

2ª. Lista de Exercícios (cedida pela Profa. Roberta Lima Gomes)**Período: 2008/1****Prof^ª. Patrícia Dockhorn Costa****email: pdcosta@inf.ufes.br**

Data de Entrega: não há. O objetivo da lista é ajudar no estudo individual dos alunos. Soluções de questões específicas poderão ser discutidas em sala de aula, conforme interesse dos alunos.

- 1) Qual a relação entre programa e processo?
- 2) O que é o BCP? Qual é o seu conteúdo típico?
- 3) O que é a “imagem” de um processo?
- 4) Defina interrupção. Quais os tipos de interrupção? Como se dá o processamento de uma interrupção? Qual a relação entre interrupção e multiprogramação?
- 5) Explique as funções dos escalonadores de curto, médio e longo prazo.
- 6) Defina *turnaround time* e *throughput*.
- 7) O que significa um processo sofrer preempção?
- 8) A maioria dos escalonadores Round Robin usa um quantum de tamanho fixo. Dê um argumento em favor de um quantum pequeno. Agora pense em um argumento que justifique um quantum grande.
- 9) Considere o seguinte algoritmo de alocação por prioridade, preemptivo, baseado em prioridades que mudam dinamicamente. Números de prioridades maiores indicam prioridades mais altas. Quando um processo está esperando para entrar em execução (na fila de prontos), sua prioridade muda segundo uma taxa α ; quando está em execução, sua prioridade muda segundo uma taxa β . Todos os processos têm prioridade 0 quando entram na fila de prontos. Valores diferentes para os parâmetros α e β podem determinar muitos algoritmos de alocação diferentes.
 - a) Qual algoritmo é obtido com $\alpha > \beta > 0$?
 - b) Qual algoritmo é obtido com $\alpha < \beta < 0$?
 - c) $\alpha = -\beta$
- 10) Como funciona o escalonamento por múltiplas filas com realimentação? Qual a relação entre o quantum e o nível de prioridade de cada fila nesse esquema?
- 11) Qual dos métodos de escalonamento discutidos em sala de aula poderia ser modificado para acomodar alguns processos de tempo real (processos que *devem* ter uma resposta dentro de um

certo período de tempo) misturados com os outros? Para aqueles métodos que você poderia modificar, explique como você faria isso. Para aqueles que isso não é possível, explique o porquê.

- 12) Cinco processos, de A até E, chegam ao computador ao mesmo tempo. Eles têm seus tempos de processamento estimados em 10, 6, 2, 4 e 8 minutos respectivamente. Suas prioridades (atribuídas externamente) são 3, 5, 2, 1 e 4, respectivamente, sendo 5 o representante da prioridade mais alta. Nenhum dos processos faz I/O. Para cada um dos algoritmos de escalonamento abaixo, determine o tempo médio de turnaround dos processos. Ignore o overhead causado pela troca de contexto.
- (a) Round Robin
 - (b) Escalonamento com prioridade
 - (c) FIFO (ordem de execução: 10, 6, 2, 4, 8)
 - (d) SJF
- Para (a), assuma que a fila começa em A, indo em ordem até E (quantum = 4).

- 13) Suponha que os processos seguintes fiquem prontos para execução nos tempos indicados:

Processo	Duração da fase de uso da UCP	Tempo de Chegada
P1	8	0,0
P2	4	0,4
P3	1	1,0

- a) Desenhe diagramas de Gantt ilustrando a execução destes processos usando os algoritmos de alocação FCFS, SJF e STRF (SJF preemptivo).
 - b) Qual o tempo de processamento médio para esses processos em cada um desses algoritmos?
- 14) Considere um sistema operacional cuja máquina de estados inclui os estados *Ready* e *Ready*, *Suspended*. Suponha que seja hora do S.O. despachar um processo e que existam nesse momento processos tanto no estado *Ready* como no estado *Ready*, *Suspended*, e que pelo menos um processo no estado *Ready*, *Suspended* possui prioridade maior do que qualquer processo no estado *Ready*. Duas políticas extremas seriam: (a) sempre despachar um processo no estado *Ready*, de forma a minimizar *swapping*; e (b) sempre dar preferência ao processo de mais alta prioridade, mesmo que isso possa significar a ocorrência de *swapping* quando este não é necessário. Sugira uma política intermediária (explique e crie um algoritmo) que tente balancear prioridade e desempenho.
- 15) Considere um sistema que possui duas filas de escalonamento, com prioridades 0 e 1, sendo que somente pode ser escalonado um processo da fila de prioridade 1 não existindo processos na fila de prioridade 0. Sabendo que o algoritmo utilizado nas duas filas é o Round-Robin, escreva o pseudo-código dos procedimentos *insere(p)*, onde p é o índice da tabela de descritores de processos e que possui um campo que contém a prioridade dos processos e *r= seleciona()*, que retorna o índice da tabela de descritores que descreve o processo selecionado. Cite duas situações em que cada procedimento é chamado.
- 16) No UNIX, um processo pode encontrar-se no estado *Kernel Running* enquanto o sistema pode apresentar dois contextos de execução, *process context* e *system context*. Explique a diferença entre eles.
- 17) Explique porque normalmente os Sistemas UNIX priorizam processos I/O-bound em relação aos processos CPU-bound.
- 18) Em algumas implementações do UNIX, o kernel é não-preemptivo. O que isto significa? Quais as vantagens e desvantagens desta abordagem?