

Ficha 1 – Apresentação do sistema operativo Linux

O Linux na perspetiva do utilizador

Tópicos abordados:

- Apresentação do Sistema Operativo
- Acesso à área de trabalho
- Introdução à shell
- Identidade do utilizador
- Redirecionamento de canais padrão
- *Pipes*
- Organização da informação (sistema de ficheiros)
- Sistema de ajuda
- Comandos principais
- Ligações (*hard & soft links*)
- Controlo de processos
- Outros comandos
- Exercícios

Duração prevista: 2 aulas

©2020: {vitor.carreira, patricio.domingues, leonel.santos}@ipleiria.pt

1 Apresentação do Sistema Operativo

O Linux é um sistema operativo (SO), variante do UNIX, que é:

- **Multitarefa** - capacidade para a execução “simultânea” de vários programas.
- **Multiutilizador** - capacidade para o suporte “simultâneo” de vários utilizadores.
- **Interativo** – capacidade de aceitar comandos introduzidos pelo utilizador.
- **Portável** – capacidade de ser executado em vários tipos de dispositivos.
- Sistema de ficheiros hierárquico.
- Tratamento uniforme para ficheiros, diretorias e dispositivos de E/S.

Existem largas dezenas de distribuições de Linux. O sítio DistroWatch fornece uma lista não exaustiva das distribuições existentes (<https://distrowatch.com/>). Esta UC faz uso da distribuição “lubuntu”, uma distribuição da família do ubuntu, ela própria derivada da distribuição Debian.

2 Acesso à Área de Trabalho

Identificação

- **login:** nome que identifica o utilizador no sistema.
- **password:** palavra que permite o acesso do utilizador à sua área de trabalho.

O Linux nas aulas práticas de SO

Dado que, as máquinas dos laboratórios têm como software de sistema o Windows, o Linux será usado recorrendo a uma máquina virtual, conseguida pelo uso do software VMWare Player. Por outro lado, uma vez que os laboratórios são partilhados, por vários turnos da UC de sistemas operativos, é conveniente ter uma máquina virtual Linux (Lubuntu) para cada turno.

Assim, para executar o SO Linux terá que efetuar um dos seguintes passos:

1. No seu portátil:
 - a. Copiar o ficheiro compactado da máquina virtual “**Lubuntu1804**” disponível na pasta: **c:\vm** de um computador do laboratório para o portátil e descompactar para uma diretoria à sua escolha **OU, em alternativa**, fazer download diretamente através do link disponibilizado no Moodle.
 - b. Executar o VMWare Workstation Player e abrir o ficheiro Ubuntu.vmx.
2. No PC do laboratório:
 - a. Abrir uma sessão de Windows;
 - b. Fazer apenas uma das opções seguintes (por ordem de importância):
 - i. Caso exista o ficheiro compactado da máquina virtual “**Lubuntu1804**” disponível na pasta: **c:\vm**, fazer uma cópia do mesmo para o Ambiente de Trabalho" ou para "Os Meus Documentos" da sua conta e descompacte-o;
 - ii. Caso contrário, descarregar o ficheiro 7Z a partir do endereço indicado na página moodle da UC
 - c. Executar o VMWare Player e abrir o ficheiro Lubuntu.vmx da diretoria escolhida no ponto b);

NOTA: A máquina virtual lubuntu18.04 tem como utilizador: user e a palavra-passe é: password

2.1 Acesso a uma sessão Linux

Depois de carregada, a imagem do sistema operativo Linux (Lubuntu), apresenta um cenário idêntico ao da figura seguinte.

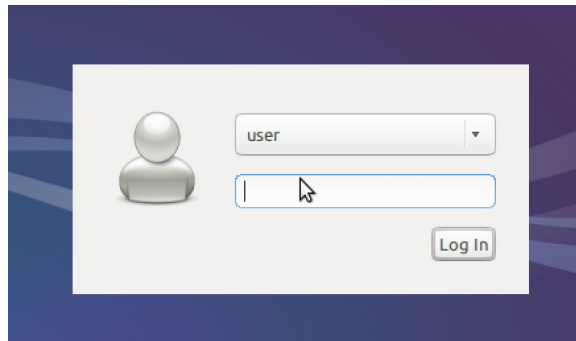


Figura 1 –Pedido de password do utilizador seleccionado (login)

Depois de escolher o utilizador com o qual quer fazer *login*, o sistema pede a palavra-chave. Depois de introduzidas as credenciais de acesso (seleccionar utilizador e escrever palavra-chave) o sistema operativo vai exibir a interface gráfica. Dado que as aulas práticas privilegiam o uso da consola de texto, como modo de interação com o sistema operativo, cada estudante deverá abrir esta interface usando o sistema de menus apresentado na Figura 3. Em alternativa poderá adicionar a consola ao ambiente trabalho clicando com o botão direito do rato sobre a aplicação LXTerminal (ver Figura) ou utilizando o atalho CTRL + ALT + T.

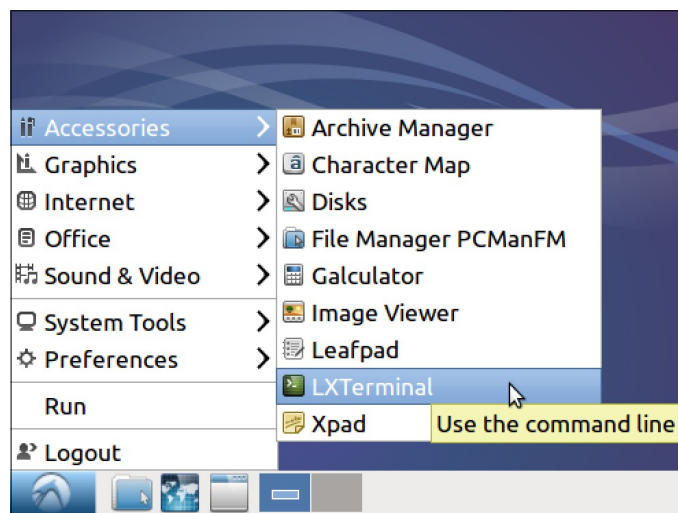


Figura 2 – Interface gráfica e carregamento da consola

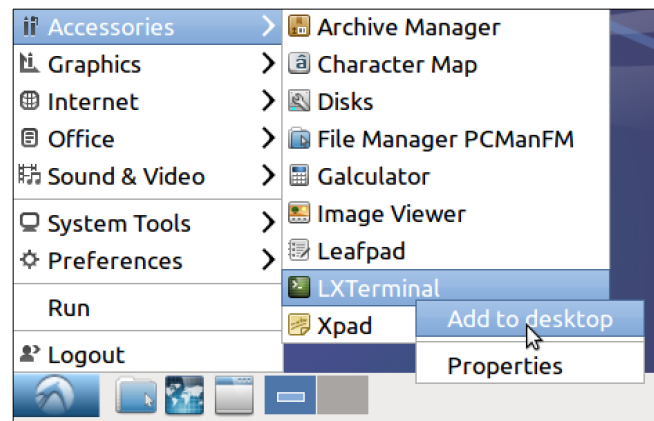


Figura 3 – Adicionar a consola ao ambiente de trabalho

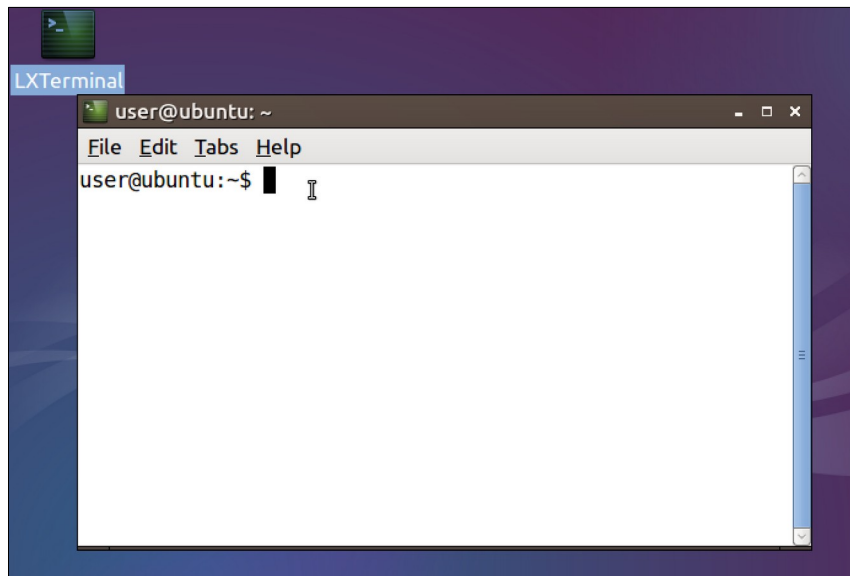


Figura 4 – A consola textual em execução

NOTA: use a combinação de teclas CTRL + L para limpar o conteúdo da consola.

3 Shell

A *shell* (concha) é um programa que aceita comandos introduzidos pelo utilizador e os converte para instruções ao sistema operativo. Por outras palavras, a *shell* é o programa com o qual o utilizador interage. No modo de texto, a *shell* aparece como sendo uma linha de comandos, na qual o utilizador escreve os comandos que pretende executar.

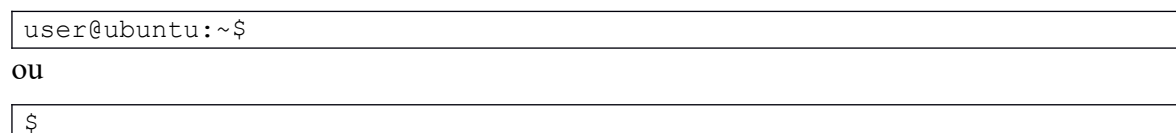


Figura 5 – Exemplos da linha de comandos do Linux

Tal como se pode verificar nos exemplos anteriores, na linha em que se poderá introduzir o comando a executar pelo sistema, normalmente aparece uma denominada *prompt* com o seguinte formato:

utilizador@nomeMaquina:diretoriaCorrente\$.

Para executar um comando, basta escrevê-lo na linha de comando, carregando na tecla de ENTER quando estiver totalmente escrito. A primeira palavra de uma linha de comando é sempre o nome do comando a ser executado. Tudo o resto são argumentos para o comando.

```
$ ls
$ ls -al
$ ls -a -l
```

Figura 6 – Exemplos de comandos

Na Figura 6:

- **ls** é o comando, ao passo que **-a** e **-l** são os argumentos (também chamados de parâmetros) da linha de comando.

Alguns utilitários

Existem alguns utilitários característicos do Unix que pode experimentar:

- **cal** - visualizar o calendário de um mês (ou ano) especificado
 - Exemplo: `$cal 3 2028`
- **date** - visualizar a hora corrente do servidor
 - Exemplo: `$date`
 - Saída do comando: `data corrente`
- **ps** - lista processos
 - **ps -u** lista todos os processos do utilizador
 - **ps aux** lista todos os processos do utilizador
- **clear** - limpa o ecrã (em alternativa pode fazer CTRL+L)

Exercício

Execute o seguinte comando, interpretando a saída produzida: `uname -a`

Histórico de comandos

As distribuições do Linux normalmente armazenam a informação referente aos últimos 500 comandos introduzidos por cada utilizador. Poderá ter-se acesso a essa informação:

- comando interno *history*.
- Tecla de seta/cursor para cima, que poderá ser conjugada com as outras teclas de seta para facilitar a edição de linhas de comando.

Completar palavras na linha de comando com o “TAB”

Na *shell*, A tecla TAB permite completar palavras na linha de comando (mecanismo designado de *tab completion*), com base nos ficheiros existentes na diretoria referenciada. Por exemplo, suponha que na diretoria corrente, existem os seguintes ficheiros: `main.c`, `funcoes.c`, `funcoes.h`, `funcoesAux.h`. Caso se pretenda visualizar o conteúdo do ficheiro `main.c`, pode escrever-se na linha de comando:

```
$ cat main.c
```

Contudo, se tivéssemos escrito apenas `cat m`, e carregado na tecla TAB, a *shell* completaria a nossa palavra acrescentando a parte em falta ao “m”, isto é, “ain.c”, dado que na diretoria corrente apenas, aquele nome de ficheiro se inicia por ‘m’. Agora, imagine-se outra linha de comando:

```
$ cat fu
```

Carregando na tecla TAB, levaria ao acréscimo de “ncoes”, sendo que neste caso existem três opções: (`funcoes.c`, `funcoes.h`, `funcoesAux.h`). Neste caso o sistema de completar palavras termina precisamente, quando as duas opções divergem. E se se carregasse novamente na tecla TAB, ser-nos-iam mostradas as várias hipóteses possíveis (neste caso: `funcoes.c`, `funcoes.h`, `funcoesAux.h`).

Outra forma de completar automaticamente as palavras, consiste em carregar duas vezes na tecla TAB. Por exemplo, se a linha de comando apenas teve a letra *p*, a *shell* indica o número de comandos disponíveis (\$PATH), perguntando se nós os queremos visualizar todos.

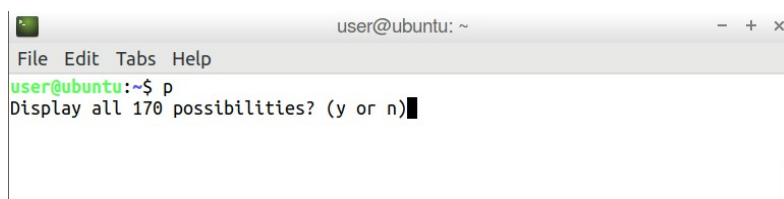


Figura 7 – “Duplo TAB” numa linha de comando vazia

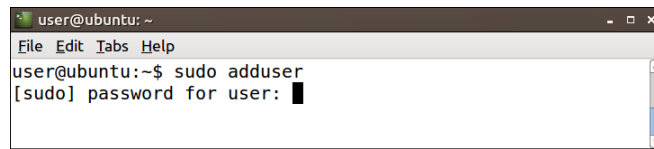
Exercício 1a

Escreva a letra **x**, na linha de comando, e carregue duas vezes em TAB.

3.1 Execução de comandos em modo privilegiado

No Ubuntu, para executar comandos que necessitam privilégios de administração pode utilizar-se o comando **sudo** antes do comando real que se pretende executar. Se o comando, que necessita de privilégios de administração, é o **adduser** (explicado a seguir), então para o poder executar tem que se escrever na linha de comandos:

\$ sudo adduser



```
user@ubuntu: ~  
File Edit Tabs Help  
user@ubuntu:~$ sudo adduser  
[sudo] password for user: █
```

Figura 8 - Execução de comando em modo de administração

É importante referir que para que o comando **adduser** seja executado são necessárias duas condições importantes:

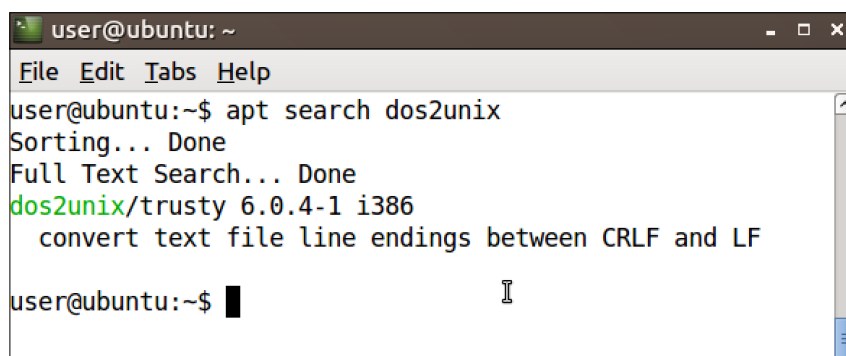
1. A conta atual a ser utilizada pertencer ao grupo dos utilizadores que podem executar o comando sudo (o utilizador USER pertence a este grupo);
user adm cdrom sudo dip plugdev nopasswdlogin lpadmin
sambashare
2. Saber a password do utilizador atual (que tem que pertencer ao grupo sudo).

3.2 Instalação de pacotes de software

A maioria das distribuições Linux permite a instalação de aplicações e utilitários adicionais na forma de pacotes (*packages*). Cada distribuição possui o seu próprio gestor de pacotes que permite instalar, remover e pesquisar pacotes existentes para a distribuição em causa. Cabe também ao gestor de pacotes efetuar a gestão de todas as dependências necessárias para instalar uma dada aplicação/utilitário.

Nos sistemas Ubuntu (Lubuntu incluído), o gestor de pacotes de linha de comando possui o nome “**apt**”. O utilitário “**apt**” disponibiliza várias opções, a destacar:

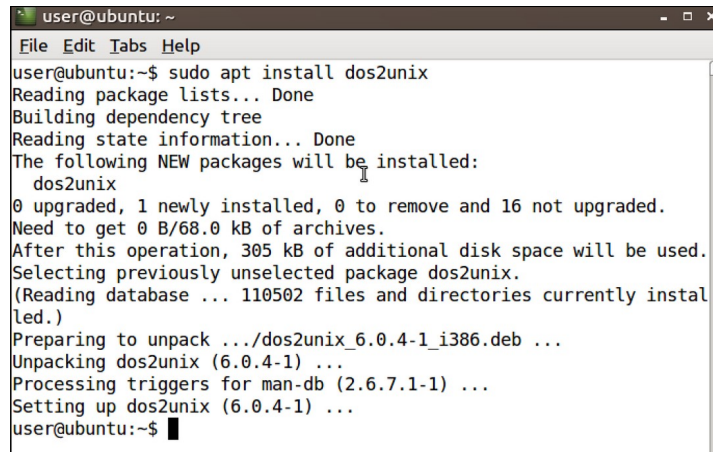
- **apt update** – atualiza a lista de pacotes disponíveis. Esta opção deve ser invocada periodicamente para atualizar a lista de pacotes disponíveis. Requer a execução em modo privilegiado;
- **apt list --installed** – lista os pacotes instalados.
- **apt search <termo>** – permite pesquisar o termo indicado nas descrições da lista de pacotes disponíveis. Exemplo:



```
user@ubuntu: ~  
File Edit Tabs Help  
user@ubuntu:~$ apt search dos2unix  
Sorting... Done  
Full Text Search... Done  
dos2unix/trusty 6.0.4-1 i386  
  convert text file line endings between CRLF and LF  
user@ubuntu:~$ █
```

Figura 9 – Exemplo apt search

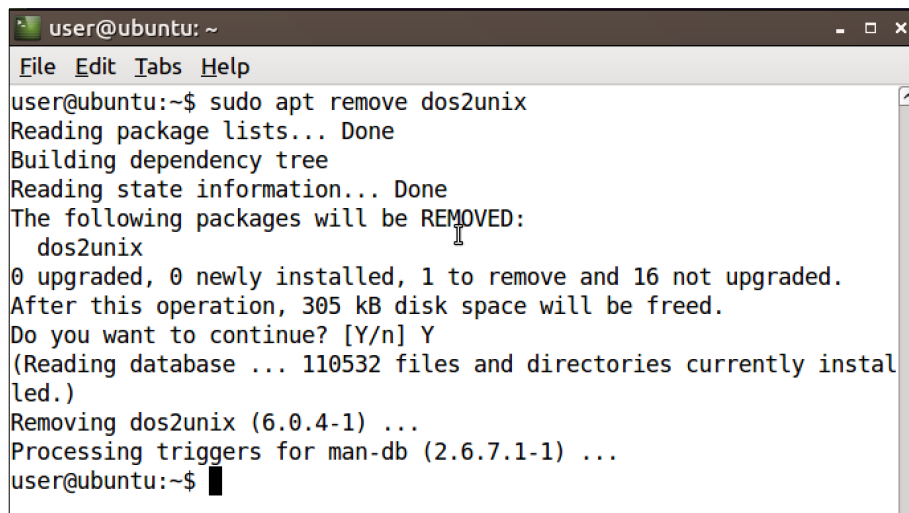
- **apt install <pacote>** – permite instalar um novo pacote e respetivas dependências (ex: **sudo apt install dos2unix**). Requer a execução em modo privilegiado. Exemplo:



```
user@ubuntu: ~  
File Edit Tabs Help  
user@ubuntu:~$ sudo apt install dos2unix  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  dos2unix  
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.  
Need to get 0 B/68.0 kB of archives.  
After this operation, 305 kB of additional disk space will be used.  
Selecting previously unselected package dos2unix.  
(Reading database ... 110502 files and directories currently installed.)  
Preparing to unpack .../dos2unix_6.0.4-1_i386.deb ...  
Unpacking dos2unix (6.0.4-1) ...  
Processing triggers for man-db (2.6.7.1-1) ...  
Setting up dos2unix (6.0.4-1) ...  
user@ubuntu:~$
```

Figura 10 – Exemplo apt install

- **apt remove <pacote>** – permite remover um pacote previamente instalado e todas as dependências que não estejam em uso (ex: **sudo apt remove dos2unix**). Requer a execução em modo privilegiado;



```
user@ubuntu: ~  
File Edit Tabs Help  
user@ubuntu:~$ sudo apt remove dos2unix  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages will be REMOVED:  
  dos2unix  
0 upgraded, 0 newly installed, 1 to remove and 16 not upgraded.  
After this operation, 305 kB disk space will be freed.  
Do you want to continue? [Y/n] Y  
(Reading database ... 110532 files and directories currently installed.)  
Removing dos2unix (6.0.4-1) ...  
Processing triggers for man-db (2.6.7.1-1) ...  
user@ubuntu:~$
```

Figura 11 – Exemplo apt remove

Exercício 1b

- Instale os pacotes *tree*, *curl* e *dos2unix*. O que sucedeu coma instalação do pacote *curl*?
- Execute o comando da linha abaixo, explicando a saída:
`apt show curl`

3.3 Configuração do fuso horário

O fuso horário associado ao sistema operativo pode ser configurado recorrendo ao seguinte comando:

```
sudo dpkg-reconfigure tzdata
```


Exercício

Execute o comando `sudo dpkg-reconfigure tzdata` por forma a configurar o fuso horário do sistema operativo.

3.4 Editores de texto

A máquina virtual disponibilizada possui um editor de texto simples chamado “**Leafpad**” que poderá utilizar para guardar os programas e *scripts* que irá elaborar ao longo da UC. Caso pretenda, poderá instalar outros editores com capacidades mais avançadas como coloração de código baseado na extensão do ficheiro. Deixamos como sugestões o i) gedit e o ii) geany.

4 Identidade do utilizador

- *Login*
- UID – User Identifier (número que identifica um utilizador no sistema)
- GID – Group Identifier (número que identifica o grupo do utilizador no sistema)

As máquinas preferem os números para identificar os utilizadores (é mais eficiente para armazenamento e processamento).

Comandos para acesso à identificação

Para obtermos informação acerca do utilizador e do grupo a que ele pertence, podemos usar os comandos **id** e **groups** (Figura 12).

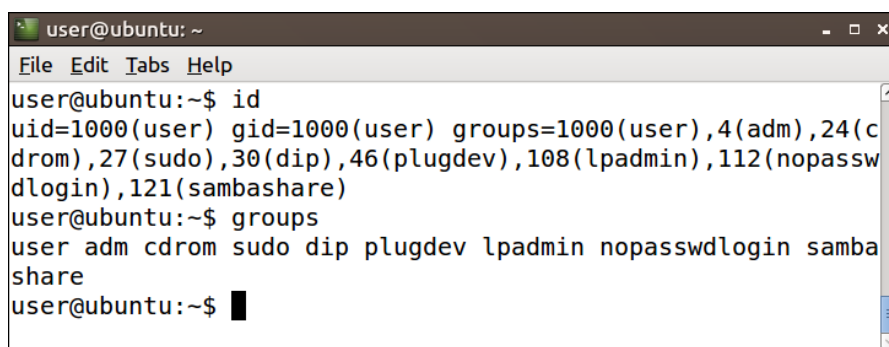
A terminal window titled 'user@ubuntu: ~' with a menu bar (File, Edit, Tabs, Help). The terminal shows the execution of two commands. First, 'id' is run, resulting in the output: 'uid=1000(user) gid=1000(user) groups=1000(user),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lpadmin),112(nopasswdlogin),121(smbashare)'. Then, 'groups' is run, resulting in the output: 'user adm cdrom sudo dip plugdev lpadmin nopasswdlogin samba share'. The prompt returns to 'user@ubuntu:~\$'.

Figura 12 – Comandos id e groups

Na figura anterior, é visível a sigla **gid**, que significa a identificação do grupo primário e serve para definir permissões no sistema de ficheiros, enquanto **groups** define a listagem dos grupos, aos quais o utilizador pertence.

5 Redireccionamento dos canais padrão para ficheiro

Operadores de redireccionamento:

> ou 1>	redireccionamento do stdout
2>	redireccionamento do stderr
>> ou 1>>	redireccionamento com junção do stdout
2>>	redireccionamento com junção do stderr
<	redireccionamento do stdin

2>&1 redirecionamento do stderr para o stdout

1>&2 redirecionamento do stdout para o stderr

&> ficheiro redirecionamento de todas as saídas para ficheiro

NOTA: o operador 2>&1, 1>&2 e &> ficheiro devem ocorrer no final da linha de comando

Exemplos:

```
$ ls > data.txt
```

```
$ ls >> data.txt
```

```
$ mail user1 < mensagem.txt
```

```
$ cat ficheiro1 ficheiro2 > ficheiro3
```

```
$ ls abc /tmp > saida_stdout+stderr.txt 2>&1
```

```
$ ls abc /tmp &> all.txt
```

Exercício *redirecionamento*

- Envie a data do sistema (*date*) para o ficheiro data.txt

6 Pipes (|)

O *pipe* redireciona a saída de um comando para a entrada de outro. No exemplo seguinte, o comando **wc -l** devolve o número de linhas do resultado do comando **ps aux**.

```
$ ps aux | wc
```

```
$ ls -laR | more
```

Quando é empregue a simbologia **|&**, são passadas as saídas **stdout** e **stderr** para o comando seguinte. O exemplo seguinte mostra os metadados de um ficheiro que não existe e do diretório **/tmp**, originando a listagem do diretório corrente no **stdout** e uma mensagem de erro no **stderr**. Essa mensagem é passada ao comando seguinte, neste caso, ao comando **nl**.

```
$ ls -l /tmp nome_nao_existe |& nl
```

```
1      ls: cannot access 'nao_existe.txt': No such file or directory
2      /tmp:
3      total 20
4      -rw----- 1 user user    0 Feb 21 11:13 config-err-mM12rg
5      drwx----- 2 user user 4096 Feb 21 11:13 ssh-4AchVM5FIK6k
```

Exercício *pipes*

a) O que faz o seguinte comando?

```
$ ps -ef | wc -l
```

b) O que faz o seguinte comando?

```
$ ps -ef |& wc -l
```

7 Organização da Informação

Sistema de ficheiros em UNIX

À semelhança de outros sistemas operativos, o Linux está organizado, numa hierarquia em árvore, com diretorias e subdiretorias (Figura 13)

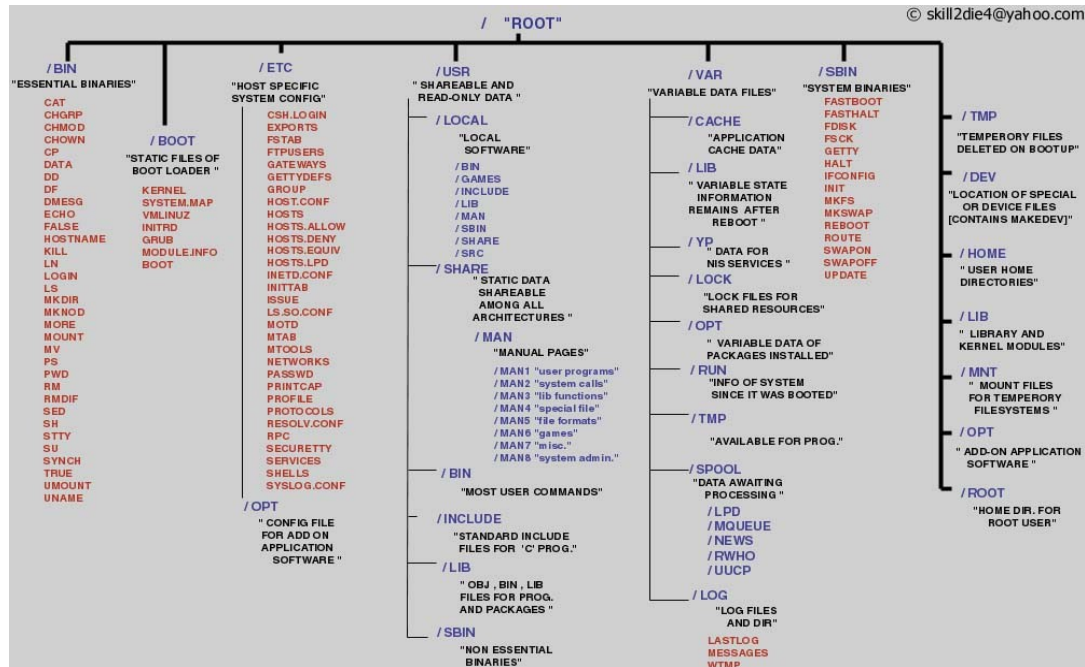


Figura 13 – Hierarquia de diretorias em Linux

Ficheiros e diretorias

7.1 Generalidades

- Os ficheiros são identificados pelo nome.
- O nome dos ficheiros pode conter quaisquer caracteres, com exceção do carácter “/”.
- O número máximo de caracteres para o nome de um ficheiro é 255.
- Associado ao conceito de ficheiro existe o conceito de diretoria, que é um conjunto de ficheiros e/ou subdiretorias.
- Podemos nos referir a um ficheiro através do seu caminho (*path name*), ou seja, o nome do ficheiro e respetiva diretoria.

7.2 Diretoria home

- É a diretoria para a qual o utilizador é levado quando entra no sistema.
- O carácter “~” identifica a diretoria home

7.3 Percorrer a árvore de diretorias

Para mudar de uma diretoria para outra utilize o comando `cd`, (Figura 14), ou seja, *change directory*. Note-se que o carácter de separação de diretorias em Linux é o carácter ‘/’ (*forward slash*) em vez do carácter ‘\’ (*backslash*) do Windows.

```
$ cd dir1
$ cd dir1/dir2
$ cd .. (movimenta-se para a diretoria imediatamente anterior)
$ cd ~ (movimenta-se para a diretoria home)
$ cd - (movimenta-se para a última diretoria conhecida)
```

Figura 14 – Exemplos do comando `cd`

7.4 Noção de pertença de ficheiros e diretorias

Todos os ficheiros e diretorias pertencem a um determinado utilizador. Para obter essa informação utilize o comando `ls`, com as opções `-a`, (lista todos os ficheiros, incluindo os escondidos, cujo nome se inicia por ponto “.”) e `-l` (lista em formato longo), ver Figura 15.

```

user@ubuntu: ~
File Edit Tabs Help
user@ubuntu:~$ ls -l -a /bin
total 12084
drwxr-xr-x  2 root root   4096 Apr 28 2018 .
drwxr-xr-x 22 root root   4096 Aug  1 2018 ..
-rwxr-xr-x  1 root root 1235608 Apr  4 2018 bash
-rwxr-xr-x  3 root root  30384 Jan 29 2017 bunzip2
-rwxr-xr-x  1 root root 1775376 Dec 12 2017 busybox
-rwxr-xr-x  3 root root  30384 Jan 29 2017 bzip2
lrwxrwxrwx  1 root root    6 Jan 29 2017 bzip2 -> bzip2
-rwxr-xr-x  1 root root   2140 Jan 29 2017 bzip2diff
lrwxrwxrwx  1 root root    6 Jan 29 2017 bzip2diff -> bzip2diff
-rwxr-xr-x  1 root root   4877 Jan 29 2017 bzip2grep
lrwxrwxrwx  1 root root    6 Jan 29 2017 bzip2grep -> bzip2grep
-rwxr-xr-x  1 root root   3642 Jan 29 2017 bzip2grep
-rwxr-xr-x  3 root root  30384 Jan 29 2017 bzip2
-rwxr-xr-x  1 root root   9768 Jan 29 2017 bzip2recover
lrwxrwxrwx  1 root root    6 Jan 29 2017 bzless -> bzless
-rwxr-xr-x  1 root root   1297 Jan 29 2017 bzmore

```

Figura 15 – Execução do comando `ls -la` para lista o diretório `/bin`

Interpretação da informação mostrada pelo comando `ls`:

- Coluna 1 – Permissões de utilizador/grupo/outros para o ficheiro/diretoria;
- Coluna 2 – Para diretorias, define o número de diretorias contidas na mesma. Para ficheiros, indica o número de referências para o mesmo;
- Coluna 3 – Identifica o nome do utilizador que é dono do ficheiro/diretoria;
- Coluna 4 – Identifica o nome do grupo que é dono do ficheiro/diretoria;
- Coluna 5 – Ficheiro: Tamanho em bytes | Diretoria: meta-dados;
- Coluna 6 – Data/hora de modificação do ficheiro/diretoria;
- Coluna 7 – Nome do ficheiro/diretoria/ligações.

Exercício 1c – Efetue a listagem da diretoria home (~)

- a) Identifique as linhas que correspondem a diretorias
- b) Identifique os donos e grupos dos ficheiros listados

Manipular diretorias e ficheiros

Comando	Descrição
<code>ls [diretoria ou ficheiro]</code>	Lista o conteúdo de uma diretoria (equivalente ao <code>dir</code>). Se nada for especificado lista a diretoria corrente.
<code>ls -l</code>	Lista com todos os pormenores
<code>ls -la</code>	Lista tudo com todos os pormenores
<code>pwd</code>	Mostra a diretoria corrente
<code>cd [dir]</code>	Muda para a diretoria <code>dir</code> . Se não for especificada a diretoria então muda para a diretoria home. Diretorias especiais: <ul style="list-style-type: none"> <code>.</code> diretoria corrente <code>..</code> diretoria anterior <code>~</code> <i>home directory</i>
<code>mkdir <dir></code>	Cria a diretoria

cp origem destino	Copia o ficheiro de origem para o ficheiro de destino
rm <diretoria ou ficheiro(s)>	Remove o(s) ficheiro(s) especificados. No caso de ser uma diretoria, esta tem de estar vazia. Adicionar -r para remover diretorias e subdiretorias.
cat <nomeFicheiro>	Mostra o conteúdo de um ficheiro
more <nomeFicheiro>	Mostra o conteúdo de um ficheiro passo a passo

Tabela 1- Comandos principais para manipular diretorias e ficheiros

8 Sistema de ajuda

Man (*manual*)

O Linux possui um sistema de ajuda embutido no sistema operativo, que pode ser acedido através do comando **man** (abreviatura de *manual*):

Sintaxe: man [secção] <comando>

Exercício 1d

- Obter ajuda sobre o comando `ls`
- Obter ajuda sobre o comando `man`

Módulo interativo do man – Alguns comandos

Comando	Efeito
<space>	Avança para a próxima página
Q	Termina
B	Recua para a página anterior
<enter>	Mostra a próxima linha
P	Volta para a página inicial, indicando o número de linhas da página
/palavra	Pesquisa “palavra” no texto da página
n	Procura a próxima ocorrência da última pesquisa
b	Procura a ocorrência anterior da última pesquisa
h	Exibe ajuda sobre os comandos interativos da página de manual

Tabela 2 - Alguns Comandos interativos do man

Algumas opções do comando man

Opção	Efeito
-k <palavra>	Pesquisa palavra nos cabeçalhos das páginas de manual, mostrando todos os cabeçalhos onde se encontre a “palavra” Exemplo: man -k passwd
-a <palavra>	Mostra todas as páginas onde se encontre no cabeçalho a “palavra”

Tabela 3 - Algumas opções do man

Exercício 1e -

O que fazem os seguintes comandos?

- a) whatis <palavra>
- b) apropos <palavra>
- c) whereis <comando>
- d) which <comando>

Secções do Manual

O manual encontra-se organizado em secções:

- Secção 1 – Comandos e utilitários
- Secção 2 – Chamadas ao sistema
- Secção 3 – Programação (funções nativas do C, bibliotecas)
- (...)

Por vezes, existem palavras que estão descritas em mais do que uma secção, para aceder à descrição numa secção, esta deve ser especificada.

Sintaxe: man <secção> palavra

Figura 24 – Utilização do comando man

Exercício 1f

Verifique, o resultado da execução dos seguintes comandos:

- a) man printf
- b) man 3 printf
- c) man -k printf

9 Comandos principais

Listagem de diretorias

\$ **ls** [opt]

- a Lista todos os ficheiros de uma diretoria
- F Lista as diretorias com o carácter '/', a seguir ao nome da diretoria e o carácter '*', a seguir ao nome dos ficheiros executáveis.
- l Lista os detalhes dos ficheiros e diretorias
- R Lista o conteúdo de todos os subdiretorias
- t Lista o conteúdo do diretorias por ordem de modificação

Exercício 1g

- a) Efetue a listagem detalhada de todos os ficheiros da diretoria atual
- b) Faça uma listagem recursiva e detalhada da diretoria atual

Criar / remover diretorias

\$ **mkdir** <nome_diretoria>

Cria uma diretoria com o nome "nome_diretoria"

\$ **rmdir** <nome_diretoria>

Remove a diretoria com o nome "nome_diretoria". A diretoria deve estar vazia.

Exercício 1h – a partir da diretoria home (~)

- a) Crie a diretoria **temp**
- b) Dentro da diretoria **temp** crie a diretoria **testes**
- c) Crie a diretoria **trabalhos**
- d) Crie a diretoria **aulas**

Mudar de diretoria

\$ **cd** [diretoria1]

muda para diretoria 1

\$ **cd** [opt]

opt:

- .. Muda para a diretoria ascendente
- ~ Muda para a diretoria **home**

Exercício 1i

- a) Mude para a diretoria **home**
- b) Mude para a diretoria **trabalhos**
- c) Mude para a diretoria ascendente

Copiar ficheiros e diretorias

\$ **cp** [opt] origem destino

Copia o ficheiro origem para destino

\$ **cp** ficheiro1 diretoria1

Copia o ficheiro1 para o diretoria1

\$ **cp -r** diretoria1 diretoria2

Copia o diretoria1 para diretoria2

Exercício 1j

- a) Mude para a diretoria **home**
- b) Copie o ficheiro **/etc/services** para a diretoria **trabalhos**

Remover ficheiros e diretorias

\$ **rm** [opt] ficheiro1

Remove o ficheiro1

\$ **rm -r** diretoria1 (muito cuidado ao utilizar)

Remove recursivamente a diretoria1

\$ **rmdir** diretoria1

Remove a diretoria1 se estiver vazia

Exercício 1k

- a) Mude para a diretoria **home**
- b) Remova a diretoria **temp**

Mover ficheiros e diretorias

\$ **mv** [opt] ficheiro1 ficheiro2

Move ou altera o nome de ficheiro1 para o ficheiro2

\$ **mv** [opt] dir1 dir2

Move ou altera o nome da dir1 para dir2

\$ **mv** [opt] ficheiro1 dir1

Move o ficheiro1 para a dir1

\$ **mv -i** ficheiro1 dir1

Move o ficheiro1 para dir1. Caso exista, em dir1 um ficheiro com o nome ficheiro1, é perguntado ao utilizador o que pretende fazer

Exercício 11

- a) Mude para a diretoria **home**
- b) Mova a diretoria **trabalhos** para dentro da diretoria **aulas**

Permissões

Utilizador	Grupo	outros
r w x	r w x	r w x
4 2 1	4 2 1	4 2 1

Legenda:

r – read, w – write, x – execute
4 – read, 2 – write, 1 – execute

\$ **ls -l**

Mostra as permissões de acesso

\$ **chmod 754** ficheiro1

Altera as permissões para r w x r - x r - -

\$ **chmod u+x** ficheiro1

Dar permissões ao utilizador para executar o ficheiro1

\$ **chmod g+rx** ficheiro2

Dar permissões ao grupo para leitura e execução ao ficheiro2

\$ **chmod o+rx** ficheiro2

Dar permissões aos outros para leitura e execução ao ficheiro2

\$ **chmod a+r** ficheiro3

Dar permissões a todos para leitura ao ficheiro3

\$ **chmod a-r** ficheiro3

Retirar permissões a todos para leitura ao ficheiro3

\$ **chown user** ficheiro4

Alterar o dono do ficheiro4 para o utilizador user

\$ **chgrp grupoA** ficheiro5

Alterar o grupo do ficheiro5 para grupoA

\$ **chown user:grupoA** ficheiro6

Alterar o dono do ficheiro6 para o utilizador user e o grupo para o grupoA

Exercício 1m

- a) Mude para a diretoria **home**
- b) Altere as permissões à diretoria **trabalhos** para: r w x r - x r - -
- c) Dê permissões ao grupo apenas para escrita na diretoria **trabalhos**
- d) Altere o dono e grupo da diretoria **trabalhos** para um utilizador e grupo à sua escolha (exemplo www-data e proxy)

Visualizar conteúdo de ficheiros

\$ **cat** ficheiro1 ficheiro2 ...

Mostra o conteúdo de todos os ficheiros especificados

\$ **more** ficheiro1

Mostra o conteúdo do ficheiro1 página a página

\$ **tail** ficheiro1

Mostra as últimas linhas de um ficheiro

\$ **tail -n xx** ficheiro1

Mostra as últimas xx linhas do ficheiro1

\$ **head** ficheiro1

Mostra as primeiras linhas do ficheiro1

\$ **head -n xx** ficheiro1

Mostra as primeiras xx linhas do ficheiro1

Exercício 1n

- Mostre todo o conteúdo do ficheiro **services** da diretoria **trabalhos**
- Mostre o conteúdo do ficheiro **services** página a página
- Mostre as primeiras 5 linhas do ficheiro **services**
- Mostre as últimas 2 linhas do ficheiro **services**

Conceito de metacaracteres

- **?** (qualquer caracter na posição indicada)
- ***** (qualquer caracter em qualquer posição)
- **[abc...]** (qualquer dos caracteres enumerados)
- **[a-e]** (qualquer dos caracteres na gama indicada)

Exemplo: \$cp *.txt ./FilesTexto

 \$ rm *.*?

10 Ligações

Hard link (ficheiros)

\$ **ln** [opt] <origem> <nome_do_link>

Os nomes origem e nome_do_link apontam para o mesmo ficheiro que obrigatoriamente deve existir.

Soft link (ficheiros e diretorias)

\$ **ln -s** <dir1> <dir2>

É criada uma ligação simbólica de dir2 para dir1. A diretoria dir1 pode não existir no sistema de ficheiros. Neste caso, o comando ls mostra a respetiva ligação, dir2, com fundo a vermelho.

Exercício 1o

- Crie um link de nome services_sombra.txt para o ficheiro **services**
- Crie uma ligação simbólica job para a diretoria **aulas**

11 Controlo de processos

\$ **jobs**

Mostra informações sobre os processos

\$ fg <identificação do processo>

Coloca em foreground o processo especificado.

\$ bg <identificação do processo>

Coloca em background o processo identificado por Identificação do processo. Caso não, seja especificado qualquer processo como argumento, este comando coloca o último processo parado em background.

Exercício 1pa

Execute e verifique o funcionamento dos seguintes comandos:

- a. sleep 5
- b. CTRL + Z
- c. jobs
- d. bg
- e. jobs
- f. fg

O utilitário ps

O utilitário *ps* (process show) serve para obter dados referentes aos processos existentes no sistema. O utilitário *ps* de sistemas linux suporta dois estilos:

- estilo BSD: as opções são especificadas sem o hífen “-” (exemplo: **ps ax**)
- estilo UNIX/Linus: as opções são especificadas com o hífen “-” (exemplo: **ps -ef**)

\$ ps [opt]

Mostra os processos do utilizador

- e mostra todos os processos (UNIX)
- f output com detalhe (UNIX)
- a mostra todos os processos (BSD)
- u output com detalhe (BSD)
- x mostra processos que não associados a um terminal (BSD)

\$ kill -9 <Identificação_processo>

Termina o processo identificado por Identificação_processo

Exercício 1pb

Apresente uma listagem de todos os processos, onde conste também os donos dos mesmos.

Exercício 1pc

Execute e verifique o funcionamento das seguintes linhas de comando:

- a) ps auxf (processos e respetiva dependência hierárquica pai/filho)
- b) ps -Ao pid,tid,cmd,args,psr (formato definido pelo utilizador)

12 Compactar e arquivar

Utilitário tar e algumas das principais opções.

tar

JULIA EVANS
@b0rk

<p>The .tar file format combines many files into one file.</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">a.txt b.txt dir/c.txt</div> <p>.tar files aren't compressed by themselves. Usually you gzip them: .tar.gz or .tgz!</p>	<p>Usually when you use the 'tar' command, you'll run some incantation</p> <p>To unpack a tar.gz, use:</p> <pre>tar -xzf file.tar.gz</pre> <div style="display: flex; align-items: center;"><div style="text-align: center; margin-right: 10px;"><p>what's xzf?</p></div><div style="text-align: center; margin-left: 10px;"><p>let's learn!</p></div></div>	<p>-x is for <u>ex</u>tract</p> <p>into the current directory by default (change with -C)</p>
		<p>-c is for <u>cr</u>eat</p> <p>makes a new tar file!</p>
<p>-t is for list</p> <p>lists the contents of a tar archive</p>	<p>tar can compress / decompress</p> <ul style="list-style-type: none">-z gzip format (.gz)-j bzip2 format (.bz2)-J xz format (.xz) <p>& more! see the man page ☺</p>	<p>putting it together</p> <p><u>list</u> contents of a .tar.<u>bz2</u>:</p> <pre>\$tar -tjxf file.tar.bz2</pre> <p style="text-align: center;"><small>verbose</small></p> <p><u>create</u> a .tar.<u>gz</u></p> <pre>\$tar -czf file.tar.gz dir/</pre> <p style="text-align: center;"><small>files to go in the archive</small></p>
<p>-f is for file</p> <p>which tar file to create or unpack</p>		

13 Outros comandos

\$ **whoami**

Identificação do utilizador

\$ **find** [<caminho>] [<expressão>]

Procura ficheiros e diretorias com o nome especificado

- name <padrão> → mostra os que satisfazem o padrão
- iname <padrão> → como -name mas case insensitive
- type <tipo> → mostra os do tipo: f ficheiros, d diretorias, l links
- ctime [+|-] → ficheiros modificados à mais ou menos n dias
- newer ficheiro → mostra ficheiros modificados mais recentemente que 'ficheiro'

Exemplos: `find / -name testo.txt – print`
`find –iname ‘*txt’`
`find –newer ficheiro`

\$ wc <opt>ficheiro

Devolve o número de linhas, palavras e carateres de um ficheiro

- l devolve o número de linhas do ficheiro
- w devolve o número de palavras do ficheiro
- m devolve o número de carateres do ficheiro
- c devolve o número de bytes do ficheiro

Exemplo: `wc –lwc ficheiro1`

Exercício 1q

- a) Procure a diretoria trabalhos a partir da sua diretoria home
- b) Quantos carateres tem o ficheiro `/etc/resolv.conf`?

\$ w

Identificação do utilizador e das tarefas que está a executar

\$ hostname

Nome da máquina à qual está ligado

\$ which ficheiro

Devolve o caminho absoluto da localização do ficheiro

\$ sleep segundos

Para durante o número de segundos especificados

Exemplo: `sleep 2`

\$ df

Mostra o espaço em disco livre

\$ du

Mostra o espaço em disco usado

\$ history

Mostra os comandos efetuados

Exemplo: `history`

Exemplo: `!30`

\$ tar cvf ./mail.tar ./Mail

O conteúdo da diretoria Mail é organizado, num ficheiro que terá o nome mail.tar.

\$ tar xvf ./mail.tar

O conteúdo do ficheiro mail.tar é restaurado para a informação original.

\$ tar cvfz ./mail.tgz ./Mail

O conteúdo do diretoria Mail é organizado, num ficheiro que terá o nome mail.tar e simultaneamente é comprimido, ação esta efetuada pela opção z.

\$ tar xvfz ./mail.tgz

O conteúdo do ficheiro mail.tar é restaurado para a informação original, sendo descomprimido pela opção z, na situação de ter sido previamente comprimido.

\$ gzip -9 mail.tar

Comprime ao máximo o ficheiro mail.tar, criando o ficheiro mail.tar.gz.

\$ gzip -d mail.tar.gz

Descomprime o ficheiro mail.tar

\$ seq <número>

Mostra um sequência de 1 até ao número (>1)

\$ seq <número_inicial> <número_final>

Mostra uma sequência do número inicial até ao número final

14 Encadeamento de comandos

i) **cmd1; cmd2; cmd3;...**

Os comandos cmd1, cmd2, cmd3 são executados consecutivamente de forma independente. O “;” atua como separador entre os comandos.

ii) **cmd1 || cmd2**

O comando cmd1 é sempre executado. Contudo, cmd2 só é executado se a execução de cmd1 não for bem sucedida.

Exemplo:

a) **ls -l ficheiro_nao_existe || echo "comando 2"**

b) **ls -l || echo "comando 2 is not run"**

iii) `cmd1 && cmd2`

O comando `cmd1` é sempre executado. Contudo, `cmd2` só é executado se a execução de `cmd1` for bem sucedida.

Exemplo:

a) `ps aux && ls /proc`

15 Bibliografia adicional

“The Linux Command Line”, William E. Shotts, Jr. (licença *creative common*) - <http://linuxcommand.org/tlcl.php>, 2019

16 Exercícios (extra aula)

1. Descreva a ação dos seguintes comandos, recorrendo, se necessário ao manual eletrónico (`man`).

- a) `sleep 36 ; echo “Está na hora” &`
- b) `cat horário_1 horário_2 > diário.txt`
- c) `ls -l > ficheiro_1 2 > ficheiro_2`
- d) `find . -name “rede.?” -print`
- e) `who | grep eiso_ | wc -l`
- f) `chmod 655 ~/public_html`
- g) `tar cvf mymail.tar ~/Mail`
- h) `cp -r ./Mail ./lixo`
- i) `rm -r ./lixo`
- j) `ps -aux`
- k) `kill 1990`
- l) `find / -mtime -1 \! -type -d -print`
- m) `cat /etc/inetd.conf | wc -l`
- n) `ps aux | less`

2. Que, comando ou comandos utilizaria para executar, as seguintes ações?

- a) Executar o programa `sleep 60`, em *background* e simultaneamente enviar as mensagens de saída para o ficheiro `/tmp/sleep.log`.
- b) Saber o número de utilizadores da máquina.
- c) Contar o número de processos atualmente a correr, no sistema operativo.