

Fatec



SÃO PAULO
GOVERNO DO ESTADO

Faculdade de Tecnologia do Estado de São Paulo

Apostila de GNU/Linux

Instrutor: Professor Lucas Pereira da Rocha

Sumário

A evolução do GNU/Linux	3
Entendendo o Software Open Source e suas licenças.....	3
Sistemas operacionais populares.....	3
O Básico sobre linha de comando.....	5
Usando a linha de comando para conseguir ajuda	6
Manipulando arquivos do sistema.....	8
Estrutura de Diretórios GNU/Linux.....	10
Encontrar arquivos de sistema	13
Fluxos, pipes (canalização) e redirecionamentos de saída.....	13
Processar fluxos de texto usando filtros	14
Manipulando shell.....	15
Editores de Texto.....	16
Gerenciamento de Pacotes	17
Instalação de Programas com DPKG.....	19
Manipulação de contas de usuário/grupos.....	20
Gerenciamento de Processos.....	21
Manipulando permissões.....	22
Identificando o hardware do sistema operacional	23
Criando Partições no HD	25
Montar disco de armazenamento no GNU/Linux	27
Manipulando permissões.....	27
Configuração da rede	28
Arquivos de log.....	30

A evolução do GNU/Linux

O GNU/Linux é a integração de dois projetos de sucesso, o primeiro a se apresentar foi o projeto GNU por **Richard Stallman**, que tinha o sonho de criar um sistema operacional gratuito que poderia ser utilizado de forma gratuita, o projeto começou com o desenvolvimento de softwares como edição de texto, compiladores, mas na época ainda faltava a parte principal do sistema operacional, o kernel. O segundo projeto foi quando o um jovem finlandês chamado Linux Torvalds disponibilizou para o mundo a primeira versão do Linux, um kernel “Unix-like”, a partir desses projetos foi possível unir os softwares GNU com o Kernel, formando GNU/Linux.

O sucesso do projeto GNU/Linux ocorreu pela filosofia de desenvolvimento por meio colaborativo, o sistema operacional e os softwares contínuos nele são com códigos abertos e protegido por lei para não serem comercializados.

Entendendo o Software Open Source e suas licenças

Software Open Source é um software que pode ser compartilhado, modificado e utilizado desde que não seja comercializado em nenhuma fase apresentada anteriormente. A licença principal para código aberto é

GNU GPL - GNU General Public License ([Licenças - Projeto GNU - Free Software Foundation](#)) sua filosofia é:


- Liberdade 0** - liberdade para rodar o programa para quaisquer propósitos;
- Liberdade 1** - liberdade para estudar como o programa trabalha e adaptá-lo às suas necessidades;
- Liberdade 2** - liberdade de redistribuir cópias de forma que você possa ajudar outras pessoas;
- Liberdade 3** - liberdade para melhorar o programa e disponibilizar as melhorias para o público;
- Liberdade 4** comunidade possa se beneficiar. Ter acesso ao código fonte é essencial também para isso.

Obs.: se você utilizar algum software licenciado sob a licença GPL, você será obrigado a distribuir o código fonte.

Sistemas operacionais populares

Uma distribuição é a utilização do kernel Linux e softwares GNU desenvolvidos por outras comunidades ou grupos, para criar um sistema operacional personalizado.

Atualmente existem centenas de distribuições, algumas mais famosas que outras. Em sua maioria, as distribuições GNU/Linux são mantidas por grandes comunidades de colaboradores:

V · D · E	Distribuições Linux		[Esconder]
Android	/e/ · Android-x86 · Fire OS · LineageOS (CyanogenMod) · MIUI · One UI · Replicant		
Arch Linux	Antergos · ArchOne · Chakra · EndeavourOS · Hyperbola GNU · Manjaro Linux · Parabola GNU		
Debian	AlineX · Crunchbang · Damn Small Linux · Debian-BR-CDD · Deepin · Demolinux · Devuan · Dreamlinux · Elivecd · Finnix · gNewSense · GnuLinEx · Kali Linux · Libertas · Linux Caixa Mágica · Linspire · Mk-Auth · Muriqui Linux · Musix · MX Linux · Progeny Debian · Raspbian · Resulinux · Rxart · Sacix · Skolelinux · Slax · SparkyLinux · SteamOS · Tails · The Linux Router Project · Univention Corporate Server · Webconverger		
	Knoppix	Arco-Debian Linux · Gnoppix · Kanotix · Kurumin · Linux Knoppix STD · Paipix	
		Oficiais: Gobuntu · Ubuntu Kylin · Ubuntu Mobile · Ubuntu Server · Ubuntu Touch · Ubuntu TV Comunidade: Edubuntu · Fluxbuntu · Kubuntu · Lubuntu · Ubuntu Budgie · Ubuntu GNOME · Ubuntu MATE · Ubuntu Studio · Xubuntu	
	Ubuntu	Derivadas: BackBox · BackTrack · elementary OS · Emmabuntüs · Goobuntu · GOS · Ichthux · Jolicloud · KDE neon · Linuxaccessivel · Linux Educacional · LinuxFX · Linux Lite · Linux Mint · Mangaka Linux · Mythbuntu · Peppermint OS · Pinguy OS · Pop! OS · Poseidon Linux · Sabily · Symphony OS · Super OS · Trisquel · Ultimate Edition · ZeVenOS · Zorin OS · Nova (sistema operacional)	
	Xubuntu	Emmabuntüs · GalliumOS · UberStudent	
Fedora	BLAG Linux and GNU · Fox Linux · Freedows · Qubes OS · Yellow Dog Linux		
Red Hat	Asianux · CentOS · Conectiva · Oracle Linux · Rocky Linux · Scientific Linux · Turbolinux · Red Flag Linux · Red Star OS		
Mandriva	ALT Linux · Mageia · PCLinuxOS		
Gentoo Linux	Chromium OS (Chrome OS) · Funtoo Linux · Guaranix · Litrix Linux · Pentoo · Sabayon Linux · Ututo · VidaLinux		
Slackware	Cytrun Linux · Draco GNU/Linux · Kya Linux · MoviX · Vector Linux · Zenwalk		
Independentes	Alpine Linux · Coyote Linux · CRUX · GoboLinux · Guix System Distribution · Lunar Linux · Parted Magic · Puppy Linux · openSUSE (SUSE Linux Enterprise) · SliTaz · Solus		
Outras	BrazilFW · Dizinha Linux · Dual OS · Dingux · Dyne:bolic · Endian Firewall · FREESCO · F.I.R.E · Foresight Linux · Hikarunix · IPCop · Lightweight Portable Security · Lycoris Desktop/LX · MeeGo · Mobilinux · PHLAK · PS2 Linux · RouterOS · SLS Linux · Tizen · Xandros		
<div><div> Categoria</div><div> Comparação</div><div> Lista</div><div> Commons</div></div>			

Fonte (Wikipedia) [SUSE Linux – Wikipédia, a enciclopédia livre \(wikipedia.org\)](https://pt.wikipedia.org/wiki/SUSE_Linux)

Distribuições Livres de Custos

Mantidas por comunidades de colaboradores sem fins lucrativos:

Exemplos são: Debian, Slackware, Gentoo, Knoppix e CentOS, entre outras.

Distribuições Corporativas

Mantidas por empresas que vendem o suporte ao seu sistema.

Exemplos: RedHat, Ubuntu, Suse e Mandriva.

Distribuições Live

São distribuídas em mídias com o intuito de rodarem a partir delas, sem a necessidade de serem instaladas no disco de armazenamento.

Distribuições From Scratch

São desenvolvidas do zero, ou seja, utilizam um kernel Linux, alguns programas GNU e a grande maioria das suas particularidades é desenvolvida especificamente para ela.

Exemplos: Debian, RedHat, Gentoo, Slackware.

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

Distribuições Provenientes (Baseadas)

Aproveitam ferramentas e bases já desenvolvidas por outras distribuições. Distribuições baseadas usam distribuições "From Scratch" para alcançar seus objetivos mais rápido, dando maior atenção ao propósito da distribuição.

Exemplos: Ubuntu, Dre-amLinux, Kubuntu, Slax e Linux Mint.

O Básico sobre linha de comando

Terminal Virtual

"Terminal virtual" é a utilização do sistema operacional em modo texto (sem interface gráfica), com essa funcionalidade o usuário (que pode ser multi-usuários) terá condições de utilizar recursos disponíveis no GNU/Linux.

Logon

Logon é a entrada do usuário no sistema operacional.

Shell

Interface entre o usuário e o kernel do sistema e por meio dele, podemos digitar os comandos. O "Shell" padrão do GNU/Linux é o **Bash**.

Super usuário: Chamado de "root" é o administrador do sistema, e seu diretório (pasta) padrão é o `"/root"`, o "Shell" é identificado com `"#"` (jogo da velha).

Usuário comum: Não tenha poderes administrativos no sistema. O diretório padrão para os usuários é o `"/home"`. O "Shell" é identificado com `"$"` (cifrão).

Histórico de comandos

Comando history

O terminal armazena os últimos 500 comandos digitados no terminal .

```
#history
```

Logout

Logout é a saída do sistema. Ela é feita por um dos comandos abaixo:

```
$ logout
```

```
$ exit
```

```
$ <CTRL>+D
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

Desligando o Computador

Para desligar o computador pode-se utilizar um dos comandos abaixo:

```
# shutdown -h now
```

Opções:

-h para desligar;

-r para reiniciar.

Comandos essenciais para a nossa movimentação dentro do sistema.

```
# pwd ( O comando "pwd" exibe o diretório atual )
```

```
# ls ( O comando "ls" é utilizado para listar o conteúdo dos diretórios )
```

O comando "**cd**" é utilizado para entrar ou sair da estruturas de pastas do sistemas

```
# cd ( Entra no diretório HOME do usuário )
```

```
# cd ~ ( Entra no diretório HOME do usuário )
```

```
# cd / ( Diretório raiz do sistema, inicio da estrutura de diretórios )
```

```
# cd /etc ( Diretório dos arquivos de configuração do sistema operacional )
```

```
# cd .. ( Retorna ao diretório anterior )
```

```
# cd - ( Sobe um nível na árvore de diretórios )
```

```
# cd /usr/share/doc ( Entra em um diretório específico )
```

Usando a linha de comando para conseguir ajuda

O GNU/Linux nos proporciona varias formas de obter ajuda, são elas:

How-to's (Passo a passo)

Os "How-to's" são documentos que focam uma necessidade específica, como montar um "firewall", instalar uma "Placa de rede", configurar placas de rede, configurar um servidor web e muitos outros. O diretório de "How-to's" do GNU/Linux é o "**/usr/share/doc**".

Se desejamos saber como configurar um "firewall", podemos consultar os arquivos do diretório:

```
# cd /usr/share/doc/iptables/
```

Comandos de ajuda

Existem diversos comandos de ajuda no GNU/Linux, vamos ver alguns:

Para comandos externos, o “help” aparece como parâmetro. Por exemplo:

```
# [comando] --help
```

```
# ls --help
```

```
# cp --help
```

```
# ip --help
```

Obs.: O parâmetro --help só funciona com o comando e não comando + parâmetro. Ex.: **ip add --help**

Comando whatis

O comando “**whatis**” busca informações do comando.

```
# whatis vim
```

```
# man -f vim ( equivalente )
```

Comando man

O comando “**man**” contém toda a documentação de um software, comando, arquivo de configuração, bibliotecas e separadas por níveis.

Os manuais do sistema são divididos nos seguintes níveis:

man 1 -> Programas executáveis e comandos do “Shell”;

man 2 -> Chamadas de sistema (funções providas pelo Kernel);

man 3 -> Chamadas de bibliotecas (funções como bibliotecas do sistema);

man 4 -> Arquivos de dispositivo (Localizados normalmente no “/dev”);

man 5 -> Arquivos de configuração e convenções;

man 6 -> Jogos;

man 7 -> Variados (incluindo pacotes de macros e convenções);

man 8 -> Comandos de administração do sistema (normalmente usado so-mente pelo root);

man 9 -> Rotinas de Kernel.

```
# man [seção] [comando] ( Sintaxe do comando “man”)
```

```
# man <nível> intro ( Todos os níveis de manuais possuem sua determinada introdução)
```

```
# man passwd ( abre o local: “/usr/share/man/man1” para o “passwd”)
```

```
# man 5 passwd ( Veremos agora o manual do arquivo de usuários “passwd”)
```

Podemos consultar quais manuais estão disponíveis dentro do próprio diretório:

```
# ls /usr/share/man/
```

Já se nosso sistema estiver em inglês é preciso usar o parâmetro “-L pt_BR”, para que possamos ver os manuais em nosso idioma:

```
# man comando ( Ver manual no idioma da instalação do sistema operacional )  
# man -L pt_BR comando ( -L pt_BR opção para ver o manual em português. SO em inglês )
```

Comando whereis

O comando “**whereis**” é utilizado para mostrar a localização do binário, arquivo de configuração (caso exista) e a localização das páginas de manuais.

```
# whereis <arquivo>  
# whereis vim ( localização das páginas de manuais do comando “vim” )
```

Comando which

O comando “**which**” mostra a localização do binário do comando.

```
# which <comando>  
# which vim
```

Manipulando arquivos do sistema

O comando ls

O comando “ls” lista os arquivos ou diretórios (quem criou, data de criação, tamanho, dono e grupo):

```
# ls -l /  
drwxr-xr-x 4 lucas FATEC 144 Sep 10 10:48 / boot
```

Analisando a saída temos:

drwxr-xr-x	indica que é um diretório (Pasta)
-rwxr-xr-x	indica que é arquivo
drwx-----	indica a permissão para o dono do arquivo
d---rwx---	indica a permissão para o grupo do arquivo
d-----rwx	indica a permissão para o outros do arquivo
lucas FATEC	indica que o dono da pasta é o usuário lucas e o grupo é FATEC
sep 10 07:40	indica data e hora da criação
/boot	pasta boot do sistema

Algumas opções para o comando ls:

```
# ls -la / root ( A opção “-a” lista todos arquivos, inclusive os ocultos )  
# ls -lha /etc/*.conf ( lista arquivos que termina com .conf )  
# ls -ld /* n ( Lista todos diretórios que termina com n )
```


Criar arquivo

Para criar arquivo utilize o comando **"touch"**:

```
# touch fatec.txt
```

```
# touch notas.pdf notas.pdf notas3.pdf notas4.pdf
```

Obs.: Outra opção para criar um arquivo é utilizar o editor de texto e salvá-lo.

Curingas

São expressões regulares que podem substituir uma palavra completa , somente uma letra, seja para listar, copiar, apagar, etc. São usados cinco tipos de curingas no GNU/Linux:

Os 5 tipos de curingas mais utilizados (*, ?, [], , !) podem ser usados juntos.

* - Utilizado para um nome completo ou restante de um arquivo/diretório;

? - Esse curinga pode substituir uma ou mais letras em determinada posição;

! - exclui da operação

```
# touch curriculo{1,2,3}.txt curriculo{4,5}.new
```

```
# ls ( Listar os arquivos )
```

```
curriculo1.txt curriculo2.txt curriculo3.txt curriculo4.newcurriculo5.new
```

```
ls * ( "*" para visualizar todos os arquivos do diretório )
```

```
curriculo1.txt curriculo2.txt curriculo3.txt curriculo4.newcurriculo5.new
```

```
# ls *new* ( Listar todos os arquivos que tenham "new" no nome )
```

```
curriculo4.new curriculo5.new
```

```
# ls *.txt ( Lista arquivos que terminam com .txt )
```

```
curriculo1.txt curriculo2.txt curriculo3.txt procedimentos.txt
```

```
# ls curriculo?.txt Lista arquivos com o nome curriculo e terminem com .txt
```

```
curriculo1.txt curriculo2.txt curriculo3.txt
```

Para listar todos os arquivos que começam com o nome "curriculo", tenham qualquer caractere entre o número "1-3" e terminem com ".txt".

```
# ls curriculo[1-3].txt
```

```
curriculo1.txt curriculo2.txt curriculo3.txt
```

Listar todos .txt exceto o curriculo2.txt :

```
# ls curriculo [!2]. txt curriculo1.txt curriculo3.txt
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

Para listar os arquivos “curriculo4.new” e “curriculo5.new” podemos usar os seguintes métodos:

```
# ls *.new# ls *new*
# ls curriculo?.new
# ls curriculo[4,5].*
# ls curriculo[4,5].new
```

Criando diretórios

O comando “**mkdir**” é utilizado para criar um diretório no sistema:

```
# mkdir RH
# mkdir RH/Processos
# mkdir RH/Processos/curriculos
```

A opção “-p” permite que toda essa estrutura seja criada em uma única linha:

```
# mkdir -p RH/Processos/curriculos
```

Removendo arquivos/diretórios

O comando “**rm**” é utilizado para apagar arquivos, diretórios e subdiretórios estejameles vazios ou não:

```
/# rm -ri curriculo.txt      ( Remover o curriculo.txt com confirmação )
# rm -rf RH                  ( Remover a pasta RH sem confirmação )
```

Copiar arquivos/diretórios

O comando “**cp**” serve para fazer cópias de arquivos e diretórios:

```
# cp arquivo-origem arquivo-destino
# cp arquivo-origem caminho/diretório-destino/
# cp -R diretório-origem nome-destino
# cp -R diretório-origem caminho/diretório-destino/
```

4.1.7 Mover ou renomear arquivos/diretórios

O comando “**mv**” serve tanto para renomear um arquivo quanto para movê-lo:

```
# mv arquivo novo-nome
# mv diretório caminho/diretório-destino/
# mv teste teste4      ( Renomeando arquivo )
# mv teste4 /tmp        ( Movendo arquivo )
```

Estrutura de Diretórios GNU/Linux

Os diretórios do GNU/Linux tem a seguinte estrutura, cada diretório tem uma função específica para o sistema operacional.

/									
bin	cdrom	etc	lib	mnt	proc	root	var	sys	
boot	dev	home	media	opt	sbin	srv	tmp	usr	

Diretório /

ls --color / Diretório raiz que ficam todos os demais diretórios do sistema.

Diretório /bin

O diretório “/bin” guarda os comandos essenciais para o funcionamento do sistema (/bin/ls; /bin/cp; /bin/mkdir; /bin/cat;).

ls /bin

Diretório /boot

No diretório “/boot” estão os arquivos estático necessários à inicialização do sistema e o gerenciador de “boot”.

ls / boot (local de arquivos estáticos para inicialização do sistema)

Diretório /dev

No diretório “/dev” ficam todos os arquivos de dispositivos. O GNU/Linux faz a comunicação com os periféricos por meio de “links” especiais que ficam armazenados nesse diretório, facilitando assim o acesso aos mesmos.

ls /dev

Diretório /etc

No diretório “/etc” estão os arquivos de configuração do sistema. Nesse diretório vamos encontrar vários arquivos de configuração, tais como: “scripts” de inicialização do sistema, tabela do sistema de arquivos, configuração padrão para “logins” dos usuários, etc.

ls /etc

cat /etc/passwd

Diretório /lib

No diretório “/lib” estão as bibliotecas compartilhadas e módulos do kernel. As bibliotecas são funções que podem ser utilizadas por vários programas.

ls /lib (Cada kernel tem seus próprios módulos, que ficam em: /lib/modules/<versão do kernel>/kernel)

uname -r (Versão do Kernel)

lsmod (Para visualizar os tipos de módulos)

Diretório /media

ls /media

Ponto de montagem para dispositivos removíveis, tais como:

Hd / cd / dvd / disquete / pendrive / câmera digital

Diretório /mnt

Este diretório é utilizado para montagem temporária de sistemas de arquivos.

ls /mnt

Diretório /opt

Normalmente, é utilizado por programas proprietários ou que não fazem parte oficialmente da distribuição.

```
# ls /opt
```

Diretório /sbin

O diretório “/sbin” guarda os comandos utilizados para inicializar, reparar, restaurar e/ou recuperar o sistema. Isso quer dizer que esse diretório também contém comandos essenciais, mas os mesmos são utilizados apenas pelo usuário administrador “root”. Entre os comandos estão:

```
# ls /sbin
```

```
( /sbin/halt; /sbin/ifconfig; /sbin/init; /sbin/iptables )
```

Obs.: Os usuários comuns não podem executar comandos do /sbin que alterem o sistema, apenas alguns para visualização.

Exemplo:

Visualizar IP configurado na placa eth0:

```
$ /sbin/ifconfig eth0 ( Alterar IP da placa de rede )
```

```
$ /sbin/ifconfig eth0 192.168.200.100
```

Diretório /srv

```
#ls /srv ( Diretório para dados de serviços fornecidos pelo sistema )
```

```
# cd /srv/www ( servidor web )
```

```
# cd /srv/ftp ( servidor ftp )
```

Diretório /tmp

```
# ls /tmp ( Diretório para armazenamento de arquivos temporários )
```

Diretório /usr

O diretório “/usr” contém programas que não são essenciais ao sistema e que seguem o padrão GNU/Linux. Por exemplo, navegadores, gerenciadores de janelas, etc.

```
# ls /usr
```

Fique atento as diferenças entre:

/bin - binários essenciais ao sistema

/usr/bin - binários não essenciais ao sistema

/usr/local/bin - scripts criados pelo usuário

Diretório /var

O diretório “/var” local onde o sistema armazena arquivos de informações do sistemas, mais conhecido como “logs”.

```
# ls /var
```

```
# ls /var/log/syslog ( Arquivo principal de logs, para ver todos entre em /etc/syslog )
```

Diretório /proc

O “/proc” é um diretório virtual, mantido pelo **kernel**, onde encontramos a configuração atual do sistema, dados estatísticos, dispositivos já montados, interrupções, endereços e estados das portas físicas, dados sobre as redes, etc.

```
# ls /proc
# more /proc/interrupts    ( Informações das IRQs dos dispositivos.)
# more /proc/meminfo       ( Informações da memória )
```

Diretório /sys

Dentro do diretório “/sys” podemos encontrar o quase o mesmo conteúdo do “/proc”, mas de uma forma bem mais organizada para nós administradores.

```
# ls /sys
```

Diretórios /home e /root

```
# ls /home /root
```

O “/home” contém os diretórios pessoais dos usuários cadastrados no sistema.

O “/root” é o diretório pessoal do super usuário “root”.

Encontrar arquivos de sistema

Find

O comando “find” procura por arquivos/diretórios no disco, ao adicionar parametros se pode obter data de modificação, tamanho, etc.

```
find [diretório] [opções/expressão]
# find /etc -name *.Conf
```

Limita a recursividade de busca na árvore de diretórios.

```
# find /etc -maxdepth 1 -name *.Conf    ( a busca será feita apenas no diretório, sem buscar por diretório )
```

Caso seja antecedido por “-”, procura por arquivos que foram acessados entre [num] minutos atrás eo momento atual.

```
# find / -uid 1000          ( Procura por arquivos que pertençam ao usuário com o “uid 1000” [num]. )
# find / -user aluno       ( Procura por arquivos que pertençam ao usuário “aluno” [nome] )
# find / -perm 644         ( Procura por arquivos que possuem os modos de permissão [modo] )
# find / -size +1M         ( Procura por arquivos que tenham o tamanho [num] )
# find . -size +1000k      ( Procura no diretório atual e nos subdiretórios com que1000 kbytes (1Mbyte) )
```

Fluxos, pipes (canalização) e redirecionamentos de saída

IMPORTANTE: No mundo GNU/Linux, a maioria das operações são realizadas por meio de scripts.

Trabalhando com entrada e saída de dados

Formas de redirecionar o fluxo de dados:

> (maior): Direciona a saída do comando para um arquivo, substituindo o seu conteúdo;

>> (maior-maior): Direciona a saída do comando para um arquivo, adicionando o texto ao final do arquivo;

| (barra vertical ou pipe): Utiliza a saída do primeiro comando como argumento dos segundo comando; tee: mostra saída na tela e redireciona para um arquivo ou outra localização ao mesmo tempo;

O direcionador >

O direcionador > encaminha a informação uma determinada atividade (Sobreescreve as informações que já estão no arquivo)

```
# ls / > raiz
```

```
# cat raiz
```

O direcionador >>

O direcionador >> encaminha a informação uma determinada atividade, adicionando no fim o conteúdo. (Adiciona a nova informação no final do arquivo)

```
# ls / >> hoje      ( O resultado do comando ls foi adicionada ao final do arquivo hoje )
```

O direcionador |

Conhecido como pipe, sua função é retornar uma informação através do uso de dois ou mais comandos

Primeiro visualize o conteúdo do arquivo /etc/passwd:

```
# cat /etc/passwd
```

Resultado do junção do comando CAT e TR, o retorno será a troca das letras de minúsculas para maiúsculas

```
# cat /etc/passwd | tr "a-z" "A-Z"
```

Processar fluxos de texto usando filtros

Uma necessidade constante dos administradores é encontrar informações dentro dos arquivos filtros se tornam necessários para a procura de informações.

Filtragem : grep e egrep

Comando grep

```
# grep root /etc/passwd
```

Resposta do comando: root:x:0:0:root:/root:/bin/bash

```
# grep -v bash      ( Busca por tudo menos bash -v a palavra )
```

```
# grep -in debian /etc/passwd      ( Para buscar a palavra "Debian" no arquivo "/etc/passwd" utilize a opção "-i" para ignorar maiúsculas e minúsculas)
```

Resposta do comando: 60:Debian-exim:x:123:132::/var/spool/exim4:/bin/false

Dica:

-i - Ignora diferença entre maiúsculas e minúsculas;

-n - Mostra o número de cadalinha encontrada;

```
# grep -i debian /etc/passwd -A 2 -B 1 ( busca as duas próximas linhas e uma linha anterior )
```

6.1.3 Comando egrep

Para uma busca mais avançada utilize o “egrep”. Para realizar uma busca em uma linha que contenha a palavra “root” ou “aluno”:

```
# egrep "root|debian" /etc/passwd ( busca avançada por uma linha que tenha “root” ou “aluno” )
```

```
# egrep [dD]ebian /etc/passwd ( contenham a palavra Debian ou debian )
```

```
# egrep ^b /etc/passwd ( Procurar por linhas que começam com a letra “b” )
```

```
# egrep false$ /etc/passwd ( Busca por linha que terminam com a palavra “false” )
```

Manipulando shell

Arquivos de Login

Quando uma “bash” é executada como uma “shell” de “login” interativo ela lê e executa o arquivo “/etc/profile”, se ele existir. Esse arquivo deve conter as configurações gerais que se aplicam a todos os usuários do sistema. Após ler o “/etc/profile”, ela irá procurar por um dos arquivos:

```
~/.Profile    ~/.Bashrc    ~/.Bash_history
```

Arquivos /etc/issue e /etc/motd

Os arquivos “/etc/issue” e “/etc/motd” são usados para mostrar mensagens para os usuários e não interferem na parte operacional do sistema. A diferença entre os arquivos “/etc/issue” e “/etc/motd”, é que o primeiro exibe uma mensagem para o usuário antes que o mesmo faça “login” no sistema, enquanto o segundo exibe uma mensagem após o usuário se “logar” no sistema. Há ainda o arquivo “/etc/issue.net”, que contém a mensagem exibida em “logins” remotos.

```
# cat /etc/motd
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

Veja um exemplo de “/etc/issue” no Debian:

```
# cat /etc/issue
```

```
Debian GNU/Linux 11 \n \l
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

Os caracteres "(n)" e "(l)" no arquivo "/etc/issue" representam respectivamente o nome do servidor e do terminal em que o usuário está logado.

Editores de Texto

Introdução teórica

A grande maioria das configurações em sistemas GNU/Linux são feitas editando diretamente arquivos de configuração em modo texto, mediante a isso, conhecer as funcionalidade de um editor de texto pelo shell é primordial para edição e produtividade.

Editor de texto VIM

Ao digitar vim seguido da tecla ENTER, o editor é inicializado, o texto inserido é todo editado por comandos no teclado, conforme tabela abaixo:

```
1 # vim texto
2 # Para inserir digite: i
3 i => Entra no modo de inserção antes do cursor
4
5 # Para sair do modo de inserção digite: ESC
6
7 # Para inserir uma linha abaixo do cursor digite: o
8 o => Insere uma linha abaixo do cursor e entra no modo de inserção
9
10 # Para sair do modo de inserção digite: ESC
11
12 # Para inserir uma linha acima do cursor digite: O
13 O => Insere uma linha acima do cursor e entra no modo de inserção
14
15 # Para sair do modo de inserção digite: ESC
16
17 # Para desfazer a última alteração digite: u
18
19 # Para refazer digite:      CTRL+R
20
21 # Para numerar as linhas digite: :set number
22
```



```
23 # Para copiar a segunda linha digite: :2y
24
25 # Para colar na linha abaixo do cursor digite: p
26
27 # Para ir para a primeira linha digite: gg
28
29 # Para colar na linha acima do cursor, "3vezes" digite: 3P
30
31 # Para salvar as alterações digite: :w
32
33 # Para sair do arquivo sem salvar digite: :q ou para forçar aida sem salvar: :
    q!
```

Comandos para salvar o texto:

:wq ou **:x** - Salva o arquivo e sai 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.

Gerenciamento de Pacotes

O que é um pacote?

Os diversos programas para GNU/Linux são distribuídos em forma de pacotes específicos para cada distribuição. Pacotes são conjuntos de binários pré-compilados, bibliotecas, checagem de dependências, arquivos de configuração, que são instalados facilmente no sistema operacional.

Os pacotes nos sistemas baseados em Debian têm uma extensão característica: “.deb”.

Gerenciador de pacotes?

Gerenciador de pacotes tem a função de interpretar a necessidade de cada um dos pacotes, para que eles possam ser instalados e/ou removidos.

Gerenciando Pacotes no Debian

Gerenciador de pacotes utilizado no Debian é “**aptitude**” ou “**apt-get**”, seu arquivo de configuração está localizado no “**/etc/apt/sources.list**”. O arquivo “**/etc/apt/sources.list**” contém os locais onde o “APT” encontrará os pacotes, a versão da distribuição que será verificada (stable, testing, unstable) e a seção que será copiada (main, non-free, contrib, non-US).

Instalação, Remoção e Atualização

Os comandos acima sincronizam a lista de pacotes disponíveis para instalação nos servidores remotos, com uma lista local:

```
# aptitude update
```

```
# apt-get update
```

Para procurarmos por um pacote que desejamos instalar, podemos fazer uma busca pelo comando abaixo:

```
# aptitude search <argumento>
```

```
# apt-cache search <argumento>
```

Obter informações sobre um pacote conhecido:

```
# aptitude show <pacote>
```

```
# apt-cache show <pacote>
```

Instalar pacote

```
# apt-get install pacote
```

```
# aptitude install pacote
```

Remover pacote:

```
# aptitude remove pacote
```

```
# apt-get remove pacote
```

Remover arquivos de configuração e suas dependências no sistema:

```
# aptitude purge pacote
```

```
# apt-get autoremove --purge pacote
```

Removendo pacotes que não serão mais usados

Quando você instala um pacote o apt busca das fontes listadas em **"/etc/apt/sources.list"** os arquivos necessários e os guarda em um repositório local **"/var/cache/apt/archives/"**, e então faz a instalação. Em algum tempo o repositório local pode crescer e ocupar muito espaço em disco. Felizmente o apt fornece uma ótima ferramenta para lidar com seu repositório local, o método "clean" do apt-get.

```
# apt-get clean
```

Atualizar pacotes instalados

Para atualizar os pacotes já instalados, para a última versão que está no repositório:

```
# aptitude upgrade
```

```
# apt-get upgrade
```

Atualização da distro

O sistema pode ser atualizado de tempos em tempos ou por questões de segurança. Para instalar todas as atualizações disponíveis, usa-se o “aptitude” com a instrução “safe-upgrade”. Dependendo da velocidade de conexão, este processo pode levar bastante tempo.

```
# aptitude safe-upgrade
```

Instalação de Programas com DPKG

Pacotes Debian - DPKG

O DPKG é um programa que é a base do Sistema de Gerenciamento de Pacotes para distribuições GNU/Linux baseadas em Debian capaz de instalar, remover e manipular arquivos .deb>

DPKG

```
# dpkg --help
```

Verifique se ele está instalado no sistema:

```
# dpkg -l <programa>
```

```
# dpkg -l | grep <programa>
```

Verifique quais programas estão instalados no sistema:

```
# dpkg -l | less
```

Verifique o status de um pacote instalado:

```
# dpkg -s pacote
```

Instale o programa pacote.deb:

```
# dpkg -i pacote.deb
```

Determine onde estão instalados todos os arquivos do aplicativo pacote:

```
# dpkg -L pacote | less
```

Determine onde serão instalados todos os arquivos do programa flashplayer:

```
# dpkg -c pacote.deb | less
```

Remova o programa pacote

```
# dpkg -r pacote
```

Verifique que foi removido:

```
# dpkg -l
```

Apague seus arquivos de configuração:

```
# dpkg -P pacote
```

DICA:

Alguns pacotes têm problemas de dependências e não são instalados, até que suas dependências sejam satisfeitas, para isso utilize o apt-get ou aptitude com a opção -f para resolver. Por exemplo, vamos tentar instalar o google-chrome, faça o download do pacote.deb:

```
# dpkg -i google-chrome-stable_current_i386.deb
```

O aplicativo não pode ser instalado porque existem pré-requisitos para sua instalação, para resolver estas dependências automaticamente, digite:

```
# aptitude -f install
```

```
# apt-get -f install
```

Manipulação de contas de usuário/grupos

Segurança Básica e Identificação de Tipos de Usuários

Criando Usuários

```
#adduser professor-LucasRocha ( cria login de forma automática, mas não aceita login com ponto (lucas.rocha) )
```

```
# useradd lucas.rocha -d /home/lucas.rocha -s /bin/bash ( comando useradd permite . mas terá que configurar )  
# mkdir /home/lucas.rocha ( Criar a pasta home para o usuário )  
# chown -R lucas.rocha:lucas.rocha /home/lucas.rocha ( Aplicar permissões para acesso apenas do usuário )  
# passwd lucas.rocha ( Criar senha do usuário )
```

```
#cat /etc/passwd ( ver os usuários no sistema )  
# id lucas.rocha ( informações id e grupos que o usuário participa )
```

Criar Grupos

```
# groupadd CPD
```

Adicionar usuário a um grupo:

```
#adduser lucas.rocha CPD
```

Verificar quais grupos o usuário lucas.rocha participa:

```
#groups lucas.rocha
```

Remover usuário do grupo CPD:

```
# deluser lucas.rocha CPD
```

Gerenciamento de Processos

Como fazemos para gerenciar os processos?

Os processos são gerenciados pelo kernel, o sistema operacional possibilita que verificamos quais processos estão sendo executados e até manipular os processos que são criados pelo usuário logado.

Visualizando processos:

```
#ps -aux
```

```
# pstree
```

```
#top ou top -i
```

Finalizando processos com killall

```
# killall -s <signal> <programa>
```

```
# killall cron ( kill +nome_do_programa )
```

O comando kill

```
# kill -<signal> <pid>
```

```
# kill -9 1267
```

O Comando lsof

O comando lsof é um dos mais importantes comandos para quem administra sistemas Linux, principalmente na área de segurança. Este comando lista todos os arquivos abertos por todos os processos

Ver se algum processo está escutando uma porta na rede suspeita, ou conectado ao IP suspeito. Por exemplo, vários scripts de invasão ficam escondidos no sistema (com nomes de outros processos), conectados a servidores de desconhecidos.

Comandos LSOFF

```
# lsof
```

```
# lsof -c k ( Por processos cujos nomes começam pela letra "k" )
```

```
# lsof -c bash ( processos cujo nome começa com "bash" )
```

```
# lsof -c bash -u ^aluno ( processos cujos nomes começam por "bash", mas exclui aluno )
```

```
# lsof -u aluno ( Nos mostrar os processos abertos pelo usuário aluno )
```

```
# lsof +p 1 ( Processos cujo PID é 1 )
```

```
# lsof +D /tmp ( Busca por todas as instâncias abertas do diretório /tmp )
```

```
# lsof -i :22 ( Conexões da porta 22, que é a porta do ssh )
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

Manipulando permissões

Conceitos permissões

Habilitando permissão por letras:

w – Habilita a escrita r – Habilita a leitura x – habilita a execução

Habilitando permissão por octetos:

- 1 → Representa a opção de execução (x);
- 2 → Representa a opção a opção de escrita (w);
- 4 → Representa a opção a opção de leitura (r).

Permissão por caracteres

d rw x-----	indica a permissão para o dono do arquivo	700
d--- rw x---	indica a permissão para o grupo do arquivo	070
d----- rw x	indica a permissão para o outros do arquivo	007

Permissão por octetos

Manipulando permissões em arquivos por caractere

```
chmod u+wx fatec.txt      ( apenas o dono do arquivo pode manipular o arquivo )
chmod g+wx fatec.txt      ( apenas o grupo do arquivo pode manipular o arquivo )
chmod o+wx fatec.txt      ( apenas outros do arquivo pode manipular o arquivo )
```

Manipulando permissões em arquivos por octeto

```
chmod 700 fatec.txt      ( apenas o dono do arquivo pode manipular o arquivo )
chmod 070 fatec.txt      ( apenas o grupo do arquivo pode manipular o arquivo )
chmod 007 fatec.txt      ( apenas outros do arquivo pode manipular o arquivo )
```

Manipulando permissões em pasta por caractere

```
chmod u+wx CPD      ( apenas o dono do arquivo pode manipular o arquivo )
chmod g+wx CPD      ( apenas o grupo do arquivo pode manipular o arquivo )
chmod o+wx CPD      ( apenas outros do arquivo pode manipular o arquivo )
```

Manipulando permissões em pasta por octeto

```
chmod 700 CPD      ( apenas o dono do arquivo pode manipular o arquivo )
chmod 070 CPD      ( apenas o grupo do arquivo pode manipular o arquivo )
chmod 007 CPD      ( apenas outros do arquivo pode manipular o arquivo )
```

Manipulando dono do arquivo/pasta e grupo

O comando para manipular o dono do arquivo/pasta e grupo é CHOWN

```
# ls -l
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA      ( Retorno do comando ls - l )
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA      ( Retorno do comando ls - l dono do arquivo )
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA      ( Retorno do comando ls - l responsavel pelo grupo )
```

Trocar o dono do arquivo de root para lucas:

```
# chmod lucas:root FATEC-PASTA
# ls -l
# drwx----- 1 lucas root 0 Oct 21 05:24 FATEC-PASTA      ( dono do arquivo é o usuário lucas )
```

Trocar o responsavel do grupo de de root para lucas:

```
# chmod lucas:lucas FATEC-PASTA
# drwx----- 1 lucas lucas 0 Oct 21 05:24 FATEC-PASTA      ( dono do arquivo é usuário lucas )
```

Identificando o hardware do sistema operacional

Introdução teórica

O kernel é o responsavel pela comunicação entre o interface utilizada pelo usuário (grafica ou modo texto) o hardware do computador. Sistema operacional GNU/Linux tem na sua estrutura a pasta /dev responsavel por organizar o hardware.

13.1.1 Explorando o /dev

Em sistemas como GNU/Linux existe o conceito de dispositivos; praticamente tudo na é tratado como sendo um dispositivo e pode ser acessado pelo seu respectivo arquivo localizado no diretório “/dev”.

O diretório “/dev” consiste de um “filesystem” (sistema de arquivos) especial e pode ser de dois tipos: “devfs” ou “udev”. O primeiro é mais antigo, tendo sido substituído pelo segundo a partir do kernel 2.6.12. Uma das diferenças entre os dois é que no “devfs” os arquivos de dispositivos são criados uma única vez, dessa forma, o diretório “/dev” contém os dispositivos para todos os hardwares suportados pelo Linux, não importando se eles existem de fato na máquina ou não. Com o “udev” os dispositivos são criados de acordo com a disponibilidade no sistema. Dessa forma, o diretório contém apenas os arquivos de dispositivo para o “hardware” presentes na máquina.

```
# lspci
```

```
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
```

```
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
```

```
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | E-mail: lucas.rocha50@fatec.sp.gov.br

00:02.0 VGA compatible controller: VMware SVGA II Adapter

00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)

00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service

00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)

00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB

00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)

lsusb

Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet

Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

free -m

```
root@debian:/home/user# free -m
              total        used         free       shared    buff/cache   available
Mem:           3581         734          819         237        2027        2369
Swap:           0           0           0
```

du -h

```
srvspo02:~ # df -h
Filesystem      Size
/dev/sda2       45G
udev            1.8G
/dev/sda3       2.0G
/dev/sda4       1.3T
/dev/sdb1       1.8T
/dev/sdc1       1.8T
/dev/sdc2       1.8T
/dev/sdc3       1.8T
/dev/sdd1       1.4T
```

O que é uma partição?

Uma partição é um espaço do disco que se destina a receber um sistema de arquivos no GNU/Linux pode ser EXT2, EXT3, EXT4

Tipos de partições

Existem três tipos possíveis de partições: primária, estendida e lógica.

Dispositivos de armazenamento no GNU/Linux

O padrão dos dispositivos “SATA” e “SCSI” começam por “sd”.

/dev/sda (Primeiro HD SATA)

/dev/sda1 (Primeiro HD SATA – PRIMEIRA PARTIÇÃO)

/dev/sda2 (Primeiro HD SATA – SEGUNDA PARTIÇÃO)

/dev/sda3 (Primeiro HD SATA – TERCEIRA PARTIÇÃO)

/dev/sda4 (Primeiro HD SATA – QUARTA PARTIÇÃO)

/dev/sdb (Segundo HD SATA)

/dev/sdb1 (Segundo HD SATA – PRIMEIRA PARTIÇÃO)

/dev/sdb2 (Segundo HD SATA – SEGUNDA PARTIÇÃO)

/dev/sdb3 (Segundo HD SATA – TERCEIRA PARTIÇÃO)

/dev/sdb4 (Segundo HD SATA – QUARTA PARTIÇÃO)

df -h (verificar os disco de armazenamento)

fdisk -l (Informações do disco, montagem, tamanho)

cfdisk (Gerenciador de particionamento de disco)

Criando Partições no HD

Particionamento com CFDISK

A ferramenta “cfdisk” é um particionador por linha de comando e varios recursos.

cfdisk /dev/sda

```

Disk: /dev/sda
Size: 8 GiB, 8589934592 bytes, 16777216 sectors
Label: gpt, identifier: C9FC07FE-F133-7742-A25D-B4E90CE14420

  Device      Start      End      Sectors    Size Type
  >> Free space  2048    16777182  16775135    8G

[ New ] [ Quit ] [ Help ] [ Write ] [ Dump ]
  
```

Curso: GNU/Linux

Professor: Lucas Pereira da Rocha | E-mail: lucas.rocha50@fatec.sp.gov.br

Professor: Lucas Pereira da Rocha | **E-mail:** lucas.rocha50@fatec.sp.gov.br

nfs - Sistema de arquivos remoto NFS.

proc - Sistema de arquivos Linux Process Information.

jfs (Journaling File System) : criado pela IBM para uso em servidores corporativos, teve seu código liberado.

xfs : desenvolvido originalmente pela Silicon Graphics e posteriormente disponibilizado o código fonte.

swap - Em alguns lugares ele é mencionado como um Sistema de Arquivos, mas SWAP é um espaço reservado para troca de dados com a memória RAM.

Devices e UUID

“UUID - Universally Unique Identifier” é código de representação do hardware físico, o sistema operacional se refere a ele na montagem de uma participação.

Para descobrirmos o “UUID” de nossa partição podemos utilizar “blkid”:

```
# blkid
/dev/sda1: UUID="f541a97e-ef19-4e47-b305-b535a75c932a" TYPE="ext3" LABEL="MAIN"
/dev/sda3: UUID="7C444A56444A12F6" TYPE="ntfs" LABEL="WIN"
/dev/sda5: TYPE="swap"
/dev/sda6: UUID="69ff8ed5-c09b-49b6-b21d-328e90243efa" TYPE="ext3" LABEL="HOME"
/dev/sda7: UUID="2c070d34-5c6e-4504-8d4b-9a8fa910548d" TYPE="ext3" LABEL="STORAGE"
/dev/sda8: UUID="489B-5A22"
```

```
# blkid /dev/sdb1
```

Caso não obtenha resposta é porque a partição não tem um arquivo de sistema. Determine o ponto de montagem:

Se não existir um ponto de montagem, basta criar um diretório vazio no local apropriado, em geral no “/media” ou “/mnt” e executar o comando para montá-lo.

Montar disco de armazenamento no GNU/Linux

```
# mount -t <tipo> -o <opções> <dispositivo> <ponto-de-montagem>
# mount /dev/sdX /mnt/PASTA
# mount /dev/sda1 /mnt/PASTA_BACKUP ( Mountando a partição SDA1 na pasta PASTA_BACKUP dentro /mnt )
# mount /dev/sdb2 /mnt/BACKUP_CPD ( Mountando a partição SDB2 na pasta BACKUP_CPD dentro /mnt )
# mount /dev/sdb2 -t ntfs /mnt/CPD ( Montagem da partição especificando o tipo do sistema de arquivos )
) Opções para -t: -t vfat / -t ext2 / -t ext3
```

Manipulando permissões

O comando para manipular o dono do arquivo/pasta e grupo é CHOWN

```
# ls -l
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA ( Retorno do comando ls - l )
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA ( Retorno do comando ls - l dono do arquivo )
# drwx----- 1 root root 0 Oct 21 05:24 FATEC-PASTA ( Retorno do comando ls - l responsavel pelo grupo )
```

Trocar o dono do arquivo de root para lucas:

```
# chmod lucas:root FATEC-PASTA
```

```
# ls -l
```

```
# drwx----- 1 lucas root 0 Oct 21 05:24 FATEC-PASTA
```

(dono do arquivo é o usuário lucas)

Trocar o responsável do grupo de de root para lucas:

```
# chmod lucas:lucas FATEC-PASTA
```

```
# drwx----- 1 lucas lucas 0 Oct 21 05:24 FATEC-PASTA
```

(dono do arquivo é usuário lucas)

Configuração da rede

Conceitos de IP estático e dinâmico

IP estático é a configuração que o administrador fornece no sistema operacional para o computador participar de uma rede.

IP dinâmico o computador busca na rede as informações IP, máscara, gateway e dns para o computador participar de uma rede. Essa resposta com as informações podem vir por um servidor DHCP que pode estar em um servidor, roteador entre outros.

Configurando a Rede

A configuração da rede baseia-se em três etapas:

Configuração do número IP e sua máscara de rede;

Configuração do "Gateway";

Configuração dos servidores "DNS".

Configurando IP e Máscara

Além da interface "loopback", podemos configurar outras interfaces, basta que elas estejam presentes fisicamente e sejam suportadas pelo kernel. Para visualizar essas configurações em nossas interfaces de redes, utilizamos o comando "ifconfig".

Instale o pacote net-tools:

```
# aptitude install net-tools
```

Para atribuir um endereço IP para uma placa de rede utilizamos esta sintaxe:

```
# ifconfig <interface> <IP>
```

Exemplo:

Atribuir endereço IP 172.16.0.100 para a interface "eth0"

```
# ifconfig eth0 172.16.0.100
```

```
# ifconfig eth0 172.16.0.100 netmask 255.255.0.0
```

Visualize o endereço da interface:

```
# ifconfig eth0
```

Criar uma interface virtual:

```
# ifconfig eth0:0 10.0.0.1
```

Onde: :0 é ativada o segundo ip na placa de rede

```
# ifconfig eth0:0
```

Configurando o gateway

Gateway é a porta de entrada e saída da nossa rede. Pensando em host, configuramos o “gateway” para termos acesso a internet:

```
# route add default gw IP
```

Adicionando uma rota padrão:

```
# route add default gw 172.16.0.1      ( IP: 172.16.0.1 é o gateway da minha rede )
# route -n                             ( Verificando rotas disponiveis no meu GNU/Linux )
# route del default                     ( remover a rota padrão )
```

Configuração dos DNS Servers

Para não ter que memorizar todos os endereços IP que precisamos acessar, foi criado um serviço de rede chamado “DNS”. Este faz a tradução de nomes para endereços IP. Para configurar os servidores de “DNS” na máquina local, precisamos editar “resolv.conf” localizado em “/etc”.

```
# vim /etc/resolv.conf
nameserver 8.8.8.8
Nameserver IP_outroDNS
```

Teste a configuração com um ping para um site :

```
# ping www.fatec.sp.gov.br
```

Traceroute é uma ferramenta que permite descobrir o caminho feito pelos pacotes desde a sua origem até o seu destino.

```
# traceroute www.fatec.sp.gov.br
```

Configuração estática de rede

O arquivo de configuração da rede na distribuição Debian é “/etc/network/interfaces” e com as seguintes linha conseguiremos habilitar as configurações:

```
# vim /etc/network/interfaces
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static
address 192.168.0.10
netmask 255.255.255.0
broadcast 192.168.0.255
network 192.168.0.0
gateway 192.168.0.1
```

Reinicie o serviço:

```
# invoke-rc.d networking stop
```

```
# invoke-rc.d networking start
```

Arquivos de log

Log e sua importância

Logs são vitais para a segurança da informação, são armazenados em arquivos e suas informações podem ser utilizadas para tomar uma determinada ação. Logs registram informações do sistema operacional, tráfego de rede, autenticações e etc.

Arquivos de logs no GNU/Linux

<code>#/var/log/</code>	(Pasta de logs do sistema operacional)
<code>#/var/log/syslog</code>	(Arquivo principal de logs do sistema operacional)
<code>#/var/log/messages</code>	(Logs do sistema e kernel)
<code>#/var/log/auth</code>	(Logs de autenticação)

Obtendo informações do sistema através dos arquivos de Logs

Para utilizar de forma efetiva as informações, utilize `cat`, `less`, `more`, `head`, e `pipe` (|)

<code>#less /var/log/syslog</code>	(Paginando informações no arquivo syslog)
<code>#cat /var/log/syslog egrep -i "acesso restrito"</code>	(syslog sendo direcionado para uma pesquisa de conteúdo)
<code>#head -n 10 /var/log/syslog</code>	(filtro para mostrar apenas as 10 primeiras linhas do syslog)
<code>#tail -n 10 /var/log/syslog</code>	(filtro para mostrar apenas as 10 últimas linhas do syslog)
<code># tail -f /var/log/syslog</code>	(Verificando o log em tempo real)