Resp P1 SISTEMAS OPERACIONAIS 2024

1. (1,0 pontos) Cite duas diferenças entre os threads de nível de usuário e os de nível de kernel. Sob que circunstâncias um tipo é melhor do que o outro?

(1) Threads de nível de usuário são desconhecidos pelo kernel, enquanto o kernel está ciente dos threads de nível de kernel.
(2) Em sistemas que utilizem mapeamento tanto M:1 quanto M:N, os threads de usuário são alocados ao schedule pela biblioteca de threads e o kernel organiza o schedule dos threads de nível de kernel.
(3) Os threads do kernel não precisam estar associados a um processo, enquanto todo thread de usuário pertence a um processo.
Os threads do kernel são geralmente mais dispendiosos para manter do que os threads de usuário, já que eles devem ser representados com uma estrutura de dados do kernel.

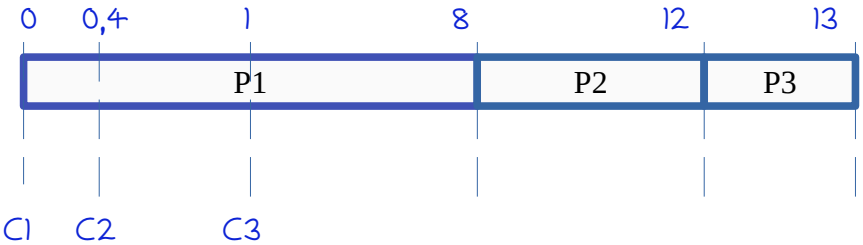
2. (1,0 ponto) Explique a diferença entre scheduling com e sem preempção.
O scheduling preemptivo permite que um processo seja interrompido no meio de sua execução, tomando a CPU e alocando-a a outro processo. O scheduling não preemptivo garante que um processo deixe o controle da CPU somente quando terminar o seu pico de CPU corrente.

3. (2,0 pontos) Suponhamos que os processo a seguir chegassem para execuação nos tempos indicados. Cada processo será executado durante o período de tempo listado. Ao responder às perguntas, use o scheduling sem preempção e baseie as decisões nas informações disponíveis no momento em que a decisão tiver de ser tomada.

Processo	Tempo de Chegada	Duração do Pico
P_1	0.0	8
P_2	0.4	4
P_3	1.0	1

a) Qual o tempo médio de turnaround desses processos com o algoritmo de scheduling FCFS?
b) O algoritmo SJF deveria melhorar o desempenho, mas observe que optamos por executar o processo P 1 no mo- mento 0 porque não sabíamos que dois processos mais curtos estavam para chegar. Calcule qual será o tempo médio de turnaround se a CPU for deixada ociosa du- rante a primeira unidade de tempo 1 para então o scheduling SJF ser usado. Lembre que os processos P1 e P2 estão esperando durante esse tempo ocioso e, portanto, seu tempo de espera pode aumentar. Esse algoritmo poderia ser chamado de scheduling de conhecimento futuro.

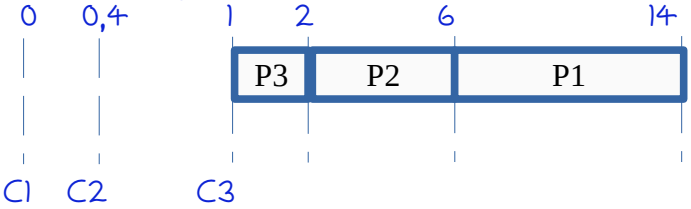
a) FCFS
Assim:



$TU_{P1} = 8 - 0 = 8$
$TU_{P2} = 12 - 0,4 = 11,6$
$TU_{P3} = 13 - 1 = 12$

$TU_{médio} = (8+11.6+12)/3 = 10,53$

b)Scheduling de conhecimento futuro. Estamos sabendo que P3 tem o pico menor



TU_{P1} = 14 - 0 =14
TU_{P2} = 6 - 0,4 = 5,6
TU_{P3} = 2-1 = 1

$TU_{médio}$ = (14+5.6+1)/3 = **6,86**

4. (1,0 ponto) Como em tempo de programação pode-se verificar se um processo é o processo pai ou processo filho?
Se a variável identificadora do processo criado for igual a zero então é o processo filho.

5. (1,0 ponto) Quais as diferenças entre o multiprocessamento simétrico e assimétrico? Explique as diferenças quanto a arquitetura.

Simétrico	Assimétrico
Cada processador executa as tarefas no SO	Apenas o processador mestre executa as tarefas do SO
Processadores com mesma arquitetura	Arquiteturas iguais ou diferentes
Cada processador pode executar processos tarefa conforme uma fila pronta ou privada	O processador mestre define as tarefas aos escravos
Comunicação por memória compartilhada	Não há necessidade de se comunicar, pois são controlados pelo Mestre
Se um processador falhar a capacidade do sistema é reduzida	Se um processador falhar o mestre seleciona um escravo para substituir
Processadores precisam ser sincronizados por motivo de equilíbrio de carga. Evitar ociosidade.	Quem controla a estrutura de dados é o Mestre.

6. (2,0 pontos) Explique a diferença entre uma chamada de sistema (syscall) e uma API.

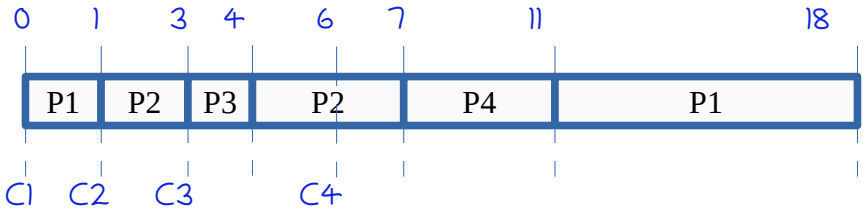
Na maioria das linguagens de programação, o sistema de suporte ao tempo de execução (um conjunto de funções que faz parte das bibliotecas incluídas com o compilador) fornece uma interface de chamadas de sistema que serve como uma ponte para as chamadas de sistema disponibilizadas pelo sistema operacional.
A interface de chamadas de sistema intercepta as chamadas de função da API e invoca as chamadas de sistema necessárias dentro do sistema operacional. Normalmente, um número é associado a cada chamada de sistema, e a interface de chamadas de sistema mantém uma tabela indexada de acordo com esses números. A interface de chamadas de sistema invoca, então, a chamada de sistema desejada no kernel do sistema operacional e retorna o status da chamada de sistema e quaisquer valores de retorno.

7. (2,0 pontos) Construa o diagrama de Gant para o algoritmo SFJ preemptivo e não preemptivo. Apresente o tempo de espera de cada processo.

Processo	Tempo de Chegada	Duração do Pico
P_1	0	8
P_2	1	5
P_3	3	1
P_4	6	4

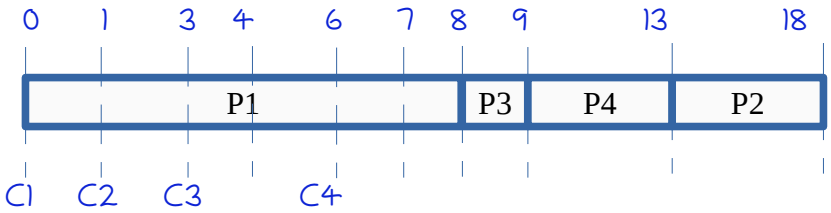
TE = TF - TC - TAM

SJF preemptivo



T1 = 18 - 0 - 8 = 10
T2 = 7 - 1 - 5 = 1
T3 = 4 - 3 - 1 = 0
T4 = 11 - 6 - 4 = 1

SJF não preemptivo



T1 = 8 - 0 - 8 = 0
T2 = 18 - 1 - 5 = 12
T3 = 9 - 3 - 1 = 5
T4 = 13 - 6 - 4 = 3