

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

SSC0140 – Sistemas Operacionais I  
Prof<sup>a</sup> Dr<sup>a</sup> Sarita Mazzini Bruschi

**Simulador de Escalonamento de Processos**

Elias Italiano Rodrigues, 7987251  
Gabriel Tessaroli Giancristofaro, 4321350  
Rodolfo Megiato de Lima, 7987286

SÃO CARLOS  
2013

## Sumário

1. Introdução.....	2
2. Desenvolvimento.....	3
2.1 Implementação.....	3
2.2 Testes.....	4
3. Conclusão.....	5

# 1. Introdução

O intuito desse trabalho é realizar a simulação do escalonamento de processos e o gerenciamento de memória pelo qual o sistema operacional é responsável. Nessa primeira parte do projeto, apenas o conceito do escalonamento de processos está implementado, deixando então para a segunda parte o gerenciamento de memória.

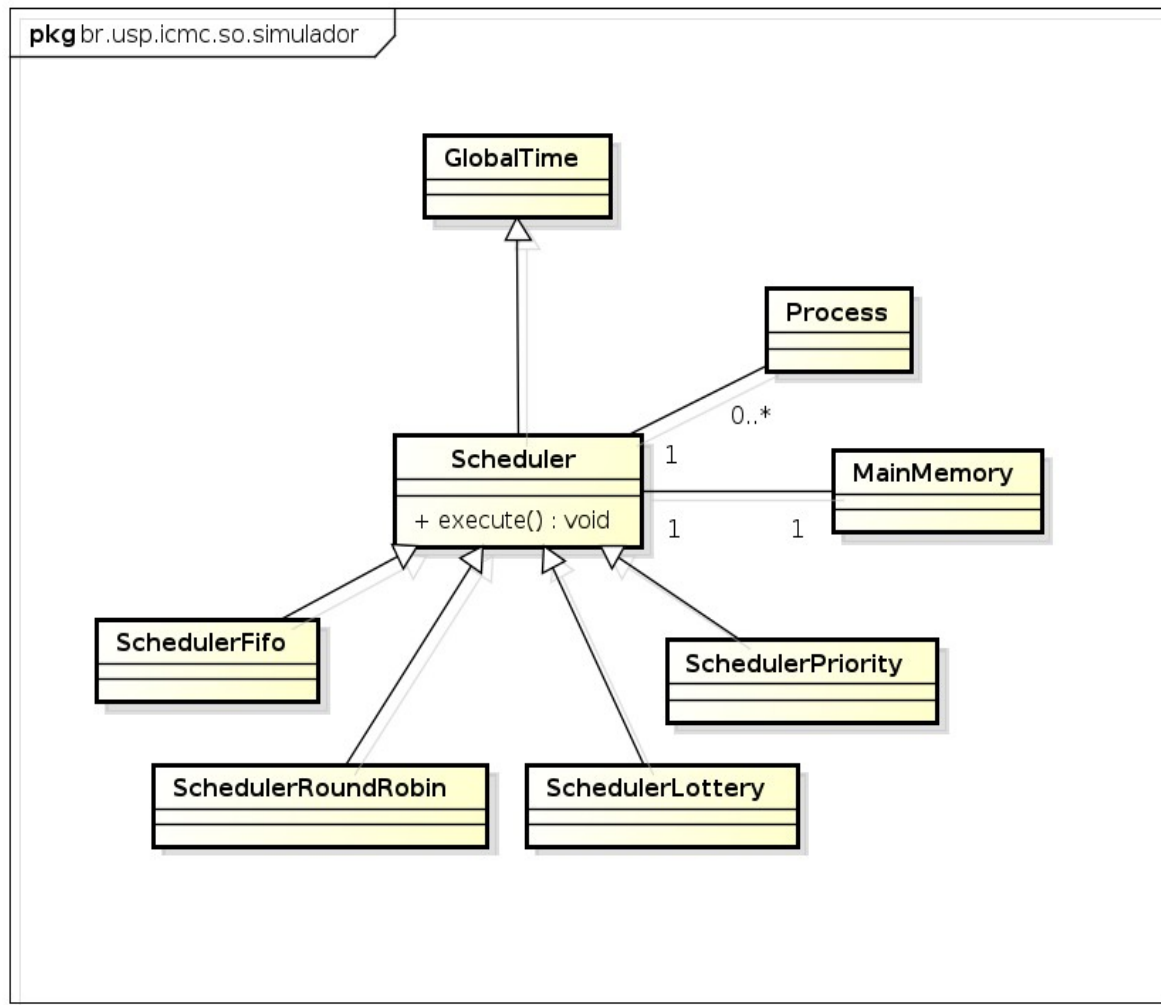
O escalonamento dos processos tem base em algoritmos de escolha para qual processo irá ganhar a oportunidade de ser executado pelo processador. Nesse simulador, utilizamos os algoritmos: FIFO, que escolhe sempre o primeiro da fila de processos prontos para executar; Round-robin, que executa o primeiro processo da fila por um período de tempo pré-definido (chamado de *quantum*), se o processo ainda não terminou seu processamento, ele volta para o fim da fila; Loteria, que escolhe de forma aleatória um processo na fila de processos prontos e o coloca para executar por um quantum de tempo, voltando para a fila se este ainda não terminou sua execução e por fim, o algoritmo de Prioridades, que utiliza de várias filas com processos de prioridades iguais alocados em cada uma delas, executando sempre primeiro, os processos das filas que possuem maior prioridade comparados com os demais.

Para que um processo seja colocado para executar no processador, ele necessita ser chaveado para tal, sendo então computado nessa simulação, uma unidade de tempo para cada chaveamento de contexto do processador. Também foram monitorados os tempos em que os processos deixam o processador e quando terminam de ser executados, além do turnaround médio dos processos e o *throughput*, no final da execução de cada um dos algoritmos.

## 2. Desenvolvimento

### 2.1 Implementação

A implementação do projeto foi feita em Java, baseada em um código-fonte disponibilizado para o trabalho. O modelo a seguir representa a relação entre as classes utilizadas:



A classe *GlobalTime* representa uma entidade que contabiliza o tempo decorrido de execução do sistema.

A classe *Process* caracteriza um processo no sistema que é escalonado pela classe *Scheduler* simulando a execução dos processos como se eles realmente fossem alocados ao processador.

A classe *MainMemory* implementa a memória principal do sistema, onde os processos são alocados enquanto esperam para ser executados por completo.

As classes filhas de *Scheduler*: *SchedulerFifo*, *SchedulerRoundRobin*, *SchedulerLottery* e *SchedulerPriority* são as classes que realmente foram implementadas desde o princípio e são elas que representam, de fato, os escalonadores do sistema. Contém a implementação do método virtual `void execute()`, o qual realmente realiza o escalonamento dos processos de acordo com cada algoritmo.

## 2.2 Testes

Foram realizados os testes com os algoritmos de escalonamento de acordo com alguns casos de teste que são gerados de maneira aleatória, já que o tempo de execução dos processos, a quantidade de memória utilizada e a prioridade destes é passada por parâmetro para a criação dos processos como valores aleatórios.

Segue abaixo um exemplo de execução de um escalonamento de 5 processos de acordo com o algoritmo Round-robin:

### PROCESSES

pid	requiredExecutionTime	requiredMemory	quantumTime	priority
0	4	9	4	-
1	7	6	4	-
2	5	1	4	-
3	3	4	4	-
4	2	4	4	-

### ROUND-ROBIN

```
-----
Now is: 0.0
Switching time: 1.0
Process 0 ended at 5.0

Now is: 5.0
Switching time: 1.0
Process 1 leaving at 10.0

Now is: 10.0
Switching time: 1.0
Process 2 leaving at 15.0

Now is: 15.0
Switching time: 1.0
Process 3 ended at 19.0

Now is: 19.0
Switching time: 1.0
Process 4 ended at 22.0

Now is: 22.0
Switching time: 1.0
Process 1 ended at 26.0

Now is: 26.0
Switching time: 1.0
Process 2 ended at 28.0
```

### TURNAROUND TIME

-----  
20.0

### THROUGHPUT

-----  
0.17857143

### **3. Conclusão**

A implementação da simulação do escalonador de processos está concluída, restando agora para a segunda parte do projeto, o desenvolvimento do simulador de gerenciamento de memória, o qual irá alocar memória para os processos que serão criados dinamicamente. Dessa forma, teremos um simulador completo, aplicando alguns dos conhecimentos adquiridos na disciplina de Sistemas Operacionais I.