

Capítulo 7

Editores de texto

7.1. Objetivos

- Conhecer os diversos editores textos;
- Explorar o editor de textos nano;
- Explorar o editor de textos Vim;

7.2. Introdução teórica

A grande maioria das configurações em sistemas GNU/Linux são feitas editando-se diretamente arquivos de configuração em modo texto. Para que essa tarefa seja executada com sucesso, é preciso conhecer alguns editores de texto dentre eles: vi, vim, nano, pico, mcedit, ed, emacs dentre muitos outros:

- **vi** - Sem dúvida nenhuma o editor mais famoso de todos os tempos, presente em quase todas as distribuições;
- **vim** - Uma versão melhorada do vi, Vim significa VImproved e trás diversas facilidades sem perder os conceitos do vi;
- **nano** - Editor padrão de muitas distribuições como Debian e Ubuntu, esse editor diferente do vim é muito fácil de ser usado;
- **pico** - Muito parecido com o nano, este está presente nas distribuições Slackware e Gentoo;
- **mcedit** - Editor muito fácil e completo que está presente nas distribuições Red-Hat, CentOS, Seu grande diferencial é a possibilidade da utilização do mouse;
- **ed** - O editor mais simples de texto presente no mundo Unix, o ed é um editor de linha para terminais aonde não é possível abrir uma janela de edição;

- **emacs** - Poderoso editor de "tudo", o emacs também é muito conhecido no mundo GNU/LINUX por fazer muitas coisas diferenciadas de um editor de texto;



*O editor **od**, pode ser importante para prova. Sua principal funcionalidade é mostrar a saída de um arquivo em formatos como octal, ASCII entre outros.*

Nesse capítulo vamos abordar apenas a utilização dos editores nano e vim.

7.3. Editor Nano

O nano é o editor padrão de textos do Debian, e distribuições baseadas nele. Esse editor é muito fácil de ser usado, e sua interface é muito intuitiva e agradável.

Para abrirmos o editor devemos chamar o seguinte comando:

```
$ nano
```

^G Ajuda
^X Sair

^O Gravar
^J Justificar

^R Ler o Arquivo
^W Onde está?

^Y Página Anterior
^V Próxima Página

^K Recortar Texto
^U Colar Txt

^C Pos Atual
^T Para Spell

Ilustração 5: Menu do Nano

Ao ser chamado, este editor irá apresentar um tela em branco com um rodapé semelhante a esse:

Vamos analisar essas funções:



*Lembrando que **^G** é igual a Ctrl + G e assim por diante ...*

- **^G Get Help** - Apresenta uma tela de ajuda os mais diversos comandos e uma breve explicação sobre o editor;
- **^X Exit** - Sai do editor, lembrando que se o arquivo não estiver salvo, essa opção irá te pedir para salvar;

- **^O WriteOut** - Salva ou sobrescreve um arquivo;
- **^J Justify** - Justifica o arquivo inteiro;
- **^R Read File** - Abre um arquivo;
- **^W Where Is** - Procura por uma ocorrência dentro do arquivo;
- **^Y Prev Page** - Move o cursor para pagina anterior;
- **^V Next Page** - Move o cursor para próxima pagina;
- **^K Cut Text** - Corta a linha em que o cursor está posicionado;
- **^U UnCut Text** - Cola a linha recortada na posição atual do cursor
- **^C Cur Pos** - Mostra informações sobre a posição do cursor;
- **^T To Spell** - Auto correção, lembrando que é necessário ter o comando spell instalado;

Como podemos ver usar o editor de textos nano, não é uma das tarefas mais difíceis no GNU/Linux. Vamos ver agora o editor Vim



As variáveis relacionadas com os editores podem ser valiosas na prova. Uma delas é a própria variável EDITOR

```
EDITOR=nano visudo
EDITOR=vim visudo
```

Observem que a variável define o editor que abrirá um programa que chama o editor padrão. Para definirmos quem é o editor padrão podemos usar o aplicativo update-alternatives .

```
update-alternatives --config editor
```

7.4. Editor Vim

O Vi é o editor básico do GNU/Linux, está disponível em grande parte das distribuições do GNU/Linux , mesmo naquelas que vêm em apenas um disquete.

Hoje em dia, as distribuições usam uma versão mais completa e com mais

em `/etc/vim/vimrc`. Para configurar o seu editor de textos, basta descomentar as funcionalidades desejadas, e copiar o arquivo para o seu home como **.vimrc**

```
$ cp /etc/vim/vimrc ~/.vimrc
```

7.5. Prática Dirigida

1) Faça uma cópia de segurança do arquivo `/etc/passwd` para seu diretório home:

```
$ cp /etc/passwd ~
```

2) Vamos testar alguns comandos básicos do vim:

```
$ vim passwd
```

7.5.1. Teste os comandos de Edição

1) Comandos básicos de inserção de texto:

- **i** - Insere texto antes do cursor
- **a** - Insere texto depois do cursor
- **r** - Substitui texto no início da linha onde se encontra o cursor
- **A** - Insere texto no final da linha onde se encontra o cursor
- **o** - Adiciona linha abaixo da linha atual
- **O** - Adiciona linha acima da linha atual
- **Ctrl + h** - Apaga o último caractere

Teste os comandos de Movimentação

2) Comandos básicos de movimentação:

- **Ctrl+f** - Move o cursor para a próxima tela
- **Ctrl+b** - Move o cursor para a tela anterior
- **H** - Move o cursor para a primeira linha da tela
- **M** - Move o cursor para o meio da tela
- **L** - Move o cursor para a última linha da tela
- **h** - Move o cursor um caractere à esquerda
- **j** - Move o cursor para a próxima linha
- **k** - Move o cursor para linha anterior
- **l** - Move o cursor um caractere à direita
- **w** - Move o cursor para o início da próxima palavra
- **W** - Move o cursor para o início da próxima palavra, separadas por espaço
- **b** - Move o cursor para o início da palavra anterior
- **B** - Move o cursor para o início da palavra anterior, separadas por espaço
- **0(zero)** - Move o cursor para o início da linha atual
- **^** - Move o cursor para o primeiro caractere não branco da linha atual
- **\$** - Move o cursor para o final da linha atual
- **nG** - Move o cursor para a linha n
- **:n** - Move o cursor para a linha n
- **gg** - Move o cursor para a primeira linha do arquivo
- **G** - Move o cursor para a última linha do arquivo

Escreva um texto e teste comandos de Localização**3) Comandos básicos para localizar texto:**

- **/palavra** - Busca pela palavra ou caractere em todo o texto
- **?palavra** - Move o cursor para a ocorrência anterior da palavra
- **n-** Repete o último comando / ou ?
- **N** - Repete o último comando / ou ?, na direção reversa
- **Ctrl+g** - Mostra o nome do arquivo, o número da linha atual e o total de linhas

Teste também os comandos de Alteração

4) Comandos básicos para alteração de texto:

- **x** - Deleta o caractere que está sob o cursor ;
- **dw** - Deleta a palavra, da posição atual do cursor até o final ;
- **dd** - Deleta a linha atual, e copia o conteúdo para área de transferência ;
- **D** - Deleta a linha a partir da posição atual do cursor até o final ;
- **:A,Bd** - Deleta da linha A até a linha B, copia para área de transferência ;
- **rx** - Substitui o caractere sob o cursor pelo especificado em x ;
- **u** - Desfaz a última modificação ;
- **U** - Desfaz todas as modificações feitas na linha atual ;
- **J** - Une a linha corrente a próxima ;
- **yy** - **E**Copia 1 linha para a área de transferência ;
- **yNy** - Copia N linhas para a área de transferência ;
- **p** - Cola o conteúdo da área de transferência ;
- **Np** - Cola N vezes o conteúdo da área de transferência ;
- **cc** - Apaga o conteúdo da linha, e copia para área de transferência ;
- **cNc** - Apaga o conteúdo de N linhas, e copia para área de transferência ;
- **:%s/string1/string2/g** - Substitui "string1" por "string2" ;

Por fim teste os comandos de Execução**5) Comandos para salvar o texto:**

- **:wq ou :x** - Salvam o arquivo e saem do editor
- **:w nome_do_arquivo** - Salva o arquivo corrente com o nome especificado
- **:w! nome_do_arquivo** - O mesmo que :w, mas forçando sobrescrita
- **:q** - Sai do editor
- **:q!** - Sai do editor sem salvar as alterações realizadas



Algumas dicas de vi para a lpi:

:setnumber

:syntax on

:noai

:set hlsearch

:set background=dark

7.6. Exercício Teórico

1) Como removo as linhas 40 a 60?

2) Como vou para o início do arquivo?

3) Como apago as 8000 primeiras linhas ?

4) Como vou para a linha 25 do arquivo?

5) Qual a sequência de teclas que apaga 4 linhas, sendo elas a linha atual e mais 3 linhas abaixo do cursor?

6) Como faço uma cópia do meu arquivo para a seguinte path: /tmp/backup?

7) Como eu copio 5 linhas, e depois colo?

8) Como eu acho todas as palavras ``linux" dentro do texto?

9) Um arquivo tem 620 linhas, eu quero ir para a última linha do arquivo.

10) Substituir todos os /, por @ dentro do arquivo?

7.7. Laboratório

- 1) Crie um arquivo chamado dados_pessoais;
- 2) Dentro dele, coloque um dado em cada linha: Nome completo, e-mail e um site de sua preferência. Grave o arquivo;
- 3) Agora, modifique algum dado e tente sair do editor sem salvar.;
- 4) De dentro do Vi, faça uma cópia desse arquivo para dados_pessoais2;
- 5) Faça uma cópia da sua linha de e-mail 5 vezes;
- 6) Posicione o cursor na primeira linha do arquivo, sem usar as teclas direcionais;
- 7) Vá direto para a linha 3, sem usar as teclas direcionais;
- 8) Vá direto para o final do arquivo, sem usar as teclas direcionais;
- 9) Recorte a linha do seu nome, coloque-a no final do arquivo;
- 10) Apague a linha que tem seu nome e depois tente desfazer a sua última ação;
- 11) Faça uma cópia do arquivo /etc/inittab para o seu HOME. (cp /etc/inittab ~) ;
- 12) Ainda no arquivo, entre no modo visual.;
- 13) Apague as 5 primeiras linhas do arquivo.;
- 14) Substitua todas as string init (minúsculo) por INIT (maiúsculo) ;
- 15) Localize a string ``shutdown", usando o método de busca;
- 16) Substituir das linhas 22 a 28 a string ``wait" pela string ``esperar";
- 17) Dentro do arquivo, localize todas as palavras ``respawn" e apague todas elas;
- 18) Apague as linhas 19 e 24;
- 19) Mova as linhas 35,36,37 e 38 para o final do arquivo;
- 11) Salve o arquivo utilizando um novo nome dentro de /root;