Проект документации FreeBSD

введение для новых участников

Аннотация

Спасибо за то, что стали частью проекта FreeBSD Documentation Project. Ваш вклад чрезвычайно важен, и мы это ценим.

Это руководство содержит информацию, необходимую для начала работы с Проектом документации FreeBSD (FDP), включая инструменты, программное обеспечение и философию, лежащую в основе Проекта документации.

Это работа в процессе. Исправления и дополнения всегда приветствуются.

Содержание

Предисловие . . . 5
Приглашения командной оболочки . . . 5
Типографические соглашения . . . 5
Примечания, советы, важная информация, предупреждения и примеры . . . 5
Благодарности . . . 6
1. Обзор . . . 7
1.1. Документация в экосистеме FreeBSD . . . 7
1.2. Быстрый старт . . . 8
1.3. Набор документации FreeBSD . . . 12
2. Инструменты . . . 14
2.1. Необходимые инструменты . . . 14
2.2. Необязательные инструменты . . . 14
3. Рабочая копия . . . 15
3.1. Документация и Справочник . . . 15
3.2. Выбор каталога . . . 15
3.3. Извлечение копии . . . 15
3.4. Обновление рабочей копии . . . 16
3.5. Отмена изменений . . . 16
3.6. Создание Diff . . . 16
3.7. Ссылки Git . . . 17
4. Структура каталогов документации . . . 18
4.1. Верхний уровень, doc/ . . . 18
4.2. Каталоги . . . 18
4.3. Информация, относящаяся к документу . . . 19
4.4. Книги: books/ . . . 19
4.5. Статьи: articles/ . . . 21
4.6. Управление списками участников . . . 21
5. Процесс сборки документации FreeBSD . . . 26
5.1. Преобразование AsciiDoc в выходные форматы . . . 26
5.2. Набор инструментов для сборки документации FreeBSD . . . 28
5.3. Информация о Makefile в дереве документации . . . 28
6. Основы Asciidoctor . . . 38
6.1. Обзор . . . 38
6.2. Заголовки . . . 39
6.3. Абзацы . . . 39
6.4. Списки . . . 40
6.5. Ссылки . . . 41
6.6. Изображения и иконки . . . 42

6.7. Заключение . . . 44
7. Розеттский камень . . . 45
7.1. Сравнение Docbook и AsciiDoc . . . 45
8. Переводы . . . 49
8.1. Что означают i18n и l10n? . . . 49
8.2. Существует ли рассылка для переводчиков? . . . 49
8.3. Нужны дополнительные переводчики? . . . 49
8.4. Какие языки мне нужно знать? . . . 49
8.5. Какой софт нужно знать? . . . 49
8.6. Как узнать, кто ещё, может быть, переводит на тот же язык? . . . 50
8.7. Никто не переводит на мой язык. Что мне делать? . . . 50
8.8. Я перевел часть документации, куда мне её отправить? . . . 51
8.9. Я единственный, кто работает над переводом на этот язык, как мне отправить свой перевод? . . . 51
8.10. Можно ли включать в перевод текст, специфичный для языка или страны? . . . 52
9. Переводы PO . . . 54
9.1. Введение . . . 54
9.2. Быстрый старт . . . 54
9.3. Создание новых переводов . . . 55
9.4. Перевод . . . 57
9.5. Советы переводчикам . . . 57
9.6. Сборка переведенного документа . . . 58
9.7. Отправка новой переведённой версии . . . 58
10. Переводы в Weblate . . . 60
10.1. Введение . . . 60
10.2. Как стать переводчиком FreeBSD . . . 60
10.3. Представьтесь . . . 60
10.4. Вход в Weblate . . . 61
10.5. Найти команду локализации для участия . . . 62
10.6. Перевод онлайн на Weblate . . . 63
10.7. Перевод в автономном режиме . . . 64
10.8. Перевод на основе автоматических предложений . . . 65
10.9. Вычитка и проверка качества в Weblate . . . 68
10.10. Сборка переведенного документа . . . 69
10.11. Отправка переводов . . . 72
10.12. FAQ (Часто задаваемые Вопросы) . . . 74
11. Справочник . . . 75
11.1. Введение . . . 75
11.2. Разделы . . . 75
11.3. Язык разметки . . . 75
11.4. Пример структуры страницы Справочника . . . 80

11.5. Тестирование . . . 82
11.6. Примеры страниц Справочника для использования в качестве шаблонов . . . 83
11.7. Ресурсы . . . 83
12. Стиль написания . . . 84
12.1. Советы . . . 84
12.2. Рекомендации . . . 85
12.3. Руководство по стилю . . . 86
12.4. Список специальных символов . . . 87
12.5. Проверка стиля с помощью Vale . . . 87
13. Настройка редактора . . . 90
13.1. Vim . . . 90
13.2. Emacs . . . 92
13.3. nano . . . 93
14. Товарные знаки . . . 96
14.1. Символы товарных знаков . . . 96
14.2. Цитирование товарных знаков . . . 96
15. Смотрите также . . . 98
15.1. Проект документации FreeBSD . . . 98
15.2. Hugo . . . 98
15.3. AsciiDoctor . . . 98
15.4. HTML . . . 98
Приложение A: Примеры . . . 99
A.1. **Книга** AsciiDoctor . . . 99
A.2. **Статья** AsciiDoctor . . . 100

Предисловие

Приглашения командной оболочки

В этой таблице показаны приглашения системы по умолчанию и приглашение суперпользователя. В примерах используются эти приглашения, чтобы указать, от какого типа пользователя выполняется пример.

Пользователь	Приглашение
Обычный пользователь	%
`root`	#

Типографические соглашения

Эта таблица описывает типографские соглашения, используемые в этой книге.

Значение	Примеры
Имена команд.	Используйте `ls -l` для вывода списка всех файлов.
Имена файлов.	Измените файл .login.
Вывод компьютера на экран.	`У вас есть почта.`
Что вводит пользователь, в отличие от выводимого компьютером на экран.	`% date +"Время: %H:%M"` `Время: 09:18`
Ссылки на руководства.	Используйте su(1) для смены пользователя.
Имена пользователей и групп.	Только `root` может это сделать.
Выделение текста.	Пользователь *обязан* сделать это.
Текст, который пользователь должен заменить на фактический текст.	Для поиска ключевого слова в руководствах введите `man -k ключевое_слово`
Переменные окружения.	`$HOME` устанавливается в домашний каталог пользователя.

Примечания, советы, важная информация, предупреждения и примеры

Примечания, предупреждения и примеры выделены в тексте.



Примечания представлены следующим образом и содержат информацию, на которую следует обратить внимание, так как она может повлиять на

действия пользователя.



Советы представлены таким образом и содержат полезную для пользователя информацию, например, показывая более простой способ выполнения чего-либо.



Важная информация представлена таким образом. Обычно это дополнительные шаги, которые может потребоваться выполнить пользователю.



Предупреждения выглядят так и содержат информацию о возможном ущербе, если инструкции не будут соблюдены. Этот ущерб может быть физическим — для оборудования или пользователя — или нефизическим, например, случайным удалением важных файлов.

Пример 1. Пример примера

Примеры представлены следующим образом и обычно содержат пошаговые инструкции или результаты выполнения определённых действий.

Благодарности

Моя благодарность Сью Блейк, Патрику Дурусо, Джону Гамильтону, Питеру Флинну и Кристоферу Мэдену, которые нашли время прочитать ранние черновики этого документа и предоставили множество ценных замечаний и критики.

Глава 1. Обзор

Добро пожаловать в Проект документации FreeBSD (FDP). Качественная документация крайне важна для успеха FreeBSD, и мы очень высоко ценим ваш вклад.

Этот документ описывает организацию FDP, как писать и отправлять документацию, а также как эффективно использовать доступные инструменты.

Все желающие могут внести свой вклад в FDP. Готовность помочь — единственное требование для участия.

Это руководство показывает, как:

- Понять роль документации и её место в экосистеме.
- Определите, какие части FreeBSD поддерживаются FDP.
- Установить необходимые инструменты и файлы документации.
- Внести изменения в документацию.
- Представить изменения для проверки и включения в документацию FreeBSD.

1.1. Документация в экосистеме FreeBSD

Все документы создаются для пользы читателей, а не их авторов или сопровождающих. Они должны адаптироваться к читателю, а не ожидать, что читатель адаптируется к ним.

Никогда не вините читателя за:

- невозможность легко или вообще использовать документ
- если документ показался ему непонятным
- непонимание документа или того, как его применить
- если он не нашел явный ответ или шаги, чтобы логически прийти к нему

Вместо этого подтвердите, что документ:

- недоступный
- запутанный
- трудно понимаемый или применимый
- неполный

Затем создайте документ:

- более доступный
- менее запутанный
- более ясный
- более полный

Используйте следующие методы:

- Примените лучшие практики доступности, чтобы исправить выявленную проблему и любые подобные, которые обнаружите
- переработайте или уточните запутанную структуру или язык
- добавьте соответствующие примеры к части, которая трудна для понимания или применения
- заполните пробелы или добавьте недостающие промежуточные этапы

1.2. Быстрый старт

Некоторые подготовительные шаги необходимо выполнить перед редактированием документации FreeBSD. Сначала подпишитесь на рассылку Список рассылки Проекта Документации FreeBSD. Некоторые участники команды также общаются в IRC-канале `#bsddocs` на EFnet. Эти люди могут помочь с вопросами или проблемами, связанными с документацией.

1.2.1. Процесс установки FreeBSD

1. Установите эти пакеты. *Мета-порт* `docproj` устанавливает все приложения, необходимые для работы с документацией FreeBSD.

   ```
   # pkg install docproj
   ```

2. Установите локальную рабочую копию документации из репозитория FreeBSD в ~/doc (см. Рабочая копия).

   ```
   % git clone https://git.FreeBSD.org/doc.git ~/doc
   ```

3. Отредактируйте файлы документации, которые требуют изменений. Если файлу нужны значительные изменения, обратитесь за советом в список рассылки.

   Просмотрите вывод и отредактируйте файл, чтобы исправить указанные проблемы, затем повторно запустите команду для поиска оставшихся проблем. Повторяйте, пока все ошибки не будут устранены.

4. ***Всегда*** собирайте и проверяйте изменения перед отправкой. Запуск команды `make` в подкаталогах `documentation` или `website` сгенерирует документацию в формате HTML.

   ```
   % make
   ```

   Для сокращения времени компиляции может быть скомпилирован только один язык:

```
% make DOC_LANG=en
```

Результаты сборки сохраняются в ~/doc/documentation/public/en/articles/ и ~/doc/documentation/public/en/books/.

5. Просмотрите вывод сборки и убедитесь, что правки не содержат опечаток, проблем с версткой или ошибок. Если в процессе сборки обнаружены ошибки, отредактируйте проблемные файлы, чтобы исправить все возникшие проблемы, затем снова запустите команду сборки, пока все ошибки не будут устранены.

6. Добавьте все файлы с помощью `git add .`, затем просмотрите изменения с помощью `git diff`. Например:

```
% git add .
% git diff --staged
```

Убедитесь, что все необходимые файлы включены, затем зафиксируйте изменение в вашей локальной ветке и создайте патч с помощью `git format-patch`

```
% git commit
% git format-patch -U999999 origin/main
```

Параметр `-U999999` добавляет в файл патча максимальное количество контекста, что делает процесс рецензирования проще.

Файл патча, созданный с помощью `git format-patch` в локальном каталоге, будет содержать идентификатор автора и адреса электронной почты, что упрощает применение разработчиками (с помощью `git am`) и правильное указание авторства.



Чтобы упростить применение патча коммиттерами в их рабочей копии дерева документации, пожалуйста, сгенерируйте файл .diff из корня вашего дерева документации.

В приведённом выше примере были внесены изменения в раздел **bsdinstall** Руководства.

7. Отправьте патч или diff-файл с помощью веб-системы Problem Report. В поле `Component` выберите ту часть документации, к которой относится изменение. В поле `Summary` введите *краткое описание проблемы*. В поле `Description` введите краткое описание изменений и любые важные детали о них. Используйте кнопку **[Add an attachment]**, чтобы прикрепить патч или diff-файл. Наконец, нажмите кнопку **[Submit Bug]**, чтобы отправить ваш diff в систему отчётов об ошибках.

1.2.2. Процесс установки GNU/Linux

1. Установите эти пакеты в системах на основе apt, таких как Debian или Ubuntu. В других дистрибутивах GNU/Linux названия пакетов могут отличаться. В случае сомнений обратитесь к менеджеру пакетов вашего дистрибутива.

```
# apt install asciidoctor hugo ruby-asciidoctor-pdf ruby-rouge git bmake
```

2. Установите локальную рабочую копию документации из репозитория FreeBSD в ~/doc (см. Рабочая копия).

```
% git clone https://git.FreeBSD.org/doc.git ~/doc
```

3. Отредактируйте файлы документации, которые требуют изменений. Если файлу нужны значительные изменения, обратитесь за советом в список рассылки.

   Просмотрите вывод и отредактируйте файлы, чтобы исправить обнаруженные проблемы, затем снова запустите команду, чтобы найти оставшиеся проблемы. Повторяйте, пока все ошибки не будут устранены.

4. Всегда собирайте и тестируйте изменения перед их отправкой. Запуск `bmake` в подкаталогах `documentation` или `website` сгенерирует документацию в формате HTML.

```
% bmake run LOCALBASE=/usr
```

5. Добавьте все файлы с помощью `git add .`, затем просмотрите изменения с помощью `git diff`. Например:

```
% git add .
% git diff --staged
```

   Убедитесь, что все необходимые файлы включены, затем зафиксируйте изменение в вашей локальной ветке и создайте патч с помощью `git format-patch`

```
% git commit
% git format-patch -U999999 origin/main
```

   Параметр `-U999999` добавляет в файл патча максимальное количество контекста, что делает процесс рецензирования проще.

   Файл патча, созданный с помощью `git format-patch` в локальном каталоге, будет содержать идентификатор автора и адреса электронной почты, что упрощает применение разработчиками (с помощью `git am`) и правильное указание авторства.



Чтобы упростить применение патча коммиттерами в их рабочей копии дерева документации, пожалуйста, сгенерируйте файл .diff из корня вашего дерева документации.

6. Отправьте патч или diff-файл с помощью веб-системы Problem Report. При использовании веб-формы укажите в поле *Summary* краткое описание проблемы. Выберите компонент `Documentation`. В поле *Description* введите краткое описание проблемы из поля *Summary* и добавьте *patch* в поле *Keywords*. Используйте кнопку **[Add an attachment]**, чтобы прикрепить патч или diff-файл. Наконец, нажмите кнопку **[Submit Bug]**, чтобы отправить ваш diff в систему отчётов об ошибках.

1.2.3. Процесс установки macOS®

1. Установите эти пакеты с помощью Homebrew и RubyGem.

```
$ brew install hugo ruby git bmake
```

2. Добавьте Ruby в Path.

```
$ echo 'export PATH="$(brew --prefix ruby)/bin:$PATH"' >> ~/.zshrc
$ echo 'export PATH="$(brew --prefix hugo)/bin:$PATH"' >> ~/.zshrc
$ echo 'export GEM_PATH="$(gem environment gemdir)"' >> ~/.zshrc
$ echo 'export PATH="${GEM_PATH}/bin:$PATH"' >> ~/.zshrc
$ source ~/.zshrc
```

3. Установите пакет rouge с помощью RubyGem.

```
$ sudo gem install rouge asciidoctor asciidoctor-pdf asciidoctor-epub3
```

4. Установите локальную рабочую копию документации из репозитория FreeBSD в ~/doc (см. Рабочая копия).

```
$ git clone https://git.FreeBSD.org/doc.git ~/doc
```

5. Отредактируйте файлы документации, которые требуют изменений. Если файлу нужны значительные изменения, обратитесь за советом в список рассылки.

   Просмотрите вывод и отредактируйте файлы, чтобы исправить обнаруженные проблемы, затем снова запустите команду, чтобы найти оставшиеся проблемы. Повторяйте, пока все ошибки не будут устранены.

6. Всегда собирайте и тестируйте изменения перед их отправкой. Запуск `bmake` в подкаталогах `documentation` или `website` сгенерирует документацию в формате HTML.

```
$ bmake run USE_RUBYGEMS=YES RUBY_CMD="$(brew --prefix ruby)/bin/ruby"
HUGO_CMD="$(brew --prefix hugo)/bin/hugo"
```

7. Добавьте все файлы с помощью `git add .`, затем просмотрите изменения с помощью `git diff`. Например:

   ```
   % git add .
   % git diff --staged
   ```

   Убедитесь, что все необходимые файлы включены, затем зафиксируйте изменение в вашей локальной ветке и создайте патч с помощью `git format-patch`

   ```
   % git commit
   % git format-patch -U999999 origin/main
   ```

   Параметр `-U999999` добавляет в файл патча максимальное количество контекста, что делает процесс рецензирования проще.

   Файл патча, созданный с помощью `git format-patch` в локальном каталоге, будет содержать идентификатор автора и адреса электронной почты, что упрощает применение разработчиками (с помощью `git am`) и правильное указание авторства.

   Чтобы упростить применение патча коммиттерами в их рабочей копии дерева документации, пожалуйста, сгенерируйте файл .diff из корня вашего дерева документации.

8. Отправьте патч или diff-файл с помощью веб-системы Problem Report. При использовании веб-формы укажите в поле *Summary* краткое описание проблемы. Выберите компонент `Documentation`. В поле *Description* введите краткое описание проблемы из поля *Summary* и добавьте *patch* в поле *Keywords*. Используйте кнопку **[Add an attachment]**, чтобы прикрепить патч или diff-файл. Наконец, нажмите кнопку **[Submit Bug]**, чтобы отправить ваш diff в систему отчётов об ошибках.

1.3. Набор документации FreeBSD

FDP отвечает за четыре категории документации FreeBSD.

- *Руководство*: Руководство представляет собой всеобъемлющий онлайн-ресурс и справочник для пользователей FreeBSD.
- *FAQ*: В разделе Часто задаваемых вопросов (FAQ) используется формат коротких вопросов и ответов для решения вопросов, часто задаваемых в различных почтовых рассылках и форумах, посвящённых FreeBSD. Такой формат не подразумевает длинных и развёрнутых ответов.
- *Справочник*: Страницы Справочника (man-страницы) системы на английском языке

обычно не создаются FDP, так как они являются частью базовой системы. Однако FDP может перефразировать части существующих руководств, чтобы сделать их понятнее или исправить неточности.

- *Веб-сайт*: Это основное представительство FreeBSD в интернете, доступное по адресу https://www.FreeBSD.org/ и на множестве зеркал по всему миру. Веб-сайт обычно становится первым знакомством нового пользователя с FreeBSD.

Команды переводчиков отвечают за перевод Руководства и веб-сайта на разные языки. На данный момент руководства (man-страницы) не переводятся.

Исходные тексты документации для веб-сайта FreeBSD, Handbook и FAQ доступны в репозитории документации по адресу `https://cgit.freebsd.org/doc/`.

Исходный код страниц справочника доступен в отдельном репозитории, расположенном по адресу `https://cgit.freebsd.org/src/`.

Документация сообщений о фиксациях доступна с помощью `git log`. Сообщения о фиксациях также архивируются по ссылке Коммиты во все ветки репозитория документации.

Веб-интерфейсы для обоих репозиториев доступны по адресам https://cgit.freebsd.org/doc/ и https://cgit.freebsd.org/src/.

Большое количество авторов участвовало в написании руководств или инструкций по FreeBSD. Некоторые из этих документов хранятся в рамках файлов FDP. В других случаях авторы предпочли опубликовать документацию отдельно. FDP стремится предоставить ссылки на как можно большее количество такой внешней документации.

Глава 2. Инструменты

Для управления документацией FreeBSD и её преобразования в различные выходные форматы используется несколько программных инструментов. Некоторые из них обязательны и должны быть установлены перед работой с примерами из следующих глав. Другие являются опциональными, расширяя возможности или упрощая процесс создания документации.

2.1. Необходимые инструменты

Установите `docproj` *мета-порт*, как показано в обзорной главе из Коллекции портов. Эти приложения необходимы для работы с документацией FreeBSD. Далее приведены дополнительные заметки об отдельных компонентах.

2.2. Необязательные инструменты

Эти приложения не обязательны, но могут упростить работу с документацией или расширить её возможности.

2.2.1. Программное обеспечение

Vim (editors/vim) — популярный редактор для работы с Asciidoctor.

Emacs (editors/emacs).

Оба инструмента включают специальный режим, который сокращает количество необходимых нажатий клавиш и помогает снизить количество ошибок.

Глава 3. Рабочая копия

Рабочая копия — это копия дерева документации репозитория FreeBSD, загруженная на локальный компьютер. Изменения вносятся в локальную рабочую копию, тестируются, а затем отправляются в виде патчей для включения в основной репозиторий.

Полная копия дерева документации может занимать 550 мегабайт дискового пространства. Рекомендуется выделить не менее гигабайта, чтобы было место для временных файлов и тестовых версий в различных выходных форматах.

Git используется для управления файлами документации FreeBSD. Он устанавливается через devel/git, который также имеет облегчённую версию под названием git-lite:

```
# pkg install git-lite
```

3.1. Документация и Справочник

Документация FreeBSD — это не только книги и статьи. Справочник (man) для всех команд и конфигурационных файлов также являются частью документации и входят в сферу ответственности FDP. Используются два репозитория: `doc` для книг и статей и `src` для операционной системы и справочника. Для редактирования справочника необходимо отдельно получить репозиторий `src`.

Репозитории могут содержать несколько версий документации и исходного кода. Новые изменения почти всегда вносятся только в последнюю версию, называемую `main`.

3.2. Выбор каталога

Документация FreeBSD традиционно хранится в /usr/doc/, а исходный код системы с руководствами — в /usr/src/. Эти деревья каталогов могут быть перемещены, и пользователи могут разместить рабочие копии в других местах, чтобы избежать конфликтов с существующей информацией в основных каталогах. В следующих примерах используются ~/doc и ~/src — подкаталоги домашнего каталога пользователя.

3.3. Извлечение копии

Загрузка рабочей копии из репозитория называется *клоном* и выполняется командой `git clone`. В этом примере клонируется последняя версия (`main`) основного дерева документации:

```
% git clone https://git.FreeBSD.org/doc.git ~/doc
```

Получение исходного кода для работы со справочником (man) выполняется аналогично:

```
% git clone https://git.FreeBSD.org/src.git ~/src
```

3.4. Обновление рабочей копии

Документы и файлы в репозитории FreeBSD изменяются ежедневно. Пользователи вносят изменения в файлы и фиксируют их довольно часто. Даже спустя короткое время после первоначального получения копии репозитория, между локальной рабочей копией и основным репозиторием FreeBSD уже могут появиться различия. Чтобы обновить локальную версию с учётом изменений, внесённых в основной репозиторий, используйте команду `git pull` в каталоге с локальной рабочей копией:

```
% cd ~/doc
% git pull --ff-only
```

Приучите себя к полезной привычке использовать `git pull` перед редактированием файлов документации. Кто-то другой мог недавно изменить этот файл, и ваша локальная рабочая копия не будет содержать последних изменений, пока вы не обновите её. Редактировать самую свежую версию файла гораздо проще, чем пытаться объединить более старую, отредактированную локальную версию с новой версией из репозитория.

3.5. Отмена изменений

Иногда оказывается, что изменения были не нужны, или автор просто хочет начать заново. Файлы можно «сбросить» к их исходному состоянию с помощью `git restore`. Например, чтобы отменить правки, сделанные в _index.adoc, и вернуть его в исходное состояние:

```
% git restore _index.adoc
```

3.6. Создание Diff

После завершения редактирования файла или группы файлов, различия между локальной рабочей копией и версией в репозитории FreeBSD должны быть собраны в один файл для отправки. Эти *diff*-файлы создаются путём перенаправления вывода команды `git diff` в файл:

```
% cd ~/doc
% git diff > doc-fix-spelling.diff
```

Присвойте файлу осмысленное имя, отражающее его содержимое. В приведённом примере указаны исправления орфографии для всего дерева документации.

Если файл с различиями должен быть отправлен через веб-интерфейс "Сообщить о проблеме в FreeBSD", добавьте расширение .txt, чтобы дать простой и прямолинейной веб-форме понять, что содержимое представляет собой обычный текст.

Будьте осторожны: `git diff` включает все изменения в текущем каталоге и его подкаталогах. Если в рабочей копии есть файлы с правками, которые ещё не готовы к

отправке, укажите список только тех файлов, которые нужно включить:

```
% cd ~/doc
% git diff disks/_index.adoc printers/_index.adoc > disks-printers.diff
```

3.7. Ссылки Git

Эти примеры демонстрируют базовое использование Git. Более подробная информация доступна в Книге по Git и документации Git.

Глава 4. Структура каталогов документации

Файлы и каталоги в дереве **doc**/ следуют структуре, преследующей цели:

1. Сделать удобным автоматическое преобразование документа в другие форматы.
2. Способствовать согласованности между различными организациями, занимающимися документацией, чтобы упростить переход между работой над разными документами.
3. Сделать удобным выбор места в дереве для размещения новой документации.

В дополнение, дерево документации должно поддерживать документы на многих разных языках. Важно, чтобы структура дерева документации не навязывала каких-либо конкретных настроек по умолчанию или культурных предпочтений.

4.1. Верхний уровень, doc/

В разделе **doc**/ есть три подраздела, документация и веб-сайт имеют одинаковую структуру.

Каталог	Использование
documentation	Содержит все статьи и книги в формате AsciiDoc. Содержит подкаталоги для дальнейшей категоризации информации по языкам.
tools	Содержит набор инструментов для перевода документации и веб-сайта с использованием Weblate. Экземпляр Weblate доступен здесь.
shared	Содержит файлы, не относящиеся к различным переводам документации. Включает подкаталоги для дальнейшей категоризации информации по языкам, а также три файла для хранения данных об авторах, выпусках и зеркалах. Этот каталог является общим для `documentation` и `website`.
* website*	Содержит ссылку FreeBSD website в формате AsciiDoc. Содержит подкаталоги для дальнейшей категоризации информации по языкам.

4.2. Каталоги

Эти каталоги содержат документацию и веб-сайт. Документация организована в подкаталоги ниже этого уровня, следуя структуре каталогов Hugo.

Каталог	Использование
* archetypes*	Содержит шаблоны для создания новых статей, книг и веб-страниц. Дополнительную информацию смотрите здесь.
config	Содержат файлы конфигурации Hugo. Один основной файл и по одному файлу для каждого языка. Для получения дополнительной информации смотрите здесь.

Каталог	Использование
content	Содержат книги, статьи и веб-страницы. Для каждого доступного перевода документации существует отдельный каталог, например `en` и `zh-tw`.
data	Содержит пользовательские данные для сборки веб-сайта в формате TOML. Этот каталог используется для хранения событий, новостей, пресс-релизов и т.д. Для получения дополнительной информации ознакомьтесь здесь.
static	Содержат статические ресурсы. Изображения, бюллетени безопасности, pgpkeys и т.д. Подробнее смотрите здесь.
* themes*	Содержите шаблоны в виде файлов `.html`, которые определяют внешний вид веб-сайта. Для получения дополнительной информации посмотрите здесь.
tools	Содержит инструменты, используемые для улучшения сборки документации. Например, для генерации оглавления книг и т.д.
beastie.png	Это изображение не нуждается в представлении ;)
LICENSE	Лицензия документации, общих материалов и веб-сайта. Лицензия BSD с 2 пунктами.
Makefile	Файл **Makefile** определяет процесс сборки документации и веб-сайта.

4.3. Информация, относящаяся к документу

Этот раздел содержит особые примечания о конкретных документах, которыми управляет FDP.

4.4. Книги: books/

Книги написаны в AsciiDoc.

Для каждой книги FreeBSD тип документа AsciiDoc (также известный как doctype) — это `book` (книга). Книги содержат `part` (части), каждая из которых включает несколько `chapter` (глав).

Когда документ преобразуется в HTML 5 (с использованием встроенного бэкенда `html5`):

- Уровень раздела AsciiDoc 0 (`=`) в начале `главы книги` будет `<h1>`
- Уровень секции AsciiDoc 1 (`==`) должен использоваться для первого логического раздела главы и будет преобразован в `<h2>`
- Уровень раздела AsciiDoc 2 (`===`) должен использоваться для первого логического подраздела и будет преобразован в `<h3>`

– и так далее. Ссылка: Заголовки и уровни разделов | Документация Asciidoctor.

4.4.1. Физическая организация

В каталоге books находится множество файлов и каталогов, все с одинаковой структурой.

_index.adoc

Файл **_index.adoc** определяет некоторые переменные AsciiDoc, которые влияют на преобразование исходного кода AsciiDoc в другие форматы, а также содержит оглавление, список примеров, список рисунков, список таблиц и раздел с аннотацией.

book.adoc

Файл **book.adoc** определяет некоторые переменные AsciiDoc, которые влияют на преобразование исходного кода AsciiDoc в другие форматы, а также включает оглавление, список примеров, список рисунков, список таблиц, раздел с аннотацией и все главы. Этот файл используется для генерации PDF с помощью `asciidoctor-pdf` и для создания книги в виде одной страницы `html`.

part*.adoc

Файлы **part*.adoc** содержат краткое введение к одной части книги.

directory/_index.adoc

Каждая глава Руководства хранится в файле с именем **_index.adoc** в отдельном каталоге от других глав.

Например, вот пример заголовка одной главы:

```
---
title: Chapter 8. Configuring the FreeBSD Kernel
part: Part II. Common Tasks
prev: books/handbook/multimedia
next: books/handbook/printing
---

[[kernelconfig]]
= Настройка ядра FreeBSD
...
```

При создании HTML5-версии Handbook это приведёт к формированию файла **kernelconfig/index.html**.

Быстрый взгляд покажет, что существует множество каталогов с отдельными файлами **_index.adoc**, включая **basics/_index.adoc**, **introduction/_index.adoc** и **printing/_index.adoc**.



Не называйте главы или каталоги в соответствии с их порядком в Руководстве. Этот порядок может измениться при реорганизации содержания Руководства. Реорганизация должна быть возможной без

переименования файлов, за исключением случаев, когда целые главы перемещаются вверх или вниз по иерархии.

4.5. Статьи: articles/

Статьи написаны в AsciiDoc.

Статьи организованы как документ AsciiDoc `article`. Статьи разделены на разделы (`=`), подразделы (`==`, `===`) и так далее.

4.5.1. Физическая организация

На каждый статью приходится один файл **_index.adoc**.

_index.adoc

Файл **_index.adoc** содержит все переменные AsciiDoc и контент.

Например, это образец одной статьи, структура довольно похожа на главу книги:

```
---
title: Why you should use a BSD style license for your Open Source Project
authors:
  - author: Bruce Montague
    email: brucem@alumni.cse.ucsc.edu
trademarks: ["freebsd", "intel", "general"]
---

= Почему вы должны использовать лицензию в стиле BSD для вашего Open Source проекта
:doctype: article :toc: macro :toclevels: 1 :icons: font :sectnums: :source-
highlighter: rouge :experimental:

'''

toc::[]

[[intro]]
== Введение
```

4.6. Управление списками участников

Проект FreeBSD отмечает вклад участников в нескольких различных списках в документации и печатных материалах. В этом разделе описано, как команда документации управляет изменениями в этих списках.

4.6.1. Списки отношений наставника и нового коммиттера

Начиная с FreeBSD 7.0, FreeBSD поддерживает три списка отношений

наставник/подопечный — один для исходного кода, один для портов и один для документации. Эти файлы имеют формат **«.dot»** и предназначены для использования с популярным пакетом для построения графиков graphics/graphviz[], доступным как пакет или порт FreeBSD.



dot(1) устанавливается как часть пакета или порта: graphics/graphviz[]. Программа dot читает файлы в формате **".dot"** и создаёт графическое изображение направленного графа.

Три файла часто служат учебным опытом для новых коммиттеров всех трёх команд, которым поручается добавить себя и своего наставника в соответствующий файл в качестве их первого коммита. Каждый файл содержит раздел «current» для новых коммиттеров, раздел «alumni» для случаев возврата прав на коммит и раздел «mentor / mentee», отображающий взаимоотношения. Формат каждой записи поясняется в начале файла.

4.6.2. Общий список участников

Участники проекта FreeBSD поддерживаются в формате статьи. Исходный файл для управления статьёй **Участники** находится по адресу:

```
doc
 /documentation
   /content
     /{language}
       /articles
         /contributors
           _index.adoc – Contains a list of include files that apply to each section.
           _index.po – Translation page
           contrib-develinmemoriam.adoc - content of “In Memoriam” section
           contrib-develinmemoriam.po – Translation page
```

Копии этого каталога contributors могут существовать в других каталогах с контентом на различных языках.

Обратите внимание, что файл contrib-develinmemoriam.adoc также находится в этом каталоге. Дополнительную информацию смотрите ниже.

Файл contributors/_index.adoc представляет собой набор включаемых файлов. Включаемые файлы перечислены в разделе, специфичном для Hugo, в исходном файле. Раздел разделён на несколько частей с помощью операторов "ifdef::". Существует подраздел для вывода на веб-сайт и подраздел для вывода вне веб-сайта (включая PDF).

Текст каждого раздела страницы **Участники** содержит оператор "include::". Например, запись для "Экс-участники менеджера портов" выглядит как `include::{include-contrib-portmgralumni}[]`. Это подключает текст об экс-участниках менеджера портов в итоговый вывод.

Чтобы внести изменения, отредактируйте соответствующий include-файл:

```
include-contrib-committers:     ~/doc/shared/contrib-committers.adoc
include-contrib-corealumni:     ~/doc/shared/contrib-corealumni.adoc
include-contrib-develalumni:    ~/doc/shared/contrib-develalumni.adoc
include-contrib-portmgralumni:  ~/doc/shared/contrib-portmgralumni.adoc
include-contrib-additional:     ~/doc/shared/contrib-additional.adoc
include-contrib-386bsd:         ~/doc/shared/contrib-386bsd.adoc

Также отредактируйте файл authors.adoc:  ~/doc/shared/authors.adoc
и все связанные с ним переводы.
```

В общем случае, если добавляется новый человек, требуется лишь одно изменение — его можно добавить в соответствующий включаемый файл. Если человек переводится из статуса «текущий» в «бывший», необходимо два изменения: одно для удаления и одно для добавления в соответствующие файлы. Порядок записей для всех файлов указан в таблице ниже.

Назначение	**Якорь раздела**	**Файл в ~/doc/shared/**	**Спецификация порядка**
Разработчики FreeBSD	include-contrib-committers	contrib-committers.adoc	упорядочено по алфавиту по фамилии
Бывшие члены основной команды	include-contrib-corealumni	contrib-corealumni.adoc	приблизительно в обратном хронологическом порядке
Бывшие члены команды разработчиков	include-contrib-develalumni	contrib-develalumni.adoc	приблизительно в обратном хронологическом порядке
Бывшие члены команды управления портами	include-contrib-portmgralumni	contrib-portmgralumni.adoc	приблизительно в обратном хронологическом порядке
Дополнительные участники проекта FreeBSD	include-contrib-additional	contrib-additional.adoc	упорядочено по алфавиту по имени
Участники, предоставившие патчи для 386BSD Patch Kit	include-contrib-386bsd	contrib-386bsd.adoc	упорядочено по алфавиту по имени
Участники проекта центрального сервера	Файл include не используется	contributors/_index.adoc	неупорядоченное
Прямое финансирование	Файл include не используется	contributors/_index.adoc	неупорядоченное

Назначение	Якорь раздела	Файл в ~/doc/shared/	Спецификация порядка
Участники, предоставившие оборудование	Файл include не используется	contributors/_index.adoc	неупорядоченное
Особые участники	Файл include не используется	contributors/_index.adoc	неупорядоченное

4.6.3. Раздел "Памяти"

Когда уведомляют сообщество BSD о смерти члена сообщества, следует использовать следующую процедуру:

1. Используйте файл ~/doc/shared/authors.adoc для поиска имени человека и ссылки на атрибут, например `{foobsd}`.
2. Если они являются текущим членом одной или нескольких команд проекта FreeBSD в ~/doc/website/content/en/administration.adoc, удалите все упоминания их атрибута. Также выполните следующие правки:
   - ~/doc/shared/contrib-committers.adoc - Удалите ссылку на атрибут.
   - ~/doc/shared/contrib-corealumni.adoc - Если они являются *действующим* членом основной команды, создайте запись с указанием дат начала и окончания.
   - ~/doc/shared/contrib-develalumni.adoc - Добавьте ссылку на атрибут и даты активности в качестве коммиттера.
   - ~/doc/shared/contrib-portmgralumni.adoc - Добавьте ссылку на атрибут при необходимости.
   - ~/doc/shared/contrib-additional.adoc — Удалите запись.
   - ~/doc/shared/contrib-386bsd.adoc - Это исключительно исторический документ. Изменения не требуются.
3. В файле ~/doc/shared/authors.adoc закомментируйте (используя одну обратную косую черту '\') адрес электронной почты, чтобы избежать создания ссылки "mailto:". См. пример для `itojun` ниже:

```
[shared/authors.adoc]

[..]

:itojun-name: Jun-ichiro Itoh
:itojun-email: \itojun@FreeBSD.org
:itojun: {itojun-name} <{itojun-email}>

[..]
```

4. Поскольку участник скончался (что следует перепроверить), добавьте его в файл contrib-

develinmemoriam.adoc "Памяти ушедших".

Постарайтесь найти информацию, подкрепленную фактами, о его вкладе в FreeBSD за прошедшие годы и добавьте её к записи в файле. Для этого может потребоваться запрос в списках рассылки разработчиков, обращение к коллегам, связь с Фондом FreeBSD или поиск в журналах коммитов.

Чтобы узнать дату их первого коммита, используйте:

```
% cd ~/src
% git log --reverse --author=foobsd     # search for first commit of foobsd
```

Это выведет их коммиты в обратном порядке. Дата первого коммита будет вверху.

Убедитесь, что формат дат соответствует другим записям:

```
(год начала предоставления права коммита - год окончания предоставления права
коммита; RIP год ухода)

Например:

* Foo BSD (2007 - 2010; RIP 2016)
```

Проверьте порядок записей в файле.

Назначение	**Якорь раздела**	**Файл в ~/doc/documentation/content/{язык}/articles/contributors/**	**Спецификация порядка**
Команда разработчиков: Памяти ушедших	contrib-develinmemoriam.adoc	contrib-develinmemoriam.adoc	приблизительно в обратном хронологическом порядке

Смотрите файл "In Memoriam" для похожих записей.

5. Наконец, если применимо, переместите запись участника с правами коммиттера из раздела «current» в раздел «alumni» соответствующего списка отношений наставник / подопечный с указанием соответствующей даты. Изменять отношения наставник / подопечный не требуется.

Глава 5. Процесс сборки документации FreeBSD

Эта глава описывает организацию процесса сборки документации и использование make(1) для управления этим процессом.

5.1. Преобразование AsciiDoc в выходные форматы

Из одного исходного файла AsciiDoc можно получить различные типы выходных данных.

Форматы	Тип файла	Описание
`html`	HTML	Глава статьи (`article`) или книги (`book`).
`pdf`	PDF	Portable Document Format.
`epub`	EPUB	Электронная публикация. Формат файла ePub.

5.1.1. Преобразование в html

Для преобразования документации и веб-сайта в `html` используйте один из следующих примеров.

Пример 2. Собрать документацию

```
% cd ~/doc/documentation
% make
```

Пример 3. Собрать веб-сайт

```
% cd ~/doc/website
% make
```

Пример 4. Собрать весь проект документации

```
% cd ~/doc
% make -j2
```

Ниже приведены примеры сложных сборок:

Пример 5. Собрать документацию на английском и испанском языках с подробными и отладочными сообщениями

```
% cd ~/doc/documentation
% make DOC_LANG="en es" HUGO_ARGS="--verbose --debug"
```

Пример 6. Собрать и предоставить контент с помощью внутреннего веб-сервера Hugo

```
% cd ~/doc/documentation
% make run
```

Этот веб-сервер по умолчанию работает на `localhost`, порт `1313`.

Для обслуживания контента с помощью внутреннего веб-сервера Hugo, привязанного к определённому IP-адресу и порту:

```
% make run BIND=192.168.15.10 HUGO_ARGS="-p 8080"
```

Имя хоста (`hostname`) также может быть установлено в качестве базового URL для внутреннего веб-сервера Hugo:

```
% make run BIND=192.168.15.10 HOSTNAME=example.com
```

Пример 7. Сборка документации в html для использования в автономном режиме

```
% cd ~/doc/documentation
% make html
```

Для сжатия вывода в формате html добавьте `DOC_HTML_ARCHIVE=1`:

```
% cd ~/doc/documentation
% DOC_HTML_ARCHIVE=1 make html
```

5.1.2. Рендеринг в pdf

Для преобразования документации в `pdf` используйте один из следующих примеров.

Пример 8. Собрать все документы в pdf

```
% cd ~/doc/documentation
```

```
% make pdf
```

Пример 9. Собрать все статьи в pdf

```
% cd ~/doc/documentation
% make pdf-articles
```

Пример 10. Собрать все книги в формате pdf

```
% cd ~/doc/documentation
% make pdf-books
```

Пример 11. Сборка документов в pdf для определённых языков

```
% cd ~/doc/documentation
% make DOC_LANG="en" pdf
```

Это соберет все документы на английском языке в формате pdf.

```
% cd ~/doc/documentation
% make DOC_LANG="en fr" pdf-books
```

Это соберет все книги на английском и французском языках в формате pdf.

5.2. Набор инструментов для сборки документации FreeBSD

Вот инструменты, используемые для сборки и установки документации FDP.

- Основным инструментом сборки является make(1), а именно Berkeley Make.
- Hugo
- AsciiDoctor
- Git

5.3. Информация о Makefile в дереве документации

В проекте документации есть три файла Makefile для сборки всей или части документации.

- Файл Makefile в каталоге documentation предназначен только для сборки документации.

- Файл Makefile в каталоге website предназначен только для сборки веб-сайта.
- Makefile в корне дерева исходников собирает как документацию, так и веб-сайт.

Файл Makefile в подкаталогах также поддерживает `make run` для обслуживания собранного содержимого с помощью внутреннего веб-сервера Hugo. По умолчанию этот веб-сервер работает на порту 1313.

5.3.1. Makefile в каталоге documentation

Этот Makefile имеет следующую форму:

```
# Generate the FreeBSD documentation
#
# Copyright (c) 2020-2026, The FreeBSD Documentation Project
# Copyright (c) 2020-2026, Sergio Carlavilla <carlavilla@FreeBSD.org>
#
# Targets intended for use on the command line
#
# all (default) -   generate the books TOC and compile all the documentation
# clean     -   removes generated files
# run       -   serves the built documentation site for local browsing
# pdf       -   build PDF versions of the articles and books.
# html      -   build HTML versions of the articles and books for
#           offline use.
#           If variable DOC_HTML_ARCHIVE is set, all documents will be
#           archived/compressed, and only these files will be kept in the public
#           directory.
# epub      -   build EPUB versions of the articles and books (Experimental).
#
# The run target uses hugo's built-in webserver to make the documentation site
# available for local browsing.  The documentation should have been built prior
# to attempting to use the `run` target.  By default, hugo will start its
# webserver on port 1313.

MAINTAINER=carlavilla@FreeBSD.org ①

# List of languages without book translations
ARTICLEONLY_LANGS=  bn-bd da ko tr
# List of languages without article translations
BOOKONLY_LANGS=     mn

# List of all languages we have content for
ALL_LANGUAGES=  bn-bd da de el en es fr hu it ja ko mn nl pl pt-br ru tr zh-cn zh-tw
②

LOCALBASE?= /usr/local

RUBY_CMD =  ${LOCALBASE}/bin/ruby ③
HUGO_CMD =  ${LOCALBASE}/bin/hugo ④
HUGO_ARGS?= --verbose --minify
```

```
HUGO_OFFLINE_ARGS?=     --environment offline --verbose --minify
ASCIIDOCTOR_CMD=    ${LOCALBASE}/bin/asciidoctor
ASCIIDOCTORPDF_CMD= ${LOCALBASE}/bin/asciidoctor-pdf

.if defined(DOC_LANG) && !empty(DOC_LANG)
LANGUAGES=  ${DOC_LANG:S/,/ /g}
.if  ${LANGUAGES:Men} == "" && ${.TARGETS:Mpdf*} == "" && ${.TARGETS:Mhtml*} == ""
.warning "Warning: cannot skip 'en'; adding it back"
LANGUAGES+= en
.endif
.else
LANGUAGES=  ${ALL_LANGUAGES}
.endif

RUBYLIB =   ../shared/lib
.export RUBYLIB

RUN_DEPENDS=    ${HUGO_CMD} \
        ${LOCALBASE}/bin/asciidoctor \
        ${LOCALBASE}/bin/rougify

.ifndef HOSTNAME
.  ifdef BIND
.HOST=$(BIND)
.  else
.HOST=localhost
.  endif
.else
.HOST=$(HOSTNAME)
.endif

# Strip the languages with only articles from the list of languages we
#  will use to build books.
BOOK_LANGS= ${LANGUAGES}
.for a in ${ARTICLEONLY_LANGS}
BOOK_LANGS:=    ${BOOK_LANGS:N${a}}
.endfor

# Strip the languages with only books from the list of languages we
#  will use to build articles.
ARTICLE_LANGS= ${LANGUAGES}
.for a in ${BOOKONLY_LANGS}
ARTICLE_LANGS:= ${ARTICLE_LANGS:N${a}}
.endfor

# Take the list of all languages, and take out the ones we have been
#   asked for.  We'll feed this to hugo.
SKIP_LANGS=
.for a in ${ALL_LANGUAGES}
.if  ${LANGUAGES:M${a}} == ""
SKIP_LANGS+=    ${a}
```

```
.endif
.endfor

.ORDER: all run ⑤

.ORDER: requirements ⑥
.ORDER: starting-message
.ORDER: starting-message build
.ORDER: build

all: requirements starting-message generate-pgpkeys-txt build
run: requirements starting-message generate-pgpkeys-txt run-local

# clean does not call pdf-clean as that is a subset of hugo-clean
clean: hugo-clean pgp-clean

requirements:
.for dep in ${RUN_DEPENDS}
.if !exists(${dep})
    @(echo ${dep} not found, please run 'pkg install docproj'; exit 1)
.endif
.endfor

requirements-pdf:
.if !exists(${LOCALBASE}/bin/asciidoctor-pdf)
    @(echo ${LOCALBASE}/bin/asciidoctor-pdf not found, please run 'pkg install
rubygem-asciidoctor-pdf'; exit 1)
.endif

requirements-epub:
.if !exists(${LOCALBASE}/bin/asciidoctor-epub3)
    @(echo ${LOCALBASE}/bin/asciidoctor-epub3 not found, please run 'pkg install
rubygem-asciidoctor-epub3'; exit 1)
.endif

starting-message: .PHONY ⑦
    @echo ---------------------------------------------------------------
    @echo                    Building the documentation
    @echo  included languages: ${LANGUAGES}
    @echo  excluded languages: ${SKIP_LANGS}
    @echo ---------------------------------------------------------------

generate-pgpkeys-txt: static/pgpkeys/pgpkeys.txt

static/pgpkeys/pgpkeys.txt: static/pgpkeys/*key
    ${RUBY_CMD} ./tools/global-pgpkeys-creator.rb

run-local: .PHONY ⑧
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} server \
        ${HUGO_ARGS} -D $(BIND:D--bind=$(BIND)) --baseURL="http://$(.HOST):1313"
```

```
build: .PHONY ⑨
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} ${HUGO_ARGS}

build-offline: .PHONY
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} ${HUGO_OFFLINE_ARGS}

pgp-clean: .PHONY
    rm -f static/pgpkeys/pgpkeys.txt

hugo-clean: .PHONY
    rm -rf resources public

#
# PDF targets
# Use DOC_LANG to choose the language, e.g., make DOC_LANG="en fr" pdf-books
#
pdf: pdf-articles pdf-books

pdf-books: requirements-pdf
.for _lang in ${BOOK_LANGS}
    ./tools/asciidoctor.sh books ${_lang} pdf
.endfor

pdf-articles: requirements-pdf
.for _lang in ${ARTICLE_LANGS}
    ./tools/asciidoctor.sh articles ${_lang} pdf
.endfor

pdf-clean: pdf-articles-clean pdf-books-clean

pdf-books-clean:
.for _lang in ${BOOK_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/books
    -rmdir ${.CURDIR}/public/${_lang}
.endfor
    -rmdir ${.CURDIR}/public/

pdf-articles-clean:
.for _lang in ${ARTICLE_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/articles
.if !exists(${.CURDIR}/public/${_lang}/books)
    rm -fr ${.CURDIR}/public/${_lang}
.endif
.endfor
    -rmdir ${.CURDIR}/public

#
# HTML targets
#
html: build-offline html-clean-global html-clean-articles html-clean-books html-
archive html-archive-clean-files
```

```
html-clean: hugo-clean

html-clean-global:
    rm -fr ${.CURDIR}/public/index.html
    rm -rf pgpkeys js

html-clean-articles:
.for _lang in ${ARTICLE_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/index.html
    rm -fr ${.CURDIR}/public/${_lang}/articles/index.html
.endfor

html-clean-books:
.for _lang in ${BOOK_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/books/index.html
.endfor

html-archive:
.if defined(DOC_HTML_ARCHIVE)
.for _lang in ${ARTICLE_LANGS}
    ./tools/asciidoctor.sh articles ${_lang} archive
.endfor
.for _lang in ${BOOK_LANGS}
    ./tools/asciidoctor.sh books ${_lang} archive
.endfor
.endif

html-archive-clean-files:
.if defined(DOC_HTML_ARCHIVE)
    find ${.CURDIR}/public/ ! -name '*.pdf' ! -name '*.tar.gz' -type f -delete
    find ${.CURDIR}/public/ -type d -empty -delete
.endif

#
# EPUB targets
# Use DOC_LANG to choose the language, e.g., make DOC_LANG="en fr" epub-books
#
epub: epub-articles epub-books

epub-books: requirements-epub
    @echo ----------------------------------------------------------------
    @echo !!! EPUB output is experimental !!!
    @echo
    @echo Asciidoctor EPUB3 is currently alpha software. Use accordingly. Although the
    @echo bulk of AsciiDoc content is converted, there's still work needed to fill in
    @echo gaps where conversion is incomplete or unstyled.
    @echo https://docs.asciidoctor.org/epub3-converter/latest/#project-status
    @echo ----------------------------------------------------------------
.for _lang in ${BOOK_LANGS}
    ./tools/asciidoctor.sh books ${_lang} epub
```

```
.endfor

epub-articles: requirements-epub
    @echo ----------------------------------------------------------------
    @echo !!! EPUB output is experimental !!!
    @echo
    @echo Asciidoctor EPUB3 is currently alpha software. Use accordingly. Although the
    @echo bulk of AsciiDoc content is converted, there's still work needed to fill in
    @echo gaps where conversion is incomplete or unstyled.
    @echo https://docs.asciidoctor.org/epub3-converter/latest/#project-status
    @echo ----------------------------------------------------------------
.for _lang in ${ARTICLE_LANGS}
    ./tools/asciidoctor.sh articles ${_lang} epub
.endfor

epub-clean: epub-articles-clean epub-books-clean

epub-books-clean:
.for _lang in ${BOOK_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/books
    -rmdir ${.CURDIR}/public/${_lang}
.endfor
    -rmdir ${.CURDIR}/public/

epub-articles-clean:
.for _lang in ${ARTICLE_LANGS}
    rm -fr ${.CURDIR}/public/${_lang}/articles
.if !exists(${.CURDIR}/public/${_lang}/books)
    rm -fr ${.CURDIR}/public/${_lang}
.endif
.endfor
    -rmdir ${.CURDIR}/public
```

① Флаг `MAINTAINER` указывает, кто является сопровождающим данного Makefile.

② Флаг `ALL_LANGUAGES` указывает, на каких языках должно быть сгенерировано оглавление.

③ Флаг `RUBY_CMD` указывает расположение бинарного файла Ruby.

④ `HUGO_CMD` — флаг, указывающий расположение бинарного файла Hugo.

⑤ Директивы `.ORDER` используются для обеспечения беспроблемного выполнения нескольких заданий make.

⑥ Цель `all` собирает документацию и помещает результат в **~/doc/documentation/public**.

⑦ `starting-message` показывает сообщение в CLI, чтобы уведомить пользователя о том, что процесс выполняется.

⑧ `run-local` запускает веб-сервер hugo на порту 1313 или на случайном свободном порту, если указанный порт уже занят.

⑨ `build` собирает документацию и помещает результат в **~/doc/documentation/public**.

5.3.2. Makefile в каталоге website

Этот Makefile имеет следующий вид:

```
# Generate the FreeBSD website
#
# Copyright (c) 2020-2026, The FreeBSD Documentation Project
# Copyright (c) 2020-2026, Sergio Carlavilla <carlavilla@FreeBSD.org>
#
# Targets intended for use on the command line
#
# all (default) -   generate the releases.toml and compile all the website
# run   -           serves the built website for local browsing
#
# The run target uses hugo's built-in webserver to make the built website
# available for local browsing.  The website should have been built prior
# to attempting to use the `run` target.  By default, hugo will start its
# webserver on port 1313.

MAINTAINER=carlavilla@FreeBSD.org ①

# List of all languages we have content for
ALL_LANGUAGES=  de el en es fr hu it ja nl ru tr zh-cn zh-tw

LOCALBASE?= /usr/local

RUBY_CMD =  ${LOCALBASE}/bin/ruby ②
HUGO_CMD =  ${LOCALBASE}/bin/hugo ③
HUGO_ARGS?= --verbose
RUBYLIB =   ../shared/lib
.export RUBYLIB

.ifndef HOSTNAME
.  ifdef BIND
.HOST=$(BIND)
.  else
.HOST=localhost
.  endif
.else
.HOST=$(HOSTNAME)
.endif

.if defined(DOC_LANG) && !empty(DOC_LANG)
LANGUAGES=      ${DOC_LANG:S/,/ /g}
.if  ${LANGUAGES:Men} == ""
.warning "Warning: cannot skip 'en'; adding it back"
LANGUAGES+= en
.endif
.else
LANGUAGES=  ${ALL_LANGUAGES}
```

```
.endif

# Take the list of all languages, and take out the ones we have been
#   asked for via DOC_LANG.  We'll feed this to hugo.
SKIP_LANGS=
.for a in ${ALL_LANGUAGES}
.if ${LANGUAGES:M${a}} == ""
SKIP_LANGS+=    ${a}
.endif
.endfor

.ORDER: all run ④

.ORDER: starting-message generate-releases
.ORDER: starting-message build
.ORDER: generate-releases build
.ORDER: build post-build
.ORDER: post-build end-message

all: starting-message generate-releases build post-build end-message ⑤
run: starting-message generate-releases run-local
clean: hugo-clean releases-clean

starting-message: .PHONY ⑥
    @echo "------------------------------------------------------------"
    @echo "Building the website started on $$(date)"
    @echo " included languages: ${LANGUAGES}"
    @echo " excluded languages: ${SKIP_LANGS}"
    @echo "------------------------------------------------------------"

end-message: .PHONY
    @echo "------------------------------------------------------------"
    @echo "Building the website completed on $$(date)"
    @echo "------------------------------------------------------------"

generate-releases: data/releases.toml ⑦

data/releases.toml:
    ${RUBY_CMD} ./tools/releases-toml.rb

run-local: .PHONY ⑧
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} server \
        ${HUGO_ARGS} -D $(BIND:D--bind=$(BIND)) --baseURL="http://$(.HOST):1313"

build: .PHONY ⑨
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} ${HUGO_ARGS}

post-build: cgi-permissions

cgi-permissions:
    @chmod 555 ./public/cgi/*.cgi
```

```
hugo-clean:
    rm -fr public resources

releases-clean:
    rm -f data/releases.toml
```

① Флаг `MAINTAINER` указывает, кто является сопровождающим данного Makefile.

② Флаг `RUBY_CMD` указывает расположение бинарного файла Ruby.

③ `HUGO_CMD` — флаг, указывающий расположение бинарного файла Hugo.

④ Директивы `.ORDER` используются для обеспечения беспроблемного выполнения нескольких заданий make.

⑤ Цель `all` собирает веб-сайт и помещает результат в **~/doc/website/public**.

⑥ `starting-message` показывает сообщение в CLI, чтобы уведомить пользователя о том, что процесс выполняется.

⑦ `generate-releases` вызывает скрипт, используемый для преобразования переменных AsciiDoc в переменные TOML. После этого преобразования переменные релизов можно использовать как в AsciiDoc, так и в пользовательских шаблонах Hugo.

⑧ `run-local` запускает веб-сервер hugo на порту 1313 или на случайном свободном порту, если указанный порт уже занят.

⑨ `build` собирает веб-сайт и помещает результат в **~/doc/website/public**.

Глава 6. Основы Asciidoctor

Большая часть документации FDP написана с использованием AsciiDoc. В этой главе объясняется, что это значит, как читать и понимать исходный код документации, а также используемые методы. Для получения полного справочника по возможностям Asciidoctor обратитесь к документации Asciidoctor. Некоторые примеры, используемые в этой главе, взяты из краткого справочника по синтаксису AsciiDoc.

6.1. Обзор

В первые дни существования компьютеров электронный текст был простым. Существовало несколько наборов символов, таких как ASCII или EBCDIC, но на этом всё и заканчивалось. Текст был текстом, и вы видели именно то, что получали. Никаких изысков, никакого форматирования, никакого интеллекта.

Неизбежно, этого оказалось недостаточно. Когда текст представлен в формате, пригодном для машинной обработки, ожидается, что машины смогут использовать и обрабатывать его интеллектуально. Авторы хотят указывать, что определённые фразы должны быть выделены, добавлены в глоссарий или преобразованы в гиперссылки. Имена файлов могут отображаться моноширинным шрифтом при просмотре на экране, но курсивом при печати или в любом другом из множества вариантов представления.

Однажды надеялись, что искусственный интеллект (ИИ) сделает это легко. Компьютер прочитает документ и автоматически определит ключевые фразы, имена файлов, текст, который читатель должен ввести, примеры и многое другое. К сожалению, в реальной жизни всё оказалось не так просто, и компьютерам до сих пор требуется помощь, прежде чем они смогут осмысленно обрабатывать текст.

Точнее говоря, им нужна помощь в определении, что есть что. Рассмотрим этот текст:

Для удаления /tmp/foo используйте rm(1).

```
% rm /tmp/foo
```

Читателю легко понять, какие части являются именами файлов, какие — командами для ввода, какие — ссылками на страницы руководства и так далее. Однако компьютер, обрабатывающий документ, не может надёжно определить это. Для этого нам нужна разметка.

Предыдущий пример фактически представлен в этом документе следующим образом:

```
To remove */tmp/foo*, use man:rm[1].

[source,shell]
----
% rm /tmp/foo
```

```
----
```

6.2. Заголовки

Asciidoctor поддерживает шесть уровней заголовков. Если тип документа `article`, можно использовать только один заголовок уровня 0 (`=`). Если тип документа `book`, то может быть несколько заголовков уровня 0 (`=`).

Вот пример заголовков в документе типа `article`.

```
= Заголовок документа (Уровень 0)

== Заголовок раздела на уровне 1

=== Заголовок раздела на уровне 2

==== Заголовок раздела на уровне 3

===== Заголовок раздела на уровне 4

====== Заголовок раздела на уровне 5

== Другой заголовок раздела на уровне 1
```



Уровни разделов нельзя пропускать при вложении разделов.

Следующий синтаксис неверен.

```
= Заголовок документа

== Уровень 1

==== Уровень 3
```

6.3. Абзацы

Абзацы не требуют специальной разметки в AsciiDoc. Абзац определяется одной или несколькими последовательными строками текста. Чтобы создать новый абзац, оставьте одну пустую строку.

Например, это заголовок с двумя абзацами.

```
= Это заголовок

Это первый абзац.
Это тоже первый абзац.
```

```
А это второй абзац.
```

6.4. Списки

Asciidoctor поддерживает несколько типов списков, наиболее распространённые — `ordered` и `unordered`. Для получения дополнительной информации о списках см. AsciiDoc Syntax Quick Reference.

6.4.1. Упорядоченные списки

Для создания нумерованного списка используйте символ `.`.

Например, это нумерованный список.

```
. Первый пункт
. Второй пункт
.. Подпункт второго пункта
. Третий пункт
```

И это будет отображено как.

1. Первый пункт
2. Второй пункт
    a. Подпункт второго пункта
3. Третий пункт

6.4.2. Неупорядоченные списки

Для создания маркированного списка используйте символ `*`.

Например, это ненумерованный список.

```
* Первый пункт
* Второй пункт
** Подпункт второго пункта
* Третий пункт
```

И это будет отображено как.

- Первый пункт
- Второй пункт
    - Подпункт второго пункта
- Третий пункт

6.5. Ссылки

6.5.1. Внешние ссылки

Чтобы указать на другой веб-сайт, следует использовать макрос `link`.

```
link:https://www.FreeBSD.org[FreeBSD]
```



Как описано в документации Asciidoctor, макрос `link` не требуется, когда цель начинается со схемы URL, такой как `https`. Тем не менее, рекомендуется всё равно использовать его, чтобы гарантировать корректное отображение ссылки в Asciidoctor, особенно в языках с нелатинской письменностью, таких как японский.

6.5.2. Ссылки на другую книгу или статью

Для указания на другую книгу или статью следует использовать переменные Asciidoctor. Например, если мы находимся в статье `cups` и хотим сослаться на `ipsec-must`, необходимо выполнить следующие шаги.

1. Включите файл urls.adoc из папки ~/doc/shared.

   ```
   include::shared/{lang}/urls.adoc[]
   ```

2. Затем создайте ссылку с использованием переменной Asciidoctor на статью `ipsec-must`.

   ```
   extref:{ipsec-must}[Статья IPSec-Must]
   ```

   И это будет отображено как.

   Статья IPSec-Must



Макрос `extref` определён как расширение. Он предназначен для корректного отображения ссылки в различных выходных форматах

6.5.3. Ссылки на тот же файл или на другой файл в той же книге

Книги структурированы в разных каталогах для поддержания удобной организации. Чтобы создать ссылку из одного подкаталога книги в другой подкаталог той же книги, используйте макрос `crossref`:

```
crossref:doc-build[documentation-makefile, Эта ссылка]
```

И это будет отображено как

Эта ссылка



Макрос `crossref` определен как расширение. Он предназначен для формирования корректной ссылки в различных выходных форматах



Используйте макрос `crossref` для внутридокументных ссылок. Хотя указание имени текущего документа может быть неудобным, это гарантирует корректное отображение ссылки в различных выходных форматах



Не используйте макрос `xref` или его сокращение `<< >>`. Они не работают хорошо во всех выходных форматах.



Когда второй параметр макроса `crossref` содержит запятую, он должен быть заключен в двойные кавычки. В противном случае запятая интерпретируется как разделитель параметров, и текст после запятой отбрасывается.

Для примера:

```
crossref:plist[documentation-comma, "First, second"]
```

Без кавычек в качестве текста ссылки использовалось бы только слово `First`, а остальная часть была бы потеряна.

6.6. Изображения и иконки

Изображения и иконки играют ключевую роль в улучшении общего пользовательского опыта. Эти визуальные элементы стратегически интегрированы для передачи информации, пояснения концепций и создания визуально привлекательного интерфейса.

6.6.1. Изображения

Изображения помогают проиллюстрировать сложные концепции, делая их более понятными для пользователей.

Первым шагом будет добавление изображения в каталог `images` по пути:

- ~/website/static/images/ для веб-сайта.
- ~/documentation/static/images/ для документации.

Например, чтобы добавить новое изображение в процесс установки FreeBSD, изображение сохраняется по пути ~/documentation/static/images/books/handbook/bsdinstall/new-image3.png.

Следующим шагом будет настройка атрибутов Asciidoctor `images-path` и `imagesdir`.

В качестве примера мы используем заголовок статьи Подготовка релизов FreeBSD.

```
= Подготовка релизов FreeBSD
:doctype: article

[...]

:images-path: articles/freebsd-releng/ ①

[...]

:imagesdir: ../../../images/{images-path} ②

[...]
```

① Ссылается на путь внутри папки /static/images.

② Делает ссылку на атрибут Asciidoctor.

Как только изображение окажется в нужном пути и атрибуты Asciidoctor будут настроены в документе, можно использовать макрос `image`.

Вот пример:

```
image::new-image3.png[New step in the FreeBSD install process]
```



Для улучшения доступности обязательно добавлять описательный текст к каждому изображению.

6.6.2. Иконки

Иконки служат интуитивно понятными символами для быстрого распознавания и навигации.

Первым шагом для использования иконок является добавление свойства `icons` в раздел свойств Asciidoctor в начале каждого документа.

```
:icons: font
```

После установки свойства иконки Asciidoctor можно добавить иконку, поддерживаемую Font Awesome.

Это пример использования иконки `envelope`:

```
icon:envelope[link=mailto:test@example.com, title="contact"]
```



Для повышения доступности веб-сайта атрибут `title` является обязательным.

6.7. Заключение

Это заключение введения в Asciidoctor. Из-за ограничений по объёму и сложности некоторые аспекты не были рассмотрены глубоко (или вообще не затронуты).

Глава 7. Розеттский камень

7.1. Сравнение Docbook и AsciiDoc

Эта шпаргалка пытается показать различия между Docbook и AsciiDoc.

Таблица 1. Сравнение Docbook и AsciiDoc

Языковая функция	**Docbook**	**AsciiDoc**
Жирный	жирный	*жирный*
Курсив	<emphasis>Курсив</emphasis>	_Курсив_
Моноширинный	<literal>Моноширинный</literal>	`Моноширинный`
Абзац	<para>Это абзац</para>	Это абзац
Клавиша	<keycap>F11</keycap>	kbd:[F11]
Ссылки	`<link xlink:href="https://www.freebsd.org/where/">Загрузить FreeBSD</link>`	`link:https://www.freebsd.org/where/[Загрузить FreeBSD]`
Разделы	`<sect1 xml:id="id"><title>Раздел 1</title></sect1>`	`[[id]]= Раздел 1`
Неупорядоченный список	`<itemizedlist><listitem><para>Когда следует собирать собственное ядро.</para></listitem><listitem><para>Как выполнить инвентаризацию оборудования.</para></listitem></itemizedlist>`	`* Когда необходимо собрать собственное ядро.* Как выполнить инвентаризацию оборудования.`

Языковая функция	Docbook	AsciiDoc
Упорядоченный список	<orderedlist> <listitem> <para>Один</para> </listitem> <listitem> <para>Два</para> </listitem> <listitem> <para>Три</para> </listitem> <listitem> <para>Четыре</para> </listitem> </orderedlist>	. Один . Два . Три . Четыре
Словарный список (Variable list)	<variablelist> <varlistentry> <term>amd64</term> <listitem> <para>Это наиболее распространённая десктопная...</para> </listitem> </varlistentry> </variablelist>	amd64:: Это наиболее распространённая десктопная...
Исходный код	<screen> &prompt.root; <userinput>mkdir -p /var/spool/lpd/lp</userinput> </screen>	[source,shell] ---- # mkdir -p /var/spool/lpd/lp ----
Неформатируемый блок	<programlisting> include GENERIC ident MYKERNEL options IPFIREWALL options DUMMYNET options IPFIREWALL_DEFAULT_TO_ACCEPT options IPDIVERT </programlisting>	.... include GENERIC ident MYKERNEL options IPFIREWALL options DUMMYNET options IPFIREWALL_DEFAULT_TO_ACCEPT options IPDIVERT ....

Языковая функция	Docbook	AsciiDoc
Изображения	<figure xml:id="bsdinstall-newboot-loader-menu"> <title>Меню загрузчика FreeBSD</title> <mediaobject> <imageobject> <imagedata fileref="bsdinstall/bsdinstall-newboot-loader-menu"/> </imageobject> <textobject> </literallayout> Поддержка ASCII-графики более не предоставляется. </literallayout> </textobject> <textobject> <phrase>Меню загрузчика FreeBSD с вариантами 1-6 для загрузки в многопользовательском режиме, однопользовательском режиме, перехода в командную строку загрузчика, перезагрузки, выбора ядра для загрузки и параметров загрузки</phrase> </textobject> </mediaobject> </figure>	[[bsdinstall-newboot-loader-menu]] .Меню загрузчика FreeBSD image::bsdinstall/bsdinstall-newboot-loader-menu[Меню загрузчика FreeBSD с вариантами 1-6: загрузка многопользовательского режима, загрузка однопользовательского режима, переход в командную строку загрузчика, перезагрузка, выбор ядра для загрузки и параметров загрузки]
Включение файла	не доступно	include::chapter.adoc[]

Языковая функция	Docbook	AsciiDoc
Таблицы	<table xml:id="partition-schemes" frame="none" rowsep="1" pgwide="1"> <title>Схемы разделов</title> <tgroup cols="2" align="left"> <thead> <row> <entry align="left"> Сокращение</entry> <entry align="left"> Описание</entry> </row> </thead> <tbody> <row> <entry>APM</entry> <entry>Карта разделов Apple, используется в PowerPC(R).</entry> </row> </tbody> </tgroup> </table>	[[partition-schemes]] .Схемы разделов [cols="1,1", frame="none", options="header"] |=== | Сокращение | Описание |APM |Apple Partition Map, используется на PowerPC(R). |===
Блоки-уведомления	<tip> <para>Это подсказка</para> </tip>	[ПОДСКАЗКА] ==== Это подсказка ====

Глава 8. Переводы

Это FAQ для людей, занимающихся переводом документации FreeBSD (FAQ, Руководство, учебные пособия, man-страницы и другие) на различные языки.

Он *очень* сильно основан на FAQ по переводу из Немецкого проекта документации FreeBSD, изначально написанном Франком Грюндером elwood@mc5sys.in-berlin.de и переведённом обратно на английский Берндом Варкеном bwarken@mayn.de.

8.1. Что означают i18n и l10n?

i18n означает интернационализацию, а l10n — локализацию. Это просто удобные сокращения.

i18n можно прочитать как «i», за которым следует 18 букв, а затем «n». Аналогично, l10n — это «l», за которым следует 10 букв, а затем «n».

8.2. Существует ли рассылка для переводчиков?

Да. Различные группы переводчиков имеют свои собственные списки рассылки. В списке проектов перевода есть дополнительная информация о списках рассылки и веб-сайтах, которые поддерживаются каждым проектом перевода. Кроме того, существует freebsd-translators@freebsd.org для общих обсуждений, связанных с переводом.

8.3. Нужны дополнительные переводчики?

Да. Чем больше людей работает над переводом, тем быстрее он будет завершен, и тем быстрее изменения в англоязычной документации отразятся в переведенных документах.

Вам не нужно быть профессиональным переводчиком, чтобы помочь.

8.4. Какие языки мне нужно знать?

В идеале вы должны хорошо знать письменный английский и, разумеется, свободно владеть языком, на который переводите.

Английский язык не является строго обязательным. Например, можно сделать венгерский перевод FAQ из испанского перевода.

8.5. Какой софт нужно знать?

Настоятельно рекомендуется поддерживать локальную копию репозитория FreeBSD Git (по крайней мере, документации). Это можно сделать, выполнив:

```
% git clone https://git.FreeBSD.org/doc.git ~/doc
```

git.FreeBSD.org — это публичный `git` сервер.



Для этого потребуется установка пакета git-lite package.

Вы должны уверенно пользоваться git. Это позволит вам видеть, какие изменения были внесены между разными версиями файлов, из которых состоит документация.

Например, чтобы посмотреть различия между ревизиями `abff932fe8` и `2191c44469` файла documentation/content/en/articles/committers-guide/_index.adoc, выполните:

```
% git diff abff932fe8 2191c44469 documentation/content/en/articles/committers-
guide/_index.adoc
```

Пожалуйста, ознакомьтесь с полным объяснением использования Git в FreeBSD в Руководстве FreeBSD.

8.6. Как узнать, кто ещё, может быть, переводит на тот же язык?

Страница переводов проекта документации содержит список известных на данный момент переводческих инициатив. Если кто-то уже работает над переводом документации на ваш язык, пожалуйста, не дублируйте их усилия. Вместо этого свяжитесь с ними, чтобы узнать, как вы можете помочь.

Если на этой странице не указан переводчик для вашего языка, отправьте сообщение в список рассылки Список рассылки Проекта Документации FreeBSD, возможно, кто-то уже планирует сделать перевод, но ещё не объявил об этом.

8.7. Никто не переводит на мой язык. Что мне делать?

Поздравляем, вы только что запустили "Проект перевода документации FreeBSD на ваш язык". Добро пожаловать на борт.

Сначала решите, есть ли у вас свободное время. Поскольку сейчас вы единственный, кто работает над вашим языком, вам придется взять на себя ответственность за рекламу своей работы и координацию добровольцев, которые, возможно, захотят вам помочь.

Напишите письмо в рассылку проекта документации, объявив о том, что вы собираетесь переводить документацию, чтобы страница переводов проекта документации могла быть актуальной.

Если в вашей стране уже есть кто-то, предоставляющий услуги зеркалирования FreeBSD, вам следует связаться с ними и спросить, можете ли вы получить место для вашего проекта на их веб-сервере, а также, возможно, электронный адрес или услуги рассылки.

Затем выберите документ и начните перевод. Лучше всего начать с чего-то относительно небольшого — например, с FAQ или одного из руководств.

8.8. Я перевел часть документации, куда мне её отправить?

Это зависит от обстоятельств. Если вы уже работаете с командой переводчиков (например, с японской или немецкой командой), то у них есть свои процедуры обработки представленной документации, которые описаны на их веб-страницах.

Если вы единственный, кто работает над конкретным языком (или вы отвечаете за проект перевода и хотите отправить свои изменения обратно в проект FreeBSD), то вам следует отправить свой перевод в проект FreeBSD (см. следующий вопрос).

8.9. Я единственный, кто работает над переводом на этот язык, как мне отправить свой перевод?

Во-первых, убедитесь, что ваш перевод правильно организован. Это означает, что он должен встраиваться в существующую структуру документации и сразу же собираться.



Hugo требует коды языков в нижнем регистре. Например, вместо `pt_BR` Hugo использует `pt-br`.

В настоящее время документация FreeBSD хранится в корневом каталоге с названием documentation/. Каталоги ниже него именуются согласно языковым кодам, на которых они написаны, как определено в ISO639 (/usr/share/misc/iso639 в версии FreeBSD новее 20 января 1999 года).

Если ваш язык может быть закодирован различными способами (например, китайский), то ниже должны находиться каталоги для каждого из предоставленных форматов кодировки.

Наконец, у вас должны быть каталоги для каждого документа.

Например, гипотетический перевод на шведский язык может выглядеть так:

```
documentation/
  content/
    sv/
      books/
        faq/
          _index.adoc
```

`sv` — это название перевода в форме lang. Обратите внимание на два Makefile, которые будут использоваться для сборки документации.

Используйте команду `git diff` для создания diff и отправьте его в систему рецензирования.

```
% git diff > sv-faq.diff
```

Вы должны использовать Bugzilla, чтобы отправить отчёт, указав, что вы предоставили документацию. Было бы очень полезно, если бы другие люди проверили ваш перевод перед отправкой, так как маловероятно, что человек, который будет его применять, свободно владеет языком.

Кто-то (скорее всего, руководитель проекта документации, в настоящее время Группа Менеджеров Дерева Документации <doceng@FreeBSD.org>) проверит ваш перевод и убедится, что он собирается. В частности, будут проверены следующие моменты:

1. Работает ли `make` в каталоге root корректно?

Если возникнут проблемы, тот, кто будет рассматривать вашу заявку, свяжется с вами для их решения.

Если проблем не возникнет, ваш перевод будет включён как можно скорее.

8.10. Можно ли включать в перевод текст, специфичный для языка или страны?

Мы бы предпочли, чтобы вы этого не делали.

Например, предположим, что вы переводите Handbook на корейский язык и хотите добавить раздел о розничных продавцах в Корее в ваш Handbook.

Нет никаких реальных причин, по которым эта информация не должна быть в английской (или немецкой, или испанской, или японской, или...) версиях также. Вполне возможно, что англоговорящий человек в Корее попытается приобрести копию FreeBSD, находясь там. Это также помогает усилить воспринимаемое присутствие FreeBSD по всему миру, что не так уж и плохо.

Если у вас есть информация, специфичная для вашей страны, пожалуйста, отправьте её в виде изменения в англоязычное Руководство (через Bugzilla), а затем переведите это изменение обратно на ваш язык в локализованной версии Руководства.

Спасибо.

8.10.1. Обращение к читателю

В английских документах читатель обращается на «you», здесь нет различия между формальным и неформальным обращением, как в некоторых языках.

Если вы переводите на язык, где такое различие существует, используйте ту форму, которая обычно применяется в другой технической документации на вашем языке. Если сомневаетесь, используйте умеренно вежливую форму.

8.10.2. Нужно ли включать какую-либо дополнительную информацию в мои переводы?

Да.

Заголовок английской версии каждого документа будет выглядеть примерно так:

```
 ---
 title: Why you should use a BSD style license for your Open Source Project
 releaseinfo: "$FreeBSD: head/en_US.ISO8859-1/articles/bsdl-gpl/article.xml 53942
2020-03-01 12:23:40Z carlavilla $"
 trademarks: ["freebsd", "intel", "general"]
 ---

 = Why you should use a BSD style license for your Open Source Project
```

Точный шаблон может меняться, но он всегда будет включать строку `$FreeBSD$` и фразу `The FreeBSD Documentation Project`. Обратите внимание, что часть `$FreeBSD$` автоматически раскрывается Git, поэтому для новых файлов она должна быть пустой (просто `$FreeBSD$`).

Документы, переведённые вами, должны включать собственную строку FreeBSD, а также изменять строку `FreeBSD Documentation Project` на `The FreeBSD_язык_Documentation Project`.

В дополнение, вы должны добавить третью строку, указывающую, на какую версию английского текста опирается перевод.

Итак, испанская версия этого файла может начинаться с:

```
 ---
 title: Soporte para segundos intercalares en FreeBSD
 releaseinfo: "$FreeBSD: head/es_ES.ISO8859-1/articles/leap-seconds/article.xml 53090
2019-06-01 17:52:59Z carlavilla $"
 ---

 = Soporte para segundos intercalares en FreeBSD
```

Глава 9. Переводы PO

9.1. Введение

Система GNU gettext предоставляет переводчикам удобный способ создания и поддержки переводов документов. Переводимые строки извлекаются из исходного документа в файл PO (Portable Object). Переводы строк добавляются с помощью отдельного редактора. Эти строки могут использоваться напрямую или собираться в полную переведённую версию исходного документа.

9.2. Быстрый старт

Предполагается, что процедура, описанная в разделе Быстрый старт, уже выполнена. Опция `TRANSLATOR` необходима и уже включена по умолчанию в порте textproc/docproj.

Этот пример демонстрирует создание испанского перевода краткой статьи Високосные секунды.

Процедура: Установка PO-редактора

1. Редактор PO необходим для редактирования файлов переводов. В этом примере используется editors/poedit.

```
# pkg install poedit
```

Процедура: Начальная настройка

Когда создаётся новый перевод, структура каталогов должна быть создана или скопирована из оригинальной английской версии:

1. Создайте каталог для нового перевода. Исходный текст статьи на английском находится в ~/doc/documentation/content/en/articles/leap-seconds/. Испанский перевод будет расположен в ~/doc/documentation/content/es/articles/leap-seconds/. Путь идентичен, за исключением названия языкового каталога. Исходный текст статьи на английском находится в ~/doc/en/articles/leap-seconds/.

```
% mkdir ~/doc/documentation/content/es/articles/leap-seconds
```

2. Скопируйте _index.po из исходного документа в каталог перевода:

```
% cp ~/doc/documentation/content/en/articles/leap-seconds/_index.po \
  ~/doc/documentation/content/es/articles/leap-seconds/
```

Предположим, что документ или язык также переводится через Weblate. В таком случае

рекомендуется взять файл `.po` оттуда и загрузить переведённый документ обратно в платформу, чтобы централизовать усилия по переводу через Weblate и избежать повторной работы.

См. как загрузить файлы `.po` в главе Перевод офлайн в Weblate.

Процедура: Перевод

Используйте PO-редактор для ввода переводов в PO-файл. Доступно несколько различных редакторов. Здесь показан poedit из пакета:editors/poedit[].

```
% poedit documentation/content/es/articles/leap-seconds/_index.po
```

Процедура: Создание переведенного документа

1. Сгенерируйте переведенный документ:

   ```
   % cd ~/doc
   % ./tools/translate.sh documentation es articles/leap-seconds
   ```

   Имя сгенерированного документа соответствует имени оригинала на английском языке, обычно `_index.adoc`.

2. Проверьте сгенерированный файл, преобразовав его в HTML и просмотрев в веб-браузере:

   ```
   % cd ~/doc/documentation
   % make
   ```

9.3. Создание новых переводов

Первым шагом в создании нового переведённого документа является поиск или создание каталога для его размещения. FreeBSD размещает переведённые документы в подкаталоге с названием, соответствующим языку и региону, в формате lang. Здесь *lang* — это двухбуквенный код в нижнем регистре.

Таблица 2. Названия языков

Язык	Регион	Имя каталога с переводами
Английский	Соединённые Штаты	en
Бенгальский	Бангладеш	bn-bd
Датский	Дания	da

Язык	Регион	Имя каталога с переводами
Немецкий	Германия	de
Греческий	Греция	el
Испанский	Испания	es
Французский	Франция	fr
Венгерский	Венгрия	hu
Итальянский	Италия	it
Японский	Япония	ja
Корейский	Корея	ko
Монгольский	Монголия	mn
Голландский	Нидерланды	nl
Польский	Польша	pl
Португальский	Бразилия	pt-br
Русский	Россия	ru
Турецкий	Турция	tr
Китайский	Китай	zh-cn
Китайский	Тайвань	zh-tw

Переводы находятся в подкаталогах основного каталога документации, который в данном случае предполагается как ~/doc/documentation/, как показано в разделе Быстрый старт. Например, немецкие переводы расположены в ~/doc/documentation/content/de/, а французские — в ~/doc/documentation/content/fr/.

Каждый языковой каталог содержит отдельные подкаталоги с названиями по типу документов, обычно articles/ и books/.

Объединение этих имён каталогов даёт полный путь к статье или книге. Например, французский перевод статьи NanoBSD находится в ~/doc/documentation/content/fr/articles/nanobsd/, а монгольский перевод Руководства — в ~/doc/documentation/content/mn/books/handbook/.

При переводе документа на новый язык необходимо создать новый каталог языка. Если каталог языка уже существует, достаточно создать подкаталог в каталоге articles/ или books/.

Пример 12. Создание испанского перевода Руководства портировщика

Создайте новый перевод на испанский язык Руководства портировщика. Оригинал находится в книге ~/doc/documentation/content/en/books/porters-handbook/.

1. Каталог для книг на испанском языке ~/doc/documentation/content/es/books/ уже существует, поэтому требуется только создать подкаталог для Руководства портировщика:

```
% cd ~/doc/documentation/content/es/books
% mkdir porters-handbook
```

2. Скопируйте содержимое из переводимой книги:

```
% cd porters-handbook
% cp -R ~/doc/documentation/content/en/books/porters-handbook/* .
```

Теперь структура документа готова для начала перевода с помощью команды po4a.

9.4. Перевод

Система gettext значительно сокращает количество элементов, за которыми нужно следить переводчику. Строки, подлежащие переводу, извлекаются из исходного документа в файл PO. Затем с помощью редактора PO вводятся переведённые версии каждой строки.

Система перевода FreeBSD PO не перезаписывает PO-файлы, поэтому этап извлечения можно выполнять в любое время для обновления PO-файла.

Редактор PO используется для редактирования файла. В этих примерах показан editors/poedit, так как он прост и имеет минимальные требования. Другие редакторы PO предоставляют функции, облегчающие процесс перевода. В Коллекции портов доступно несколько таких редакторов, включая devel/gtranslator.

Важно сохранить PO-файл. Он содержит всю работу, проделанную переводчиками.



Перевод документов онлайн оказывается наиболее простым способом перевода документации в FreeBSD, так как позволяет нескольким пользователям совместно работать над одним файлом, эффективно распределяя нагрузку. Для получения дополнительной информации обратитесь к следующей главе: Переводы в Weblate.

9.5. Советы переводчикам

9.5.1. Сохранение AsciiDoc макросов

Сохраните макросы AsciiDoc, которые указаны на английском языке.

Пример 13. Сохранение AsciiDoc макросов

Оригинал на английском:

```
msgid ""
```

```
"This example shows the creation of a Spanish translation of the short "
"extref:{leap-seconds}[Leap Seconds] article."
```

Испанский перевод:

```
msgid ""
"Este ejemplo muestra la creación de un artículo con poco contenido como el
artículo "
"extref:{leap-seconds}[Leap Seconds]."
```

9.5.2. Сохранение пробелов

Сохраняйте существующие пробелы в начале и конце строк, которые нужно перевести. В переведённой версии эти пробелы также должны присутствовать.

9.5.3. Дословные теги

Содержимое некоторых тегов должно быть скопировано дословно, без перевода:

- `man(1)`
- `package`
- `link`
- `image`
- `include`
- `Admonitions`
- `id's`
- `Heading tags`
- `source`

9.6. Сборка переведенного документа

Переведённую версию оригинального документа можно создать в любое время. При этом все непереведённые части документа будут включены в итоговый документ на английском языке. Большинство PO-редакторов имеют индикатор, показывающий, какая часть перевода завершена. Это позволяет переводчику легко увидеть, когда переведено достаточно строк, чтобы сборка итогового документа имела смысл.

Глава о Weblate есть полный пример того, как Собрать переведённый документ.

9.7. Отправка новой переведённой версии

Подготовьте новые файлы перевода для отправки. Это включает добавление файлов в систему контроля версий, установку дополнительных свойств для них, а затем создание

diff-файла для отправки.

Файлы различий, созданные в этих примерах, можно прикрепить к отчёту об ошибке в документации или обзору кода.

Пример 14. Перевод статьи NanoBSD на испанский язык

1. Создайте diff новых файлов из базового каталога ~/doc/ так, чтобы полный путь отображался вместе с именами файлов. Это помогает коммиттерам определить целевой языковой каталог.

```
% cd ~/doc
% git diff documentation/content/es/articles/nanobsd/ >
/tmp/es_nanobsd.diff
```

Глава о Weblate содержит полный пример того, как отправить новый перевод.

Глава 10. Переводы в Weblate

10.1. Введение

Эта глава описывает основные шаги для присоединения к команде переводчиков FreeBSD, перевода онлайн в Weblate или офлайн, а также содержит простые рекомендации по переводу, вычитке и тестированию. Основное внимание уделено части, связанной с переводом.

Исходные документы (статьи и книги) находятся на портале документации.

Weblate — это веб-ориентированное программное обеспечение с открытым исходным кодом, специализирующееся на локализации. Проект FreeBSD поддерживает собственный экземпляр.

10.2. Как стать переводчиком FreeBSD

Вот простые шаги для начала перевода статей и книг проекта документации FreeBSD.

1. Создайте учётную запись на FreeBSD Weblate с помощью электронной почты или вашего аккаунта GitHub.
2. Подписаться на список рассылки Список рассылки переводчиков FreeBSD.
3. Представьтесь и попросите присоединиться к языковой команде. Если языковая команда не существует, попросите создать её. Самопредставление обязательно. Оно увеличивает шансы на одобрение доступа на запись.
4. Войдите в Weblate с новым аккаунтом.
5. Найдите языковую команду и выберите первый документ для перевода.
6. Создайте учётную запись в Bugzilla, чтобы отправлять переводы после завершения работы над документом. Проект документации также принимает Pull Requests на GitHub с переводами.



Все файлы переводов и документы должны соответствовать Лицензии документации FreeBSD; если это неприемлемо, пожалуйста, не регистрируйтесь и не присылайте никаких исправлений или переводов.

10.3. Представьтесь

Предоставьте краткое самопредставление в рассылке Список рассылки переводчиков FreeBSD, чтобы начать процесс предоставления доступа. Это позволит координатору языка или администратору выдать необходимые права новому пользователю Weblate для начала перевода.

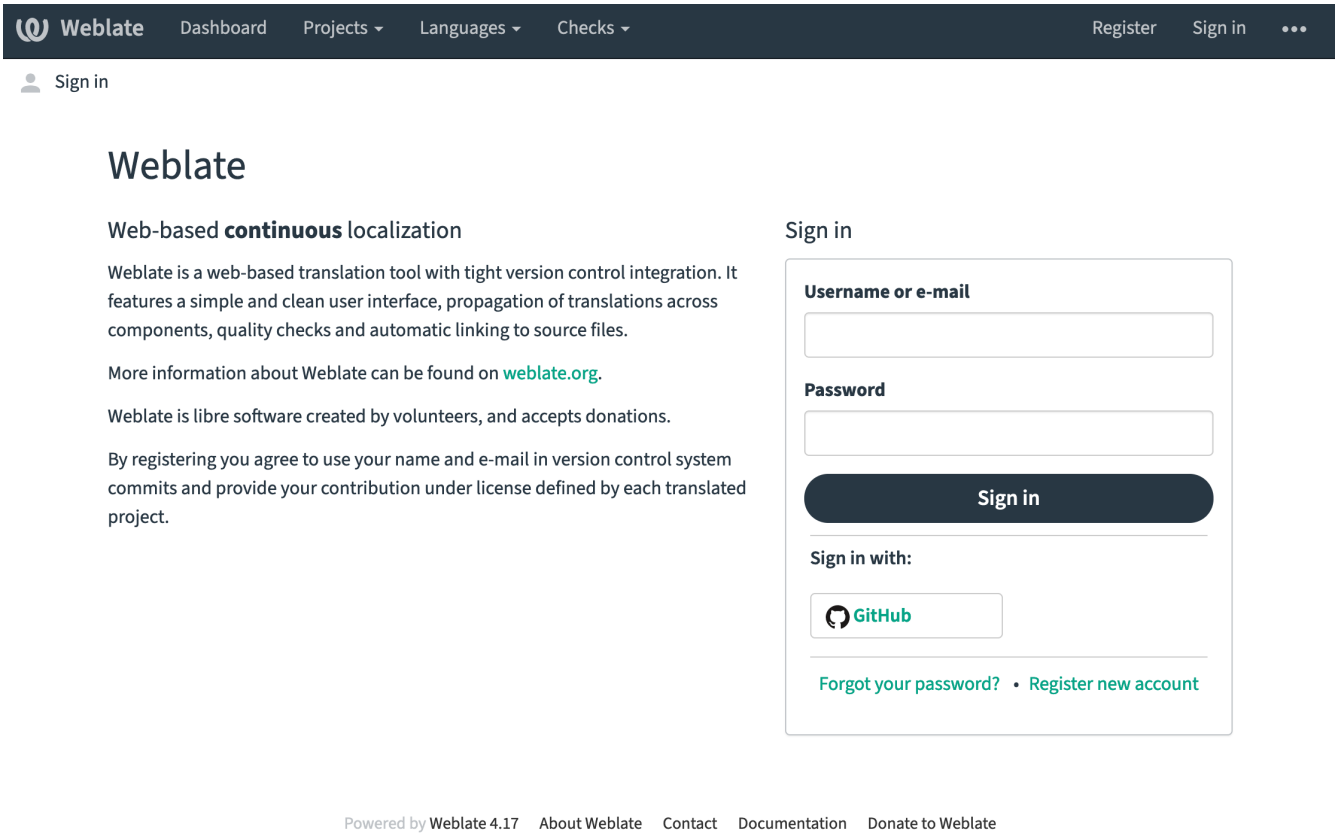
Ниже приведён пример того, как может выглядеть такое письмо.

```
Subject: Self-Introduction: Name and language

Name:      Name (use preferred name)
Location:  City, country (optional)
Login:     username or email (essential)
Language:  Language to translate (essential)
Profession or student status: (optional)
About You: (free format -- info which you feel comfortable sharing with
  others: company, school, other affiliation, historical qualifications, other
  projects you have worked on, level and type of computer skills, other relevant
skills,
  etc.)
You and the FreeBSD Project: (free format: other FreeBSD projects of
  interest, comments, etc.)
```

10.4. Вход в Weblate

Откройте https://translate-dev.freebsd.org/ и Войдите (Sign in).



Используйте имя пользователя, адрес электронной почты или учётную запись GitHub для входа.

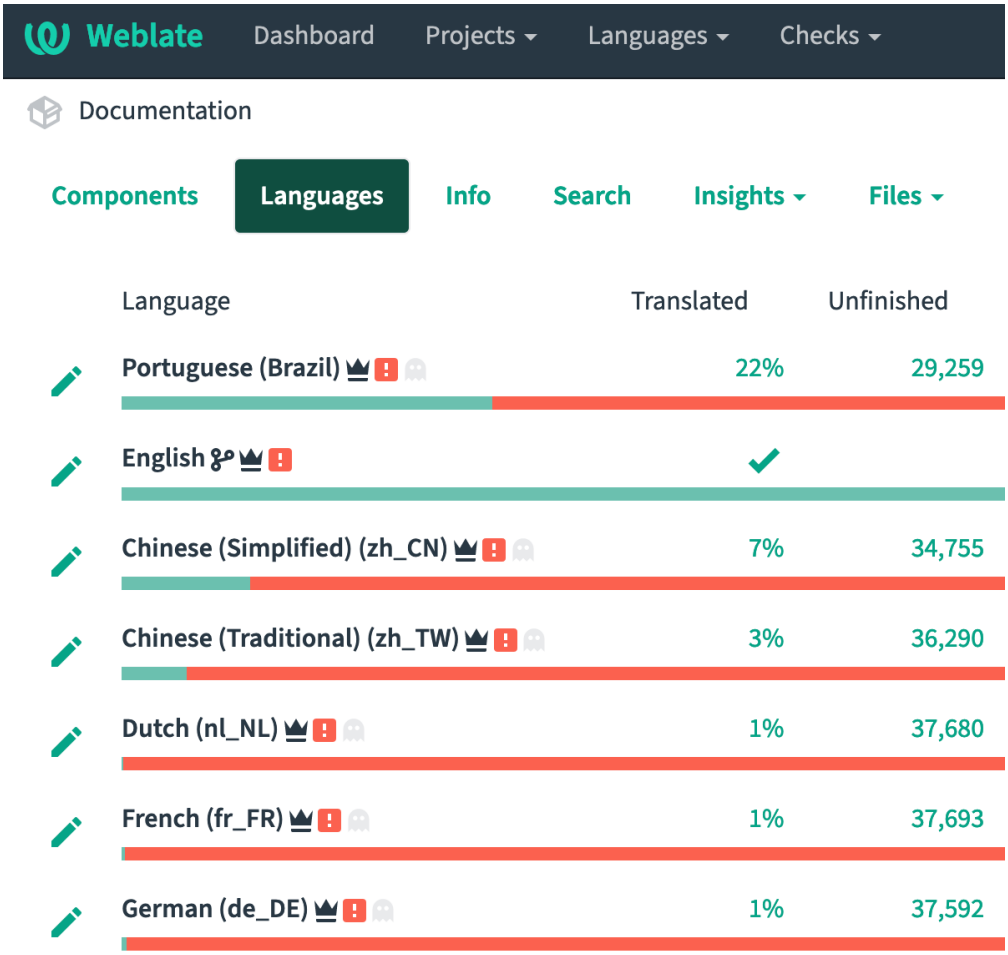
Профиль пользователя содержит ваши предпочтения, имя и адрес электронной почты. Имя и адрес будут использоваться в коммитах; сохраняйте эту информацию актуальной.

На экземпляре Weblate FreeBSD все переводы будут зафиксированы в freebsd-doc-translate (промежуточном репозитории на GitHub), а не напрямую в freebsd-doc. Переводчики должны брать файлы PO gettext (`.po`), преобразовывать их в `.adoc` и отправлять через Bugzilla или GitHub, чтобы переведённый документ был опубликован или обновлён в портале документации. Подробнее см. в следующих разделах.

Weblate будет фиксировать изменения ежедневно, как минимум в `freebsd-doc-translate`, если есть новые переведённые строки.

10.5. Найти команду локализации для участия

Нажмите Проекты, выберите Документация, затем нажмите Языки и увидите все доступные языки.



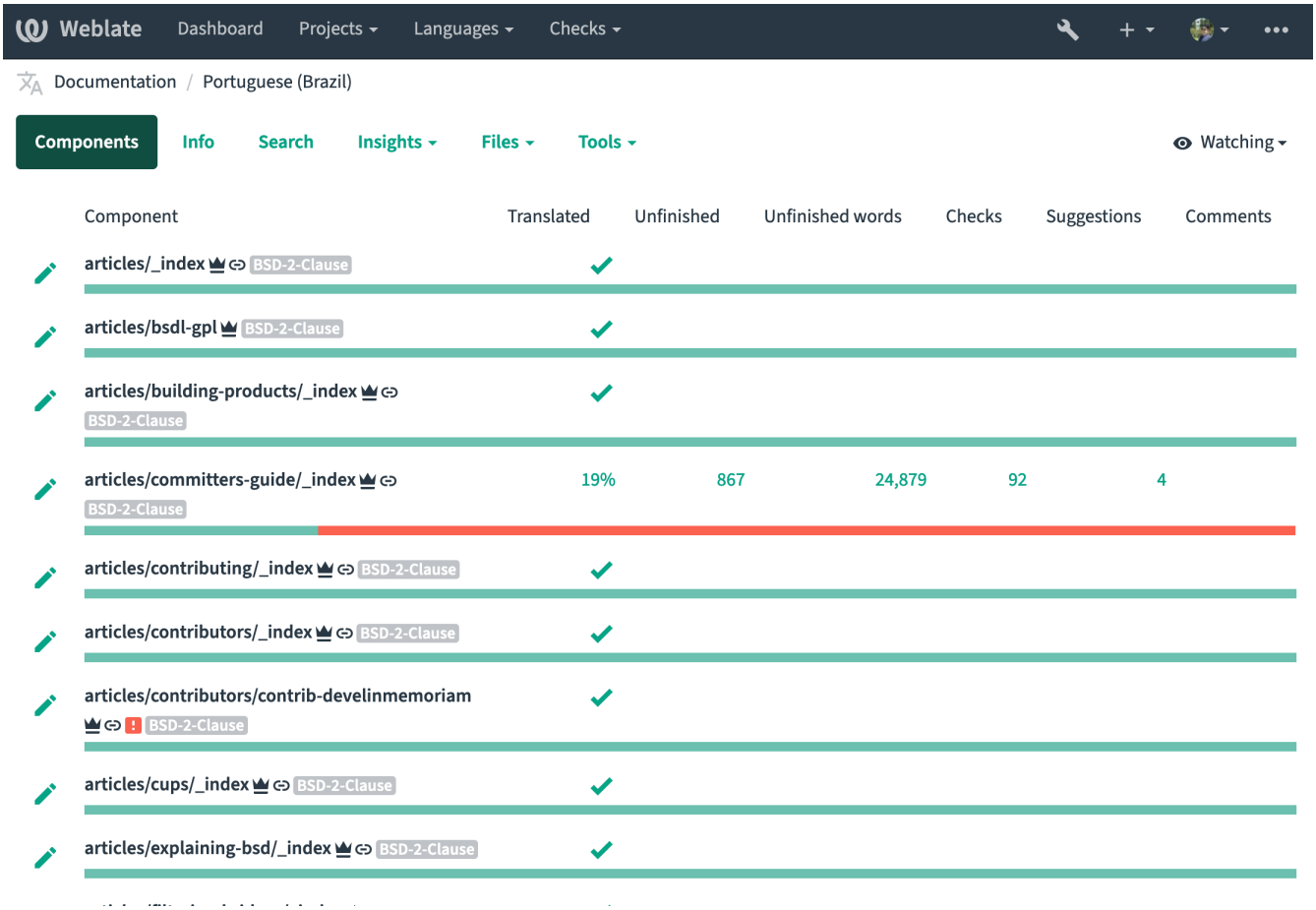
Обратите внимание, что некоторые языки и переведённые документы уже доступны в портале документации и репозиториях.

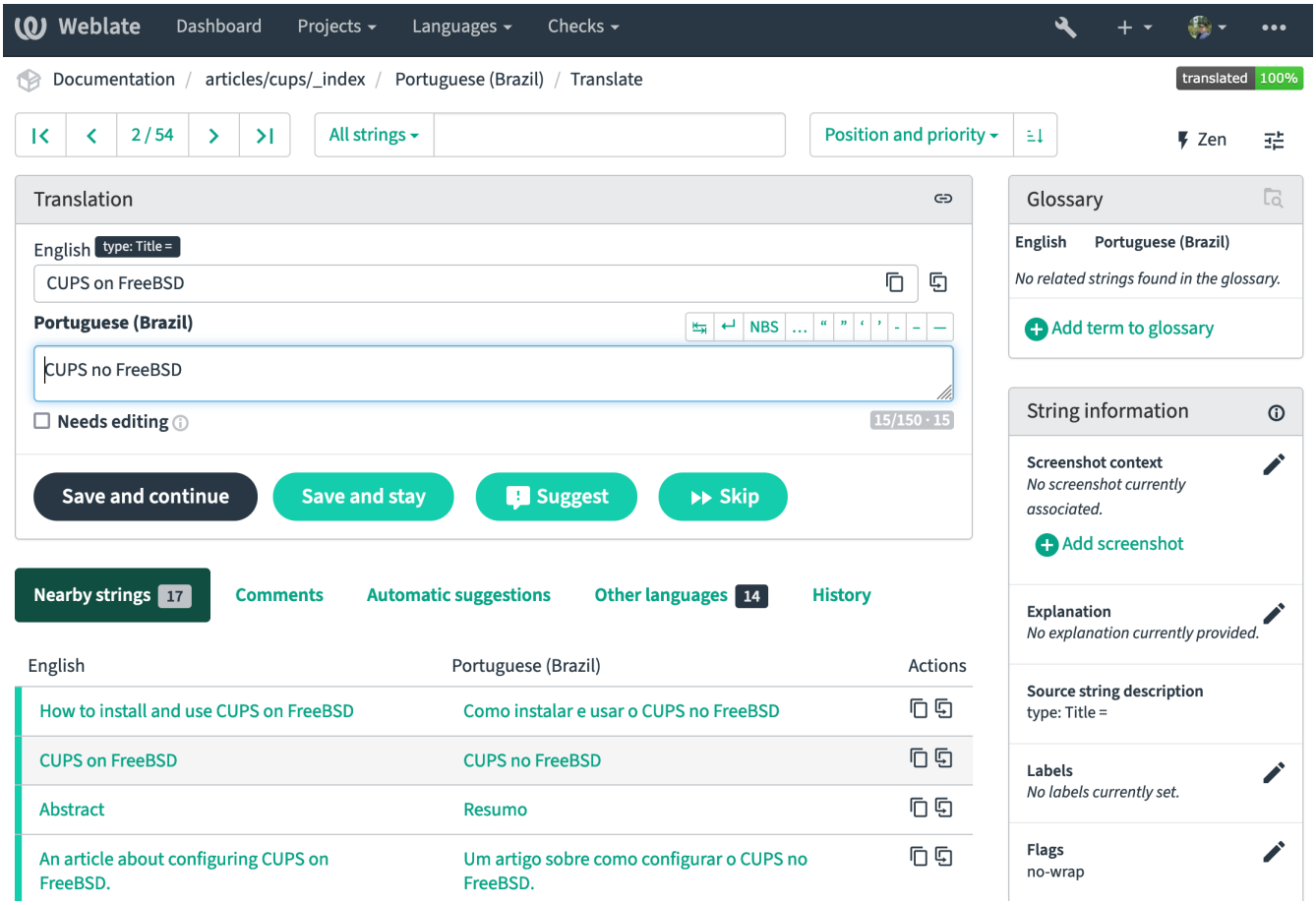
Если желаемый язык для перевода недоступен в Weblate, пожалуйста, свяжитесь с координаторами по языкам, прежде чем запрашивать создание нового языка. Если ответа не последует, обратитесь по адресу: Группа Менеджеров Дерева Документации <doceng@FreeBSD.org>.

10.6. Перевод онлайн на Weblate

Перевод документов онлайн оказывается наиболее простым способом перевода документации в FreeBSD, так как позволяет пользователям совместно работать над одним файлом, распределяя нагрузку.

После того как координатор или администратор предоставит пользователю доступ к определённому языку, кнопка сохранения станет активной, и пользователь сможет начать перевод.



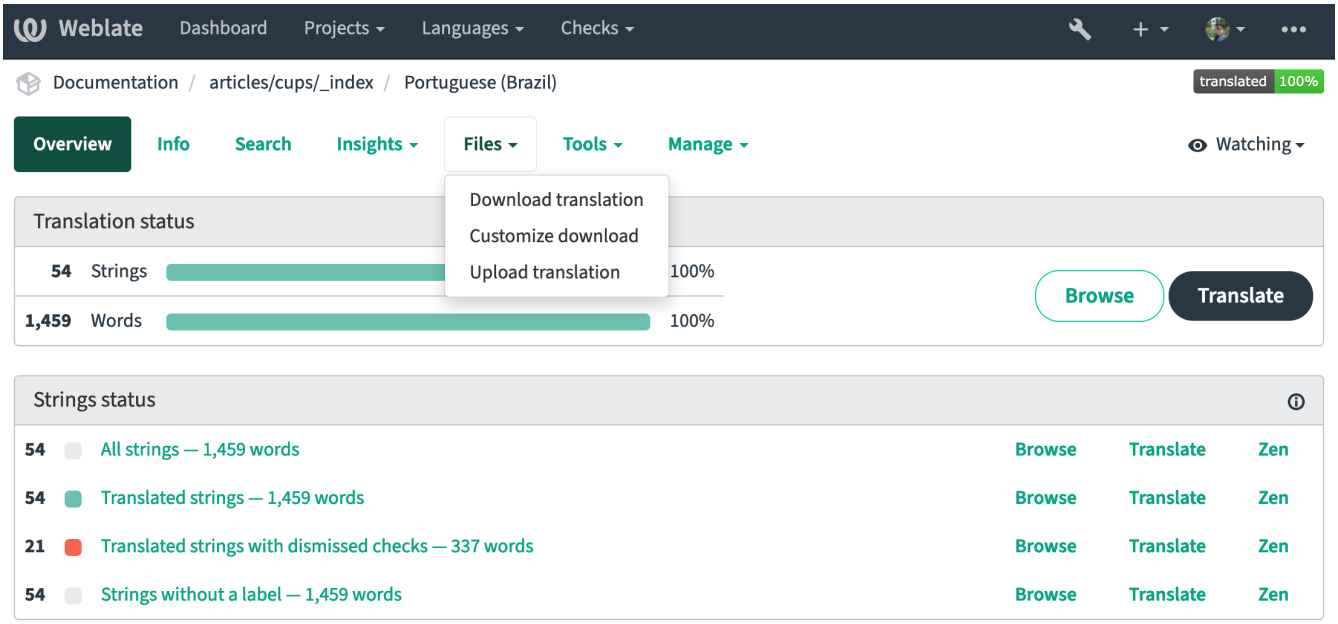


В Weblate есть набор ссылок, ведущих к непосредственному переводу. Перевод далее разделён на отдельные проверки, такие как `Не переведено` или `Требует проверки`. Если весь документ переведён без ошибок, ссылка `Все переводы` всё равно доступна на случай, если потребуется проверка. Также можно воспользоваться полем поиска, чтобы найти конкретную строку или термин.

В документации Weblate содержится дополнительная информация о переводах, например, сочетания клавиш и другие советы по работе с инструментом перевода.

10.7. Перевод в автономном режиме

Weblate на FreeBSD использует файлы перевода PO gettext. Пользователи, знакомые с файлами PO gettext, которые хотят переводить офлайн, могут загружать и выгружать переводы на странице документа в Weblate, выбрав пункт в меню `Files`.



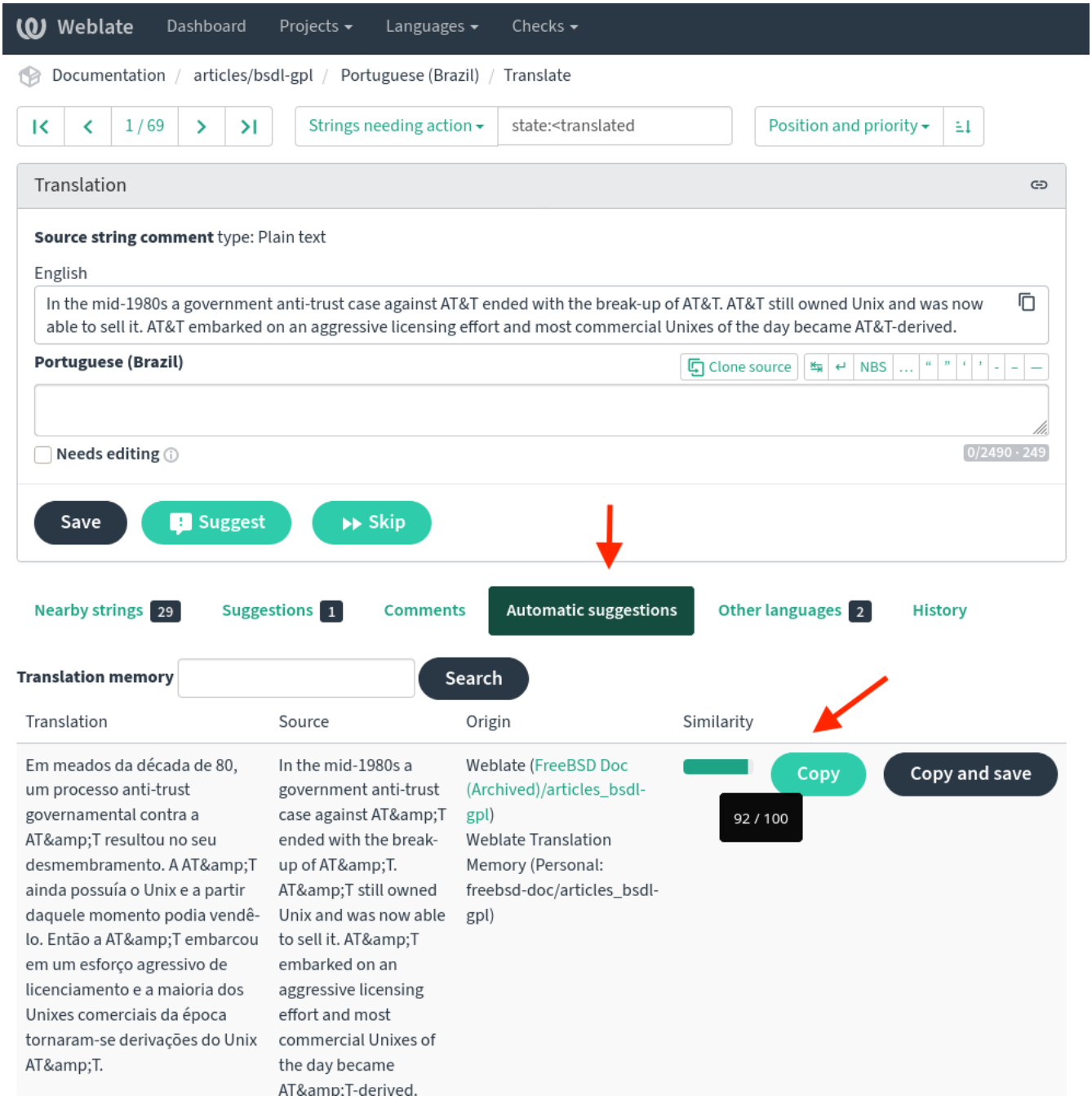
10.8. Перевод на основе автоматических предложений

Языки, использующие Weblate до миграции на Hugo/Asciidoctor, могут использовать эту функцию в Weblate для экономии времени.

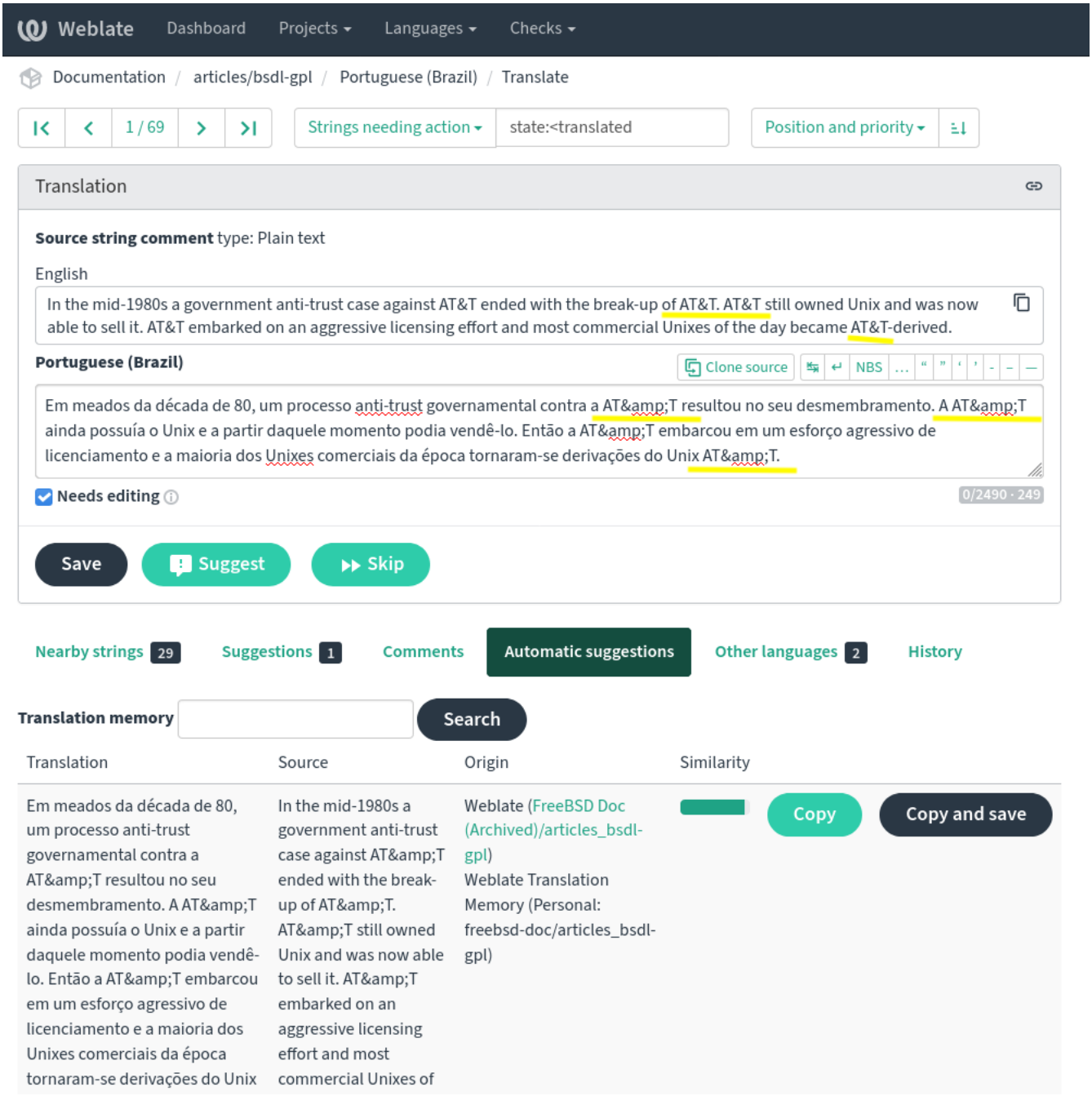
Эта функция Weblate использует память переводов, созданную другими компонентами и проектами на том же сервере. Для этого прежние переводы Weblate сохранены на том же сервере в режиме только для чтения.

Строки, совпадающие с `100/100` по степени схожести, могут быть скопированы и сохранены напрямую. Остальным строкам потребуется как минимум незначительная корректировка.

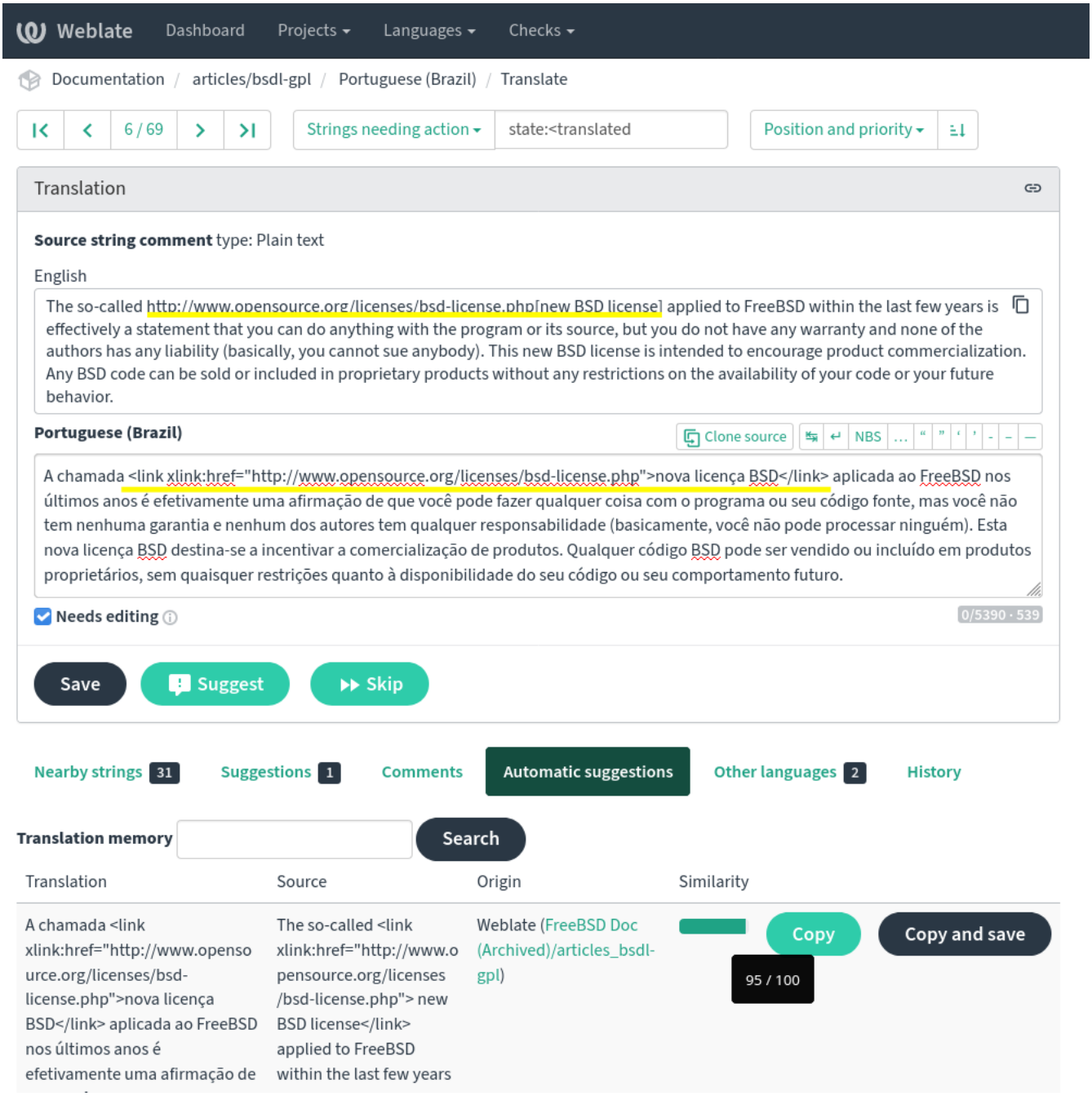
Некоторые примеры:



С переходом на Hugo/Asciidoctor документы используют UTF-8. Некоторые HTML-сущности следует заменить. Некоторые строки, такие как ссылки, требуют изменений в разметке.

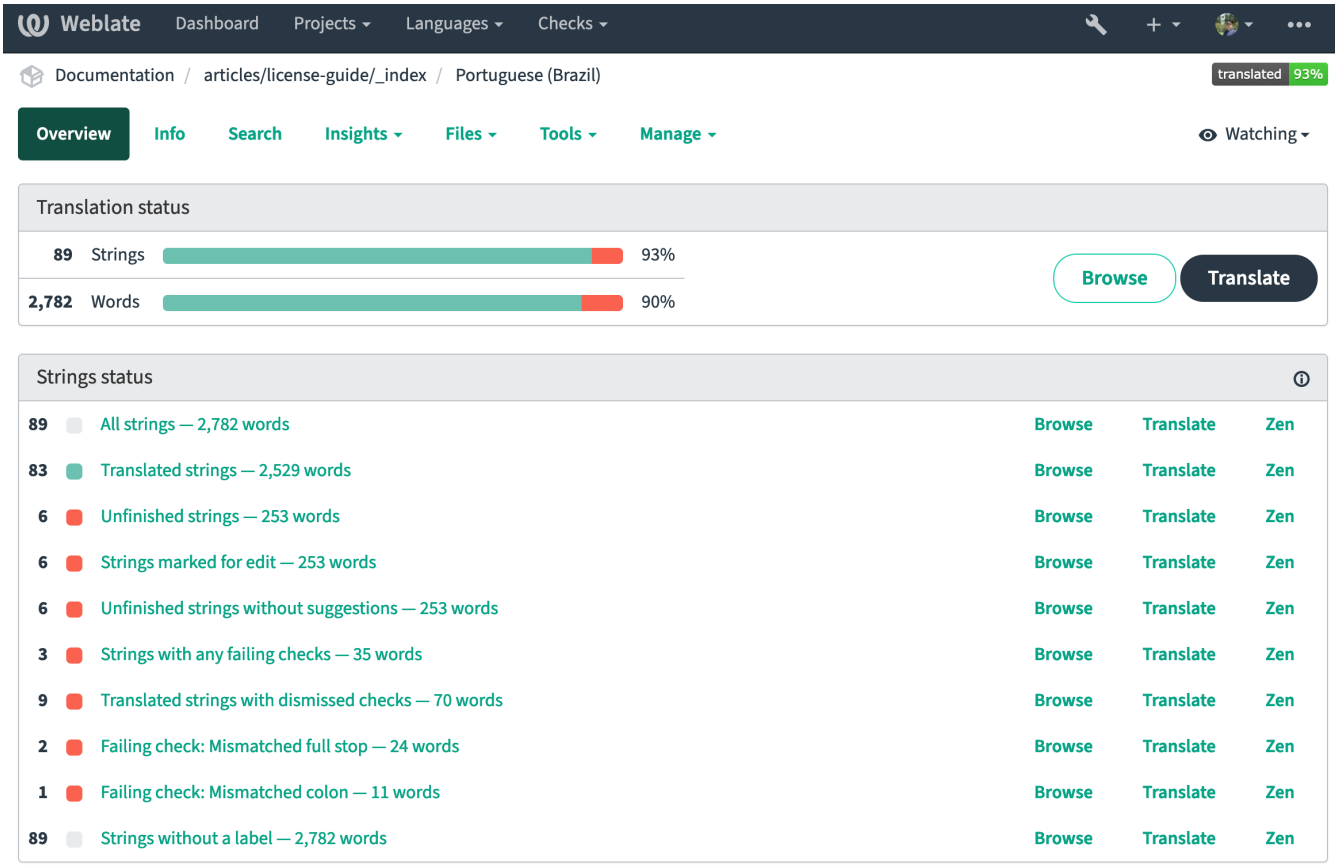

Weblate
Dashboard
Projects
Languages
Checks
Documentation / articles/bsdl-gpl / Portuguese (Brazil) / Translate
1 / 69
Strings needing action
state:<translated
Position and priority
Translation
Source string comment type: Plain text
English
In the mid-1980s a government anti-trust case against AT&T ended with the break-up of AT&T. AT&T still owned Unix and was now able to sell it. AT&T embarked on an aggressive licensing effort and most commercial Unixes of the day became AT&T-derived.
Portuguese (Brazil)
Clone source
NBS
Em meados da década de 80, um processo anti-trust governamental contra a AT&T resultou no seu desmembramento. A AT&T ainda possuía o Unix e a partir daquele momento podia vendê-lo. Então a AT&T embarcou em um esforço agressivo de licenciamento e a maioria dos Unixes comerciais da época tornaram-se derivações do Unix AT&T.
Needs editing
0/2490 · 249
Save
Suggest
Skip
Nearby strings 29
Suggestions 1
Comments
Automatic suggestions
Other languages 2
History
Translation memory
Search
Translation
Source
Origin
Similarity
Em meados da década de 80, um processo anti-trust governamental contra a AT&T resultou no seu desmembramento. A AT&T ainda possuía o Unix e a partir daquele momento podia vendê-lo. Então a AT&T embarcou em um esforço agressivo de licenciamento e a maioria dos Unixes comerciais da época tornaram-se derivações do Unix
In the mid-1980s a government anti-trust case against AT&T ended with the break-up of AT&T. AT&T still owned Unix and was now able to sell it. AT&T embarked on an aggressive licensing effort and most commercial Unixes of
Weblate (FreeBSD Doc (Archived)/articles_bsdl-gpl)
Weblate Translation Memory (Personal: freebsd-doc/articles_bsdl-gpl)
Copy
Copy and save

Ссылки:

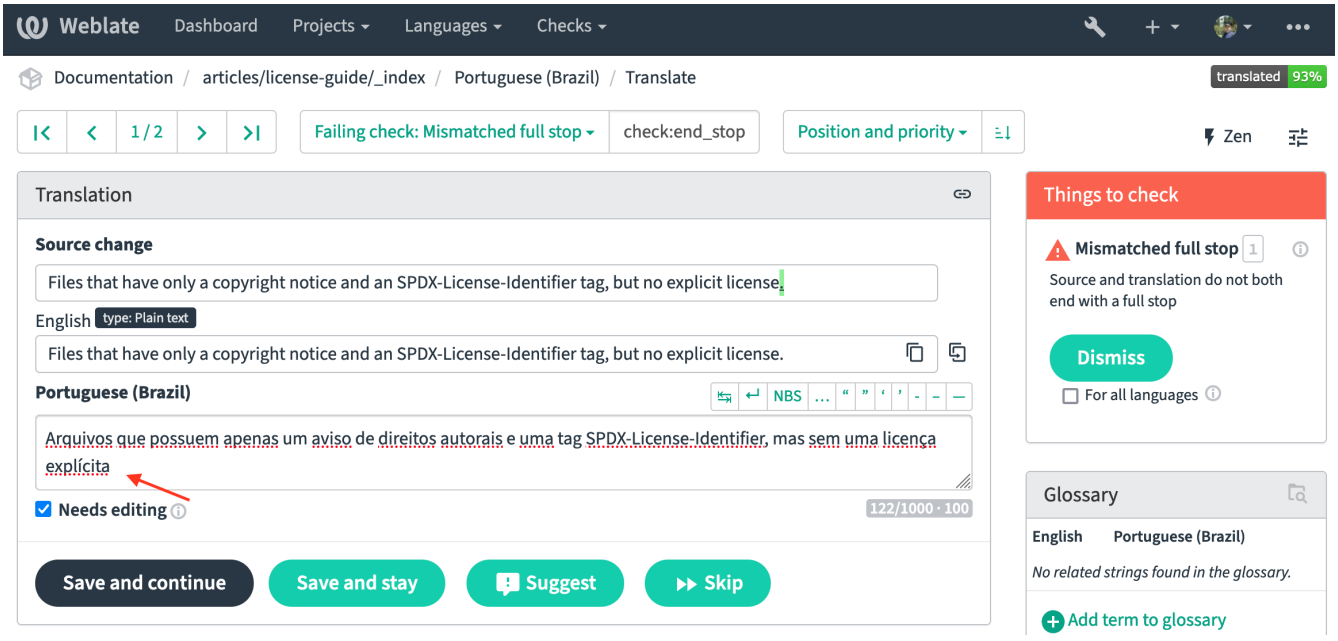


10.9. Вычитка и проверка качества в Weblate

Панель документа `Project/Language/Document` отображает статус перевода и состояние строк для этого документа. Эта страница удобна для вычитки и проверки качества.



В этом примере две строки пропустили точку; переход по этой ссылке покажет только те строки, которые нужно пересмотреть/перевести.



Переводчики и рецензенты часто ценят возможность видеть переведённые строки в контексте.

10.10. Сборка переведенного документа

Проект не использует непрерывную интеграцию и непрерывную поставку для сборки переводов. Ведутся исследования по внедрению этой возможности.



Следующий пример использует GitHub, так как Weblate также находится на GitHub . Обратите внимание, что этот репозиторий доступен только для чтения, но Pull Requests принимаются.

Для локальной сборки перевода выполните следующие шаги:

Процедура: Клонирование необходимых репозиториев

1. Клонирование репозитория `freebsd-doc`:

```
% git clone https://github.com/freebsd/freebsd-doc.git ~/freebsd-doc
```

2. Клонирование репозитория `freebsd-doc-translate`:

```
% git clone https://github.com/freebsd/freebsd-doc-translate.git ~/freebsd-doc-
translate
```

Процедура: Копирование файла перевода в `freebsd-doc`

Имея оба репозитория, скопируйте перевод из `freebsd-doc-translate` в `freebsd-doc`. Пример перевода статьи Руководства для коммиттеров на испанском языке.

```
% cp ~/freebsd-doc-translate/documentation/content/es/articles/committers-
guide/_index.po \
~/freebsd-doc/documentation/content/es/articles/committers-guide/
```

Процедура: Преобразование файла перевода (`.po`) в `.adoc`

Перейдите в корень `freebsd-doc`.

```
% cd ~/freebsd-doc
```

Преобразовать файл `.po` в `.adoc`

```
% ./tools/translate.sh documentation es articles/committers-guide
```

По умолчанию: только файлы с более чем восемьюдесятью процентами переведённых строк будут преобразованы в `.adoc`.

Чтобы игнорировать это ограничение:

```
% KEEP_ENV=0 ./tools/translate.sh documentation es articles/committers-guide
```

Некоторые документы, такие как книги, содержат множество PO-файлов gettext. Всегда копируйте их все при переводе и сборке. Файлы, которые не были переведены, будут преобразованы с исходными (английскими) строками.



Структура каталогов является основополагающей. Всегда следуйте структуре каталогов английского документа.

Процедура: Сборка переведенного документа

Наконец, часть сборки.

Перейдите в каталог документации, так как сборка веб-сайта FreeBSD не требуется.

```
% cd documentation
```

И соберите документацию. Обратите внимание, что `en` всегда добавляется по умолчанию при сборке любого другого языка.

```
% DOC_LANG=es make
```

Эта команда соберет только английскую и испанскую документацию портала FreeBSD. Результат будет сохранен в каталоге public; откройте его в браузере. Обратите внимание, что некоторые индексные файлы могут перенаправлять браузер на онлайн-страницу.

Еще один хороший вариант — собрать и предоставить контент с помощью встроенного веб-сервера Hugo:

```
% DOC_LANG=es make run
```

По умолчанию веб-сервер прослушивает `localhost`; чтобы изменить это поведение, укажите нужный IP-адрес в значении параметра `BIND`.

```
% DOC_LANG=es make run BIND=192.168.15.10
```

Это собирает и предоставляет содержимое с помощью встроенного веб-сервера Hugo, оставляя его открытым, и при любых изменениях файлов автоматически пересобирает их.

Чтобы внести необходимые изменения в перевод, выполните следующие шаги для

повторной синхронизации всех компонентов:

- Исправьте строку перевода на Weblate.
- Заставьте Weblate зафиксировать изменения в разделе `Document/Manage/Commit`.
- Синхронизируйте локальный репозиторий Weblate `freebsd-doc-translate` с помощью команды `git pull origin main`.
- Скопируйте перевод снова в `freebsd-doc`.
- Преобразуйте перевод в .adoc с помощью скрипта `./tools/translate.sh`.
- Hugo пересоберет файл и не будет собирать весь набор, если использовалась команда `make run`, или повторно выполнит `make`.



Повторяйте предыдущие шаги столько раз, сколько необходимо, пока документ не будет готов к публикации.

Глава Процесс сборки документации содержит информацию о преобразовании в HTML и PDF.

10.11. Отправка переводов

Пример отправки обновления для статьи на бразильском португальском Committer's Guide.

Проверка репозитория

После выполнения шагов из раздела Сборка переведенного документа, перейдите в корневой каталог `freebsd-doc` и просмотрите, что будет включено в коммит. Для просмотра списка изменяемых файлов и различий в их содержимом:

```
% git status
% git diff
```

Просмотрите вывод, и если какие-либо файлы, не связанные с обновлением перевода Руководства коммиттера, были изменены или добавлены, выполните соответствующее действие по их откату или удалению соответственно, прежде чем продолжить.

Всегда включайте файл PO gettext (`.po`) и переведенный документ в Hugo/Asciidoctor (`.adoc`).

Создать новую ветку и зафиксировать изменения

Создайте ещё одну ветку для разделения работы, что поможет при будущих обновлениях в локальном репозитории.

```
% git checkout -b committers-guide_pt-br
```

Зарегистрировать локальный коммит.

```
% git add .
% git commit
```

Пример сообщений коммитов для переводов:

```
pt-br/committers-guide: Sync with en XXXXXXX
```

Где `XXXXXXX` — это ревизия git(1), хранящаяся в репозитории Weblate в файле ~/freebsd-doc-translate/revision.txt.

Если это первый перевод статьи:

```
Add Korean translation of Leap Seconds article
```

После выполнения коммита будет отображено сообщение, если git(1) ранее не был настроен. Следуйте инструкциям и укажите имя и адрес электронной почты, используемые в Weblate. Этот шаг важен для правильного учёта вклада участников.

Затем проверьте весь коммит, просмотрите изменения, а также имя автора и адрес электронной почты.

```
% git show
```

Сгенерировать патч

Далее создайте файл git-format-patch(1).

```
% git format-patch main
0001-pt-br-committers-guide-Sync-with-en-XXXXXXX.patch
```

Прикрепите патч 0001-pt-br-committers-guide-Sync-with-en-XXXXXXX.patch к отчёту об ошибке в FreeBSD Bugzilla.

Включите следующую информацию в отчёт:

Таблица 3. Поля Bugzilla

Поле	**Значение**
продукт (product)	Documentation
Компонент (Component)	Books & Articles

Поле	Значение
Сводка (Summary)	Тот же текст, что и в локальном коммите
Описание (Description)	Укажите, что инструкции в этом руководстве были выполнены, включая вычитку и другие необходимые шаги. Включите сведения, которые могут помочь с анализом и продвижением отчёта.
CC (Необязательно)	Если в языке есть координаторы, укажите их адреса электронной почты в поле CC.

Для тех, кто знаком с git(1) и GitHub: вместо отправки исправления через Bugzilla, можно использовать запрос на включение изменений (pull request) в GitHub (укажите имя и адрес, которые вы используете в Weblate).

https://github.com/freebsd/freebsd-doc/ является вторичным зеркалом. Изменения в дереве `doc` могут вносить только люди, имеющие права на коммит (`doc commit bit`).

Когда переводчики продолжают присылать качественные исправления, другие коммиттеры могут выдвинуть их для получения прав на запись (бит коммита для документации для переводов), учётной записи FreeBSD и сопутствующих привилегий.

Список дополнительных участников FreeBSD включает некоммиттеров, чьи изменения были закоммичены в дерево `doc`.

Если вы сомневаетесь в каком-либо действии, напишите в Список рассылки переводчиков FreeBSD.

10.12. FAQ (Часто задаваемые Вопросы)

10.12.1. Нужно ли переводить все сообщения об авторских правах?

Каждая языковая команда решает этот вопрос для своего языка; в команде `pt-br` (бразильский португальский) было решено не переводить эти сообщения.

Глава 11. Справочник

11.1. Введение

Справочник, часто сокращаемый до *man-страниц*, изначально задумывались как легкодоступные напоминания о синтаксисе команд, деталях драйверов устройств или форматах конфигурационных файлов. Со временем они стали крайне полезным справочным материалом для быстрого ознакомления прямо из командной строки для пользователей, системных администраторов и программистов.

Хотя они предназначены скорее как справочные материалы, а не учебные руководства, разделы EXAMPLES в manual pages часто содержат подробные примеры использования.

Страницы Справочника обычно отображаются интерактивно с помощью команды man(1). Когда пользователь вводит `man ls`, выполняется поиск страницы Справочника, соответствующей `ls`. Первый найденный результат отображается.

11.2. Разделы

Страницы справочника (man-страницы) сгруппированы по *разделам*. Каждый раздел содержит man-страницы для определённой категории документации:

Номер раздела	Категория
1	Общие команды
2	Системные вызовы
3	Функции библиотек
4	Интерфейсы ядра
5	Форматы файлов
6	Игры
7	Разное
8	Системный администратор
9	Разработчик ядра

11.3. Язык разметки

Для man-страниц использовались различные формы разметки и программы отображения. FreeBSD использовала groff(7) и более новую mandoc(1). Большинство существующих man-страниц FreeBSD, а также все новые, используют разметку в формате mdoc(7). Это простая построчная разметка, обладающая достаточной выразительностью. В основном она семантическая: части текста помечаются по их назначению, а не по тому, как они должны выглядеть при отображении. Существует некоторая разметка, основанная на внешнем виде, которую обычно лучше избегать.

Исходный код страницы справочника обычно интерпретируется и отображается на экране в интерактивном режиме. Исходные файлы могут быть обычными текстовыми файлами или сжатыми с помощью gzip(1) для экономии места.

Страницы справочника также могут быть преобразованы в другие форматы, включая PostScript для печати или генерации PDF. См. man(1).

11.3.1. Разделы Справочника

Страницы справочника состоят из нескольких стандартных разделов. Каждый раздел имеет заголовок в верхнем регистре, а разделы для определённого типа man-страниц следуют в строго определённом порядке. Для man-страниц категории 1 (Общие команды) разделы идут в следующем порядке:

Название Раздела	**Описание**
NAME (ИМЯ)	Название команды
SYNOPSIS (СИНТАКСИС)	Формат опций и аргументов
DESCRIPTION (ОПИСАНИЕ)	Описание назначения и использования
ENVIRONMENT (ОКРУЖЕНИЕ)	Настройки окружения, влияющие на работу
EXIT STATUS (СТАТУС ВЫХОДА)	Коды ошибок, возвращаемые при завершении
EXAMPLES (ПРИМЕРЫ)	Примеры использования
COMPATIBILITY (СОВМЕСТИМОСТЬ)	Совместимость с другими реализациями
SEE ALSO (СМ. ТАКЖЕ)	Перекрестная ссылка на связанные страницы руководства
STANDARDS (СТАНДАРТЫ)	Совместимость со стандартами, такими как POSIX
HISTORY (ИСТОРИЯ)	История реализации
BUGS (ИЗВЕСТНЫЕ ОШИБКИ)	Известные ошибки
AUTHORS (АВТОРЫ)	Люди, которые создали команду или написали справочную страницу.

Некоторые разделы являются необязательными, и сочетание разделов для конкретного типа справочной страницы может различаться. Примеры наиболее распространённых типов приведены далее в этой главе.

11.3.2. Макросы

Разметка mdoc(7) основана на *макросах*. Строки, начинающиеся с точки, содержат макрокоманды, каждая длиной в две или три буквы. Например, рассмотрим эту часть man-страницы ls(1):

```
.Dd December 1, 2015  ①
.Dt LS 1
.Sh NAME  ②
```

```
.Nm ls
.Nd list directory contents
.Sh SYNOPSIS  ③
.Nm  ④
.Op Fl -libxo  ⑤
.Op Fl ABCFGHILPRSTUWZabcdfghiklmnopqrstuwxy1,  ⑥
.Op Fl D Ar format  ⑦
.Op Ar  ⑧
.Sh DESCRIPTION  ⑨
For each operand that names a
.Ar file
of a type other than
directory,
.Nm
displays its name as well as any requested,
associated information.
For each operand that names a
.Ar file
of type directory,
.Nm
displays the names of files contained
within that directory, as well as any requested, associated
information.
```

① Определены *Дата документа* (.Dd) и *Название документа* (.Dt).

② Для раздела NAME определяется *Заголовок раздела* (.Sh). Затем определяется *Имя* (.Nm) команды и *Описание имени* (.Nd) в одну строку.

③ Начинается раздел SYNOPSIS (.Sh). В этом разделе описываются параметры командной строки и аргументы, которые принимаются.

④ *Имя* (`.Nm`) уже определено, и его повторение здесь просто отображает заданное значение в тексте.

⑤ Отображается *необязательный флаг* (.Op Fl) `-libxo`. Макрос `Fl` добавляет тире в начало флагов, поэтому в руководстве он отображается как `--libxo`.

⑥ Отображается длинный список необязательных флагов, состоящих из одного символа.

⑦ Определён необязательный флаг `-D`. Если указан флаг `-D`, за ним должен следовать *аргумент*. Аргумент представляет собой *формат* — строку, которая указывает ls(1), что и как отображать. Подробности о строке формата приведены далее на странице руководства.

⑧ Определен последний необязательный аргумент. Поскольку для аргумента не указано имя, используется значение по умолчанию `file …`.

⑨ *Заголовок раздела* (.Sh) для раздела DESCRIPTION определен.

При отображении с помощью команды `man ls` результат выводится на экран следующим образом:

```
LS(1)                   FreeBSD General Commands Manual                   LS(1)
```

```
NAME
     ls - list directory contents

SYNOPSIS
     ls [--libxo] [-ABCFGHILPRSTUWZabcdfghiklmnopqrstuwxy1,] [-D format]
        [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls
     displays its name as well as any requested, associated information.  For
     each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested,
     associated information.
```

Необязательные значения указаны в квадратных скобках.

11.3.3. Руководство по разметке

Язык разметки mdoc(7) не является очень строгим. Для ясности и согласованности проект FreeBSD Documentation добавляет некоторые дополнительные рекомендации по стилю:

Только первая буква макросов пишется с заглавной буквы

Всегда используйте заглавную букву для первого символа макроса и строчные буквы для остальных символов.

Начинайте новые предложения с новых строк

Начинайте новое предложение с новой строки, не начинайте его на той же строке, что и существующее предложение.

Обновите `.Dd` при внесении значительных изменений в справочную страницу

Дата документа сообщает читателю, когда страница руководства была последний раз обновлена. Важно обновлять её при внесении значительных изменений в страницы руководства. Незначительные правки, такие как исправления орфографии или пунктуации, которые не влияют на использование, могут быть сделаны без обновления `.Dd`.

Приведите примеры

Показывайте читателю примеры, когда это возможно. Даже простые примеры ценны, потому что то, что тривиально для автора, не обязательно тривиально для читателя. Хорошей целью являются три примера. Простой пример демонстрирует минимальные требования, серьезный пример показывает реальное использование, а углубленный пример демонстрирует неочевидную или нестандартную функциональность.

Включить лицензию BSD

Включите лицензию BSD в новые руководства. Предпочтительная лицензия доступна в Руководстве коммиттера.

11.3.4. Приемы разметки

Добавьте пробел перед пунктуацией на строке с макросами. Пример:

```
.Sh SEE ALSO
.Xr geom 4 ,
.Xr boot0cfg 8 ,
.Xr geom 8 ,
.Xr gptboot 8
```

Обратите внимание, как запятые в конце строк `.Xr` размещены после пробела. Макрос `.Xr` ожидает два параметра: имя внешней страницы руководства и номер раздела. Пробел отделяет пунктуацию от номера раздела. Без пробела внешние ссылки будут некорректно указывать на разделы `4,` или `8,`.

11.3.5. Важные макросы

Здесь будут показаны некоторые очень распространённые макросы. Для большего количества примеров использования см. mdoc(7), groff_mdoc(7) или выполните поиск реальных примеров в каталогах /usr/share/man/man*. Например, чтобы найти примеры макроса `.Bd` (*Begin display*):

```
% find /usr/share/man/man* | xargs zgrep '.Bd'
```

Организационные макросы

Некоторые макросы используются для определения логических блоков страницы руководства.

Организационный макрос	Использование
`.Sh`	Заголовок раздела. За ним следует название раздела, традиционно записанное в верхнем регистре. Можно рассматривать их как названия глав.
`.Ss`	Подзаголовок раздела. За ним следует название подраздела. Используется для разделения раздела `.Sh` на подразделы.
`.Bl`	Начало списка (Begin list). Начать перечень элементов.
`.El`	Конец списка (End List).
`.Bd`	Начало отображения (Begin display). Начать специальную область текста, например, область с отступом.
`.Ed`	Завершить отображение (End display).

Внутристрочные макросы

Многие макросы используются для разметки текста внутри строк.

Внутристрочный макрос	Использование
`.Nm`	Имя (Name). При первом использовании вызывается с именем в качестве параметра, затем используется без параметра для отображения уже определённого имени.
`.Pa`	Путь к файлу (Path to a file). Используется для обозначения имён файлов и путей к каталогам.

11.4. Пример структуры страницы Справочника

Этот раздел демонстрирует минимально желаемое содержание man-страниц для нескольких распространённых категорий руководств.

11.4.1. Раздел 1 или 8 — Команда

Предпочтительная базовая структура для раздела 1 или 8 — Команда:

```
.Dd August 25, 2017
.Dt EXAMPLECMD 8
.Os
.Sh NAME
.Nm examplecmd
.Nd "command to demonstrate section 1 and 8 man pages"
.Sh SYNOPSIS
.Nm
.Op Fl v
.Sh DESCRIPTION
The
.Nm
utility does nothing except demonstrate a trivial but complete
manual page for a section 1 or 8 command.
.Sh SEE ALSO
.Xr exampleconf 5
.Sh AUTHORS
.An Firstname Lastname Aq Mt flastname@example.com
```

11.4.2. Раздел 4 — Драйверы устройств

Предпочтительная базовая структура для раздела 4 — драйверы устройств:

```
.Dd August 25, 2017
.Dt EXAMPLEDRIVER 4
```

```
.Os
.Sh NAME
.Nm exampledriver
.Nd "driver to demonstrate section 4 man pages"
.Sh SYNOPSIS
To compile this driver into the kernel, add this line to the
kernel configuration file:
.Bd -ragged -offset indent
.Cd "device exampledriver"
.Ed
.Pp
To load the driver as a module at boot, add this line to
.Xr loader.conf 5 :
.Bd -literal -offset indent
exampledriver_load="YES"
.Ed
.Sh DESCRIPTION
The
.Nm
driver provides an opportunity to show a skeleton or template
file for section 4 manual pages.
.Sh HARDWARE
The
.Nm
driver supports these cards from the aptly-named Nonexistent
Technologies:
.Pp
.Bl -bullet -compact
.It
NT X149.2 (single and dual port)
.It
NT X149.8 (single port)
.El
.Sh DIAGNOSTICS
.Bl -diag
.It "flashing green light"
Something bad happened.
.It "flashing red light"
Something really bad happened.
.It "solid black light"
Power cord is unplugged.
.El
.Sh SEE ALSO
.Xr example 8
.Sh HISTORY
The
.Nm
device driver first appeared in
.Fx 49.2 .
.Sh AUTHORS
```

```
.An Firstname Lastname Aq Mt flastname@example.com
```

11.4.3. Раздел 5 — Файл конфигурации

Предпочтительная базовая структура для раздела 5 — файл конфигурации:

```
.Dd August 25, 2017
.Dt EXAMPLECONF 5
.Os
.Sh NAME
.Nm example.conf
.Nd "config file to demonstrate section 5 man pages"
.Sh DESCRIPTION
.Nm
is an example configuration file.
.Sh SEE ALSO
.Xr example 8
.Sh AUTHORS
.An Firstname Lastname Aq Mt flastname@example.com
```

11.5. Тестирование

Проверка новой страницы руководства может быть непростой задачей. К счастью, существуют инструменты, которые могут помочь в этом. Некоторые из них, например man(1), не ищут в текущем каталоге. Хорошей практикой является добавление префикса `./` к имени файла, если новая страница руководства находится в текущем каталоге. Также можно использовать абсолютный путь.

Используйте линтер mandoc(1) для проверки на ошибки парсинга:

```
% mandoc -T lint ./mynewmanpage.8
```

Используйте textproc/igor для проверки страниц Справочника:

```
% igor ./mynewmanpage.8
```

Еще один полезный инструмент — textproc/vale. Он не поддерживает синтаксис mdoc(7), но обработанную страницу руководства можно прочитать из стандартного ввода:

```
% man ls | vale
```

textproc/vale обладает высокой степенью настраиваемости. Рекомендуется ознакомиться с его документацией.

Используйте man(1) для проверки конечного результата ваших изменений:

```
% man ./mynewmanpage.8
```

Вы можете использовать col(1) для фильтрации вывода man(1) и удаления backspace-символов перед загрузкой результата в ваш любимый редактор для проверки орфографии:

```
% man ./mynewmanpage.8 | col -b | vim -R -
```

Проверка орфографии с использованием полнофункциональных словарей рекомендуется и может быть выполнена с помощью textproc/hunspell или textproc/aspell в сочетании с textproc/en-hunspell или textproc/en-aspell соответственно. Например:

```
% aspell check --lang=en --mode=nroff ./mynewmanpage.8
```

11.6. Примеры страниц Справочника для использования в качестве шаблонов

Некоторые страницы Справочника могут служить подробными примерами.

Страница Справочника	**Путь к расположению исходного кода**
cp(1)	/usr/src/bin/cp/cp.1
vt(4)	/usr/src/share/man/man4/vt.4
crontab(5)	/usr/src/usr.sbin/cron/crontab/crontab.5
gpart(8)	/usr/src/sbin/geom/class/part/gpart.8

11.7. Ресурсы

Ресурсы для авторов справочных страниц:

- man(1)
- mandoc(1)
- style.mdoc(5)
- groff_mdoc(7)
- Практические руководства UNIX: mdoc
- История man-страниц UNIX

Глава 12. Стиль написания

12.1. Советы

Техническая документация может быть улучшена за счёт последовательного применения нескольких принципов. Большинство из них можно разделить на три цели: *быть понятным*, *быть полным* и *быть кратким*. Эти цели могут противоречить друг другу. Хорошее изложение заключается в нахождении баланса между ними.

12.1.1. Будьте понятны

Ясность чрезвычайно важна. Читатель может быть новичком или изучать документ на неродном языке. Стремитесь к простому, понятному тексту, который чётко объясняет концепции.

Избегайте витиеватой или украшенной речи, шуток или разговорных выражений. Пишите как можно проще и яснее. Простой текст легче понять и перевести.

Делайте объяснения как можно короче, проще и понятнее. Избегайте пустых фраз вроде "для того чтобы", которые обычно можно заменить на "чтобы". Не используйте снисходительные слова вроде "по сути". Избегайте латинских терминов, таких как "i.e." или "cf.", которые могут быть непонятны за пределами академических или научных кругов.

Следует писать в формальном стиле. Необходимо избегать прямого обращения к читателю. Например, следует использовать формулировку "скопировать файл в /tmp", а не "вы можете скопировать файл в /tmp".

Приводите понятные, корректные и *проверенные* примеры. Простой пример лучше, чем отсутствие примера. Хороший пример ещё лучше. Не приводите плохих примеров, которые можно распознать по извинениям или фразам вроде «но на самом деле так делать не стоит». Плохие примеры хуже, чем их отсутствие. Приводите хорошие примеры, потому что *даже если предупредить читателя, чтобы он не использовал пример в таком виде*, он всё равно, скорее всего, воспользуется им как есть.

Избегайте *неопределённых выражений*, таких как «должен», «может», «попытаться» или «смог бы». Эти слова создают впечатление, что автор не уверен в фактах, и вызывают сомнения у читателя.

Аналогично, давайте инструкции в виде повелительных команд: не "вам следует сделать это", а просто "сделайте это".

12.1.2. Будьте полными

Не делайте предположений о способностях или уровне навыков читателя. Сообщите им то, что им необходимо знать. Давайте ссылки на другие документы, чтобы предоставить справочную информацию без необходимости её повторного изложения. Поставьте себя на место читателя, предугадайте вопросы, которые у него возникнут, и ответьте на них.

12.1.3. Будьте краткими

Хотя функции должны быть описаны полностью, иногда информации так много, что читателю сложно найти нужную деталь. Баланс между полнотой и лаконичностью — это непростая задача. Один из подходов — включить введение, затем раздел «быстрый старт» с описанием наиболее распространённого сценария, а после него — подробный справочный раздел.

12.2. Рекомендации

Для обеспечения единообразия среди множества авторов документации FreeBSD были разработаны рекомендации, которым следует придерживаться при написании.

Используйте американское написание английских слов

Существует несколько вариантов английского языка с разным написанием одних и тех же слов. Если варианты написания отличаются, используйте американский английский. Например, «color», а не «colour», «rationalize», а не «rationalise», и так далее.



Использование британского варианта английского языка может быть допустимо в случае предоставленной статьи, однако орфография должна быть единообразной во всем документе. Остальные документы, такие как книги, веб-сайты, справочные страницы и т. д., должны использовать американский вариант английского языка.

Не используйте сокращения

Не используйте сокращения. Всегда пишите фразу полностью. "Don’t use contractions" является неправильным вариантом.

Избегание сокращений способствует более формальному тону, повышает точность и немного облегчает работу переводчиков.

Используйте серийную запятую

В списке элементов внутри абзаца разделяйте каждый элемент запятой. Последний элемент отделяйте от остальных запятой и словом "and".

Для примера:

This is a list of one, two and three items (Вот список из одного, двух и трёх элементов).

Это список из трёх элементов: "one", "two" и "three" или список из двух элементов: "one" и "two and three"?

Лучше быть точным и включать серийную запятую:

This is a list of one, two, and three items (Вот список из одного, двух и трёх элементов).

Избегайте избыточных фраз

Избегайте избыточных фраз. В частности, "команда", "файл" и "man команда" часто избыточны.

Например, команды:

Неверно: Используйте команду `git` для обновления исходного кода.

Правильно: Используйте `git` для обновления исходников.

Имена файлов:

Неверно: ... в имени файла /etc/rc.local...

Правильно: ... в /etc/rc.local...

Ссылки на страницы Справочника (во втором примере используется `man:[]` с сущностью csh(1)):

Неправильно: Дополнительную информацию смотрите в `man csh`.

Правильно: См. csh(1).

Для получения дополнительной информации о стиле написания текстов см. «Элементы стиля» Уильяма Странка.

12.3. Руководство по стилю

Чтобы сохранить единообразие исходного кода документации при редактировании множеством разных людей, пожалуйста, следуйте этим стилевым соглашениям.

12.3.1. Одно предложение на строку

Используйте семантические разрывы строк в документации, метод, называемый "одно предложение на строку". Идея этого метода заключается в том, чтобы помочь пользователям писать и читать документацию. Для получения дополнительной информации об этом методе прочитайте страницу Semantic Line Breaks.

Это пример, который не использует "одно предложение на строку".

```
All human beings are born free and equal in dignity and rights. They are endowed with
reason and conscience and should act towards one another in a spirit of brotherhood.
```

И вот пример, использующий этот метод.

```
All human beings are born free and equal in dignity and rights.
They are endowed with reason and conscience and should act towards one another in a
spirit of brotherhood.
```

12.3.2. Аббревиатуры

Акроним должен быть определён при первом упоминании в документе, например:

«Network Time Protocol (NTP)». После того как акроним определён, используйте только акроним, если контекст не требует полного термина. Обычно акроним определяется один раз в главе или документе.

Все аббревиатуры должны быть заключены в символ `.

12.4. Список специальных символов

Этот список специальных символов показывает правильный синтаксис и результат их использования в документации FreeBSD. Если нужного символа нет в списке, задайте вопрос на Список рассылки Проекта Документации FreeBSD.

Имя	Синтаксис	Выводится
Авторское право (copyright)	(C)	(C)
Зарегистрировано	(R)	®
Товарный знак	(TM)	™
Тире	--	—
Многоточия	...	…
Одиночная стрелка вправо	->	→
Двойная правая стрелка	=>	⇒
Одиночная стрелка влево	<-	←
Двойная левая стрелка	<=	⇐

12.5. Проверка стиля с помощью Vale

Для обеспечения ясности и единообразия во всей документации и на страницах веб-сайта в дерево документации были добавлены стили Vale. Vale — это мощный линтер для создания пользовательских правил написания текста, который можно использовать в различных сценариях. В настоящее время Vale можно применять как инструмент командной строки, в CI/CD-процессах и интегрировать в предпочитаемый редактор.

Следующая таблица описывает текущие названия правил и их соответствующую степень серьёзности.

Имя	Степень серьезности
FreeBSD.BrandTerms	ошибка
FreeBSD.ConsciousLanguage	warning
FreeBSD.Contractions	предложение
FreeBSD.EOLSpacing	warning
FreeBSD.Hang	warning
FreeBSD.Hyphens	warning

Имя	Степень серьезности
FreeBSD.Spacing	ошибка
FreeBSD.SuperfluousOptArgInLinks	предложение
Vale.Avoid	ошибка
Vale.Repetition	ошибка
Vale.Spelling	ошибка
Vale.Terms	ошибка

12.5.1. Текущие правила Vale

1. FreeBSD.BrandTerms: В соответствии с правилами авторского права Фонда FreeBSD, **freebsd** следует писать как **FreeBSD**. Аналогично, у каждого крупного поставщика и компании есть свои правила написания их торговых марок и брендов. Следует проявлять уважение к брендам других и уделять время правильному написанию PostgreSQL, Node.js, Let's Encrypt и т. д. Отсутствующие названия брендов следует добавлять в файл .vale/styles/FreeBSD/BrandTerms.yml в репозитории `doc`.
2. FreeBSD.ConsciousLanguage: Это правило предлагает использовать осознанный язык, чтобы по возможности избегать чувствительных слов, указывающих на цвет кожи, возраст, расу или сексуальную ориентацию людей.
3. FreeBSD.Contractions: Не следует использовать сокращенные слова. Это правило исключает все сокращения и рекомендует использовать полные слова.
4. FreeBSD.EOLSpacing: В большинстве документов присутствуют пробелы в конце строк, что не является желательной ситуацией.
5. FreeBSD.Hang: Термин `Hang` часто используется для обозначения ситуации, когда приложение перестает отвечать. Это правило предлагает более подходящую формулировку.
6. FreeBSD.Hyphens: Часто наречия, оканчивающиеся на 'ly', добавляются с дефисом, что является ошибкой.
7. FreeBSD.Spacing: Часто двойные пробелы трудно заметить невооруженным глазом, и здесь это решается.
8. FreeBSD.SuperfluousOptArgInLinks: Рекомендуется оставлять квадратные скобки пустыми в макросах `link:`, если отображаемый текст совпадает с URL.
9. Vale.Avoid: Запрещает использование терминов из списка **НЕ ИСПОЛЬЗОВАТЬ** в документации The FreeBSD Project. Если обнаружено слово, которое не должно присутствовать в документации, его следует добавить в файл .vale/styles/Vocab/Terms/reject.txt в репозитории `doc`. На данный момент список пуст.
10. Vale.Repetition: Одни и те же слова часто набираются дважды при отходе от клавиатуры и возвращении к работе. Это правило находит повторяющиеся слова и предупреждает пользователей.
11. Vale.Spelling: В настоящее время в документации и на веб-сайте используется смесь написания en_US и en_GB. Vale поставляется со встроенным словарем, который

использует строго en_US и не принимает варианты написания en_GB для любых слов.

12. Vale.Terms: Обеспечивает соблюдение **ПРЕДПОЧТИТЕЛЬНЫХ** терминов для проекта FreeBSD. В настоящее время список терминов пуст, и специфичные для FreeBSD термины будут добавляться постепенно. Если какое-либо слово считается правильным, но отсутствует в словаре, его следует добавить в файл .vale/styles/Vocab/Terms/accept.txt в репозитории `doc`.

Дополнительные правила будут введены в будущем по мере необходимости.

12.5.2. Использование Vale

Vale можно использовать из командной строки, а также из редактора или IDE. Установить textproc/vale можно следующим образом:

```
$ pkg install vale
```

Использование Vale в командной строке

Предполагая, что репозиторий `doc` был склонирован в ~/doc, для запуска потребуются следующие команды:

```
% cd ~/doc
% vale .
```



Vale — это программа с высокой нагрузкой на процессор и память из-за специфики приложения, и она может долго не выводить результаты на экран. Лучше запускать её для конкретных папок или файлов, а не для всего репозитория `doc`, так как это уже выполняется в CI-пайплайне.

Использование Vale в редакторах

Vale работает с основными популярными редакторами, такими как editors/vim, editors/emacs, editors/vscode. В настоящее время необходимая конфигурация для editors/vim описана в Vim. Конфигурация для editors/emacs находится в разработке.

Глава 13. Настройка редактора

Настройка конфигурации текстового редактора может ускорить и упростить работу с документами, а также помочь им соответствовать рекомендациям FDP.

13.1. Vim

Установите из editors/vim, затем следуйте инструкциям по настройке в разделе Конфигурация. Более опытные пользователи могут использовать полноценный линтер, например Ale, который также может работать как клиент Language Server Protocol для Vim.

13.1.1. Использование

Создатели страниц Справочника могут использовать следующие сочетания клавиш для переформатирования:

- Нажмите `P`, чтобы переформатировать абзацы или выделенный текст в режиме Visual.
- Нажмите `T`, чтобы заменить группы из восьми пробелов на табуляцию.

В документацию добавлен линтер Vale для проверки грамматических и стилистических ошибок. Vale поддерживает различные редакторы и IDE.

Vale может быть уже установлен как зависимость мета-порта textproc/docproj. Если нет, установите textproc/vale с помощью:

```
$ pkg install vale
```

Установите Ale для интеграции в editors/vim, чтобы использовать textproc/vale.

```
% mkdir -p ~/.vim/pack/vendor/start
% git clone --depth 1 https://github.com/dense-analysis/ale.git
~/.vim/pack/vendor/start/ale
```

Пользователи, использующие менеджеры плагинов для editors/vim, не нуждаются в вышеописанном и должны следовать инструкциям своего менеджера плагинов для установки Ale.

В данный момент из-за ошибки в Vale необходимо скопировать конфигурацию Vale в домашний каталог. Учитывая, что репозиторий был склонирован в ~/doc, скопируйте следующим образом:

```
% cp -R ~/doc/.vale* ~/
```

13.1.2. Конфигурация

Отредактируйте файл ~/.vimrc, добавив следующие строки в конец файла:

~/.vimrc

```
if has("autocmd")
  au BufNewFile,BufRead *.adoc call Set_ADOC()
  au BufNewFile,BufRead *.[1-9] call Set_MAN()
endif " has(autocmd)

function Set_Highlights()
  "match ExtraWhitespace /^\s* \s*\|\s\+$/
  return 0
endfunction " Set_Highlights_Adoc()

function Set_Highlights_MAN()
  highlight default link OverLength ErrorMsg
  match OverLength /\%71v.\+/
  return 0
endfunction " Set_Highlights_MAN()

function ShowSpecial()
  setlocal list listchars=tab:>>,trail:*,eol:$
  hi def link nontext ErrorMsg
  return 0
endfunction " ShowSpecial()

function Set_COMMON()
  setlocal number
  setlocal shiftwidth=2
  setlocal tabstop=8
  setlocal softtabstop=2
  setlocal formatprg="fmt -p"
  setlocal autoindent
  setlocal smartindent
  call ShowSpecial()
  call Set_Highlights()
  return 0
endfunction " Set_COMMON()

function Set_ADOC()
  setlocal syntax=asciidoc
  setlocal filetype=asciidoc
  call Set_COMMON()
  return 0
endfunction " Set_ADOC()

function Set_MAN()
  setlocal syntax=man
  setlocal filetype=man
```

```
  setlocal textwidth=70
  " Rewrap paragraphs
  noremap P gqj
  " Replace spaces with tabs
  noremap T :s/        /\t/<CR>
  call Set_COMMON()
  call Set_Highlights_MAN()
  return 0
endfunction " Set_Man()

let g:ale_fixers = {
\   '*': ['remove_trailing_lines', 'trim_whitespace'],
\}
let g:ale_linters = {
\   'asciidoc': ['vale'],
\}
let g:ale_fix_on_save = 1
```



Приведённая выше конфигурация автоматически удалит конечную строку, конечные пробелы и множественные пробелы, которые могут отображать дополнительные нежелательные изменения в выводе `git diff`. В таких случаях обязательно укажите это в сообщении коммита.

13.2. Emacs

Установка из editors/emacs или editors/emacs-devel.

13.2.1. Автоматизированная проверка правописания с Flycheck и Igor

Пакет Flycheck доступен из Milkypostman's Emacs Lisp Package Archive (MELPA). Если MELPA ещё не добавлен в packages-archives Emacs, его можно добавить, выполнив

```
(add-to-list 'package-archives '("melpa" . "http://stable.melpa.org/packages/") t)
```

Добавьте строку в файл инициализации Emacs (один из ~/.emacs, ~/.emacs.el или ~/.emacs.d/init.el), чтобы сделать это изменение постоянным.

Для установки Flycheck выполните

```
(package-install 'flycheck)
```

Создайте проверяющий модуль Flycheck для textproc/igor с помощью выполнения

```
(flycheck-define-checker igor
  "FreeBSD Documentation Project sanity checker.
```

```
See URLs https://www.freebsd.org/docproj/ and
http://www.freshports.org/textproc/igor/."
  :command ("igor" "-X" source-inplace)
  :error-parser flycheck-parse-checkstyle
  :modes (nxml-mode)
  :standard-input t)

  (add-to-list 'flycheck-checkers 'igor 'append)
```

Еще раз, добавьте эти строки в файл инициализации Emacs, чтобы изменения стали постоянными.

13.2.2. Специфичные настройки документации FreeBSD

Чтобы применить настройки, специфичные для проекта документации FreeBSD, создайте файл .dir-locals.el в корневом каталоге репозитория документации и добавьте в него следующие строки:

```
;;; Directory Local Variables
;;; For more information see (info "(emacs) Directory Variables")

((nxml-mode
  (eval . (turn-on-auto-fill))
  (fill-column . 70)
  (eval . (require 'flycheck))
  (eval . (flycheck-mode 1))
  (flycheck-checker . igor)
  (eval . (add-to-list 'rng-schema-locating-files "~/.emacs.d/schema/schemas.xml"))))
```

13.3. nano

Установка из editors/nano.

13.3.1. Конфигурация

В текущей версии nano нет файла подсветки синтаксиса для adoc/asciidoc. Поэтому создадим его с нуля, используя текстовый редактор для создания нового файла или добавления строк в ~/.nanorc со следующим содержимым:

~/.nanorc

```
syntax "asciidoc" "\.(adoc|asc|asciidoc)$"
# main header
color red "^====+$"
# h1
color red "^==[[:space:]].*$"
color red "^----+$"
# h2
```

```
color magenta "^===[[:space:]].*$"
color magenta "^~~~~+$"
# h4
color green "^====[[:space:]].*$"
color green "^\^\^\^\^\^+$"
# h5
color brightblue "^=====[[:space:]].*$"
color brightblue "^\+\+\+\+\++$"
# attributes
color brightgreen ":.*:"
color brightred "\{[a-z0-9]*\}"
color red "\\\{[a-z0-9]*\}"
color red "\+\+\+\{[a-z0-9]*\}\+\+\+"
# Paragraph Title
color yellow "^\..*$"
# source
color magenta "^\[(source,.+|NOTE|TIP|IMPORTANT|WARNING|CAUTION)\]"
# Other markup
color yellow ".*[[:space:]]\+$"
color yellow "_[^_]+_"
color yellow "\*[^\*]+\*"
color yellow "\+[^\+]+\+"
color yellow "`[^`]+`"
color yellow "\^[^\^]+\^"
color yellow "~[^~]+~"
color yellow "'[^']+'"
color cyan "`{1,2}[^']+'{1,2}"
# bullets
color brightmagenta "^[[:space:]]*[\*\.-]{1,5}[[:space:]]"
# anchors
color brightwhite "\[\[.*\]\]"
color brightwhite "<<.*>>"
# trailing whitespace
color ,blue "[[:space:]]+$"
# multiples of eight spaces at the start a line
# (after zero or more tabs) should be a tab
color ,blue "^([TAB]*[ ]{8})+"
# tabs after spaces
color ,yellow "( )+TAB"
# highlight indents that have an odd number of spaces
color ,red "^(([ ]{2})+|(TAB+))*[ ]{1}[^ ]{1}"
```

Обработать файл для создания встроенных табуляций:

```
% perl -i'' -pe 's/TAB/\t/g' ~/.nanorc
```

13.3.2. Использование

Укажите дополнительные полезные опции при запуске редактора:

```
% nano -AKipwz -T8 _index.adoc
```

Пользователи csh(1) могут определить алиас в ~/.cshrc, чтобы автоматизировать эти параметры:

```
alias nano "nano -AKipwz -r 70 -T8"
```

После определения псевдонима параметры будут добавлены автоматически:

```
% nano _index.adoc
```

Глава 14. Товарные знаки

Для всех документов в рамках проекта FreeBSD Documentation Project необходимо указывать зарегистрированные товарные знаки, а также принято указывать другие товарные знаки, и это является требованием для каждого автора и участника.

14.1. Символы товарных знаков

Добавьте символ товарного знака (™, ® или другой) к первому упоминанию зарегистрированного названия и всегда при использовании логотипов. Используйте эквивалентную ASCII-последовательность, которая будет отображаться как соответствующий символ Юникода. Также пишите зарегистрированное название в соответствии с его правилами использования товарного знака.

Если сомневаетесь, изучите сайт владельца товарного знака, сайт продукта и/или сайт Ведомства по патентам и товарным знакам США для поиска товарных знаков.

14.2. Цитирование товарных знаков

Проект документации FreeBSD предоставляет шаблон для указания товарных знаков, что также позволяет избежать их дублирования в документах.

Сначала найдите товарный знак в разделе Copyright в шаблоне проекта, затем добавьте его в тег `trademarks` в разделе `Front Matter` документа, который расположен в начале каждого документа.

Вот пример `Front Matter` из статьи Вклад в FreeBSD:

```
---
title: Contributing to FreeBSD
authors:
  - author: Jordan Hubbard
  - author: Sam Lawrance
  - author: Mark Linimon
description: How to contribute to the FreeBSD Project
trademarks: ["freebsd", "ieee", "general"]
weight: 15
tags: ["Contributing", "FreeBSD", "Non-Programmer Tasks", "Programmer Tasks"]
---
```

Товарные знаки `freebsd`, `ieee` и `general` будут автоматически отображаться при сборке документа следующим образом:

```
FreeBSD is a registered trademark of the FreeBSD Foundation.

IEEE, POSIX, and 802 are registered trademarks of Institute of Electrical and
Electronics Engineers, Inc. in the United States.
```

```
Many of the designations used by manufacturers and sellers to distinguish their
products are claimed as trademarks. Where those designations appear in this document,
and the FreeBSD Project was aware of the trademark claim, the designations have been
followed by the “™” or the “®” symbol.
```

Если товарный знак отсутствует в шаблоне проекта, его необходимо предоставить. Любой разработчик или участник может обновлять товарные знаки.

Теги товарных знаков `freebsd` и `general` обычно присутствуют во всех документах.

Глава 15. Смотрите также

Этот документ намеренно не является исчерпывающим обсуждением AsciiDoc и проекта документации FreeBSD. Для получения дополнительной информации о них рекомендуется посетить следующие веб-сайты.

15.1. Проект документации FreeBSD

- Веб-страницы проекта документации FreeBSD
- Руководство по FreeBSD

15.2. Hugo

- Hugo
- Документация Hugo

15.3. AsciiDoctor

- AsciiDoctor
- Документационный портал AsciiDoctor

15.4. HTML

- Консорциум Всемирной паутины
- Спецификация HTML 5
- Спецификация CSS
- Sass

Приложение A: Примеры

Эти примеры не являются исчерпывающими — они не содержат всех элементов, которые может быть желательно использовать, особенно в преамбуле документа. Для дополнительных примеров использования AsciiDoctor изучите исходный AsciiDoc код для этого и других документов, доступных в Git-репозитории **doc** или онлайн, начиная с https://cgit.freebsd.org/doc/.

A.1. Книга AsciiDoctor

Пример 15. **Книга** *AsciiDoctor*

```
---
title: An Example Book
authors:
  - author: The FreeBSD Documentation Project
copyright: 1995-2021 The FreeBSD Documentation Project
releaseinfo: ""
trademarks: ["general"]
---

= Пример книги
:doctype: book
:toc: macro
:toclevels: 2
:icons: font
:xrefstyle: basic
:relfileprefix: ../
:outfilesuffix:
:sectnums:
:partnums:
:chapter-signifier: Chapter
:part-signifier: Part
:source-highlighter: rouge
:experimental:
:skip-front-matter:
:book: true
:pdf: false


:chapters-path:


[abstract] Аннотация

Раздел аннотации

'''
```

```
toc::[]

:sectnums!:

include::{chapters-path}preface/_index.adoc[leveloffset=+1]

:sectnums:

include::{chapters-path}parti.adoc[lines=7..18]

include::{chapters-path}chapter-name/_index.adoc[leveloffset=+1]
```

A.2. Статья AsciiDoctor

Пример 16. **Статья** *AsciiDoctor*

```
---
title: An Example Article
authors:
  - author: Your name and surname
    email: foo@example.com
trademarks: ["general"]
---

= Пример статьи
:doctype: article
:toc: macro
:toclevels: 1
:icons: font
:sectnums:
:source-highlighter: rouge
:experimental:

'''

toc::[]

== Мой первый раздел

Это первый раздел в моей статье.

=== Мой первый подраздел

Это первый подраздел в моей статье.
```