

Nome:

Instruções: Esta prova é composta de quatro questões totalizando 10 (dez) pontos. Responda as questões de forma sucinta e clara.
BOA PROVA!

1) (3,0) Considere que 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.

Sabendo que o SO que executa os processos implementa um escalonador com múltiplas filas (sem retroalimentação), adotando FIFO para processos com prioridade ≤ 2 e SJF para os demais processo, desenhe o diagrama de Gantt ilustrando o escalonamento dos processos, além de calcular seus respectivos tempos de turnaround e tempo médio de espera.

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. Além disso, vamos identificar em qual fila cada processo será alocado, de acordo com sua prioridade. Temos, portanto:

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

Além disso, sabemos que:

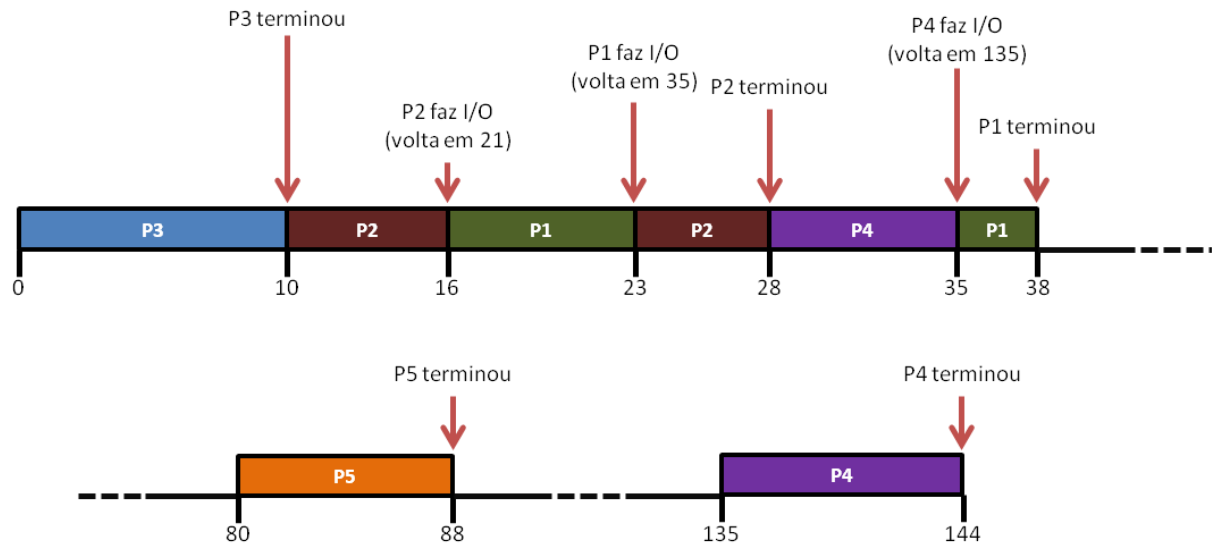
Tempo de turnaround = Instante do término - Instante de criação

e que, portanto:

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

Vamos considerar quem não há preempção entre filas, embora sejam aceitas as duas respostas, visto que essa informação não foi especificada no enunciado.

Temos, portanto, o seguinte diagrama de escalonamento:



Processo	Tempo de Turnaround	Tempo de Espera
P1	$38 - 3 = 35$	$35 - 12 - 10 = 13$
P2	$28 - 4 = 24$	$24 - 5 - 11 = 8$
P3	$10 - 0 = 10$	$10 - 0 - 10 = 0$
P4	$144 - 20 = 124$	$124 - 100 - 16 = 8$
P5	$88 - 80 = 8$	$8 - 0 - 8 = 0$
Média	-	$(13 + 8 + 0 + 8 + 0) / 5 = 29 / 5 = 5,8$

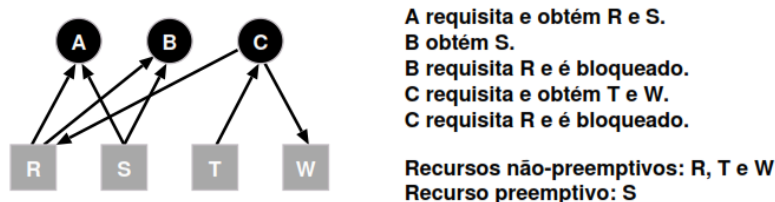
2) (2,0) O que é e como funciona DMA? Qual a relação deste conceito com a multiprogramação?

A técnica de DMA (direct memory access) é usada para que os dispositivos possam acessar diretamente a memória, evitando que o processador seja o responsável pela transferência de dados entre a memória e cada dispositivo.

Quando um processo deseja fazer uma operação de E/S ele solicita este acesso ao SO através de uma chamada de sistema. O controle da execução é desviado para o SO, que salva o contexto do processo e prossegue com o atendimento da solicitação. É então feito um pedido de acesso para a controladora do dispositivo. O processo solicitante é bloqueado e o SO prossegue com a sua execução, possivelmente resultando no escalonamento de outro processo. Quando o dado fica pronto na controladora é gerada uma interrupção e o SO é chamado para tomar providência. Neste momento é que deve ser feita a cópia dos dados da memória do dispositivo para a região de memória do processo solicitante. Sem o DMA é o SO que realiza esta tarefa. Já com o DMA o SO faz uma requisição para que a controladora de DMA realize esta cópia e é liberado para escalonar outros processos. Quando a cópia é finalizada pela controladora de DMA, é gerada uma interrupção e o SO entra em ação para desbloquear o processo solicitante e movê-lo para a fila de prontos.

O DMA maximiza as oportunidades de concorrência geradas pela multiprogramação, visto que libera o processador da tarefa de copiar os dados entre dispositivo e memória. Vale lembrar que existe um custo de configuração do DMA que só é compensado quando o mesmo é usado para copiar grandes blocos de dados. Além disso, o DMA requer acesso exclusivo ao barramento de memória, impedindo que o processador tenha acesso a outros dados na memória.

3) (2,0) Um aluno de sistemas operacionais fez a figura dada a seguir para um trabalho do curso. Na figura, é mostrado o grafo de recursos para os passos ao lado do grafo e a conclusão do aluno em relação aos possíveis impasses (deadlocks) no grafo. Se você acha que a figura do trabalho está correta basta responder que sim mas, se você acha que a figura está errada, aponte os erros que existem nela.



Conclusão: um impasse ocorre devido ao ciclo A-R-B-S-A

A figura não está correta, pois existem quatro erros nela. O primeiro erro ocorreu quando B obteve S. Como S é preemptivo, este recurso deve ser retirado de A. Sendo assim a aresta $S \rightarrow A$ não deve existir no desenho. O segundo erro ocorreu na requisição de R por B, pois a aresta deveria estar orientada de B para R, já que B foi bloqueado devido a R ser

não preemptivo e estar alocado a A. O terceiro erro ocorreu na requisição de W por C pois, como C conseguiu obter W com sucesso, a aresta deveria estar orientada de W para C. Finalmente, o último erro diz respeito à conclusão do aluno sobre o impasse: mesmo depois de corrigir a orientação incorreta da aresta descrita no primeiro erro, não existe um ciclo orientado envolvendo os recursos R e S e os processos A e B.

4) (3,0) Considere o pseudo-código abaixo para o problema do leitor escrito e responda:

```
void Escritor() {  
    ProduzDado();  
    P(Acesso);  
    Escreve();  
    V(Acesso);  
}  
  
void Leitor() {  
    P(Exclusao);  
    Nleitores++;  
    if ( Nleitores == 1 ) P(Acesso);  
    V(Exclusao);  
    Leitura();  
    P (Exclusao);  
    Nleitores--;  
    if ( Nleitores == 0 ) V(Acesso);  
    V(Exclusao);  
    ProcessaDado();  
}
```

- a) (1,0) Qual deve ser o valor inicial dos semáforos Acesso e Exclusao e da variável Nleitores para que o programa funcione corretamente?

Acesso=1, Exclusao=1 e Nleitores=0

- b) (1,0) Se tivermos 10 leitores executando a função leitura e 1 escritor é executado, quais serão os valores dos semáforos Acesso e Exclusao e da variável Nleitores? Sobre qual(is) semáforo(s) se dá o bloqueio do escritor?

Acesso=0, Exclusao=1, Nleitores=10. O bloqueio se dá sobre o semáforo acesso.

- c) (1,0) Descreva uma situação onde os escritores sofram starvation.

O starvation ocorre quando novos leitores chegam sempre antes do último leitor terminar, de forma que o semáforo acesso nunca seria liberado para gerar a oportunidade para execução de um escritor.