

Guilherme Almeida Lopes

Laboratório 04: Segurança Gerência de processo

Relatório técnico de laboratório solicitado pelo professor Rodrigo Campiolo na disciplina de Sistemas Operacionais do Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Universidade Tecnológica Federal do Paraná – UTFPR

Departamento Acadêmico de Computação – DACOM

Bacharelado em Ciência da Computação – BCC

Campo Mourão

Setembro / 2023

Resumo

Este documento descreve o resultados obtido no laboratório 04, sobre gerência de processos.

Palavras-chave: sistemas operacionais. linux. Processo.

Sumário

1	Introdução	4
2	Fundamentação	4
3	Materiais	4
4	Procedimentos e Resultados	4
	4.1 Questão 1	4
	4.2 Questão 2	7
	4.3 Questão 3	8
5	Discussão dos Resultados	9
6	Conclusões	9
7	Referências	9

1 Introdução

Neste documento consta a fundamentação do conteúdo apresentado, os materiais utilizados para a realização do laboratório, os procedimentos efetuados e seus resultados, a discussão de tais resultados obtidos e, por fim, a conclusão.

2 Fundamentação

A segurança é uma função muito crítica do sistema operacional, por determinar quais usuário tem acesso a determinados recursos, como também as auditoria, que registram todas as operações realizadas pelo sistema operacional.

A Área de segurança apresenta três grandes pilares, como citado por Carlos maziero (??), para que os sistemas de um modo geral sejam considerados seguros eles tem que ter Confiabilidade, Integridade, Disponibilidade.

A Integridade é um dos pilares fundamentais da segurança de sistemas computacionais, que consiste em buscar o estado que somente os usuários autorizados podem modificar ou destruir recursos do sistema. A Disponibilidade é outro dos pilares da segurança, que consiste em garantir que os recursos estejam disponíveis a qualquer momento pelo usuário daquele recurso. A Confiabilidade é o último dos pilares, que visa garantir que somente os usuários autorizados tem acesso aos recursos devidos.

3 Materiais

- Distribuição Linux
- Virtualbox
- Comandos do sistema

4 Procedimentos e Resultados

4.1 Questão 1

Foi simulado o algoritmo FCFS tanto no modo multiprogramação quanto no modo monoprogramação no simulador OS Sim, e suas estatísticas foram obtidas, respectivamente. Dito isso, foram desenhados os diagramas de Gantt respectivos, sendo a cor amarela o processo bloqueado por operações E/S e a cor verde o processo em execução.

Process Scheduling Information										
Efficiency (%)		1,00								
Throughput (processes/time unit)		0,20								
Avg. Turnaround Time (time)		17,7								
Avg. Waiting Time (time)		11,2								
Avg. Response Time (time)		5,00								
PID	Name	Priority	Submissi...	Periodic	CPU	Response	Waiting	Turnarou...	% CPU	% IO
5	P4	1	0	-	4	11	11	15	0.26666...	0.0
2	P2	1	0	-	6	2	10	17	0.375	0.14285...
4	P3	1	0	-	6	7	11	19	0.35294...	0.25
1	P1	1	0	-	4	0	13	20	0.23529...	0.42857...

Figura 1 – Estatísticas da simulação no modo multiprogramação.

Process Scheduling Information										
Efficiency (%)		0,77								
Throughput (processes/time unit)		0,15								
Avg. Turnaround Time (time)		17,2								
Avg. Waiting Time (time)		10,7								
Avg. Response Time (time)		10,7								
PID	Name	Priority	Submissi...	Periodic	CPU	Response	Waiting	Turnarou...	% CPU	% IO
1	P1	1	0	-	4	0	0	7	1.0	0.42857...
2	P2	1	0	-	6	7	7	14	0.46153...	0.14285...
4	P3	1	0	-	6	14	14	22	0.3	0.25
5	P4	1	0	-	4	22	22	26	0.15384...	0.0

Figura 2 – Estatísticas da simulação no modo monoprogramação.

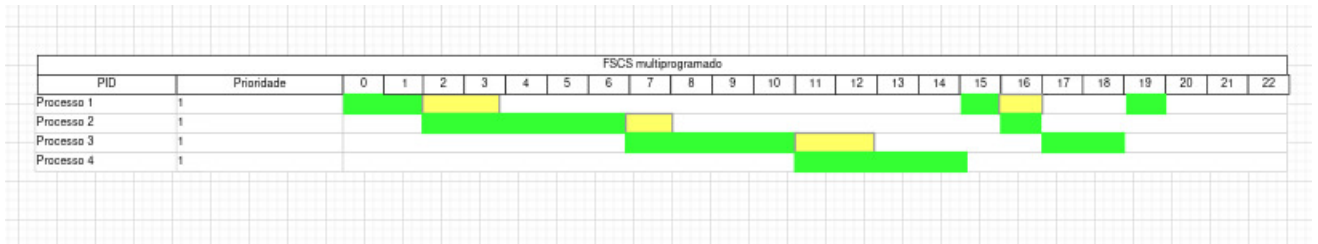


Figura 3 – Diagrama de Gantt para o modo multiprogramação.

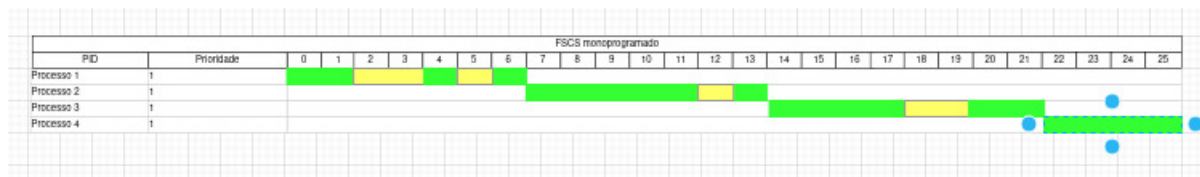


Figura 4 – Diagrama de Gantt para o modo monoprogramação.

Comparando a eficiência do sistema nas duas configurações de operações, os resultados exibem uma eficiência de 100% no modo multiprogramação, em que a CPU está sempre ocupada, enquanto no modo monoprogramação, essa eficiência é menor, com 77%. Isso ocorre pois diversos processos podem ser executados concorrentemente na multiprogramação, o que evita a inércia da CPU.

Ademais, a respeito do tempo médio de espera para ambas as execuções, não houve muita discrepância, sendo de 11,2 na multiprogramação e 10,7 na monoprogramação. No entanto, a distribuição dos tempos de espera entre os processos é consideravelmente discrepante no modo monoprogramação, variando de 0 unidades de tempo para o processo com PID 1 (P1) a 22 unidades de tempo para o processo com PID 5 (P4). Isso ocorre pois existe uma execução sequencial dos processos, o que faz com que haja um tempo de espera mais longo para os processos que estão por vir logo após. Já na multiprogramação, os processos possuem tempos de espera mais uniformes, todos próximos de 11 unidades de tempo, devido à execução "simultânea" dos processos.

Por fim, no geral, a multiprogramação exhibe aspectos de desempenho superiores à monoprogramação, com exceção do tempo de espera. No entanto, ao considerar o desempenho individual dos processos, existem processos, como é o caso do processo com PID 1 (P1), que possuem tempos de espera e de resposta extremamente baixos, sendo eles nulos. Isso ocorre pois todos os processos chegaram ao mesmo tempo. Na multiprogramação não existem valores tão baixos para cada processo, mas sim uma média geral que se aproxima de cada tempo de cada processo individual.

4.2 Questão 2

Foi simulado o algoritmo de prioridades tanto como preemptivo e não preemptivo e obtidos os resultados respectivos, desenhados nos diagramas de Gantt, sendo, novamente, a cor amarela o processo bloqueado por operações E/S, a cor verde o processo em execução e a cor vermelha os processos bloqueados pela fila.

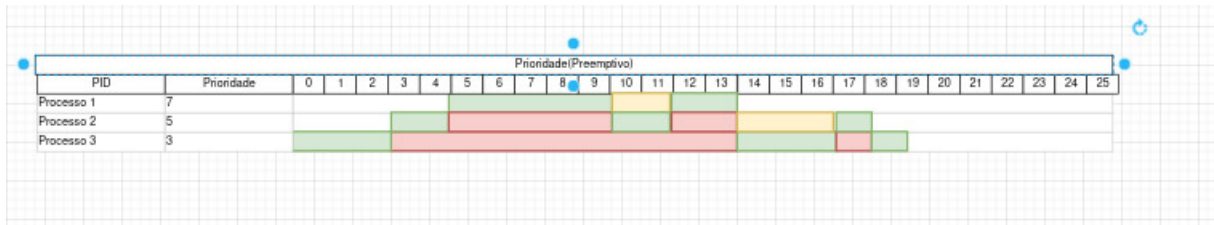


Figura 5 – Diagrama de Gantt para preemptivo.

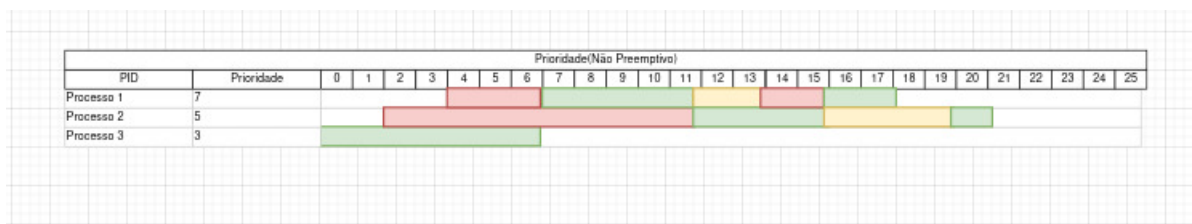


Figura 6 – Diagrama de Gantt para não preemptivo.

Além disso, as estatísticas dos tempo de resposta, de espera e de duração para P1, P2 e P3 foram obtidos para o preemptivo (Figura 7) e não preemptivo (Figura 8)

	Resposta	Espera	Duração
Processo 1	7	2	9
Processo 2	5	10	15
Processo 3	6	12	18

Figura 7 – Preemptivo.

	Resposta	Espera	Duração
Processo 1	7	6	13
Processo 2	5	13	18
Processo 3	6	0	6

Figura 8 – Não preemptivo.

Por fim, em uma análise geral, o preemptivo exibe aspectos de desempenho superiores ao não preemptivo, com exceção do tempo médio de espera, que se mostra mais elevado no preemptivo.

4.3 Questão 3

Após feita a simulação para os algoritmos FCFS - monoprogramado e multiprogramado -, SJF - preemptivo e não preemptivo -, prioridade - preemptivo e não preemptivo - e Round Robin com o quantum de 2, foram obtidas as estatísticas presentes nas Figura 9 e 10.

Algoritmo	Eficiência %	Throughput	Tempo médio	Tempo Espera	Tempo de Resposta
FCFS(monoprogramado)	73	0,15	14	7,5	7,5
FCFS(multiplicação)	100	0,21	13,75	7,25	2,25
SJF(não preemptivo)	100	0,21	11	4,5	3,5
SJF(preemptivo)	100	0,21	10,5	4	3,5
Prioridade(preemptivo)	95	0,2	11,5	5	0,75
Prioridade(não preemptivo)	100	0,21	13,5	7	1,25
Round Robin (quantum = 2)	100	0,21	12,75	6,25	1

Figura 9 – Tabela de comparação de desempenho entre os algoritmos.

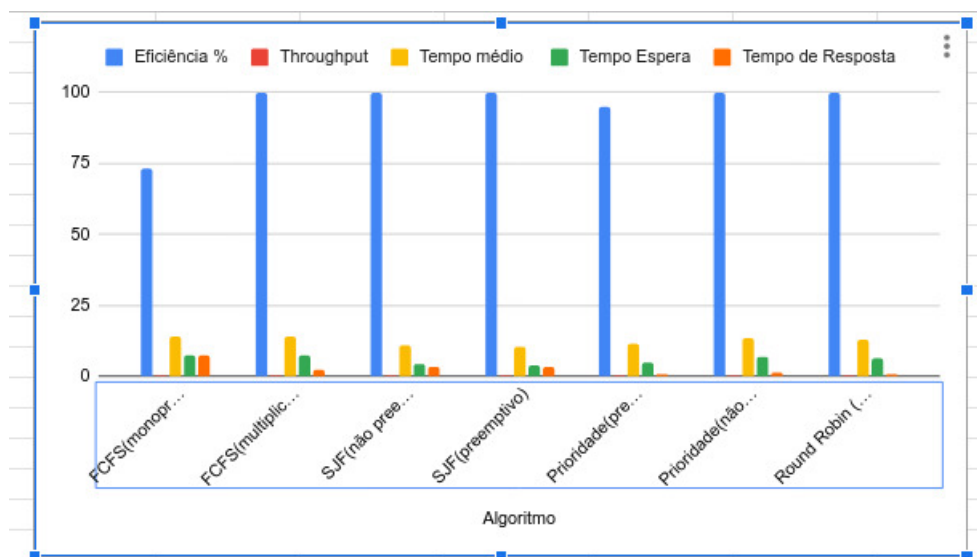


Figura 10 – Gráfico de comparação de desempenho entre os algoritmos.

Em geral, os resultados apresentados nos algoritmos mostram desempenhos aproximados, com exceção do FCFS monoprogramado. Tal algoritmo apresentou uma eficiência de 73%, enquanto quase todos os outros algoritmos apresentaram eficiência de 100%. Além disso, seu throughput foi o único abaixo de 0,21, sendo ele de 0,15. Ademais, o tempo médio, o tempo de espera e o tempo de resposta, principalmente, foram consideravelmente elevados se comparados com os outros algoritmos. Todos esses resultados ocorrem pois a CPU fica em inércia por um período, já que ela aguarda o processo em I/O terminar.

A respeito dos outros algoritmos (com exceção do FCFS monoprogramado, já comentado anteriormente), a eficiência é de 100% em todos menos no de prioridade preemptiva. Tal algoritmo também possui um throughput diferente dos outros, apesar de ser uma diferença quase desprezível, sendo ele de 0,2, enquanto o restante é de 0,21.

Já em relação aos tempos médio e de espera, os algoritmos de SJF tiveram resultados melhores, sendo o preemptivo ligeiramente superior ao não preemptivo. Porém, essas vantagens não permaneceram em relação ao tempo de resposta, os quais possuem o pior tempo de resposta entre os algoritmos, desconsiderando o FCFS monoprogramado. O algoritmo que obteve o melhor tempo de resposta foi o de prioridade, sendo o preemptivo, novamente, ligeiramente superior ao não preemptivo. Por fim, ao observar a média geral de todas as estatísticas, o Round Robin é o algoritmo que tem uma média mais próxima, sem ter muita discrepância entre os resultados.

5 Discussão dos Resultados

Após análises aprofundadas das estatísticas de diversos algoritmos de escalonamento, e suas comparações realizadas, foi possível verificar que, para otimizar o desempenho de um sistema, é preciso considerar qual dos algoritmos atende melhor às demandas, levando em conta se os processos são I/O-bound ou CPU-bound.

6 Conclusões

Portanto, torna-se evidente como é importante ter pleno conhecimento dos algoritmos e seus resultados na CPU para, assim, entender quais algoritmos obtêm melhores resultados em situações específicas.

7 Referências