

# Sistemas Operativos 2021 / 2022

## Licenciatura em Engenharia Informática

### Lab. 02 – Programação com C em Linux

Nesta aula pretende-se que os alunos fiquem com uma noção da programação com C em ambiente Linux, do compilador *gcc* e da utilização de *makefiles*.

#### Ex. 1 – Edição com vim e compilação com gcc

---

Na máquina virtual criada no laboratório anterior, abra o terminal, navegue até à pasta Desktop (cd ~/Desktop) e crie uma pasta de nome “lab2”. Dentro da pasta anterior, crie outra de nome “ex1” e navegue novamente até à mesma. Usando o editor *vim*, crie o ficheiro “hello.c” com o seguinte código:

```
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

Se o vim não estiver instalado faça *sudo apt-get install vim* para o instalar. Após a criação do código C, é necessário compilar usando o *gcc*:

```
$ gcc hello.c
```

Verifique usando o comando *ls* que o gcc gerou o ficheiro executável “a.out”. Por defeito, o compilador gera o ficheiro *a.out* se não for indicado outro nome. Para forçar outro nome para o executável usa-se o parâmetro “-o”:

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

Para executar o ficheiro basta então:

```
$ ./hello
Hello World!
```

O prefixo “./” indica à shell que se pretende invocar um comando ou *script* que está no directório local.

## Ex. 2 – Criação de programa em C

---

Em `~/Desktop/lab2/ex2/main.c`, edite, compile e execute um programa que peça um número ao utilizador e escreva o seu quadrado no ecrã (stdout).

## Ex. 3 – Utilização e compilação de *header files*

---

Numa pasta “ex3” crie três ficheiros (*main.c*, *math.c* e *math.h*). Implemente, no ficheiro *math.c*, as seguintes funções que deverão ser definidas no ficheiro *math.h*.

```
/* math.h */
int square(int val);
int cube(int val);
```

Use o ficheiro *main.c* para implementar um programa que peça um número e escreva o seu quadro e cubo usando as funções definidas acima. Para compilar terá de indicar ao *gcc* que pretende incluir o ficheiro *math.c*:

```
$ gcc -o main main.c math.c
```

Garanta que não tem erros de compilação nem *warnings*!

## Ex. 4 – Utilização de *makefiles*

---

Um *makefile* é um ficheiro que contém um conjunto de regras utilizados pelo *GNU Make*, uma ferramenta que controla a geração automática de executáveis e outros ficheiros (<https://www.gnu.org/software/make/>).

Copie o conteúdo da pasta do exercício anterior para uma nova pasta “ex4” e crie um ficheiro de nome “makefile” com as seguintes regras:

```
build:
    gcc -o main main.c math.c

run:
    ./main
```

Execute o comando “make” no terminal. Por defeito, o *make* executa a primeira regra de um *makefile* pelo que resultará na compilação e criação do executável *main*.

```
$ make
gcc -o main main.c math.c
```

Para executar qualquer uma das outras regras, basta executar o *make* com o nome da regra como argumento:

```
$ make run
```

### Ex. 5 – Make clean

---

Altere o *makefile* anterior de modo a incluir uma nova regra “clean” de modo a que o comando *make clean* remova o executável criado pelo comando *make build*.

### Ex. 6 – Make go

---

Inclua uma nova regra “go” de modo a que o comando *make go* compile e execute o programa criado. Esta regra deverá fazer uso das regras “build” e “run” criadas anteriormente.

Depois altere a regra “go” de modo a executar o comando “ls” após a compilação e execução.

(fim de enunciado)