

CURSO:
TECNOLOGIA EM TELEMÁTICA

Escalonamento com OS *Simulator*

DATA:
04/03/24

2024.1


PROFESSOR:
DOMINGOS SÁVIO SOARES FELIPE

Prática – Políticas de escalonamento de processos utilizando o OS *Simulator*

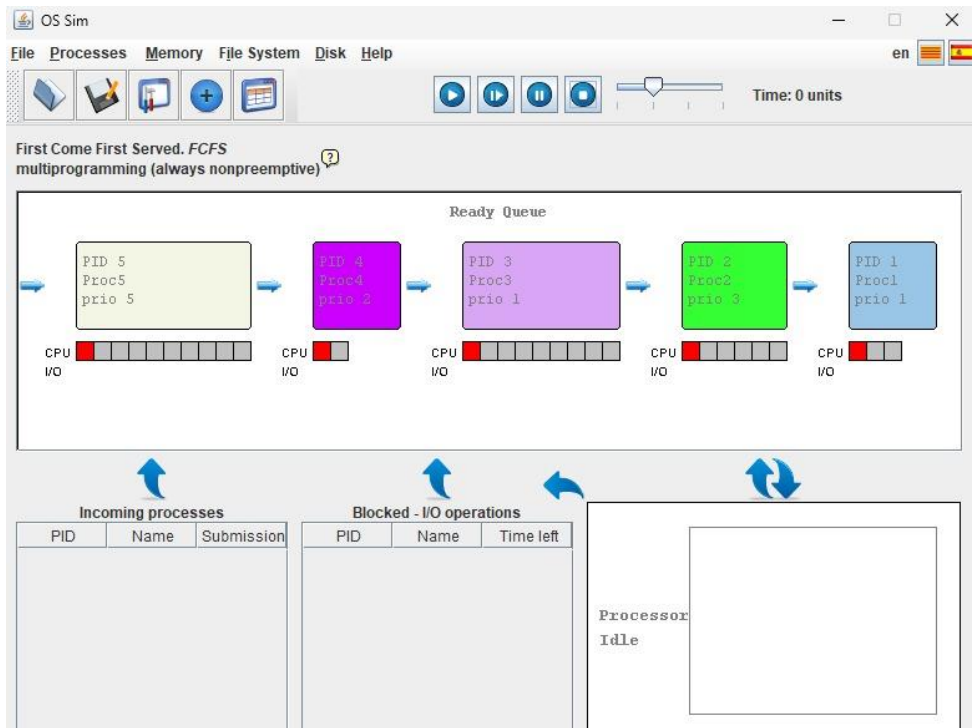
Alunos: Carla Beatriz da Silva Teixeira | Anderson Kleber Santos Da Silva

Considerando os processos da tabela abaixo, configure o OSSIM, em momentos diferentes, evidentemente, para escalonar de acordo com FCFC, SJF, RR (quantum 1 e 3) e por prioridade e faça o que é pedido nos itens logo após a tabela.

Processo	Burst (processo)	CPU bound/ IO bound	Prioridade	Tempo de Chegada
p1	3	3	0	2
p2	6	4/2	2	3
p3	9	6/3	0	1
p4	2	2	1	1
p5	10	8/2	4	2

1. Insira aqui as telas de informações de cada um dos escalonamentos dos processos (opção *Data and Statistics* 

FCFC



OS Sim

File Processes Memory File System Disk Help

en

Time: 30 units

Process Scheduling Information

Efficiency (%) 1,00

Throughput (processes/time unit) 0,17

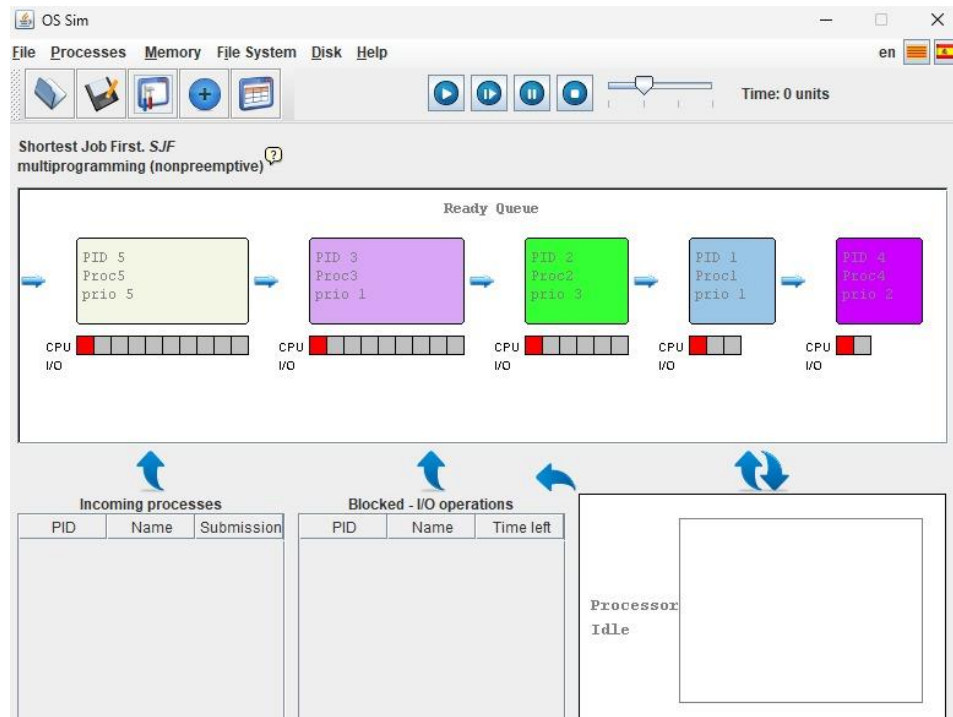
Avg. Turnaround Time (time) 16,00

Avg. Waiting Time (time) 10,00

Avg. Response Time (time) 10,00

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	Proc1	1	0	-	3	0	0	3	1.0	0.0
2	Proc2	3	0	-	6	3	3	9	0.6666666...	0.0
3	Proc3	1	0	-	9	9	9	18	0.5	0.0
4	Proc4	2	0	-	2	18	18	20	0.1	0.0
5	Proc5	5	0	-	10	20	20	30	0.3333333...	0.0

SJF



Process Scheduling Information

Efficiency (%) 1,00

Throughput (processes/time unit) 0,17

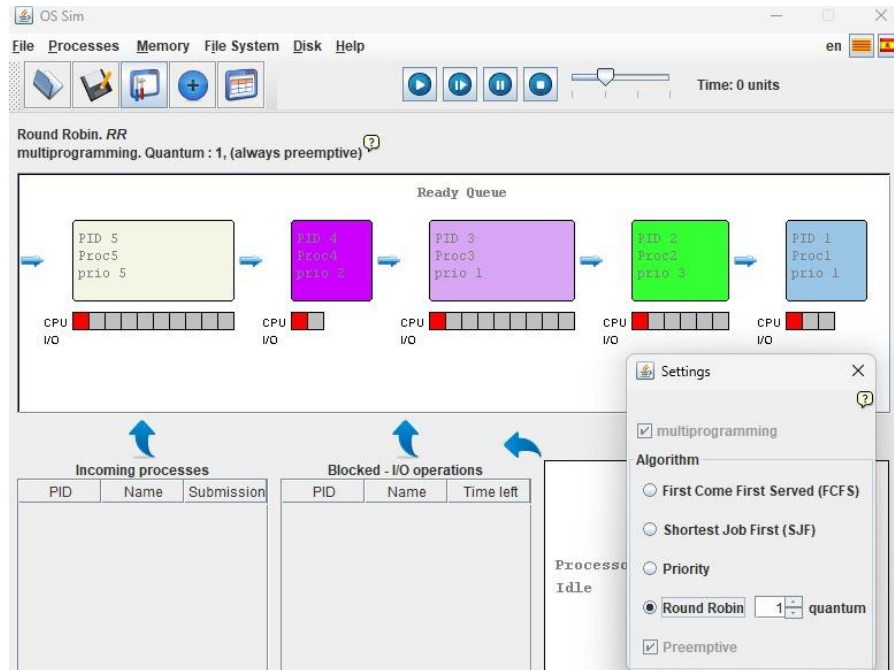
Avg. Turnaround Time (time) 13,60

Avg. Waiting Time (time) 7,60

Avg. Response Time (time) 7,60

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
4	Proc4	2	0	-	2	0	0	2	1.0	0.0
1	Proc1	1	0	-	3	2	2	5	0.6	0.0
2	Proc2	3	0	-	6	5	5	11	0.5454545...	0.0
3	Proc3	1	0	-	9	11	11	20	0.45	0.0
5	Proc5	5	0	-	10	20	20	30	0.3333333...	0.0

RR QUANTUM 1



Process Scheduling Information

Efficiency (%) 1,00

Throughput (processes/time unit) 0,17

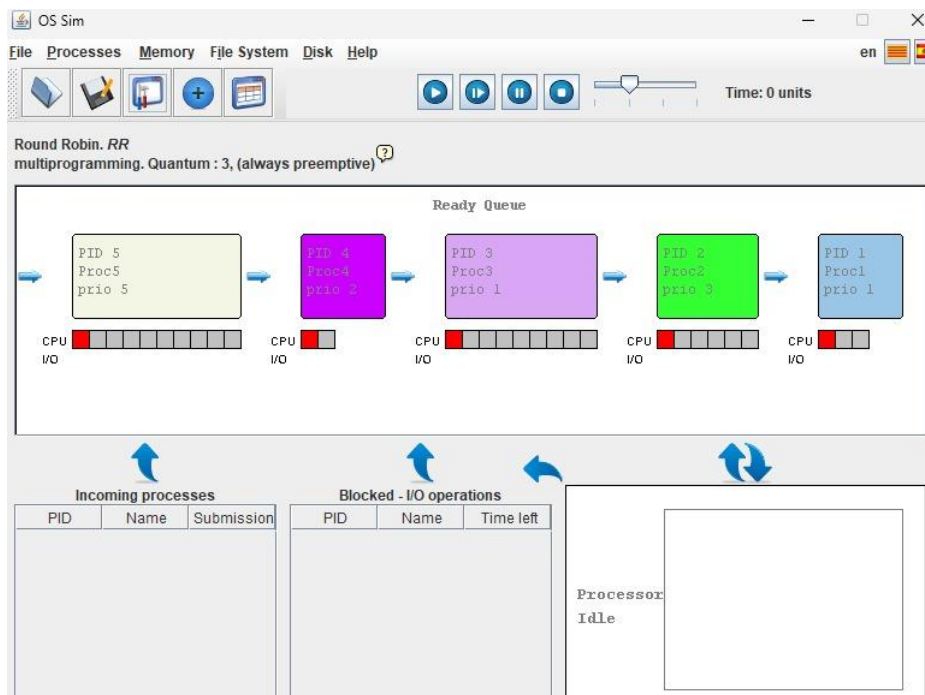
Avg. Turnaround Time (time) 19,80

Avg. Waiting Time (time) 13,80

Avg. Response Time (time) 2,00

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
4	Proc4	2	0	-	2	3	7	9	0.222222...	0.0
1	Proc1	1	0	-	3	0	8	11	0.272727...	0.0
2	Proc2	3	0	-	6	1	15	21	0.285714...	0.0
3	Proc3	1	0	-	9	2	19	28	0.321428...	0.0
5	Proc5	5	0	-	10	4	20	30	0.333333...	0.0

RR QUANTUM 3



Process Scheduling Information										
Efficiency (%)		1,00								
Throughput (processes/time unit)		0,17								
Avg. Turnaround Time (time)		17,40								
Avg. Waiting Time (time)		11,40								
Avg. Response Time (time)		5,80								
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	Proc1	1	0	-	3	0	0	3	1.0	0.0
4	Proc4	2	0	-	2	9	9	11	0.1818181...	0.0
2	Proc2	3	0	-	6	3	11	17	0.3529411...	0.0
3	Proc3	1	0	-	9	6	17	26	0.3461538...	0.0
5	Proc5	5	0	-	10	11	20	30	0.3333333...	0.0

2. O que é possível inferir, em termos comparativos, quanto às políticas/algoritmos de escalonamento, considerando os critérios utilização da CPU, tempo de *turnaround* médio, tempo médio de espera e tempo médio de resposta? Justifique.

É possível observar através dos testes que as configurações feitas em cada processo dentro de cada tipo de escalonamento definem diretamente como o processo vai ser executado, em que momento e qual o tempo que levará para se concluir cada processo (throughput) e em geral, o processo todo do início ao fim (nesse caso, os 5 processos inseridos e configurados no programa). Outro ponto muito interessante é a questão da eficiência de cada processo que é interferido diretamente pelo tipo de escalonamento que está sendo usado e quantos processos vão ter que rodar até que se finalize todos os processos em espera.

3. A partir do que foi observado, em qual contexto ou cenário (caracterização de processos – cpu e/ou io bound e quantidade de bursts) cada algoritmo parece ser mais apropriado? Justifique.

Em minha análise a partir dessa atividade, o escalonamento FCFS se enquadra menor quando se tem menos processos e estes não possuem um certo nível de categorização e prioridade que possam afetar diretamente o processo do sistema em geral. Para sistemas que usam e recebem muitos processos ao mesmo tempo, o mais adequado seria o SFJ onde pode-se definir as prioridades e as mesmas são respeitadas mediante a configuração feita no programa.

Já o RR, seja do tipo quantum 1 ou 3, possuem as configurações diretas de prioridades, porém elas sofrem alterações diretas e automáticas caso o sistema julgue necessário, não seguindo de fato as prioridades estipuladas, mas sim a categorização que o processo define.

Desta forma, percebe-se que cada processo tem sua devida funcionalidade e que o “melhor cenário” varia dependendo da necessidade do usuário final, a quantidade de processos que ele terá no sistema em uso e quais são suas prioridades de controle e de efetividade.