



## **Guião de aula**

### **Introdução ao Sistema Operativo Linux**

---

#### **1. Introdução**

O Unix é um sistema operativo criado no início da década de 70. É um sistema multiutilizador desenvolvido na linguagem de programação C. Atualmente existem diversas versões (por exemplo, BSD Unix, System V, HPUX, Solaris).

O Unix é um sistema operativo criado no início da década de 70 por Linus Torvalds. Na verdade, o Unix refere-se ao nome do núcleo do sistema operativo, que é o componente fundamental responsável por controlar o *hardware* e oferecer um conjunto de abstrações, mecanismos e facilidades para sua utilização por programadores e utilizadores. O desenvolvimento do Unix, desde sua origem, deve-se à colaboração de diversos programadores, tendo também a participação de empresas em tempos recentes. Todas as contribuições são abertas e gratuitas, independente se vindas de pessoas ou empresas.

Existem muitas distribuições Linux, sendo algumas das mais conhecidas o Ubuntu, Fedora, Debian, Linux Mint, ArchLinux e OpenSuSE. A escolha da distribuição a ser usada depende do gosto e conhecimento do utilizador e, em certos casos, do uso específico que se pretende.

Tradicionalmente, um sistema operativo Linux oferece uma interface de utilizador baseada na linha de comando. Essa interface, em modo texto, está sempre disponível, sendo apresentada pelo menos durante a inicialização do sistema. Em seus primórdios existia apenas esta interface, porém atualmente existem também interfaces gráficas incorporadas nas distribuições Linux.

#### **2. Exercícios propostos**

Todos os exercícios propostos são para ser realizados através da linha de comando existente no Linux.

2.1. Executar o seguinte comando: `man man`

2.1.1. Como sair do `man`?

2.1.2. Com fazer pesquisa dentro do `man`?

2.2. Criar a diretoria `exerc1` na diretoria `HOME`.

2.2.1. O que é a diretoria `HOME`?

2.3. Mudar a diretoria corrente para `exerc1`

2.4. Criar as diretorias `teste1` e `teste2` dentro da diretoria `exerc1`

2.5. Executar o seguinte comando na diretoria `teste1` (ver secção 5.6 da documentação auxiliar)

```
echo aa > aa; echo bb > bb; echo cc > cc
```

2.6. Copiar o conteúdo da diretoria `teste1` para a diretoria `teste2`

2.7. Copiar a diretoria `teste1` para a diretoria `teste2`

2.8. Mudar a diretoria corrente para a sua diretoria `teste1`

2.9. Alterar as permissões do ficheiro `aa` de modo a apenas o dono ter permissões de leitura e escrita sobre o ficheiro (ver secção 4.4 da documentação auxiliar)

2.10. Executar o seguinte comando

```
echo ls -l > myls
```

2.11. Na diretoria `teste1` executar o comando `mysls` (ver notas de secção 5.1)

2.12. Visualizar os processos com o comando `ps`

2.13. Criar o ficheiro `processos` com a informação sobre os processos pertencentes ao vosso utilizador

2.14. Utilizar o comando `ls` e o comando `wc` para contar o número de ficheiros e diretorias existentes na diretoria raiz (ver secção 5.6).

2.15. Executar o comando `sleep 10; echo 'passaram 10 segundos'`

2.16. Executar o comando `(sleep 130; echo 'passaram 130 segundos')` em background

2.17. Visualizar os processos

2.18. Terminar o processo correspondente à execução do comando `sleep 130`

2.19. Executar o comando `(sleep 250; echo 'passaram 250 segundos')` em background

2.20. Executar o comando `sleep 240`

2.21. Suspender o processo do comando `sleep 240` e passá-lo para background

2.21.1. Qual a diferença entre `Ctrl+C` e `Ctrl+Z`

2.22. Continuar a execução do processo do comando `sleep 240` (em background)

2.23. Visualizar a lista de jobs em execução

2.24. Passar o processo do comando `sleep 240` para foreground.

2.25. Terminar o processo do comando `sleep 240`

2.26. Criar o ficheiro `metacaracter` com o seguinte conteúdo:

```
aquí tem *  
aquí tem caracter no fim: caracter  
aquí nao tem caracter no fim  
este tambem tem no fim caracter  
e este tem * outra vez
```

2.27. Justificar as diferenças resultantes da execução dos seguintes comandos:

```
grep * metacaracter  
grep '*' metacaracter
```

2.28. Procurar as linhas do ficheiro `~/exerc1/teste/metacaracter` que terminam com a palavra `caracter`

### 3. Exercícios extras

3.1. Procurar em `/usr/include` os ficheiros (e respetivas linhas) onde está definida a constante `NULL` (na linguagem de programação C).

Uma constante é definida do seguinte modo: `#define <nome da constante> <valor da constante>`