



UNIVERSIDADE DE ÉVORA

Simulador de Escalonamento de Processos

Sistemas Operativos I

Professor: Luís Rato

Realizado por: Miguel de Carvalho 43108

3 de Abril de 2020

1 Introdução

Neste trabalho foi solicitado a realização de um programa que simule o **Escalonamento de Processos** num modelo de 3 estados. Na figura abaixo está representado o diagrama que descreve o modelo de 3 estados:

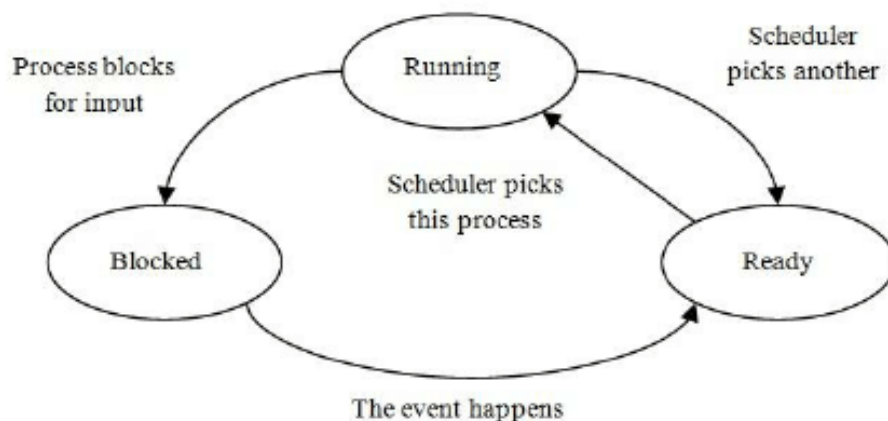


Figura 1: Diagrama de 3 Estados

O **Escalonador de Processos** faz parte do **Sistema Operativo** e é responsável por decidir em que momento cada processo estará no CPU. Existem muitos algoritmos de escalonamento para realizar essa decisão.

Neste trabalho serão utilizados o algoritmo **FCFS** e o **Round Robin (RR)**:

- O **FCFS** é um algoritmo de escalonamento não preemptivo que prioriza os processos pela ordem de chegada. Executa o processo todo do início ao fim sem o interromper, até estar concluído. Quando aparece um novo processo e ainda existe um em execução, esse novo irá para a fila de espera.
- O **Round Robin (RR)** é um algoritmo de escalonamento preemptivo que apresenta um funcionamento igual ao do **FCFS**, mas com tempo limite de execução, o **Quantum**. Ou seja, quando o processo se encontra em execução este irá ser interrompido quando o tempo de execução for igual ao **Quantum** e irá para a fila de espera (**READY**)

2 Implementação

Primeiramente comecei por pensar como deveria proceder para realizar o trabalho, e na primeira tentativa comecei por desenvolver o **FCFS**, mas na segunda tentativa cheguei a conclusão de que não seria necessário realizar um programa para a implementação do **FCFS** e outro para o **Round Robin**, pois ambos são iguais, excepto existir um limite de execução no **RR**.

Poderei então meter um limite muito grande na execução, no **Quantum**, e estarei perante o **FCFS**.

Comecei por proceder há criação das queues que iriam ser utilizados para o **READY**, **RUN** e **BLOCKED** e das respetivas funções essenciais para a sua manipulação.

O próximo desafio foi proceder a leitura do input (ficheiro com os processos e os tempos), para isso foi necessário proceder a criação de uma **Struct** para os processos.

Por conseguinte, comecei por desenvolver a função que iria realizar as trocas dos processos entre os 3 estados. Esta função foi a mais difícil de realizar devido a complexidade dos requisitos para os processos mudarem de estados. Foi necessário também proceder a criação de novas funções para facilitar alguns procedimentos que eram realizados de forma repetitiva.

3 Execução

Quando o utilizador executar o **Simulador de Escalonamento** irá ser questionado acerca do nome do ficheiro de input e qual algoritmo pretende utilizar.

- Caso o utilizador escolha o **FCFS**, o **QUANTUM** irá ter um valor de **10000**, assim nunca irá obrigar o programa a passar para o estado ready.
- Caso o utilizador escolha o **Round Robin**, será perguntado ao utilizador qual o **Quantum** desejado.

4 Análise de Resultados

Em suma, este trabalho fez-me entender melhor como funciona o escalonador e as condições que cada algoritmo usa para proceder há mudança dos processos entre os estados e as diferenças de tempo.

5 Problemas

Neste trabalho só existe um problema e consiste no facto de que em algumas execuções o ultimo processo existente no ficheiro não é lido, mas este problema só acontece no **Linux**. Por outro lado no **Windows** este problema não ocorre.

6 Conclusão

Em suma, este trabalho fez-me entender melhor como funciona o escalonador e as condições que cada algoritmo usa para proceder há mudança dos processos entre os estados e as diferenças de tempo.