

# Introdução ao Linux

- O GNU/Linux é um sistema operacional livre
  - Indepe de uma empresa
  - É mantido por comunidades ao redor do mundo
  - Código fonte aberto, permite modificações
  - É distribuído em licenças GPL.
- A Free Software Foundation (FSF) é a responsável pelo desenvolvimento do Projeto GNU.
  - Fundação criada por Richard Stallman,
- O Projeto GNU foi iniciado em 1984 para desenvolver um sistema operacional completo, compatível com o UNIX, e que fosse Software Livre (Hurd, 1990)



# *Introdução ao Linux*

- GNU é um acrônimo, significa:
  - “GNU's Not UNIX - GNU não é UNIX”
- O Linux foi idealizado por Linus Torvalds em 1991
- Implementou um microkernel baseado no Minix
  - Minix é um sistema operacional baseado no UNIX desenvolvido por Andrew Tanenbaum.



# *Características do Linux*

- É desenvolvido de forma voluntária por programadores ao redor do mundo.
- Multitarefa real, multiusuário, suporte a nomes extensos de arquivos e diretórios (255 caracteres) e proteção entre processos executados na RAM.
- Suporte a multiplataformas tais como: SPARC, ALPHA, PowerPC, ARM, Intel, sejam RISC ou CISC.
- Conectividade com outros sistemas operacionais como: Apple (MAC-OS), Sun (Solaris), ALPHA (True-64), e Windows.
- Convive sem conflito com outros sistemas operacionais na mesma máquina/disco.

# *Características do Linux*

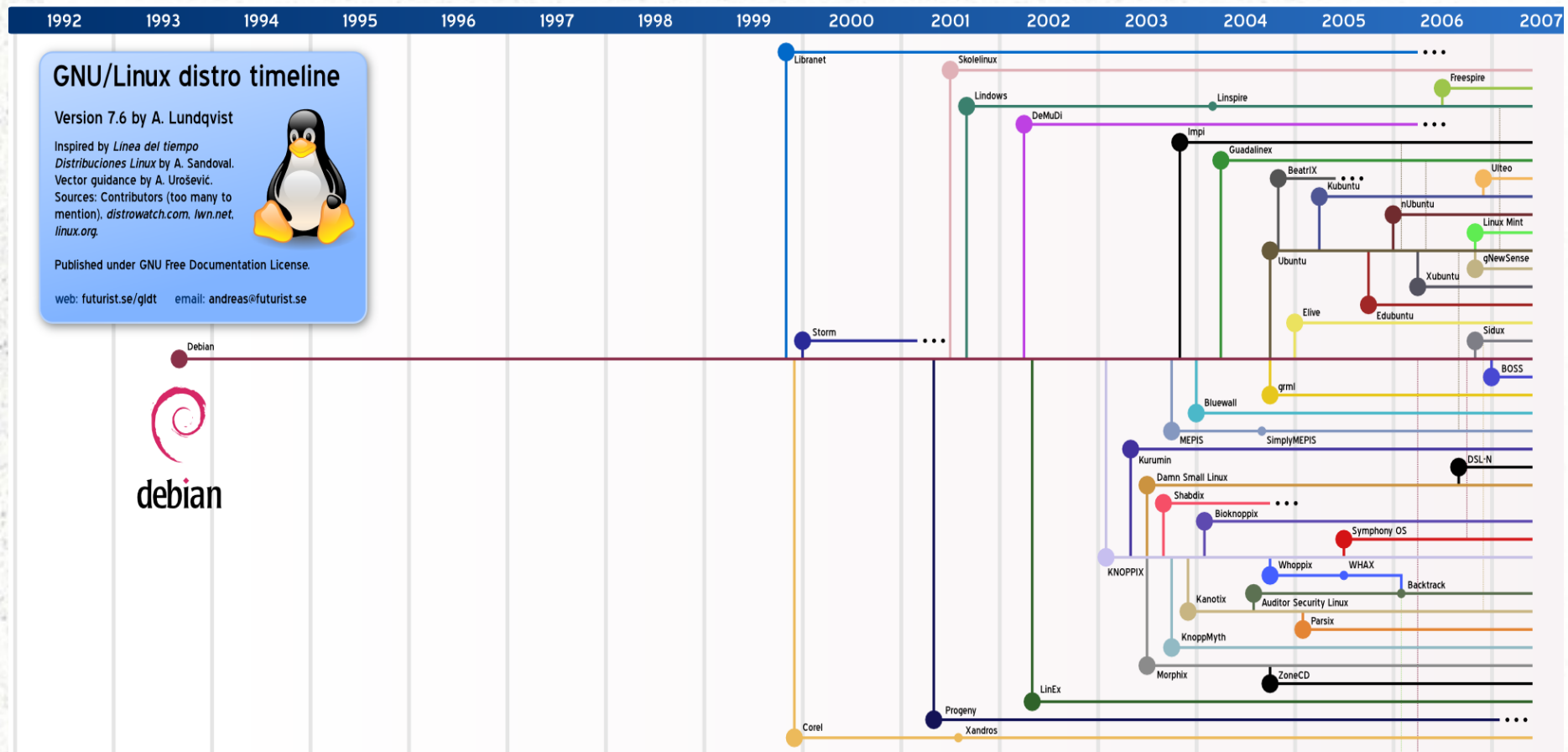
- Multitarefa: Permite que diversos programas rodem ao mesmo tempo.
- Processos: vários serviços disponibilizados pelo sistema que estão rodando em background.
- Daemon: processos podem ser reiniciados sem reiniciar a máquina. Exemplo: rede, X, firewall, etc.
- Suporte a terminais Virtuais (~ 63 consoles).
- Sistema de arquivos: Pode utilizar (acessar) vários sistemas de arquivos tais como: ext2, ext3, ext4, xfs, btrfs, fat, ntfs, nfs e cifs.
- Sistema de permissões hierarquizadas, para root (administrador) e usuários

# *Distribuições*

- Uma distribuição Linux é um sistema operacional Unix-like incluindo o kernel Linux e outros softwares de aplicação, formando um conjunto.
- Há basicamente duas formas de distribuições
  - As mantidas por organizações comerciais, como a Red-Hat, Ubuntu, SUSE, etc.
  - As mantidas por comunidades como Debian, Gentoo, Slackware, CentOS, etc.
- Distro como Red-Hat, Suse possuem projetos open source. Fedora e Opensuse

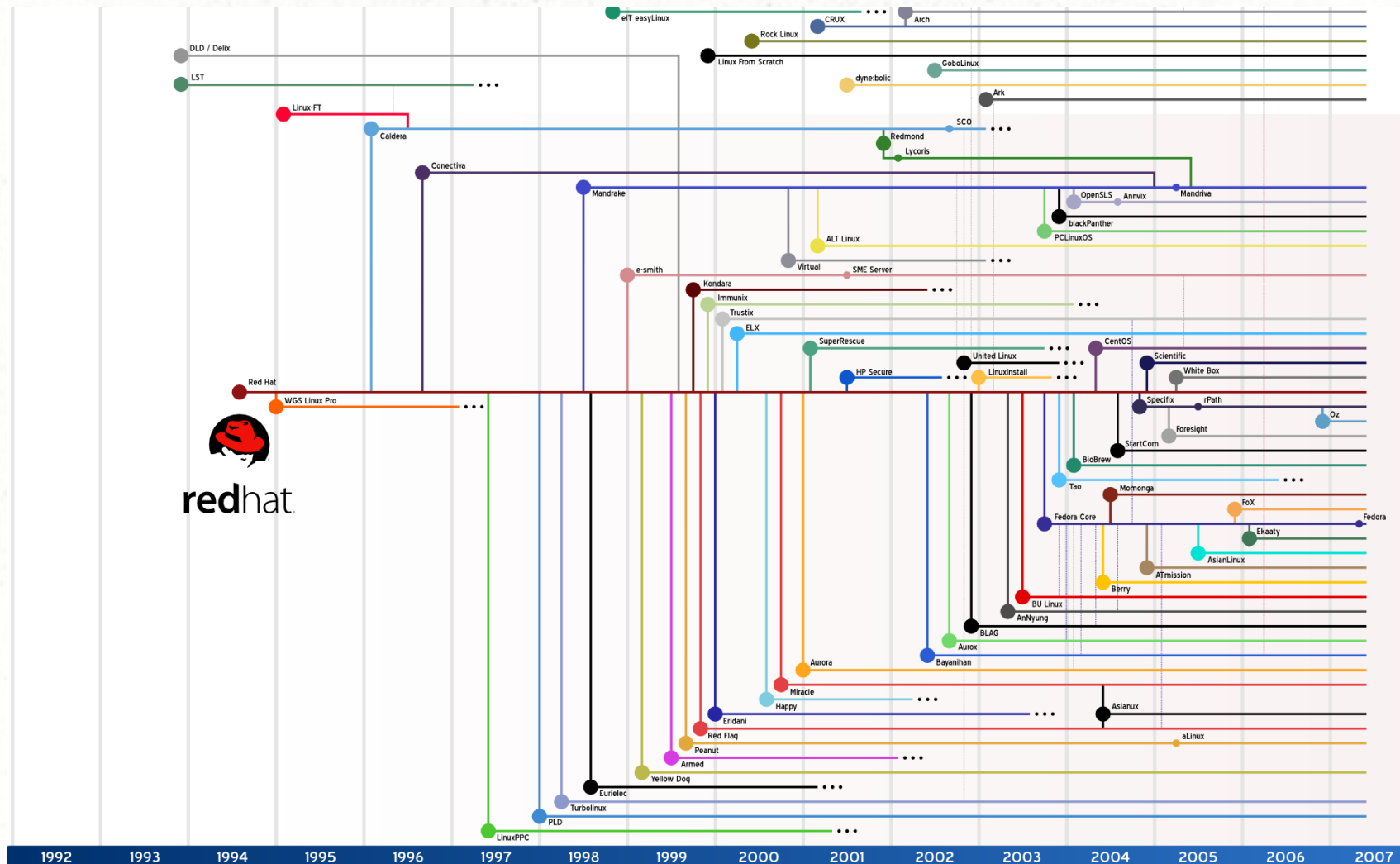
# Distribuições

- Há distribuições que servem de base para outras distribuições, chamadas de meta distribuições por exemplo Debian: Ubuntu, linuxmint



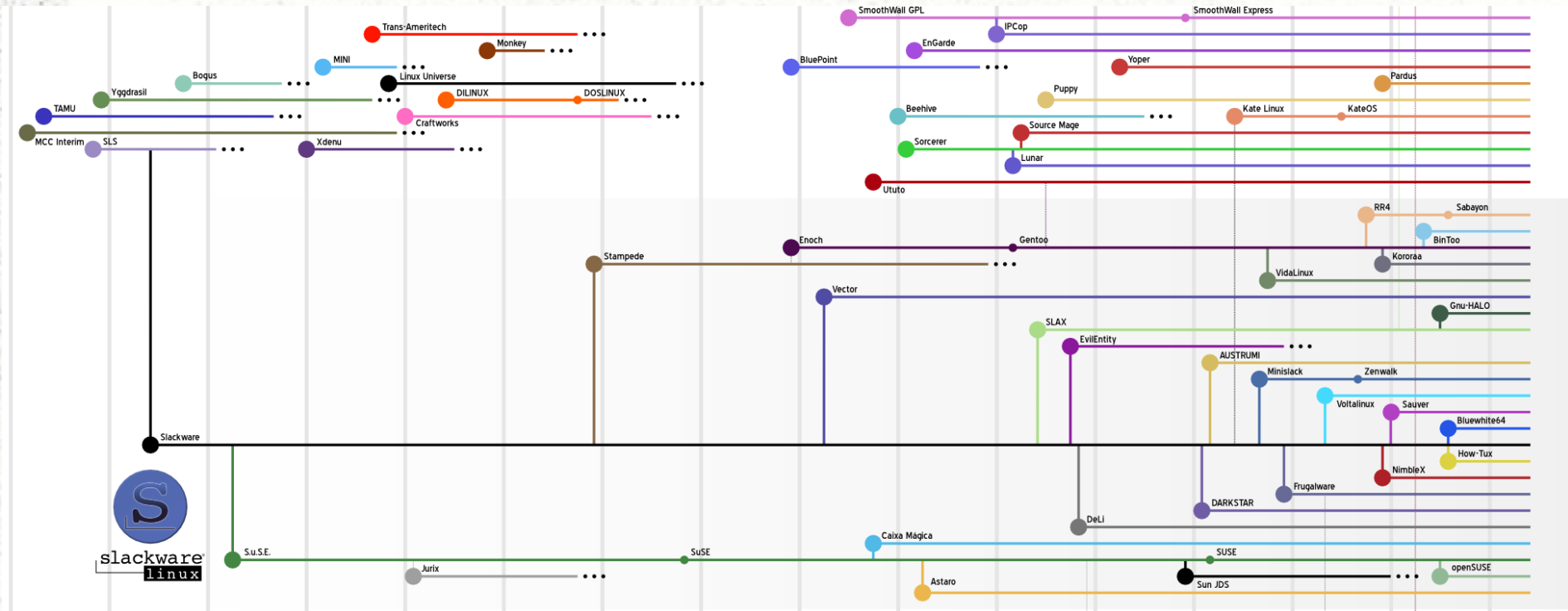
# Distribuições

- Red-Hat



# Distribuições

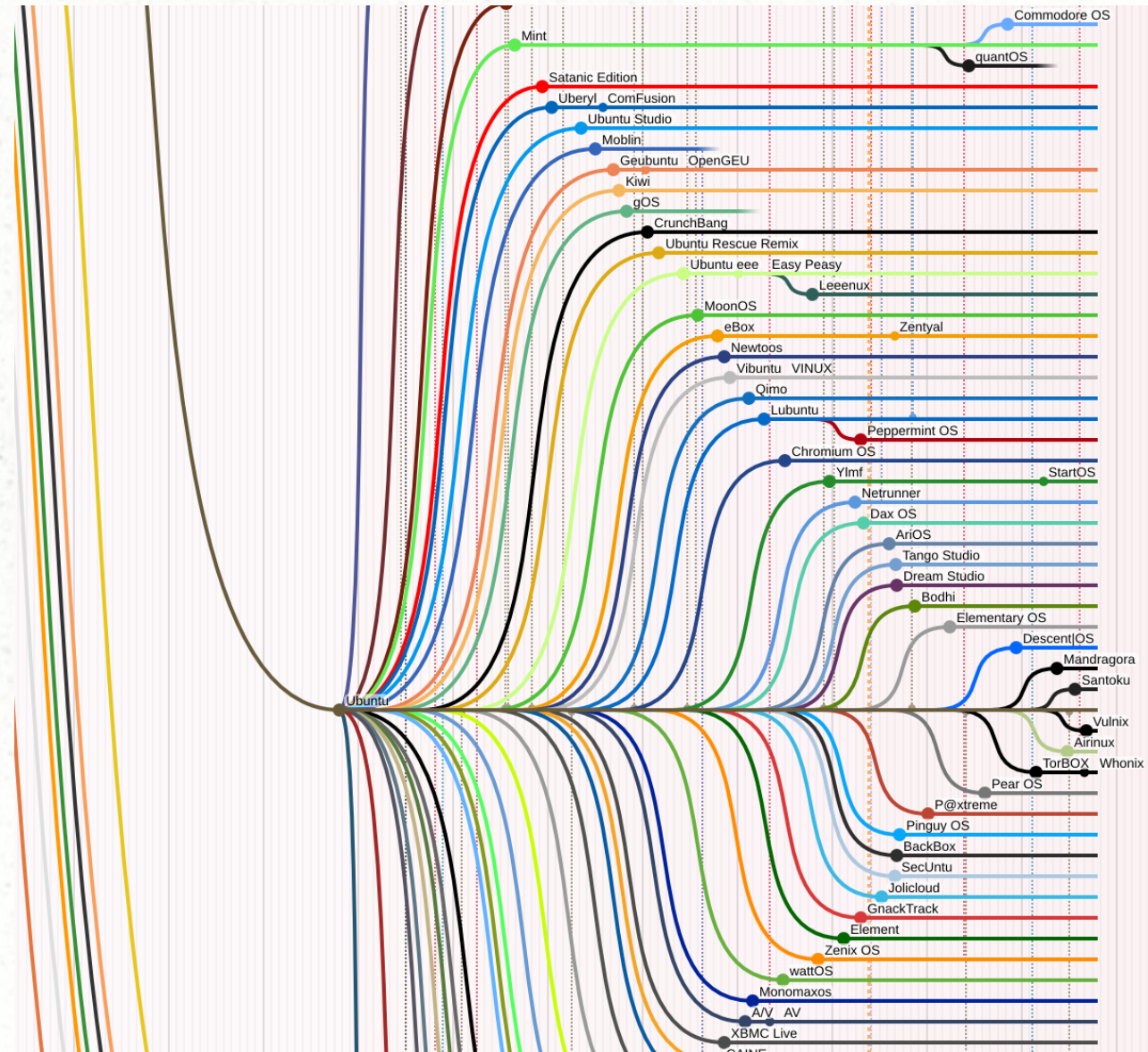
- Slackware





# Distribuições

- Ubuntu



# Distribuições

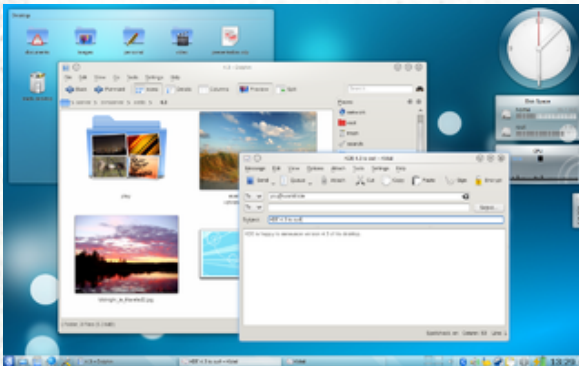
Logos das distribuições mais populares



# *Distribuições*

## Interfaces com o usuário, interface gráfica

- Kde, Gnome, Cinnamon, Enlightenment, Fluxbox, KDE Plasma, Lxde, Mate, Xfce, Unity.



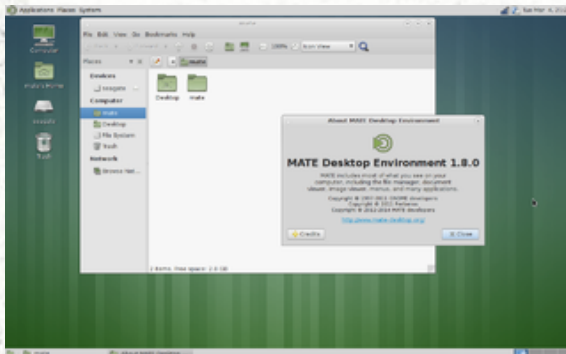
Kde Plasma



GNOME



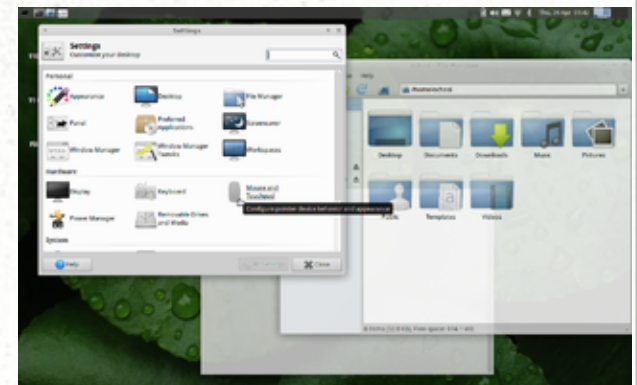
Unity



Mate



Cinnamon



Xfce

# *Distribuições*

## **Bibliografia/Referências**

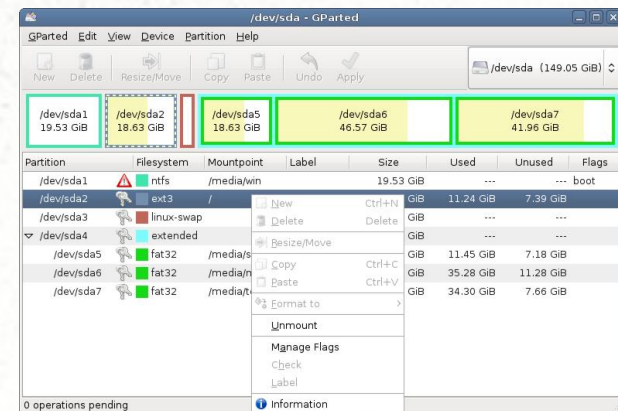
- <http://www.gnu.org/>
- <http://distrowatch.com/>
- <http://futurist.se/gldt/>
- <https://www.vivaolinux.com.br/linux/>
- <https://en.wikipedia.org/wiki/Linux>
-

# Partições no linux

- O Linux precisa de pelo menos duas partições
  - Uma partição para a raiz “/”, é a partir da raiz que será instalado o sistema
  - Outra partição para a “*swap*”, arquivo de troca.
- Há várias opções de sistemas de arquivos, sendo o ext4 o mais utilizado atualmente. Para a “*swap*” é utilizado um sistema de arquivo especial.
- As partições são numeradas.
  - Nos sistemas com MBR (32bits, < 2TB) somente 4 partições primárias (1 a 4), ou 3 primárias e as outras estendidas (63 IDE, 15 SCSI)
  - Nos sistemas com GPT (64bits,UFEI > 2TB) até 128 partições

```
mauricio@Floyd-Desktop ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
/dev/sda5          908206080 923828223 15622144   7,5G 82 Linux swap
/dev/sda6          923830272 976771071 52940800   25,3G 83 Linux
mauricio@Floyd-Desktop ~ $
mauricio@Floyd-Desktop ~ $ sudo fdisk -l /dev/sda
Disk /dev/sda: 465,8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00086769

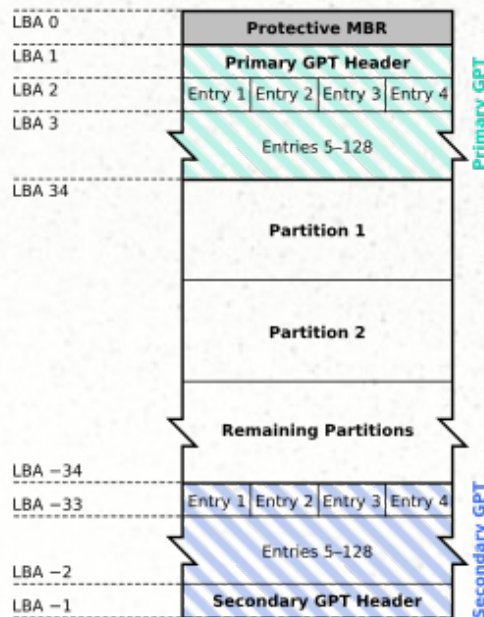
Dispositivo Inicializar   Start      Fim        Setores   Size Id Tipo
/dev/sda1 *              2048      68359469   68357422   32,6G 83 Linux
/dev/sda2                68360192 126953471 58593280   28G 83 Linux
/dev/sda3               126953472 908204031 781250560 372,5G 83 Linux
/dev/sda4               908206078 976771071 68564994   32,7G 5 Estendida
/dev/sda5               908206080 923828223 15622144   7,5G 82 Linux swap
/dev/sda6               923830272 976771071 52940800   25,3G 83 Linux
```



# Partições no linux

- Diferenças entre os sistemas de partições MBR e GPT
  - MBR (32bits, < 2TB) 4 partições primárias (1 a 4) as outras estendidas
  - GPT (64bits,UFEI > 2TB) até 128 partições
  - [https://pt.wikipedia.org/wiki/Tabela\\_de\\_Partici%C3%A7%C3%A3o\\_GUID](https://pt.wikipedia.org/wiki/Tabela_de_Partici%C3%A7%C3%A3o_GUID)
  - [http://www.wikiwand.com/en/BIOS\\_boot\\_partition](http://www.wikiwand.com/en/BIOS_boot_partition)

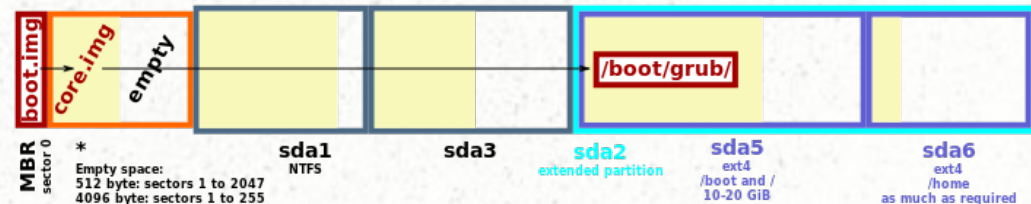
## GUID Partition Table Scheme



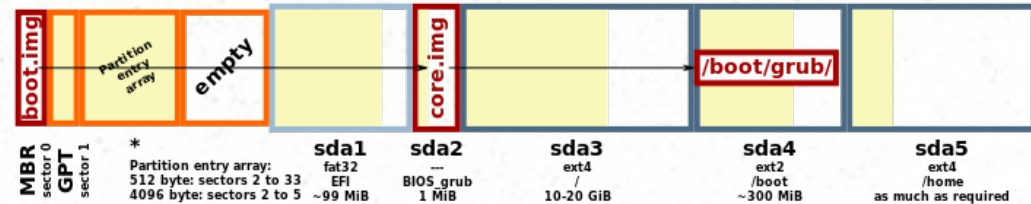
## GNU GRUB 2

Locations of *boot.img*, *core.img* and the */boot/grub/* directory

Example 1: An MBR-partitioned hard disk with sector size of 512 or 4096 bytes



Example 2: A GPT-partitioned hard disk with sector size of 512 or 4096 bytes



# Sistemas de arquivos no Linux

- Sistema de arquivos: diretório raiz ou “/”, no shell

```
mauricio@Floyd-Desktop ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
RX bytes:78227861 (78.2 MB) TX bytes:4294422 (4.2 MB)  
IRQ:20 Memória:fb200000-fb220000  
  
lo  
Link encap:Loopback Local  
inet end.: 127.0.0.1 Masc:255.0.0.0  
endereço inet6: ::1/128 Escopo:Máquina  
UP LOOPBACK RUNNING MTU:65536 Métrica:1  
pacotes RX:1777 erros:0 descartados:0 excesso:0 quadro:0  
Pacotes TX:1777 erros:0 descartados:0 excesso:0 portadora:0  
colisões:0 txqueuelen:1  
RX bytes:322102 (322.1 KB) TX bytes:322102 (322.1 KB)  
  
mauricio@Floyd-Desktop ~ $ ls -al /  
total 112  
drwxr-xr-x 23 root root 4096 Jun 27 10:26 .  
drwxr-xr-x 23 root root 4096 Jun 27 10:26 ..  
drwxr-xr-x 2 root root 12288 Jul 31 09:06 bin  
drwxr-xr-x 3 root root 4096 Jul 31 09:07 boot  
drwxr-xr-x 2 root root 4096 Jun 27 10:24 cdrom  
drwxr-xr-x 20 root root 4840 Ago 30 09:04 dev  
drwxr-xr-x 154 root root 12288 Ago 30 10:49 etc  
drwxr-xr-x 4 root root 4096 Abr 12 2016 home  
lrwxrwxrwx 1 root root 32 Jun 27 10:26 initrd.img -> boot/initrd.img-4.4.0-53-generic  
drwxr-xr-x 25 root root 4096 Jun 27 11:08 lib  
drwxr-xr-x 2 root root 4096 Jun 27 11:06 lib64  
drwx----- 2 root root 16384 Jun 27 10:15 lost+found  
drwxr-xr-x 3 root root 4096 Jul 3 17:13 media  
drwxr-xr-x 3 root root 4096 Jun 27 17:56 mnt  
drwxr-xr-x 2 root root 4096 Dez 13 2016 opt  
dr-xr-xr-x 212 root root 0 Ago 30 09:04 proc  
drwx----- 7 root root 4096 Jun 27 11:25 root  
drwxr-xr-x 35 root root 1120 Ago 30 10:51 run  
drwxr-xr-x 2 root root 12288 Jul 31 09:06/sbin  
drwxr-xr-x 2 root root 4096 Dez 13 2016 srv  
dr-xr-xr-x 13 root root 0 Ago 30 09:04 sys  
drwxrwxrwt 19 root root 4096 Ago 30 11:30 tmp  
drwxr-xr-x 10 root root 4096 Dez 13 2016 usr  
drwxr-xr-x 11 root root 4096 Dez 13 2016 var  
lrwxrwxrwx 1 root root 29 Jun 27 10:26 vmlinuz -> boot/vmlinuz-4.4.0-53-generic  
mauricio@Floyd-Desktop ~ $
```

# *Sistemas de arquivos*

- **/bin**: armazena os binários (bin de binário, executáveis) de alguns comandos básicos do sistema, como:
  - su, tar, cat, rm, pwd, etc. (~16MB ).
- **/boot**: armazena os arquivos de inicialização
  - O Kernel ou os Kernels, é possível ter mais de um.
  - O gerenciador de boot do sistema (Grub).
- **/dev**: Diretório que contém os dispositivos, é o exemplo mais exótico (diferente) da estrutura de diretório no Linux (UNIX).
  - Todos os arquivos contidos aqui, como: /dev/hda , /dev/sda, /dev/dsp , /dev/mouse, etc., não são arquivos, mas ponteiros para dispositivos de hardware.



# *Sistemas de arquivos*

- **/usr:** de Unix System Resources. Este é o diretório com mais arquivos em qualquer distribuição Linux.
  - É neste diretório que ficam os binários e bibliotecas de todos os principais programas.
  - A pasta “/usr/bin” (bin de binário), armazena cerca de 2.000 programas e atalhos para programas numa instalação típica.
  - A pasta “/usr/lib” (lib de library), armazena as bibliotecas
  - /usr/man: Estão os manuais (help) dos comandos.
  - /usr/share: Informações adicionais de programas (comandos) e exemplos.

# *Sistemas de arquivos*

- **/etc:** Aqui estão os arquivos de configuração do sistema.
  - Os arquivos são texto puro (ASCII). E normalmente terminam em “.conf”, por exemplo: adduser.conf , kernel-img.conf ,etc.
  - O arquivo “passwd” , contém as informações de cada usuário do sistema desde o root a usuários especiais.
  - Há uma estrutura de subdiretórios para cada aplicação
- **/home:** Contém os diretórios de todos os usuários menos do **root**.
- **/lib:** Aqui estão as lib (bibliotecas) compartilhadas necessárias aos programas no sistema raiz.

# *Sistemas de arquivos*

- **/root**: Contém os diretórios do superusuário (administrador) do sistema.
  - Em algumas distribuições é utilizado um usuário (primeiro) como administrador via comando *sudo*.
- **/tmp**: Pasta para arquivos temporários por exemplo do Kde e Gnome (gerenciadores de janelas)
- **/mnt**: Utilizado para realizar a montagem de outro tipo de sistema de arquivos
- **/media/**: montagem automática de dispositivos, como CDs e pendrive ou Hds externos

# *Sistemas de arquivos*

- **/proc**: É um sistema de arquivos especial que é criado na memória pelo kernel. Apresenta informações do sistema e dispositivos.
  - Por exemplo, informações da CPU estão no arquivo “/proc/cpuinfo” e da memória em “/proc/meminfo”
- **/var**: este diretório contém os arquivos que são alterados constantemente
  - /var/log: aqui estão os arquivos de log (monitoração) da máquina:
  - /var/log/wtmp, histórico dos logins dos usuários
  - /var/log/messages, contém as mensagens do kernel e de vários programas
  - /var/log/syslog, mensagens de vários programas

# Comandos básicos no linux

- No terminal ou console há o que chamamos de interpretador de comandos, ou shell.
- BASH Shell (bash): Bourne-Again Shell, é uma extensão do sh, sendo utilizado como padrão nos linux.
- Nos prompts do Linux, quando acaba com o símbolo \$, indica que o login é um usuário normal.
  - *mauricio@deep:~\$* , Este é o shell do usuário “mauricio” na máquina “deep” e no diretório “/home/mauricio”.
  - *mauricio@deep:/etc\$* , o usuário “mauricio” no diretório “/etc”.
- Se estiver logado como root, ao invés do símbolo \$, no final do prompt existirá o símbolo #.
  - *root@deep:/etc#*

# Comandos básicos no linux

- A tecla “**Tab**” no bash autocompleta, muito útil para relembrar de um determinado comando.
  - mauricio@deep:/etc\$ k
  - Display all 268 possibilities? (y or n)
- Os comandos no bash são executados da forma
  - “**Comando** <opção> <argumento>”
- “**cd**”: Este comando serve para mudar (navegar) de diretório
  - “cd /” : muda de onde estiver para o diretório raiz
  - “cd /etc”: muda de onde estiver para o diretório “/etc”
  - “cd ..” : sobe um diretório de onde você estiver.

# Comandos básicos no linux

- **“ls”**: Serve para listar os arquivos e diretórios dentro da pasta atual.
  - Por exemplo: \$ “ls -ahl”
- **“cp”**: Este é o comando usado para copiar arquivos de uma pasta a outra.
  - “cp [opções] <arquivo\_origem> <arquivo\_destino>”
  - -i : Modo interativo. Pergunta se você quer sobrescrever ou não (confirmações)
  - -v : Mostra o que está sendo copiado.
  - -R : Copia recursivamente (diretórios e subdiretórios).

# Comandos básicos no linux

- **“mv”**: O mv serve para mover arquivos de um lugar para o outro.
  - “mv <arquivo\_origem> <arquivo\_destino>”
  - -i : Modo interativo. Pergunta se você quer sobrescrever ou não (confirmações)
  - -v : Mostra o que está sendo movido.
  - -f : Copia sem solicitar confirmação.
- **“rm”**: O rm serve para remover tanto arquivos quanto diretórios, de acordo com os parâmetros usados.
  - “rm [opções] <arquivo>”
  - Para remover um arquivo simples
  - o parâmetro "-f", Para que ele não pedir confirmação
  - “rm -rf <arquivos/diretorio>



# Comandos básicos no linux

- **“mkdir”**: Este serve para criar novos diretórios,
  - "mkdir ~/teste "
  - É possível também criar pastas recursivamente, criando-se todas as pastas necessárias até chegar a que você pediu, adicionando o parâmetro "-p"
  - "mkdir -p ~/tmp/rasp/teste
- **“locate”**: Este é um comando muito útil, ele permite encontrar arquivos de forma muito rápida.
  - Ele procura apenas dentro de uma base de dados que contém os nomes de todos os arquivos.
  - Esta base é gerada ao rodar o comando “updatedb”,
  - É necessário ser root para executar o “updatedb”.

# Comandos básicos no linux

- “**su**”: No Linux, existe uma separação clara entre o root e os demais usuários do sistema.
  - O root é o único que pode alterar a configuração do sistema e usar a maior parte das ferramentas de configuração.
  - Exemplo: “\$ su user2”, será solicitada a senha do user2.
- Se você está logado como usuário qualquer e der o comando “su” sem nome de usuário, será solicitada a senha do root e, quando ela for fornecida, passará a ser super-usuário (root).
- “**sudo**”: Este comando é utilizado em algumas distro baseadas no Ubuntu para permitir acesso à administração do sistema. Deve preceder o comando que se deseja executar com “root”
  - Exemplo: “\$ sudo cat /etc/passwd”

# *Comandos básicos no linux*

- “**cat**”: Serve para ver o conteúdo de um arquivo. Por exemplo,
  - "cat /etc/passwd" mostra o conteúdo arquivo "passwd" na tela do console.
  - O problema do cat é que ele mostra o conteúdo do arquivo todo na tela, não permite navegação.
- Ele permite concatenar arquivos, por exemplo.
  - “cat /etc/passwd /etc/syslog.conf > /tmp/teste\_cat.txt”
  - Para ver o resultado “cat /tmp/teste\_cat.txt”

# *Comandos básicos no linux*

- **“clear”**: Limpa a tela, uma forma de colocar ordem na casa antes de executar novos comandos. Ao invés de digitar, você pode pressionar “Ctrl+L”, que é o atalho de teclado para ele.
- **“du”**: O du permite ver uma lista com o espaço ocupado por cada pasta dentro do diretório atual.
  - É uma forma rápida de encontrar grandes arquivos ou pastas que estão consumindo muito espaço.
  - Em geral utilizo “du -h”, onde o -h (human)
  - Exemplo: “\$du -sh ./\*”, mostra os tamanhos de cada pasta a partir da atual (.).

# Comandos básicos no linux

- “**more**”: Este comando permite que o usuário visualize uma tela para frente por vez, em um longo corpo de texto, assim como pesquisar esse texto.
  - Pressionar a barra de espaço faz pular para frente uma página, enquanto pressionar Enter moverá para frente uma linha por vez.
- “**grep**”: O grep permite filtrar a saída de um determinado comando, de forma que ao invés de um monte de linhas, você veja apenas a informação que está procurando.
  - Ele é frequentemente usado em conjunto com o pipe, sobretudo em scripts.

# Comandos básicos no linux

- | **(pipe)**: Junto com as setas de redirecionamento (> e >>), o pipe (|) é muito usado em scripts e comandos diversos.
  - Ele permite fazer com que a saída de um comando seja direcionada (enviada) para outro ao invés de ser mostrada na tela.
  - Parece muito exótica, mas acaba sendo incrivelmente útil, pois permite “combinar” diversos comandos que originalmente não teriam nenhuma relação entre si, de forma que eles façam alguma coisa específica.
- Por exemplo, para o comando “cat” quando a tela ficou cheia
  - “cat /etc/passwd |more”
  - “\$ sudo cat /var/log/syslog|grep error |more”

# *Comandos básicos no linux*

- **“history”** : mostra o histórico, pode ser utilizado junto com o grep para filtrar, procurar um determinado comando
- **&**: Este é um parâmetro que permite rodar aplicativos mantendo o terminal livre. No Linux, todos os aplicativos, mesmo os gráficos, podem ser chamados a partir de uma janela de terminal.
  - O problema é que, ao chamar algum aplicativo, o terminal ficará bloqueado até que o aplicativo seja finalizado, obrigando-lhe a abrir um para cada programa.
  - “~\$ edit teste.txt &”: roda o edit com o arquivo test.txt e libera o terminal.

# Comandos básicos no linux

- “**tar**”: era usado originalmente para backups em fitas magnéticas.
  - “\$ tar cvf <nome\_arquivo>.tar <arq1> <arq2>”.
  - Esse comando criará um novo arquivo, especificado pelo nome de arquivo *nome\_arquivo.tar* contendo os arquivos *arq1* e *arq2*.
- Outra opções:
  - “\$ tar czvf teste.tar.gz teste/” :(inclui todos os arquivos do diretório e subdiretórios de teste/ compactando-os automaticamente)
  - “\$ tar tzvf teste.tar.gz” : mostra o conteúdo de nosso arquivo teste.tar.gz
  - “\$ tar xzvf teste.tar.gz” :extraí o conteúdo do arquivo teste.tar.gz no diretório atual



# *Permissões no linux*

- No GNU/Linux, cada arquivo tem uma permissão para três categorias diferentes.
  - “own” (dono), “group” (grupo) e “others” (outros)
- As permissões são atributos dos arquivos/diretório que especificarão se ele pode ser lido, executado ou escrito. Assim cada usuário possui permissões de leitura, escrita e execução.
  - Indicadas por: “r”, “w” e “x” , respectivamente.
- Há indicação nas permissões também para diretório “d”.
- As permissões são controladas pelo comando “chmod”
  - O dono: é a pessoa que criou o arquivo ou o diretório
  - O grupo: permitir que vários usuários diferentes tenham acesso a um mesmo arquivo
  - Todos: é a categoria de usuários que não são donos ou não pertencem ao grupo do arquivo
- O comando “chown” determina quem é o dono e o grupo do arquivo ou diretório.

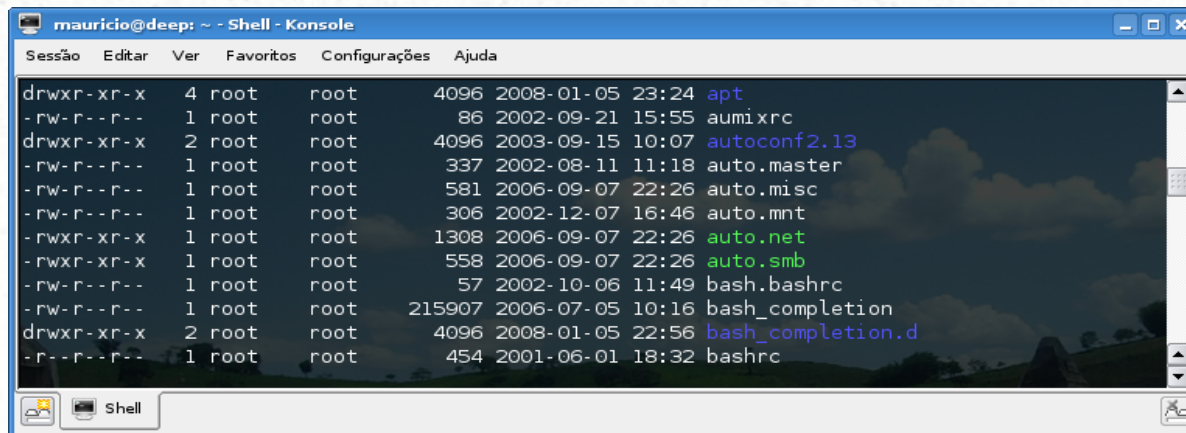
# Permissões no linux

- O chmod atua sobre os três níveis; usuário 'u', grupo 'g' ou outros 'o'. Utiliza letras ou modo octal.
  - No modo octal usa números de 0 (zero) a 7 (sete).

<b>Valor</b>	<b>Arquivo</b>	<b>Diretório</b>
0	permissão negada	permissão negada
1	permissão de execução	permissão para entrar
2	permissão de gravação	permissão para gravar
3	permissão de gravação e execução	permissão de entrar e gravar
4	permissão de leitura	permissão para listar o conteúdo
5	permissão de leitura e execução	permissão de listar e entrar
6	permissão de leitura e gravação	permissão de listar e gravar
7	soma de todas as permissões	soma de todas as permissões

# Permissões no linux

- No modo não octal, são utilizadas letras e os símbolos – e + para remover ou acrescentar uma permissão.
  - Ex. `$ chmod [ugoa]{-+}[rwx] nome_do_arquivo_ou_diretório`
  - `$ chmod u+w teste.txt` : adiciona perm. de escrita “w” para o usuário “u”
  - `$ chmod ug+rw teste.txt` : adiciona perm. de leitura e escrita para o usuário e para o grupo.
  - `$ chmod 755 teste.txt`: altera as permissões para **rw** dono, **rx** grupo e para os outros
  - `$ chmod 644 teste.txt`: altera as permissões para **rw** dono, **r** grupo e para os outros



```
maurício@deep: ~ - Shell - Konsole
Sessão  Editar  Ver  Favoritos  Configurações  Ajuda
drwxr-xr-x  4 root  root    4096 2008-01-05 23:24 apt
-rw-r--r--  1 root  root     86 2002-09-21 15:55 aumixrc
drwxr-xr-x  2 root  root    4096 2003-09-15 10:07 autoconf2.13
-rw-r--r--  1 root  root     337 2002-08-11 11:18 auto.master
-rw-r--r--  1 root  root     581 2006-09-07 22:26 auto.misc
-rw-r--r--  1 root  root     306 2002-12-07 16:46 auto.mnt
-rwxr-xr-x  1 root  root    1308 2006-09-07 22:26 auto.net
-rwxr-xr-x  1 root  root     558 2006-09-07 22:26 auto.smb
-rw-r--r--  1 root  root     57 2002-10-06 11:49 bash.bashrc
-rw-r--r--  1 root  root  215907 2006-07-05 10:16 bash_completion
drwxr-xr-x  2 root  root    4096 2008-01-05 22:56 bash_completion.d
-r--r--r--  1 root  root     454 2001-06-01 18:32 bashrc
```

# *Instalação do Linux*

- Antes de instalar é necessário saber
  - a) A máquina compartilhará outro sistema operacional (SO)
  - b) A utilização será Servidor ou Desktop
  - c) Compatibilidade de Hardware
- Se for compartilhada com outro SO (mais de dois).
  - O compartilhamento é no mesmo disco
  - O compartilhamento é em discos separados.
- Definir o particionamento do disco
- A nomenclatura que o Linux utiliza para os discos é:
  - sda: para o primeiro
  - sdb: para o segundo, etc
- Nos discos IDE(antigos) era utilizando hda, hdb, hdc e hdd