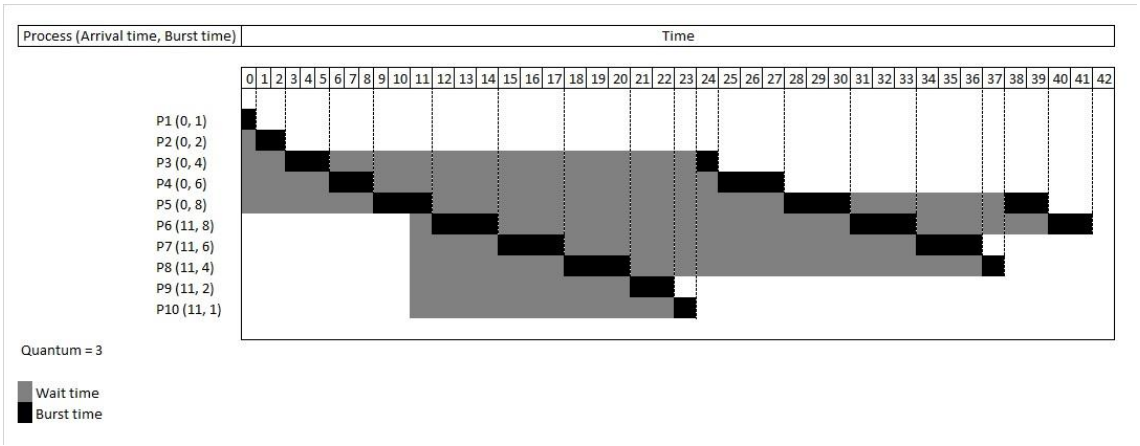
Universidade de Évora

Curso de Engenharia Informática

Sistemas Operativos 2019/2020

Trabalho 1 – Escalonamento



Trabalho realizado por:

Dinis Matos nº42738

Introdução

O trabalho realizado tem como objetivo ser um simulador de escalonamento dum sistema operativo.

Estrutura do trabalho

Implementação de filas

Para a execução do programa é feita uma implementação de filas à parte para depois ser utilizada para o escalonamento.

Esta implementação tem uma estrutura de fila, cujo nome é *struct Queue*, tem um método que cria um estrutura de fila nova, um método boleano que diz se a fila está cheia, outra que diz se está vazia. Também tem um método de retorna o tamanho atual da fila, outro que coloca um valor no fim da fila e mais um que retira e retorna o primeiro valor da fila. Por fim, tem duas funções. Uma que retorna o início da fila e outra que retorna o fim da fila.

Includes e Defines

É declarado inicialmente as bibliotecas “stdio.h”, “stdlib.h”, “string.h”, “stdbool.h”, a implementação de filas e três variáveis iniciais, Quantum (no caso do escalonamento ser Round Robin), Size (para definir o tamanho de palavras) e nomeficheiro (nome do input).

Structs

A estrutura “Processo” é inicializada com os seguintes argumentos: sequencia (que contém a sequencia alternada do tempo de CPU e um tipo de pedido I/O), quantidadesequencia, pid (número de Pid), inicio (instante em que entra).

Funções

- contalinhas

  Função que recebe um nome e que retorna o número de linhas do ficheiro com esse nome. Se este não existir o programa acaba.

- lerinput

  Recebe uma ordem de estruturas e um ficheiro. Depois esta função coloca a informação desejada no respetivo sítio dessa estrutura, ainda corrigindo algumas falhas, por exemplo alguns espaços a mais, **porém** na linha final é necessário ter pelo menos um espaço no final para esta função captar toda a informação corretamente.

- openfileteste

  Esta função que recebe uma ordem de estruturas e um nome. Se um ficheiro com o nome existir a função "lerinput" será chamada com uma ordem de estruturas e um ficheiro. Se este não existir o programa acaba.

- retirarprimeiro

  Função que contém dois argumentos. Uma ordem e o tamanho dessa mesma. Depois esta função retira apenas o primeiro elemento dessa ordem e reduz o tamanho dessa ordem por um.

- inicio_ready

  Quando o tempo de inicio do processo, coincidir com o instante atual, esta função coloca esse processo na fila *ready*.

- blocked_ready

  Quando o tempo dos processos na fila *blocked* acaba, esses mesmos passam para a fila *ready*; é o objetivo desta função.

- prints

  Esta função é responsável por todos os *outputs* do programa.

- ciclo

  Função que executa "vezes"(variável) ciclos de escalonamento e que utiliza as funções prints, inicio_ready e blocked_ready.

- execucao

  Esta função é responsável pela execução do escalonamento escolhido, tanto para FCFS e Round Robin (se Quantum>0 então terá uma execução de acordo com o escalonamento Round Robin).

- main

  Principal função do programa que conta as linhas do ficheiro, guarda as variáveis pretendidas e executa a função "execucao".

Input/Output

Input

```
100 0 1 3 10 3 6
101 0 4 4 2
200 1 2 5 1  2 3
300 1 7 6 1
```

Output FCFS

```
0 | READY 101 | RUN 100 | BLOCKED
1 | READY 200 300 | RUN 101 | BLOCKED 100
2 | READY 200 300 | RUN 101 | BLOCKED 100
3 | READY 200 300 | RUN 101 | BLOCKED 100
4 | READY 200 300 100 | RUN 101 | BLOCKED
5 | READY 300 100 | RUN 200 | BLOCKED 101
6 | READY 300 100 | RUN 200 | BLOCKED 101
7 | READY 100 | RUN 300 | BLOCKED 101 200
```

```
8 | READY 100 | RUN 300 | BLOCKED 101 200
9 | READY 100 101 | RUN 300 | BLOCKED 200
10 | READY 100 101 | RUN 300 | BLOCKED 200
11 | READY 100 101 | RUN 300 | BLOCKED 200
12 | READY 100 101 200 | RUN 300 | BLOCKED
13 | READY 100 101 200 | RUN 300 | BLOCKED
14 | READY 101 200 | RUN 100 | BLOCKED 300
15 | READY 101 200 | RUN 100 | BLOCKED 300
16 | READY 101 200 | RUN 100 | BLOCKED 300
17 | READY 101 200 | RUN 100 | BLOCKED 300
18 | READY 101 200 | RUN 100 | BLOCKED 300
19 | READY 101 200 | RUN 100 | BLOCKED 300
20 | READY 101 200 300 | RUN 100 | BLOCKED
21 | READY 101 200 300 | RUN 100 | BLOCKED
22 | READY 101 200 300 | RUN 100 | BLOCKED
23 | READY 101 200 300 | RUN 100 | BLOCKED
24 | READY 200 300 | RUN 101 | BLOCKED 100
25 | READY 200 300 | RUN 101 | BLOCKED 100
26 | READY 300 | RUN 200 | BLOCKED 100
27 | READY 100 | RUN 300 | BLOCKED 200
28 | READY | RUN 100 | BLOCKED 200
29 | READY 200 | RUN 100 | BLOCKED
30 | READY 200 | RUN 100 | BLOCKED
31 | READY 200 | RUN 100 | BLOCKED
32 | READY 200 | RUN 100 | BLOCKED
33 | READY 200 | RUN 100 | BLOCKED
34 | READY | RUN 200 | BLOCKED
35 | READY | RUN 200 | BLOCKED
36 | READY | RUN 200 | BLOCKED
```

Output Round Robin – Quantum = 3

```
0 | READY 101 | RUN 100 | BLOCKED
1 | READY 200 300 | RUN 101 | BLOCKED 100
2 | READY 200 300 | RUN 101 | BLOCKED 100
3 | READY 200 300 | RUN 101 | BLOCKED 100
4 | READY 300 100 101 | RUN 200 | BLOCKED
5 | READY 300 100 101 | RUN 200 | BLOCKED
6 | READY 100 101 | RUN 300 | BLOCKED 200
7 | READY 100 101 | RUN 300 | BLOCKED 200
8 | READY 100 101 | RUN 300 | BLOCKED 200
9 | READY 101 300 | RUN 100 | BLOCKED 200
10 | READY 101 300 | RUN 100 | BLOCKED 200
11 | READY 101 300 200 | RUN 100 | BLOCKED
12 | READY 300 200 100 | RUN 101 | BLOCKED
13 | READY 200 100 | RUN 300 | BLOCKED 101
14 | READY 200 100 | RUN 300 | BLOCKED 101
15 | READY 200 100 | RUN 300 | BLOCKED 101
16 | READY 100 300 | RUN 200 | BLOCKED 101
17 | READY 300 101 | RUN 100 | BLOCKED 200
18 | READY 300 101 | RUN 100 | BLOCKED 200
19 | READY 300 101 200 | RUN 100 | BLOCKED
20 | READY 101 200 100 | RUN 300 | BLOCKED
21 | READY 200 100 | RUN 101 | BLOCKED 300
22 | READY 200 100 | RUN 101 | BLOCKED 300
23 | READY 100 | RUN 200 | BLOCKED 300
24 | READY 100 | RUN 200 | BLOCKED 300
25 | READY 100 | RUN 200 | BLOCKED 300
26 | READY | RUN 100 | BLOCKED 300
27 | READY 300 | RUN 100 | BLOCKED
28 | READY 300 | RUN 100 | BLOCKED
```

```
29 | READY 100 | RUN 300 | BLOCKED
30 | READY | RUN 100 | BLOCKED
31 | READY | RUN | BLOCKED 100
32 | READY | RUN | BLOCKED 100
33 | READY | RUN | BLOCKED 100
34 | READY | RUN 100 | BLOCKED
35 | READY | RUN 100 | BLOCKED
36 | READY | RUN 100 | BLOCKED
37 | READY | RUN 100 | BLOCKED
38 | READY | RUN 100 | BLOCKED
39 | READY | RUN 100 | BLOCKED
```