

Relatório: Algoritmos Escalonadores

1. Casos de teste

Foram utilizadas 3 listas de processos (abaixo):

- Apenas processos curtos;
- Apenas processos longos;
- Processos de duração aleatória;

Curto			
Chegada	ID	Burst	Priority
0	P1	10	3
2	P2	1	1
2	P3	2	4
3	P4	9	5
16	P5	5	2
20	P6	4	5
23	P7	8	4
23	P8	7	1
29	P9	1	2
35	P10	2	3

Longo			
Chegada	ID	Burst	Priority
10	P1	105	3
35	P2	84	1
119	P3	123	4
190	P4	71	5
200	P5	99	2
39	P6	200	2
220	P7	147	1
299	P8	121	4
139	P9	85	3
157	P10	113	5

Mistos			
Chegada	ID	Burst	Priority
0	P1	10	3
2	P2	1	1
119	P3	123	4
3	P4	1	5
200	P5	99	2
20	P6	55	3
72	P7	9	1
15	P8	194	2
149	P9	72	5
55	P10	42	4

2. Resultados

2.1. Considerações Gerais

FCFS: para todos os casos, é um algoritmo basicamente imprevisível, com desempenho variável na medida da aleatoriedade na ordem de chegada dos processos.

Priority: nos gráficos obtidos, apresentam um desempenho similar ao do SJF, mas levemente inferior, pois precisam satisfazer a limitação adicional da prioridade (pré-estabelecida como parâmetro).

2.2. Turnaround

No geral, a preempção contribui para a redução do turnaround, sendo mais notável no caso misto, e com efeito pouco significativo no caso longo.

Em todos os casos, o algoritmo RR apresentou o pior desempenho nessa categoria.

2.3. Tempo de Espera

As conclusões para tempo de espera foram exatamente as mesmas do tempo de turnaround, sendo os gráficos obtidos bastante similares.

2.4. Tempo de Resposta

Nesta categoria, RR apresenta um resultado nitidamente superior no caso dos processos longos, mas perde para o SJFP nos demais casos.

Para processos curtos, RR se mostrou inferior também aos algoritmos Priority, mesmo com a limitação extra.

2.5. Quantidade de Trocas de Contexto

Nos casos longo e médio, RR apresenta uma quantidade muito maior de trocas de contexto do que todos os demais. Entretanto, no caso curto, os algoritmos de preempção

obtiveram um resultado inferior, com mais trocas de contexto que os demais. Isso acontece porque o quantum do RR ($q = 10$) tem magnitude comparável aos tempos de burst dos processos curtos.

Em todos os casos, a preempção aumentou a quantidade de trocas de contexto.

2.6. Valores Invariantes

Variaram com os casos de teste, mas não com os algoritmos as seguintes métricas:

- **Tempo de CPU**

- Pois a soma dos tempos de burst é constante.

- **Tempo de Processamento**

- Pois todos os algoritmos conseguem processar no tempo mínimo, já que não há interrupções a gerenciar, apenas a ordem dos processos.

- **Throughput**

- Como o tempo de processamento é constante, o throughput também o é (devido à sua fórmula: quantidade de processos / tempo de processamento).

2.7. Gráficos





