

1ª. Prova de Sistemas Operacionais I / Sistemas de Programação II – 2009/2

Aluno: .....

- 1) (3,0) O sistema operacional Linux utiliza dois algoritmos de escalonamentos, um de tempo compartilhado e outro baseado em prioridades, para tratar tarefas de tempo real. Cada processo é associado a uma classe de escalonamento. No algoritmo de escalonamento para processos de tempo compartilhado, o Linux usa um algoritmo por prioridades, baseado em créditos. Cada processo possui um determinado número de créditos (inicialmente, o número de créditos é igual à prioridade do processo); o processo com o maior número de créditos na fila de prontos é selecionado pelo escalonador. A cada interrupção do temporizador (1ms), o processo em execução perde um crédito; quando seu crédito chega a zero, o escalonador é ativado para selecionar outro processo para ganhar o processador. Se nenhum processo na fila de prontos tiver créditos, o algoritmo faz nova atribuição de créditos a todos os processos (inclusive aos processos bloqueados), de acordo com a seguinte regra:  $\text{créditos} := \text{créditos} / 2 + \text{prioridade}$ .

Avalie o algoritmo de escalonamento por prioridade usado pelo Linux quanto ao tempo total de execução dos processos (turnaround). Para tanto, considere o seguinte volume de trabalho (processos de A a D chegam no sistema ao mesmo tempo):

Ordem	Processo	Surto de CPU*	Duração de I/O	Tempo total de CPU	Prioridade
1	A	1	4	3	2
2	B	2	6	4	2
3	C	---	---	6	2
4	D	---	---	6	2

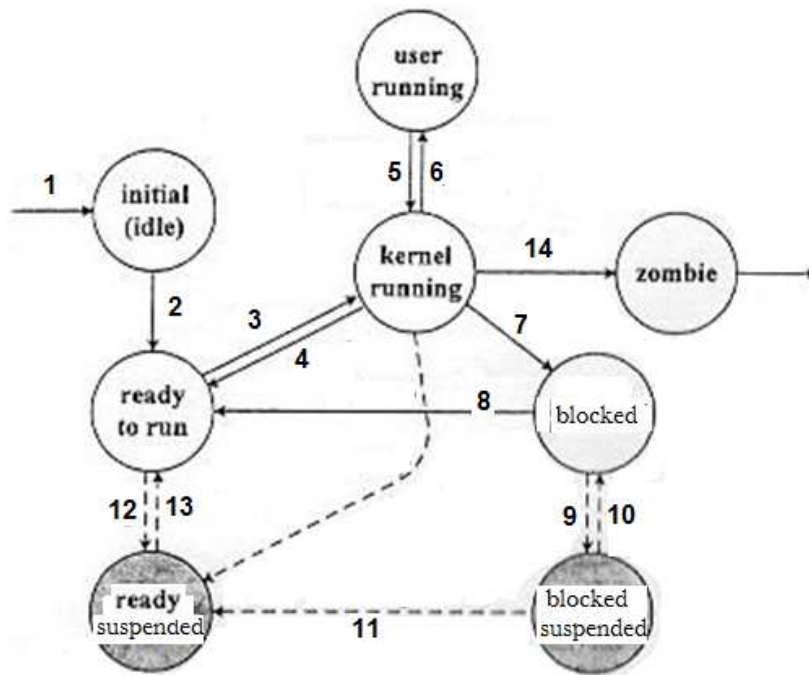
(\*) tempo de CPU necessário antes de cada solicitação de I/O (processos A e B ficam alternando entre surtos de CPU e em operações de I/O).

Ilustre usando o diagrama abaixo a evolução no tempo do (i) estado e do (ii) número de créditos de cada processo. Utilize a seguinte nomenclatura de estados: *Pronto (P)*; *Rodando (R)*; *Bloqueado (B)*; *Terminado (T)*. Por exemplo: - “P2” indica que o processo está no estado *Pronto* com 2 créditos.  
- “R0” indica que o processo está no estado *Rodando* consumindo o seu último crédito

	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																						
B																						
C																						
D																						

- 2) (2,0) Quais das afirmações seguintes são verdadeiras e quais são falsas? Justifique/explique suas respostas.
- Um algoritmo de escalonamento é preemptivo se o processador não puder ser retirado à força de um processo.
  - Sistemas de tempo real geralmente usam escalonamento de processador preemptivo.
  - Sistemas de tempo compartilhado geralmente necessitam que o kernel do S.O. seja preemptivo.
  - Uma deficiência do algoritmo de escalonamento por prioridade não-preemptivo é a inversão de prioridades.

3) (2,0) Dado o diagrama abaixo, escreva e explique os eventos que provocam cada uma das transições entre estados.



4) (3,0) Faça o Diagrama de Precedência para o seguinte código:

```

c2 = 0;
c1 = fork();          /* fork number 1 */
if (c1 == 0)
    c2 = fork();     /* fork number 2 */
else
    wait(c1);
if (c2 > 0)
    fork();          /* fork number 3 */
exit();
  
```

*Boa Prova!*