

Sistema Operacional

Unidade 6 – Editando e visualizando textos

Ubuntu 12.04 LTS
Precise Pangolin



QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

INTRODUÇÃO	3
UTILIZANDO OS EDITORES DE TEXTO MODO TEXTO	5
VI.....	5
Modo inserção	6
Modo comando	6
Comandos para alteração de texto	7
Exemplo de utilização do VI	7
NANO (pico)	8
Exemplo de utilização do Nano	9
COMANDOS DE VISUALIZAÇÃO DE CONTEÚDO	10
cat	10
tac	10
head	10
tail	10
nl	11
wc	11
Atributos de wc.....	11
diff	11
split	12
sort	12
(pipe).....	12
more	12
less	13
grep	13
ARQUIVOS IMPORTANTES DO SISTEMA OPERACIONAL	13
Arquivos que estão no diretório “/etc”.....	13
Arquivos úteis que se encontram no diretório “/proc”	13
BIBLIOGRAFIA.....	14

INTRODUÇÃO

A partir de agora vamos aprender a criar arquivos de texto a partir de editores de texto. No Linux temos dois tipos de editores de texto, os editores em modo texto e os editores em modo gráfico.

Os editores em modo gráfico são utilizados geralmente por usuários, já os editores em modo texto serão utilizados por usuários avançados (técnicos) para alterar arquivos de configuração e *scripts* de forma rápida, principalmente através da rede.

Exemplos de editores em modo texto: *vi, vim, nano, emacs, mcedit, ee, elvis, joe, entre outros.*

Vejamos alguns exemplos de editores de texto. A figura 1 mostra o editor de texto “Nano”.

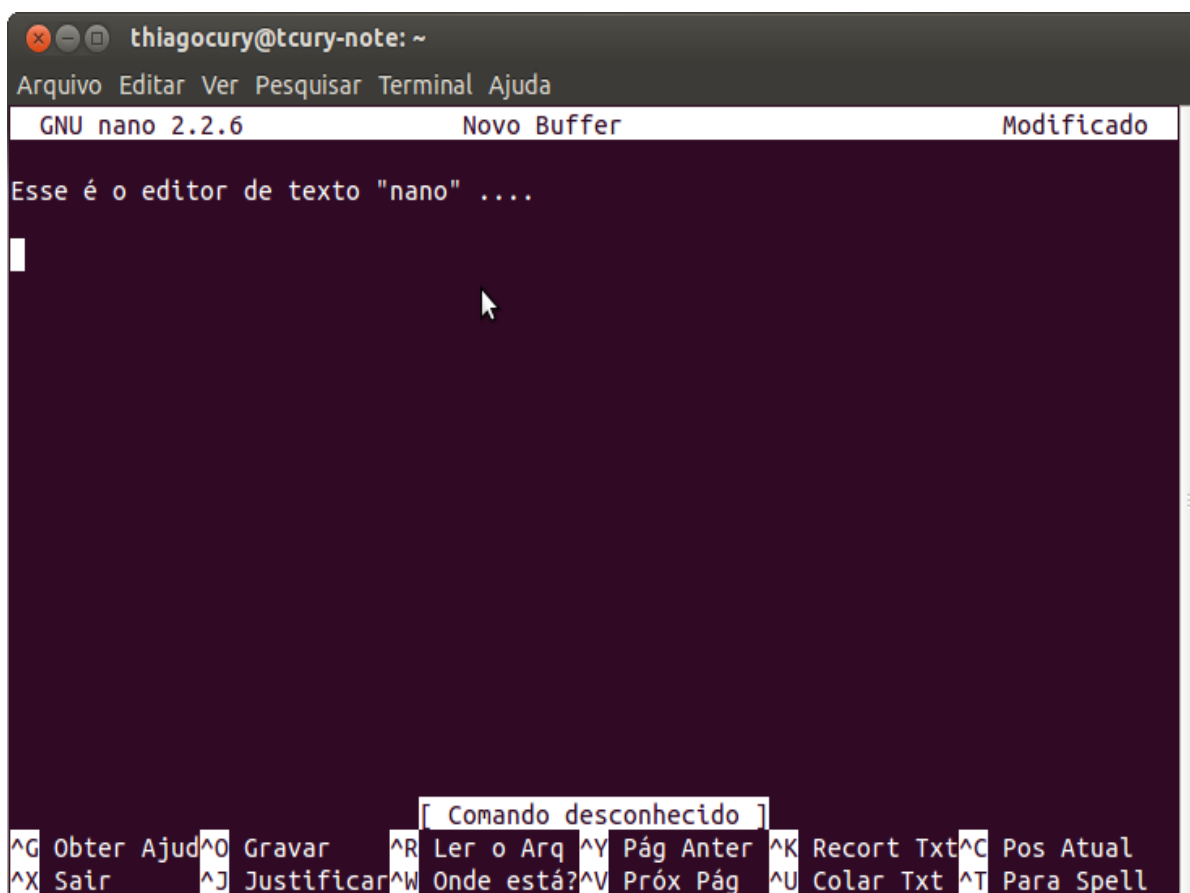


Figura 1 - Editor de Texto Nano

A figura 2 mostra o editor de texto “VI”.

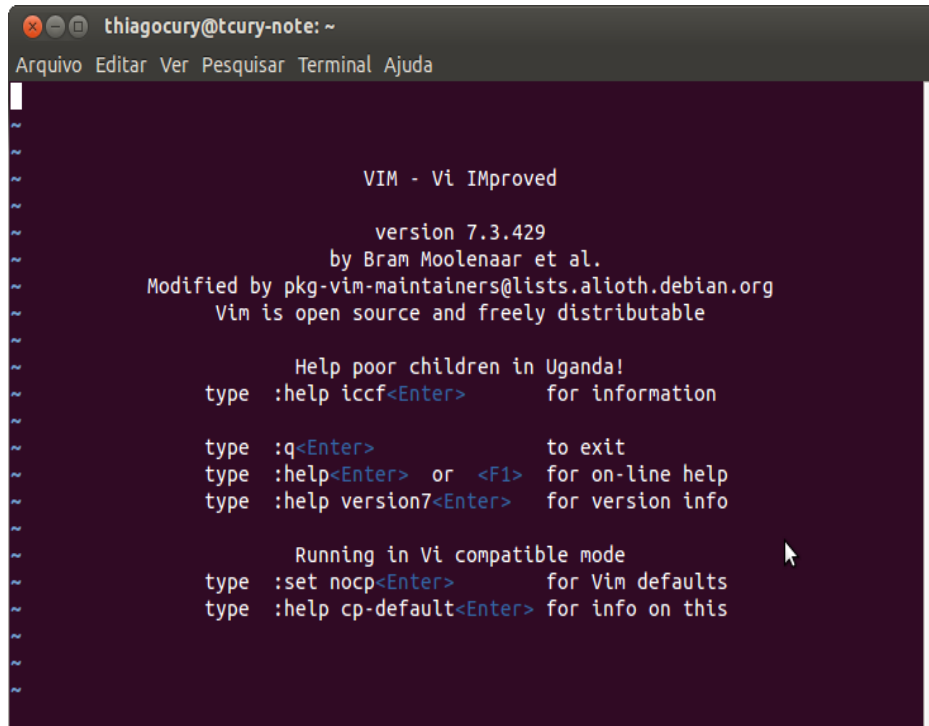


Figura 2 - Editor de Texto VI

Exemplo de editores modo gráfico: *Writer, Kwriter, Gedit, Kedit, entre outros.*

A figura 3 mostra o editor de texto “Gedit”. Ele é similar ao editor de texto bloco de notas (*Notepad*) do sistema operacional Windows.

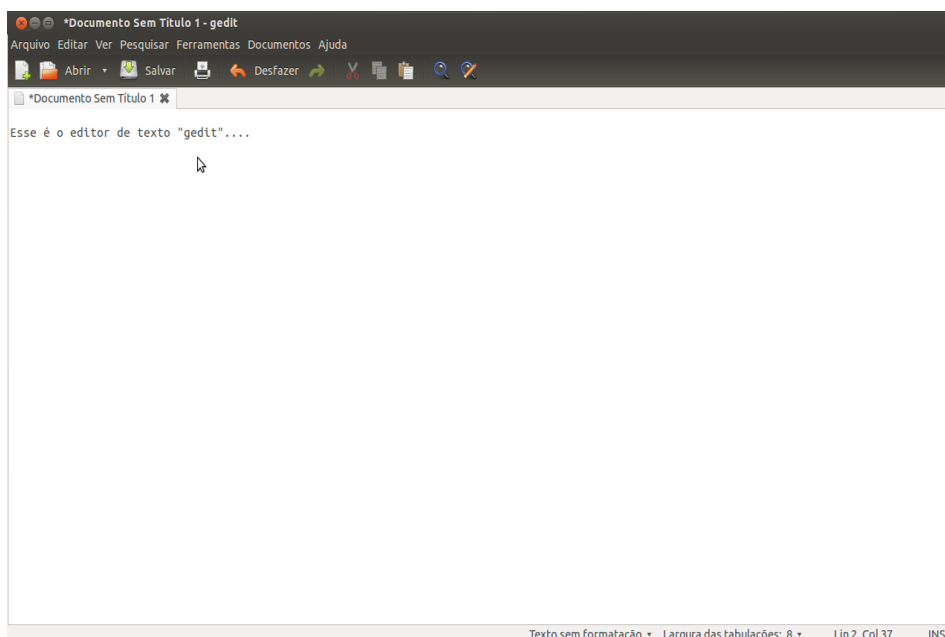


Figura 3 - Editor de texto Gedit

A Figura 4 mostra o editor de texto “Writer”, similar ao editor de texto da suíte de aplicativos da *Microsoft* chamado de *Word*.

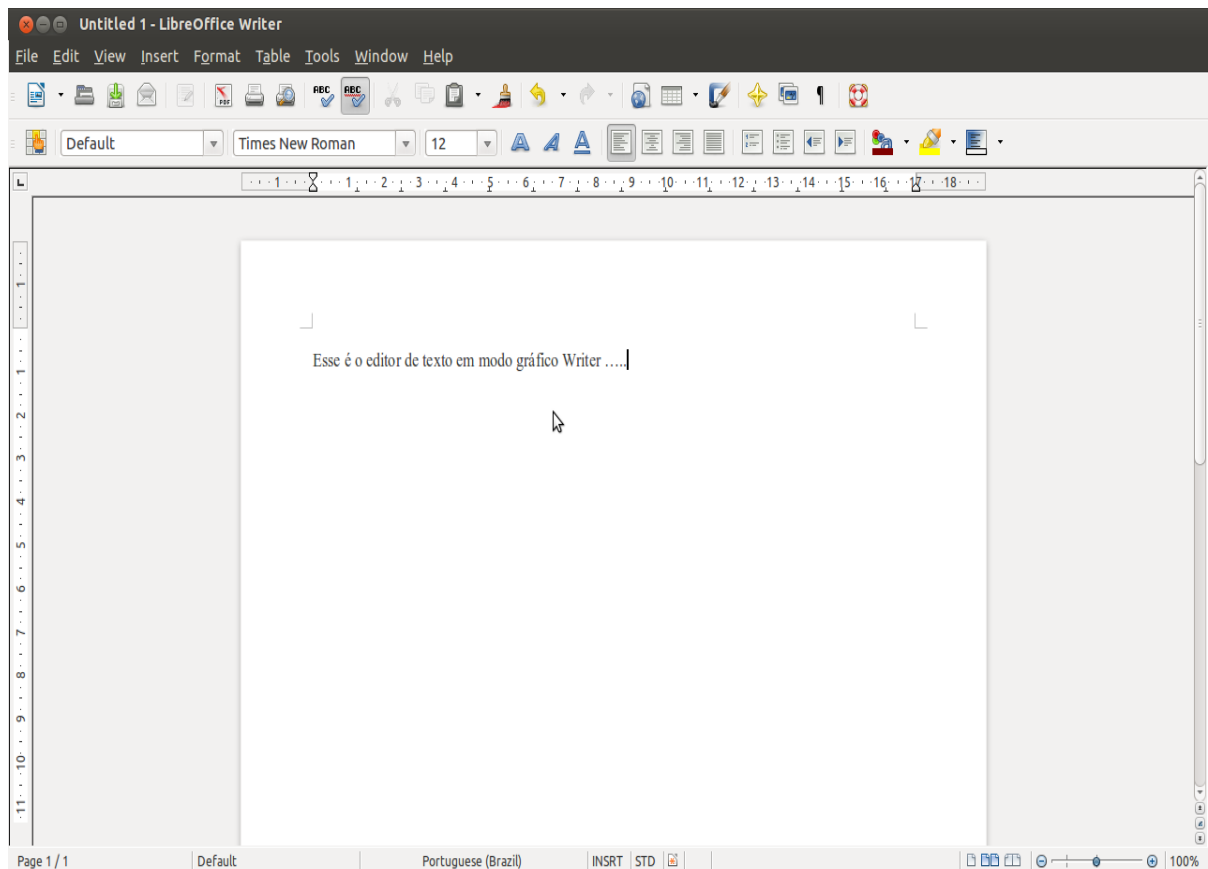


Figura 4 - Editor de texto Writer

UTILIZANDO OS EDITORES DE TEXTO MODO TEXTO

VI

O editor de texto “VI” se popularizou no mundo Linux por estar presente no sistema operacional UNIX. O VI é muito utilizado por usuários Linux e pode ser encontrado em praticamente todas as distribuições.

Para abrir o editor “VI”, abra o terminal e digite “vi”.

O VI é um editor de texto que tem dois modos de trabalho: modo comando e modo inserção.

Modo inserção

Esse modo será utilizado sempre que precisamos inserir um texto. Para mudar para o modo inserção, pressione *esc* e em seguida uma das teclas na tabela logo abaixo:

i	Insere texto antes do cursor
a	Insere texto depois do cursor
r	Insere texto no início da linha onde o cursor está
A	Insere texto no final da linha onde o cursor está
o	Adiciona linha abaixo da atual
O	Adiciona linha acima da atual
ctrl + h	Apaga o último caractere

Após pressionar alguma destas teclas, não acontecerá nada no modo visual do VI, mas você pode digitar o texto que ele já está no modo inserção.

Modo comando

Esse modo serve para inserir comandos para: salvar o arquivo, atualizar um arquivo, sair, sair sem salvar, entre outros.

ATENÇÃO: Ao salvar um arquivo onde o nome precise de espaço, não se esqueça de colocar as “contra barras”.

Abaixo estão os comandos mais básicos do VI. Eles servem para salvar arquivos com os nomes especificados, e sair do editor. Para ativar o modo comando, pressione *esc* e em seguida, um dos comandos abaixo.

Apenas salva o arquivo no diretório corrente

:w nomeDoArquivo

Salva o arquivo atual no arquivo especificado, sobrescrevendo-o

:w! nomeDoArquivo

Apenas sai do editor de texto

:q

Salva o arquivo com o nome especificado e sai do editor

:wq nomeDoArquivo

Sai sem salvar as alterações realizadas no arquivo

:q!

Comandos para alteração de texto

x	Apaga o caractere que está sob o cursor
dw	Apaga a palavra a partir do cursor
dd	Apaga toda a linha onde se encontra o cursor
D	Apaga a linha a partir da posição atual do cursor

SAIBA MAIS: Você pode encontrar outros comandos sobre o editor de texto VI. A lista de comandos é bem extensa.

<http://aurelio.net/curso/material/vim-ref.html>

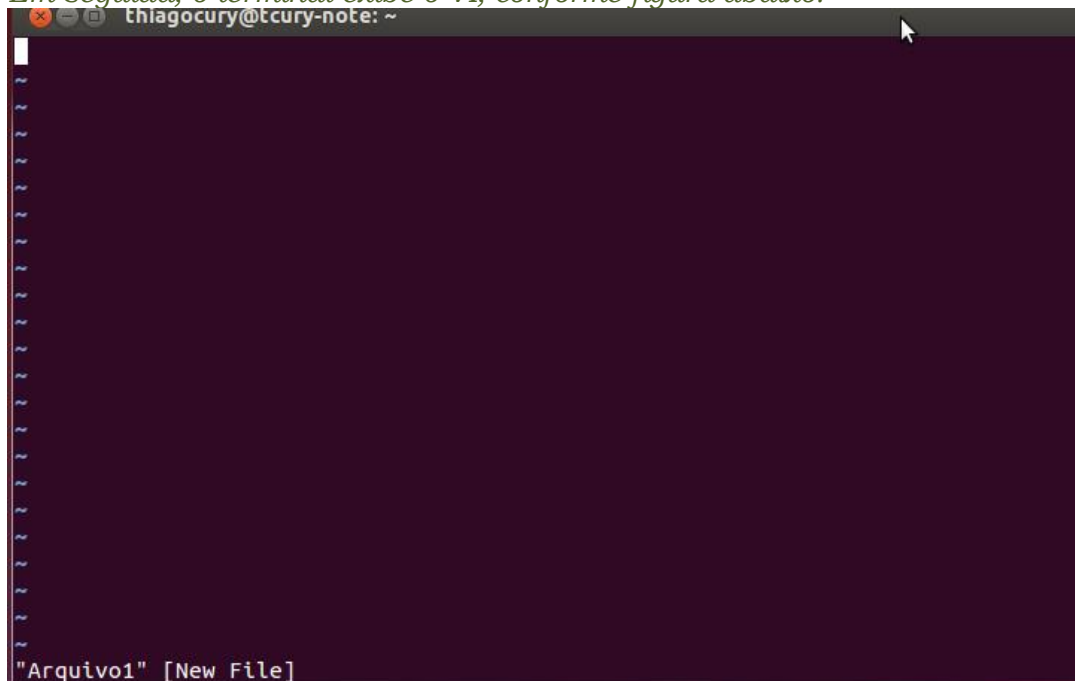
Exemplo de utilização do VI

Vamos criar um arquivo de texto utilizando o editor VI. Para isso, vamos abrir o terminal.

No terminal digite:

```
vi Arquivo1
```

Em seguida, o terminal exibe o VI, conforme figura abaixo.



Digite “i” para ativar o modo inserção e em seguida digite o texto. A figura abaixo mostra um exemplo.

Após digitar o texto, pressione **esc** e digite **:wq** (salva e sai do editor). Você pode visualizar o arquivo criado usando o comando *ls* para listar os arquivos, e depois usando o *cat* você pode visualizar o conteúdo dele, conforme a figura abaixo:

NANO (pico)

Você pode digitar o texto normalmente, porém para salvar ou sair precisamos utilizar teclas de atalhos. O Nano exibe um *menu* no canto inferior da tela, com algumas opções. Por exemplo:

- 

Obs: O acento ^ (circunflexo) significa que devemos apertar a tecla “CTRL”.

Exemplo de utilização do Nano

No terminal, digite:

```
nano Arquivo2
```

Em seguida, o editor abrirá e está pronto para que você digite o texto. A imagem abaixo mostra um exemplo de texto criado no Nano.

```
thiagocury@tcury-note: ~
GNU nano 2.2.6      Arquivo: Arquivo2      Modificado

Digitando um texto no Nano
O Nano é uma outra opção de editor de texto para o modo texto.
Simples de utilizar, abaixo visualizamos a lista de opções.
lembre-se que o ^ representa a tecla CTRL.

^G Obter Ajud ^O Gravar ^R Ler o Arq ^Y Pág Anter ^K Recort Txt ^C Pos Atual
^X Sair ^J Justificar ^W Onde está? ^V Próx Pág ^U Colar Txt ^T Para Spell
```

Quando concluirmos a digitação, pressione CTRL + O para gravar, em seguida confirme pressionando *Enter*. Para sair do editor, use CTRL + X. Use *ls* para listar o diretório e você verá o Arquivo2 na listagem.

```
thiagocury@tcury-note: ~
thiagocury@tcury-note:~$ ls
Área de Trabalho  Arquivo2      Downloads      Imagens  Música  Vídeos
Arquivo1          Documentos    examples.desktop  Modelos  Público
thiagocury@tcury-note:~$
```

COMANDOS DE VISUALIZAÇÃO DE CONTEÚDO

Após criar um arquivo de texto, podemos visualizar o conteúdo do arquivo diretamente no terminal, sem a necessidade de abrir um editor. Para isso podemos utilizar os comandos de visualização, que estão listados a seguir.

cat

Serve para visualizar todo conteúdo do arquivo.

Exemplo – visualizar o conteúdo do Arquivo1

```
cat Arquivo1
```

tac

Serve para visualizar todo conteúdo do arquivo, porém de forma invertida. Note que o comando “tac” é literalmente a inversão do comando “cat”.

Exemplo – visualizar o conteúdo do Arquivo1, invertido.

```
tac Arquivo1
```

head

Com esse comando podemos visualizar o topo do arquivo, ou seja, por padrão ele mostra as 10 primeiras linhas do arquivo de texto. Mas podemos modificar e pedir para mostrar só a 1ª linha do arquivo, por exemplo.

Exemplo – visualizando somente a primeira linha do Arquivo1

```
head -1 Arquivo1
```

tail

Com esse comando podemos visualizar a “cauda” do arquivo, ou seja, as 10 últimas linhas do arquivo de texto. Assim como o comando *head*, podemos modificar e pedir para mostrar só as três primeiras linhas do arquivo, por exemplo.

Exemplo – visualizando somente a última linha do Arquivo1

```
tail -1 Arquivo1
```

nl

Esse comando exibe o conteúdo do arquivo de texto, numerando todas as linhas.

Exemplo: visualizando o Arquivo1 com as linhas numeradas.

```
nl Arquivo1
```

wc

Lista, na ordem, o número de linhas, palavras e caracteres de um arquivo de texto.

Atributos de wc

-m	Exibe somente o número de caracteres do arquivo indicado
-l	Exibe quantas linhas tem o arquivo de texto.
-w	Realiza a contagem de quantas palavras tem no arquivo de texto.

Exemplo

```
wc Arquivo1
```

A saída do comando mostrará três números seguidos do nome do arquivo em questão. O **primeiro** número significa o número de linhas, o **segundo** número significa o número de palavras que tem no arquivo em questão, e o **terceiro** número significa o número de caracteres que tem no arquivo de texto.

Exibe quantas linhas tem o arquivo de texto.

```
wc -l Arquivo1
```

diff

Serve para diferenciar o conteúdo entre dois arquivos de texto, mostrando somente o que tem em um e não tem no outro.

Exemplo: compara o Arquivo1 e o Arquivo2, mostrando o conteúdo diferente entre eles, primeiramente o que tem no Arquivo1 e não tem no Arquivo2, e depois o que tem no Arquivo2 e não tem no Arquivo1

```
diff Arquivo1 Arquivo2
```

split

Serve para dividir um arquivo de texto. Quando dividimos um arquivo, ele criará um arquivo resultante que inicie com o prefixo “x” iniciando os nomes, seguidos de um sufixo “aa”, “ab”, “ac”, etc. Vejamos alguns exemplos:

Divide um arquivo chamado Arquivo1 em arquivos contendo no máximo duas linhas.

```
split -l 2 Arquivo1
```

Divide um arquivo chamado Arquivo2 em arquivos contendo no máximo 18 bytes.

```
split -b 18 Arquivo2
```

sort

Serve para ordenar a saída de um arquivo de texto. Ele consegue ordenar por número ou por caractere.

Mostra o conteúdo do arquivo ordenando as linhas em ordem alfabética

```
sort Arquivo2
```

| (pipe)

O *pipe* é utilizado para redirecionar as entradas e saídas dos comandos. Podemos utilizar o *pipe* em conjunto, por exemplo, com o comando “*more*”, “*less*”, “*grep*” e outros.

more

O comando *more* serve para mostrar um arquivo paginado, de uma forma que possamos visualizar linha a linha ou página a página. Para visualizar linha a linha é só ficar pressionando a tecla *enter*. Se desejar visualizar de página em página é só ir pressionando a barra de espaço. Vejamos alguns exemplos de uso do *more*.

Exibe o arquivo cpuinfo do diretório /proc paginado

```
cat /proc/cpuinfo |more
```

Exibe o histórico de comandos utilizados no terminal de forma paginada

```
history |more
```

less

O comando *less* é similar ao *more*, porém este ao invés de visualizar linha a linha ele permite a navegação no arquivo, permitindo ir para esquerda, direita, para baixo e para cima utilizando as setas de direção.

Obs: Para **sair** do comando *less*, precisamos apertar a tecla “q”.

Exemplos

```
history |less
cat /proc/cpuinfo |less
```

grep

Serve para realizar filtros na saída padrão de algum comando ou no conteúdo de algum arquivo de texto. Vejamos alguns exemplos.

Filtra o arquivo passwd buscando por um usuário que comece com a palavra root.

```
grep ^root /etc/passwd
```

Mostra todo arquivo passwd através do comando cat, filtrando pelo usuário root.

```
cat /etc/passwd |grep root
```

ARQUIVOS IMPORTANTES DO SISTEMA OPERACIONAL

Arquivos que estão no diretório “/etc”.

/etc/passwd	Contem informações dos usuários.
/etc/group	Arquivo onde são armazenados todos os grupos do sistema
/etc/issue	Verifica a versão da distribuição em uso
/etc/shadow	Arquivo com as senhas dos usuários criptografadas.

Arquivos úteis que se encontram no diretório “/proc”

/proc/meminfo	Informações sobre a memória
/proc/devices	Informações sobre dispositivos instalados
/proc/version	Versão do <i>kernel</i>
/etc/shadow	Arquivo com as senhas dos usuários criptografadas.
/proc/cpuinfo	Informações sobre o processador
/proc/interrupts	Interrupções do <i>hardware</i>

BIBLIOGRAFIA

JARGAS, Aurélio Marinho. **Editor VI - Guia de Referência**. Disponível em <http://aurelio.net/curso/material/vim-ref.html>. Acesso em agosto de 2012.

SHIELDS, Ian. Aprenda Linux, 101: **Filtros e fluxos de texto: Manipulando textos na linha de comando usando o GNU textutils**. Disponível em <http://www.ibm.com/developerworks/br/linux/library/l-lpic1-v3-103-2/>. Acesso em agosto de 2012.

SILVA, Arlei Silveira. **Comandos Básicos**. Disponível em <http://wiki.ubuntu-br.org/ComandosBasicos>. Acesso em Agosto de 2012.

SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em <http://www.guiafoca.org/>. 2010.