



# Linux para Infraestrutura

Prof. André Nasserála  
nasserála@gmail.com

# Calculadora BC

- Cálculos em linha de comando, digite: # bc
- Depois digite a operação desejada e enter, exemplo:
- $25+3+8$  : 36
- Pra encerrar e sair do "bc" digite quit;
- Exemplos:
- echo "25+3+8" | bc
- echo "25/3+8" | bc
- echo "scale=2; 20/3" | bc
- echo "scale=4; 20/3" | bc

# Comando PS

- Este comando exibe informações sobre os processos que estão executando na máquina.
- Algumas opções do comando:
  - -a : mostra os processos de todos os usuários.
  - -A ou -e : mostra todos os processo.
  - -f : mostra a árvore de execução de comandos.
  - -g grupo : mostra os processos de um determinado nome de grupo.
  - -x : mostra os processos que não foram iniciados no console.
  - -u : fornece o nome do usuário e a hora de início do processo.

# Comando PS

- # ps
- Exibe os processos sendo executados no terminal.
  
- # ps -aux
- Exibe todos os processos do sistema, independente de terminal.
  
- # ps -g adm
- Exibe os processos do grupo *adm*.

# Comando kill

- Este comando envia um sinal para um processo.
- Algumas opções do comando
- -l : lista os nomes dos sinais.
- -l número : fornece o nome do sinal identificado pelo número fornecido.
- -s sinal : especifica o sinal a ser enviado, onde sinal pode ser um nome ou um número.

# Comando kill

- Neste exemplo temos dois processos sendo executados: bash(841) e teste(1314).
- Para finalizar o processo teste, basta digitar
- kill 1314
- Caso o processo não seja encerrado, você pode forçar o término do processo com o seguinte comando
- kill -9 1314

# Pacotes RPM

- O RPM é um poderoso gerenciador de pacotes que permite ao administrador instalar, remover e obter informações sobre pacotes.
- Com o RPM é possível também reparar um banco de dados danificado, construir pacotes a partir de arquivos fonte, verificar a assinatura digital de pacotes RPM, simular uma instalação, entre outras coisas.
- O RPM oferece uma grande gama de funcionalidades, no entanto, serão mostrados apenas as mais utilizadas, devendo o administrador consultar a documentação do aplicativo para obter mais detalhes.

# Pacotes RPM

- Instalando e Atualizando Pacotes RPM
- Instalar pacotes utilizando o comando rpm é bastante simples. Abra um terminal e, estando no mesmo diretório onde está o pacote que se deseja instalar, digite:
- **# rpm -i pacote-versao.i386.rpm**
- A opção -i informa ao comando que você deseja instalar um pacote.
- É recomendado que o administrador utilize também as opções v (verbose - modo detalhado) e h (inclui linhas de progresso) ao instalar um pacote, ou seja, rpm -ivh mostra mais informações sobre o andamento do processo.

# Pacotes RPM

- Caso o pacote a ser instalado necessite que outro pacote tenha sido previamente instalado, o RPM apresentará uma mensagem de erro mostrando quais dependências não foram atendidas para a instalação deste pacote.
- Será necessário então instalar os pacotes indicados para poder então instalar com sucesso o pacote que originalmente se queria instalar.
- Será mostrado mais adiante um modo de fazer isso automaticamente.
- Caso se deseje atualizar um pacote já instalado no sistema por uma versão mais nova em vez do -i utilize a opção -U, assim o pacote mais antigo será removido, o pacote novo será instalado e as configurações serão mantidas.

# Pacotes RPM

- Removendo Pacotes RPM
- Para remover um pacote com o rpm abra um terminal e digite:
- **# rpm -e nome\_do\_pacote**
- Ao remover um pacote não é necessário utilizar o nome do pacote completo, isto é, não utilize nome-versão-release.rpm, mas sim apenas o nome do pacote. Utilize a versão apenas se existirem duas versões do pacote instalado e se desejar remover uma delas.
- Caso algum pacote instalado no sistema dependa do pacote que se deseja remover, o RPM não fará a desinstalação e emitirá uma mensagem de erro informando que dependências seriam quebradas com isto. Caso seja realmente necessário, o administrador poderá utilizar a opção --nodeps para evitar que o RPM faça essa verificação, mas isso não é recomendado, pois poderá danificar o sistema.

# Pacotes RPM

- Obtendo Mais Informações Sobre os Pacotes
- O RPM pode ser utilizado para obter mais informações sobre os pacotes, tanto os já instalados quanto os não instalados. A forma básica para o modo de consulta é `rpm -q[opção] pacote`. Será mostrado a seguir, algumas das consultas mais comuns utilizando-se o RPM:
- Obtendo informações de um pacote:

```
$ rpm -qi bash
Name       : bash           Relocations: (not relocatable)
Version    : 2.05b           Vendor: Conectiva
Release    : 42313cl       Architecture: i386
Group      : Base         License: GPL
Size       : 772573
Install Date: Qui 04 Mar 2004 11:26:13 BRT
Build Date  : Sex 26 Dez 2003 01:52:22 BRT
Build Host  : mapi8.distro.conectiva
Source RPM  : bash-2.05b-42313cl.src.rpm
Signature   : (none)
Packager    : Conectiva S.A. <security at conectiva.com.br>
URL         : http://www.gnu.org/software/bash
Summary     : GNU Bourne Again Shell (bash)
Description :
Bash é um interpretador de comandos compatível com sh,
que executa comandos lidos da entrada padrão ou de um arquivo.
Bash também incorpora características úteis das shells Korn e
C (ksh e csh). O Bash tem sido desenvolvido para ser uma
implementação compatível com a especificação IEEE Posix para
shells e ferramentas (IEEE Working Group 1003.2).
```

# Pacotes RPM

- Para obter informações de um pacote que não está instalado, utilize a opção `-p`, seguida do nome do arquivo do pacote.
- O administrador poderá utilizar o `rpm` para obter informações sobre as dependências de um pacote. Poderá descobrir que outros pacotes dependem do pacote que se quer consultar, bem como descobrir de que pacotes o pacote consultado necessita. Os dois exemplos abaixo mostram como descobrir essas duas informações, respectivamente:

```
# rpm -q --whatrequires glib
gtk+-1.2.10-45456cl
xmms-1.2.10-52293cl
bonobo-1.0.22-46388cl

# $ rpm -q --requires glib
/sbin/ldconfig
/sbin/ldconfig
libc.so.6
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1.2)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.3)
libdl.so.2
libdl.so.2(GLIBC_2.0)
libdl.so.2(GLIBC_2.1)
libpthread.so.0
libpthread.so.0(GLIBC_2.0)
libpthread.so.0(GLIBC_2.3.2)
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rpmlib(ScriptletInterpreterArgs) <= 4.0.3-1
```

# Pacotes RPM

- Para descobrir a qual pacote pertence um arquivo do sistema utilize a opção `-qf` arquivo, como no exemplo abaixo:
- **# rpm -qf /bin/bash**
- **bash-2.05b-42313cl**
- E como um último exemplo, veja como listar todos os pacotes instalados no sistema:
- **# rpm -qa**
- Para procurar por um pacote específico em todos os pacotes instalados em seu sistema, você pode utilizar o comando anterior, juntamente com o `grep`. Se você quiser, por exemplo, encontrar todos os pacotes do XFree instalados em seu sistema, pode fazer o seguinte:
- **# rpm -qa | grep xfree**

# Pacotes RPM

- **Recuperando o Banco de Dados RPM**
- Se por um motivo ou outro o banco de dados de pacotes corromper-se, o rpm pode recuperá-lo. Caso seja necessário fazer isso, basta utilizar o comando abaixo:
- **# rpm --rebuilddb**
- **Utilizando Pacotes de Fontes**
- Usualmente, além dos pacotes que contém os binários dos aplicativos, encontram-se também pacotes contendo os arquivos fontes dos aplicativos. O pacote fonte, ao ser instalado, copia seus arquivos para a o diretório /usr/src/rpm, permitindo assim que os usuários do sistema possam estudar como determinado programa é feito e até mesmo alterá-lo de acordo com sua vontade.

# Pacotes RPM

- É possível utilizar um pacote fonte para construir um pacote RPM contendo os arquivos binários adequados à arquitetura da máquina em que ele será utilizado.
- Se você possui o arquivo fonte e deseja construir um pacote contendo os binários para que o programa possa efetivamente ser instalado, utilize o comando rpm com a opção `--rebuild pacote.src.rpm`.
- O pacote será construído e colocado no diretório `/usr/src/rpm/RPMS/arquitetura`, onde arquitetura é o processador para o qual o pacote foi compilado, normalmente i386.
- Proceda então a instalação como faria normalmente.

# Pacotes RPM

- **Gerenciador YUM**

- O yum é escrito em python e é utilizado pelo Fedora e Red Hat a muitos anos. Ele já provou que funciona e, mesmo com algumas críticas em relação a sua velocidade se comparada com outros gerenciadores de pacotes, ele faz o trabalho, mesmo que um pouco mais lento.
- A configuração padrão do yum está no arquivo `/etc/yum.conf` e existe um arquivo de configuração para cada repositório no diretório `/etc/yum.repos.d/`.
- Esses arquivos, quando instalados, são suficientes para que o instalador do Red Hat/Fedora adicione todos os pacotes dos repositórios padrões do sistema. Se quiser conhecer mais opções para alterar o arquivo de configuração, o manual do yum.conf poderá lhe ajudar muito.

# Pacotes RPM

- O yum é bem focado. A maioria das pessoas preferirá utilizar alguma interface gráfica para ele, mas conhecendo os comandos diretos do sistema é uma grande idéia no caso do seu servidor X não funcionar ou quando conectado remotamente em um servidor.
- Para instalar pacotes com o yum, use a seguinte linha de comando:
- **# yum install zsh**
- Esse comando irá instalar o pacote zsh e qualquer dependência que ele possa ter.
- Você pode especificar mais de um pacote ao mesmo tempo:
- **# yum install zsh httpd php**

# Pacotes RPM

- Se não tiver certeza de qual pacote deve chamar, você pode procurar nos arquivos de metadados usando o comando de busca. Em nossa demonstração, se você estiver trabalhando com algum código python e precisa da interface do MySQL disponível, mas não sabe qual o nome do pacote que ele precisa, procure por ele:
- **# yum search MySQL | grep python**
- Com esse comando, você está procurando por qualquer pacote relacionado ao MySQL, e filtrando a resposta aos pacotes que contenham a palavra python no nome do pacote. A primeira opção nessa busca será o MySQL-python, que é o pacote que você está procurando.
- Se quiser uma lista de pacotes disponíveis, você pode usar o comando "list". Isso irá listar todos os pacotes disponíveis informando os que estão instalados. Isso é muito útil se você estiver utilizando um sistema 64 bits e precisa de pacotes 32 bits. Exemplo:

# Pacotes RPM

- `[root@fw3w ~]# yum list openssl`
- Loaded plugins: fastestmirror, presto
- Loading mirror speeds from cached hostfile
- \* base: centos-mirror.hostdime.com.br
- \* extras: centos.redeminastelecom.com.br
- \* rpmforge: ftp-stud.fht-esslingen.de
- \* updates: centos-mirror.hostdime.com.br
- Installed Packages
- `openssl.i686 1.0.0-4.el6_0.2 @updates`

# Pacotes RPM

- Para fazer uma atualização completa do S.O.:
- **# yum update**
- Para remover um pacote do seu sistema, use o comando "remove".
- **# yum remove httpd**
- Isso irá remover pacotes de dependências também. Para ver informações completas de um pacote, como a versão, arquitetura, e descrição, use:
- **# yum info httpd**
- Para saber se um pacote precisa ser atualizado, mas sem rodar a função de atualização, use:
- **# yum check-update**

# Dispositivos de Armazenamento - FDISK

- Com o fdisk podemos criar novas partições em um disco, até 4 partições primárias e infinitas partições lógicas, dependendo somente do tamanho do disco (levando em conta que cada partição requer um mínimo de 40MB).
- Também podemos modificar ou deletar partições já existentes ou recém criadas no disco.
- Volume físico: Disco rígido físico;
- Partição primária: Divisão lógica de um HD usada para instalação de um Sistema Operacional;
- Partição lógica: : Divisão lógica de um HD usada para guardar informações gerais. Não é bootável.

# Dispositivos de Armazenamento - FDISK

- Antes de criar uma nova partição, modificar ou deletar, precisaremos verificar todas as partições que existem no disco. Então para esta missão usaremos o "fdisk -l", como abaixo:

```
[root@fw3w ~]# fdisk -l
```

```
Disk /dev/sda: 40.1 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x45844584
```

```
Dispositivo Boot      Start      End      Blocks  Id System
/dev/sda1  *           1         32      256000  83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2           32        415     3072000  82  Linux swap / Solaris
Partition 2 does not end on cylinder boundary.
/dev/sda3          415       4871    35791872  83  Linux
[root@fw3w ~]# █
```

# Dispositivos de Armazenamento - FDISK

- Para ver apenas as partições de um determinado disco podemos incluir no comando o nome do disco como segue:
- `# fdisk -l /dev/sda`
- Para ver o "help" com todos os comandos que podem ser usados no disco podemos usar o `fdisk` sem a opção `"-l"` e usar o comando `"m"`, como no exemplo:

```
[root@fw3w ~]# fdisk /dev/sda
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
```

```
Comando (m para ajuda): m
```

```
Comando - ação
```

```
a alterna a opção "inicializável"
b edita rótulo BSD no disco
c alterna a opção "compatibilidade"
d exclui uma partição
l lista os tipos de partição conhecidos
m mostra este menu
n cria uma nova partição
o cria uma nova tabela de partições DOS vazia
p mostra a tabela de partições
q sai sem salvar as alterações
s cria um novo rótulo de disco Sun vazio
t altera a identificação da partição para o sistema
u altera as unidades das entradas mostradas
v verifica a tabela de partições
w grava a tabela no disco e sai
x funcionalidade adicional (somente para usuários avançados)
```

# Dispositivos de Armazenamento - FDISK

- Digamos que você queira deletar algumas partições do disco.
- Primeiro vamos ver as partições, usando agora o comando "p", faça o seguinte:

```
Comando (m para ajuda): p
```

```
Disk /dev/sda: 40.1 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x45844584
```

Dispositivo	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	32	256000	83	Linux
Partition 1 does not end on cylinder boundary.						
/dev/sda2		32	415	3072000	82	Linux swap / Solaris
Partition 2 does not end on cylinder boundary.						
/dev/sda3		415	4871	35791872	83	Linux

# Dispositivos de Armazenamento - FDISK

- Visto, vamos deletar sda2 usando o comando "d", desta maneira:
- **Comando (m para ajuda): d**
- Número da partição (1-9): 1
- E gravar as alterações usando o comando "w", assim:
- **Comando (m para ajuda): w**
- A tabela de partições foi alterada!
- Chamando ioctl() para reler tabela de partições.
- Sincronizando discos.

# Dispositivos de Armazenamento - FDISK

- Podemos criar uma nova partição de um tamanho específico utilizando o comando "n", conforme abaixo:
- **Comando (m para ajuda): n**
- **Comando – ação**
- **l lógica (5 ou superior)**
- **p partição primária (1-4)**
- **Primeiro setor (305801216-625141759, padrão 305801216):**
- **Usando valor padrão 305801216**
- **Last setor, +setores or +size{K,M,G}**
- **(305801216-312111103, padrão 312111103):**
- **Usando valor padrão 312111103**

# Dispositivos de Armazenamento - FDISK

- No caso o comando perguntou se desejava criar uma partição lógica ou uma partição primária, para o caso escolheremos uma partição lógica teclando "l".
- Logo o comando faz as perguntas que vão definir o tamanho e a localização da partição. No caso simplesmente demos "enter" para usar os valores padrão e todo o espaço disponível em disco para a partição.
- Lembre-se que a cada modificação é preciso salvar:
- **Comando (m para ajuda): w**
- **A tabela de partições foi alterada!**
- **Chamando ioctl() para reler tabela de partições.**
- **Sincronizando discos.**

# Dispositivos de Armazenamento - MKFS

- Depois de criada a partição deve ser formatada com o comando "mkfs", de acordo com o tipo de sistema de arquivos desejado. Digamos que tenhamos criado a partição "9", alguns exemplos seriam:
  - 1) Para ext4:
    - # mkfs.ext4 /dev/sda9
  - 2) Para reiserfs:
    - # mkfs.reiserfs /dev/sda9
  - 3) Para ntfs (Windows XP e outros):
    - # mkfs.ntfs /dev/sda9
  - Partição criada e formatada, agora é só usar.

# Dispositivos de Armazenamento - MKFS

- Para ver o tamanho de uma partição podemos usar a opção "-s" como no exemplo:
- # fdisk -s /dev/sda9
- 3154944
- O número apresentado é o tamanho da partição em blocos, para este caso isso equivale a cerca de 3100MB.
- Outros exemplos:
- # fdisk -s /dev/sda1
- 79629573
- # fdisk -s /dev/sda5
- 6007808

# Usando o comando mount

- Geralmente deve ser informado como parâmetro um dispositivo a ser montado e o ponto de montagem.
- Existe outra possibilidade de montagem manual informando somente o dispositivo, ou somente o ponto de montagem.
- Quando isso acontece, o dispositivo ou o ponto de montagem devem existir configurados no arquivo `/etc/fstab`:
  - `$ mount [opções] dispositivo`
  - `$ mount [opções] diretório`

# Usando o comando mount

- Arquivo `/etc/fstab`: contém as informações de montagem dos dispositivos que são montados durante o processo de carga do sistema.
- O `mount` também possibilita as seguintes notações de dispositivo sejam usadas para se referir a um dispositivo:

Notação	Descrição
<code>/dev/sdaX</code>	Especifica o dispositivo pelo nome dele no sistema;
<code>LABEL=&lt;rótulo&gt;</code>	Especifica o dispositivo pelo rótulo do sistema de arquivos;
<code>UUID=&lt;uuid&gt;</code>	Especifica o dispositivo pelo UUID do sistema de arquivos;
<code>PARTLABEL=&lt;rótulo&gt;</code>	Especifica o dispositivo pelo rótulo da partição;
<code>PARTUUID=&lt;uuid&gt;</code>	Especifica o dispositivo pelo UUID da partição;

# Usando o comando mount

- As opções mais frequentes são:
- -a Monta todos os dispositivos especificados no arquivo `/etc/fstab` que não tem a opção `noauto` selecionada;
- -r Monta o sistema de arquivos do dispositivo como somente leitura;
- -w Monta o sistema de arquivos do dispositivo para leitura e gravação;
- -o Especifica as opções de montagem;
- -t tipo Especifica o tipo do sistema de arquivos do dispositivo. Exemplo: `ext2`, `ext3`, `ext4`, `iso9660`, `msdos`, `xfs`, `btrfs`, `nfs`, etc.
- Para montar os diversos sistemas de arquivos, o Linux precisa estar com o suporte a estes sistemas de arquivos habilitados no Kernel ou carregado na forma de um módulo.

# Usando o comando mount

- Arquivo `/etc/filesystems`: fornece uma lista dos sistemas de arquivos que estão habilitados e aceitos pelo mount:

```
$ cat /etc/filesystems
xfs
ext4
ext3
ext2
nodev proc
iso9660
vfat
```

# Usando o comando mount

- Exemplos de uso do mount:
- Para montar um cartão USB:
- `# mount /dev/sdb1 /mnt`
- Para montar uma partição Windows (vfat) em `/dev/sda1` no ponto de montagem `/mnt` para somente para leitura:
- `# mount /dev/sda1 /mnt -r -t vfat`
- Para remontar a partição raiz como somente leitura:
- `# mount -o remount,r /`

# Usando o comando mount

- Para montar uma imagem ISO, a opção “-o loop” deve ser usada:
- `# mount /tmp/imagem.iso /media/iso -o loop`
- Para que o mount remonte todos os pontos de montagem definidos em `/etc/fstab`:
- `# mount -a`
- Se não for passado nenhum parâmetro ou opção, o mount mostra todos os dispositivos montados, incluindo os pseudo-file-system `/proc`, `/dev`, `/run` e `/sys`.
- Ele também mostra as opções de montagem, como leitura e gravação, bem como o sistema de arquivos:

# Usando o comando mount

```
$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=1001736k,nr_inodes=250434,mode=755)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
/dev/xvda1 on / type xfs (rw,noatime,attr2,inode64,noquota)
/dev/xvdf on /home type ext4 (rw,noatime,data=ordered)
```

- Para montar um dispositivo usando o UUID dele, primeiro deve-se ver qual é o UUID do dispositivo com o comando blkid:
- # blkid
- /dev/sda1: UUID="1c3b15b1-cd13-4a73-b2a8-449fa3aa039f" TYPE="xfs"
- # mount UUID="1c3b15b1-cd13-4a73-b2a8-449fa3aa039f" /mnt
- A vantagem de se usar o UUID é que esse ID do disco não muda nunca.
- Já a nomenclatura do disco pode ser alterada quando um novo disco é inserido ou alterado de lugar no cabo.
- Desta forma, um disco nomeado como /dev/sda, pode virar /dev/sdb. Se forem referenciados pelo UUID, esse número não altera.

# Usando o comando mount

- Arquivo `/etc/mtab`
- O sistema mantém um arquivo chamado `/etc/mtab` que contém as informações sobre os dispositivos que estão montados no sistema.
- Ele pode conter os pontos de montagem indicados no arquivo `/etc/fstab` e também os dispositivos montados manualmente.
- O arquivo `/etc/mtab` é mantido pelo sistema, através do comando `mount`.
- O conteúdo deste arquivo é muito similar ao arquivo `/proc/mounts`.
- O Linux usa dois mecanismos diferentes para acompanhar os dispositivos montados.
- Um é implementado (leia sobre `deploy`) no kernel e expõe esta informação ao espaço do usuário via arquivo `/proc/mounts`.
- O outro mecanismo mantém a contabilidade dos pontos de montagem pelo utilitário `mount` usando o arquivo `/etc/mtab`.

# Usando o comando mount

- Arquivo /proc/partitions
- Cada disco detectado no sistema irá aparecer no arquivo /proc/partitions.
- É neste arquivo que o Kernel mantém uma lista dos discos detectados, suas partições e tamanho.

```
$ cat /proc/partitions
major minor #blocks name
 8     0 488386584 sda
 8     1   524288 sda1
 8     2  2097152 sda2
 8     3  2097152 sda3
 8     4 483666951 sda4
 8    16 117220824 sdb
 8    17 117219328 sdb1
 8    48  31950720 sdd
 8    49   204800 sdd1
 8    50  31744000 sdd2
```

# Usando o comando mount

- A ferramenta lsblk pode fornecer a mesma informação de uma forma muito mais clara:

```
$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0  0 465,8G  0 disk
├─sda1 8:1  0  512M  0 part /boot/efi
├─sda2 8:2  0    2G  0 part [SWAP]
├─sda3 8:3  0    2G  0 part
└─sda4 8:4  0 461,3G  0 part
sdb   8:16 0 111,8G  0 disk
└─sdb1 8:17 0 111,8G  0 part
sdd   8:48 1  30,5G  0 disk
├─sdd1 8:49 1   200M  0 part
└─sdd2 8:50 1  30,3G  0 part /media/uira/UNTITLED
```

- Neste computador existem dois discos rígidos: sda com 4 partições, e sdb com uma única partição.
- E ainda existe um pendrive de 30 GB, montado no diretório /media/uira/UNTITLED.

# Usando o comando umount

- O comando umount é utilizado para desmontar dispositivos montados pelo comando mount.
- O umount sincroniza o conteúdo do disco com o conteúdo dos buffers (memória própria para transferência de dados) e libera o diretório de montagem.
- Ele tanto pode receber como parâmetro o dispositivo que está montado, quanto o diretório do ponto de montagem:
- `$ umount [opções] dispositivo`
- Ou
- `$ umount [opções] diretório`

# Usando o comando umount

- As opções disponíveis são:
- -a Desmonta todos os dispositivos listados no arquivo `/etc/mtab`, que é mantido pelo comando `mount` como referência de todos os dispositivos montados;
- -f Força que um dispositivo seja desmontado;
- -t tipo Desmonta somente os dispositivos que contenham o sistema de arquivos especificado no tipo.
  
- Nenhum arquivo pode estar em uso no sistema de arquivos que será desmontado.
- Exemplos:
- `$ umount /dev/cdrom`
- Ou
- `$ umount /media/cdrom`