

Questão 1

atribuição de créditos

	Tempo (ms)																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	R1	B1	B1	B1	B1	P1	P1	R0	B2	B2	B2	B2	P2	P2	R1	T						
B	P2	R1	R0	B0	B0	B0	B0	B0	B2	P2	P2	P2	R1	R0	T							
C	P2	P2	P2	R1	R0	P0	P0	P0	R1	R0	P0	P0	P0	P0	P0	R1	R0	T				
D	P2	P2	P2	P2	P2	R1	R0	P0	P2	P2	R1	R0	P0	P0	P0	P2	P2	R1	R0	T		

$$T_A = 15 \quad T_B = 14 \quad T_C = 17 \quad T_D = 19$$

$$\bar{T} = \frac{15 + 14 + 17 + 19}{4} = 16,25$$

- vantagem p/ processos I/O bound

Questão 2

a) Falso: um algoritmo de escalonamento é preemptivo justamente por, em algumas situações, forçar a preempção do processo em execução fazendo com que o mesmo perca a posse da CPU. Por exemplo, quando um processo mais prioritário se torna "pronto", ou qdo o quantum do processo em execução finaliza.

b) Verdadeiro: algoritmos de escalonamento preemptivos oferecem melhores tempos de resposta já que processos em execução podem ser preemptados. Mas vale ressaltar que um algoritmo preemptivo não é suficiente para oferecer garantias de resposta (se o kernel não for preemptivo)

c) Falso: o mais importante é implementar um algoritmo de escalonamento preemptivo, oferecendo tempos de resposta médio baixo. Não há necessidade de garantias de tempo de resposta (o que seria obtido com um kernel preemptivo)

d) Verdadeiro: caso um processo com prioridade 10 ganhe a posse da CPU, e nesse meio tempo um processo de prioridade 5 (esquema onde prioridade 5 > prioridade 10) se torna "pronto", este último (mais prioritário) terá que esperar o primeiro (menos prioritário) liberar a CPU

Questão 3

- 1) Um processo é criado devido a uma requisição do usuário, de um outro processo, ou do próprio Kernel
- 2) O processo recém criado é "admitido" pelo escalonador de longo prazo (quando há recursos suficiente, como memória)
- 3) O processo é escalonado para executar
- 4) O processo é "preemptado", perde a CPU para um processo mais prioritário, ou fim de quantum
- 5) Processo muda modo de execução par Kernel mode ~~para~~ devido a: (i) SVC; (ii) ocorrência de exceção; (iii) ocorrência de interrupção
- 6) O processo retorna p/ modo usuário ao fim de: (i) SVC; (ii) exceção; (iii) Tratamento de interrupção
- 7) O processo em execução é bloqueado para esperar a ocorrência de um evento; processo permanece em memória.
- 8) O evento pelo qual o processo estava esperando ocorreu, retornando à fila de prontos; processo permanece em memória
- 9) O escalonador de médio prazo transfere da memória p/ área de swap (disco) o processo bloqueado, suspendendo este processo
- 10) O evento pelo qual um processo bloqueado e suspenso ocorreu, tornando este processo pronto, mas o processo permanece na área de swap

- 10) Um processo bloqueado e suspenso é trazido novamente para memória mas ainda está bloqueado. Isso pode ocorrer para ~~processos~~ que, na ocorrência do evento pelo qual o processo espera ele entre mais rapidamente no estado "pronto".
- 12) Um processo no estado pronto é suspenso, sendo transferido da memória para área de swap. Isso ocorre porque o escalonador de médio prazo precisa liberar memória e certamente não há mais processos bloqueados em memória.
- 13) Um processo pronto e suspenso é trazido novamente para memória pelo escalonador de médio-prazo visto que há memória disponível.
- 14) Um processo é finalizado (i) normalmente, ou (ii) devido à ocorrência de um erro ou sinal.

Questão 4

