



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Sistemas Operacionais – Lista III
Professor: Wellington Franco

Data: 30/06/2021 – 07/07/2021

Nome: _____

- 1) Nos sistemas operacionais, o escalonamento de processos consiste em:
 - a. Priorizar o processo a ser executado.
 - b. Selecionar um processo da fila de ready e alocar a CPU para o mesmo.
 - c. Alterar a ordem dos processos para utilização da CPU e demais recursos.
 - d. Transferir um processo na fila de wait para a fila de ready.
 - e. Executar processos mais demorados antes dos mais rápidos.

- 2) São benefícios de threads, exceto:
 - a. Melhora a capacidade de resposta de uma aplicação
 - b. Facilita o compartilhamento de recurso no processo
 - c. Escalabilidade do sistema com a utilização dos multiprocessadores
 - d. Apesar de facilitar o compartilhamento dos recursos, não ocorre economia desses recursos pelo SO.

- 3) Com relação aos modelos de threads de usuários e kernel, é incorreto afirmar que no modelo muitos-para-um:
 - a. Mapeia muitos threads do usuário para um do kernel
 - b. Gerenciamento é feito pela biblioteca de threads no espaço do usuário, sendo eficiente
 - c. Várias threads podem acessar o kernel por vez, assim ocorre paralelismo
 - d. O processo pode ser bloqueado se um thread fizer um system call bloqueadora

4) Com relação aos modelos de threads de usuários e kernel, é incorreto afirmar que no modelo um-para-um:

- a. Mapeia cada thread do usuário para um do kernel
- b. Fornece mais concorrência, uma vez que cada thread usuário interage com um de kernel, assim pode ter system call bloqueadora.
- c. Permite que várias threads sejam executadas em paralelo.
- d. Apesar cada thread de usuário necessitar da criação de um thread de kernel, não é custoso para o SO.

5) Com relação aos semáforos, é incorreto afirmar que:

- a. Os semáforos de contagem podem ser usados para controlar o acesso a um determinado recurso composto por uma quantidade finita de instâncias.
- b. O semáforo é inicializado com a quantidade de recursos disponíveis.
- c. Cada processo que desejar usar um recurso, executa uma operação de wait()
- d. Quando o processo libera um recurso, ele executa uma operação de signal()
- e. Se a contagem for N (chegou no final), todos os processos ficam bloqueados esperando a liberação do recurso.

6) O problema de starvation pode acontecer nos semáforos, quando os processos obedecem que regra na fila de espera:

- a. FIFO (First in, First out)
- b. LIFO (Last In, First out)
- c. SJF (Menor esforço)
- d. Round Robin
- e. Por prioridade

7) O SO utiliza o algoritmo FCFS,

Processo	Duração do pico
P1	20

P2	21
P3	15
P4	19

Os processos chegam na ordem P1, P2, P3 e P4. Calcule o tempo médio de turnaround.

8) O SO utiliza o algoritmo SJF com preempção,

Processo	Duração do pico	Tempo de chegada
P1	20	1
P2	11	2
P3	15	3
P4	19	5

Calcule o tempo médio de turnaround.

9) O SO utiliza o algoritmo de Por prioridade sem preempção,

Processo	Duração do pico	Tempo de chegada	Prioridade
P1	16	5	1
P2	23	4	1
P3	22	3	3
P4	18	2	2

O processo com maior prioridade é o que tem o número menor de prioridade. Calcule o tempo médio de espera.

10) O SO utiliza o algoritmo Round-Robin, quantum=2,

Processo	Duração do pico	Tempo de chegada
P1	18	1
P2	16	4
P3	15	6
P4	22	7

Calcule o tempo médio de espera.

11)

Cinco processos são criados na seguinte ordem: P1 , P2 , P3 , P4 e P5, com os seguintes tempos:

Processo	Tempo de Serviço	Prioridade	Tempo de chegada
<i>P1</i>	13	3	0
<i>P2</i>	11	4	4
<i>P3</i>	7	1	5
<i>P4</i>	8	2	7
<i>P5</i>	16	5	10

Ilustre a execução dos processos através de um diagrama usando os seguintes esquemas de escalonamento:

- (a) FIFO
- (b) SJF
- (c) prioridade (número de prioridade menor implica prioridade maior)
- (d) circular com fatia de tempo = 4 u.t.

Desconsidere E/S ou tempo de escalonamento ou troca de contexto entre processos.

Mostre os tempos de **turnaround** individuais por processo e a média resultante.