Universidade Federal de Viçosa *campus* Rio Paranaíba
Sistemas de Informação 2022-1
SIN351 - Sistemas Operacionais
Prof. Rodrigo Moreira

Exercícios Processos & Escalonamento

1. (Adaptado de Tanenbaum) Cinco processos, A a E, chegam a uma CPU ao mesmo tempo. Eles têm tempos de execução estimados em 10, 6, 2, 4 e 8 minutos. Suas prioridades são 3, 5, 2, 1 e 4, respectivamente, sendo 5 a prioridade mais alta. Para cada um dos seguintes algoritmos de escalonamento, determine o tempo médio de retorno. Considere a sobrecarga de chaveamento de 3 segundos.

   - *Round Robin* com quantum de 2 minutos
   - *Prioridades* com quantum de 2 minutos

2. Considere o seguinte conjunto de processos, com o tempo dado em segundos. Considere que os processos chegaram na ordem P1, P2, P3, P4, P5, nos tempos 1, 3, 7, 7, 12. Quantum=2. Sobrecarga de chaveamento = 1. Determine o tempo médio de resposta e de retorno para cada algoritmo:

Processo	Execução
P1	10
P2	1
P3	2
P4	1
P5	5

3. Considerando que um escalonador do tipo *Round-Robin* é implementado através de uma lista encadeada onde cada elemento da lista *ready* é um ponteiro para um descritor de processos. Responda :
   - (A) Qual o efeito de se colocar dois elementos na lista *ready* com ponteiros para um mesmo descritor processo ?
   - (B) Quais poderiam ser as vantagens e desvantagens deste esquema?
   - (C) Como você modificaria o comportamento «normal» do algoritmo Round Robin para obter o mesmo efeito sem duplicar os ponteiros na lista de *ready*?

4. Um tipo de escalonador empregado em sistemas operacionais é o de múltiplas filas com realimentação (filas multinível). A ideia neste caso é classificar os processos de acordo com suas diferentes necessidades de *CPU* e de *I/O* (*CPU bound* e *I/O bound*) criando diversas filas com prioridades diferentes. Se um processo utiliza muito tempo de *CPU* ele é transferido de uma fila de mais alta prioridade para uma fila de mais baixa prioridade.
   Responda:

- (A) Este esquema privilegia os processos *I/O bound* ? Em caso afirmativo, qual a argumentação (justificativa) para este tipo de privilégio ?
- (B) Neste esquema existe a probabilidade de um processo *CPU bound* sofra «starvation» (postergação infinita) ? Caso esta possibilidade seja real como pode ser resolvido este problema ?
- (C) Qual critério é empregado para classificar um processo nas diversas filas quando ele é criado ?
- (D) Como o sistema operacional pode determinar se um processo é *CPU bound* ou *I/O bound*, movendo-o de uma fila para outra, se esta característica só pode ser conhecida em tempo de execução ?

5. Considere o seguinte conjunto de 5 processos onde chegada é o instante de tempo que o processo se tornou apto pela primeira vez, t é o tempo necessário a execução do processo (tempo total de CPU) e p a sua prioridade.

Processo	Chegada	Tempo de CPU	Prioridade
P_0	0	60	9
P_1	15	25	10
P_2	15	15	9
P_3	65	15	10
P_4	70	10	11

Assumindo que a execução inicie no tempo zero, desenhe o diagrama de execução desses 5 processos, isto é, quem está ocupando a *CPU* em cada instante de tempo, considerando as seguintes políticas de escalonamento:

- FIFO não preemptivo
- SJF não preemptivo
- Round Robin (quantum=10, não preemptivo por prioridade)
- Múltiplas filas, preemptivo por prioridades, com política FIFO em cada fila.

Supor que processos com prioridades de valores numéricos menores são os mais prioritários. Em caso de empates, considerar como critério de desempate o pid do processo (o processo de menor *pid* ganha a disputa.

6. Quando uma interrupção ou uma chamada de sistema transfere o controle para o sistema operacional, uma área de pilha no kernel, separada da pilha do processo interrompido, é geralmente usada. Por quê?