Primer do Projeto de Documentação do FreeBSD para Novos Colaboradores

Resumo

Obrigado por fazer parte do Projeto de Documentação do FreeBSD. A sua contribuição é extremamente valiosa.

Esta cartilha cobre os detalhes necessários para começar a contribuir com o Projeto de Documentação do FreeBSD, incluindo ferramentas, software e a filosofia por trás do Projeto de Documentação.

Este documento é um trabalho em andamento. Correções e adições são sempre bem vindas.

Índice

Prefácio . . . 5
- Prompts do Shell . . . 5
- Convenções Tipográficas . . . 5
- Notas, Dicas, Informações Importantes, Avisos e Exemplos . . . 5
- Agradecimentos . . . 6

1. Visão geral . . . 7
- 1.1. Documentação no Ecossistema FreeBSD . . . 7
- 1.2. Introdução . . . 8
- 1.3. Conjunto de Documentação do FreeBSD . . . 12

2. Ferramentas . . . 14
- 2.1. Ferramentas Obrigatórias . . . 14
- 2.2. Ferramentas Opcionais . . . 14

3. A Área de Trabalho . . . 15
- 3.1. Documentação e Páginas de Manual . . . 15
- 3.2. Escolhendo um Diretório . . . 15
- 3.3. Baixando uma Cópia . . . 15
- 3.4. Atualizando . . . 16
- 3.5. Revertendo Alterações . . . 16
- 3.6. Criando um Diff . . . 16
- 3.7. Referências Git . . . 17

4. Estrutura de Diretórios da Documentação . . . 18
- 4.1. O Nível Superior, doc/ . . . 18
- 4.2. Os Diretórios . . . 18
- 4.3. Informação Específica de Documentação . . . 19
- 4.4. Os Livros: books/ . . . 19
- 4.5. Os Artigos: articles/ . . . 20

5. Processo de Compilação da Documentação do FreeBSD . . . 22
- 5.1. Renderizando AsciiDoc . . . 22
- 5.2. O Conjunto de Ferramentas de Compilação da Documentação do FreeBSD . . . 24
- 5.3. Compreendendo o Makefile na Árvore de Documentação . . . 24

6. Primer Asciidoctor . . . 34
- 6.1. Visão geral . . . 34
- 6.2. Cabeçalhos . . . 35
- 6.3. Parágrafos . . . 35
- 6.4. Listas . . . 36
- 6.5. Links . . . 36
- 6.6. Imagens e Ícones . . . 38
- 6.7. Conclusão . . . 39

7. Rosetta Stone . . . 40
7.1. Comparação entre Docbook e AsciiDoc . . . 40
8. Traduções . . . 44
8.1. O que significa i18n e l10n? . . . 44
8.2. Existe uma lista de discussão para tradutores? . . . 44
8.3. São necessários mais tradutores? . . . 44
8.4. Quais idiomas eu preciso saber? . . . 44
8.5. Quais softwares eu preciso conhecer? . . . 44
8.6. Como eu faço para descobrir se já existem outras pessoas traduzindo documentos para o meu idioma? . . . 45
8.7. Ninguém mais está traduzindo para o meu idioma. O que eu faço? . . . 45
8.8. Eu já tenho alguns documentos traduzidos, para onde eu devo enviá-los? . . . 46
8.9. Eu sou a única pessoa trabalhando na tradução para este idioma, como faço para enviar minha tradução? . . . 46
8.10. Posso incluir um texto específico do idioma ou do país em minha tradução? . . . 47
9. Traduções PO . . . 49
9.1. Introdução . . . 49
9.2. Introdução . . . 49
9.3. Criando Novas Traduções . . . 50
9.4. Traduzindo . . . 52
9.5. Dicas para Tradutores . . . 52
9.6. Compilando um Documento Traduzido . . . 53
9.7. Submetendo a Nova Tradução . . . 53
10. Traduções Weblate . . . 55
10.1. Introdução . . . 55
10.2. Como se Tornar um Tradutor do FreeBSD . . . 55
10.3. Apresente-se . . . 55
10.4. Entrar no Weblate . . . 56
10.5. Encontre uma Equipe de Idiomas para Participar . . . 57
10.6. Traduzindo Online no Weblate . . . 57
10.7. Traduzindo Offline . . . 59
10.8. Tradução baseada em Sugestões Automáticas . . . 59
10.9. Revisão e Verificações de Qualidade do Weblate . . . 62
10.10. Compilando o Documento Traduzido . . . 63
10.11. Enviando Traduções . . . 66
10.12. FAQ (Perguntas Frequentes) . . . 68
11. Páginas de Manual . . . 69
11.1. Introdução . . . 69
11.2. Seções . . . 69
11.3. Marcação . . . 69
11.4. Exemplo de Estruturas de Página de Manual . . . 74

11.5. Testando . . . 76
11.6. Exemplos de páginas de manuais para usar como modelos . . . 77
11.7. Recursos . . . 77
12. Estilo de escrita . . . 78
12.1. Dicas . . . 78
12.2. Diretrizes . . . 79
12.3. Guia de estilo . . . 80
12.4. Uma frase por linha . . . 80
12.5. Siglas . . . 80
12.6. Lista de Caracteres Especiais . . . 81
12.7. Linting com Vale . . . 81
13. Configuração do Editor . . . 84
13.1. Vim . . . 84
13.2. Emacs . . . 86
13.3. nano . . . 87
14. Marcas Registradas . . . 90
14.1. Símbolos de Marca Registrada . . . 90
14.2. Citação de Marca Registrada . . . 90
15. Veja também . . . 92
15.1. O projeto de documentação do FreeBSD . . . 92
15.2. Hugo . . . 92
15.3. AsciiDoctor . . . 92
15.4. HTML . . . 92
Apêndice A: Exemplos . . . 93
A.1. AsciiDoctor **book** . . . 93
A.2. **Artigo** AsciiDoctor . . . 94

Prefácio

Prompts do Shell

Esta tabela mostra o prompt padrão do sistema e o prompt do super usuário. Os exemplos usam estes prompts para indicar com qual usuário o exemplo foi executado.

Usuário	Prompt
Usuário normal	%
`root`	#

Convenções Tipográficas

Esta tabela descreve as convenções tipográficas utilizadas neste livro.

Propósito	Exemplos
Nome dos comandos.	Utilize `ls -l` para listar todos os arquivos.
Nome dos arquivos.	Edite .login.
Saída de um programa na tela do computador.	`You have mail.`
O que o usuário digita, quando contrastado com a saída do programa na tela do computador.	`% date +"The time is %H:%M"` `The time is 09:18`
Referência a uma página de manual.	Utilize su(1) para assumir outro nome de usuário.
Nome de usuário e de grupos de usuários.	Apenas o `root` pode fazer isso.
Ênfase.	O usuário *deve* fazer isso.
Texto que o usuário deve substituir com o texto real.	Para buscar por uma palavra chave nas páginas de manual, digite `man -k keyword`
Variáveis de ambiente.	`$HOME` aponta para o diretório inicial do usuário.

Notas, Dicas, Informações Importantes, Avisos e Exemplos

Notas, avisos e exemplos aparecem ao longo do texto.

Notas são representadas desta forma, e contêm informações para as quais se deve ficar atento, pois podem afetar o que o usuário faz.

Dicas são representadas desta forma, e contêm informações úteis para o usuário, como as que mostram uma maneira mais fácil de fazer alguma coisa.

Informações importantes são representadas desta forma. Normalmente elas destacam passos extras que o usuário pode precisar realizar.

Avisos são representados deste modo, e contêm informações de alerta sobre possíveis danos se não seguir as instruções. Estes danos podem ser físicos, para o equipamento ou para o usuário, ou podem ser não-físicos, tal como a deleção inadvertida de arquivos importantes.

Exemplo 1. Uma Amostra de Exemplo

Os exemplos são representados deste modo, e normalmente contêm exemplos passo a passo, ou mostram os resultados de uma determinada ação.

Agradecimentos

Meu muito obrigado a Sue Blake, Patrick Durusau, Jon Hamilton, Peter Flynn, e Christopher Maden, por terem gasto parte do seu tempo lendo os primeiros rascunhos deste documento e por terem oferecido muitos comentários e críticas construtivas para este trabalho.

Capítulo 1. Visão geral

Seja bem vindo ao Projeto de Documentação do FreeBSD.(FDP). Documentação de boa qualidade é muito importante para o sucesso do FreeBSD, e nós valorizamos muito suas contribuições.

Este documento descreve como o FDP é organizado, como escrever e como submeter documentos, e como utilizar de forma efetiva as ferramentas que estão disponíveis.

Todos são bem vindos para se juntar ao FDP. A vontade de contribuir é o único requisito de adesão.

Este primer mostra como:

- Compreender o papel da documentação e seu lugar no ecossistema.
- Identificar quais partes do FreeBSD são mantidas pelo FDP.
- Instalar as ferramentas e arquivos de documentação necessários.
- Realizar alterações na documentação.
- Enviar de volta alterações para revisão e inclusão na documentação do FreeBSD.

1.1. Documentação no Ecossistema FreeBSD

Todos os documentos são para o benefício de seus leitores, não de seus escritores ou zeladores. Eles devem se adaptar ao leitor e não esperar que o leitor se adapte a eles.

Nunca culpe o leitor por:

- ser incapaz de fazer uso de um documento facilmente ou de tudo
- achar o documento confuso
- não entender o documento ou como utilizá-lo
- não encontrar uma resposta explícita ou preencher lacunas com sucesso (ou conectando pontos) para raciocinar em direção a uma

Em vez disso, reconheça que o documento é:

- inacessível
- confuso
- difícil de entender ou utilizar
- incompleto

Em seguida, faça o documento:

- mais acessível
- menos confuso
- mais claro
- mais completo

Use os seguintes métodos:

- aplique as melhores práticas de acessibilidade para corrigir o problema relatado e quaisquer outros semelhantes que você encontrar
- refazer ou esclarecer a estrutura ou linguagem confusa
- adicionar exemplos relevantes para a parte que é difícil de entender ou aplicar
- preencha as lacunas ou adicione os degraus que faltam

1.2. Introdução

Algumas etapas preparatórias devem ser seguidas antes de editar a documentação do FreeBSD. Primeiro, se registre na lista de discussão do projeto de documentação do FreeBSD. Alguns membros do time também interagem no IRC, canal `#bsddocs` na rede EFnet. Estas pessoas podem ajudar com questões e problemas envolvendo documentação.

1.2.1. Processo de instalação do FreeBSD

1. Instale esses pacotes. O *meta-port* `docproj` instala todos os aplicativos necessários para editar e compilar a documentação do FreeBSD.

   ```
   # pkg install docproj
   ```

2. Obtenha uma cópia local da árvore de documentação do FreeBSD em ~/doc (ver A Área de Trabalho).

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

3. Edite os arquivos de documentação que precisam de alterações. Se um arquivo precisar de grandes mudanças, consulte a lista de discussão para obter informações.

   Revise a saída e edite o arquivo para corrigir os problemas informados e, em seguida, execute novamente o comando para verificar os problemas restantes. Repita até que todos os erros sejam resolvidos.

4. ***Sempre*** realize testes de compilação e revise as alterações antes de submeter algo. Execute `make` no diretório `documentation` ou `website` para gerar a documentação no formato HTML.

   ```
   % make
   ```

   Para reduzir o tempo de compilação, apenas um idioma pode ser compilado:

   ```
   % make DOC_LANG=en
   ```

A saída da compilação é armazenada em ~/doc/documentation/public/en/articles/ e ~/doc/documentation/public/en/books/.

5. Revise a saída da compilação e certifique-se de que as edições não contenham erros de digitação, problemas de layout ou erros. Se algum erro for encontrado durante o processo de compilação, edite os arquivos com erro para corrigir quaisquer problemas que apareçam e, em seguida, execute o comando de compilação novamente até que todos os erros sejam resolvidos.
6. Adicione todos os arquivos com `git add .`, então revise o diff com `git diff`. Por exemplo:

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

Certifique-se de que todos os arquivos necessários estejam incluídos, então confirme a mudança em seu branch local e gere um patch com `git format-patch`

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

Patch gerado com `git format-patch` incluirá a identidade do autor e endereços de e-mail, tornando mais fácil para os desenvolvedores aplicarem (com `git am`) e dar os devidos créditos.

> Para tornar mais fácil para os committers aplicarem o patch em sua cópia de trabalho da árvore de documentação, por favor, gere o .diff da base de sua árvore de documentação.

No exemplo acima, foram feitas alterações na parte bsdinstall do Handbook.

7. Submeta o patch or arquivo diff pela web para o sistema de Relatórios de Problema. Se estiver usando o formulário web, insira um Sumário com *[patch] descrição curta do problema*. Selecione o Componente `Documentation`. No campo de Descrição, insira uma breve descrição das alterações e quaisquer detalhes importantes sobre elas. Use o botão **[Add an attachment]** para anexar o patch ou arquivo diff. Finalmente, pressione o botão **[Submit Bug]** para enviar seu diff para o sistema de relatório de problemas.

1.2.2. Processo de instalação GNU/Linux

1. Instale esses pacotes em sistemas baseados em apt como Debian ou Ubuntu. Em outras distribuições GNU/Linux os nomes dos pacotes podem mudar. Consulte o gerenciador de pacotes da sua distribuição em caso de dúvida.

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

2. Obtenha uma cópia local da árvore de documentação do FreeBSD em ~/doc (ver A Área de Trabalho).

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

3. Edite os arquivos de documentação que precisam de alterações. Se um arquivo precisar de grandes mudanças, consulte a lista de discussão para obter informações.

   Revise a saída e edite os arquivos para corrigir os problemas informados e, em seguida, execute novamente o comando para verificar os problemas restantes. Repita até que todos os erros sejam resolvidos.

4. Sempre compile e teste as alterações antes de enviá-las. A execução de `bmake` nos subdiretórios `documentation` ou `website` irá gerar a documentação em formato HTML.

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

5. Adicione todos os arquivos com `git add .`, então revise o diff com `git diff`. Por exemplo:

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

   Certifique-se de que todos os arquivos necessários estejam incluídos, então confirme a mudança em seu branch local e gere um patch com `git format-patch`

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

   Patch gerado com `git format-patch` incluirá a identidade do autor e endereços de e-mail, tornando mais fácil para os desenvolvedores aplicarem (com `git am`) e dar os devidos créditos.

   > Para tornar mais fácil para os committers aplicarem o patch em sua cópia de trabalho da árvore de documentação, por favor, gere o .diff da base de sua árvore de documentação.

6. Submeta o patch ou arquivo diff file pela web para o sistema de Relatórios de Problema. Se estiver usando o formulário web, insira um Sumário com uma *breve descrição do problema*. Selecione o Componente `Documentation`. No campo de Descrição, insira uma breve descrição das alterações e quaisquer detalhes importantes sobre elas e adicione *patch* no campo *Keywords*. Use o botão **[Add an attachment]** para anexar o patch ou arquivo diff. Finalmente, pressione o botão **[Submit Bug]** para enviar seu diff para o sistema de relatório de problemas.

1.2.3. Processo de instalação do macOS®

1. Instale esses pacotes usando o Homebrew e o RubyGem.

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

2. Adicione o Ruby ao Path.

```
$ echo 'export GEM_PATH="/usr/local/lib/ruby/gems/3.1.0"' >> ~/.zshrc
$ echo 'export PATH="$(brew --prefix ruby)/bin:$PATH"' >> ~/.zshrc
$ source ~/.zshrc
```

3. Adicione o alias do git ao Homebrew, pois `git format-patch` do git fornecido pela Apple não funcionará com o Phabricator.

```
$ echo 'alias git=/usr/local/bin/git' >> ~/.zshrc
$ source ~/.zshrc
```

4. Instale o pacote rouge usando RubyGem.

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

5. Obtenha uma cópia local da árvore de documentação do FreeBSD em ~/doc (ver A Área de Trabalho).

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

6. Edite os arquivos de documentação que precisam de alterações. Se um arquivo precisar de grandes mudanças, consulte a lista de discussão para obter informações.

   Revise a saída e edite os arquivos para corrigir os problemas informados e, em seguida, execute novamente o comando para verificar os problemas restantes. Repita até que todos os erros sejam resolvidos.

7. Sempre compile e teste as alterações antes de enviá-las. A execução de `bmake` nos subdiretórios `documentation` ou `website` irá gerar a documentação em formato HTML.

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

8. Adicione todos os arquivos com `git add .`, então revise o diff com `git diff`. Por exemplo:

```
% git add .
```

```
% git diff --staged
```

Certifique-se de que todos os arquivos necessários estejam incluídos, então confirme a mudança em seu branch local e gere um patch com `git format-patch`

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

Patch gerado com `git format-patch` incluirá a identidade do autor e endereços de e-mail, tornando mais fácil para os desenvolvedores aplicarem (com `git am`) e dar os devidos créditos.

> Para tornar mais fácil para os committers aplicarem o patch em sua cópia de trabalho da árvore de documentação, por favor, gere o .diff da base de sua árvore de documentação.

9. Submeta o patch ou arquivo diff file pela web para o sistema de Relatórios de Problema. Se estiver usando o formulário web, insira um Sumário com uma *breve descrição do problema*. Selecione o Componente `Documentation`. No campo de Descrição, insira uma breve descrição das alterações e quaisquer detalhes importantes sobre elas e adicione *patch* no campo *Keywords*. Use o botão **[Add an attachment]** para anexar o patch ou arquivo diff. Finalmente, pressione o botão **[Submit Bug]** para enviar seu diff para o sistema de relatório de problemas.

1.3. Conjunto de Documentação do FreeBSD

O FDP é responsável por quatro categorias de documentação do FreeBSD.

- *Handbook*: O Handbook almeja ser um compreensivo recurso de referência online para os usuários do FreeBSD.
- *FAQ*: O FAQ utiliza um formato curto de pergunta e resposta para abordar dúvidas que são frequentemente realizadas nas listas de discussão e fóruns dedicados ao FreeBSD. Este formato não permite respostas longas e detalhadas.
- *Páginas de Manual*: As páginas de manual do sistema de língua inglesa geralmente não são escritas pelo FDP, pois fazem parte do sistema base. Contudo, o FDP pode reformular partes das páginas de manual existentes para torná-las mais claras ou para corrigir imprecisões.
- *Web site*: Esta é a presença principal do FreeBSD na web, visite https://www.FreeBSD.org/ e muitos mirrors ao redor do mundo. O site é tipicamente o primeiro contato de um usuário novo com o FreeBSD.

As equipes de tradução são responsáveis por traduzir o manual e o site para diferentes idiomas. As páginas do manual não são traduzidas no momento.

Código fonte do site do FreeBSD, Handbook, e FAQ estão disponíveis no repositório de documentação em `https://cgit.freebsd.org/doc/`.

Código fonte das páginas de manual estão disponíveis em um repositório diferente localizado em `https://cgit.freebsd.org/src/`.

As mensagens de commit de documentação podem ser visualizadas com `git log`. As mensagens de commit também são arquivadas em Mensagens de commit de todas as branches do repositório doc.

Endereço web para ambos os repositórios disponíveis em https://cgit.freebsd.org/doc/ e https://cgit.freebsd.org/src/.

Muitas pessoas tem escrito tutoriais e artigos how-to sobre FreeBSD. Alguns são armazenados como parte dos arquivos FDP. Em outros casos, o autor decidiu manter a documentação separada. O FDP esforça-se para fornecer links para o máximo possível dessas documentações externas.

Capítulo 2. Ferramentas

Várias ferramentas são utilizadas para gerenciar a documentação do FreeBSD e renderizá-la para diferentes formatos. Algumas dessas ferramentas são necessárias e devem ser instaladas antes de trabalhar com os exemplos nos capítulos a seguir. Algumas são opcionais, adicionando recursos ou tornando a tarefa de criar documentação mais simples.

2.1. Ferramentas Obrigatórias

Instale o *meta-port* `docproj` como mostrado no capítulo de introdução da coleção de Ports. Estes pacotes são necessários para trabalhar com a documentação do FreeBSD. Informações adicionais específicas de alguns componentes serão informadas abaixo.

2.2. Ferramentas Opcionais

Essas ferramentas não são necessárias, mas podem facilitar o trabalho na documentação ou adicionar recursos.

2.2.1. Software

Vim (editors/vim)

Um editor popular para trabalhar com AsciiDoctor.

Emacs (editors/emacs)

Ambos editores incluem um modo especial para editar documentos. Esse modo inclui comandos para reduzir a quantidade de digitação necessária e ajudar a reduzir a possibilidade de erros.

Capítulo 3. A Área de Trabalho

A *área de trabalho* é uma cópia da árvore de documentação do repositório do FreeBSD baixada no computador local. As alterações são feitas na cópia de trabalho local, testadas e enviadas como patches para serem submetidas no repositório principal.

Uma cópia completa da árvore de documentação pode ocupar 700 megabytes de espaço em disco. Tenha um gigabyte de espaço total para ter sobra para arquivos temporários e versões de teste dos diversos formatos de saída.

O Git é utilizado para gerenciar os arquivos de documentação do FreeBSD. Ele é obtido pela instalação do pacote devel/git, e também por uma versão mais leve chamado git-lite:

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

3.1. Documentação e Páginas de Manual

A documentação do FreeBSD não é formada apenas por livros e artigos. Páginas de manual para todos os comandos e arquivos de configuração também fazem parte da documentação e fazem parte do território do FDP. Dois repositórios estão envolvidos: `doc` para os livros e artigos, e `src` para o sistema operacional e páginas de manual. Para editar páginas de manual, o repositório `src` deve ser obtido separadamente.

Repositórios podem conter várias versões de documentação e código-fonte. Novas modificações quase sempre são feitas apenas para a versão mais recente, chamada `main`.

3.2. Escolhendo um Diretório

A documentação do FreeBSD é tradicionalmente armazenada em /usr/doc/, e o código fonte do sistema com páginas de manual em /usr/src/. Essas árvores são realocáveis e os usuários podem armazenar as cópias de trabalho em outros locais para evitar interferir nas informações existentes nos diretórios principais. Os exemplos a seguir utilizam ~/doc e ~/src, ambos subdiretórios do diretório pessoal do usuário.

3.3. Baixando uma Cópia

O processo de download de um repositório é chamado de *clone* e é feito com um `git clone`. Este exemplo faz a clonagem de uma cópia da versão mais recente (`main`) do repositório de documentação principal:

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

O checkout do código-fonte para trabalhar nas páginas de manual é muito semelhante:

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

3.4. Atualizando

Os documentos e arquivos no repositório do FreeBSD mudam diariamente. As pessoas modificam arquivos e submetem alterações com frequência. Mesmo após um checkout inicial, já haverá alterações entre a cópia de trabalho local e o repositório principal do FreeBSD. Para atualizar a versão local com as mudanças que foram feitas no repositório principal, execute `git pull` no diretório que contém a cópia de trabalho local:

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

Adquira o hábito protetor de usar `git pull` antes de editar arquivos de documentos. Alguém pode ter editado aquele arquivo recentemente, e a cópia de trabalho local não incluirá as últimas mudanças até que tenha sido atualizada. Editar a versão mais recente de um arquivo é muito mais fácil do que tentar combinar um arquivo local editado mais antigo com a versão mais recente do repositório.

3.5. Revertendo Alterações

De vez em quando acontece que algumas mudanças não eram necessárias, ou o escritor só quer começar novamente. Arquivos podem ser "resetados" para sua forma anterior com `git restore`. Por exemplo, para apagar as alterações feitas no _index.adoc e redefini-las para o formato sem modificação:

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

3.6. Criando um Diff

Após finalizar as alterações em um arquivo ou grupo de arquivos, as diferenças entre a cópia de trabalho local e a versão no repositório do FreeBSD devem ser coletadas em um único arquivo para ser submetido. Estes arquivos diff são produzidos redirecionando a saída de `git diff` para um arquivo:

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

Dê ao arquivo um nome significativo que identifique o conteúdo. O exemplo acima é para correção ortográfica em toda a árvore de documentação.

Se o arquivo diff for enviado com a interface web "Submit a FreeBSD problem report", adicione uma extensão .txt para que o formulário web identifique que o conteúdo do arquivo é texto plano.

Tenha cuidado: `git diff` inclui todas as alterações feitas no diretório atual e em quaisquer subdiretórios. Se houver arquivos na cópia de trabalho com edições que ainda não estão prontas para serem enviadas, forneça uma lista apenas dos arquivos a serem incluídos:

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

3.7. Referências Git

Estes exemplos demonstram um uso muito básico do Git. Mais detalhes estão disponíveis no Git Book e na Documentação do Git.

Capítulo 4. Estrutura de Diretórios da Documentação

Arquivos e diretórios no repositório **doc**/ seguem uma estrutura destinada a:

1. Facilitar a conversão do documento para outros formatos.
2. Promover a consistência entre as diferentes organizações de documentação, e assim facilitar a alternância entre diferentes documentos.
3. Facilitar a decisão de onde a nova documentação deve ser colocada.

Além disso, o repositório de documentação deve acomodar documentos em vários idiomas e codificações diferentes. É importante que a estrutura do repositório de documentação não imponha quaisquer padrões particulares ou preferências culturais.

4.1. O Nível Superior, doc/

Existem dois tipos de diretório em **doc**/, documentation e website, ambos compartilham a mesma estrutura.

Diretório	Uso
documentation	Contém todos os artigos e livros em formato AsciiDoc. Contém subdiretórios para categorizar ainda mais as informações por idiomas.
tools	Contém um conjunto de ferramentas usadas para traduzir a documentação e o site usando Weblate. A instância do Weblate pode ser acessada aqui.
shared	Contém arquivos que não são específicos para as várias traduções da documentação. Contém subdiretórios para categorizar ainda mais as informações por idiomas e três arquivos para armazenar as informações dos autores, lançamentos e espelhos. Este diretório é compartilhado entre `documentation` e o `website`.
website	Contém o site do FreeBSD no formato AsciiDoc Contém subdiretórios para categorizar ainda mais as informações por idiomas.

4.2. Os Diretórios

Esses diretórios contêm a documentação e o website. A documentação está organizada em subdiretórios abaixo deste nível, seguindo o estrutura de diretórios Hugo.

Diretório	Uso
archetypes	Contém templates para criar novos artigos, livros e páginas web. Para mais informações, veja aqui.
config	Contém os arquivos de configuração do Hugo. Um arquivo principal e um arquivo por idioma. Para mais informações, veja aqui.

Diretório	Uso
content	Contém os livros, artigos e páginas web. Existe um diretório para cada tradução disponível da documentação, por exemplo `en` e `zh-tw`.
data	Contem dados personalizados para compilar o site no formato TOML. Este diretório é usado para armazenar os eventos, notícias, imprensa, etc. Para mais informações, veja aqui.
static	Contem ativos estáticos. Imagens, avisos de segurança, pgpkeys, etc. Para mais informações, veja aqui.
themes	Contém os modelos na forma de arquivos `.html` que especificam a aparência do site. Para mais informações, veja aqui.
tools	Contém ferramentas usadas para aprimorar a construção da documentação. Por exemplo, para gerar o índice dos livros, etc.
beastie.png	Esta imagem não precisa de introdução ;)
LICENSE	Licença da documentação e site. Licença BSD de 2 cláusulas.
Makefile	O **Makefile** que executa o processo de compilação da documentação e do website.

4.3. Informação Específica de Documentação

Esta seção contém informações específicas sobre documentos gerenciados pelo FDP.

4.4. Os Livros: books/

Os livros são escritos em AsciiDoc.

Para cada livro do FreeBSD, o tipo de documento AsciiDoc (também conhecido como doctype) é `book`. Os livros possuem `part`es, cada uma contendo vários `capítulos` (chapter).

Quando o documento é convertido para HTML 5 (usando o backend `html5` embutido):

- A seção AsciiDoc nível 0 (`=`) no início de um `capítulo` de um `livro` será `<h1>`
- A seção AsciiDoc nível 1 (`==`) deve ser usada para a primeira seção lógica de um capítulo e será `<h2>`
- A seção AsciiDoc nível 2 (`===`) deve ser usada para a primeira subseção lógica e será `<h3>`

– e assim por diante. Referência: Títulos e Níveis de Seção | Asciidoctor Docs.

4.4.1. Organização Física

Existem vários arquivos e diretórios no diretório books, todos com a mesma estrutura.

4.4.1.1. _index.adoc

O arquivo **_index.adoc** define algumas variáveis AsciiDoc que afetam como o código AsciiDoc é

convertido para outros formatos e lista o Índice, a Tabela de Exemplos, a Tabela de Figuras, a Tabela de Tabelas e a seção de resumo.

4.4.1.2. book.adoc

O arquivo **book.adoc** define algumas variáveis AsciiDoc que afetam como o código AsciiDoc é convertido para outros formatos e lista o Índice, a Tabela de Exemplos, a Tabela de Figuras, a Tabela de Tabelas, a seção de resumo e todos os capítulos. Este arquivo é usado para gerar o PDF com `asciidoctor-pdf` e para gerar o livro em uma página `html`.

4.4.1.3. part*.adoc

Os arquivos **part*.adoc** armazenam uma breve introdução de uma parte do livro.

4.4.1.4. directory/_index.adoc

Cada capítulo do Handbook é armazenado em um arquivo chamado **_index.adoc** em um diretório separado dos outros capítulos.

Por exemplo, este é um exemplo do cabeçalho de um capítulo:

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

[[kernelconfig]]
= Configurando o kernel do FreeBSD
...
```

Quando a versão HTML5 do Handbook é produzida, será gerado o **kernelconfig/index.html**.

Uma breve olhada mostrará que existem muitos diretórios com arquivos **_index.adoc** individuais, incluindo **basics/_index.adoc**, **introduction/_index.adoc**, e **printing/_index.xml**.

> Não nomeie capítulos ou diretórios com a ordenação do Handbook. Essa ordenação pode mudar conforme o conteúdo do Handbook é reorganizado. A reorganização deve ser realizada sem renomear arquivos, a menos que capítulos inteiros sejam promovidos ou rebaixados dentro da hierarquia.

4.5. Os Artigos: articles/

Os artigos são escritos em AsciiDoc.

Os artigos são organizados como um `artigo` do AsciiDoc. Os artigos são divididos em seções (`=`) e subseções (`==`, `===`) e assim por diante.

4.5.1. Organização Física

Existe um arquivo **_index.adoc** por artigo.

4.5.1.1. _index.adoc

O arquivo **_index.adoc** contém todas as variáveis AsciiDoc e o conteúdo.

Por exemplo, este é um exemplo de um artigo, a estrutura é muito semelhante a um capítulo de livro:

```
---
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"]
---

= Por que você deve usar uma licença de estilo BSD em seu Projeto Open Source
:doctype: article
:toc: macro
:toclevels: 1
:icons: font
:sectnums:
:sectnumlevels: 6
:source-highlighter: rouge
:experimental:

'''

toc::[]

[[intro]]
== Introdução
```

Capítulo 5. Processo de Compilação da Documentação do FreeBSD

Este capítulo cobre a organização do processo de compilação da documentação e como o make(1) é usado para controlá-lo.

5.1. Renderizando AsciiDoc

Diferentes tipos de formato podem ser gerados a partir de um único arquivo AsciiDoc.

Formatos	Tipo de Arquivo	Descrição
`html`	HTML	Um capítulo de `artigo` ou `livro`.
`pdf`	PDF	Portable Document Format.
`epub`	EPUB	Electronic Publication. ePub file format.

5.1.1. Renderizando em html

Para renderizar a documentação e o site em `html`, use um dos seguintes exemplos.

Exemplo 2. Compile a documentação

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

Exemplo 3. Compile o website

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

Exemplo 4. Compile todo o projeto de documentação

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

Exemplos avançados de compilação são apresentados abaixo:

Exemplo 5. Compile a documentação em Inglês e Espanhol com mensagens de debug e no modo verboso

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

```
% make DOC_LANG="en es" HUGO_ARGS="--verbose --debug"
```

Exemplo 6. Compile e sirva o conteúdo com o servidor web interno do Hugo

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

Este servidor web roda em `localhost`, porta `1313` por padrão.

Para servir o conteúdo com o servidor web interno do Hugo em um endereço IP e porta específicos:

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

Um `hostname` também pode ser definido como url base para o servidor web interno do Hugo:

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

Exemplo 7. Compile a documentação em html para uso offline

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

Para compactar a saída em html, adicione `DOC_HTML_ARCHIVE=1`:

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

5.1.2. Renderizando em pdf

Para gerar a documentação em `pdf`, use um dos seguintes exemplos.

Exemplo 8. Compilar todos os documentos em pdf

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

Exemplo 9. Compilar todos os artigos em pdf

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

Exemplo 10. Compilar todos os livros em pdf

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

Exemplo 11. Compilar todos os documentos em pdf de um idioma em específico

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

Este comando irá compilar todos os documentos em pdf do idioma Inglês.

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

Este comando irá compilar todos os livros em pdf dos idiomas Inglês e Francês.

5.2. O Conjunto de Ferramentas de Compilação da Documentação do FreeBSD

Essas são as ferramentas usadas para compilar e instalar a documentação FDP.

- A principal ferramenta de compilação é o make(1), especificamente o Berkeley Make.
- Hugo
- AsciiDoctor
- Git

5.3. Compreendendo o Makefile na Árvore de Documentação

Existem três arquivos Makefile para compilar em parte ou todo o projeto de documentação.

- O Makefile no diretório documentation irá compilar apenas a documentação.
- O Makefile no diretório website irá compilar apenas a website.

- O Makefile na raíz irá compilar a documentação e o site.

O Makefile dos subdiretórios também suportam `make run` para servir o conteúdo compilado com o servidor web interno do Hugo. Este servidor web roda na porta 1313 por padrão.

5.3.1. Makefile da Documentação

Este Makefile suporta o seguinte:

```
# Generate the FreeBSD documentation
#
# Copyright (c) 2020-2021, The FreeBSD Documentation Project
# Copyright (c) 2020-2021, 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 A flag `MAINTAINER` especifica quem é o mantenedor deste Makefile.

② A flag `ALL_LANGUAGES` especifica em quais idiomas o índice deve ser gerado.

③ A flag `RUBY_CMD` especifica a localização do binário Ruby.

④ A flag `HUGO_CMD` especifica a localização do binário Hugo.

⑤ As diretivas `.ORDER` são usadas para garantir que vários makes possam ser executados sem problemas.

⑥ O target `all` gera os índices dos livros ("TOCs"), compila a documentação e coloca o resultado em **~/doc/documentation/public**.

⑦ `starting-message` mostra uma mensagem no console para mostrar ao usuário que o processo está em execução.

⑧ `run-local` executa o servidor web hugo na porta 1313, ou uma porta livre aleatória se esta já estiver em uso.

⑨ `build` compila a documentação e coloca o resultado em **~/doc/documentation/public**.

5.3.2. Makefile do Website

Este é o Makefile:

```
# Generate the FreeBSD website
#
# Copyright (c) 2020-2021, The FreeBSD Documentation Project
# Copyright (c) 2020-2021, 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 "----------------------------------------------------------------"

starting-message: .PHONY ⑦
    @echo ----------------------------------------------------------------
    @echo                   Building the website
    @echo ----------------------------------------------------------------

generate-releases: data/releases.toml ⑧

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

run-local: .PHONY ⑨
    HUGO_DISABLELANGUAGES="${SKIP_LANGS}" ${HUGO_CMD} server \
        ${HUGO_ARGS} -D $(BIND:D--bind=$(BIND)) --baseURL="http://$(.HOST):1313"

build: .PHONY ⑩
    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 A flag `MAINTAINER` especifica quem é o mantenedor deste Makefile.

② A flag `RUBY_CMD` especifica a localização do binário Ruby.

③ A flag `HUGO_CMD` especifica a localização do binário Hugo.

④ As diretivas `.ORDER` são usadas para garantir que vários makes possam ser executados sem problemas.

⑤ O target `all` compila o website e coloca o resultado em **~/doc/website/public**.

⑥ `starting-message` mostra uma mensagem no console para mostrar ao usuário que o processo está em execução.

⑦ `generate-releases` chama o script usado para converter as variáveis AsciiDoc em variáveis TOML. Com esta conversão, as variáveis de releases podem ser utilizadas no AsciiDoc e nos templates personalizados do Hugo.

⑧ `run-local` executa o servidor web hugo na porta 1313, ou uma porta livre aleatória se esta já estiver em uso.

⑨ `build` compila o website e coloca o resultado em **~/doc/website/public**.

Capítulo 6. Primer Asciidoctor

A maioria das documentações do FDP é escrita em AsciiDoc. Este capítulo explica o que isso significa, como ler e entender o código da documentação e as técnicas usadas. Para obter uma referência completa dos recursos do Asciidoctor, consulte a documentação do Asciidoctor. Alguns exemplos usados neste capítulo foram retirados da Referência rápida de sintaxe AsciiDoc.

6.1. Visão geral

Nos primórdios da era computacional, o texto eletrônico era simples. Havia poucos conjuntos de caracteres como ASCII ou EBCDIC, e apenas isso. Texto era texto, e o que você lia era realmente o texto que você tinha. Sem frescuras, sem formatação, sem inteligência.

Inevitavelmente, isso não era suficiente. Quando o texto está em um formato utilizável por computadores, espera-se que eles possam usá-lo e manipulá-lo de maneira inteligente. Os autores querem indicar que certas frases devem ser enfatizadas, adicionadas a um glossário ou transformadas em hiperlinks. Os nomes dos arquivos podem ser apresentados em uma fonte de estilo "typewriter" para exibição na tela do computador, ou como "itálico" quando impressos, ou qualquer outra opção dentre uma infinidade de opções para apresentação.

Esperava-se que a Inteligência Artificial (IA) facilitasse isso. O computador leria o documento e identificaria automaticamente frases-chave, nomes de arquivos, textos que o leitor deveria digitar, exemplos e outros tipos. Infelizmente, na vida real não foi dessa forma, e os computadores ainda precisam de assistência antes que possam processar o texto de maneira significativa.

Mais precisamente, eles precisam de ajuda para identificar o que é o quê. Considere este texto:

Para remover /tmp/foo, utilize o rm(1).

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

É fácil identificar quais partes são nomes de arquivos, quais são comandos a serem digitados, quais partes são referências a páginas de manual e assim por diante. Mas o computador que processa o documento não consegue. Para isso, precisamos utilizar markup.

O exemplo anterior é representado neste documento da seguinte forma:

```
Para remover */tmp/foo*, utilize o man:rm[1].

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

6.2. Cabeçalhos

Asciidoctor suporta seis níveis de cabeçalhos. Se o tipo de documento for `artigo`, apenas um nível 0 (`=`) pode ser usado. Se o tipo de documento for `livro`, pode haver vários títulos de nível 0 (`=`).

Estes são exemplos de cabeçalhos em um `artigo`.

```
= Título do Documento (Nível 0)

== Título da Seção de Nível 1

=== Título da Seção de Nível 2

==== Título da Seção de Nível 3

===== Título da Seção de Nível 4

====== Título da Seção de Nível 5

== Outro Título de Seção de Nível 1
```

Os níveis de seção não podem ser ignorados ao aninhar seções.

A sintaxe a seguir não está correta.

```
= Título do Documento

== Nível 1

==== Nível 3
```

6.3. Parágrafos

Os parágrafos não precisam de marcação especial no AsciiDoc. Um parágrafo é definido por uma ou mais linhas consecutivas de texto. Para criar um novo parágrafo, deixe uma linha em branco.

Por exemplo, este é um título com dois parágrafos.

```
= Este é o título

Este é o primeiro parágrafo. Este também é o primeiro parágrafo.

E este é o segundo parágrafo.
```

6.4. Listas

Asciidoctor suporta alguns tipos de listas, as mais comuns são `ordenadas` e` não ordenadas`. Para obter mais informações sobre listas, consulte o Referência Rápida da Sintaxe AsciiDoc.

6.4.1. Listas ordenadas

Para criar uma lista ordenada, use o caractere `.`.

Por exemplo, esta é uma lista ordenada.

```
. Primeiro item
. Segundo item
.. Sub-segundo item
. Terceiro item
```

E isso seria processado como.

1. Primeiro item
2. Segundo item
   a. Sub-segundo item
3. Terceiro item

6.4.2. Listas não ordenadas

Para criar uma lista não ordenada, use o caractere `*`.

Por exemplo, esta é uma lista não ordenada.

```
* Primeiro item
* Segundo item
** Sub-segundo item
* Terceiro item
```

E isso seria processado como.

- Primeiro item
- Segundo item
  - Sub-segundo item
- Terceiro item

6.5. Links

6.5.1. Links externos

Para apontar para outro site, a macro `link` deve ser usada.

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

> Como a documentação do Asciidoctor descreve, a macro `link` não é necessária quando o link começa com um esquema de URL como `https`. No entanto, é uma boa prática fazer o uso assim para garantir que o Asciidoctor renderize o link corretamente, especialmente em idiomas não latinos como o Japonês.

6.5.2. Links para outro livro ou artigo

Para apontar para outro livro ou artigo, as variáveis Asciidoctor devem ser usadas. Por exemplo, se estamos no artigo `cups` e queremos apontar para `ipsec-must`, esses passos devem ser usados.

1. Inclua o arquivo urls.adoc da pasta ~/doc/shared.

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

2. Em seguida, crie um link usando a variável Asciidoctor para o artigo `ipsec-must`.

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

   E isso seria processado como.

   IPSec-Must article

   > A macro `extref` é definida como uma extensão. Ela é projetada para renderizar o link correto entre as diferentes saídas

6.5.3. Links para o mesmo arquivo ou para outro arquivo no mesmo livro

Os livros são estruturados em diretórios diferentes para manter um layout sensato. Para criar um link de um subdiretório de um livro para outro subdiretório do mesmo livro, use a macro `crossref`:

```
crossref:doc-build[documentation-makefile]
```

E isso seria renderizado como

Este link

> A macro `crossref` é definida como uma extensão. Ela é projetada para renderizar o link correto entre as diferentes saídas

Use a macro `crossref` para links intra-documento também. Embora possa ser inconveniente escrever o nome do documento atual, ela garante que o link correto seja renderizado entre as diferentes saídas

Não use nem a macro `xref` nem seu atalho `<< >>`. Eles não funcionam bem em todos os formatos de saída.

6.6. Imagens e Ícones

Imagens e ícones desempenham um papel crucial na melhoria da experiência geral do usuário. Esses elementos visuais são estrategicamente integrados para transmitir informações, esclarecer conceitos e fornecer uma interface visualmente envolvente.

6.6.1. Imagens

As imagens ajudam a ilustrar conceitos complexos, tornando-os mais acessíveis aos usuários.

O primeiro passo será adicionar a imagem no diretório de imagens no caminho:

- ~/website/static/images/ para o website.
- ~/documentation/static/images/ para a documentação.

Por exemplo, para adicionar uma nova imagem ao processo de instalação do FreeBSD, a imagem será salva no caminho ~/documentation/static/images/books/handbook/bsdinstall/new-image3.png.

O próximo passo será configurar os atributos do Asciidoctor `images-path` e `imagesdir`.

Usaremos como exemplo o cabeçalho do artigo Engenharia de Releases do FreeBSD.

```
= Engenharia de Release do FreeBSD
:doctype: article

[...]

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


[...]
```

① Faz referência ao caminho dentro da pasta /static/images.

② Faz referência ao atributo Asciidoctor.

Uma vez que a imagem esteja no caminho correto e os atributos do Asciidoctor tenham sido configurados no documento, a macro `image` pode ser usada.

Este é um exemplo:

```
image::new-image3.png[Nova etapa no processo de instalação do FreeBSD]
```

Para melhorar a acessibilidade, é obrigatório adicionar texto descritivo a cada imagem.

6.6.2. Ícones

Os ícones servem como símbolos intuitivos para reconhecimento e navegação rápidos.

O primeiro passo para usar ícones é adicionar a propriedade `icons` à seção de propriedades do Asciidoctor, no topo de cada documento.

```
:icons: font
```

Depois que a propriedade do ícone do Asciidoctor for definida, um ícone suportado pela Font Awesome pode ser adicionado.

Este é um exemplo de como usar o ícone `envelope`:

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

Para melhorar a acessibilidade do site, o atributo `title` é obrigatório.

6.7. Conclusão

Esta é a conclusão deste primer do Asciidoctor. Por razões de espaço e complexidade, várias assuntos não foram abordadas em profundidade (ou completamente).

Capítulo 7. Rosetta Stone

7.1. Comparação entre Docbook e AsciiDoc

Esta rosetta stone tenta mostrar as diferenças entre Docbook e AsciiDoc.

Tabela 1. Comparação entre Docbook e AsciiDoc

Recurso da Linguagem	Docbook	AsciiDoc
Negrito	bold	*bold*
Itálico	<emphasis>Italic</emphasis>	_Italic_
Monoespaço	<literal>Monospace</literal>	`Monospace`
Parágrafo	<para>This is a paragraph</para>	This is a paragraph
Teclado	<keycap>F11</keycap>	kbd:[F11]
Links	`<link xlink:href="https://www.freebsd.org/where/">Download FreeBSD</link>`	`link:https://www.freebsd.org/where/[Download FreeBSD]`

Seções	`<sect1 xml:id="id">`
	`  <title>Section 1</title>`
	`</sect1>`
		`[[id]]`
		`= Section 1`

Lista não ordenada	`<itemizedlist>`
	`  <listitem>`
	`  <para>When to build a custom kernel.</para>`
	`  </listitem>`
	
	`  <listitem>`
	`  <para>How to take a hardware inventory.</para>`
	`  </listitem>`
	`</itemizedlist>`
		`* When to build a custom kernel.`
		`* How to take a hardware inventory.`

Recurso da Linguagem	Docbook	AsciiDoc
Lista ordenada	<orderedlist> <listitem> <para>One</para> </listitem> <listitem> <para>Two</para> </listitem> <listitem> <para>Three</para> </listitem> <listitem> <para>Four</para> </listitem> </orderedlist>	. One . Two . Three . Four
Lista de variáveis	<variablelist> <varlistentry> <term>amd64</term> <listitem> <para>This is the most common desktop...</para> </listitem> </varlistentry> </variablelist>	amd64:: This is the most common desktop...
Código fonte	<screen> &prompt.root; <userinput>mkdir -p /var/spool/lpd/lp</userinput> </screen>	[source,shell] ---- # mkdir -p /var/spool/lpd/lp ----
Bloco literal	<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 ....

Recurso da Linguagem	Docbook	AsciiDoc

Imagens	<figure xml:id="bsdinstall-newboot-loader-menu">
	  <title>FreeBSD Boot Loader Menu</title>
	
	  <mediaobject>
	  <imageobject>
	  <imagedata fileref="bsdinstall/bsdinstall-newboot-loader-menu"/>
	  </imageobject>
	  <textobject>
	  </literallayout>ASCII art replacement is no longer supported.</literallayout>
	  </textobject>
	  <textobject>
	  <phrase>The FreeBSD loader menu, with options 1-6 to boot
	  multi-user, boot single user, escape to loader prompt, reboot,
	  select a kernel to load, and select boot options</phrase>
	  </textobject>
	  </mediaobject>
	</figure>
		[[bsdinstall-newboot-loader-menu]]
		.FreeBSD Boot Loader Menu
		image::bsdinstall/bsdinstall-newboot-loader-menu[The FreeBSD loader menu, with options 1-6 to boot multi-user, boot single user, escape to loader prompt, reboot, select a kernel to load, and select boot options]

Includes	n/a	include::chapter.adoc[]

Recurso da Linguagem	Docbook	AsciiDoc
Tabelas	<table xml:id="partition-schemes" frame="none" rowsep="1" pgwide="1"> <title>Partitioning Schemes</title> <tgroup cols="2" align="left"> <thead> <row> <entry align="left" >Abbreviation</entry> <entry align="left" >Description</entry> </row> </thead> <tbody> <row> <entry>APM</entry> <entry>Apple Partition Map, used by PowerPC(R).</entry> </row> </tbody> </tgroup> </table>	[[partition-schemes]] .Partitioning Schemes [cols="1,1", frame="none", options="header"] |=== | Abbreviation | Description |APM |Apple Partition Map, used by PowerPC(R). |===
Advertências	<tip> <para>This is a tip</para> </tip>	[TIP] ==== This is a tip ====

Capítulo 8. Traduções

Este é o FAQ para pessoas que estão traduzindo a documentação do FreeBSD (FAQ, Handbook, tutoriais, páginas de manual e outros) para diferentes idiomas.

Ele é *fortemente* baseado na tradução do FAQ do Projeto de Documentação Alemão do FreeBSD, originalmente escrito por Frank Gründer elwood@mc5sys.in-berlin.de e traduzido de volta para o Inglês por Bernd Warken bwarken@mayn.de.

8.1. O que significa i18n e l10n?

i18n significa internacionalização e l10n significa localização. São apenas uma abreviação.

i18n pode ser lido como "i" seguido por 18 letras, seguido por "n". Da mesma forma, l10n é "l" seguido por 10 letras, seguido por "n".

8.2. Existe uma lista de discussão para tradutores?

Sim. Diferentes grupos de tradução têm suas próprias listas de discussão. A lista dos projetos de tradução possui mais informações sobre as listas de discussão e sites web mantidos por cada projeto de tradução. Além disso, existe a freebsd-translators@freebsd.org para discussão geral de tradução.

8.3. São necessários mais tradutores?

Sim. Quanto mais pessoas trabalham na tradução, mais rápido ela será finalizada, e mais rapidamente as mudanças na documentação em Inglês serão refletidas nos documentos traduzidos.

Você não precisa ser um tradutor profissional para poder ajudar.

8.4. Quais idiomas eu preciso saber?

Idealmente, você deverá possuir um bom conhecimento de Inglês escrito, e obviamente, precisará ser fluente no idioma para o qual está traduzindo.

Inglês não é estritamente necessário. Por exemplo, você poderia fazer uma tradução Húngara do FAQ da tradução em Espanhol.

8.5. Quais softwares eu preciso conhecer?

É fortemente recomendado que você mantenha uma cópia local do repositório Git do FreeBSD (pelo menos a parte da documentação). Isso pode ser feito executando:

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

git.FreeBSD.org é um servidor público `git`.

Será necessário que o pacote git-lite esteja instalado.

Você deverá ter conhecimentos básicos de git. Ele permitirá que você veja o que mudou entre as diferentes versões dos arquivos que compõem a documentação.

Por exemplo, para ver as diferenças entre as revisões `abff932fe8` e `2191c44469` de documentation/content/en/articles/committers-guide/_index.adoc, execute:

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

Por favor, veja a explicação completa de como usar o Git no FreeBSD no FreeBSD Handbook.

8.6. Como eu faço para descobrir se já existem outras pessoas traduzindo documentos para o meu idioma?

A página do Projeto de Tradução da Documentação lista os trabalhos de tradução que são conhecidos. Se outras pessoas já estiverem trabalhando na tradução de documentação para o seu idioma, por favor, não duplique os esforços. Em vez disso, entre em contato para saber como você pode ajudar.

Se não existir nenhum projeto de tradução para o seu idioma listado nesta página, envie uma mensagem para a lista de discussão do projeto de documentação do FreeBSD para o caso de alguém estar pensando em fazer a tradução, mas ainda não tiver anunciado nada.

8.7. Ninguém mais está traduzindo para o meu idioma. O que eu faço?

Parabéns, você acabou de iniciar o "Projeto de Tradução da Documentação do FreeBSD em seu idioma aqui". Bem vindo a bordo.

Primeiro, pense se você terá o tempo necessário. Uma vez que você é a única pessoa trabalhando no seu idioma no momento, será sua a responsabilidade de publicar o seu trabalho e coordenar qualquer voluntário que queira ajudá-lo.

Escreva um email para a lista de discussão do Projeto de Documentação, anunciando que você irá traduzir a documentação, assim a página do Projeto de Traduções de Documentação poderá ser atualizada.

Se já existir alguém em seu país provendo o espelhamento de serviços do FreeBSD, você deve contacta-lo e perguntar se você pode ter algum espaço web para seu projeto, e se possível um endereço de email ou mesmo um serviço de lista de discussão.

Então escolha um documento e comece a traduzir. É melhor começar com algo razoavelmente pequeno—como o FAQ ou um dos tutoriais.

8.8. Eu já tenho alguns documentos traduzidos, para onde eu devo enviá-los?

Isso depende. Se você já está trabalhando com uma equipe de tradução (tal como a equipe Japonesa, ou a equipe Alemã) então ela terá seus próprios procedimentos para manipular a documentação submetida, e estes serão descritos em seus web sites.

Se você for a única pessoa trabalhando em um determinado idioma (ou se você é o responsável pelo projeto de tradução e quer submeter suas mudanças de volta para o projeto FreeBSD) então você deve enviar sua tradução ao Projeto FreBSD (veja pergunta seguinte).

8.9. Eu sou a única pessoa trabalhando na tradução para este idioma, como faço para enviar minha tradução?

Primeiro, verifique se sua tradução está organizada corretamente. Isso significa que ele deve cair na árvore de documentação existente e ser compilada de maneira correta.

Os diretórios abaixo desse são nomeados de acordo com o código do idioma em que estão escritos, conforme definido na ISO639 (/usr/share/misc/iso639 em uma versão do FreeBSD mais recente que 20 de janeiro de 1999).

Hugo precisa dos códigos de idioma em letras minúsculas. Por exemplo, em vez de `pt_BR`, Hugo utiliza `pt-br`.

Atualmente a documentação do FreeBSD é armazenada em um diretório de nível superior chamado documentation/. Os diretórios abaixo desse são nomeados de acordo com o código do idioma em que estão escritos, conforme definido na ISO639 (/usr/share/misc/iso639 em uma versão do FreeBSD mais recente que 20 de janeiro de 1999).

Se o seu idioma puder ser codificado de maneiras diferentes (por exemplo, Chinês), deve haver diretórios abaixo desse, um para cada formato de codificação fornecido.

Finalmente, você deve ter diretórios para cada documento.

Por exemplo, em uma hipotética tradução Sueca ficaria assim:

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

`sv` é o nome da tradução, no formato lang. Repare nos dois Makefiles, que serão usados para compilar a documentação.

Utilize o comando git diff para gerar a alteração e envia-la ao sistema de revisão.

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

Você deve usar o Bugzilla para enviar um relatório indicando que você enviou a documentação. Seria muito útil se você conseguir outras pessoas para checar sua tradução primeiro, já que é improvável que a pessoa que irá fazer o commit seja fluente no idioma.

Alguém (provavelmente o Gerente do Projeto de Documentação, atualmente Equipe de Engenharia de Documentação <doceng@FreeBSD.org>) irá então pegar sua tradução e checar se ela compila. Em particular, os seguintes itens serão analisados:

1. O `make` no diretório root funciona corretamente?

Se houver algum problema, quem estiver validando a submissão irá entrar em contato para que seja feito as correções.

Se não houver problemas, sua tradução será disponibilizada o mais rápido possível.

8.10. Posso incluir um texto específico do idioma ou do país em minha tradução?

Nós preferimos que você não faça isso.

Por exemplo, suponha que você esteja traduzindo o Handbook para o Coreano e queira incluir uma seção sobre varejistas na Coreia em seu Handbook.

Não há razão pela qual esta informação não deva estar nas versões em Inglês (ou Alemão, ou Espanhol, ou Japonês, ou ...). É possível que uma pessoa que fale Inglês na Coréia possa tentar obter uma cópia do FreeBSD enquanto esteja ali. Isso também ajuda a aumentar a presença perceptível do FreeBSD ao redor do mundo, o que não é uma coisa ruim.

Se você tiver informações específicas do país, submeta-as como uma alteração do Handbook em Inglês (usando o Bugzilla) e em seguida, traduza a alteração de volta para o seu idioma no Handbook traduzido.

Obrigado.

8.10.1. Dirigindo-se ao leitor

Nos documentos em Inglês, o leitor é tratado por "você", não há distinção entre formal/informal como existe em alguns idiomas.

Se você estiver traduzindo para um idioma que tenha esta distinção, use qualquer forma que normalmente é usada em outras documentações técnicas. Na dúvida, use a forma mais educada.

8.10.2. Preciso incluir informações adicionais nas minhas traduções?

Sim.

O cabeçalho da versão em Inglês de cada documento será algo parecido com isto:

```
 ---
 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
```

O forma exata pode mudar, mas sempre incluirá uma linha $FreeBSD$ e a frase `The FreeBSD Documentation Project`. Note que a parte do $FreeBSD é expandida automaticamente pelo Git, portanto ela deve estar vazia (apenas `$FreeBSD$`) para novos arquivos.

Seus documentos traduzidos devem incluir sua própria linha FreeBSD, e mudar a linha `FreeBSD Documentation Project` para `The FreeBSD language Documentation Project`.

Você deve ainda adicionar uma terceira linha que indicará qual revisão do texto em inglês o texto é baseado.

Assim, a versão em Espanhol desse arquivo pode começar com:

```
 ---
 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
```

Capítulo 9. Traduções PO

9.1. Introdução

O GNU gettext oferece aos tradutores uma maneira fácil de criar e manter traduções de documentos. Sequências traduzíveis são extraídas do documento original em um arquivo PO (Portable Object). Versões traduzidas das strings são inseridas com um editor separado. As strings podem ser usadas diretamente ou incorporadas em uma versão traduzida completa do documento original.

9.2. Introdução

Supõe-se que o procedimento mostrado no Quick Start já tenha sido executado. A opção `TRANSLATOR` é necessária e já está ativada por padrão no port textproc/docproj.

Este exemplo mostra a criação de uma tradução em Espanhol do pequeno artigo Leap Seconds.

Procedimento. Instale um Editor PO

1. É necessário um editor PO para editar os arquivos de tradução. Este exemplo utiliza o editors/poedit.

```
# pkg install poedit
```

Procedimento. Configuração Inicial

Quando uma nova tradução é criada pela primeira vez, a estrutura do diretório e o Makefile devem ser criados ou copiados do original em Inglês:

1. Crie um diretório para a nova tradução. O código fonte do artigo em Inglês está em ~/doc/documentation/content/en/articles/leap-seconds/. A tradução em Espanhol estará em ~/doc/documentation/content/es/articles/leap-seconds/. O caminho é o mesmo, exceto pelo nome do diretório de idiomas.

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

2. Copie o _index.po do documento original para o diretório de tradução:

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

Supondo que o documento ou idioma também esteja sendo traduzido via Weblate. Nesse caso, é bom pegar o arquivo `.po` de lá e carregar o documento de tradução de volta na plataforma, centralizando os esforços de tradução via Weblate para evitar retrabalho.

Veja como baixar os arquivos `.po` no capítulo Traduzindo Offline no Weblate.

Procedimento. Tradução

Use um editor PO para inserir as traduções no arquivo PO. Existem vários editores diferentes disponíveis. O poedit do editors/poedit é mostrado aqui.

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

Procedimento. Gerando um Documento Traduzido

1. Gere o documento traduzido:

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

   O nome do documento gerado corresponde ao nome do original em Inglês, geralmente _index.adoc.

2. Verifique o arquivo gerado renderizando-o para HTML e exibindo-o com um navegador web:

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

9.3. Criando Novas Traduções

O primeiro passo para criar um novo documento traduzido é localizar ou criar um diretório para mantê-lo. O FreeBSD coloca documentos traduzidos em um subdiretório nomeado para seu idioma e região no formato lang . *lang* é um código minúsculo de dois caracteres.

Tabela 2. Nomes de Idioma

Idioma	Região	Nome do Diretório da Tradução
English	United States	en
Bengali	Bangladesh	bn-bd
Danish	Denmark	da
German	Germany	de
Greek	Greece	el
Spanish	Spain	es

Idioma	Região	Nome do Diretório da Tradução
French	France	fr
Hungarian	Hungary	hu
Italian	Italy	it
Japanese	Japan	ja
Korean	Korea	ko
Mongolian	Mongolia	mn
Dutch	Netherlands	nl
Polish	Poland	pl
Portuguese	Brazil	pt-br
Russian	Russia	ru
Turkish	Turkey	tr
Chinese	China	zh-cn
Chinese	Taiwan	zh-tw

As traduções estão em subdiretórios do diretório principal da documentação, aqui assumido como ~/doc/documentation/ como apresentado na Introdução. Por exemplo, as traduções em Alemão estão localizadas em ~/doc/documentation/content/de/ e as traduções em Francês estão em ~/doc/documentation/content/fr/.

Cada diretório de idiomas contém subdiretórios separados para os tipos de documentos, geralmente articles/ e books/.

A combinação desses nomes de diretórios fornece o caminho completo para um artigo ou livro. Por exemplo, a tradução Francesa do artigo NanoBSD está em ~/doc/documentation/content/fr/articles/nanobsd/, e a tradução em Mongol do manual está em ~/doc/documentation/content/mn/books/handbook/.

Um novo diretório de idioma deve ser criado ao traduzir um documento para um novo idioma. Se o diretório de idiomas já existir, somente um subdiretório no diretório articles/ ou books/ será necessário.

Exemplo 12. Criando uma Tradução em Espanhol do Porter's Handbook

Crie uma nova tradução em Espanhol do Porter's Handbook. O original é um livro em ~/doc/documentation/content/en/books/porters-handbook/.

1. O diretório de livros em Espanhol ~/doc/documentation/content/es/books/ já existe, portanto, apenas um novo subdiretório para o Porter's Handbook é necessário:

```
% cd ~/doc/documentation/content/es/books
```

```
% mkdir porters-handbook
```

2. Copie o conteúdo do livro original:

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

Agora a estrutura do documento está pronta para o tradutor começar a tradução com o `poedit`.

9.4. Traduzindo

O sistema gettext reduz bastante o número de itens que devem ser rastreados por um tradutor. As strings a serem traduzidas são extraídas do documento original em um arquivo PO. Em seguida, um editor PO é usado para inserir as traduções de cada string.

O sistema de tradução PO do FreeBSD não sobrescreve os arquivos PO, portanto a etapa de extração pode ser executada a qualquer momento para atualizar o arquivo PO.

Um editor PO é usado para editar o arquivo. editors/poedit é usado nestes exemplos porque é simples e tem requisitos mínimos. Outros editores PO oferecem recursos para facilitar o trabalho de tradução. A Coleção de Ports oferece vários desses editores, incluindo o devel/gtranslator.

É importante preservar o arquivo PO. Ele contém todo o trabalho que os tradutores fizeram.

> Traduzir documentos online tende a ser o método mais fácil para tradução de documentos no FreeBSD, pois permite que vários usuários colaborem no mesmo arquivo, distribuindo a carga de trabalho de forma eficaz. Para obter mais detalhes, consulte o próximo capítulo, Traduções Weblate.

9.5. Dicas para Tradutores

9.5.1. Preservando macros AsciiDoc

Preserve as macros AsciiDoc que são mostradas no original em Inglês.

Exemplo 13. Preservando macros AsciiDoc

Inglês original:

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

Tradução para o Espanhol:

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

9.5.2. Preservando Espaços

Preserve os espaços existentes no início e no final das strings a serem traduzidas. A versão traduzida também deve ter esses espaços.

9.5.3. Tags

O conteúdo de algumas tags devem ser copiadas igualmente, sem realizar tradução:

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

9.6. Compilando um Documento Traduzido

Uma versão traduzida do documento original pode ser criada a qualquer momento. Quaisquer porções não traduzidas do original serão incluídas em Inglês no documento resultante. A maioria dos editores PO tem um indicador que mostra quanto da tradução foi realizada. Isso torna mais fácil para o tradutor ver quantas strings foram traduzidas para tornar a compilação do documento final utilizável.

O capítulo Weblate fornece um exemplo completo sobre Compilando o Documento Traduzido.

9.7. Submetendo a Nova Tradução

Prepare os novos arquivos de tradução para envio. Isso inclui adicionar os arquivos ao sistema de controle de versão, definir propriedades adicionais e criar um diff para a submissão.

Os arquivos diff criados por esses exemplos podem ser anexados a um relatório de bug de documentação ou revisão de código.

Exemplo 14. Tradução Espanhola do Artigo NanoBSD

1. Crie um diff dos novos arquivos a partir do diretório base ~/doc/ para que o caminho completo seja mostrado com os nomes dos arquivos. Isso ajuda os committers a identificar o diretório do idioma de destino.

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

O capítulo Weblate fornece um exemplo completo sobre Enviando Traduções.

Capítulo 10. Traduções Weblate

10.1. Introdução

Este capítulo descreve alguns passos básicos para ingressar na equipe de tradutores do FreeBSD, traduzindo online no Weblate ou offline, e algumas sugestões simples sobre tradução, revisão e teste. O foco é na parte da tradução.

Os documentos originais (artigos e livros) estão no portal de documentação.

Weblate é um software web de código aberto com foco em idiomas; o projeto FreeBSD possui uma instância local.

10.2. Como se Tornar um Tradutor do FreeBSD

A seguir seguem os passos para começar a traduzir artigos e livros do Projeto de Documentação do FreeBSD.

1. Crie uma conta na instância Weblate do FreeBSD com um endereço de e-mail ou sua conta GitHub.
2. Inscreva-se na lista de discussão sobre as traduções do FreeBSD.
3. Apresente-se e peça para participar de uma equipe de idiomas. Se a equipe de idiomas não existir, peça para criá-la. A auto-apresentação é essencial. Isso aumenta suas chances de ser aprovado para acesso de escrita.
4. Acesse o Weblate com a nova conta.
5. Encontre a equipe de idiomas e escolha um documento inicial para traduzir.
6. Crie uma conta no Bugzilla para enviar as traduções depois de terminar um documento. O projeto de documentação também está aceitando Pull Requests do GitHub com envio de traduções.

> Todos os arquivos de tradução e documentos devem seguir a Licença de Documentação do FreeBSD; se isso não for aceitável, por favor, não se inscreva ou envie patches ou traduções.

10.3. Apresente-se

Forneça uma breve auto-apresentação na lista de discussão sobre as traduções do FreeBSD para iniciar o processo de concessão de acesso. Isso permitirá que um coordenador ou administrador de idiomas forneça as permissões necessárias para que o novo usuário do Weblate comece a traduzir.

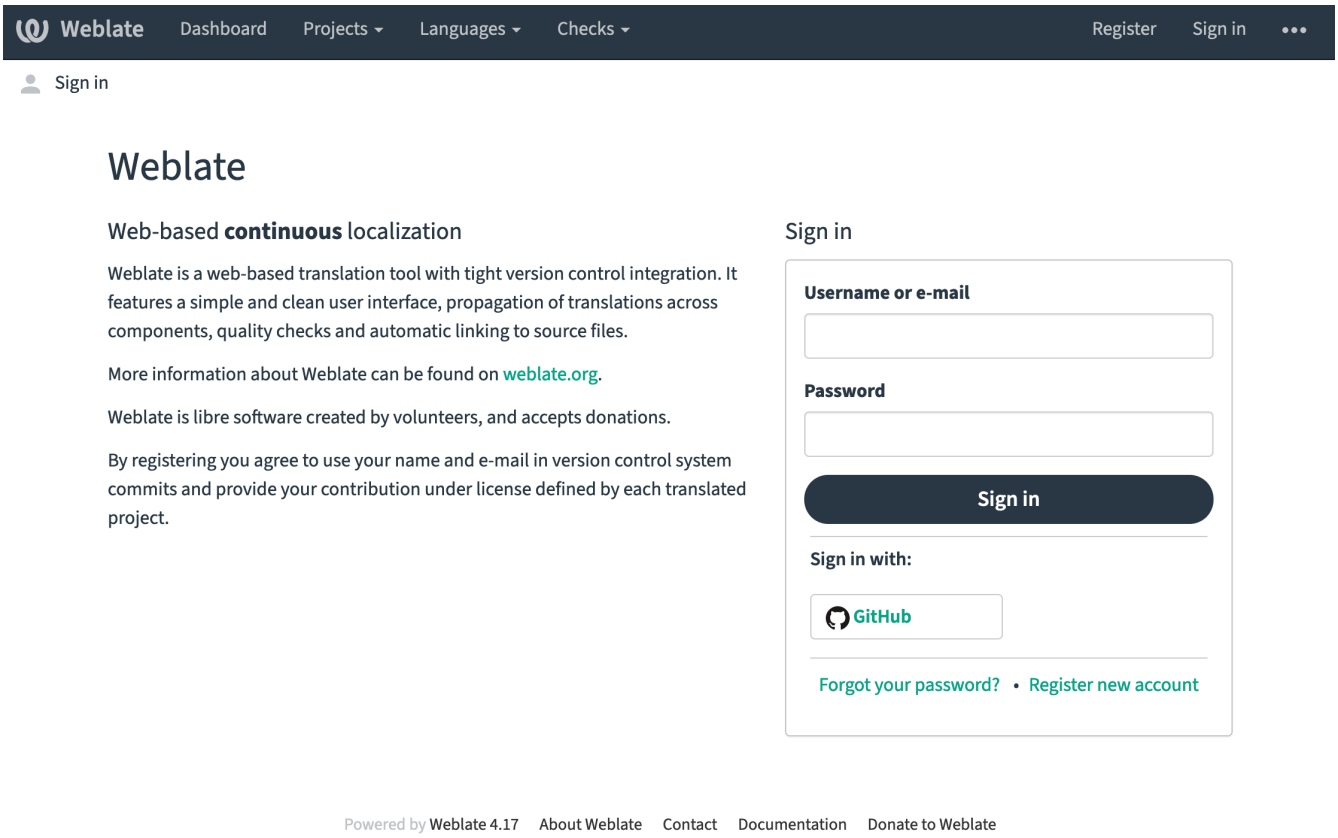
A seguir está um exemplo de como tal e-mail poderia parecer.

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

```
Name:      Name (usar nome preferido)
Location:  City, country (opcional)
Login:     username or email (essencial)
Language:  Language to translate (essencial)
Profession or student status: (opcional)
About You: (formato livre -- informações com as quais você se sente à vontade para
compartilhar
  others: empresa, escola, outra afiliação, habilitações históricas, outras
  projetos em que você trabalhou, nível e tipo de conhecimento de informática, outras
habilidades relevantes,
  etc.)
You and the FreeBSD Project: (formato livre: outros projetos FreeBSD de interesse,
comentários, etc.)
```

10.4. Entrar no Weblate

Abra https://translate-dev.freebsd.org/ e `Entre`.



Use um nome de usuário, endereço de e-mail ou conta do GitHub para fazer login.

O perfil do usuário contém suas preferências, nome e endereço de e-mail. O nome e o endereço serão usados nos commits; mantenha esta informação atualizada.

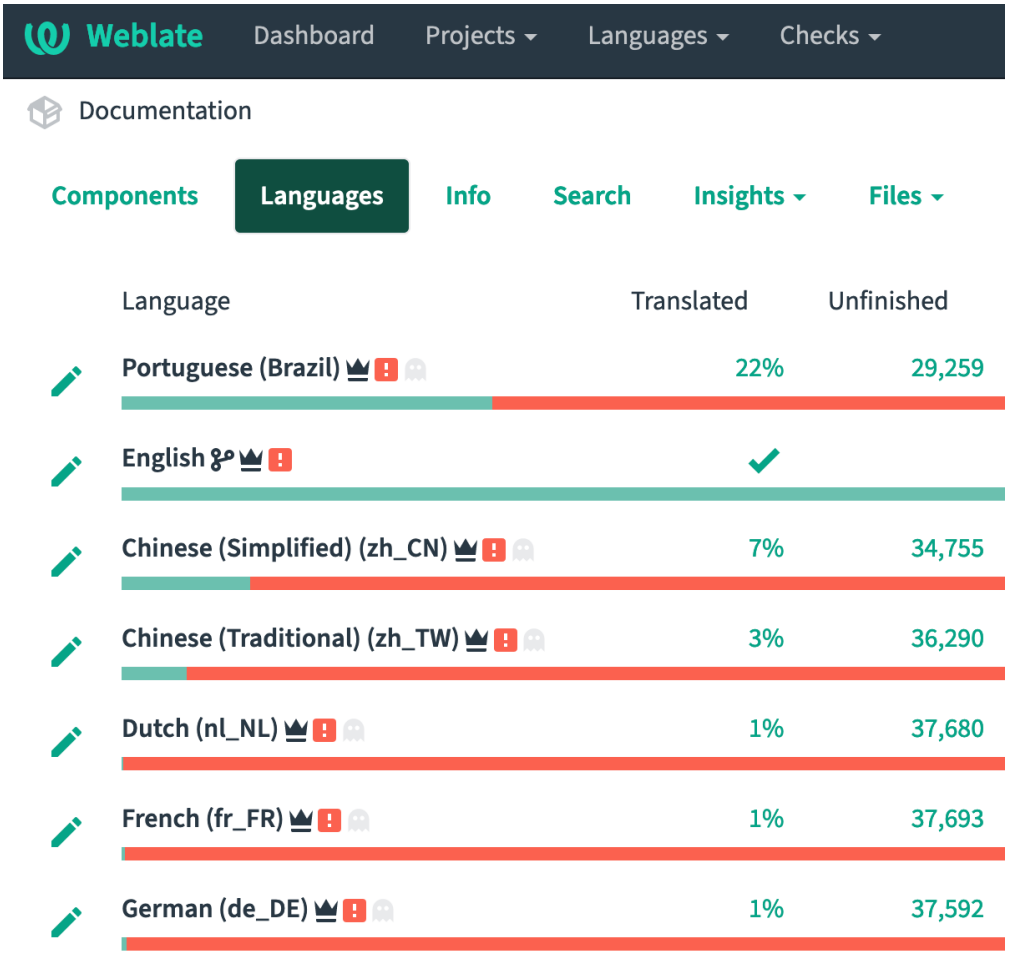
Na instância Weblate do FreeBSD, todas as traduções serão enviadas para o freebsd-doc-translate (um repositório intermediário no GitHub), e não diretamente para o freebsd-doc. Os tradutores devem pegar os arquivos PO gettext (`.po`), converte-los para `.adoc` e submete-los via Bugzilla ou

GitHub para publicar ou atualizar o documento traduzido no portal de documentação. Veja mais nas seções a seguir.

O Weblate irá enviar os commits diariamente, pelo menos, para o `freebsd-doc-translate`, se quaisquer novas strings forem traduzidas.

10.5. Encontre uma Equipe de Idiomas para Participar

Clique em `Projetos`, escolha `Documentação`, depois clique em `Idiomas` e veja todos os idiomas disponíveis.



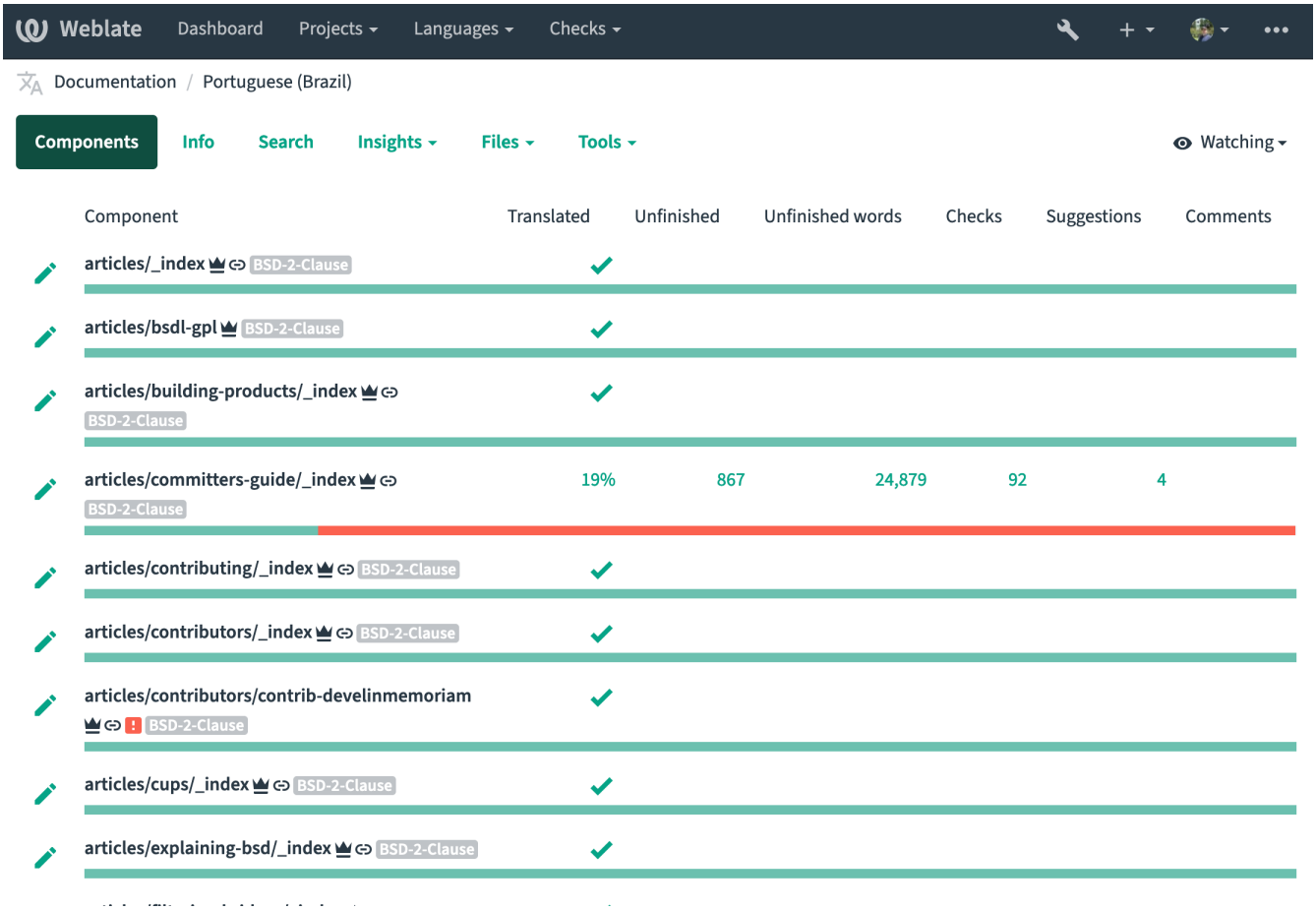
Observe que alguns idiomas e documentos traduzidos já existem no portal de documentação e repositórios.

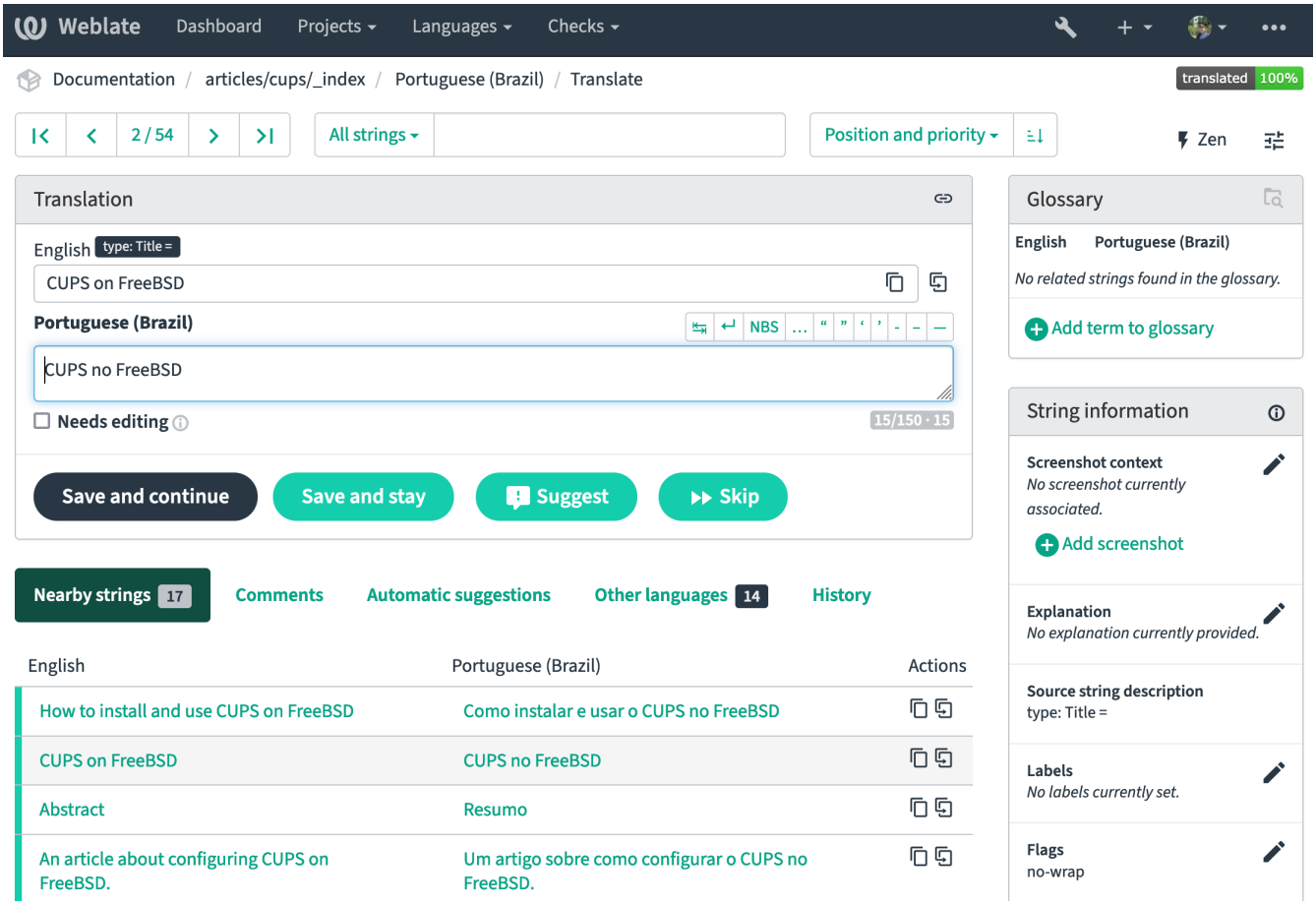
Se o idioma desejado para tradução não estiver disponível no Weblate, entre em contato com os coordenadores de idiomas antes de solicitar a criação de um novo idioma. Se não houver resposta, entre em contato com a Equipe de Engenharia de Documentação <doceng@FreeBSD.org>.

10.6. Traduzindo Online no Weblate

Traduzir documentos online tende a ser o método mais fácil para tradução de documentos no FreeBSD, pois permite que os usuários trabalhem no mesmo arquivo, distribuindo a carga de trabalho.

Assim que um coordenador ou administrador conceder acesso a um idioma específico para um usuário, o botão salvar será habilitado para que esse usuário possa começar a traduzir.



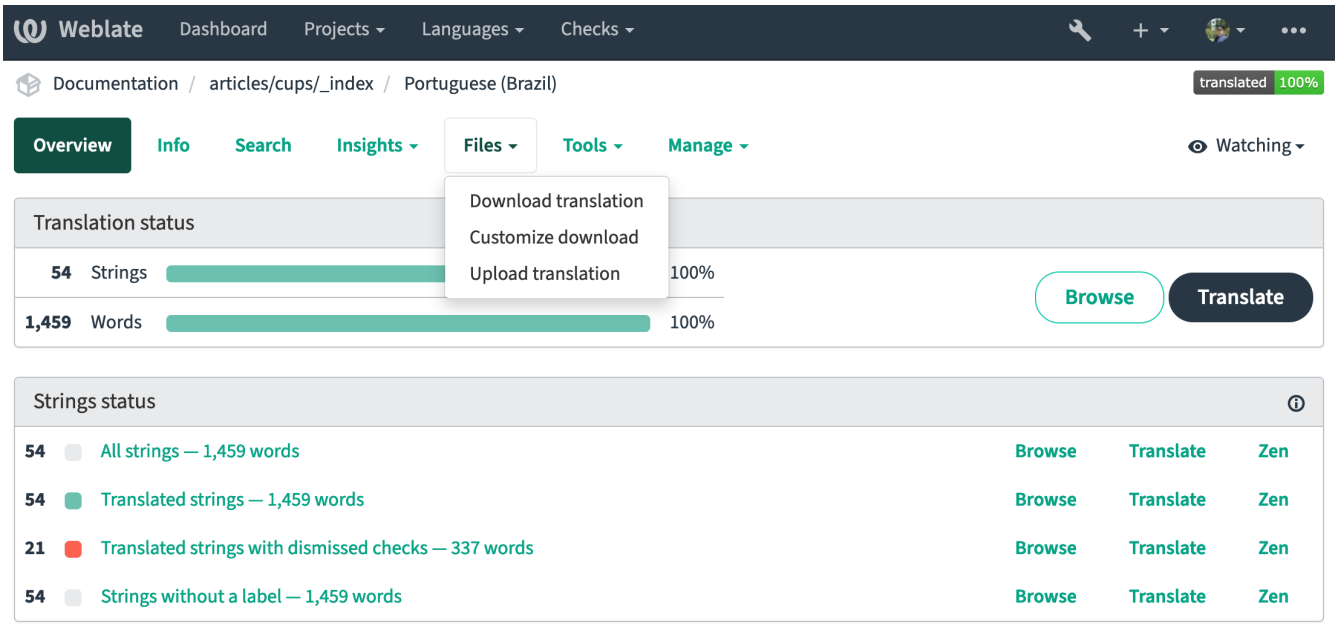


O Weblate possui um conjunto de links que levam à tradução. A tradução é dividida em verificações individuais, como `Não traduzido` ou `Precisando de revisão`. Se todo o documento estiver traduzido sem nenhum erro, o link `Todas as traduções` ainda estará disponível caso uma revisão seja necessária. Como alternativa, o campo de pesquisa pode ser usado para localizar uma string ou termo específico.

Na documentação do Weblate, há mais informações sobre traduções, como atalhos de teclado e outras dicas sobre a ferramenta de tradução.

10.7. Traduzindo Offline

O Weblate no FreeBSD usa arquivos PO gettext para traduções. Os usuários familiarizados com os arquivos PO gettext que desejam traduzir offline podem baixar e enviar as traduções através da página do documento no Weblate clicando na seção `Arquivos`.



10.8. Tradução baseada em Sugestões Automáticas

Os idiomas que usavam Weblate antes da migração para Hugo/Asciidoctor podem usar esse recurso do Weblate para economizar tempo.

Este recurso do Weblate usa a Memória de Tradução gerada pelos outros componentes e projetos no mesmo servidor. As antigas traduções do Weblate estão hospedadas no mesmo servidor como somente leitura por causa disso.

Strings que correspondem a `100/100` em similaridade podem ser copiadas e salvas diretamente. Outras strings precisarão de pelo menos um pequeno ajuste.

Alguns exemplos:

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

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 AT&T.	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.	Weblate (FreeBSD Doc (Archived)/articles_bsdl-gpl) Weblate Translation Memory (Personal: freebsd-doc/articles_bsdl-gpl)	Copy Copy and save 92 / 100

Com a migração para Hugo/Asciidoctor, os documentos passaram a utilizar UTF-8. Algumas entidades HTML devem ser substituídas. Algumas strings, como links, requerem alterações na marcação.

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

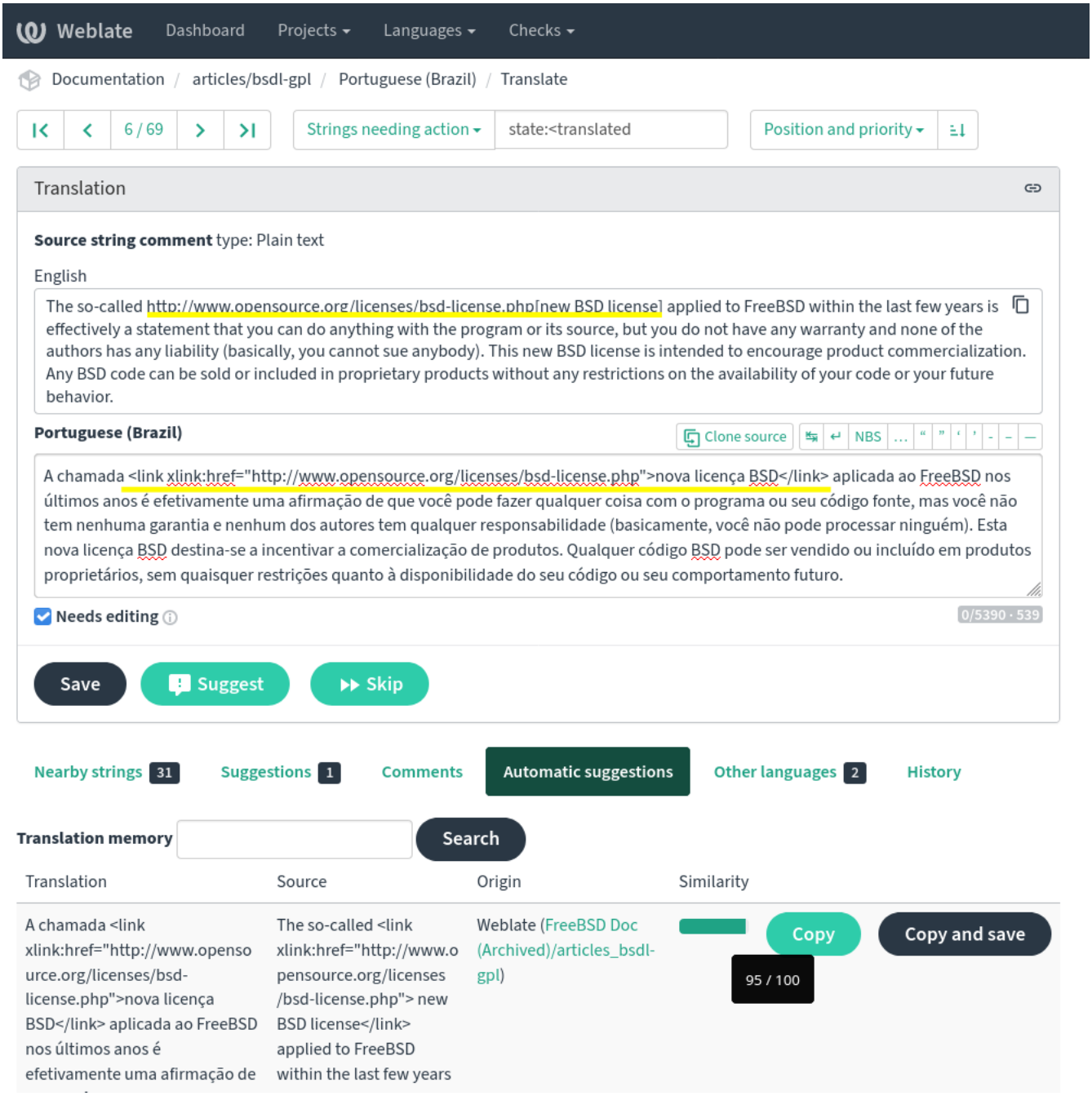
Save Suggest Skip

Nearby strings 29 Suggestions 1 Comments Automatic suggestions Other languages 2 History

Translation memory Search

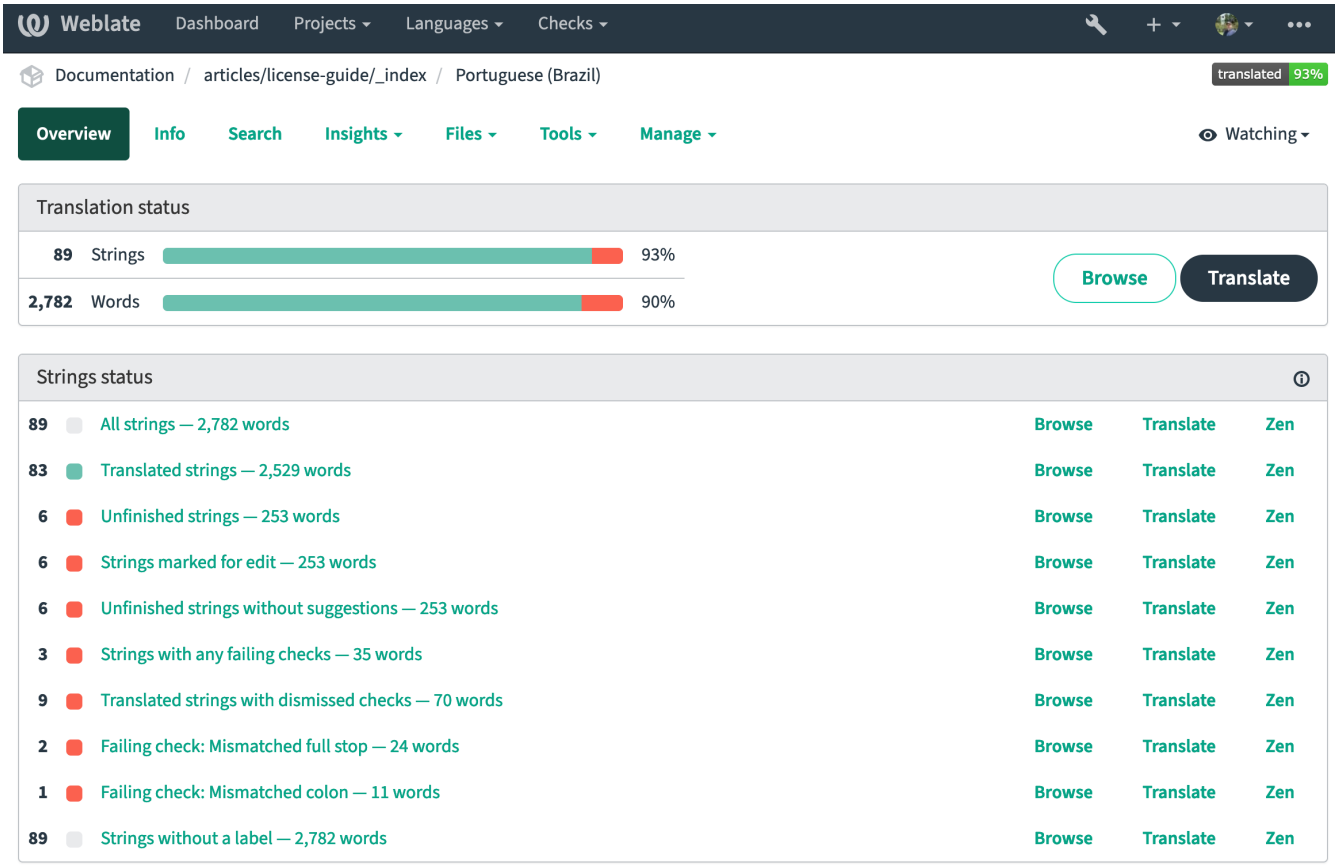
Translation	Source	Origin	Similarity		
Em meados da década de 80, um processo anti-trust governamental contra a AT&T resultou no seu desmembramento. A AT&T ainda possuía o Unix e a partir daquele momento podia vendê-lo. Então a AT&T embarcou em um esforço agressivo de licenciamento e a maioria dos Unixes 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

Links:

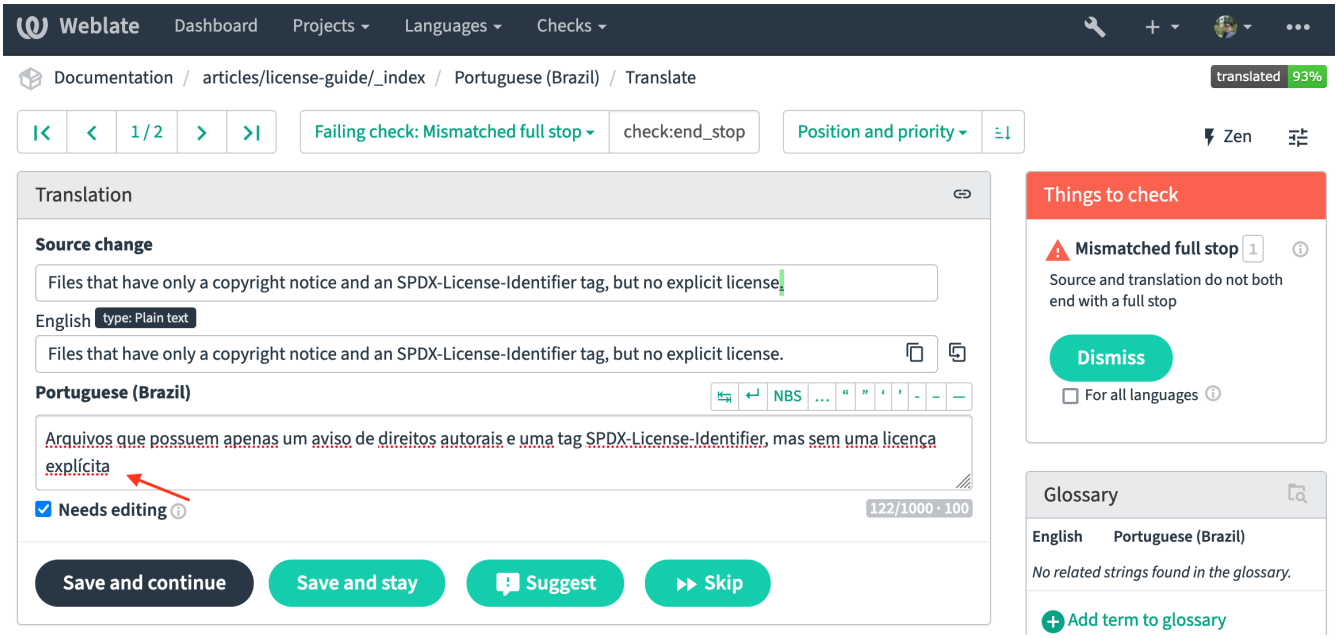


10.9. Revisão e Verificações de Qualidade do Weblate

O dashboard do documento `Projeto/Idioma/Documento` mostra o status da tradução e o status das strings desse documento. Esta página é útil para revisão e verificações de qualidade.



Neste exemplo, falta o ponto final em duas strings; clicando nesse link será mostrado apenas as strings a serem revisadas/traduzidas.



Tradutores e revisores geralmente gostam da visualização das strings traduzidas em seu contexto.

10.10. Compilando o Documento Traduzido

O projeto não usa integração contínua e entrega contínua (CI/CD) para gerar as traduções. Existem estudos para a sua disponibilização.

O exemplo a seguir usa o GitHub, pois o Weblate também está no GitHub. Observe que este repositório é um espelho somente leitura, mas as Pull Requests são aceitas.

Para gerar a tradução localmente, siga estas etapas:

Procedimento: Clone os repositórios necessários

1. Clone o repositório `freebsd-doc`:

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

2. Clone o repositório `freebsd-doc-translate`:

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

Procedimento: Copie um arquivo de tradução para `freebsd-doc`

Com ambos os repositórios, copie a tradução de `freebsd-doc-translate` para `freebsd-doc`. Exemplo de tradução do artigo do Committer's Guide em espanhol.

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

Procedimento: Converter um arquivo de tradução (`.po`) para `.adoc`

Vá para a raiz do `freebsd-doc`.

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

Traduzir (converter) o arquivo `.po` para `.adoc`

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

Por padrão: apenas arquivos com mais de oitenta por cento de strings traduzidas serão convertidos para `.adoc`.

Para ignorar esse limite:

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

Alguns documentos, como livros, possuem muitos arquivos PO gettext. Sempre copie todos eles ao traduzir e compilar. Arquivos que não foram traduzidos serão convertidos com as strings de origem (inglês).

A estrutura de diretórios é fundamental. Siga sempre a estrutura de diretórios dos documentos em inglês.

Procedimento: Gerar o documento traduzido

Por último, a parte da compilação.

Entre no diretório de documentação porque não há necessidade de compilar o site do FreeBSD.

```
% cd documentation
```

E compile a documentação. Observe que `en` é sempre adicionado por padrão ao compilar qualquer outro idioma.

```
% DOC_LANG=es make
```

Este comando irá compilar apenas os documentos em inglês e espanhol do portal de documentação do FreeBSD. A saída estará no diretório public; abra-o em um navegador. Observe que alguns arquivos de índice podem redirecionar o navegador para a página online.

Outra boa opção é compilar e servir o conteúdo com o servidor web interno do Hugo:

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

Por padrão, o servidor web escuta em `localhost`; Para substituir esse comportamento, especifique o endereço IP desejado no valor do parâmetro `BIND`.

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

Isso compila e serve o conteúdo com o servidor web interno de Hugo e deixa ele em espera, quando algum arquivo for alterado, ele o recompila automaticamente.

Para fazer os ajustes necessários na tradução, siga as etapas abaixo para sincronizar novamente todos os componentes:

- Corrija a string de tradução em Weblate.

- Force o Weblate a realizar o commit das alterações na aba `Document/Manage/Commit`.
- Sincronize o repositório Weblate local `freebsd-doc-translate` com o comando `git pull origin main`.
- Copie a tradução novamente para `freebsd-doc`.
- Converta a tradução para .adoc com o script `./tools/translate.sh`.
- Hugo irá recompilar o arquivo e não todo o conjunto se `make run` foi utilizado; ou execute novamente o `make`.

> Siga os passos anteriores quantas vezes forem necessárias até que o documento esteja pronto para ser publicado.

O capítulo Processo de Compilação da Documentação inclui informações sobre a renderização para HTML e PDF.

10.11. Enviando Traduções

Exemplo de envio de atualização do artigo Committer's Guide em Português do Brasil.

Verifique o repositório

Depois de seguir os passos em Compilando o Documento Traduzido, vá para a raiz do `freebsd-doc` e visualize o que deve ser submetido. Para obter uma visão geral dos arquivos a serem alterados e das diferenças no conteúdo do arquivo:

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

Revise a saída e, se algum arquivo não relacionado à atualização da tradução do Committer's Guide foi alterado ou adicionado, tome a ação apropriada de revertê-lo ou removê-lo, respectivamente, antes de continuar.

Sempre inclua o arquivo PO gettext (`.po`) e o documento traduzido em Hugo/Asciidoctor (`.adoc`).

Crie uma nova branch e faça o commit

Crie outra branch para separar o trabalho, o que ajudará em futuras atualizações no repositório local.

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

Registre o commit local.

```
% git add .
```

```
% git commit
```

Exemplo de mensagens de commit para traduções:

```
pt-br/committers-guide: Sincronizar com en XXXXXXX
```

Onde XXXXXXX é a revisão git(1) armazenada no repositório Weblate ~/freebsd-doc-translate/revision.txt.

Se for a primeira tradução de um artigo:

```
Adicionar tradução Coreana do artigo Leap Seconds
```

Uma mensagem será exibida após o commit se o git(1) não tiver sido configurado anteriormente. Siga as instruções e forneça o nome e o endereço de e-mail usado no Weblate. Esta etapa é crucial para creditar o trabalho dos contribuidores.

Em seguida, verifique todo o commit, revise as alterações e o nome e e-mail do autor.

```
% git show
```

Gerar um patch

Em seguida, gere um arquivo git-format-patch(1).

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

Anexe o patch 0001-pt-br-committers-guide-Sync-with-en-XXXXXXX.patch a um relatório de problema no Bugzilla do FreeBSD.

Inclua as seguintes informações no relatório:

Tabela 3. Campos do Bugzilla

Campo	Valor
produto	Documentação
Componente	Artigos e Livros
Resumo	O mesmo que o commit local

Campo	Valor
Descrição	Declare que as instruções deste guia foram seguidas, incluindo revisão e outras etapas necessárias. Inclua informações que podem ajudar na triagem e no progresso do relatório.
CC (Opcional)	Se o idioma tiver coordenadores, inclua seus endereços de e-mail no campo CC.

Para pessoas familiarizadas com git(1) e GitHub: em vez de enviar o patch através do Bugzilla, um pull request do GitHub pode ser usado (use o nome e o endereço de e-mail utilizado no Weblate).

https://github.com/freebsd/freebsd-doc/ é um espelho secundário. Alterações na árvore `doc` podem ser feitas apenas por pessoas que possuem um `doc` commit bit.

Quando os tradutores continuam enviando patches de boa qualidade, eles podem ser nomeados por outros committers para receber acesso de escrita (um doc commit bit para traduções), uma conta FreeBSD e todas as vantagens associadas.

A lista de Contribuidores Adicionais do FreeBSD inclui não-committers cujas contribuições são submetidas à árvore `doc`.

Em caso de dúvida sobre algum procedimento, escreva para a lista de discussão sobre as traduções do FreeBSD.

10.12. FAQ (Perguntas Frequentes)

10.12.1. É necessário traduzir todas as mensagens de Copyright?

Cada equipe de idiomas decide isso para o seu próprio idioma; na equipe `pt-br` (Português do Brasil), foi decidido não traduzir essas mensagens.

Capítulo 11. Páginas de Manual

11.1. Introdução

Páginas de manual, geralmente abreviadas como *man pages*, foram concebidas como lembretes prontamente disponíveis para sintaxe de comandos, detalhes de drivers de dispositivos ou formatos de arquivos de configuração. Elas se tornaram uma referência rápida extremamente valiosa de linha de comando para usuários, administradores de sistema e programadores.

Embora tenham sido planejados como material de referência em vez de tutoriais, as seções EXEMPLOS das páginas de manual geralmente fornecem casos de uso detalhados.

Páginas de manual são geralmente mostradas interativamente pelo comando man(1). Quando o usuário digita `man ls`, uma pesquisa é executada para uma página de manual que corresponde a `ls`. O primeiro resultado correspondente é exibido.

11.2. Seções

As páginas de manual são agrupadas em *seções*. Cada seção contém páginas de manual para uma categoria específica de documentação:

Número da Seção	Categoria
1	Comandos Gerais
2	Chamadas de Sistema
3	Funções de Biblioteca
4	Interfaces do Kernel
5	Formatos de Arquivo
6	Jogos
7	Diversos
8	Gerenciamento do Sistema
9	Desenvolvedor do Kernel

11.3. Marcação

Vários formulários de marcação e programas de renderização foram usados para páginas de manual. O FreeBSD usou groff(7) e o mais recente mandoc(1). A maioria das páginas de manual do FreeBSD, e todas as novas, usam o formulário mdoc(7) de marcação. Esta é uma marcação simples baseada em linhas que é razoavelmente expressiva. É principalmente semântico: partes do texto são marcadas para o que são, em vez de como devem aparecer quando renderizadas. Existe alguma marcação baseada em aparência que geralmente é melhor evitar.

O código fonte de página de manual geralmente é interpretada e exibido na tela interativamente.

Os arquivos fontes podem ser arquivos de texto comuns ou compactados com gzip(1) para economizar espaço.

As páginas de manual também podem ser renderizadas para outros formatos, incluindo PostScript para impressão ou geração de PDF. Veja man(1).

11.3.1. Seções de Página de Manual

Páginas de manual são compostas por várias seções padrão. Cada seção tem um título em letras maiúsculas e as seções de um determinado tipo de página de manual aparecem em uma ordem específica. Para uma página de manual do Comando Geral da categoria 1, as seções são:

Nome da Seção	Descrição
NAME	Nome do comando
SYNOPSIS	Formato de opções e argumentos
DESCRIPTION	Descrição da finalidade e uso
ENVIRONMENT	Configurações de ambiente que afetam a operação
EXIT STATUS	Códigos de erro retornados na saída
EXAMPLES	Exemplos de uso
COMPATIBILITY	Compatibilidade com outras implementações
SEE ALSO	Referência cruzada para páginas de manual relacionadas
STANDARDS	Compatibilidade com padrões como POSIX
HISTORY	Histórico de implementação
BUGS	Erros conhecidos
AUTHORS	Pessoas que criaram o comando ou escreveram a página de manual.

Algumas seções são opcionais e a combinação de seções para um tipo específico de página manual pode variar. Exemplos dos tipos mais comuns são mostrados mais adiante neste capítulo.

11.3.2. Macros

A marcação mdoc(7) é baseada em macros. As linhas que começam com um ponto contêm comandos de macro, com duas ou três letras. Por exemplo, veja esta parte da página de manual do 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 O *Document date* e *Document title* são definidos.

② O *Section header* para a seção NAME é definido. Em seguida, o *Name* do comando e um *Name description* de uma linha são definidos.

③ A seção SYNOPSIS começa. Esta seção descreve as opções de linha de comando e os argumentos que são aceitos.

④ *Name* (`.Nm`) já foi definido, e repeti-lo aqui apenas exibe o valor definido no texto.

⑤ Uma *Optional Flag* chamada `-libxo` é mostrada. A macro `Fl` adiciona um traço ao início das flags, então isso aparece na página de manual como`--libxo`.

⑥ Uma longa lista de flags opcionais de um único caractere é mostrada.

⑦ Uma flag opcional `-D` é definida. Se a flag `-D` for fornecida, ele deve ser seguido por um *Argument*. O argumento é um *format*, uma string que diz ao ls(1) o que mostrar e como mostrar. Detalhes sobre a string de formato são fornecidos posteriormente na página do manual.

⑧ Um argumento opcional final é definido. Visto que nenhum nome é especificado para o argumento, o padrão de `file …` é usado.

⑨ O *Section header* para a seção DESCRIPTION é definido.

Quando renderizado com o comando `man ls`, o resultado exibido na tela é semelhante ao seguinte:

```
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.
```

Valores opcionais são mostrados entre colchetes.

11.3.3. Diretrizes de Marcação

A linguagem de marcação mdoc(7) não é muito rigorosa. Para maior clareza e consistência, o projeto de Documentação do FreeBSD adiciona algumas diretrizes de estilo adicionais:

Apenas a primeira letra das macros é maiúscula

Sempre use maiúsculas para a primeira letra de uma macro e minúscula para as letras restantes.

Comece novas frases em novas linhas

Inicie uma nova frase em uma nova linha, não a inicie na mesma linha de uma frase existente.

Atualizar `.Dd` ao fazer alterações não triviais em uma página de manual

A *Data do documento* informa o leitor sobre a última vez que a página de manual foi atualizada. É importante atualizar sempre que alterações não triviais forem feitas nas páginas de manual. Alterações triviais, como correções ortográficas ou de pontuação que não afetam o uso, podem ser feitas sem atualizar `.Dd`.

Apresentando exemplos

Apresente exemplos ao leitor sempre que possível. Mesmo exemplos triviais são valiosos, porque o que é trivial para o escritor não é necessariamente trivial para o leitor. Três exemplos são um bom objetivo. Um exemplo trivial mostra os requisitos mínimos, um exemplo afundo mostra o uso real e um exemplo detalhado demonstra uma funcionalidade incomum ou não óbvia.

Inclua a licença BSD

Inclua a licença BSD em novas páginas de manual. A licença preferencial está disponível no Guia dos Committer's.

11.3.4. Truques de Marcação

Adicione um espaço antes da pontuação em uma linha com macros. Exemplo:

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

Observe como as vírgulas no final das linhas `.Xr` foram colocadas após um espaço. A macro `.Xr` espera dois parâmetros, o nome de uma página de manual externa e um número de seção. O espaço separa a pontuação do número da seção. Sem o espaço, os links externos apontariam incorretamente para a seção `4,` ou `8,`.

11.3.5. Macros Importantes

Algumas macros muito comuns serão mostradas aqui. Para obter mais exemplos de uso, consulte mdoc(7), groff_mdoc(7), ou procure por uso real no diretório /usr/share/man/man*. Por exemplo, para procurar exemplos da macro `.Bd` *Begin display*:

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

11.3.5.1. Macros Organizacionais

Algumas macros são usadas para definir blocos lógicos de uma página de manual.

Macro Organizacional	Uso
`.Sh`	Cabeçalho da seção (Section header). Seguido do nome da seção, tradicionalmente toda em caixa alta. Pense nisso como títulos de capítulos.
`.Ss`	Cabeçalho da subseção (Subsection header). Seguido pelo nome da subseção. Usado para dividir uma seção `.Sh` em subseções.
`.Bl`	Comece a lista (Begin list). Inicie uma lista de itens.
`.El`	Terminar lista (End list).
`.Bd`	Comece a exibição (Begin display). Comece uma área especial do texto, como uma área recuada.
`.Ed`	Fim da exibição (End display).

11.3.5.2. Macros Inline

Muitas macros são usadas para marcar texto embutido.

Macro inline	Uso
`.Nm`	Nome. Chamado com um nome como parâmetro no primeiro uso, depois usado sem o parâmetro para exibir o nome que já foi definido.
`.Pa`	Caminho para um arquivo (Path to a file). Usado para marcar nomes de arquivos e caminhos de diretório.

11.4. Exemplo de Estruturas de Página de Manual

Esta seção mostra o conteúdo mínimo desejável para um página de manual para várias categorias comuns de páginas de manual.

11.4.1. Seção 1 ou 8 sobre um comando

A estrutura básica preferida para uma seção 1 ou 8 sobre um comando:

```
.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. Seção 4 sobre um Driver de Dispositivo

A estrutura básica preferida para a seção 4 sobre um driver de dispositivo:

```
.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. Seção 5 sobre um Arquivo de Configuração

A estrutura básica preferida para a seção 5 sobre um arquivo de configuração:

```
.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. Testando

O teste de uma nova página de manual pode ser um desafio quando o arquivo não está localizado no caminho de pesquisa normal da páginas de manual. man(1) também não procura no diretório atual. Se a nova página de manual estiver no diretório atual, prefixe o nome do arquivo com um `./`

Use o linter de mandoc(1) para verificar se há erros:

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

Use o textproc/igor para revisar a página do manual:

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

Outra ferramenta útil é o textproc/vale. Ele não suporta a sintaxe mdoc(7), mas a página de manual renderizada pode ser lida e analisada a partir da entrada padrão:

```
% man ls | vale
```

textproc/vale é altamente configurável. É aconselhável ler sua documentação.

Use man(1) para verificar o resultado final de suas alterações:

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

Você pode usar col(1) para filtrar a saída de man(1) e se livrar dos caracteres backspace antes de carregar o resultado em seu editor favorito para verificação ortográfica:

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

A verificação ortográfica com dicionários completos é incentivada e pode ser realizada usando textproc/hunspell ou textproc/aspell combinado com textproc/en-hunspell ou textproc/en-aspell, respectivamente. Por exemplo:

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

11.6. Exemplos de páginas de manuais para usar como modelos

Algumas destas páginas de manual são adequadas para serem usadas como exemplos detalhados.

Página de Manual	Caminho para o arquivo de origem
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. Recursos

Recursos para escritores de páginas manuais:

- man(1)
- mandoc(1)
- groff_mdoc(7)
- Practical UNIX Manuals: mdoc
- History of UNIX Manpages

Capítulo 12. Estilo de escrita

12.1. Dicas

A documentação técnica pode ser melhorada pelo uso consistente de vários princípios. A maioria destes pode ser classificada em três objetivos: *ser claro*, *ser completo* e *ser conciso*. Essas metas podem entrar em conflito umas com as outras. Uma boa escrita consiste em um equilíbrio entre elas.

12.1.1. Seja claro

A clareza é extremamente importante. O leitor pode ser um novato ou ler o documento em um segundo idioma. Esforce-se por um texto simples e descomplicado que explique claramente os conceitos.

Evite discurso florido ou embelezado, piadas ou expressões coloquiais. Escreva da maneira mais simples e clara possível. Um texto simples é mais fácil de se entender e de se traduzir.

Mantenha as explicações o mais curtas, simples e claras possíveis. Evite frases vazias como "a fim de" as quais normalmente significam apenas um "para". Evite palavras potencialmente paternalistas tais como "basicamente". Evite termos latinos como "i.e." ou "cf.", os quais podem ser desconhecidos fora de grupos acadêmicos ou científicos.

Escreva em um estilo formal. Evite dirigir-se ao leitor como "você". Por exemplo, digamos "copie o arquivo para /tmp" em vez de "você pode copiar o arquivo para /tmp".

Dê exemplos claros, corretos, e *testados*. Um exemplo trivial é melhor do que nenhum exemplo. Um bom exemplo é ainda melhor. Não dê exemplos ruins, identificáveis por desculpas ou frases como "mas realmente isso nunca deve ser feito dessa forma". Exemplos ruins são piores que nenhum exemplo. Dê bons exemplos, porque *mesmo quando avisado para não usar o exemplo como mostrado* , o leitor normalmente só usa o exemplo como mostrado.

Evite palavras vazias como "deveria", "poderia", "tentaria", ou "podia". Estas palavras implicam que o autor não tem certeza dos fatos e cria dúvidas no leitor.

Da mesma forma, dê instruções como comandos imperativos: não utilize "você deve fazer isso", mas apenas "faça isso".

12.1.2. Seja completo

Não faça suposições sobre as habilidades do leitor. Diga-lhes o que precisam saber. Dê links para outros documentos para fornecer informações básicas sem precisar recriá-las. Coloque-se no lugar do leitor, antecipe as perguntas que eles farão e responda-os.

12.1.3. Seja conciso

Embora as funcionalidades devam ser documentadas completamente, às vezes existe tanta informação que o leitor não consegue encontrar facilmente os detalhes específicos de que necessita.

O equilíbrio entre ser completo e ser conciso é um desafio. Uma abordagem é ter uma introdução e, em seguida, uma seção de "início rápido" que descreve a situação mais comum, seguida por uma seção de referência aprofundada.

12.2. Diretrizes

Para promover a consistência entre os inúmeros autores da documentação do FreeBSD, algumas diretrizes foram elaboradas para os autores seguirem.

Use a Ortografia do Inglês Americano

Existem várias variantes do Inglês, com grafias diferentes para a mesma palavra. Onde as grafias diferem, use a variante do Inglês Americano. "color", não "colour", "rationalize", não "rationalise", e assim por diante.

> O uso do Inglês Britânico pode ser aceito no caso de um artigo contribuído, no entanto, a ortografia deve ser consistente em todo o documento. Os outros documentos, como livros, site, páginas de manual, etc, devem usar o Inglês Americano.

Não use contrações

Não use contrações. Sempre soletre a frase na íntegra. "Do not" é a forma correta, "Don't" é a errada.

Evitar contrações contribui para um tom mais formal, é mais preciso e é um pouco mais fácil para os tradutores.

Use a vírgula serial

Em uma lista de itens dentro de um parágrafo, separe cada item dos outros com uma vírgula. Separe o último item dos outros com uma vírgula e a letra "e".

Por exemplo:

Esta é uma lista de um, dois e três itens.

Esta é uma lista de três itens, "um", "dois", e "três", ou uma lista de dois itens, "um" e "dois" e "três"?

É melhor ser explícito e incluir uma vírgula serial:

Esta é uma lista de um, dois, e três itens.

Evite frases redundantes

Não use frases redundantes. Em particular, "the command", "the file", e "man command" são frequentemente redundantes.

Por exemplo, comandos:

Errado: Use o comando `git` para atualizar o código fonte.

Correto: Use o `git` para atualizar o código fonte.

Nomes de arquivo:

Errado: ... no nome do arquivo /etc/rc.local...

Correto: ... no /etc/rc.local...

Referências de páginas de manual (o segundo exemplo usa `man:[]` com a entidade csh(1)):

Errado: veja `man csh` para mais informações.

Certo: Veja csh(1).

Para mais informações sobre o estilo de escrita, consulte Elements of Style de William Strunk.

12.3. Guia de estilo

Para manter o código fonte da documentação consistente quando muitas pessoas diferentes a estiverem editando, siga estas convenções de estilo.

12.4. Uma frase por linha

Use quebras de linha semântica na documentação, uma técnica chamada "uma frase por linha". A ideia dessa técnica é ajudar os usuários a escrever e ler a documentação. Para obter mais informações sobre essa técnica, leia a página Semantic Line Breaks.

Este é um exemplo que não usa "uma frase por linha".

```
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.
```

E este é um exemplo que usa a técnica.

```
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.
```

12.5. Siglas

As siglas devem ser definidas na primeira vez que aparecerem em um documento, como em: "Network Time Protocol (NTP)". Depois que o acrônimo tiver sido definido, use apenas a sigla, a menos que faça mais sentido contextualmente usar todo o termo. Siglas geralmente são definidos apenas uma vez por capítulo ou por documento.

Todas as siglas devem ser incluídas com o caractere `.

12.6. Lista de Caracteres Especiais

Esta lista de caracteres especiais mostra a sintaxe correta e a saída quando usada na documentação do FreeBSD. Se um caractere não está nesta lista, pergunte sobre ele na lista de discussão do projeto de documentação do FreeBSD.

Nome	Sintaxe	Renderizado
Copyright	(C)	©
Registrado	(R)	®
Marca Comercial	(TM)	™
Travessão	--	—
Elipses	...	…
Seta simples para a direita	->	→
Seta dupla para a direita	=>	⇒
Seta simples para a esquerda	<-	←
Seta dupla para a esquerda	<=	⇐

12.7. Linting com Vale

Para manter clareza e consistência em toda a documentação e páginas do site, estilos Vale foram introduzidos na árvore de documentação. Vale é um linter poderoso para escrever regras personalizadas e pode ser usado em vários cenários. Atualmente o Vale pode ser usado como uma ferramenta de linha de comando, para pipelines de CI/CD e integrado a um editor de texto de sua escolha.

A tabela a seguir descreve os nomes das regras atuais e as suas respectivas severidade.

Nome	Severidade
FreeBSD.BrandTerms	erro
FreeBSD.ConsciousLanguage	aviso
FreeBSD.Contractions	sugestão
FreeBSD.EOLSpacing	aviso
FreeBSD.Hang	aviso
FreeBSD.Hyphens	aviso
FreeBSD.Spacing	erro
FreeBSD.SuperfluousOptArgInLinks	sugestão
Vale.Avoid	erro
Vale.Repetition	erro
Vale.Spelling	erro

Nome	Severidade
Vale.Terms	erro

12.7.1. Regras Atuais do Vale

1. FreeBSD.BrandTerms: De acordo com as regras de direitos autorais da Fundação FreeBSD, **freebsd** deve ser escrito como **FreeBSD**. Da mesma forma, todos os principais fornecedores e empresas têm regras específicas sobre como escrever seus nomes de marcas e marcas registradas. Deve-se tomar cuidado para respeitar o valor da marca de outras pessoas e reservar um tempo para escrever PostgreSQL, Node.js, Let's Encrypt, etc. Nomes de marcas ausentes devem ser adicionados ao .vale/styles/FreeBSD/BrandTerms.yml no repositório `doc`.
2. FreeBSD.ConsciousLanguage: Esta regra propõe o uso de linguagem consciente para que palavras sensíveis apontando para a cor, idade, raça ou orientação sexual das pessoas sejam evitadas sempre que possível.
3. FreeBSD.Contractions: Palavras contraídas não devem ser usadas. Esta regra evita todas as contrações e sugere palavras completas.
4. FreeBSD.EOLSpacing: Na maioria dos documentos, espaços presentes no fim da linha (EOL) não são desejáveis.
5. FreeBSD.Hang: `Hang` é frequentemente usado para significar de que o aplicativo parou de responder. Esta norma propõe melhor redação.
6. FreeBSD.Hyphens: Muitas vezes advérbios que terminam com 'ly' são adicionados com um hífen, o que está errado.
7. FreeBSD.Spacing: Muitas vezes, os espaços duplos são difíceis de captar a olho nu e isso é abordado aqui.
8. FreeBSD.SuperfluousOptArgInLinks: Sugere colchetes vazios nas macros `link:` quando o texto exibido coincide com a URL.
9. Vale.Avoid: Impõe os termos de vocabulário **NÃO USE** para o Projeto FreeBSD. Se for encontrada alguma palavra que não deva estar na documentação, a palavra deve ser adicionada a .vale/styles/Vocab/Terms/reject.txt no repositório `doc`. A lista está vazia no momento.
10. Vale.Repetition: Muitas vezes, as mesmas palavras são digitadas duas vezes ao sair do teclado e voltar ao trabalho novamente. Esta regra encontra palavras repetidas e avisa os usuários.
11. Vale.Spelling: No momento, há uma mistura de grafias en_US e en_GB na documentação e no site. Vale vem com um dicionário embutido do qual usa estritamente en_US e não aceita a variante en_GB de nenhuma palavra.
12. Vale.Terms: Aplica os termos de vocabulário **PREFERIDO** para o Projeto FreeBSD. No momento, a lista de termos está vazia e os termos específicos do FreeBSD serão adicionados gradualmente. Se alguma palavra estiver correta e não disponível no dicionário, a palavra deve ser adicionada ao .vale/styles/Vocab/Terms/accept.txt no repositório `doc`.

Mais regras serão introduzidas nos próximos dias, quando e onde for necessário.

12.7.2. Utilizando o Vale

O Vale pode ser usado em linha de comando e em um editor de texto ou IDE. textproc/vale pode ser instalado da seguinte forma:

```
$ pkg install vale
```

12.7.2.1. Usando o Vale na linha de comando

Assumindo que o repositório `doc` foi clonado em ~/doc os seguintes comandos são necessários para executar:

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

O Vale é um programa intensivo de CPU e memória devido à natureza do aplicativo e pode demorar um pouco para mostrar qualquer saída na tela. Uma melhor maneira de executar o aplicativo é em diretórios ou arquivos específicos, em vez de todo o repositório `doc`, pois isso já é feito na pipeline de CI.

12.7.2.2. Usando Vale em editores

O Vale funciona com os principais editores tradicionais como o editors/vim, editors/emacs, editors/vscode. No momento a configuração necessária para o editors/vim estão descritas em rossref:editor-config[editor-config-vim, Vim]. As configuração para o editors/emacs está sendo desenvolvida.

Capítulo 13. Configuração do Editor

Ajustar a configuração do editor de texto pode tornar o trabalho nos arquivos da documentação mais rápido e fácil, além de ajudar os documentos a ficarem em conformidade com as diretrizes do projeto.

13.1. Vim

Instale o editors/vim e em seguida siga as instruções em Uso. Usuários mais avançados podem usar um linter mais adequado como o Ale que também pode atuar como um Protocolo de Servidor de Idiomas do Vim.

13.1.1. Uso

Os escritores de páginas de manuais podem usar os seguintes atalhos de teclado para reformatar:

- Pressione `P` para reformatar parágrafos ou texto selecionado no modo Visual.
- Pressione `T` para substituir grupos de oito espaços por um tab.

Um linter chamado Vale foi adicionado para verificar erros gramaticais e cosméticos nos documentos. O Vale possui suporte para diversos editores e IDEs.

O Vale já é instalado como uma dependência do textproc/docproj meta-port. Caso contrário, instale textproc/vale com:

```
$ pkg install vale
```

Instale o Ale para intergrar o textproc/vale com o editors/vim.

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

Os usuários que estão usando gerenciadores de plug-in no editors/vim não precisam dos passos acima e devem seguir as instruções do próprio gerenciador de plug-in para instalar o Ale.

Neste momento devido a um bug no Vale é necessário copiar a configuração do Vale para o diretório home. Considerando que o repositório foi clonado em ~/doc da seguinte forma:

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

13.1.2. Configuração

Edite o ~/.vimrc, adicionando estas linhas ao final do arquivo:

~/.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
```

A configuração acima removerá automaticamente a linha final, o espaço final e vários espaços que podem exibir alterações indesejadas adicionais na saída do `git diff`. Nesses casos, mencione isso adequadamente no log de commit.

13.2. Emacs

Instale-o a partir de editors/emacs ou editors/emacs-devel.

13.2.1. Revisão Automatizada com Flycheck e Igor

O pacote Flycheck está disponível no Emacs Lisp Package Archive da Milkypostman (MELPA). Se a MELPA ainda não estiver nos repositórios de pacotes do Emacs, ele pode ser adicionado executando

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

Adicione a linha ao arquivo de inicialização do Emacs (qualquer um deles, ~/.emacs, ~/.emacs.el, ou ~.emacs.d/init.el) para tornar esta alteração permanente.

Para instalar o Flycheck, execute

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

Crie um verificador Flycheck para textproc/igor executando

```
(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)
```

Novamente, adicione essas linhas ao arquivo de inicialização do Emacs para tornar as mudanças permanentes.

13.2.2. Configurações Específicas da Documentação do FreeBSD

Para aplicar configurações específicas para o projeto de documentação do FreeBSD, crie o arquivo .dir-locals.el no diretório raiz do repositório de documentação e adicione estas linhas ao arquivo:

```
;;; 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

Instale o aplicativo editors/nano.

13.3.1. Configuração

Atualmente não há arquivo de highlight de sintaxe adoc/asciidoc com distribuição nano. Então vamos criar um do zero e usar um editor para criar um novo arquivo ou adicionar linhas no ~/.nanorc com este conteúdo:

~/.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}"
```

Processe o arquivo para criar guias incorporadas:

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

13.3.2. Uso

Especifique opções úteis adicionais ao executar o editor:

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

Usuários do csh(1) podem definir um alias em ~/.cshrc para automatizar estas opções:

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

Depois que o alias é definido, as opções serão adicionadas automaticamente:

```
% nano _index.adoc
```

Capítulo 14. Marcas Registradas

Para todos os documentos do Projeto de Documentação do FreeBSD, é necessário citar as marcas registradas e os outros tipos são habituais, e isso é um requisito para todo escritor e colaborador.

14.1. Símbolos de Marca Registrada

Acrescente um símbolo de marca registrada (™, ® ou outros símbolos) na primeira ocorrência do nome da marca e sempre ao usar logotipos. Use a sequência ASCII equivalente, que será renderizada como o caractere Unicode real. Além disso, escreva o nome da marca registrada seguindo as diretrizes da marca registrada.

Em caso de dúvida, pesquise o site do proprietário da marca registrada, o site do produto e/ou o site United States Patent and Trademark Office trademark search.

14.2. Citação de Marca Registrada

O Projeto de Documentação do FreeBSD fornece um modelo para citar marcas registradas, o que também evita a duplicação de marcas registradas nos documentos.

Primeiro, procure a marca registrada na Seção de direitos autorais no modelo do projeto e adicione-a às tags de marcas registradas na seção 'Front Matter' do documento, localizada no início de cada documento.

O seguinte é um exemplo de `Front Matter` do artigo Contribuindo com o 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"]
---
```

As tags de marca registrada `freebsd`, `ieee` e `general` serão renderizadas automaticamente ao compilar o documento, ficando desta forma:

```
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.
```

Caso a marca não esteja presente no template do projeto, ela deve ser submetida. Qualquer desenvolvedor ou colaborador pode atualizar as marcas registradas.

As tags de marca registrada `freebsd` e `general` geralmente estão presentes em todos os documentos.

Capítulo 15. Veja também

Este documento não é deliberadamente uma discussão exaustiva de AsciiDoc e o Projeto de Documentação do FreeBSD. Para mais informações sobre estes, você é encorajado a consultar os seguintes sites.

15.1. O projeto de documentação do FreeBSD

- As Páginas Web do Projeto de Documentação do FreeBSD
- O Handbook do FreeBSD

15.2. Hugo

- Hugo
- Documentação Hugo

15.3. AsciiDoctor

- AsciiDoctor
- Portal de Documentação do AsciiDoctor

15.4. HTML

- O Consórcio World Wide Web
- A especificação HTML 5
- A especificação CSS
- Sass

Apêndice A: Exemplos

Estes exemplos não são extensos - eles não contêm todos os elementos que podem ser desejáveis de usar, particularmente em relação ao início dos documentos (Front Matter). Para mais exemplos de marcação AsciiDoctor, examine o código fonte em AsciiDoctor deste e de outros documentos disponíveis no repositório Git **doc** ou no link https://cgit.freebsd.org/doc/.

A.1. AsciiDoctor book

Exemplo 15. AsciiDoctor **book**

```
---
title: Um Livro de Exemplo
authors:
  - author: The FreeBSD Documentation Project
copyright: 1995-2021 The FreeBSD Documentation Project
releaseinfo: ""
trademarks: ["general"]
---

= Um Livro de Exemplo
:doctype: book
:toc: macro
:toclevels: 2
:icons: font
:xrefstyle: basic
:relfileprefix: ../
:outfilesuffix:
:sectnums:
:sectnumlevels: 6
:partnums:
:chapter-signifier: Chapter
:part-signifier: Part
:source-highlighter: rouge
:experimental:
:skip-front-matter:
:book: true
:pdf: false


:chapters-path:


[abstract]
Abstract

Abstract section

'''
```

```
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. Artigo AsciiDoctor

Exemplo 16. **Artigo** *AsciiDoctor*

```
---
title: Um Exemplo de Artigo
authors:
  - author: Seu nome e sobrenome
    email: foo@example.com
trademarks: ["general"]
---

= Um Exemplo de Artigo
:doctype: article
:toc: macro
:toclevels: 1
:icons: font
:sectnums:
:sectnumlevels: 6
:source-highlighter: rouge
:experimental:

'''

toc::[]

== My First Section

This is the first section in my article.

=== My First Sub-Section

This is the first sub-section in my article.
```