GABARITO
P2 - Sistemas Operacionais I
Professor: Leandro Marzulo
2012-2

Nome:

Instruções: Esta prova é composta de duas questões totalizando 12 (doze) pontos, sendo a nota máxima 10 (dez). Responda as questões de forma sucinta e clara. O uso de lápis é permitido, no entanto, pedidos de revisão serão considerados apenas para questões respondidas a caneta. BOA PROVA!

1) (6,0) Considere cinco processos com as características descritas na tabela a seguir:

Processo	Tempo da Rajada de CPU 1	Tempo da Rajada de I/O	Tempo da Rajada de CPU 2	Prioridade	Instante de criação
P1	7	12	3	5	3
P2	6	5	5	1	4
P3	10	-	-	2	0
P4	7	100	9	4	20
P5	8	-	-	3	80

Considere que cada processo faz I/O em um dispositivo independente (todos os I/Os são paralelos) e que o tempo de troca de contexto é insignificante. Saiba que, para cada processo:

Tempo de Turnaround = Tempo de Execução no processador + Tempo de I/O + Tempo de Espera

Repare que o Tempo de Execução no Processador e o Tempo de I/O de cada processo é independente do mecanismo de escalonamento. Além disso, o tempo de turnaround de cada processo pode também ser definido como o tempo decorrido entre o instante de criação do processo e o instante do seu término.

Para cada um dos mecanismos de escalonamento a seguir, desenhe o diagrama de Gantt ilustrando o escalonamento dos processos, além de calcular seus respectivos tempos de turnaround e tempo médio de espera, segundo as políticas especificadas a seguir.

Vamos calcular, para cada processo, o "Tempo de Execução no processado" e o "Tempo de I/O", pois estes são independentes do mecanismo de escalonamento. O Tempo de Execução no processado" de um processo é a soma dos tempos de rajada de CPU do mesmo. Já o "Tempo de I/O" é soma dos tempos de raja de I/O do mesmo. Temos, portanto:

Processo	Tempo de Execução no processador	Tempo de I/O
P1	7+10 = 10	12
P2	6+5 = 11	5
P3	10	0
P4	7+9 = 16	100
P5	8	0

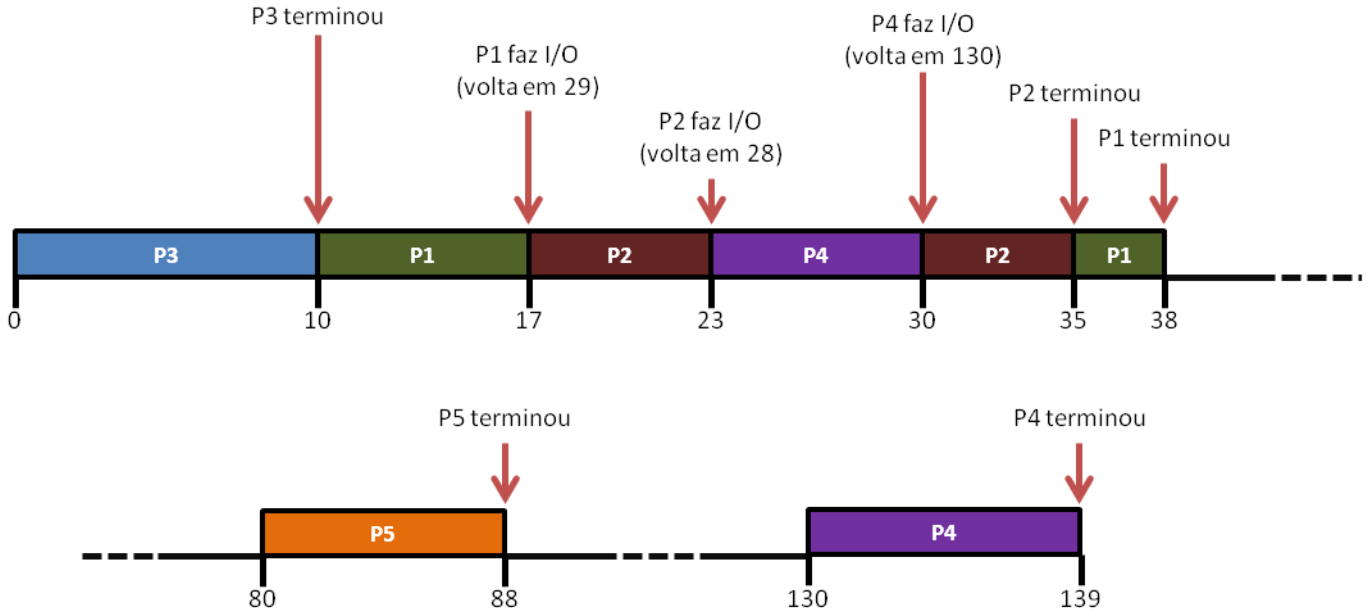
Além disso, sabemos que:

Tempo de turnaround = Instante do término - Instante de criação

e que, portanto:

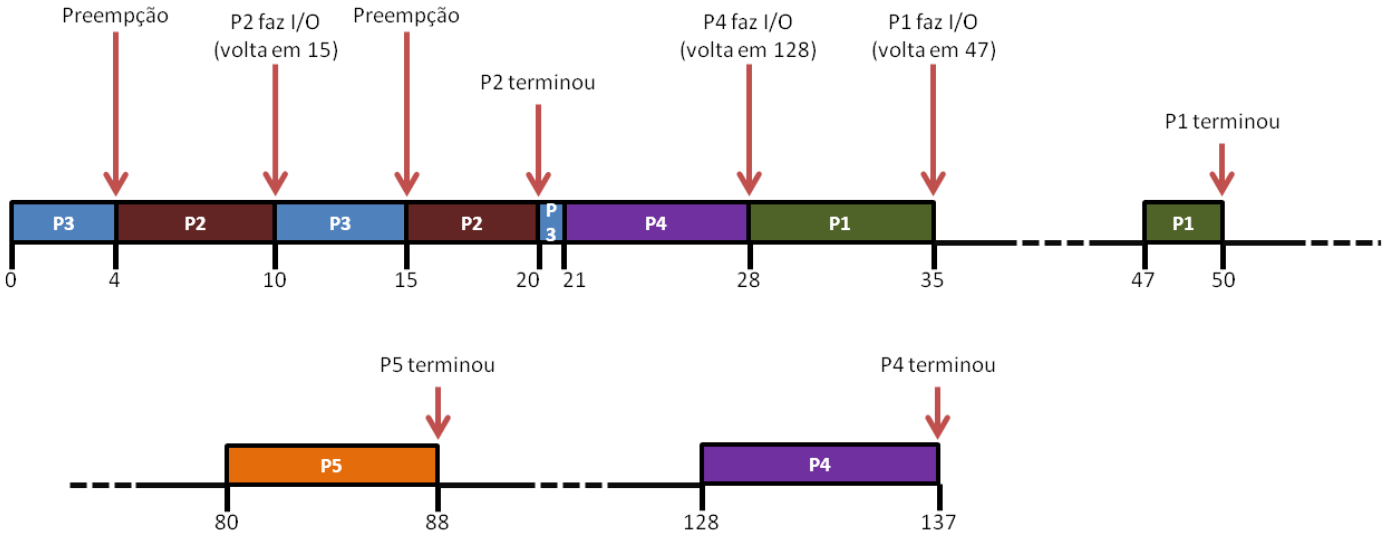
Tempo de Espera = Tempo de turnaround - Tempo de Execução no processador - Tempo de I/O

a) (2,0) FIFO



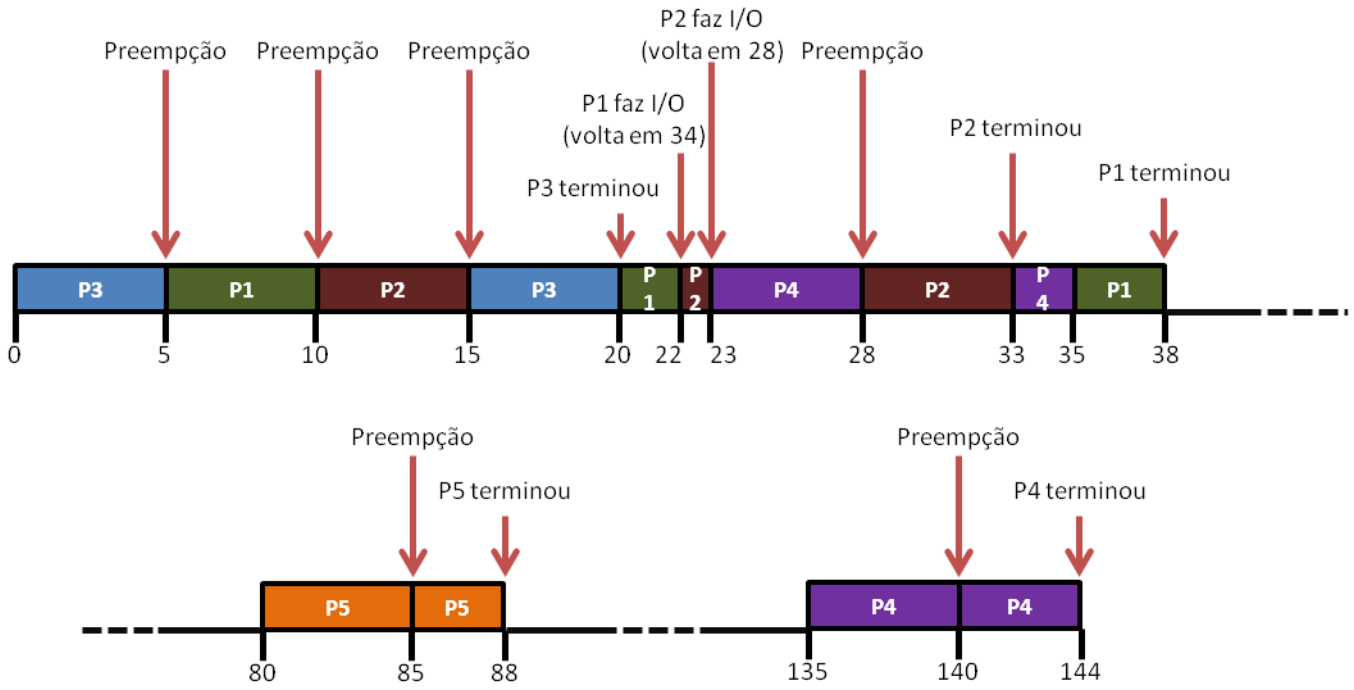
Processo	Tempo de Turnaround	Tempo de Espera
P1	38-3 = 35	35-12-10 = 13
P2	35-4 = 31	31-5-11 = 15
P3	10-0 = 10	10-0-10 = 0
P4	139-20 = 119	119-100-16 = 3
P5	88-80 = 8	8-0-8 = 0
Média	-	(13+15+0+3+0)/5 = 31/5 = 6,2

b) (2,0) Prioridade com preempção (número menor implica prioridade maior)



Processo	Tempo de Turnaround	Tempo de Espera
P1	50-3 = 47	47-12-10 = 25
P2	20-4 = 16	16-5-11 = 0
P3	21-0 = 21	21-0-10 =11
P4	137-20 = 117	117-100-16 = 1
P5	88-80 = 8	8-0-8 = 0
Média	-	(25+0+11+1+0)/5 = 37/5 = 7,4

c) (2,0) Round Robin com fatia de tempo igual a 5 u.t.



Processo	Tempo de Turnaround	Tempo de Espera
P1	38-3 = 35	35-12-10 = 13
P2	33-4 = 29	29-5-11 = 13
P3	20-0 = 20	20-0-10 =10
P4	144-20 = 124	124-100-16 = 8
P5	88-80 = 8	8-0-8 = 0
Média	-	(13+13+10+8+0)/5 = 44/5 = 8,8

2) (6,0) Considere dois processos A e B que compartilhem uma mesma fila (que comporta um máximo de 5 elementos), sendo que A coloca elementos nela e B remove elementos dela. A seguir mostramos códigos para esses processos usando o semáforo binário acesso e os semáforos de contagem vazias e cheias.

```
void ProcessoA()
{
   while (1)
   {
      P(acesso);
      P(vazias);
      InsereElementoFila();
      V(cheias);
      V(acesso);
   }
}
```

```
void ProcessoB()
{
   while (1)
   {
      P(vazias);
      P(acesso);
      RemoveElementoFila();
      V(acesso);
      V(vazias);
   }
}
```

Responda:

a) (1,0) Pode haver deadlock no código em questão? Justifique.

Sim, pois as 4 condições necessárias ocorrem:
- Exclusão mútua
- Condição de posse e espera (mais de um semáforo por processo)
- Inexistência de preempção de recursos
- Espera circular - suponha a seguinte sequência de passos:
- A fez P(acesso) e consegui fazer o decremento do semáforo
- B fez P(vazias) e conseguiu fazer o decremento do semáforo
- B faz P(acesso) e bloqueia
- A faz P(vazias) e bloqueia
Neste caso temos o ciclo A-B-A que não será desfeito.

Isso ocorre devido a um erro no uso dos semáforos explicado na questão b.

b) (1,5) O uso dos semáforos está correto? Em caso negativo, mostre o código corrigido.

Não. Segue o código correto:

```
void ProcessoA()
{
   while (1)
   {
      P(vazias);
      P(acesso);
      InsereElementoFila();
      V(acesso):
      V(cheias);
   }
}
```

```
void ProcessoB()
{
   while (1)
   {
      P(cheias);
      P(acesso);
      RemoveElementoFila();
      V(acesso);
      V(vazias);
   }
}
```

c) (1,0) Com quais valores os semáforos devem ser inicializados?

```
acesso=1;
cheias=0;
vazias=5;
```

d) (1,5) Quais são as quatro condições necessárias para a ocorrência de um deadlock? Explique cada uma delas.

- **Exclusão mútua – Pelo menos um recurso deve estar alocado em modo não-compartilhável (um processo por vez usa o recurso.**
- **Posse e espera – Um processo deve estar de posse de pelo menos um recurso e esperando para adquirir recursos adicionais que, no momento, estejam sendo mantido por outros processos.**
- **Inexistência de preempção de recursos – Recursos não podem ser interceptados, isto é, um recurso só pode ser liberado voluntariamente pelo processo que o detém.**
- **Espera circular – Deve existir um conjunto {P0, P1, ..., Pn} de processos em espera de tal modo que P0 esteja esperando por um recurso alocado a P1, P1 esteja esperando por um recurso alocado a P2, ..., Pn-1 esteja esperando por um recurso alocado a Pn e Pn esteja esperando por um recurso alocado a P0.**

e) (1,0) Quais são as três condições que devem ser garantidas em uma solução para o problema da região crítica? Explique cada uma delas.

- **Exclusão mútua: não mais do que um único processo pode executar na região crítica em qualquer dado momento**
- **Progresso: Se nenhum processo está a executar na sua secção crítica e existem processos que pretendem entrar na sua secção crítica, então apenas estes podem participar na decisão do processo que irá entrar na secção crítica e esta decisão não pode ser adiada indefinidamente.**
- **Espera limitada: Deve existir um limite de espera para o número de vezes em que é permitido a entrada a outros processos na sua secção crítica depois de um processo ter solicitado entrar na secção crítica e antes de o pedido ser garantido**.