

# **Relatório do Trabalho**

Disciplina de Sistema Operativos

Ano Letivo 2022/2023

## **Simulador de Sistema Operativo - Modelo de 5 estados**



### **Membros do grupo**

Alexandre Costa nº 48039

João Trabuço nº 52194

Tomás Oliveira nº 52812

## Descrição do trabalho

O programa proposto é composto por cinco ficheiros: `filas.h`, `processos.h`, `filas.c`, `processos.c` e `main.c`. Neste relatório, será feita uma descrição sucinta da estrutura do programa.

O ficheiro `filas.c` contém as funções para manipulação de filas, nomeadamente para criar uma fila vazia, inserir um processo na fila, remover o primeiro processo da fila e retorná-lo, verificar se a fila está vazia e ter acesso ao primeiro processo da fila sem que seja removido. Adicionalmente, este ficheiro contém duas funções que são específicas para este projeto: `Running_Para_Blocked` e `processo_blocked`. Estas funções percorrem a lista de processos e verificam se algum processo se encontra nos estados "RUNNING" ou "BLOCKED", respetivamente. Se algum processo estiver nestes estados, as funções analisam o processo e decidem se este deve passar para o estado "BLOCKED", "EXIT", "READY" ou continuar no estado "RUNNING". Caso o processo deva passar para o estado "BLOCKED" ou "READY", este é adicionado à respetiva fila.

O ficheiro `filas.h` contém as definições de estruturas e as declarações de funções para a manipulação de filas. As estruturas definidas são `Node` e `Queue`, que são utilizadas para representar um nó da fila e a própria fila, respetivamente. As funções declaradas neste ficheiro são as mesmas que estão implementadas no ficheiro `filas.c`.

O ficheiro `processos.c` contém as funções para manipulação de processos, nomeadamente para criar um novo processo, inicializar os seus campos e atualizar o seu estado.

O ficheiro `processos.h` contém as definições de estruturas e as declarações de funções para a manipulação de processos. A estrutura definida é `Processo`, que representa um processo e contém os seus campos.

Por fim, o ficheiro `main.c` é o ponto de entrada do programa. Neste ficheiro, é feita a leitura dos processos a partir de um ficheiro, é inicializada a fila de processos prontos (`Q_READY`) e a fila de processos bloqueados (`Q_BLOCKED`) e é executado o algoritmo de escalonamento dos processos. Para cada instante de tempo, é percorrida a lista de processos e é decidido se cada processo deve passar para outro estado ou continuar no estado atual. Os processos que passam para o estado "BLOCKED" são adicionados à fila de processos bloqueados, os processos que passam para o estado "READY" são adicionados à fila de processos prontos e os processos que passam para o estado "EXIT" são removidos da lista de processos e escritos num ficheiro.

## Output de programas

Exemplo 1:

```
int programas[3][8] = {  
  {1, 3, 1, 2, 2, 4, 0, 0},  
  {1, 4, 2, 4, 1, 3, 0, 0},  
  {3, 2, 1, 6, 1, 3, 0, 0}};
```

instante	proc1	proc2	proc3
0			
1	NEW	NEW	
2	RUNNING	READY	
3	RUNNING	READY	NEW
4	RUNNING	READY	READY
5	BLOCK	RUNNING	READY
6	READY	RUNNING	READY
7	READY	RUNNING	READY
8	READY	RUNNING	READY
9	READY	BLOCK	RUNNING
10	READY	BLOCK	RUNNING
11	RUNNING	READY	BLOCK
12	RUNNING	READY	READY
13	BLOCK	RUNNING	READY
14	BLOCK	RUNNING	READY
15	READY	RUNNING	READY
16	READY	RUNNING	READY
17	READY	BLOCK	RUNNING
18	READY	READY	RUNNING
19	READY	READY	RUNNING
20	READY	READY	RUNNING
21	READY	READY	RUNNING
22	READY	READY	RUNNING
23	RUNNING	READY	BLOCK
24	RUNNING	READY	READY
25	RUNNING	READY	READY
26	RUNNING	READY	READY
27	EXIT	RUNNING	READY
28		RUNNING	READY
29		RUNNING	READY
30		EXIT	RUNNING
31			RUNNING
32			RUNNING
33			EXIT