

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

SSC0140 – Sistemas Operacionais I
Prof^ª Dr^ª Sarita Mazzini Bruschi

**Simulador de Escalonamento de Processos com
Alocação de Memória**

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.....	8

1. Introdução

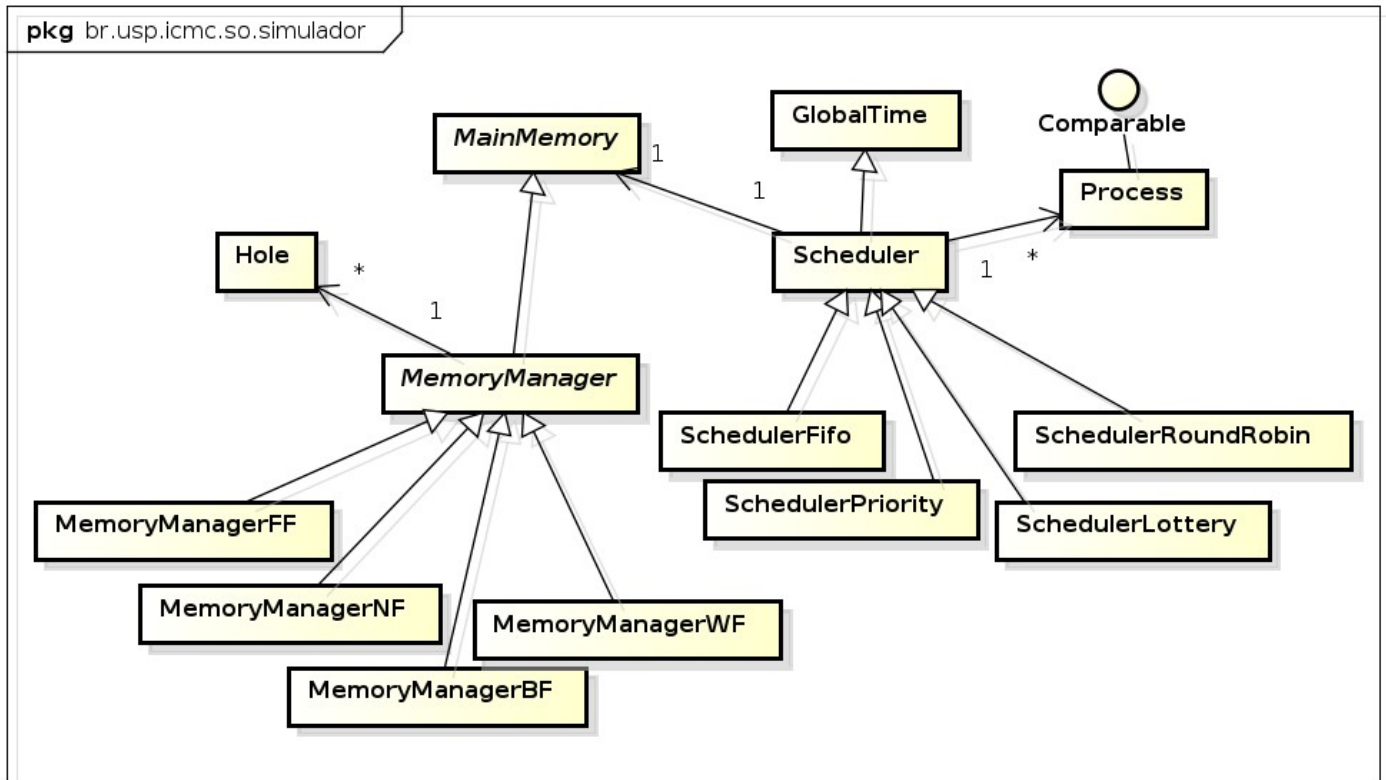
O intuito desse trabalho é realizar a simulação do escalonamento de processos e o gerenciamento de memória pelo qual o sistema operacional é responsável. Nessa segunda parte do projeto, apenas o conceito da alocação de memória foi implementado já que na primeira, o escalonamento dos processos foi concluído.

A alocação de memória tem base em algoritmos de escolha para qual região da memória principal cada processo será alocado. Nesse simulador, utilizamos os algoritmos: *First-Fit*, que escolhe sempre o primeiro espaço livre de memória suficiente para alocar o processo; *Next-Fit*, que é uma variação do *First-Fit*, escolhendo sempre o primeiro espaço livre na memória que seja suficiente para alocar o processo, mas que continua sempre de onde parou e não do início da memória a cada vez; *Best-Fit*, que escolhe sempre o espaço livre que contenha o tamanho mais próximo possível com o tamanho ocupado pelo processo e por fim, o algoritmo *Worst-Fit*, que ao contrário do *Best-Fit*, escolhe sempre o espaço livre que contenha o maior tamanho e que o processo consiga ser alocado.

2. Desenvolvimento

2.1 Implementação

A implementação do projeto foi feita em Java, baseada no código-fonte disponibilizado e incrementado com a primeira parte do trabalho – escalonamento dos processos. 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 *Hole* caracteriza o “buraco” na memória, que possui o tamanho e o ponteiro para onde ele começa.

A classe *MainMemory* implementa a memória principal do sistema, onde os processos são alocados enquanto esperam para ser executados por completo. A classe filha de *MainMemory*, *MemoryManager* é responsável pelo gerenciamento da memória principal.

As classes filhas de *MemoryManager*: *MemoryManagerFF*, *MemoryManagerNF*, *MemoryManagerBF* e *MemoryManagerWF* são as classes que realmente foram implementadas desde o princípio e são elas que representam, de fato, as classes que alocam a memória os processos na memória contendo a implementação do método virtual *int allocMemory(int blockSize, int pid)*.

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 e alocação de memória 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.

Para o teste da alocação de memória, foi utilizado um padrão de 5 processos previamente alocados de maneira aleatória, para que os algoritmos de escalonamento pudessem agir de forma correta.

Segue abaixo um exemplo de execução de um escalonamento de 10 processos de acordo com o algoritmo Round-robin para o escalonamento e First-Fit para a alocação de memória:

INITIAL PROCESSES

pid	requiredExecutionTime	requiredMemory	quantumTime	priority
1	4	5	4	2
2	3	7	4	1
3	3	8	4	4
4	4	6	4	3
5	5	5	4	1

FIRST-FIT

MEMORY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
-	-	-	-	-	-	-	-	-	-	-	-	1	1	1	1	1	-	-	-
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
-	-	-	-	-	-	-	-	-	2	2	2	2	2	2	2	-	-	-	-
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
-	-	-	-	3	3	3	3	3	3	3	3	-	-	-	-	-	-	-	-
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
-	-	-	-	4	4	4	4	4	4	-	-	-	-	-	-	-	-	-	-
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
-	-	-	-	-	-	-	-	-	-	-	-	5	5	5	5	5	-	-	-

NEW PROCESS

pid	requiredExecutionTime	requiredMemory	quantumTime	priority
6	7	10	4	-

NEW PROCESS

MEMORY

NEW PROCESS

MEMORY

5

NEW PROCESS

pid	requiredExecutionTime	requiredMemory	quantumTime	priority
9	6	9	4	-

MEMORY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
6	6	6	6	6	6	6	6	6	6	8	8	1	1	1	1	1	7	7	7
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
7	-	-	-	-	-	-	-	-	2	2	2	2	2	2	2	-	-	-	-
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
-	-	-	-	3	3	3	3	3	3	3	3	9	9	9	9	9	9	9	9
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
9	-	-	-	4	4	4	4	4	4	-	-	-	-	-	-	-	-	-	-
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
-	-	-	-	-	-	-	-	-	-	-	-	5	5	5	5	5	-	-	-

NEW PROCESS

pid	requiredExecutionTime	requiredMemory	quantumTime	priority
10	1	5	4	-

MEMORY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
6	6	6	6	6	6	6	6	6	6	8	8	1	1	1	1	1	7	7	7
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
7	10	10	10	10	10	-	-	-	2	2	2	2	2	2	2	-	-	-	-
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
-	-	-	-	3	3	3	3	3	3	3	3	9	9	9	9	9	9	9	9
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
9	-	-	-	4	4	4	4	4	4	-	-	-	-	-	-	-	-	-	-
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
-	-	-	-	-	-	-	-	-	-	-	-	5	5	5	5	5	-	-	-

ROUND-ROBIN

Now is: 0.0
Switching time: 1.0
Process 1 ended at 5.0

Now is: 5.0
Switching time: 1.0
Process 2 ended at 9.0

Now is: 9.0
Switching time: 1.0
Process 3 ended at 13.0

Now is: 13.0
Switching time: 1.0
Process 4 ended at 18.0

Now is: 18.0
Switching time: 1.0
Process 5 leaving at 23.0

Now is: 23.0
Switching time: 1.0
Process 6 leaving at 28.0

Now is: 28.0
Switching time: 1.0
Process 7 ended at 31.0

Now is: 31.0
Switching time: 1.0
Process 8 ended at 33.0

Now is: 33.0
Switching time: 1.0
Process 9 leaving at 38.0

Now is: 38.0
Switching time: 1.0
Process 10 ended at 40.0

Now is: 40.0
Switching time: 1.0
Process 5 ended at 42.0

Now is: 42.0
Switching time: 1.0
Process 6 ended at 46.0

Now is: 46.0
Switching time: 1.0
Process 9 ended at 49.0

THROUGHPUT

28.6

MEMORY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

0 Processes couldn't allocated

3. Conclusão

A implementação da simulação do escalonador de processos e a alocação de memória está concluída, complementando assim o processo iniciado no primeiro projeto. Dessa forma, temos um simulador completo, que escalona processos e aloca memória para estes, aplicando alguns dos conhecimentos adquiridos na disciplina de Sistemas Operacionais I.