

# Projecto de Sistemas Operativos 2017-18

## heatSim

### Exercício 0 (parte 1)

LEIC-A / LEIC-T / LETI  
IST

#### Abstract

Este documento pretende guiar os alunos a realizar o exercício preparatório para o projecto da disciplina de Sistemas Operativos. Este exercício consiste na concretização de uma versão sequencial do simulador **heatSim**. Este exercício não é para avaliação e deve ser realizado maioritariamente nas aulas de laboratório.

## 1 Introdução

Os alunos devem ler primeiro o documento que faz uma introdução ao projecto, antes de lerem este guia.

Para facilitar a realização da versão sequencial do programa é fornecido código fonte que concretiza já parte do programa. Nomeadamente, é fornecido:

- O código para manipulação de uma estrutura de dados matricial, que poderá ser usada para guardar a temperatura dos “pontos” da superfície.
- Um “esqueleto” do corpo principal do programa, já com a leitura dos parâmetros passados na linha de comando.

Este guia sugere alguns exercícios para os alunos se familiarizarem e completarem o código fornecido, o que será útil para os exercícios seguintes.

Os exercícios permitem ainda praticar o ciclo de desenvolvimento de aplicações em linguagem C no ambiente UNIX.

## 2 Contacto com o ambiente UNIX

1. Crie um directório no seu computador e descarregue para lá o arquivo `heatSim_ex01.zip` que está disponível na página da disciplina (no fenix), na secção “Laboratórios”.

Para extrair os ficheiros contidos no arquivo, use o comando `unzip heatSim_ex01.zip`

2. Relembre o que fazem comandos como, por exemplo, `cd`, `ls`, `cat`, `cp`, `mv`, `rm`, `mkdir` e `rmdir`.

Recordar também que a generalidade dos comandos aceitam *switches* que modificam o seu comportamento. Comparar, por exemplo, o efeito do comando `ls` simples e `ls -CAF`.

Na secção seguinte detalha-se como pode obter ajuda em ambiente UNIX.

### 3 Utilização do manual

1. Pode aceder a informação detalhada sobre comandos de sistema, programas e funções da linguagem C, usando o comando `man` (manual).

Por exemplo, para se informar sobre o uso do próprio comando `man` escrever:

```
man man
```

Para navegar nas páginas do manual podem ser usadas as setas do teclado e as teclas “PageUp” e “PageDown”. Para sair do manual basta pressionar a tecla `q`.

2. O manual encontra-se organizado em secções. Para a cadeira Sistemas Operativos, as secções mais relevantes são: 1 (comandos/utilidades), 2 (chamadas de sistema) e 3 (funções de biblioteca). Isto é relevante pois existem comandos/funções com o mesmo nome que têm propósito e funcionamento diferentes.

Por exemplo, isso observa-se para o comando `printf` que está na secção 1 e a função da linguagem C `printf` que está na secção 3. Ao evocar o manual pode especificar a que secção pretende aceder, indicando o seu número antes do nome. Experimente:

```
man printf
```

```
man 3 printf
```

3. O manual também fornece informação sobre programas/ferramentas como, por exemplo, `make` e `zip`. Outro modo de obter informação recorre directamente aos programas/ferramentas e ao uso do `switch --help`. Exemplos:

```
gcc --help
```

```
gdb --help
```

```
make --help
```

```
zip --help
```

4. O uso do manual é especialmente útil para obter informação sobre as funções do C e identificar os valores devolvidos - notar a secção `RETURN VALUE`. Este aspecto é muito importante pois os alunos devem identificar e tratar as diversas situações de erro que podem ocorrer.

### 4 Análise do código fornecido

1. Analise os ficheiros extraídos do arquivo `heatSim_ex01.zip` usando o editor da sua preferência.

Notar que é usada a estrutura de dados `DoubleMatrix2D` para representar uma matriz. Essa estrutura contém o número de linhas e de colunas da matriz e um apontador para o seu conteúdo, o qual é constituído por um vector de variáveis do tipo `double`.

O código fornecido inclui diversas funções que manipulam a estrutura `DoubleMatrix2D` e que realizam tarefas que podem ser necessárias na resolução do projecto como por exemplo: imprimir o conteúdo da matriz, copiar o conteúdo de uma matriz para outra matriz semelhante, preencher uma linha da matriz com um determinado valor, etc. A assinatura destas funções encontra-se no ficheiro `matrix2d.h` e o respectivo código no ficheiro `matrix2d.c`. Chama-se ainda a atenção para as macros `dm2dGetEntry` e `dm2dSetEntry` definidas em `matrix2d.h`.

2. Estudar a função `dm2dNew` existente no ficheiro `matrix2d.c`.

## 5 Geração e modificação do programa heatSim

1. Gere o programa `heatSim` e execute-o usando os seguintes comandos:

- `gcc -g -c matrix2d.c`
- `gcc -g -c main.c`
- `gcc -o heatSim matrix2d.o main.o`
- `./heatSim 5 10 10 0 0 10`

2. Pela análise do ficheiro `main.c` verifica-se que o programa necessita de 6 argumentos, nomeadamente:

- `N` - dimensão (útil) da matriz (na realidade haverá que considerar as arestas, pelo que a dimensão real da matriz será de  $(N + 2) \times (N + 2)$ ; as linhas e as colunas variam entre 0 e  $N + 1$ ).
- `tEsq`, `tSup`, `tDir` e `tInf` - temperaturas a atribuir às arestas esquerda, superior, direita e inferior da matriz.
- `iteracoes` - número de iterações a realizar na simulação.

Verifique que validações são feitas aos argumentos e experimente correr `./heatSim 5 0 0 0 ola 20`.

3. Adicione validações para garantir que  $N \geq 1$ , que as temperaturas são  $\geq 0$  e que o número de iterações é  $\geq 1$ .

4. Adicionalmente, para se familiarizar com as funções existentes em `matrix2d.c` modifique `main.c` para que o programa execute as seguintes operações:

- Após a criação da matriz `matrix` inicialize-a colocando o valor 0 na linha 0, o valor 1 na linha 1, etc. Imprimir a matriz.
- Após a criação da matriz `matrix_aux` inicialize-a colocando o valor 0 na coluna 0, o valor 1 na coluna 1, etc. Imprimir a matriz.
- Copie a linha 3 da matriz `matrix` para a linha 2 da matriz `matrix_aux`. Imprimir `matrix_aux`.

5. Agora que já está familiarizado com o `matrix2d.c`, implemente a função `simul` cujo esqueleto já se encontra em `main.c`. Essa função deve realizar uma simulação, com o número de iterações indicado, começando com os valores existentes em `matrix` e usando `matrix_aux` como matriz auxiliar.

6. No final, a função `main` deve imprimir o resultado da simulação.

Para referência, indica-se o resultado a obter quando executa `./heatSim 5 10 10 0 0 10`

10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	0.0000
10.0000	8.8569	7.8830	7.0624	6.1991	4.5381	0.0000
10.0000	7.8830	6.0963	4.7689	3.6157	2.2181	0.0000
10.0000	7.0624	4.7689	3.1553	2.0678	1.0914	0.0000
10.0000	6.1991	3.6157	2.0678	1.1351	0.5342	0.0000
10.0000	4.5381	2.2181	1.0914	0.5342	0.2192	0.0000
10.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000