

Disciplina: Sistema Operacional
Prof. Leandro Pisco

Introdução

Um sistema operacional, por mais complexo que possa parecer, é apenas um conjunto de rotinas executado pelo processador, de forma semelhante aos programas dos usuários. Sua principal função é controlar o funcionamento de um computador, gerenciando a utilização e o compartilhamento dos seus diversos recursos, como processadores, memórias e dispositivos de entrada e saída.

Sem o sistema operacional, um usuário para interagir com o computador deveria conhecer profundamente diversos detalhes sobre hardware do equipamento, o que tornaria seu trabalho lento e com grandes possibilidades de erros. O sistema operacional tem como objetivo funcionar como uma interface entre o usuário e o computador, tornando sua utilização mais simples, rápida e segura.

A grande diferença entre um sistema operacional e aplicações convencionais é a maneira como suas rotinas são executadas em função do tempo. Um sistema operacional não é executado de forma linear como na maioria das aplicações, com início, meio e fim. Suas rotinas são executadas concorrentemente em função de eventos assíncronos, ou seja, eventos que podem ocorrer a qualquer momento.

O nome sistema operacional, apesar de ser o mais empregado atualmente, não é o único para designar esse conjunto de rotinas. Denominação como monitor, executivo, supervisor ou controlador possuem, normalmente, o mesmo significado.

Kernel

O Kernel é o núcleo do sistema. É responsável pelas operações de baixo nível tais como: gerenciamento de memória, gerenciamento de processos, suporte ao sistema de arquivos, periféricos e dispositivos. Pode ser compilado para se adequar melhor a um determinado tipo de máquina.

Shell

O Shell é o elo entre o usuário e o sistema. Imagine o Shell como sendo um intérprete entre pessoas que falam línguas diferentes. Ele traduz os comandos digitados pelo usuário para a linguagem usada pelo kernel e vice-versa. Sem o Shell a interação entre usuário e o kernel seria bastante complexa. Ele também é uma linguagem de programação completa.

Script

Script é um arquivo que contém comandos do shell que em uma situação normal poderiam ser executados a partir do prompt. Esses comandos são executados sequencialmente dependendo de condições estruturais e de laço como "if ... then" (se ... então) e "do while" (faça enquanto). Um script também

pode usar variáveis. A entrada pode ser recebida através do prompt de comando ou de outros arquivos.

1 - História do Linux

Origem no UNIX que foi desenvolvido nos anos 70.

O UNIX é muito utilizado hoje no ambiente corporativo, educacional e agora no ambiente desktop.

Linux é uma cópia do Unix feito por Linus Torvalds, junto com um grupo de hackers pela Internet. Pretende-se que ele siga conforme com o padrão POSIX, padrão usado pelas estações UNIX.

Linus Torvalds que na época era um estudante de ciência da computação na Finlândia criou um clone do sistema Minix (sistema operacional desenvolvido por Andrew Tannenbaun que era semelhante ao UNIX) que o chamou de Linux. Hoje ele ainda detém o controle do Kernel do sistema.

Em março de 1992 surge a versão 1.0 do Linux.

Estima-se que sua base de usuários se situe hoje em torno de 10 milhões.

Ele ainda não se enquadra como sendo um sistema operacional UNIX.

Ele inclui proteção entre processos (crash protection), carregamento por demanda, redes TCP/IP, além de nomes de arquivos com até 255 caracteres, multi-tarefa real, suporte a UNICODE, shared libraries, memória virtual, etc.

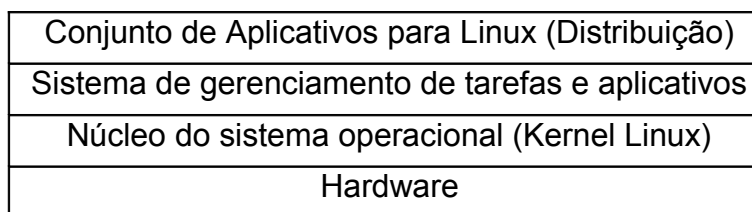
1.1 Linux como sistema operacional

Linux se refere ao Kernel

O conjunto de aplicativos que são executados no Kernel se chamam distribuição.

A função do Kernel é ser a interface entre o Hardware e os sistemas de gerenciamento de tarefas e aplicativos.

O Kernel do Linux é mais leve que o de outros sistemas operacionais para servidores. São 12 milhões de linhas de código, enquanto outras versões do Unix têm 30 milhões e o Windows NT, 50 milhões. "Isso significa que o Linux é três vezes mais rápido que outros Unix e até cinco vezes mais rápido que o Windows NT".



1.2 Características do Sistema

- Multitarefa (Várias aplicações podem ser executadas ao mesmo tempo)
- Multiusuário (Vários usuários podem utilizar o sistema ao mesmo tempo)

- É gratuito, atualizações frequentes e é desenvolvido voluntariamente por programadores experientes e colaboradores que visam a constante melhoria do sistema.
- Convive harmoniosamente no mesmo computador com outros sistemas operacionais
- Não exige um computador potente para rodar
- Não é necessário licença para o seu uso.
- Maior estabilidade em relação ao Windows
- Maior confiabilidade.
- Não precisa ser reinicializado devido a instalação de programas ou configuração de periféricos.
- Acessa discos formatados por outros sistemas operacionais
- Suporte a linguagens de programação.
- Não existem vírus no linux
- Roda aplicações windows através do WINE
- Ambiente Shell
- Suporte a diversos dispositivos e periféricos disponíveis no mercado
- Código fonte aberto, isso significa que se você for um programador, pode modificá-lo para se adequar a necessidades específicas, algo impossível de ser conseguido com o Windows.

Termos de Licenciamento

Inicialmente, Torvalds lançou o **Linux** sob uma licença que proibia qualquer uso comercial. Isso foi mudado de imediato para a Licença Pública Geral GNU. Essa licença permite a distribuição e mesmo a venda de versões possivelmente modificadas do **Linux** mas requer que todas as cópias sejam lançadas dentro da mesma licença e acompanhadas do código fonte.

GNU General Public License

GNU General Public License (Licença Pública Geral), **GNU GPL** ou simplesmente **GPL**, é a designação da licença para software livre idealizada por Richard Stallman no final da década de 1980, no âmbito do projecto GNU da Free Software Foundation (FSF).

A GPL é a licença com maior utilização por parte de projectos de software livre, em grande parte devido à sua adoção para o Linux

Multiusuário: Permite que vários usuários possam rodar o sistema operacional, e não possui restrições quanto à licença. Permite vários usuários simultâneos, utilizando integralmente os recursos de multitarefa. A vantagem disso é que o Linux pode ser distribuído como um servidor de aplicativos. Usuários podem acessar um servidor Linux através da rede local e executar aplicativos no próprio servidor.

Multiplataforma: O Linux roda em diversos tipos de computadores, sejam eles RISC ou CISC.

Multitarefa: Permite que diversos programas rodem ao mesmo tempo, ou seja, você pode estar imprimindo uma carta para sua vovó enquanto trabalha na planilha de vendas, por exemplo. Sem contar os inúmeros serviços disponibilizados pelo Sistema que estão rodando em background e você provavelmente nem sabe.

Multiprocessador: Permite o uso de mais de um processador. Já é discutida, há muitos anos, a capacidade do Linux de poder reconhecer mais de um processador e inclusive trabalhar com SMP, clusters de máquinas, na qual uma máquina central controla os processadores das outras para formar uma só máquina.

Protocolos: Pode trabalhar com diversos protocolos de rede (incluindo o TCP/IP que é nativo Unix).

Processo - Resumidamente... um programa em execução. As informações sobre vários processos em execução ficam armazenadas na tabela de processos. O Sistema Operacional controla o tempo de execução de cada processo, e ele é responsável por iniciar, parar/suspender e terminar a execução de processos. Os usuários também pode interferir na administração de processos.

Todo processo tem um processo "pai" que o chamou, portanto o processo tem nome, número de identificação, grupo ao qual pertence, etc. e está inserido na árvore hierárquica de processos. O processo pai é chamado de init.

Sistema de arquivos - É uma forma de armazenamento de arquivos em estruturas (na maneira hierárquica) de diretórios. Assim, o usuário não precisa necessitar conhecer detalhes técnicos do meio de armazenamento. Ele apenas precisa necessitar conhecer a estrutura (árvore) de diretórios para poder navegar dentro dela e acessar suas informações

Sistema de cores dos arquivos Linux.

Os arquivos Linux ao ser exibido pelo comando ls, podem exibir uma cor diferente apontando o tipo de cada arquivo.

Cor	Arquivo
Azul escuro	Diretório
Cinza	Arquivos comuns
Branco	Arquivos texto
Azul ciano	link
Verde	Arquivos binários (comandos do linux)
Amarelo	Dispositivos de entrada/saída
Magenta	compactado

Consoles virtuais: Permite que o usuário tenha mais de um console para trabalhar, sendo que em cada console você pode ter diversas tarefas sendo executadas em background e mais em foreground (segundo plano e primeiro plano).

2 - O que é uma distribuição

Ao "kernel" é freqüentemente acrescentado uma série de aplicações, formando um sistema ou distribuição Linux.

Distribuição nada mais é que um pacote do kernel do sistema operacional mais os programas que o acompanham. Este pacote, incluindo as ferramentas necessárias para sua instalação, é chamado de distribuição.

Uma distribuição atende a uma determinada necessidade.

As distribuições podem ser produzidas em diferentes versões do Kernel, podem incluir diferentes conjuntos de aplicativos, utilitários, ferramentas e módulos de driver , e podem oferecer diferentes programas de instalação e atualização para facilitar o gerenciamento do sistema.

Exemplos de distribuição

Red Hat – Famoso por suas ferramentas de instalação e atualização do sistema operacional e por seu sistema bem projetado de instalação, desinstalação e controle de pacotes de aplicativos de software.

Slackware – Era a distribuição mais popular. Não vem com RPM. Ganha em performance, mas peca na interatividade. Usado mais para servidores de rede.

Caldera OpenLink – O OpenLink 1.3 inclui o "K" Desktop Environment, uma licença não-comercial do StarOffice for Linux, Suporte Netware, uma licença do DR-DOS para compatibilidade DOS.

S.u.S.E Linux – É uma conhecida distribuição de Linux, disponível principalmente na Europa e oferecida nas versões em inglês e alemão.

Debian / GNU – Não possui uma organização comercial patrocinadora. É produzida por uma equipe de voluntários. Utiliza seu próprio sistema de gerenciamento de pacotes.

2.1.1 Conta de Acesso

Similar a conta de um banco, a conta de acesso é a maneira pela qual você se identifica no sistema. É através dela que o sistema sabe que você tem direitos de fazer uma coisa ou outra e garantir que somente você alterará seus arquivos. Isso é de suma importância porque dessa forma, você não poderá (mesmo que acidentalmente) danificar o sistema, acessando áreas vitais para o funcionamento do mesmo.

A conta de acesso é composta de Login e Senha. Tanto o login quanto a senha podem ser alfanuméricos, a senha inclusive deveria ser alfanumérica por medidas de segurança. Mesmo que só você utilize o linux, terá que usar uma conta para acessar o sistema, devido ao fato de que essa é uma

característica do próprio linux desde a sua concepção (e de qualquer sistema multiusuário) e por ser ele um sistema voltado para ambientes em rede.

Existe uma conta que tem o controle total do sistema denominada **root**, comumente chamada de *superusuário*. Com ela podemos fazer qualquer coisa que um usuário normal não poderia, inclusive danificar o sistema, se não soubermos exatamente o que estamos fazendo. Por isso todo o cuidado é pouco, e justamente por isso, essa conta não deveria ser usada no dia a dia. Deveria ter sua senha guardada e usada em situações de real necessidade, tais como tarefas de configuração e administração do sistema.

2.1.2 Sistema de Arquivos

Um sistema de arquivos é a maneira pela qual os dados são organizados, armazenados e controlados em qualquer mídia além de ser a forma como o sistema operacional representa essas informações. Ele é criado durante a formatação de uma partição. O sistema de arquivos mais utilizado no linux é o ext (extended filesystem). Podemos representar o sistema de arquivos do linux segundo a seguinte estrutura:

/ - (todos os arquivos e diretórios do sistema)

/boot – (arquivos de inicialização)

/bin – (executáveis principais)

/dev – (referência aos dispositivos)

/etc – (executáveis essenciais ao sistema, arquivos de configuração)

/usr – (comandos, bibliotecas, aplicativos)

/lib – (biblioteca básica do sistema)

/tmp – (arquivos temporários)

/home – (diretório de usuários e suas configurações)

O prompt

Quando o prompt que preceder de um comando for \$, o comando pode executado por qualquer usuário.

Quando o prompt que preceder um comando for #, o comando pode ser executado somente pelo super usuário o Root.

Comandos

No MS-DOS os comandos não poderiam ser criados pelos usuários, ou sejam, eram limitados e geralmente estáticos.

No mundo Unix e por extensão, no Linux, o conceito é diferente. Um comando é qualquer arquivo executável.

Comandos internos.

São comandos que estão dentro de um shell interpretador de comandos. Quando o shell é carregado na memória, seus comandos ficam residentes nela. A grande vantagem dos comandos internos é a velocidade, pois eles não precisam ser procurados no disco rígido, nem criam processos.
Exemplo: cd, alias, kill, logout, etc.

Comandos externos.

São comandos que estão localizados em diretórios específicos no disco rígido, como /bin e o /sbin. O Linux precisa consultar o disco rígido sempre que alguns desses comandos forem solicitados. A maioria dos comandos do Linux são externos.

Exemplo: ls, cp, rm, mv, mkdir, rmdir, etc.

Comandos:

Comando:	Clear
Descrição:	Permite limpar a tela de informações desnecessárias no momento.
Sintaxe:	Clear
Opções:	nenhuma
Exemplo:	Clear
Comando:	Login
Descrição:	Cancela a sessão atual e inicia nova sessão de usuário
Sintaxe:	login
Opções:	Nenhum
Exemplo:	Login
Comando:	Logout
Descrição:	Termina a sessão do usuário
Sintaxe:	Logout
Opções:	nenhuma
Exemplo:	Logout

Comando:	Exit
Descrição:	Encerra o shell de comandos correntes e quando aberto no ambiente gráfico
Sintaxe:	Exit
Opções:	Nenhuma
Exemplo:	Exit
Comando:	Halt
Descrição:	Usado antes de desligar o computador
Sintaxe:	Halt
Opções:	Nenhuma
Exemplo:	halt
Comando:	Man
Descrição:	Consulta os manuais on-line do sistema (também chamadas páginas de manual)
Sintaxe:	Man opções seção comando
Opções:	-a exibe todas as páginas de manual combinando um comando específico -h exibe uma mensagem de ajuda e sai -w mostra a localização de páginas de manual a serem exibidas
Exemplo:	Man ls Obs: Para sair do man tecla (Q)
Comando:	Cd
Descrição:	Muda o diretório de trabalho
Sintaxe:	Cd diretório
Opções:	Nenhuma
Exemplo:	Cd /usr/bin (vai para o diretório /usr/bin) Cd .. (sobe um degrau – nível da árvore de diretórios) Cd / (vai para o diretório raiz) Cd (retorna ao diretório pessoal do usuário)
Comando:	Pwd
Descrição:	Mostra o diretório atual
Sintaxe:	Pwd
Opções:	Nenhuma
Exemplo:	Pwd
Comando:	Ls
Descrição:	Lista o conteúdo dos diretórios
Sintaxe:	Ls opções diretório
Opções:	-a lista arquivos ocultos (todos os arquivos e diretórios ocultos começam com ponto(.)) -l faz a listagem usando o formato longo dos arquivos.

	<p>Usando o formato longo são mostrados informações sobre permissões de arquivos e diretórios</p> <ul style="list-style-type: none"> -t ordena por data de alteração. Normalmente os arquivos são listados em ordem alfabética. -r reverte a ordenação para obter ordem alfabética inversa. -R recursiva, lista também o conteúdo dos subdiretórios abaixo do especificado. -d diretório -F diferencia os tipos de arquivos concatenando caracteres aos seus nomes: / diretórios * arquivos executáveis @ links simbólicos ~ arquivos de backup = sockets pipe
Exemplo:	<pre>Ls Ls /usr/bin Ls -F Ls -l Ls -aF Ls -tl</pre> <p>Obs: É possível fazer combinações com as opções.</p>
Comando:	Touch
Descrição:	Atualiza a data de acesso ao arquivo. Caso o arquivo não exista, o touch irá criá-lo vazio por default (padrão)
Sintaxe:	Touch opções arquivo
Opções:	<ul style="list-style-type: none"> -c não cria o arquivo caso ele não exista. -m muda somente a hora de modificação -a muda somente a data de acesso.
Exemplo:	Touch arquivo1.doc
Comando:	Rm
Descrição:	Remove arquivos
Sintaxe:	Rm arquivo
Opções:	<ul style="list-style-type: none"> -f não exibe mensagens de confirmação para sobrescrever arquivos. -i pede uma confirmação para sobrescrever o arquivo destino, caso já exista um com o mesmo nome. -r apaga recursivamente o conteúdo de um diretório. -v exibe o nome de cada arquivo antes de removê-lo.
Exemplo:	Rm arquivo1.doc
Comando:	Mkdir
Descrição:	Cria diretórios (pasta)
Sintaxe:	Mkdir diretório
Opções:	-p cria diretórios pai se ainda não existirem
Exemplo:	Mkdir ITS

Comando:	Rmdir
Descrição:	Remove diretórios vazios
Sintaxe:	Rmdir diretório
Opções:	-p remove os diretórios intermediários se os mesmos estiverem vazios.
Exemplo:	Rmdir ITS

Exercício propostos.

- 1 – Explique o que é um Sistema Operacional?
- 2 – Qual a principal diferença entre um sistema operacional e um sistema convencional?
- 3 – O que é Kernel?
- 4 – O que é Shell?
- 5 – O que é Script?
- 6 – De onde originou o Linux? E quem o idealizou?
- 7 – Cite 3 características do Linux como sistema operacional.
- 8 – O que seria o GNU?
- 9 – O que é Multiusuário?
- 10 – O que é Multitarefa?
- 11 – O que é multiprocessador?
- 12 – O que são processos?
- 13 – Dados os tipos de arquivos abaixo, cite sua cor correspondente no sistema de cores.
A) Executáveis
B) Dispositivos I/O
C) Diretório
D) Link
- 14) O que é uma distribuição Linux? Dê 2 exemplos
- 15) O que é conta e qual a sua função?
- 16) Defina o que armazenamos nos diretórios padrão do Linux.

/boot -

/bin -

/dev -

/etc -

/usr -

/lib -

/tmp -

/home -

17) Qual a função do Prompt de comando?

18) Defina: Comandos internos e externos

PRÁTICA

- 1) Limpar a tela
- 2) Listar os arquivos ocultos
- 3) Listar os arquivos ordenados por data de criação
- 4) Listar os arquivos e seus atributos
- 5) Listar os arquivos ocultos e em ordem.
- 6) Listar os arquivos recursivamente.
- 7) Consultar o manual on line dos comandos clear e ls
- 8) Acessar a pasta raiz
- 9) Listar o conteúdo da pasta raiz.
- 10) Acessar as pastas do Raiz e listar o seu conteúdo.
- 11) Acessar a pasta de trabalho do usuário
- 12) Criar uma pasta chamada ITS
- 13) Criar um arquivo chamado prova
- 14) Apagar o arquivo prova
- 15) Sair da pasta ITS
- 16) Apagar a pasta ITS

Comando:	CP
Descrição:	Copia ou ou mais arquivos
Sintaxe:	Cp opções /caminho/arquivo-origem caminho/arquivo-destino
Opções:	-a preserva todos os atributos de arquivo -b faz cópia de backup antes de copiar -d copia um link mas não o caminho apontado pelo mesmo -i pede confirmação antes de sobrescrever o arquivo -r ou -R copia recursivamente arquivos em todos os subdiretórios -u copia apenas quando o arquivo que está sendo copiado é mais novo do que o destino -v mostra o nome do arquivo antes de copiá-lo
Exemplo:	Cp /exercícios/arquivo1.doc /exercícios/bakcup

Comando:	MV
Descrição:	Move ou renomeia arquivos e diretórios
Sintaxe:	MV opções /caminho/arquivo-origem caminho/arquivo-destino
Opções:	-b faz cópia de backup antes de copiar -i pede confirmação antes de sobrescrever o arquivo -v mostra o nome do arquivo antes de copiá-lo
Exemplo:	MV exercícios/arquivo1.doc /exercícios/bakcup

Comando:	LN
Descrição:	Cria links para arquivos ou diretórios. LN cria tanto links simbólicos como diretos.
Sintaxe:	LN opções caminho nome do link
Opções:	-s cria link smbólico
Exemplo:	Ls /exercícios/prova link1.doc

Terminais Virtuais.

Terminal virtual é um equipamento constituído de teclado e monitor, ligado a um mainframe, não sendo capaz de realizar nenhum processamento, senão permitir o acesso do seu operador ao sistema, entrada de dados e visualização de resultados de algum programa (processo) pelo mesmo.

O Linux suporta vários usuários logados no sistema ao mesmo tempo por meio de terminais remotos. Além disso, o Linux possui noção de terminal virtual o que permite que um usuário esteja logado em mais de uma seção simultaneamente.

Para mudar de terminal virtual use: Alt + Tecla de Função
(Teclas de função: F1 a F6)

Para mudar de terminal virtual no ambiente gráfico(X-Window) use: CTRL + ALT + tecla de função
(Teclas de função: F7 a F12)

Comando:	Echo
Descrição:	Escreve no terminal
Sintaxe:	Echo opções texto
Opções:	-e habilita a interpretação de caracteres escape no texto
Exemplo:	Echo 'Rezende Ramel'

Comando:	Banner
Descrição:	Escreve com letras grandes
Sintaxe:	Banner opções texto
Opções:	-w n – onde n é a largura das letras
Exemplo:	Banner 'Rezende'

Comando:	Date
Descrição:	Exibe a data e hora atuais, ou define nova data e hora
Sintaxe:	Date [MMDDHHmmAAAA] MM mês DD dia HH hora Mm minuto AAA ano
Opções:	Nenhum
Exemplo:	Date

Comando:	Cal
Descrição:	Mostra o calendário do mês correspondente a data atual do sistema, ou de um mês e ano especificados pelo usuário.
Sintaxe:	Cal mês ano
Opções:	Nenhum
Exemplo:	Cal 2008
Comando:	Cat
Descrição:	Concatena e mostra arquivos
Sintaxe:	Cat opções arquivo
Opções:	-A mostra todos os caracteres especiais -b numera apenas as linhas não brancas -t mostra tabulações.
Exemplo:	Cat arquivo Cat > teste1.doc (permite criar o arquivo teste1.doc utilizando redirecionamento(>)
Comando:	More
Descrição:	Mostra arquivos texto (por página- enter para mudar de página e Q para sair)
Sintaxe:	More opções arquivo
Opções:	-s reduz múltiplas linhas em branco a uma única linha em branco
Exemplo:	More arquivo1.doc
Comando:	Less
Descrição:	Mostra arquivo texto (podendo navegar dentro do arquivo usando as teclas direcionais e Q para sair)
Sintaxe:	Less opções arquivo
Opções:	-s reduz múltiplas linhas em branco a uma única linha em branco
Exemplo:	Less arquivo1.doc
Comando:	Head
Descrição:	Mostra na saída padrão uma quantidade de linhas de texto do início do arquivo que recebe como entrada
Sintaxe:	Head opções arquivo
Opções:	-N (onde N é um número) mostra as N primeiras linhas de um arquivo. -b N (onde N é um número) mostra os primeiros N blocos do arquivo -c N (onde N é um número) mostra os primeiros N caracteres (bytes) do arquivo.
Exemplo:	Head -5 /etc/lilo.conf
Comando:	WC

Descrição:	Conta palavras dos dados da entrada padrão e apresenta o resultado na saída padrão
Sintaxe:	Wc opções arquivo
Opções:	-l conta as linhas -w conta as palavras -c conta os caracteres
Exemplo:	Wc (Conta as linhas, palavras e letras) Wc -l exercício.doc

Comando:	Sort
Descrição:	Ordena os dados de entrada, escrevendo na saída.
Sintaxe:	Sort opções arquivo
Opções:	-r inverte a ordem de classificação, de ascendente para descendente
Exemplo:	Cat > alunos joao ana gustavo bruno marcos ctrl + c sort alunos

Comando:	Diff
Descrição:	Mostra a diferença entre dois arquivos texto
Sintaxe:	Diff opções arquivo1 arquivo2
Opções:	-a considera todos os arquivos como texto -b ignora seqüências de espaços e caracteres de tabulação
Exemplo:	Diff alunos1 alunos2

Editor de Textos VI ou VIM

"vi" é a sigla para "Visual Interface". A origem desse nome se deve ao seguinte fato: quando o vi foi criado (começo da década de 80), não era comum existirem editores de textos como nos dias de hoje. Naquela época, você digitava um texto mas não podia vê-lo! Isso mesmo! Em 1992, foi criado o vim (Vi IMitator), um clone fiel ao vi, porém com muitas outras funcionalidades, que só foram sendo adicionadas. Algum tempo depois, o vim passou a ser chamado de `Vi IMproved' (vi melhorado).

O vim é um dos editores de textos mais utilizados no mundo Unix. Em alguns sistemas, existe um link simbólico (/bin/vi) apontando para o /usr/vim. Em outros, o /bin/vi é o executável, só que executa diretamente o vim. Muita gente acha que usa vi, mas na verdade utiliza o vim, e eles têm algumas diferenças. O que você verá abaixo fala sobre o vim.

O vim é um editor de textos muito poderoso, ele pode: abrir vários arquivos ao mesmo tempo, possui sistema de autocorreção, auto-identificação, seleção visual, macros, seleção vertical de texto, uso de expressões regulares, sintaxe

colorida, e muito mais. Ele não é exclusivo do Unix, ou seja, pode ser executado em outras plataformas, como Amiga, MacOS, Sun, Windows entre outras.

Existe também o gvim, que é o vim em modo gráfico, com todas as funcionalidades do vim em pleno funcionamento, o que muda é apenas o modo gráfico mesmo.

O vim possui vários modos, ou seja, estados em que ele se encontra. São eles: modo de inserção, comandos, linha de comando, visual, busca e reposição. Abordarei os dois principais:

Modo de inserção e de comandos

Para identificar o modo (estado) do vim, basta visualizar o rodapé da tela.

Agora, vamos à prática. Para executar o vim, utilize:

\$ vi => Abre o vim vazio, sem nenhum arquivo e exibe a tela de apresentação.

\$ vi arquivo => Abre o arquivo de nome "arquivo".

\$ vi arquivo + => Abre o arquivo de nome "arquivo", com o cursor no final do mesmo.

\$ vi arquivo +10 => Abre o arquivo de nome "arquivo", com o cursor na linha 10.

\$ vi arquivo +/Copag => Abre o arquivo de nome "arquivo", na primeira ocorrência da palavra "Copag".

Ao executar o vim, ele inicia diretamente em modo de comando. Para comprovar, é só olhar na última linha (rodapé) e não vai haver nada lá. Isso quer dizer que você não conseguirá escrever nada, pode digitar a vontade que só vai ouvir beeps. Para começar a escrever, pressione "i" em seu teclado. O vim entra em modo de inserção, que você comprova (como falado anteriormente) pelo rodapé da tela, onde fica a seguinte

marcação:

```
-- -- INSERT --
```

Suponha que você já digitou o bastante, e quer salvar, por segurança. Pressione a tecla ESC para voltar em modo de comandos. E veja os comandos para salvar/sair:

:w => Salva o arquivo que está sendo editado no momento.

:q => Sai.

:wq => Salva e sai.

:x => Idem.

ZZ => Idem.

:w! => Salva forçado.

:q! => Sai forçado.

:wq! => Salva e sai forçado.

Então, você editou uma boa quantidade de textos e quer salvar:

:w

Agora, quer voltar a editar o texto:

i

Lembre que utilizando o "i" para inserção, a mesma se inicia inserindo texto antes do cursor. Veja agora outros subcomandos de inserção de texto:

A => Insere o texto no fim da linha onde se encontra o cursor

o => Adiciona uma linha vazia abaixo da linha corrente

O => Adiciona uma linha vazia acima da linha corrente

Ctrl + h => Apaga último carácter à esquerda

Voltando ao modo de comando:

Veja agora subcomandos para movimentação pelo texto:

Ctrl + f => Passa para a tela seguinte.

Ctrl + b => Passa 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 para carácter a esquerda.

j => Move o cursor para linha abaixo.

k => Move o cursor para linha acima.

l => Move o cursor para carácter a direita.

w => Move o cursor para o início da próxima palavra (não ignorando a pontuação).

W => Move o cursor para o início da próxima palavra (ignorando a pontuação).

b => Move o cursor para o início da palavra anterior (não ignorando a pontuação).

B => Move o cursor para o início da palavra anterior (ignorando a pontuação).

0 (zero) => Move o cursor para o início da linha corrente.

^ => Move o cursor para o primeiro carácter não branco da linha.

\$ => Move o cursor para o fim da linha corrente.

nG => Move o cursor para a linha de número "n"

(substitua n pelo número da linha)..

G => Move o cursor para a última linha do arquivo.

Copiando e colando textos no vim (utilizando o mouse)

Selecione o texto necessário com o botão esquerdo do mouse. Quando você for colar, saiba que o texto será colado a partir de onde se encontra o cursor (esse que aparece, às vezes piscando e às vezes não, quando você está digitando). Para colar, depois de ter selecionado o texto, você pode utilizar uma dessas opções:

- 1) Pressionando o botão direito do mouse;
- 2) Pressionando o botão direito + botão esquerdo juntos;
- 3) Pressionando o botão do meio do mouse (mouse de 3 botões);

Observação: Lembre-se que o vim deve estar no modo de inserção.

Usando o modo visual do vim

Entre no modo visual: v

Agora, utilize as teclas direcionais (setas) do teclado, para selecionar o texto desejado.

Pressione e cole, utilizando a tecla "p" (paste).

Veja agora como apagar um determinado texto:

Utilizando normalmente as teclas Backspace/Delete, ou entrando em modo visual (v) e pressionando a tecla Delete.

Você pode remover até o final de uma palavra, utilizando: dw

Pode também remover até o final de uma frase: d\$

Desfazendo uma ação

É claro que você pode desfazer uma ação que você considera errado, ou que errou ao digitar o texto. É só utilizar: u

Se você precisar voltar o texto na tela, utilize as teclas Ctrl + r.

Subcomandos para localização de texto

/palavra => Procura pela palavra ou caracter acima ou abaixo do texto.

?palavra => Move para a ocorrência anterior da palavra (para repetir a busca use "n").

n => Repete o último comando utilizando / ou ?.

N => Repete o último comando / ou ? ao contrário (baixo para cima).

Ctrl+g => Mostra o nome do arquivo, o número da linha corrente e o total de linhas.

Mais opções para remoção de caracteres

x => Apaga o caracter onde o cursor estiver.

dd => Apaga a linha inteira onde o cursor estive

D => Apaga a linha a partir da posição do cursor até o fim.

J => Une a linha corrente à próxima.

:5dd => Remove as próximas 7 linhas a partir da posição do atual do cursor (qualquer número).

Mais para copiar e colar

:yy => Copia a linha onde o cursor se encontra.

:5yy => Copia as próximas 5 linhas a partir da posição atual do cursor.

:p => Cola o que foi copiado na linha abaixo do cursor atual.

Opções para substituição de textos

rCHARACTER => Substitui o caracter onde o cursor se encontra pelo caracter especificado em CHARACTER.

RTEXTO => Substitui o texto corrente pelo texto digitado (sobrepõe).

cw => Remove a palavra corrente para substituição.

cc => Remove a linha corrente para substituição.

C => Substitui o restante da linha corrente, esperando o texto logo após o comando.

J => Une a linha corrente à próxima.

:s/velho/novo => Substitui a primeira ocorrência de "velho" por "novo" na linha corrente.

:% s/velho/novo => Substitui em todo o arquivo (%) a primeira ocorrência de "velho" por "novo" em cada linha.

:% s/velho/novo/g => Substitui em todo o arquivo (%), todas (g) as ocorrências de "velho" por "novo".

:% s/velho/novo/gc => Igual ao anterior, mas pedindo confirmação para cada substituição.

:% s/^String[0-9]//gc => Expressões regulares também funcionam, como no sed.

:% s/./\u&/gc => Converte para maiúsculas (\u) o primeiro caracter (.) de cada linha.

Abreviações

:ab => Mostra todas as abbr.

:abc[lear] => Remove todos.

:iab => Apenas para modo de inserção.

:iabc[lear] => Tira todos de inserção.

:cab => Apenas p/modo de comando (:).

:cabc[lear] => Tira todos os modos de comando.

:una vc => Tira ab para vc.

Observação: Pontuação, espaço ou o ENTER, disparam a expansão de uma abreviação. Porém, Ctrl+] também pode ser usado, para expandir sem adicionar caracteres.

Opções para o comando SET

:set

autowrite aw => Salva a cada alteração.

backspace bs => Comportamento backspace (1 ou 2).

errorbell eb => Campanha de erro.

expandtab et => Troca tab por espaços.

fileformat=dos ff => Converte o arquivo para DOS.
hidden hid => Preserva o buffer.
hlsearch hls => Elimina a última procura.
ignorecase ic => Case insensitive na busca.
incsearch is => Ilumina procura enquanto digita.
laststatus=2 => Mostra linha de estado.
lazyredraw lz => Não redesenha em macros.
lines=N => Número de linhas na tela.
magic => Usar mágicas na procura de padrões.
number nu => Mostra núm da linha.
report=N => Mostra aviso quando N linhas mudaram (0=sempre).
showcmd => Mostra o comando que se está fazendo.
showmatch sm => Mostra o casamento de {},[],().
smartcase scs => Assume "noic" quando tiver maiúsculas.
textwidth=N => Quebra de linha do texto.
undolevels ul=N => Guarde os N últimos comandos para desfazer (padrão=1000).
vb t_vb= => Retira o "beep" de erro.

Agora invertendo maiúsculas/minúsculas

5~ => Inverte os 5 próximos caracteres.
g~\$ => Inverte todos os caracteres até o fim da linha.
seleciona, u => Converte para minúsculas.
seleciona, U => Converte para maiúsculas.
seleciona, ~ => Inverte.

Observação: Onde está escrito "seleciona", é para fazer utilizando o modo visual (v).

Agora veja como definir coluna de quebra de linha (problema que eu tive quando iniciei no aprendizado do vim):

```
:set textwidth=N
```

Se você já estiver num arquivo pronto:

```
:set wm=5 => O número 5 aqui são as colunas que serão "cortadas".  
gqG => Até o final do arquivo.
```

Vamos ver agora o que podemos fazer pressionando a tecla "Ctrl":

É claro que é segurando Ctrl + .

No modo de COMANDO:

A => Incrementa um número (Add)
X => Decrementa um número
S => ScrollLock
L => Redesenha tela

V => Modo visual (Visual Vertical)
 G => Status do arquivo
 M => Início da próxima linha
 E => Linha abaixo sem mover cursor
 Y => Linha acima sem mover cursor
 N => Próxima linha (Next)
 P => Linha anterior (Previous)
 F => PageDown (Forward)
 B => PageUp (Backyard)
 U => PageUp / 2 (Up)
 D => PageDown / 2 (Down)

Agora, no modo de INSERÇÃO:

A => Insere o último texto inserido
 I => TAB
 S => ScrollLock
 H => BackSpace
 T => 2 tab's no início da linha (Two Tabs)
 V => Anula expansão do próximo caractere
 J => Enter - quebra de linha
 M => Enter - quebra de linha
 L => Redesenha tela
 R => Insere conteúdo do registrador [a-z] (Veja abaixo)
 K => Insere um dígrafo (Veja abaixo)
 N => Procura palavra no texto atual (Next)
 P => Procura palavra no texto atual (Previous)
 Y => Copia caractere que está acima (Yank)

Gerenciamento de processos.

Todo sistema operacional, sempre que executamos uma tarefa ou abrimos um programa o sistema cria um processo, também conhecido como JOBS, sendo assim, processo é um aplicativo ou tarefa em execução.

Todo processo gera um PID, que é um número único que identifica o processo, e não existe 2 processos como o mesmo PID.

Comandos para gerenciamento de processos.

Comando:	ps
Descrição:	Permite exibir os processos em execução e seu respectivo PID.
Sintaxe:	ps opções
Opções:	-l lista detalhes do processo -aux lista processos de todos os usuários, fornecendo o nome e o usuário e quando o processo foi iniciado, mostrando inclusive processos não associados a um terminal x
Exemplo:	PS -l

Descrição dos campos:

PID - Identificação do processo.

PPID - Parent PID. Identificação do processo pai na árvore de processos

UID - User ID. Identificação do usuário que criou o processo.

TTY - Terminal onde o processo está sendo executado.

TIME - Tempo de processamento consumido pelo processo.

CMD - Comando referente ao processo

COMMAND - Linha de comando usada para executar o processo

STAT - Estado do processo (R - Executando, S - Em espera, T - Parado, etc.)

%CPU, %MEM, VSZ, RSS - CPU e memória usados pelo processo.

Comando:	jobs
Descrição:	Lista os jobs (processos sendo executados em segundo plano).
Sintaxe:	Jobs
Exemplo:	Jobs

Para colocar um processo em segundo plano pressione CTRL+Z enquanto o processo estiver sendo executado. Exemplo:

Controlando JOBS

Comando:	BG
Descrição:	Coloca um processo em segundo plano.
Sintaxe:	BG PID
Exemplo:	BG 5685

Comando:	FG
Descrição:	Coloca um processo em Primeiro plano.
Sintaxe:	FG PID
Exemplo:	FG 5685

Exemplo:

Supondo que a saída do comando ps seja:

```
PID TTY    TIME CMD
100 pts/0 00:00:00 bash
105 pts/0 00:00:00 yes
120 pts/0 00:00:00 ps
```

E do comando jobs seja:

```
[1]+ Stopped  yes > /dev/null
```

Você pode colocar o processo em segundo plano usando o comando "bg 105" (usando o PID) ou "bg %1" (usando o número do Job).

Para colocar o processo em primeiro plano novamente use "fg <PID>" ou "fg %<JOB>". Para colocá-lo novamente em segundo plano use CTRL+Z.

Experimente executar o editor "vi" e colocá-lo em segundo plano. Para voltar para o editor digite "fg %<NR. DE JOB DO VI>". Exemplo:

- Execute "vi"
- Pressione CTRL+Z (Aparece a mensagem: [[1]+ Stopped vi)
- Execute algum outro comando (por exemplo: ls)
- Retorne para o vi com o comando: fg %1
- Para fechar o editor "vi", digite ":q!" e pressione ENTER.

OBS: Este procedimento é muito útil quando se está usando algum programa interativo e é preciso executar alguma outra operação no shell.

Comando:	&
Descrição:	Inicia um processo diretamente em segundo plano acrescentando-se um & ao final da linha de comando
Sintaxe:	& ao final da linha de comando
Exemplo:	Cp prova /bin&

Terminando processos

Comando:	Kill
Descrição:	Interrompe um processo em execução
Sintaxe:	Kill PID
Exemplo:	Kill -l %1

Normalmente o comando kill envia um aviso ao processo (sinal TERM) para que este tenha tempo de fechar seus arquivos e terminar normalmente suas operações. Para forçar o término imediato de um processo (podendo resultar em perdas de informações, etc) use "kill -s KILL <PID>".

Para interromper o processo sendo executado no momento pressione CTRL+C.

OBS: Somente o usuário "root" pode fechar os processos de outros usuários.

Comando:	Killall
Descrição:	Termina um processo através do nome.
Sintaxe:	Killall processo
Exemplo:	Kill cp

Controlando prioridades

Comando:	nice
Descrição:	É possível executar comandos com prioridade diferenciadas. Por exemplo: Se você precisa ordenar um arquivo ou compilar um programa muito grande e esta operação será demorada, você pode reduzir o impacto na execução associando prioridades mais baixas a este processo. Basta executar o processo com o comando nice.
Sintaxe:	Nice processo
Exemplo:	Nice -n 5 cp

OBS: Somente o root pode atribuir prioridades negativas (altas) a um processo.

Comando:	renice
Descrição:	Altera a prioridade de um processo em execução.
Sintaxe:	Nice processo
Exemplo:	renice 7 177 (Altera para 7 a prioridade do processo de PID 177)

OBS: Somente o usuário root pode dar mais prioridade para um processo. Os usuários normais só podem reduzir a prioridade. (Lembre-se que na verdade, quando maior o valor de nice, menor a prioridade.)

Mais sobre processos

Comando:	Nohup
Descrição:	. Normalmente um processo é terminado quando o processo pai termina. Portanto, todos os processos iniciados durante uma sessão serão terminados quando ela for fechada. O comando nohup, faz com que um processo permaneça em execução mesmo quando o shell for fechado e o usuário tenha efetuado logoff do sistema.
Sintaxe:	Nohup processo
Exemplo:	nohup cp > /dev/null &

Comando:	Su
Descrição:	Além de permitir alternar entre um usuário e outro, o comando su ainda serve para rodar processos como um

	usuário diferente.
Sintaxe:	Su usuário
Exemplo:	su aluno -c ls

PRÁTICA 1

- 1) CRIE UM ARQUIVO CHAMADO ITS.DOC
- 2) CRIE UMA PASTA CHAMADA BACKUP
- 3) COPIE O ARQUIVO ITS.DOC PARA A PASTA BACKUP
- 4) CRIE UM ARQUIVO CHAMADO REZENDE.DOC
- 5) COPIE O ARQUIVO REZENDE.DOC PARA A PASTA BACKUP
- 6) CRIE UMA PASTA CHAMADA COLEGIO
- 7) MOVER O ARQUIVO ITS.DOC DA PASTA BACKUP PARA A PASTA COLEGIO
- 8) MOVER O ARQUIVO REZENDE.DOC DA PASTA BACKUP PARA A PASTA COLEGIO
- 9) CRIA UM LINK PARA O ARQUIVO ITS.DOC
- 10) CRIAR UM LINK SIMBÓLICO PARA O AQUIVO REZENDE.DOC

PRÁTICA 2

- 1) EXIBIR A DATA ATUAL DO SISTEMA
- 2) EXIBIR O CALENDÁRIO DO ANO ATUAL
- 3) EXIBIR O CALENDÁRIO DO ANO 2050
- 4) EXIBIRO CALENDÁRIO DO ANO 3000
- 5) CRIAR UM ARQUIVO CHAMADO PROVA.TXT USANDO O COMANDO CAT COMO REDIRECIONADOR
- 6) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO CAT
- 7) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO MORE
- 8) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO LESS
- 9) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO MORE E REDIRECIONANDO PARA O ARQUIVO PROVA2.DOC
- 10) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO LESS E REDIRECIONANDO PARA O ARQUIVO PROVA3.DOC

PRÁTICA 3

- CRIE UM ARQUIVO CHAMADO LINHA.TXT COM 10 LINHAS USANDO O COMANDO CAT.
- LER AS 3 PRIMEIRAS LINHAS DO ARQUIVO LINHA.TXT
- LER AS 5 PRIMEIRAS LINHAS DO ARQUIVO LINHA.TXT

- LER AS 2 ULTIMAS LINHAS DO ARQUIVO LINHA.TXT
- LER AS 4 ULTIMAS LINHAS DO ARQUIVO LINHA.TXT
- CONTAR O NÚMERO DE LINHAS DO ARQUIVO LINHA.TXT
- CONTAR O NÚMERO DE PALAVRAS DO ARQUIVO LINHA.TXT
- CONTAR O NÚMERO DE CARACTERES DO ARQUIVO LINHA.TXT
- CONTAR O NÚMERO DE LINHAS, PALAVRAS E CARACTERES DO ARQUIVO LINHA.TXT
- EXIBIR O CONTEÚDO DO ARQUIVO LINHA.TXT USANDO OS COMANDOS CAT, MORE E LESS.

PRÁTICA 4

- 1) CRIE UM ARQUIVO CHAMADO NOMES.TXT, COM 5 NOMES ALEATÓRIOS UM EM BAIXO DO OUTRO
- 2) ORDENE O ARQUIVO NOMES.TXT
- 3) CRIE O ARQUIVO NOMEORDENADO A PARTIR DO ARQUIVO NOMES.TXT, ORDENANDO O SEU CONTEÚDO
- 4) MOSTRE A DIFERENÇA ENTRE OS ARQUIVOS NOMES.TXT E NOMEORDENADO.
- 5) CRIE UM ARQUIVO CHAMA CHAMADA.DOC COM 5 NOMES ALEATÓRIOS EM UM EM BAIXO DO OUTRO
- 6) ORDENE O ARQUIVO CHAMADA.DOC
- 7) CRIE O ARQUIVO CHAMADANOVA A PARTIR DO ARQUIVO CHAMADA.TXT, ORDENANDO O SEU CONTEÚDO
- 8) MOSTRE A DIFERENÇA ENTRE OS ARQUIVOS CHAMADA.TXT E CHAMADANOVA

PRÁTICA 5

- 17)USANDO O VI CRIE UM ARQUIVO COM 10 LINHAS CHAMADO INFO.DOC
- 18)COPIAR A LINHA 5
- 19)COLAR A LINHA 5 NO FINAL DO ARQUIVO
- 20)RECORTAR A LINHA 2
- 21)COLAR A LINHA 2 NO MEIO DO ARQUIVO
- 22)EXCLUIR A LINHA 5 E LINHA 2
- 23)SALVAR O AQUIVO
- 24)SAIR DO VI
- 25)LER O CONTEÚDO DO ARQUIVO INFO.DOC
- 26)EDITAR O ARQUIVO INFO.DOC INCLUINDO 2 LINHAS AO FINAL
- 27)SALVAR O AQUIVO
- 28)SAIR DO VI.

PRÁTICA 6

- 1) VERIFICAR TODOS OS PROCESSOS EM EXECUÇÃO.
- 2) ESCOLHA UM PROCESSO E COLOQUE-O EM SEGUNDO PLANO.
- 3) TERMINE O PROCESSO ANTERIOR

- 4) EXECUTE O VI
- 5) COLOQUE O VI EM SEGUNDO PLANO
- 6) VERIFIQUE OS PROCESSOS EM EXECUÇÃO
- 7) COLOQUE O VI EM PRIMEIRO PLANO
- 8) FINALIZE O PROCESSO DO VI

EXERCÍCIOS PROPOSTOS

- 1 - QUAL A FUNÇÃO DO COMANDO CP?
- 2 - QUAL A FUNÇÃO DO COMANDO MV?
- 3 - QUAL A FUNÇÃO DO COMANDO LN?
- 4 - QUAL A DIFERENÇA ENTRE LINK SIMBÓLICO E LINK DIRETO?
- 5 - QUAL O COMANDO USADO PARA EXIBIR A DATA ATUAL
- 6 - QUAL A FUNÇÃO DO COMANDO CAL
- 7 - QUAL A FUNÇÃO DO COMANDO CAT
- 8 - O QUE É REDIRECIONAMENTO?
- 9 - QUAL A FUNÇÃO DO COMANDO MORE?
- 10 - QUAL A FUNÇÃO DO COMANDO LESS?
- 11 - QUAL O COMANDO USADO PARA LER AS PRIMEIRAS LINHAS DE UM ARQUIVO?
- 12 - QUAL A FUNÇÃO DO COMANDO TAIL?
- 13 - QUAL A FUNÇÃO DO COMANDO WC?
- 14 - QUAL O COMANDO USADO PARA ORDENAR O CONTEÚDO DE UM ARQUIVO?
- 15 - QUAL A FUNÇÃO DO COMANDO DIFF
- 16 - QUAL A PRINCIPAL FUNÇÃO DO VI
- 17 - QUAL O COMANDO USADO PARA COPIAR LINHAS NO VI
- 18 - QUAL O COMANDO USADO PARA COLAR LINHAS NO VI
- 19 - O QUE PODEMOS FAZER NO MODO DE COMANDOS DO VI
- 20 - O QUE PODEMOS FAZER NO MODO DE LINHA DO VI
- 21 - CITE O PROCESSO PARA SALVAR UM ARQUIVO NO VI.
- 22 - QUAL O COMANDO USADO PARA SAIR DO VI
- 23 - O QUE É UM PROCESSO?
- 24 - O QUE É PID?
- 25 - QUAL A FUNÇÃO DO COMANDO PS
- 26 - O QUE É COLOCAR UM PROCESSO EM SEGUNDO PLANO?
- 27 - QUAL O COMANDO USADO PARA COLOCAR UM PROCESSO EM SEGUNDO PLANO
- 28 - QUAL A FUNÇÃO DO COMANDO & E COMO PODEMOS UTILIZÁ-LO?
- 29 - QUAL A FUNÇÃO DO COMANDO KILL?
- 30 - QUAL A FUNÇÃO DO COMANDO KILLALL?

Gerenciamento de processos.

Todo sistema operacional, sempre que executamos uma tarefa ou abrimos um programa o sistema cria um processo, também conhecido como JOBS, sendo assim, processo é um aplicativo ou tarefa em execução.

Todo processo gera um PID, que é um número único que identifica o processo, e não existe 2 processos como o mesmo PID.

Comandos para gerenciamento de processos.

Comando:	ps
Descrição:	Permite exibir os processos em execução e seu respectivo PID.
Sintaxe:	ps opções
Opções:	-l lista detalhes do processo -aux lista processos de todos os usuários, fornecendo o nome e o usuário e quando o processo foi iniciado, mostrando inclusive processos não associados a um terminal x
Exemplo:	PS -l

Descrição dos campos:

PID - Identificação do processo.

PPID - Parent PID. Identificação do processo pai na árvore de processos

UID - User ID. Identificação do usuário que criou o processo.

TTY - Terminal onde o processo está sendo executado.

TIME - Tempo de processamento consumido pelo processo.

CMD - Comando referente ao processo

COMMAND - Linha de comando usada para executar o processo

STAT - Estado do processo (R - Executando, S - Em espera, T - Parado, etc.)

%CPU, %MEM, VSZ, RSS - CPU e memória usados pelo processo.

Comando:	jobs
Descrição:	Lista os jobs (processos sendo executados em segundo plano).
Sintaxe:	Jobs
Exemplo:	Jobs

Para colocar um processo em segundo plano pressione CTRL+Z enquanto o processo estiver sendo executado. Exemplo:

Controlando JOBS

Comando:	BG
Descrição:	Coloca um processo em segundo plano.
Sintaxe:	BG PID
Exemplo:	BG 5685

Comando:	FG
Descrição:	Coloca um processo em Primeiro plano.
Sintaxe:	FG PID
Exemplo:	FG 5685

Exemplo:

Supondo que a saída do comando ps seja:

```
PID TTY    TIME CMD
100 pts/0 00:00:00 bash
105 pts/0 00:00:00 yes
120 pts/0 00:00:00 ps
```

E do comando jobs seja:

```
[1]+ Stopped  yes > /dev/null
```

Você pode colocar o processo em segundo plano usando o comando "bg 105" (usando o PID) ou "bg %1" (usando o número do Job).

Para colocar o processo em primeiro plano novamente use "fg <PID>" ou "fg %<JOB>". Para colocá-lo novamente em segundo plano use CTRL+Z.

Experimente executar o editor "vi" e colocá-lo em segundo plano. Para voltar para o editor digite "fg %<NR. DE JOB DO VI>". Exemplo:

- Execute "vi"
- Pressione CTRL+Z (Aparece a mensagem: [[1]+ Stopped vi)
- Execute algum outro comando (por exemplo: ls)
- Retorne para o vi com o comando: fg %1
- Para fechar o editor "vi", digite ":q!" e pressione ENTER.

OBS: Este procedimento é muito útil quando se está usando algum programa interativo e é preciso executar alguma outra operação no shell.

Comando:	&
Descrição:	Inicia um processo diretamente em segundo plano acrescentando-se um & ao final da linha de comando
Sintaxe:	& ao final da linha de comando
Exemplo:	Cp prova /bin&

Terminando processos

Comando:	Kill
Descrição:	Interrompe um processo em execução
Sintaxe:	Kill PID
Exemplo:	Kill -l %1

Normalmente o comando kill envia um aviso ao processo (sinal TERM) para que este tenha tempo de fechar seus arquivos e terminar normalmente suas operações. Para forçar o término imediato de um processo (podendo resultar em perdas de informações, etc) use "kill -s KILL <PID>".

Para interromper o processo sendo executado no momento pressione CTRL+C.

OBS: Somente o usuário "root" pode fechar os processos de outros usuários.

Comando:	Killall
Descrição:	Termina um processo através do nome.
Sintaxe:	Killall processo
Exemplo:	Kill cp

Controlando prioridades

Comando:	nice
Descrição:	É possível executar comandos com prioridade diferenciadas. Por exemplo: Se você precisa ordenar um arquivo ou compilar um programa muito grande e esta operação será demorada, você pode reduzir o impacto na execução associando prioridades mais baixas a este processo. Basta executar o processo com o comando nice.
Sintaxe:	Nice processo
Exemplo:	Nice -n 5 cp

OBS: Somente o root pode atribuir prioridades negativas (altas) a um processo.

Comando:	renice
Descrição:	Altera a prioridade de um processo em execução.
Sintaxe:	Nice processo
Exemplo:	renice 7 177 (Altera para 7 a prioridade do processo de PID 177)

OBS: Somente o usuário root pode dar mais prioridade para um processo. Os usuários normais só podem reduzir a prioridade. (Lembre-se que na verdade, quando maior o valor de nice, menor a prioridade.)

Mais sobre processos

Comando:	Nohup
Descrição:	. Normalmente um processo é terminado quando o processo pai termina. Portanto, todos os processos iniciados durante uma sessão serão terminados quando ela for fechada. O comando nohup, faz com que um processo permaneça em execução mesmo quando o shell for fechado e o usuário tenha efetuado logoff do sistema.
Sintaxe:	Nohup processo
Exemplo:	nohup cp > /dev/null &

Comando:	Su
Descrição:	Além de permitir alternar entre um usuário e outro, o comando su ainda serve para rodar processos como um usuário diferente.
Sintaxe:	Su usuário
Exemplo:	su aluno -c ls

PRÁTICA 1

- 1) CRIE UM ARQUIVO CHAMADO ITS.DOC
- 2) CRIE UMA PASTA CHAMADA BACKUP
- 3) COPIE O ARQUIVO ITS.DOC PARA A PASTA BACKUP
- 4) CRIE UM ARQUIVO CHAMADO REZENDE.DOC
- 5) COPIE O ARQUIVO REZENDE.DOC PARA A PASTA BACKUP
- 6) CRIE UMA PASTA CHAMADA COLEGIO
- 7) MOVER O ARQUIVO ITS.DOC DA PASTA BACKUP PARA A PASTA COLEGIO
- 8) MOVER O ARQUIVO REZENDE.DOC DA PASTA BACKUP PARA A PASTA COLEGIO
- 9) CRIA UM LINK PARA O ARQUIVO ITS.DOC
- 10) CRIAR UM LINK SIMBÓLICO PARA O AQUIVO REZENDE.DOC
- 11)

PRÁTICA 2

- 1) EXIBIR A DATA ATUAL DO SISTEMA
- 2) EXIBIR O CALENDÁRIO DO ANO ATUAL
- 3) EXIBIR O CALENDÁRIO DO ANO 2050
- 4) EXIBIR O CALENDÁRIO DO ANO 3000
- 5) CRIAR UM ARQUIVO CHAMADO PROVA.TXT USANDO O COMANDO CAT COMO REDIRECIONADOR
- 6) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO CAT
- 7) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO MORE
- 8) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO LESS
- 9) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO MORE E REDIRECIONANDO PARA O ARQUIVO PROVA2.DOC
- 10) LER O CONTEÚDO DO ARQUIVO PROVA.TXT USANDO O COMANDO LESS E REDIRECIONANDO PARA O ARQUIVO PROVA3.DOC

PRÁTICA 3

- 11) CRIE UM ARQUIVO CHAMADO LINHA.TXT COM 10 LINHAS USANDO O COMANDO CAT.
- 12) LER AS 3 PRIMEIRAS LINHAS DO ARQUIVO LINHA.TXT
- 13) LER AS 5 PRIMEIRAS LINHAS DO ARQUIVO LINHA.TXT
- 14) LER AS 2 ÚLTIMAS LINHAS DO ARQUIVO LINHA.TXT
- 15) LER AS 4 ÚLTIMAS LINHAS DO ARQUIVO LINHA.TXT
- 16) CONTAR O NÚMERO DE LINHAS DO ARQUIVO LINHA.TXT
- 17) CONTAR O NÚMERO DE PALAVRAS DO ARQUIVO LINHA.TXT
- 18) CONTAR O NÚMERO DE CARACTERES DO ARQUIVO LINHA.TXT
- 19) CONTAR O NÚMERO DE LINHAS, PALAVRAS E CARACTERES DO ARQUIVO LINHA.TXT
- 20) EXIBIR O CONTEÚDO DO ARQUIVO LINHA.TXT USANDO OS COMANDOS CAT, MORE E LESS.

PRÁTICA 4

- 9) CRIE UM ARQUIVO CHAMADO NOMES.TXT, COM 5 NOMES ALEATÓRIOS UM EM BAIXO DO OUTRO
- 10) ORDENE O ARQUIVO NOMES.TXT
- 11) CRIE O ARQUIVO NOMEORDENADO A PARTIR DO ARQUIVO NOMES.TXT, ORDENANDO O SEU CONTEÚDO
- 12) MOSTRE A DIFERENÇA ENTRE OS ARQUIVOS NOMES.TXT E NOMEORDENADO.
- 13) CRIE UM ARQUIVO CHAMADO CHAMADA.DOC COM 5 NOMES ALEATÓRIOS UM EM BAIXO DO OUTRO
- 14) ORDENE O ARQUIVO CHAMADA.DOC

- 15)CRIE O ARQUIVO CHAMADANOVA A PARTIR DO ARQUIVO CHAMADA.TXT, ORDENANDO O SEU CONTEÚDO
- 16)MOSTRE A DIFERENÇA ENTRE OS ARQUIVOS CHAMADA.TXT E CHAMADANOVA

PRÁTICA 5

- 9) USANDO O VI CRIE UM ARQUIVO COM 10 LINHAS CHAMADO INFO.DOC
- 10)COPIAR A LINHA 5
- 11)COLAR A LINHA 5 NO FINAL DO ARQUIVO
- 12)RECORTAR A LINHA 2
- 13)COLAR A LINHA 2 NO MEIO DO ARQUIVO
- 14)EXCLUIR A LINHA 5 E LINHA 2
- 15)SALVAR O AQUIVO
- 16)SAIR DO VI
- 17)LER O CONTEÚDO DO ARQUIVO INFO.DOC
- 18)EDITAR O ARQUIVO INFO.DOC INCLUINDO 2 LINHAS AO FINAL
- 19)SALVAR O AQUIVO
- 20)SAIR DO VI.

PRÁTICA 6

- 1) VERIFICAR TODOS OS PROCESSOS EM EXECUÇÃO.
- 2) ESCOLHA UM PROCESSO E COLOQUE-O EM SEGUNDO PLANO.
- 3) TERMINE O PROCESSO ANTERIOR
- 4) EXECUTE O VI
- 5) COLOQUE O VI EM SEGUNDO PLANO
- 6) VERIFIQUE OS PROCESSOC EM EXECUÇÃO
- 7) COLOQUE O VI EM PRIMEIRO PLANO
- 8) FINALIZE O PROCESSO DO VI

EXERCÍCIOS PROPOSTOS

- 29)QUAL A FUNÇÃO DO COMANDO CP?
- 30)QUAL A FUNÇÃO DO COMANDO MV?
- 31)QUAL A FUNÇÃO DO COMANDO LN?
- 32)QUAL A DIFERENÇA ENTRE LINK SIMBÓLICO E LINK DIRETO?
- 33)QUAL O COMANDO USADO PARA EXIBIR A DATA ATUAL
- 34)QUAL A FUNÇÃO DO COMANDO CAL
- 35)QUAL A FUNÇÃO DO COMANDO CAT
- 36)O QUE É REDIRECIONAMENTO?
- 37)QUAL A FUNÇÃO DO COMANDO MORE?
- 38)QUAL A FUNÇÃO DO COMANDO LESS?
- 39) QUAL O COMANDO USADO PARA LER AS PRIMEIRAS LINHAS DE UM ARQUIVO?
- 40)QUAL A FUNÇÃO COMANDO TAIL?

- 41) QUAL A FUNÇÃO DO COMANDO WC?
- 42) QUAL O COMANDO USADO PARA ORDENAR O CONTEÚDO DE UM ARQUIVO?
- 43)QUAL A FUNÇÃO DO COMANDO DIFF
- 44)QUAL A PRINCIPAL FUNÇÃO DO VI
- 45)QUAL O COMANDO USADO PARA COPIAR LINHAS NO VI
- 46)QUAL O COMANDO USADO PARA COLAR LINHAS NO VI
- 47)O QUE PODEMOS FAZER NO MODO DE COMANDOS DO VI
- 48)O QUE PODEMOS FAZER NO MODO DE LINHA DO VI
- 49)CITE O PROCESSO PARA SALVAR UM ARQUIVO NO VI.
- 50)QUAL O COMANDO USADO PARA SAIR DO VI
- 51)O QUE É UM PROCESSO?
- 52)O QUE É PID?
- 53)QUAL A FUNÇÃO DO COMANDO PS
- 54)O QUE É COLOCAR UM PROCESSO EM SEGUNDO PLANO?
- 55)QUAL O COMANDO USADO PARA COLOCAR UM PROCESSO EM SEGUNDO PLANO
- 56)QUAL A FUNÇÃO DO COMANDO & E COMO PODEMOS UTILIZÁ-LO?
- 57)QUAL A FUNÇÃO DO COMANDO KILL?
- 58)QUAL A FUNÇÃO DO COMANDO KILLALL?

Gerenciamento de sistemas de arquivos

Dispositivos de armazenamento e árvore de diretório

Embora todos os arquivos do sistema operacional Linux, estejam conectados em uma gigantesca árvore de diretórios que começa no diretório raiz, os arquivos por si mesmos residem em dispositivos de armazenamento diferentes como: Discos flexíveis, discos rígidos, cd-roms , etc.

Os arquivos Linux residentes em um dispositivo de armazenamento qualquer são organizados em um sistema de arquivos. A árvore de diretórios do Linux pode ter , vários sistemas de arquivos diferentes, localizados em dispositivos de armazenamento diferentes.

Um sistema de arquivos residente em um dispositivos de armazenamento qualquer tem seus arquivos organizados em sua própria árvore de diretórios, que permanece separada da árvore de diretórios principal, até que seja conectada a ela. Por exemplo, um drive de discos flexíveis com arquivos Linux tem sua própria árvore de diretórios e é necessário conectar essa sub-árvore de diretórios à árvore principal residente na partição do disco rígido. Até que isto seja feito, não será possível acessar os arquivos presentes no disco rígido.

Este processo de conectar a árvore de diretórios de um dispositivo de armazenamento qualquer à árvore de diretórios de um dispositivo de armazenamento qualquer à árvore principal é chamado de “montar o dispositivo de armazenamento”. Isto é feito pelo comando Mount. Para acessarmos arquivos residentes em um dispositivo de armazenamento qualquer, precisamos primeiro monta-lo em um diretório específico. Só o superusuário ROOT pode realizar esta operação.

Comando:	Mount
Descrição:	Monta sistemas de arquivos
Sintaxe:	Mount opções arquivo-de-dispositivo ponto de montagem
Opções:	-a monta todos os dispositivos no arquivo /etc/fstab Fstype pode ser: Ms-dos – sistema de arquivos fat16 do dos Vfat – sistema de arquivos fat32 do windows Nfs – sistema de arquivos do windows nt Ext2 sistema de arquivos linux Iso9960 – sistema de arquivos para montagem de cd-rom
Exemplo:	Mount /dev/fd0 /mnt/floppy

O sistema de arquivos principal é montado automaticamente, Outros sistemas de arquivos residentes em outros dispositivos de armazenamento precisam ser montados explicitamente pelo comando mount.

Antes de desligar o sistema, todos os sistemas de arquivos montados tem que ser desmontados explicitamente. Exceto o sistema de arquivos principal, que é desmontado automaticamente. O comando usado para esta operação é o comando Umount. Só o superusuário root pode realizar esta operação.

Comando:	Umount
Descrição:	desmonta sistemas de arquivos
Sintaxe:	Umount opções arquivo-de-dispositivo ponto de montagem
Opções:	Nenhuma
Exemplo:	Umount /dev/fd0 /mnt/floppy

Acesso a arquivos

O sistema Operacional Linux, trata o acesso a arquivos de forma randômica, ou seja, seus bytes ou registros podem ser lidos em qualquer ordem.

Atributos dos arquivos.

Cada arquivo tem necessariamente um nome e um conjunto de dados. Além disso, o Sistema Operacional associa a cada arquivo algumas outras informações que chamaremos de atributos de arquivos.

O sistema de arquivos do Linux permite restringir o acesso aos arquivos e diretórios permitindo que somente determinados usuários possam acessá-los. A cada arquivo e diretório é associado um conjunto de permissões. Essas permissões determinam quais usuários podem ler, escrever, ou alterar um arquivo, e no caso de arquivos executáveis como programas, quais usuários podem executá-lo. Se um usuário tem permissão de execução de um diretório, significa que ele pode realizar buscas dentro daquele diretório, e não executá-lo como se fosse um programa.

Diretório	Usuário			Grupo			Outros		
D	R	W	X	R	W	X	R	W	X
1	2	3	4	5	6	7	8	9	10

Legenda:

D – Diretório
R – Leitura
W – Escrita
X – Execução.

Exemplo:

- r w - r - - r - - teste.doc

No arquivo teste.doc o usuário pode ler e escrever, o grupo pode apenas lê e outros podem apenas ler.

Comando:	CHMOD
Descrição:	Alterar os atributos de forma simbólica ou absoluta.
Sintaxe:	CHMOD atributos
Opções:	Modo simbólico.
Exemplo:	CHMOD U+W,G-WX,O-W

No modo absoluto cada atributo tem um valor. Como tabela abaixo.

4	2	1
R	W	X

Exemplo.

- R W – R - - R - - => Valor absoluto = 644

Alterando o atributos usando o modo absoluto.

ARQUIVO: TESTE.DOC

DE: - R W X R W – R - - => VALOR ABSOLUTO 764

PARA: - R W – R - - R - - => VALOR ABSOLUTO 644

CHMOD 644 TESTE.DOC

Comando:	Locate
Descrição:	Localiza arquivos a partir de um banco de dados atualizado periodicamente
Sintaxe:	Locate padrão
Opções:	
Exemplo:	Locate *.doc

Comando:	Whereis
Descrição:	O comando localiza o binário, a fonte e a página de manual para um comando

Sintaxe:	Whereis opções arquivo
Opções:	-b busca apenas por binário -s busca apenas por fontes -m busca apenas por páginas de manual
Exemplo:	Whereis teste.doc Whereis -b teste.doc Whereis -s teste.doc

Exercícios propostos

- Qual a função do comando Mount?
- Qual a função do comando Umount?
- Como fazemos para montar uma unidade USB?
- Como fazemos para desmontar a unidade USB?
- Qual a função dos atributos dos arquivos?
- Tendo o arquivos abaixo e seus atributos, diga o que cada grupo pode fazer com os arquivos.

Arquivo:	Atributos
A) Prova	_ R W X R - - R W _
B) Teste	- R _ X R W _ R _ _
C) Exercio	_ R W X R W X R _ _

- Usando os mesmo arquivos e atributos acima, diga qual o valor absoluto de cada um.

A)
B)
C)

- Tendo os arquivos e atributos abaixo, usando o comando CHMOD, altere para o que se pede:

A) Prova	DE: _ R W X R - - R W _ Para: _ RW _ RW - R _ _
B) Teste	DE: - R _ X R W _ R _ _ Para: _ RW _ R _ _ R _ _
C) Exercio	De: _ R W X R W X R _ _ Para: _ R W _ R W _ R W _

Configurando o FSTAB - O arquivo /etc/fstab permite configurar o sistema para montar partições, cdroms, disquetes e compartilhamentos de rede durante o boot. Cada linha é responsável por um ponto de montagem. É através do /etc/fstab que o sistema é capaz de acessar o seu CD-ROM por

exemplo. O `fstab` é um dos arquivos essenciais para o funcionamento do sistema, por isso, antes de editá-lo faça sempre uma cópia de segurança, usando por exemplo o comando (como root):

```
cp /etc/fstab /etc/fstab-original
```

Para abrir o arquivo rode (novamente como root) o comando:

```
vi /etc/fstab
```

À primeira vista o `fstab` parece ser mais um daqueles arquivos indecifráveis. Ele possui uma lógica própria que parece um pouco complicada no início, mas é relativamente fácil de entender. Uma vez que você entenda a sintaxe das opções você poderá editar o `fstab` para adicionar um segundo drive de CD ou fazer com que um compartilhamento de rede seja montado automaticamente durante o boot, sem depender de configuradores automáticos.

Vamos começar dando uma olhada no `/etc/fstab`

```
# /etc/fstab: filesystem table.
#
# filesystem mountpoint type options dump pass

/dev/hda2 / reiserfs defaults 0 1
/dev/hda5 none swap sw 0 0
proc /proc proc defaults 0 0

/dev/fd0 /floppy vfat defaults,user,noauto,showexec,umask=022 0 0
/dev/cdrom /mnt/cdrom iso9660 defaults,ro,user,noexec,noauto 0 0

# partitions found by Knoppix

/dev/hda1 /mnt/hda1 reiserfs noauto,users,exec 0 0
/dev/hda2 /mnt/hda2 reiserfs noauto,users,exec 0 0
/dev/hda3 /mnt/hda3 reiserfs noauto,users,exec 0 0
/dev/hda6 /mnt/hda6 reiserfs noauto,users,exec 0 0

# Monta a partição /home, adicionado pelo instalador do Kurumin

/dev/hda3 /home reiserfs notail 1 2

# Ativa o USB

usbdevfs /proc/bus/usb usbdevfs defaults 0 0

Este é o arquivo gerado automaticamente durante a instalação.
```

Administração de Sistemas Linux

SUDO

Para administradores de sistemas que compartilham a responsabilidade de administrar sistemas com outros administradores, ou pessoas que ocasionalmente necessitam de acesso de superusuário, o sudo é bem útil. O sudo permite que se dê poderes limitados de superusuário para outras pessoas ao mesmo tempo em que o controle é mantido sobre os comandos que aqueles grupos ou usuários podem usar.

O comando sudo, como muitos outros comandos Linux que conhecemos, emprega um arquivo de configuração detalhando os comandos que podem ser usados por usuários particulares de um host. Um grupo seletivo de usuários, por exemplo, pode ter permissão de reiniciar o servidor. Usando o sudo, estes usuários escreveriam:

```
% sudo shutdown -y -g 5 -i 6
```

Os usuários então informam sua senha de login. A senha garante que ninguém que encontre uma tela desprotegida pode usar comandos para os quais não tem permissão.

O sudo usa um sistema de temporização e autoriza os usuários por um período de tempo antes de pedir que eles se autentiquem novamente. Por exemplo, será pedida a senha a um usuário a primeira vez que eles usam o sudo, mas não será pedida a senha para os próximos comandos. Após haver passado 5 minutos, entretanto, o usuário terá que informar novamente sua senha. Esta exigência ocasional de autenticação é conhecida como "ticketing".

Os usuários podem pedir por uma lista de comandos permitidos dentro da estrutura do sudo executando o comando sudo com o argumento -l conforme o exemplo:

```
$ /usr/local/bin/sudo -l  
allowed: /usr/bin/reboot  
allowed: /usr/bin/shutdown  
allowed: /usr/sbin/mount
```

Um usuário somente precisa invocar os comandos para os quais ele normalmente não tem permissão (como usuário normal) via sudo.

Os comandos sudo executados pelos usuários podem também ser registrados em um arquivo de log, que pode ser armazenado em um host central. Dependendo do seu site, isto pode tornar fácil para você revisar os dados do log quando tem que investigar um problema.

O arquivo de configuração do sudo, */etc/sudoers*, pode ser simples ou complexo, dependendo do número de grupos de usuários e os comandos que você quer configurar. O arquivo abaixo é um exemplo simples que configura

um grupo de usuários, um grupo de comandos privilegiados, e um grupo de hosts. Ele então associa os grupos de usuários com comandos que eles podem executar (usando o sudo) e sistemas nos quais eles podem executá-los.

```
#/etc/sudoers file.  
#  
# This file MUST be edited with the 'visudo' command as root.  
#  
# See the sudoers man page for the details on how to write a sudoers file.  
#  
  
# Host alias specification  
Host_Alias BIGGIES = big1, big2, big3, big4, big5, big6  
  
# User alias specification  
User_Alias OPS = john, jane, jim, joan, jeff, jan  
  
# Cmnd alias specification  
Cmnd_Alias RESTART = /usr/sbin/reboot, /usr/sbin/shutdown, /usr/sbin/init  
  
# User privilege specification  
OPS BIGGIES = RESTART
```

Este arquivo de ser copiado para todos os sistemas listados na linha Host_Alias e o comando sudo deve ser instalado para que o sudo fique operacional.

Este arquivo /etc/sudoers exemplo somente toca na sintaxe que deve estar contida no arquivo de configuração. Você deve obter alguns exemplos adicionais e ler a página man do sudoers para entender completamente as opções.

Gerenciador de BOOT.

Gerenciador de boot é um sistema instalado no Linux que permitirá ao usuário escolher entre os sistemas operacionais instalados no computador. Permitindo então 2 ou mais sistemas operacionais fiquem instalados na máquina em partições diferentes do disco. Como exemplo de gerenciador de boot, temos o Lilo e o Grub mais usado atualmente.

3 - Configurando o LILO

O LILO é um dos mais importantes softwares do Linux, é ele quem carrega o kernel no seu sistema para começar a funcionar! Entender a configuração dele é muito importante, principalmente para quem tem muitos sistemas numa mesma máquina. Com o LILO, é possível gerenciar partições e assim bootar em diferentes sistemas operacionais. É com ele que fica possível colocar por exemplo, o Windows e o Linux na mesma máquina.

O arquivo de configuração do LILO é o /etc/lilo.conf. É editando ele que você vai poder configurar o LILO conforme as suas necessidades. Um arquivo comum para uma máquina que possui dois sistemas (Linux e Windows) segue no exemplo abaixo:

```
# Arquivo de configuração do LILO

### Seção dos parâmetros globais do LILO ###
boot=/dev/hda
vga=normal

message=/boot/mensagem.txt
  prompt
  timeout=150

### Seção de partições do LILO ###

# Partição Linux
image=/boot/vmlinuz-2.2.17
  root=/dev/hda1
  label=Linux
  read-only

# Partição Windows
other=/dev/hdb1
  label=Windows
  table=/dev/hdb
```

Como você pôde ver no exemplo acima, a configuração do LILO é dividida em duas partes: configuração global e configuração de partições. A configuração global carregará consigo algumas informações sobre o funcionamento geral do LILO, como ele vai se comportar, aonde, quando, porque, cade, com quem e fazendo o que :) Na configuração de partições se define as opções de sistemas que o LILO vai bootar. Vamos ver passo-a-passo aqui as opções mostradas no exemplo acima, e também algumas opções adicionais que não estão no exemplo acima, mas você pode usar se quiser:

Curiosidades sobre o Linux.

Distribuição Ubuntu e suas derivações



Ubuntu é uma distribuição do Linux, sistema operacional baseado na distribuição Debian. Ubuntu deriva do conceito sul africano Ubuntu traduzido como: “humanidade para com os outros” ou “sou o que sou pelo que nós somos”.

O Ubuntu daqui a dois dias vai lançar a sua versão 8.04 e por isso vou falar um pouco de suas derivações. O Ubuntu é um dos S.O Linux que mais evoluíram e deram certo e com isso várias derivações foram criadas: Kubuntu, Edubuntu, Xubuntu e até uma versão usada pela Google o Goobuntu. Vamor ver um pouco de cada uma delas.



O Xubuntu é um derivado da distribuição Ubuntu que ao invés de utilizar a interface gráfica GNOME utiliza o ambiente XFCE, além de programas que o façam rodar satisfatoriamente em computadores mais antigos.



Kubuntu e Ubuntu- são o mesmo projeto, mas ao contrário do Ubuntu, que vem com o ambiente gráfico Gnome, ele vem com o KDE



Edubuntu é uma distribuição Linux desenvolvida pela Canonical Ltd baseada no SO Ubuntu para ambientes escolares. A meta fundamental é proporcionar o professor com conhecimento técnico limitado a habilidade de instalar um laboratório de informática, o Edubuntu tem esta capacidade porque tem um gerenciador de configuração, usuários e processos.



Gobuntu é uma distribuição de Linux, e uma das derivações oficiais do projeto ubuntu, sendo um sistema operativo que consiste somente estritamente em software livre, não incluindo “binary blobs” nem firmware que não seja livre.

O Fluxbuntu é uma distribuição de Linux baseada no Ubuntu 7.10 “Gutsy Gibbon” (lançada em Outubro de 2007) com o objectivo de ser mais leve e como tal funcionar em sistemas mais antigos, ou em sistemas modernos a grande velocidade. O Fluxbuntu usa o Fluxbox



Goobuntu (Google + Ubuntu), chamado também de Google OS, é o nome dado a versão do sistema operacional criado pelo Google baseado na distribuição Ubuntu do Linux para uso nos desktops da empresa.

Fonte: <http://www.ubuntu-br.org/>

3.1 Fedora Live USB Creator



Gostaria de usar o Linux, mas tem medo por não conhecer o sistema operacional ou por já está acostumado com o Windows. Bom tem várias alternativas para você usar o Linux, sem danificar o seu computador e desinstalar o S.O da Microsoft. Uma delas é Bootar o Linux pelo Pen Drive, usando o Fedora Live USB Creator, é uma ótima alternativa, tem todas as funcionalidades de um bom sistema operacional leve e gratuito, e o melhor, poderá levar seu S.O portátil para qualquer lugar. Instalei no meu pen drive de 2Gb com a versão do KDE e funcionou muito bem, ficou leve, bem rapidinho, com uma interface gráfica boa, reconheceu todo o meu hardware, inclusive a rede wireless e já vem com vários aplicativos instalados como: editor de textos, gráficos, tocador de mídia e muito mais. Aconselho a baixar e testar.



Fonte para baixar o programa

<http://www.baixaki.com.br/download/fedora-live-usb-creator.htm>

Fonte do artigo.

<http://profpisco.com/?p=2201>