Sistemas Operacionais I – 2010/1

Gabarito da 1ª. Prova

Questão 1

- Definir corretamente o que é a técnica de escalonamento de processos SJF (*Shortest Job First*)
    - Técnica de escalonamento em que a fila de processos prontos para execução é organizada de tal forma que os processos de menor expectativa de tempo de CPU (menor tamanho, menor CPU burst) são colocados no início da fila, antes dos processos mais longos (maior tempo de execução, maior CPU burst). Com essa estratégia consegue-se obter uma diminuição razoável do tempo médio de espera na fila.
    - É uma técnica de escalonamento não preemptiva.
    - *Starvation* de processos longos pode ocorrer nesta técnica.
- Associar corretamente a técnica com o *throughput* e *turnaround*
    - *Throughput*: alto (processos curtos terminam rapidamente, dando espaço para a execução de outros processos).
    - *Turnaround*: médio (dependendo da existência de processos grandes na fila e devido à característica não-preemptiva da técnica, o *turnaround* médio pode não ser necessariamente pequeno).

Questão 2

- Avaliar corretamente a influência do tamanho do quantum e tirar uma conclusão geral sobre o problema colocado:
    - Quantum grande: tende a FIFO, resultando em um baixo grau de multiprogramação.
    - Quantum pequeno: apresenta um alto grau de multiprogramação; porém, provoca grande overhead devido às inúmeras trocas de contexto.
    - Em geral, um bom tamanho de quantum deve durar, em média, o tempo necessário para que o processo realize I/O. O valor não deve ser muito grande e nem muito pequeno, de forma a reduzir o *turnaround* médio e obter um bom *throughput*.

Questão 3

- Definir corretamente o que é a SVC (Supervisor Call)
    - Interface padronizada através da qual os processos têm acesso aos serviços fornecidos pelo núcleo do sistema operacional. Constitui uma forma segura e controlada de executar instruções especiais e de realizar operações que envolvam estruturas de dados do *kernel* em benefício de processos de usuário.

- Relacioná-las com os modos de operação da UCP (*user mode / kernel mode*)
    - As SVCs executam em modo *kernel*, com acesso irrestrito ao espaço de endereçamento dos processos de usuário e à área de *kernel*. Pode executar instruções privilegiadas e acessar todas as estruturas de dados de *kernel* necessárias para a execução do serviço solicitado. Ao impedir processos de usuário rodando em *user mode* de acessar tais instruções e estruturas de dados, as SVCs determinam uma forma segura e controlada de se acessar serviços que envolvam o uso de instruções especiais e a realização de operações que manipulem estruturas de dados do *kernel* em benefício dos processos de usuário.

Questão 4

- Explicar o motivo do escalonador tradicional não favorecer processos CPU bound:
    - A explicação deve fazer referência à fórmula usada no cálculo das prioridades dos processos pelo escalonador tradicional do Unix. É a partir da observação e análise dos termos que compõem a fórmula, em particular do termo *p_cpu*, que se pode concluir que os processos CPU-bound não são favorecidos pelo esquema. Este termo é inversamente proporcional à prioridade de escalonamento: quanto maior o seu valor menor é a prioridade de escalonamento do processo. O termo *p_cpu* reflete o uso de CPU pelos processos: quanto mais um processo usa a CPU maior é o seu valor. Pode-se então concluir que quanto mais o processo usa a CPU menor é a sua chance de ser escalonado pois os processos na fila de prontos e nas filas de espera têm o seu valor de *p_cpu* crescendo a uma taxa bem menor do que o processo em execução.
    - A cada 4 CPU ticks a rotina de interrupção do relógio incrementa o valor de *p_cpu* do processo em execução e a cada 100 CPU ticks a prioridade de todos os processos é recalculada. Em função do recálculo de prioridades pode haver troca de contexto. Isso previve o starvation e favirece processos I/O bound.

Questão 5

- Desenhar corretamente a árvore de processos (vide adiante)
    - A árvore de processos conta com 16 processos: 1 processo pai e 15 outros processos gerados a partir da aplicação da SVC *fork()*.
    - Ao final da primeira iteração (i=0):
        - A árvore conta com 4 processos.
        - São impressos duas vezes a palavra Sistemas e outras duas vezes a palavra Operacionais.
    - Ao final da segunda iteração (i=1):
        - A árvore conta com 16 processos.
        - São impressos mais oito vezes a palavra Sistemas e mais oito vezes a palavra Operacionais, totalizando 10 impressões de cada uma dessas palavras.

Questão 6

- a) O processo A sofre preempção em dois instantes: *(i)* 24 para 25; e *(ii)* 59 para 60.
- b) O processo B, por ser mais prioritário, não sofre preempção em nenhum momento.

Questão 7

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**
P1	E	E	E	E	E	B	B	B	B	B	B	B	B	B	B	P	P	P	P	E	E	E	E	E	B	B	B	B	B	B	B	B	B	B	E	E	E	E
P2	P	P	P	P	P	E	E	E	E	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P3	P	P	P	P	P	P	P	P	P	E	E	E	E	E	E	E	E	E	E	P	P	P	P	P	E	E	-	-	-	-	-	-	-	-	-	-	-	-

E=Executando, P=Pronto, B=Bloqueado

- a) T=8: P1=Bloqueado, P2=Executando, P3=Pronto
- b) T=18: P1=Pronto, P2=Terminado, P3=Executando
- c) T=28: P1=Bloqueado, P2=Terminado, P3=Terminado

Árvore de Processos da Questão 5: