**Comandos da Shell**

**Cd / ->** diretório de arranque

**Ls –la ->** mais informações acerca de cada pasta

**Permissões:**

Ex: drwxr-xr-x

Wxr- permissões para mim (owner)

Xr – permissões para grupo de utilizador a que pertce o owner

x- permissões para todos os outros (x para todos os utilizadores quer dizer que pode ver tudo mas nao pode modificar. W significa que pode modificar)

**ls –laR** -> R significa visita recursiva a todas as pastas

**ls –laR | more** -> pipeline de dois comandos ( a saída de um programa serve de entrada para outro)

**cd~** -> diretório home

**cd / ; ls** -> **(; significa que são executados em sequência**)

**cd ; ls** -> (diretório home)

Como será que a shell sabe o nosso diretório home?

Vai consultar a variável Home ($Home) para ver qual é o diretório.

**ls /usr/bin/gcc\* -la** -> (\* significa qualquer coisa que comece com gcc)

**cd /tmp**

**gedit test11.txt**

**cat test11.txt** -> (mostra conteúdo do ficheiro) (cat vem de concatenar)

**cat -n teste11.txt** -> (numerar as linhas)

**cp test11.txt test12.txt** ->(copia)

**more test12.txt**->(le o ficheiro todo para memoria e mostra)

**less test12.txt** ->(mais eficiente do que o more)

**head test12.txt** ->(ler parte inicial do ficheiro)

**tail test12.txt ->**(ler parte final do ficheiro)

**> operador de redirecionamento**

**cat test\* > file1.txt ->** (concatena os dois ficheiros, todos os ficheiros que começam por test, e mostra os em file1.txt)-Substitui se já existir um file1.txt, destruindo o ficheiro inicial.

**>> operador de append no fim do ficheiro**

**cat test\* >> file1.txt** -> (permite juntar no fim, não substitui)

**cat > file2.txt ->** (cat não tendo argumento permite escrever no ficheiro diretamente sem abrir nenhum editor de texto)

**cat ->** (le o input e faz output disso)

ola

ola

**Como terminar?**

1. Ctr D para terminar entrada (input)
2. Echo $? Vai terminar abruptamente, dando um erro.

**cp file2.txt file3.txt**

**mv file3.txt file4.txt**

**cp -i file2.txt file4.txt->** (avisa se o ficheiro de destino ja existir- caso contrario perdem se ficheiros)

**rm -i file?.txt**

**? (remover todos os ficheiros que comecem por file, tenham um caractere a seguir e acabem em txt).**

**\* significa qualquer um, mas ? significa um caractere.**

**alias cp= ‘cp -i’->** sempre que invocar cp, invoca cp -i

**mkdir->** (Criar subpastas)

**mkdir dir1**

**cp\***

**man cp->** (ver todos os comandos-para sair ctrl D e q)

**tail -F -retry rootdir.txt** -> (ve o que esta no fim do ficheiro)

**ls / -larR > rootdir.txt**

**ps -**> process status

**ver os processos que estão a correr: top**

**chmod->** (change mode- tem a ver com as permissões, para mim, para o grupo e para todos os outros)

**chmod 700 script.sh (este ficheiro passa a ter permissões) :**

**rwe**

7 em octal- 111- permissoes read write execute para owner

0 em octal- 000 - permissões read write execute para group

0 em octal- 000 - permissões read write execute para others

**Para compilar um programa:**

<gcc prog.c -o prog -Wall

*Informações sobre comandos:*

man exit

man 3 exit