003. — 560 с. — ISBN: 5-8459-0382-3.
- [2] Д. Э. Кнут. Всё про $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{F}_{\text{O}}\text{N}_{\text{T}}$. — М. : Вильямс, 2003. — 384 с. — ISBN: 5-8459-0442-0.
- [3] Д. Э. Кнут. Компьютерная типография. — М. : Мир, 2003. — 686 с. — ISBN: 5-03-3361-0.
- [4] Г. Грэтцер. Первые шаги в $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. — М. : Мир, 2000. — 172 с. — ISBN: 5-03-003361-0.
- [5] М. Гуссенс, Ф. Миттельбах, А. Самарин. Путеводитель по пакету $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ и его расширению $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$. — М. : Мир, 1999. — 606 с. — ISBN: 5-03-003325-4.
- [6] М. Гуссенс, С. Ратц, Ф. Миттельбах. Путеводитель по пакету $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ и его графическим расширениям. — М. : Мир, 2002. — 621 с. — ISBN: 5-03-003388-2.
- [7] М. Гуссенс, С. Ратц. Путеводитель по пакету $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ и его Web-приложениям. — М. : Мир, 2001. — 604 с. — ISBN: 5-03-003387-4.
- [8] С. М. Львовский. Набор и вёрстка в системе $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. — М. : МЦНМО, 2003. — 448 с. — ISBN: 5-94057-091-7.
- [9] И. А. Котельников, П. З. Чеботаев. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ по-русски. — Новосибирск : Сибирский Хронограф, 2004. — 496 с. — ISBN: 5-87550-195-2.
- [10] А. И. Роженко. Искусство верстки в $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 'е / Под ред. А. С. Алексеева. — Новосибирск : Изд. ИВМиМГ СО РАН, 2005. — 398 с. — ISBN: 5-901548-25-6.
- [11] А. Э. Мильчин. Издательский словарь-справочник. — М. : ОЛМА-Пресс, 2003. — 500 с. — ISBN: 5-224-04560-6.