

# Simulador de escalonamento de processos

## Trabalho Final de Sistemas Operacionais

Daniel Fernando Jimenez

117028769

Matheus Cunha Simões

117091021

### 1. Introdução

Neste trabalho vamos desenvolver um simulador em C de escalonamento de processos usando a estratégia de seleção Round Robin com Feedback, nele vamos observar o que acontece em cada instante de tempo T para poder entender melhor como o processador lida com vários processos, várias filas de entrada e de I/O.

### 2. Objetivos

- Criar um simulador em C sem uso de bibliotecas externas nem estruturas prontas, usar o basico do basico assim como o processador faz.
- Entender a estratégia de Round Robin e seu time slice para poder ser aplicada num ambiente de geração de processos aleatórios, solicitações de I/O e fila de alta e baixa prioridade

### 3. Premissas

- A quantidade de processos foi definida por 5 (Podem ser mudadas logo no começo do código).
- Os Tempos de duração de cada tipo de I/O ficaram assim (Podem ser mudadas logo no começo do código).

I/O	Tempo
Disco	5
Fita	10
Impressora	15

- O Tempo de serviço e o tempo de chegada são gerados aleatoriamente para cada processo no começo do programa.
- Cada fatia de Tempo é de 1 unidade e o time slice da estratégia ficou estabelecida em 3 (Pode ser mudada logo no começo do código).

- A solicitação de IO para cada processo sendo executada foi randômica, a cada instante de tempo ele tem um 10% de probabilidade de sair do processador e entrar numa fila IO.

#### 4. Descrição

Nós dividimos o desenvolvimento do programa em vários passos, o primeiro passo foi a criação das bibliotecas que vão representar os processos, os dispositivos de IO e a fila. Para a criação da fila usamos a seguinte estrutura base

```
//A structure to represent a queue
typedef struct _Queue {
    int front, rear, size;
    unsigned capacity;
    int* array;
} Queue;
```

e nela criamos as funções de isFull, isEmpty, enqueue, dequeue, front e rear. Ela vai servir para simular nossa fila de alta e baixa prioridade para processos no estado de pronto.

No arquivo do processo fizemos algo mais simples porque nele só precisávamos da estrutura de controle

```
//A structure to represent a process
typedef struct _Process {
    int id;
    int startTime; //apenas para agendar o inicio na simulacao
    int timeExecuted;
    int executionTime;
    int endTime; //apenas para guardar o fim da simulação e calcular o turnaround
} Process;
```

Com isso conseguimos controlar o inicio do processo e o tempo de serviço no processador, adicionalmente criamos as três funções básicas para o processo que são: createProcess( criação do processo e seus atributos básicos), remaining(pra saber o tempo que falta pra sair do processador) e exec(simulando a execução).

Por último o IO device foi criado da seguinte maneira

```
//A structure to represent an I/O device
typedef struct _IOdevice {
    Queue *processes; //fila de processos para realizar IO
    int executed; //quanto já foi executado do processo no topo da fila
    int operationTime; //tempo da operacao de IO
} IOdevice;
```

Com essa estrutura controlamos o processo usando o dispositivo IO e o tempo que ele está levando para executar esse IO. Além disso foram criadas 4 funções básicas de controle sobre essa estrutura que são: createIOdevice(para inicializar o dispositivo), enqueueIO(para adicionar um processo ao IO), remainingIO(Para saber quanto falta para o processo acabar) e o execIO(simular a execução do processo no IO). Adicionalmente foi criado uma variável de IOdevice para cada dispositivo de IO(Disca, Fita, Impressora)

[illegible]

Criando processos:

Processo 1: Chegada no tempo = 10, tempo de serviço = 12.

Processo 2: Chegada no tempo = 38, tempo de serviço = 23.

Processo 3: Chegada no tempo = 37, tempo de serviço = 17.

Processo 4: Chegada no tempo = 34, tempo de serviço = 22.

Processo 5: Chegada no tempo = 19, tempo de serviço = 12.

t = 0: CPU ociosa.

t = 1: CPU ociosa.

t = 2: CPU ociosa.

t = 3: CPU ociosa.

t = 4: CPU ociosa.

t = 5: CPU ociosa.

t = 6: CPU ociosa.

t = 7: CPU ociosa.

t = 8: CPU ociosa.

t = 9: CPU ociosa.

t = 10: Processo 1 foi iniciado e inserido na fila de alta prioridade.

t = 10: Processo 1 retirado da fila de alta prioridade e entra em execução.

t = 13: Processo 1 sofreu preempção, inserido na fila de baixa prioridade.

t = 13: Processo 1 retirado da fila de baixa prioridade e entra em execução.

t = 16: Processo 1 sofreu preempção, inserido na fila de baixa prioridade.

t = 16: Processo 1 retirado da fila de baixa prioridade e entra em execução.

t = 19: Processo 1 sofreu preempção, inserido na fila de baixa prioridade.

t = 19: Processo 5 foi iniciado e inserido na fila de alta prioridade.

t = 19: Processo 5 retirado da fila de alta prioridade e entra em execução.

t = 22: Processo 5 sofreu preempção, inserido na fila de baixa prioridade.

t = 22: Processo 1 retirado da fila de baixa prioridade e entra em execução.

t = 25: O processo 1 finalizou sua execução.

t = 25: Processo 5 retirado da fila de baixa prioridade e entra em execução.

t = 28: Processo 5 sofreu preempção, inserido na fila de baixa prioridade.

t = 28: Processo 5 retirado da fila de baixa prioridade e entra em execução.

t = 31: Processo 5 inserido na fila de I/O para disco.

t = 31: CPU ociosa.

t = 32: CPU ociosa.

t = 33: CPU ociosa.

t = 34: Processo 4 foi iniciado e inserido na fila de alta prioridade.

t = 34: Processo 4 retirado da fila de alta prioridade e entra em execução.

t = 36: Processo 5 terminou I/O em disco e foi inserido na fila de baixa prioridade.

t = 37: Processo 4 sofreu preempção, inserido na fila de baixa prioridade.

t = 37: Processo 3 foi iniciado e inserido na fila de alta prioridade.

t = 37: Processo 3 retirado da fila de alta prioridade e entra em execução.

t = 38: Processo 2 foi iniciado e inserido na fila de alta prioridade.

t = 40: Processo 3 sofreu preempção, inserido na fila de baixa prioridade.

t = 40: Processo 2 retirado da fila de alta prioridade e entra em execução.

t = 43: Processo 2 inserido na fila de I/O para disco.

t = 43: Processo 5 retirado da fila de baixa prioridade e entra em execução.

t = 46: O processo 5 finalizou sua execução.

t = 46: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 48: Processo 2 terminou I/O em disco e foi inserido na fila de baixa prioridade.  
t = 49: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 49: Processo 3 retirado da fila de baixa prioridade e entra em execucao.  
t = 52: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 52: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 55: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 55: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 58: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 58: Processo 3 retirado da fila de baixa prioridade e entra em execucao.  
t = 60: Processo 3 inserido na fila de I/O para disco.  
t = 60: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 63: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 63: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 64: Processo 4 inserido na fila de I/O para fita magnetica.  
t = 64: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 65: Processo 3 terminou I/O em disco e foi inserido na fila de baixa prioridade.  
t = 67: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 67: Processo 3 retirado da fila de baixa prioridade e entra em execucao.  
t = 68: Processo 3 inserido na fila de I/O para impressora.  
t = 68: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 71: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 71: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 72: Processo 2 inserido na fila de I/O para impressora.  
t = 72: CPU ociosa.  
t = 73: CPU ociosa.  
t = 74: Processo 4 terminou I/O em fita magnetica e foi inserido na fila de alta prioridade.  
t = 74: Processo 4 retirado da fila de alta prioridade e entra em execucao.  
t = 77: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 77: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 80: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 80: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 83: Processo 4 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 83: Processo 3 terminou I/O em impressora e foi inserido na fila de alta prioridade.  
t = 83: Processo 3 retirado da fila de alta prioridade e entra em execucao.  
t = 86: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 86: Processo 4 retirado da fila de baixa prioridade e entra em execucao.  
t = 89: O processo 4 finalizou sua execucao.  
t = 89: Processo 3 retirado da fila de baixa prioridade e entra em execucao.  
t = 92: Processo 3 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 92: Processo 3 retirado da fila de baixa prioridade e entra em execucao.  
t = 94: O processo 3 finalizou sua execucao.  
t = 94: CPU ociosa.  
t = 95: CPU ociosa.  
t = 96: CPU ociosa.  
t = 97: CPU ociosa.  
t = 98: Processo 2 terminou I/O em impressora e foi inserido na fila de alta prioridade.  
t = 98: Processo 2 retirado da fila de alta prioridade e entra em execucao.

t = 101: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 101: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 104: Processo 2 sofreu preempcao, inserido na fila de baixa prioridade.  
t = 104: Processo 2 retirado da fila de baixa prioridade e entra em execucao.  
t = 105: O processo 2 finalizou sua execucao.

P1

Chegada:10  
Tempo de Servico:12  
Tempo de espera:13  
Turnaround(Tq):15  
Desempenho(Tq/Ts):1.250000

P2

Chegada:38  
Tempo de Servico:23  
Tempo de espera:82  
Turnaround(Tq):67  
Desempenho(Tq/Ts):2.913043

P3

Chegada:37  
Tempo de Servico:17  
Tempo de espera:77  
Turnaround(Tq):57  
Desempenho(Tq/Ts):3.352941

P4

Chegada:34  
Tempo de Servico:22  
Tempo de espera:67  
Turnaround(Tq):55  
Desempenho(Tq/Ts):2.500000

P5

Chegada:19  
Tempo de Servico:12  
Tempo de espera:34  
Turnaround(Tq):27  
Desempenho(Tq/Ts):2.250000

## 6. Conclusões

- Nas diferentes simulações feitas conseguimos entender o conceito de starvation na hora que um processo na fila baixa esperando ser atendido pelo processador, tem que ficar esperando pois chegam novos processos na fila de baixa prioridade.

- O estratégia Round Robin atende todos os processos de maneira justa, permitindo que processos curtos sejam processados sem esperar por outros processos longos que chegaram antes que ele.