



APRESENTAÇÃO

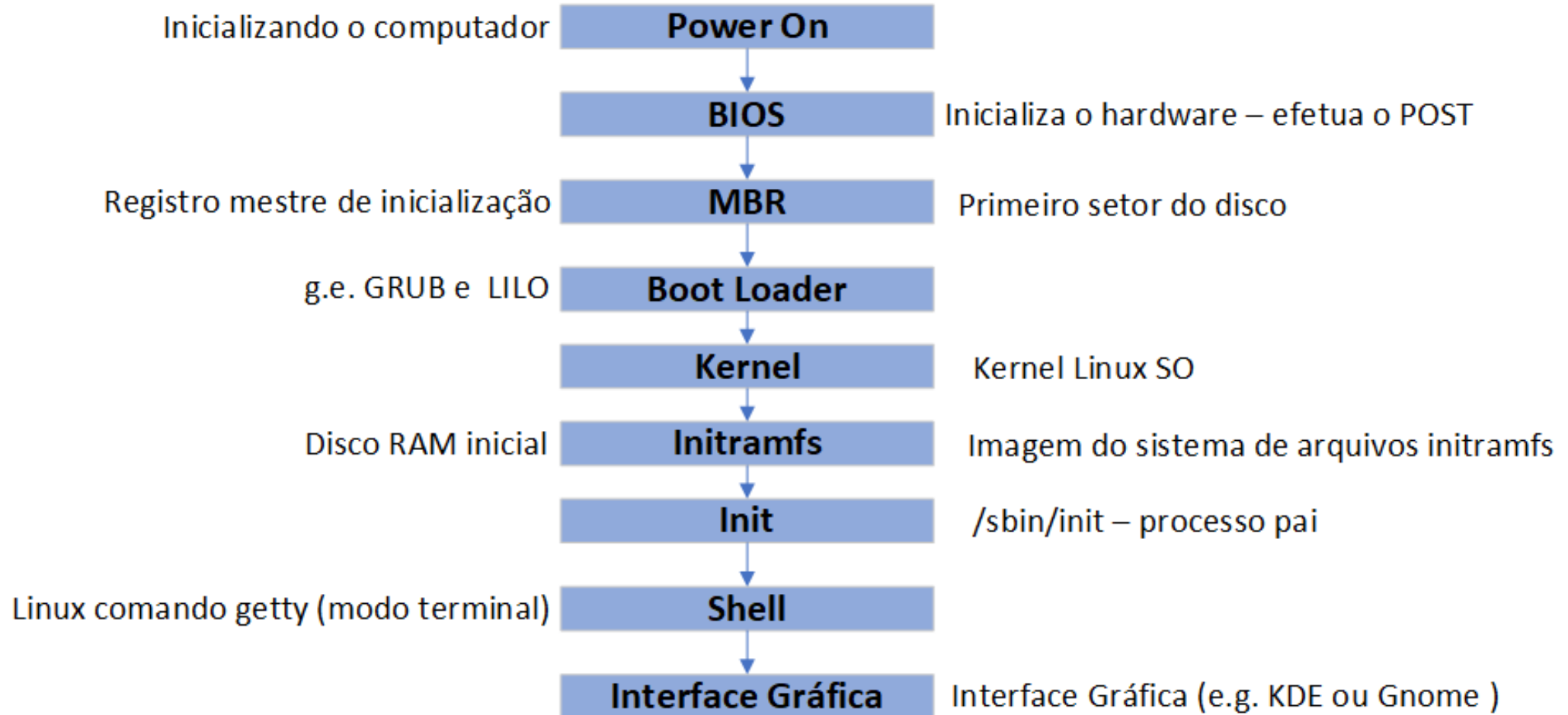
Clínica de Tecnologia da Informação e Comunicação

Sistemas Operacionais

Prof. MSc. Jhonatan Geremias
jhonatan.geremias@pucpr.br



Processo de Inicialização



Correções de Problemas na Inicialização

Erros MBR - Windows

Razões que causam problema na MBR:

- Partição do disco rígido danificada;
 - Erro de gravação no disco ou ataques de vírus;
- BIOS do sistema não suportando o disco rígido;
- Disco rígido corrompido;
- Arquivos e pastas do sistema operacional danificados;
 - O sistema operacional não pode acessar os arquivos do sistema para inicializar corretamente.



Correções de Problemas na Inicialização

Recuperar o MBR - Windows

1. Inicializar o computador utilizando um DVD de instalação do Windows;
2. Acessar opção reparar o computador;
3. Clicar em solução de problemas;
4. Abrir prompt de comando.

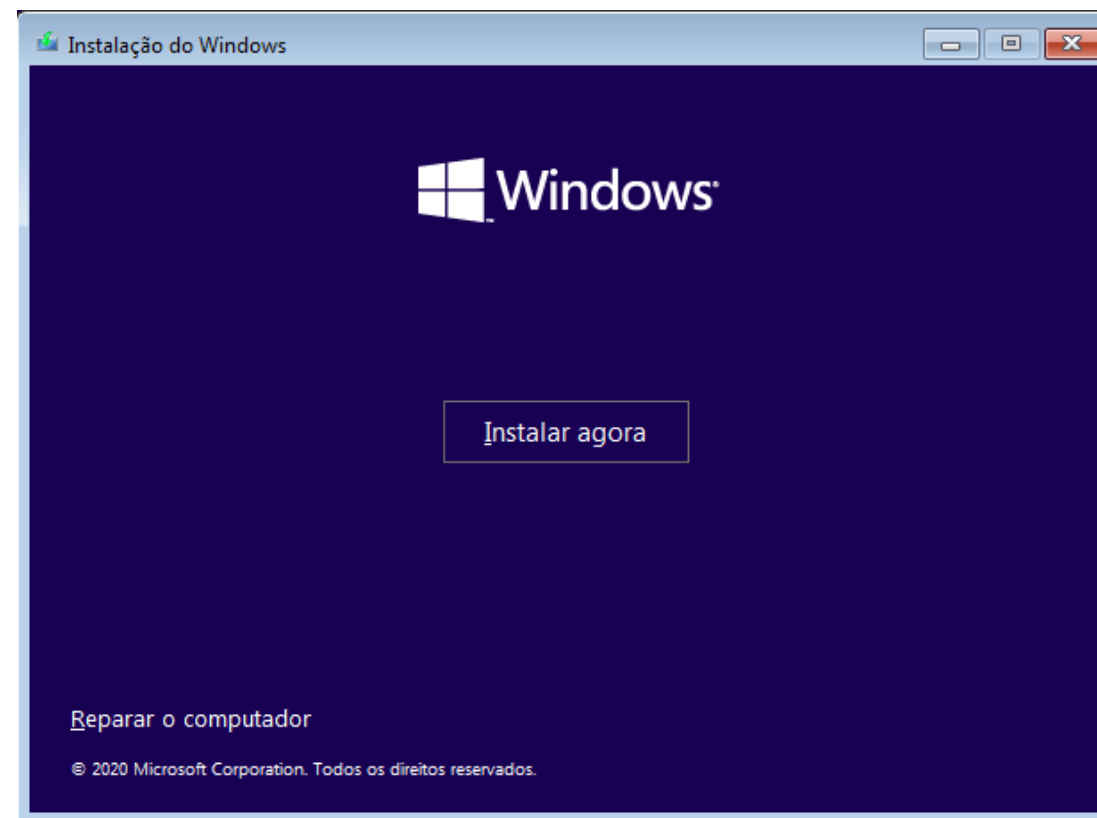
Comandos:

`bootrec /FixMbr`

`bootrec /FixBoot`

`bootrec /ScanOS`

`bootrec /RebuildBcd`



Correções de Problemas na Inicialização

Opções bootrec - Windows

- **/FixMbr** - grava o registro mestre de inicialização da partição do sistema;
- **/FixBoot** - escreve um novo setor de inicialização na partição do sistema;

Examina todos os discos em busca de instalações compatíveis com o Windows:

- **/ScanOS** - exibe as entradas que não estão no repositório de configuração de inicialização;
- **/RebuildBcd** - permite que o usuário escolha qual delas adicionar ao repositório da configuração de inicialização.



Processo de Inicialização – GRUB

- Gerenciador de inicialização;
- O GRUB reside no /boot;
- Permite selecionar qual sistema operacional será utilizado;
 - Definir a ordem de boot do SO.

```
GNU GRUB  version 2.00-19ubuntu2

Ubuntu
Advanced options for Ubuntu
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Windows 7 (loader) (on /dev/sda1)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands before booting or 'c'
for a command-line.
The highlighted entry will be executed automatically in 8s..
```



Processo de Inicialização – Características GRUB

- Permite carregar uma grande variedade de SO's;
- Altamente flexível:
 - Gerencia vários tipos de sistemas de arquivos e kernel;
 - Carrega um SO sem a necessidade de gravar uma posição física do kernel no disco;
- Interface gráfica amigável e configurável;



Processo de Inicialização – Configuração SETUP

1. Inicializar o computador utilizando um Live CD;
2. No terminal verificar as partições;
3. Criar uma pasta para definir o ponto de montagem;
4. Montar a partição raiz;
5. Acessar a partição raiz montada como root;
6. Atualize a lista do GRUB;
7. Reinstalar o GRUB.

Recuperar o GRUB

Comandos:

```
sudo fdisk -l  
sudo mkdir /mnt/hda1  
sudo mount /dev/hda1 /mnt/hda1  
sudo chroot /mnt/hda1  
update-grub  
grub-install /dev/hda
```



Correções de Problemas na Inicialização

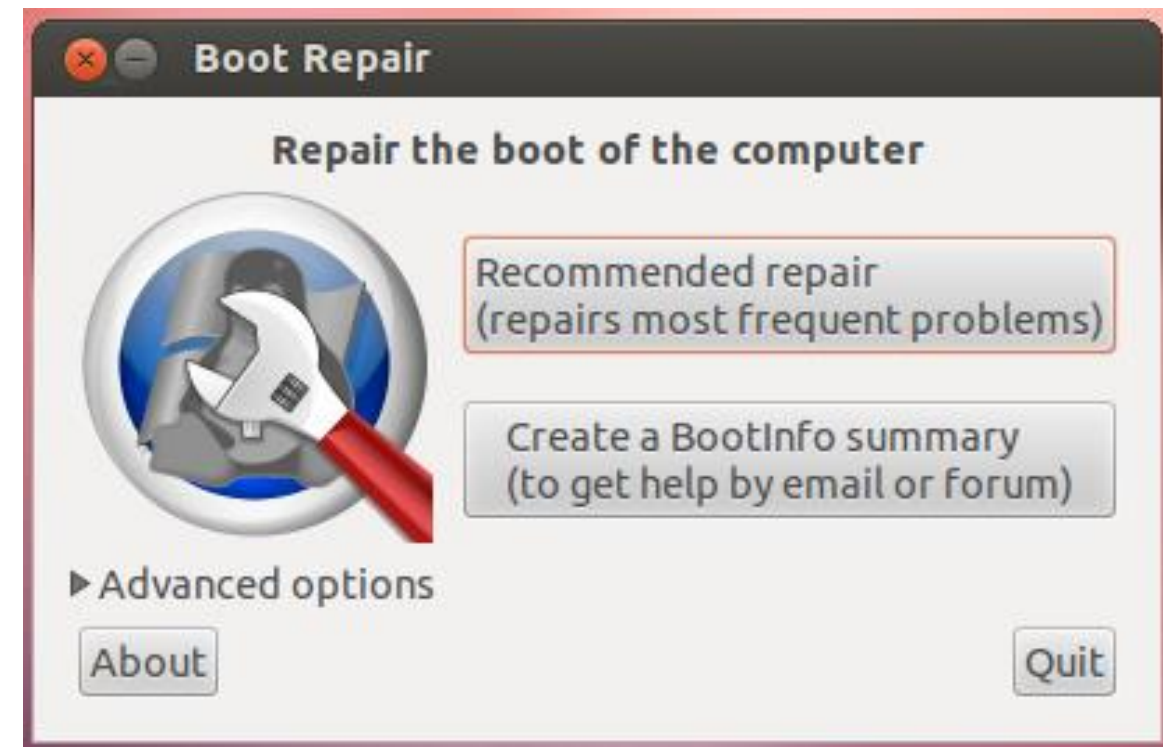
Recuperar o GRUB

Utilizando o Boot-Repair

1. Inicializar o computador utilizando um Live CD;
2. Instalar o Boot-Repair;
3. Executar o Boot-Repair.

Comandos:

```
sudo -s  
add-apt-repository ppa:yannubuntu/boot-repair  
apt-get update  
apt-get install -y boot-repair  
boot-repair
```



Correções de Problemas na Inicialização

Recuperar Sistema

Opções avançadas da inicialização do Ubuntu

```
Recovery Menu (filesystem state: read-only)
```

```
[ OK ] Finished Daily man-db regeneration.
```

clean	Try to make free space
dpkg	Repair broken packages
fsck	Check all file systems
grub	Update grub bootloader
network	Enable networking
root	Drop to root shell prompt
system-summary	System summary

```
<Ok>
```



Correções de Problemas na Inicialização

Identificar MBR corrompida - Windows

MBR danificado ou corrompido, o sistema exibirá um dos seguintes erros:

- Erro carregando o sistema operacional;
- Sistema operacional não encontrado;
- Tabela de partição inválida;
- Nenhum meio inicializável encontrado;
- Reinicie e selecione o dispositivo de inicialização adequado.



Correções de Problemas na Inicialização

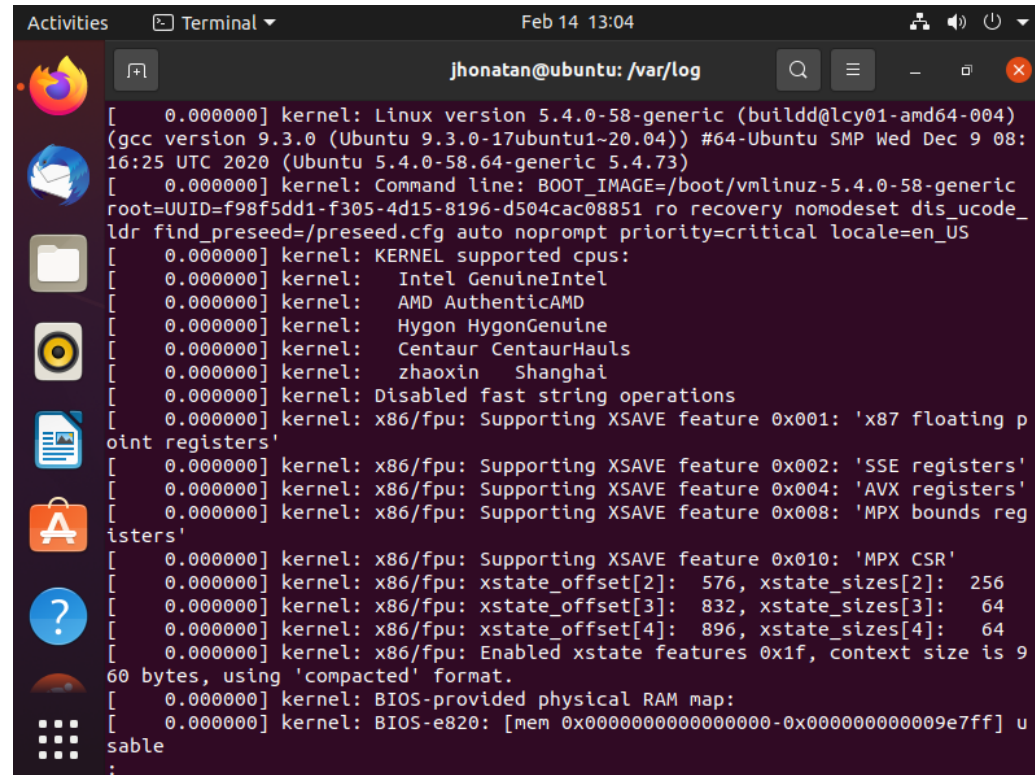
Identificar erros na inicialização - Linux

- Cada vez que é feita a inicialização no sistema Linux, o arquivo de mensagens é gravado e pode ser analisado.

Comando:

`dmesg`

`cat /var/log/dmesg | less`



The screenshot shows a terminal window titled 'Terminal' with the user 'jhonatan@ubuntu' in the directory '/var/log'. The terminal displays the output of the 'dmesg' command, showing kernel boot messages. The messages include the Linux version (5.4.0-58-generic), the boot command line, and various hardware and kernel initialization details. The terminal output is as follows:

```
[ 0.000000] kernel: Linux version 5.4.0-58-generic (build@lcy01-amd64-004)
(gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #64-Ubuntu SMP Wed Dec 9 08:
16:25 UTC 2020 (Ubuntu 5.4.0-58.64-generic 5.4.73)
[ 0.000000] kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-58-generic
root=UUID=f98f5dd1-f305-4d15-8196-d504cac08851 ro recovery nomodeset dis_ucode_
ldr find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US
[ 0.000000] kernel: KERNEL supported cpus:
[ 0.000000] kernel: Intel GenuineIntel
[ 0.000000] kernel: AMD AuthenticAMD
[ 0.000000] kernel: Hygon HygonGenuine
[ 0.000000] kernel: Centaur CentaurHauls
[ 0.000000] kernel: zhaoxin Shanghai
[ 0.000000] kernel: Disabled fast string operations
[ 0.000000] kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating p
oint registers'
[ 0.000000] kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] kernel: x86/fpu: Supporting XSAVE feature 0x008: 'MPX bounds reg
isters'
[ 0.000000] kernel: x86/fpu: Supporting XSAVE feature 0x010: 'MPX CSR'
[ 0.000000] kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] kernel: x86/fpu: xstate_offset[3]: 832, xstate_sizes[3]: 64
[ 0.000000] kernel: x86/fpu: xstate_offset[4]: 896, xstate_sizes[4]: 64
[ 0.000000] kernel: x86/fpu: Enabled xstate features 0x1f, context size is 9
60 bytes, using 'compact' format.
[ 0.000000] kernel: BIOS-provided physical RAM map:
[ 0.000000] kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009e7ff] u
sable
:
```



Operações Básicas - Linux

- `login`: abre uma nova sessão para um usuário;
- `logout`: permite desconectar um usuário de uma determinada sessão;
- `exit`: encerra uma sessão de trabalho;
- `shutdown -h now`: encerramento do sistema;
- `halt`, `reboot` e `init 6`: reinicialização da máquina;
- `dir` e `ls`: permite listar o conteúdo de um diretório;
- `cd <diretório>`: permite navegar nos diretórios;
 - `cd -`: retorna a posição do diretório anterior;
 - `cd ..` retorna um diretório na posição hierárquica;
 - `cd ~`: retorna ao diretório /home do usuário;
- `pwd`: retorna o caminho do diretório corrente.



Informações sobre o sistema

- **lsusb**: mostra informações sobre os dispositivos usb conectados;





GPU – Placa de vídeos

- As GPUs são mais conhecidas como placas de vídeo;
- A sigla significa Unidade de processamento gráfico (*Graphics Processing Unit*);
- O termo foi criado pela Sony para designar o chip gráfico do PlayStation, por volta de 1994;
- Porém começou a ser mais difundido pela NVIDIA em 1999;
 - Placas de vídeo GeForce.
 - Permitindo o uso de efeitos avançados de transformação e iluminação de polígonos.



GPU - Graphics Processing Unit

- As GPUs são compostas por diversos núcleos (muitas centenas);
- Especializadas em processar elementos gráficos 2D e 3D;
- Responsáveis em gerar as imagens que chegam em nossos monitores;
- Permitem rodar programas e jogos em uma resolução adequada;



GPU - Recursos

- A arquitetura das GPUs é balanceada pelas fabricantes de acordo com aquilo que é mais utilizado em um dado momento.
- Elas podem priorizar questões como processamento de shaders, texturas, anti-aliasing, entre outros.
 - *Shaders* são pequenos programas feitos dentro de unidades gráficas de processamento;
 - *Anti-Aliasing* – Aplicação de filtros para embaçar as bordas de uma linha, removendo a impressão do serrilhado.



Informações Sobre o Sistema

- `cat /etc/issue`: mostra a versão da distribuição;
- `uname -a`: mostra informações do sistema;
- `lspci`: mostra informações sobre as placas PCI instaladas;
- `lsusb`: mostra informações sobre os dispositivos usb conectados;
- `cat /proc/version`: traz a versão o kernel que está sendo utilizado;
- `lshw`: exibe informações sobre o hardware;
- `lsmod`: exibe os módulos do kernel que estão carregados na memória.



Informações Sobre o Sistema

- **lshw**: exibe informação sobre o hardware;
- Exporta um relatório completo em html.

Comando:

```
lshw -html > index.html
```

id:	memory
description:	System memory
physical id:	0
size:	4GiB

id:	cpu:0
product:	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz
vendor:	Intel Corp.
physical id:	1
bus info:	cpu@0
width:	64 bits
capabilities:	<i>fpu fpu_exception wp vme de pse tsc msr pae mce ss syscall nx pdpe1gb rdtscp x86-64 constant_tsc pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic hypervisor lahf_lm abm 3dnowprefetch cpuid_fault smep bmi2 invpcid mpx rdseed adx smap clflush</i>



Sistema de arquivo – Linux

1/2

- `/` - diretório raiz;
- `/root` - diretório local do superusuário (usuário root) ;
- `/bin` – diretório que armazena os arquivos binários de comandos essenciais do sistema e mais frequentemente utilizados;
- `/sbin` – arquivos essenciais, executáveis de administração do sistema;
- `/lib` – arquivos de bibliotecas compartilhados para o kernel e aplicativos;
- `/boot` – arquivos estáticos de boot/bootloader usados na inicialização;
- `/dev` – arquivos usados para acessar os dispositivos de entrada/saída;
- `/etc` – configuração do sistema da máquina local, como rede, som e vídeo.



Sistema de arquivo – Linux

2/2

- `/home` – diretório local com os arquivos dos usuários;
- `/mnt` – ponto de montagem de partição que receberá o conteúdo da mídia que for montada;
- `/tmp` – arquivos temporários gerados por alguns utilitários;
- `/usr` – diretório de instalação de programas e aplicativos;
- `/var` – informações variáveis, contém a maior parte dos arquivos que são gravados com frequência pelos programas dos sistemas
 - Ex.: caches, spools, logs, e-mails;
- `/proc` – armazena e atualiza informações do sistema em tempo real;
 - Ex.: configuração de hardware, programas em execução, recurso de memória, dispositivos PCI e muito outros.



Comandos de verificação

- `pwd`: mostra o caminho do diretório corrente;
- `hostname`: mostra o nome da máquina;
- `whoami`: mostra o nome usuário logado;
- `who` ou `w`: usuários logados na máquina;
- `rwho`: mostra usuários logados em outros computadores na rede;
- `id`: exibe identificação do usuário e grupo (**user id** e **group id**).



Comandos de verificação – Data e Hora

- **date**: exibe a data e hora atual, dados armazenados no relógio do sistema;
- **time**: quantidade de tempo gasto durante a execução de um comando;
- **timeconfig** ou **timedatectl**: ajustar a zona de horário do sistema (**timezone**);
- **uptime**: quantidade de tempo decorrido desde o último reboot;
- **hwclock**: mostrar e ajustar o relógio do hardware do seu computador.



Comandos de verificação – Histórico

- **history**: exibe os últimos 1000 comandos executados em modo texto na sessão atual;
 - Histórico mantido no arquivo **.bash_history** no diretório padrão do usuário.

Comando:

`history | more`

`history 10`

`history >> arquivo.txt`

```
jhonatan@ubuntu:~$ history 10
30  pwd
31  cd ..
32  ls
33  cd jhonatan/
34  ls
35  pwd
36  ls
37  ps -aux
38  history
39  history 10
jhonatan@ubuntu:~$
```



Comandos de verificação – Processos

Verificar os processos em execução:

- **ps**: responsável por listar os processos atuais;
- **top**: mostra todos os processos em tempo de execução.

Comando:

ps -aux

top

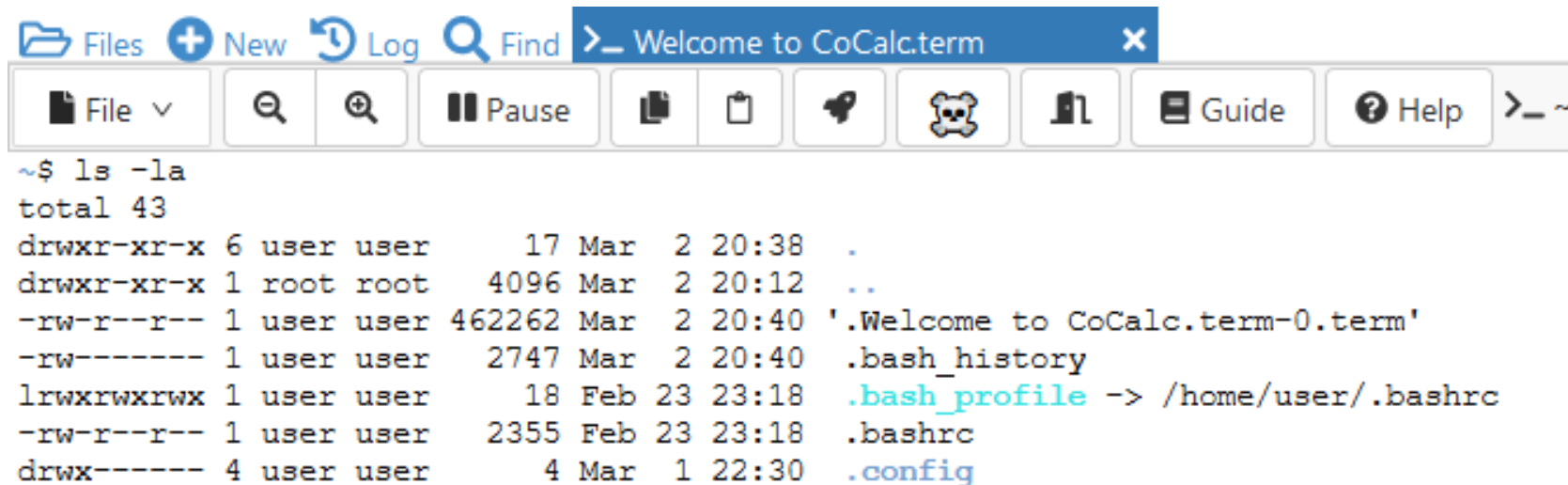
```
top - 09:07:46 up 3:14, 1 user, load average: 0.00, 0.02, 0.05
Tasks: 278 total, 1 running, 277 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.3 us, 1.7 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.0 st
MiB Mem : 3908.5 total, 1638.8 free, 1220.2 used, 1049.5 buff/cache
MiB Swap: 1162.4 total, 1162.4 free, 0.0 used. 2336.0 swap
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM
4406	jhonatan	20	0	3145828	289312	156036	S	2.6	7.2
3567	jhonatan	20	0	283636	68056	42336	S	2.0	1.7
5351	jhonatan	20	0	2421744	136580	113676	S	1.3	3.4
4940	jhonatan	20	0	4260164	246024	96752	S	1.0	6.1
5891	jhonatan	20	0	814724	51308	38780	S	0.7	1.3



Comando para Gerenciamento de Arquivos

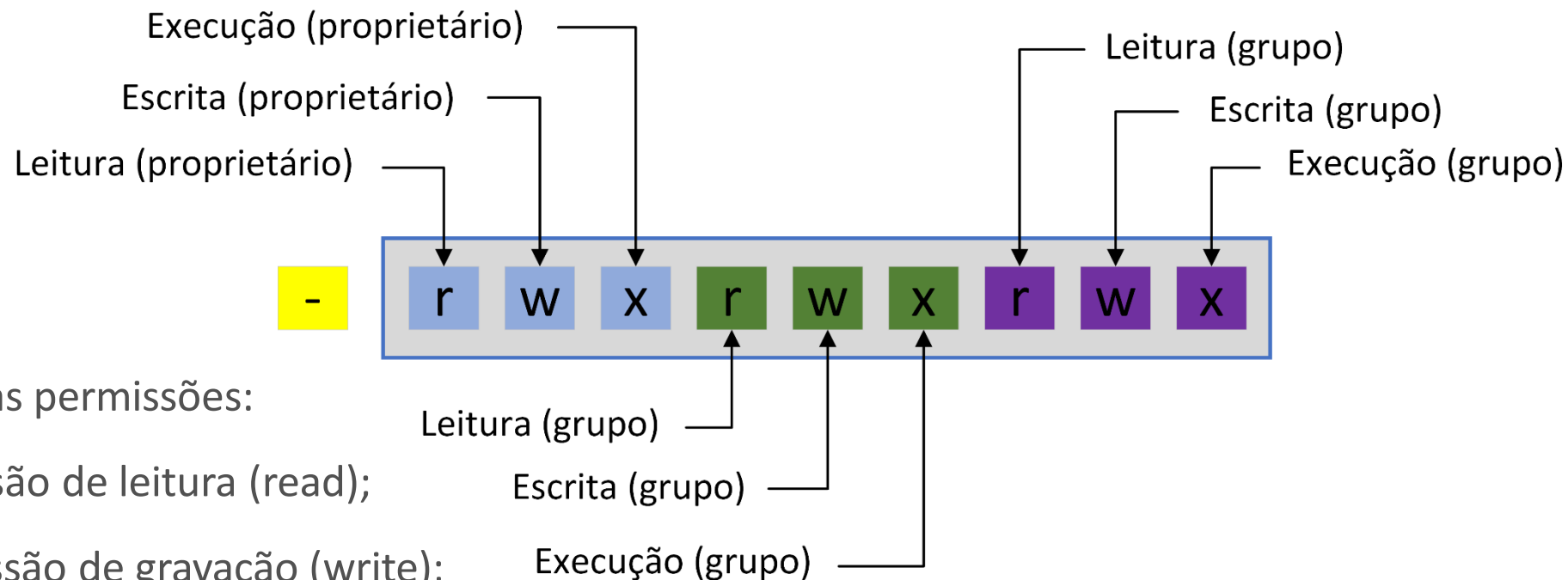
- `ls <filename>`: lista conteúdo de um diretório
 - Opções de argumentos:
 - `l` – listagem com detalhes, tipo de permissão, dono do diretório ou arquivo, grupo, etc;
 - `a` – listagem com arquivos ocultos;
 - `d` – lista somente diretórios – arquivos com `.`(ponto) no início do arquivo;
 - `F` – listagem diferenciada com caractere no nome.



```
~$ ls -la
total 43
drwxr-xr-x 6 user user    17 Mar  2 20:38 .
drwxr-xr-x 1 root root 4096 Mar  2 20:12 ..
-rw-r--r-- 1 user user 462262 Mar  2 20:40 '.Welcome to CoCalc.term-0.term'
-rw----- 1 user user   2747 Mar  2 20:40 .bash_history
lrwxrwxrwx 1 user user    18 Feb 23 23:18 .bash_profile -> /home/user/.bashrc
-rw-r--r-- 1 user user   2355 Feb 23 23:18 .bashrc
drwx----- 4 user user     4 Mar  1 22:30 .config
```



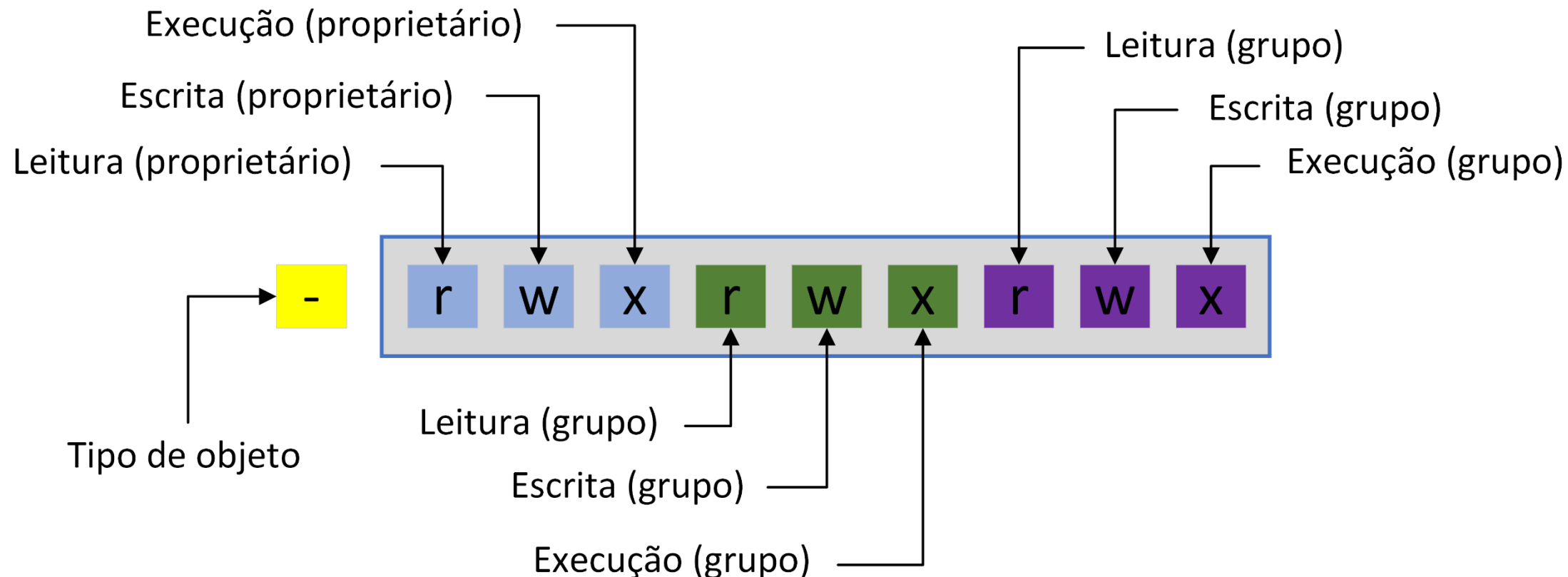
Permissões Linux



- Estrutura das permissões:
- **r** => permissão de leitura (read);
- **w** => permissão de gravação (write);
- **x** => permissão de execução (execution);
- **-** => permissão desabilitada.



Permissões Básicas - Linux



Permissões – Comando chmod

- O comando chmod permite alterar o nível de permissão de um arquivo ou diretório;
- As permissões são sempre definidas na sequência: usuário, grupo e outros;
 - A permissão pode ser definida de forma decimal ou alfabética;

- Conforme especificação:

Ex.: `chmod ugo+x`

- `r` – permissão de leitura
- `w` – permissão de gravação
- `x` – permissão de execução
- `a` – permissão atribuída a todos (*all*)
- `g` – permissão atribuída ao grupo;
- `o` – permissão atribuída a outros usuários;
- `u` – permissão atribuída ao proprietário;
- `=` – define permissão para
- `+` – adiciona a permissão
- `-` – retira permissão



Definir Permissões para o Comando chmod

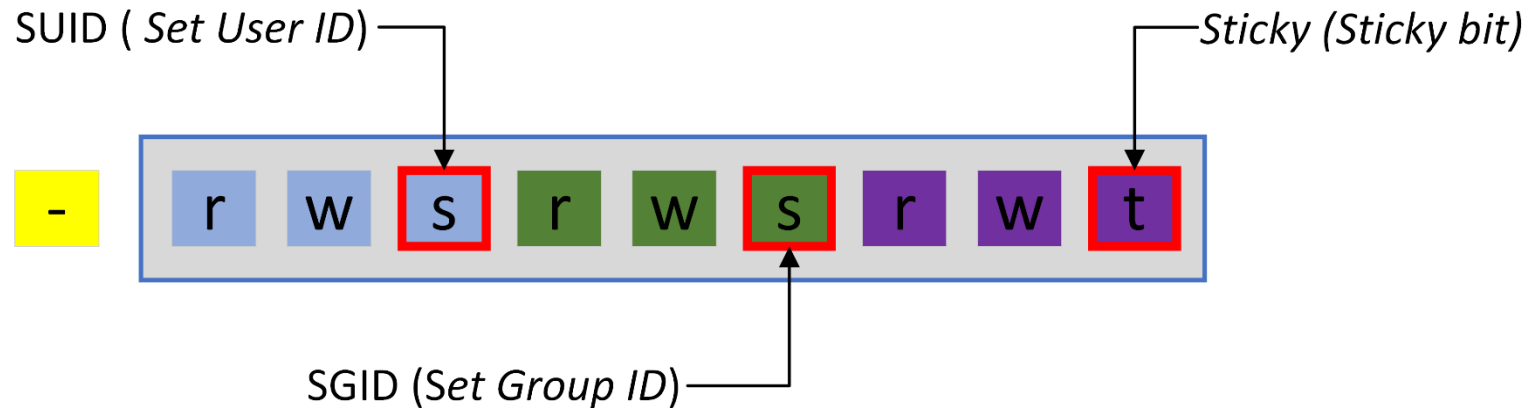
- Uma maneira para simplificar o processo é ter mente os valores de leitura, escrita e execução;
 - $r = 4$
 - $w = 2$
 - $x = 1$
- Para obter a permissão os valores decimais são somados diretamente;
 - Ex: `chmod 540 arquivo.txt`;
 - Permissão do usuário: leitura = 4, execução = 1, somando as permissões obtém o valor 5.

Permissão	Binário	Decimal
---	000	0
--X	001	1
-W-	010	2
-WX	011	3
r--	100	4
r-X	101	5
rW-	110	6
rWX	111	7

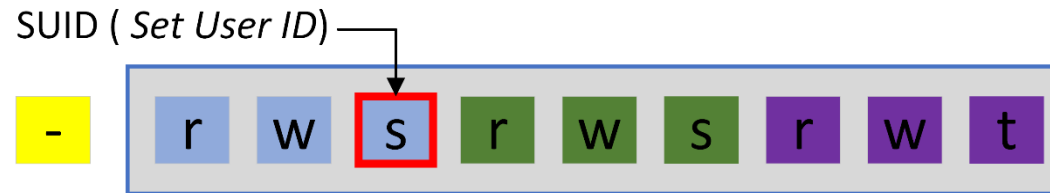


Permissões de Acesso Especiais

- Existem três modelos especiais para controle de acesso, chamados SUID (*Set User ID*), sgid (*Set Group ID*) e Sticky (*Sticky bit*);



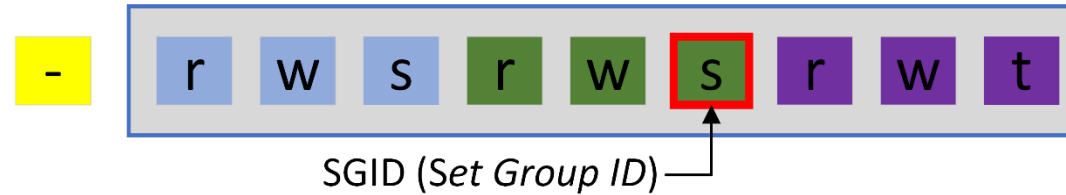
Permissões de Acesso Especiais - SUID



- A propriedade SUID é apenas para arquivos executáveis não tendo efeito sob diretórios.
 - Um arquivo executável com a propriedade SUID aplicada, o programa rodará com o ID do dono do arquivo, não com o ID do usuário que executou o programa.
 - Normalmente o usuário dono do programa executável é também dono do processo sendo executado.
 - O processo do arquivo executável utilizando o acesso SUID é executado como se estivesse sido iniciado pelo dono do arquivo.
 - A permissão de acesso especial SUID pode aparecer somente no campo proprietário;
 - Um exemplo para arquivo executável com a propriedade SUID é o arquivo `/usr/bin/passwd`.
 - Exemplo de comando: `chmod u+s arquivo.sh`



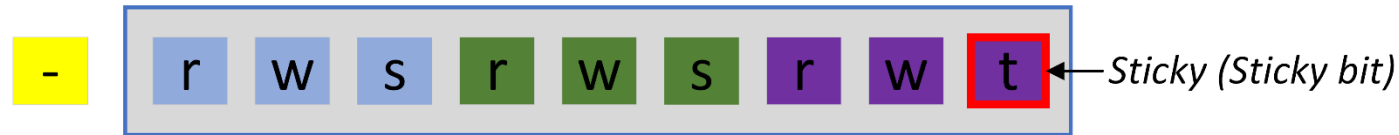
Permissões de Acesso Especiais - SGID



- A propriedade SGID tem a mesma função que o SUID para arquivos executáveis;
- Ainda o SGID tem um efeito especial aplicado em diretórios;
 - Quando SGID é aplicado em um diretório, os novos arquivos que são criados dentro do diretório assumem o mesmo ID de Grupo do diretório com a propriedade SGID aplicado;
 - A permissão de acesso especial SGID pode aparece somente no campo grupo;
 - Exemplo comando: `chmod g+s /home/equipe`



Permissões de Acesso Especiais - Sticky

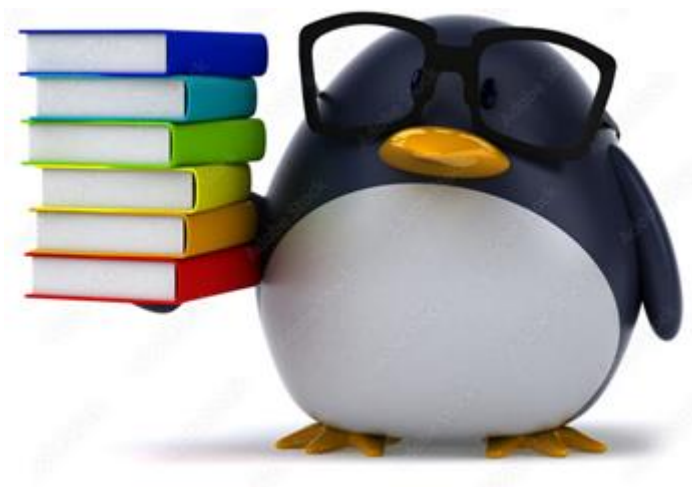


- Em arquivos executáveis, a propriedade *Sticky* faz com que o sistema mantenha uma imagem do programa em memória depois que o programa finalizar.
- Esta capacidade aumenta o desempenho;
 - Será efetuado um cache do programa para a memória;
 - A próxima vez que ele for executado, será carregado mais rápido;
- Capacidade de Aumentar a segurança;
 - A propriedade *Sticky* aplicada em diretórios impede que outros usuários deletem ou renomeie arquivos que não são donos.
 - O diretório estará em modo *append-only* (somente incremento);
 - Somente o dono do arquivo, poderá deletar ou renomear os arquivos;
 - Exemplo comando: `chmod o+t programa_pesado.sh`



Lista de Controle de Acesso Estendida

- Uma ACL (*Access Control List* / Lista de Controle de Acesso) é uma configuração de segurança que nos fornece um controle mais refinado sobre quais usuários podem acessar diretórios e arquivos específicos do que as permissões tradicionais do Linux.



Lista de Controle de Acesso Estendida

- A permissão da lista de controle de acesso (ACL) estendida é aplicada utilizando o comando **setfact**.
- Sintaxe do comando setfact:
 - Passar o parâmetro -m para adicionar a permissão,
 - Fornecer o usuário “u:<usuário>”, um sinal de dois pontos “:” e a permissão a ser atribuída
 - Permissões: leitura (r), escrita (w) e execução (x)

setfact -m u:<usuário>:<permissão> <objeto>

```
aluno@ubuntu:~/u4$ ls -la
total 8
drwxr-xr-x  2 aluno equipe1 4096 Jul  6 06:28 .
drwxr-xr-x  3 aluno equipe1 4096 Jul  6 06:26 ..
-rw-rw----+ 1 aluno equipe1    0 Jul  6 06:28 arquivo1.txt
aluno@ubuntu:~/u4$
```

getfact arquivo1.txt

```
aluno@ubuntu:~/u4$ getfacl arquivo1.txt
# file: arquivo1.txt
# owner: aluno
# group: equipe1
user::rw-
user:aluno2:rw-
user:aluno3:rw-
group::r--
mask::rw-
other::---
aluno@ubuntu:~/u4$
```



Permissões - Comando umask

- O comando `umask` permite criar uma máscara utilizada para definir as permissões de novos arquivos e diretórios;
 - Permissões aplicadas pela máscara em novos arquivos inicialmente não possui o privilégio de execução para nenhum dos usuários – valor máximo: `666 (rw-rw-rw-)` (`110-110-110`);
 - Para acesso aos diretórios essa permissão é aplicada – valor máximo: `777 (rwx-rwx-rwx)` (`111-111-111`);
- Para utilizar o `umask` é necessário subtrair o valor máximo pela permissão desejada;
 - Exemplo: de permissão para arquivo: `(rw-r-----)` a permissão seria `640`
 - Obtendo a permissão para o `umask`: $666 - 640 = 026$
 - Comando: `umask 0026`



Permissões - Comando umask

umask →

```
~$ touch arquivo1.txt
~$ mkdir pasta1
~$ ls -l
total 1
-rw-r--r-- 1 user user 0 Mar  3 22:32 arquivo1.txt
drwxr-xr-x 2 user user 2 Mar  3 22:32 pasta1
~$ umask 0017
~$ touch arquivo2.txt
~$ mkdir pasta2
~$ ls -l
total 2
-rw-r--r-- 1 user user 0 Mar  3 22:32 arquivo1.txt
-rw-rw---- 1 user user 0 Mar  3 22:33 arquivo2.txt
drwxr-xr-x 2 user user 2 Mar  3 22:32 pasta1
drwxrw---- 2 user user 2 Mar  3 22:33 pasta2
~$
```

permissão do arquivo criado antes do comando umask

permissão do diretório criado antes do comando umask

permissão do arquivo criado depois do comando umask

permissão do diretório criado depois do comando umask



Permissões - Comando umask

- O valor umask pode ser encontrado e configurado no /etc/profile ou /etc/bash.bashrc;
- Para definir as permissões com o umask é possível utilizar a notação alfabética;
 - Exemplo do comando: `umask u+rw.`

umask notação alfabética

```
~$ umask ugo+w
~$ touch arquivo.txt
~$ mkdir pasta
~$ ls -l
total 1
-rw-rw-rw- 1 user user 0 Mar  3 22:50 arquivo.txt
drwxrwxrwx 2 user user 2 Mar  3 22:50 pasta
~$ umask ugo-wx
~$ touch arquivo2.txt
~$ mkdir pasta2
~$ ls -l
total 2
-rw-rw-rw- 1 user user 0 Mar  3 22:50 arquivo.txt
-r--r--r-- 1 user user 0 Mar  3 22:51 arquivo2.txt
drwxrwxrwx 2 user user 2 Mar  3 22:50 pasta
dr--r--r-- 2 user user 2 Mar  3 22:52 pasta2
```



Permissões - Comando umask

- O valor umask pode ser encontrado e configurado no /etc/profile ou /etc/bash.bashrc;
- Para definir as permissões com o umask é possível utilizar a notação alfabética;
 - Exemplo do comando: `umask u+rw.`

umask notação alfabética

```
~$ umask ugo+w
~$ touch arquivo.txt
~$ mkdir pasta
~$ ls -l
total 1
-rw-rw-rw- 1 user user 0 Mar  3 22:50 arquivo.txt
drwxrwxrwx 2 user user 2 Mar  3 22:50 pasta
~$ umask ugo-wx
~$ touch arquivo2.txt
~$ mkdir pasta2
~$ ls -l
total 2
-rw-rw-rw- 1 user user 0 Mar  3 22:50 arquivo.txt
-r--r--r-- 1 user user 0 Mar  3 22:51 arquivo2.txt
drwxrwxrwx 2 user user 2 Mar  3 22:50 pasta
dr--r--r-- 2 user user 2 Mar  3 22:52 pasta2
```



Comando de Permissão de Acesso - chown

- O comando chown permite alterar o proprietário e grupo no qual está associado um arquivo ou diretório;
- Sintaxe do comando:
 - Alterar proprietário: `chown <usuário> <arquivo>`
 - Alterar proprietário e grupo: `chown <usuário>:<grupo> <arquivo>`

```
root@socps-pc:~/aula# ls -l
total 4
-rw-r--r-- 1 root root 14 Mar  4 09:28 arquivo.txt
root@socps-pc:~/aula# chown projeto:projeto arquivo.txt
root@socps-pc:~/aula# ls -l
total 4
-rw-r--r-- 1 projeto projeto 14 Mar  4 09:28 arquivo.txt
```



Gerenciamento de arquivos e diretórios - Linux

- `cp` – permite realizar uma cópia do arquivo;
- `mkdir` – permite criar um novo diretório;
- `mv` – permite mover ou renomear um arquivo ou diretório;
- `rm` – permite remover um arquivo ou diretório;
 - Para diretório é necessário aplicar o parâmetro `-R`;
- `rmdir` – permite remover um diretório
- `touch` – comando que permite criar um arquivo vazio;



Manipulando arquivos - Linux

- `cat` – permite ler uma arquivo
- `tail` – mostrar as últimas linhas de um arquivo;
 - Ainda é possível definir a quantidade de linhas utilizando o parâmetro `-n`;
- `|` – o pipe (`|`) permite usar dois ou mais comandos, de forma que a saída de um comando sirva como entrada para o próximo.
 - Ex.: `ls | more`
- `less` – permite fazer paginação de arquivos, pode ser utilizado para ler arquivos que ocupam mais de uma tela;
- `zless` – tem a mesma função do comando `less`, porém permite a visualização de arquivos compactados (`.gzip` e `gz`);
- `more` – permite pausar uma apresentação que ocupa mais que uma tela;
 - Não possui retorno na página que foi avançada.



Atividade – Manipulando Arquivos

Atividade Desafios

- Utilize o comando `cat` para concatenar o conteúdo de três arquivos diferentes para um novo arquivo;
- Inverta a ordem de leitura das linhas do arquivo, utilize o comando `tac`, crie um arquivo na ordem inversa;
- Insira conteúdo em um arquivo sem utilizar nenhum editor de texto (nano, vim, gedit, vi), utilize o comando `echo`;
- Crie um novo arquivo no home do usuário a partir das 3 primeiras e 3 últimas linhas do arquivo `/var/log/syslog`
 - Utilizar o comando `tail` e `head`;
 - Tente executar o comando em uma única linha (utilize o `&&`);



Encontrando Arquivos

- **find** – permite encontrar um arquivo informado em seu sistema de arquivo;
 - A procura por padrão é iniciada pelo diretório raiz “/”
 - Na sintaxe do comando é possível utilizar coringas (*,?) no nome do arquivo;
 - Ex.: `find /etc/ -name sources.list` Ex.: `find / -user “projeto” | more`
- **locate** – permite encontrar um arquivo que contenha uma *string* de busca;
 - Ex.: `locate firefox`
- **whereis** – exhibe os locais onde se encontram os arquivos executáveis, man pages, arquivos de configuração e fontes de determinado comando;
 - Ex.: `whereis firefox`
- **which** – exhibe o caminho completo até o executável que seria rodado se o nome dele fosse digitado no terminal – consulta a partir do PATH;
 - Ex.: `which firefox`



Encontrando Arquivos – Comando grep

- **grep** – o comando processa o texto linha por linha e imprime todas as linhas que correspondem ao padrão especificado;
 - O **grep** procura por um padrão especificado em uma *string* dentro de um arquivos texto;
 - Sintaxe: **grep** <argumentos> <string> <arquivo>
 - Ex.: **grep -n "href" index.html**
 - Parâmetros:
 - **s** – mensagens de erro são suprimidas;
 - **i** – as diferenças entre letras maiúsculas e minúsculas são suprimidas;
 - **v** – obtém todas as linhas que não são iguais ao parâmetro de busca;
 - **c** – retorna a ocorrências de número de linhas obtido na busca;
 - **l** – retorna apenas o nome do arquivo com o padrão especificado (coringas *);
 - **n** – apresenta o número da linha onde foi localizado o padrão de busca.



Criando Links Simbólicos

- `ln` – permite criar um link simbólico;
- Um “link” simbólico especifica um caminho para localizar o arquivo real;
- Utilizando link simbólicos é possível criar referências circulares, por exemplo “a” aponta para “b” e “b” aponta de volta para “a”;
 - Sintaxe: `ln -s <origem> <destino>`
 - Ex.: `ln -s /home/projeto/programa.v1.0.0/ programa`
- `symlinks` – verifica e corrige links simbólicos no sistema;
 - Atualmente não está vindo por padrão nas distribuições mais recentes;
 - Para instalar utilize o comando: `apt-get install symlinks`
 - Sintaxe: `symlinks <parâmetros> <caminho>`
 - Ex.: `symlinks -r -cds /home/projeto`



Atividade – Links Simbólicos

- Criar e manipular links simbólicos – comando `ln`:
 - Crie um arquivo executável para imprimir a mensagem “Utilizando link simbólico!!”;
 - O arquivo deve ser gerado a partir do compilador `gcc` utilizando a `linguagem de programação c`;
- Adicionar o arquivo criado no diretório;
 - Fora do diretório criar um link simbólico apontando para o arquivo executável;
 - Dentro do diretório crie outro link simbólico apontando para o link simbólico fora da pasta;
 - Criando uma referência circular;
 - Execute o link simbólico externo e depois o link criado dentro da pasta;
 - Dentro da pasta crie um link simbólico para um local inexistente;
 - Fora da pasta utilize o comando `syslinks`.
 - Primeiramente execute o comando apenas com o `parâmetro -r` e depois com `-r -cds`;
 - Analise cada um dos resultados.



Configuração do hardware

- No Linux existe uma nomenclatura padrão para cada um dos periféricos utilizados:
 - **hda**: primeiro disco rígido instalado na máquina (*master*);
 - **hdaX**: partição **X** do primeiro disco rígido instalado;
 - **hdb**: segundo disco rígido instalado na máquina (*slave*);
 - **hdbX**: partição **X** do segundo disco rígido instalado;
 - **fd0**: drive de disquete (**por curiosidade**);
 - **cdrom**: drive de CD-ROM instalado;
 - **OBS.: X** – número da partição;
- Os arquivos que configuram determinado hardware ficam no diretório /dev.



Tipos de sistema de arquivos

- Os tipos de sistemas de arquivos mais utilizados são:
- **ext2, ext3, ext4, XFS e btrfs** – sistema operacional Linux;
- **NTFS** – sistema operacional Windows (versões a partir do Windows 2000);
- Família FAT – sistema operacional Windows (compatível com MSDOS);
 - **FAT12, FAT16, FAT32 e VFAT**;
 - Limite de tamanho de arquivo 4GB;
 - VFAT permite utilização de nomes longos;
- **iso9660** – para montar o CD-ROM;
- **HFS, APFS e UFS** – sistemas de arquivos Mac OS.



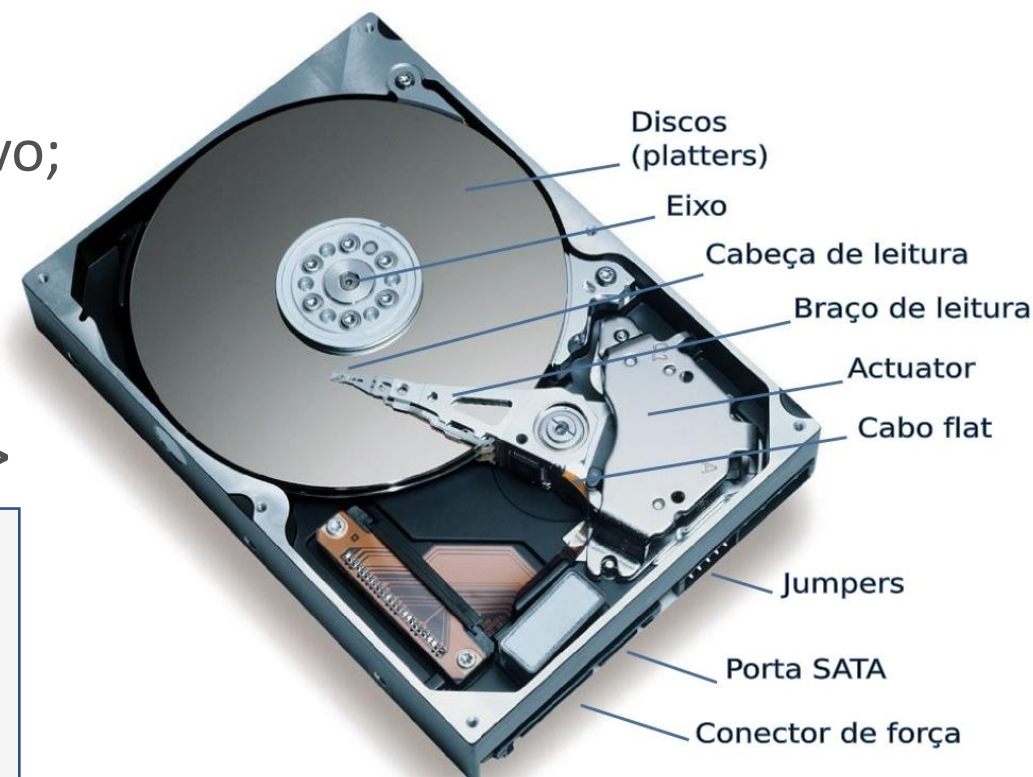
Manipular os sistemas de arquivos

- `cat /proc/filesystem`: informação dos tipos de sistemas de arquivos que podem ser utilizados na distribuição;
- `mount` : permite montar um sistema de arquivo;
 - `mount -t <tipo> <dispositivo> <diretório>`
- `umount`: desmontar o sistema de arquivos.
 - `umount <dispositivo>` ou `umount <diretório>`

Comandos:

```
mount -t ext2 /dev/sda1 /mnt/disco
```

```
umount /mnt/disco
```



Utilitários de Gerenciamento de Disco - gparted

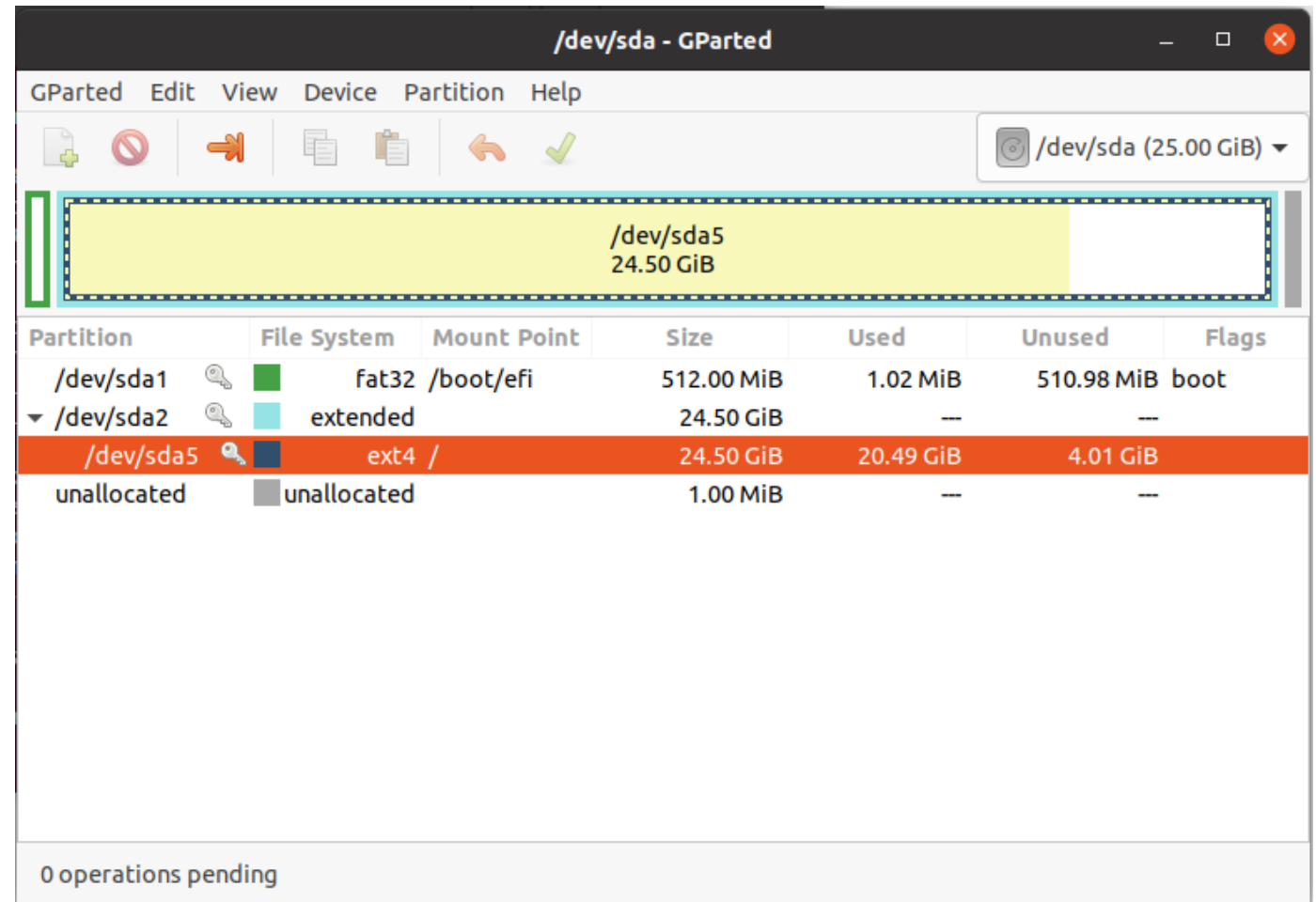
Utilitário de disco:

Comandos:

```
sudo apt-get install gparted  
sudo gparted
```

Alternativa para formatar:

```
mkfs.ext4 -L pendrive /dev/sdb3
```



Montar Disco na Inicialização - fstab

- Arquivo de inicialização do disco: `cat /etc/fstab`

```
root@ubuntu: /home/jhonatan
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda5 during installation
UUID=f98f5dd1-f305-4d15-8196-d504cac08851 /                ext4      errors=remount-ro 0      1
# /boot/efi was on /dev/sda1 during installation
UUID=C535-1448 /boot/efi          vfat      umask=0077      0      1
/swapfile                                none      swap        sw          0      0
/dev/fd0 /media/floppy0    auto      rw,user,noauto,exec,utf8 0      0
```



Verificar os Discos Montados

- `df -h`: verificar os dispositivos montados.

```
root@ubuntu:/home/jhonatan# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G   0    1.9G   0% /dev
tmpfs           391M  1.8M  390M   1% /run
/dev/sda5       24G   20G   2.8G  88% /
tmpfs           2.0G   0    2.0G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           2.0G   0    2.0G   0% /sys/fs/cgroup
/dev/loop0       56M   56M     0 100% /snap/core18/1988
/dev/loop4       63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop3      219M  219M     0 100% /snap/gnome-3-34-1804/66
/dev/loop2      218M  218M     0 100% /snap/gnome-3-34-1804/60
/dev/loop5       65M   65M     0 100% /snap/gtk-common-themes/1514
/dev/loop6       50M   50M     0 100% /snap/snap-store/467
/dev/loop7       52M   52M     0 100% /snap/snap-store/518
/dev/loop1       56M   56M     0 100% /snap/core18/1944
/dev/loop8       32M   32M     0 100% /snap/snapd/10492
/dev/loop9       32M   32M     0 100% /snap/snapd/11036
/dev/sda1       511M  4.0K  511M   1% /boot/efi
tmpfs           391M  64K  391M   1% /run/user/1000
root@ubuntu:/home/jhonatan#
```



Manipular hard disco – identificação do disco

- `blkid`: verificar a identificação do bloco do disco:

```
root@ubuntu: /home/jhonatan

root@ubuntu:/home/jhonatan# blkid
/dev/sda5: UUID="f98f5dd1-f305-4d15-8196-d504cac08851" TYPE="ext4" PARTUUID="16f8dfea-05"
/dev/loop0: TYPE="squashfs"
/dev/loop1: TYPE="squashfs"
/dev/loop2: TYPE="squashfs"
/dev/loop3: TYPE="squashfs"
/dev/loop4: TYPE="squashfs"
/dev/loop5: TYPE="squashfs"
/dev/loop6: TYPE="squashfs"
/dev/loop7: TYPE="squashfs"
/dev/sda1: UUID="C535-1448" TYPE="vfat" PARTUUID="16f8dfea-01"
/dev/loop8: TYPE="squashfs"
/dev/loop9: TYPE="squashfs"
root@ubuntu:/home/jhonatan#
```



Montar um dispositivo na inicialização

1. Adicionar novo dispositivo (hard disk);
2. Formatar disco: o utilitário de disco `gparted` é uma boa opção;
3. `blkid`: identificação do bloco do dispositivo;
4. Adicionar entrada arquivo fstab: `/etc/fstab`
 - Exemplo:
 - `UUID=958ae88d-44f3-4160-8024-f250d053714c /disco2 ext4 defaults 1 2`
5. Reiniciar a máquina e verificar se o disco foi adicionado: `df -h`



Utilitários de Disco

- **fdisk**: utilitário para particionamento de disco – **fdisk /dev/dispositivo**;
- **sfdisk**: lista tabelas de partição – inclusive partições estendidas;
- **cfdisk**: utilitário de particionamento de disco, baseado em menus;
- **parted**: utilitário para manipulação de partição (ext2 e DOS);
- **badblocks**: verifica se o disco ou disquete possui blocos danificados;
- **fsck**: permite verificar e reparar o sistema de arquivos – **fsck -t <tipo> <partição>**;
 - **fsck -t ext2 /dev/sda2**;
- **dd**: comando clonar o disco – **dd if=<origem> of=<destino>**





Obrigado!

Jhonatan Geremias

Jhonatan.geremias@pucpr.br

