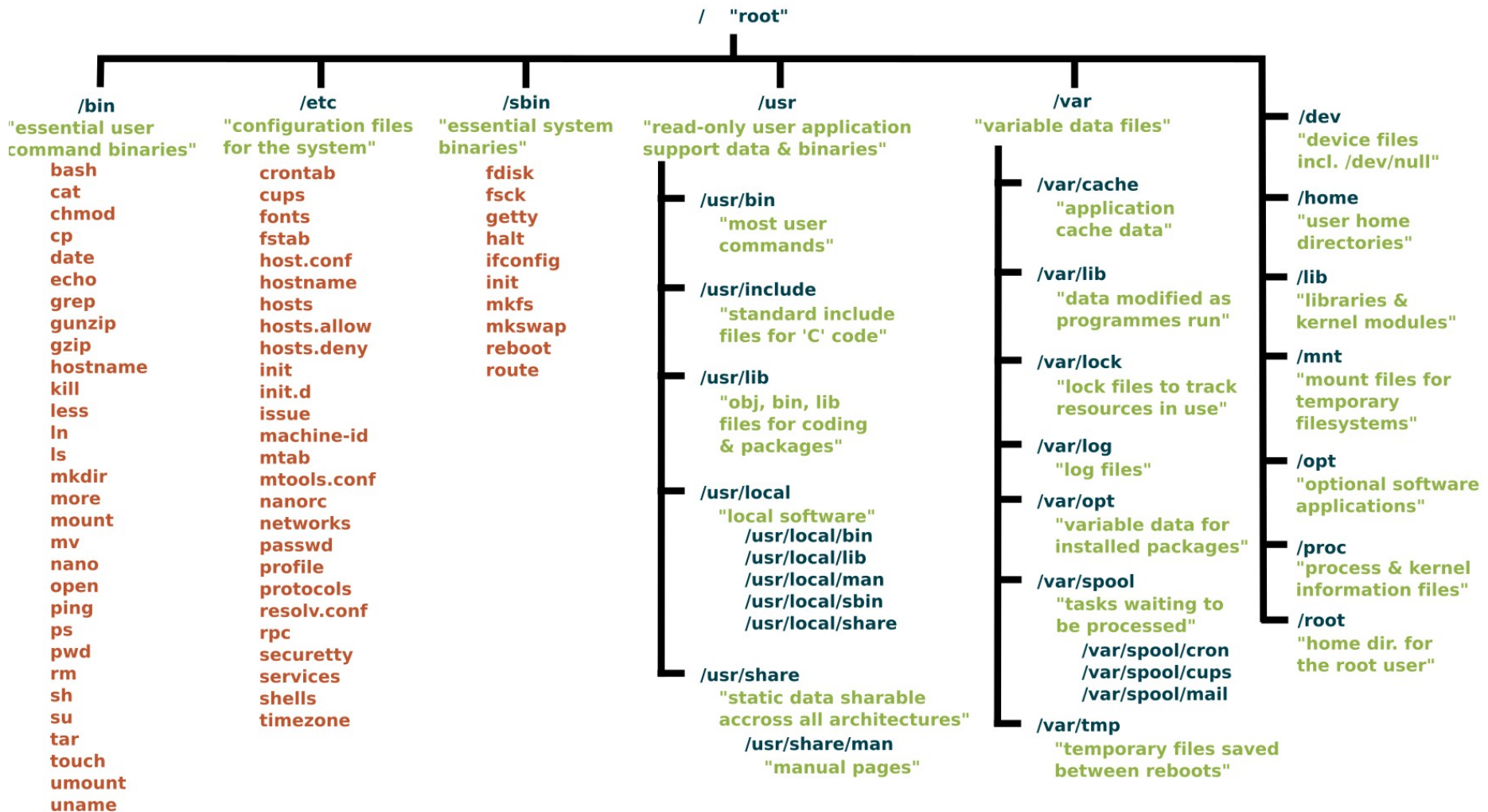


Essencial da Linha de Comando Linux

Sistema de Ficheiros (Exemplo)



Sistema de Ficheiros

Diretoria raiz/root (/)

A diretoria raiz / é o ponto de partida para toda a árvore hierárquica do sistema de ficheiros Linux

Subdiretórias

/bin	executáveis binários
/boot	ficheiros associados ao arranque do sistema
/dev	dispositivos ligados (usb, cdrom, rato, teclado)
/etc	ficheiros de configuração
/home	diretórias pessoais de cada conta de utilizador
/lib	bibliotecas de sistema
/media	montagem de dispositivos amovíveis (disquetes, cdrom)
/mnt	montagem de sistemas de ficheiros (nfs, smb)

Subdiretórias

/opt	software complementar
/proc	informação de processos/recursos do sistema
/root	diretoria inicial do administrador
/run	sistema de arquivos temporário para dados de execução
/sbin	executáveis binários utilizados pelo administrador
/srv	dados para serviços de servidor
/sys	sistema de ficheiros virtual para informações de hardware/drivers
/tmp	ficheiros temporários (a eliminar na reinicialização)
/usr	utilitários e dados/programas de utilizador de leitura apenas
/var	dados variáveis do sistema (logs, cache, filas, estados)

Linha de Comandos

Iniciar a linha de comandos: **Ctrl+Alt+T**

Não é necessário quando não temos GUI :)

~\$

nome da diretoria atual

~ significa a diretoria pessoal do utilizador

Ctrl+C, exit

\$ ^C

Ctrl+C (representado por ^C) é usado para parar o processo que está a correr

\$ **exit**

termina a sessão atual e fecha a consola se não está aberta mais nenhuma sessão

Formato Genérico dos Comandos

comando **opções** **argumentos**

comando: nome do comando a executar ou caminho para o comando, no sistema de ficheiros

opções: afetam o comportamento do comando. Opções mono-carácter são usualmente precedidas de **-**, com possibilidade de combinar as várias opções:

comando **-a** **-b** **-c** ou **comando** **-abc**

Formato Genérico dos Comandos

comando opções argumentos

opções multi-carácter são usualmente precedidas de **--**

comando **--opção**

argumentos: itens (dados) necessários ao comando

Para correr vários comandos, usar **;** para os separar:

ls **-la** /tmp **;** **ls** **-l** **--all**

man

```
$ ls --help
```

muitos comandos suportam a opção **--help**, que mostra uma descrição resumida da sintaxe

```
$ man ls
```

mostra o manual completo de comando para o comando "**ls**" (o manual é o local padrão para a documentação!)



Caminhos

```
$ cd /tmp
```

```
$ cd tmp
```

Caminho absoluto: começa com "/"

Caminho relativo: não

pwd

\$ **pwd**

mostra o caminho absoluto para a diretoria
corrente/de trabalho

cd

```
$ cd dir
```

define `dir` como nova diretoria de trabalho, sendo `dir` um caminho para a nova diretoria de trabalho

```
$ cd
```

define a nossa *home directory* como diretoria de trabalho

```
$ cd . .
```

"`..`" significa "a diretoria acima desta"

```
$ cd . ; pwd
```

"`.`" significa "esta diretoria"

ls

\$ **ls**

mostra/lista informação sobre objetos da diretoria atual

\$ **ls -la**

l -> listagem dos objetos em formato longo

a -> lista todos os nomes, incluindo os objetos ocultos (iniciados por um ponto ".")

mkdir

```
$ mkdir mydir
```

cria diretorias (vazias)

```
$ mkdir -p mydir/mysubdir
```

cria uma cadeia de diretorias, criando as intermédias se necessário

touch, nano, cat

```
$ touch myfile
```

podemos usar o **touch** para criar um ficheiro vazio

```
$ nano myfile
```

podemos editar **myfile** com o editor **nano**
(**myfile** é criado, caso não exista)

```
$ cat myfile
```

mostra o conteúdo de **myfile**

Edição de código, compilação e execução

```
$ nano hello.c
```

editar

```
$ gcc hello.c -o hello
```

compilar

```
$ ./hello
```

executar

cp

```
$ cp myfile mycopy
```

copiar ficheiros e diretorias

```
$ cp -l myfile myhl
```

criar um *hard link* ao invés de uma cópia

```
$ cp -s myfile mysl
```

criar um link simbólico

cp

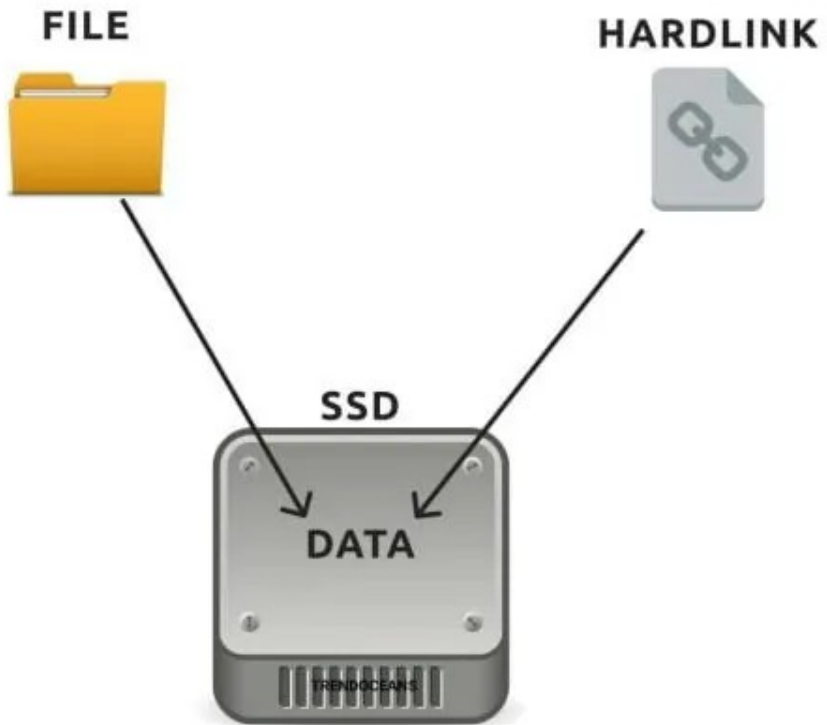
Uma **cópia** é um novo ficheiro independente, com os seus próprios dados

Um **hard link** aponta para os mesmos dados do ficheiro original

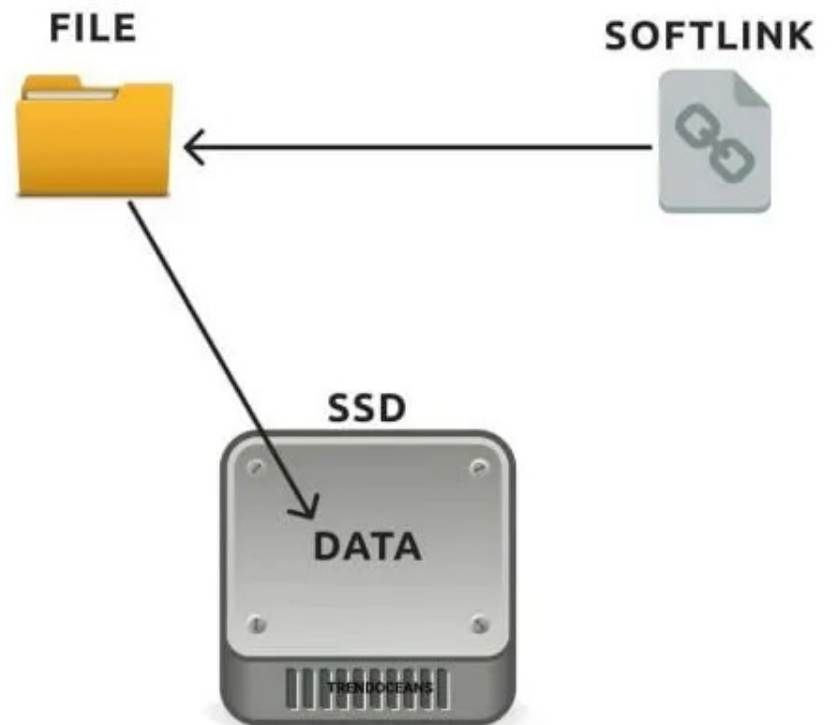
Um **soft link** aponta para o caminho do ficheiro e é quebrado se o ficheiro original for eliminado

cp

HARD LINK



SOFT LINK



mv

```
$ mv origin destination
```

mover ficheiros

rm, rmdir, rm -r

```
$ rm myfile
```

remover ficheiros

```
$ rmdir mydir
```

remover diretorias vazias

```
$ rm -r mydir
```

remover diretorias e os seus conteúdos
recursivamente

who am i

\$ **whoami**

mostra o nome de utilizador

\$ **id**

imprime IDs de utilizadores e grupos

\$ **groups**

imprimir os grupos a que o utilizador pertence

who

\$ who

mostrar quem está logado

W

\$ W

mostrar quem está logado e o que está a
fazer

sudo

\$ **sudo** **command**

executa o comando como utilizador
root/super-utilizador

\$ **sudo** **bash**

abre um novo shell com privilégios de root -
mais poder, maior responsabilidade

Nota: Se o sudo não está instalado, será necessário fazê-lo! Ver:

<https://www.cyberciti.biz/faq/how-to-install-and-configure-sudo-on-debian-linux/>

apt install

```
$ su root
```

mudar para utilizador root

```
# apt install sudo
```

instalar o sudo

```
# nano /etc/sudoers
```

adicionar

```
student ALL=(ALL:ALL) ALL
```

gravar e sair

ls -l

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

- Tipo de ficheiro e permissões
- Contagem de links/atalhos
- Utilizador
- Grupo
- Tamanho
- Data da última alteração
- Nome do ficheiro

Tipo de Ficheiro

```
$ ls -l
```

```
-rw-r--r-- 1 user1  group 573 Sep 24 22:07 myfile
```

-	Ficheiro normal
d	Diretoria
l	Atalho

Permissões

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

r read/leitura

w write/escrita

x execute/execução

- a permissão está desabilitada
para este campo

Nove definições de permissão

Permissões

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

Três conjuntos de três

Permissões

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

Primeiro conjunto: proprietário/owner

read	yes
write	yes
execute	yes

Permissões

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

Segundo conjunto: grupo/group

read	yes
write	no
execute	yes

Permissões

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

Terceiro conjunto: todos os outros. Neste exemplo, temos:

read	yes
write	no
execute	yes

chmod

```
$ chmod XXX file
```

definir permissões para um ficheiro. XXX são três dígitos octais para permissões de proprietário, grupo e outros

```
$ chmod -R XXX dir
```

altera recursivamente as permissões de uma diretoria. -R aplica as permissões recursivamente a todos os ficheiros e subdiretorias

Nota: pode ser preciso o comando sudo se não tivermos permissão para modificar o ficheiro/diretoria.

Links/Atalhos

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

Número de links/atalhos.

Utilizador

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

O proprietário do ficheiro.

Grupo

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

O grupo a que pertence o proprietário do ficheiro.

Tamanho

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

O tamanho deste ficheiro (em bytes).

Data da Última Alteração

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

A última vez que o ficheiro foi alterado.

Nome

```
$ ls -l
```

```
-rwxr-xr-x 1 user1 group 573 Sep 24 22:07 myfile
```

O nome do ficheiro.

date

\$ date

Mon Sep 23 12:29:21 AM WEST 2024

echo

```
$ echo hi
```

```
hi
```

Redirecionamento de Saída

```
$ date > mydate
```

```
$ cat mydate
```

Acrescentar

```
$ date >> log.txt
```

```
$ date >> log.txt
```

```
$ echo "End of log!" >> log.txt
```

```
$ cat log.txt
```

Útil para criar ficheiros de log, por exemplo

Pipes

```
$ ls -l | wc -l
```

Os pipes ligam a saída de um comando à entrada do comando seguinte

wc -l : conta o número de linhas de um ficheiro

Monitorização de Processos com o htop

- Instalar: `sudo apt install -y htop`
- Correr: `htop`
- Recursos: interativo, utilização por CPU/núcleo, gestão de processos
- Alternar entre consolas virtuais (ttys):
 - `Ctrl + Alt + F1` → tty1
 - `Ctrl + Alt + F2` → tty2
 - ... até tty6 na maioria dos sistemas

Recursos Linux - Livro

The Linux Command Line, William Shotts,
Fifth Internet Edition, 2019

Informação Útil

Configuração básica do teclado (Kernel e X)

https://wiki.debian.org/Keyboard#Basic_keyboard_configuration_.28Kernel_and_X.29

Como alterar o tamanho da fonte da consola

<https://www.baeldung.com/linux/font-change-virtual-console>

Exercício

- Consultar o manual de **mkdir**
- Criar as diretorias **OS/Lesson2**
- Mudar-se para a diretoria **Lesson2**
- Criar o ficheiro **hello.c** (usando o nano)
- Inprimir **"Hello World!\n"** em C
- Compilar e executar
- Apagar a diretoria **OS** e o seu conteúdo

Solução

- Consultar o manual de mkdir
man mkdir
- Criar as diretorias OS/Lesson2
mkdir -p OS/Lesson2
- Mudar-se para a diretoria Lesson2
cd OS/Lesson2
- Criar o ficheiro hello.c (use o nano)
nano hello.c

Solução

- Imprimir "Hello World!\n" em C

```
#include <stdio.h>
```

```
int main() {
```

```
    printf("Hello World!\n");
```

```
    return 0;
```

```
}
```

- Compilar e executar

```
gcc hello.c -o hello
```

```
./hello
```

Solução

- Apagar a diretoria OS e o seu conteúdo

```
cd ../..
```

```
rm -r OS
```