

FIAP

Aluno: Gabriel Kazuki Onishi

RM: 87182

Checkpoint 1 – Operating System Tuning And Cognation – Professor Sérgio Ricardo Rota

1.

A) Verdadeira, pois independentemente da quantidade de núcleos de processamento, para o programa funcionar é necessário que o processo seja executado pelo processador, e isso aconteceria tendo um ou mais núcleos.

B) Verdadeira, pois cada sistema operacional pode ter o seu conjunto de chamadas de sistema, de forma que se não forem compatíveis o programa não rodaria, já que as chamadas não existiriam naquele SO. Portanto, como é dito no enunciado que o programa rodou no sistema A e B, podemos inferir que as chamadas são compatíveis nos dois sistemas operacionais.

C) Falsa, pois ao compilar o programa, o que se obtém é justamente um conjunto de instruções a partir do sistema A para que o processo possa ser executado, e tendo em vista que o mesmo código compilado foi passado para o sistema B, não teria como o conjunto de instruções ser diferente.

D) Falsa, pois os registradores de um sistema 64 bits são diferentes de um 32 bit, de forma que para que o programa fosse executado com sucesso, os registradores teriam que ser compatíveis entre si. Ou seja, dado que o programa executou com sucesso nos dois sistemas, podemos dizer que os registradores eram compatíveis, portanto, ambos os sistemas tinham a mesma arquitetura.

E) Falsa, pois pode ser que os sistemas tenham utilizado um padrão nas suas chamadas de sistema, como o POSIX ou SUS, de forma que mesmo sendo sistemas operacionais diferentes, eles teriam semelhanças que permitiriam compatibilidade nas chamadas de sistemas e, portanto, o programa seria capaz de rodar em ambos.
2.

A) Não, pois a quantidade de memória RAM não interfere no tempo de troca de contexto, apenas aumentando a quantidade de processos que poderiam ficar no estado “Pronto”.

B) Sim, pois dado que a memória cache é mais rápida que a RAM, o processo de salvar o contexto de hardware e ler outro nos registradores seria mais veloz, reduzindo assim o tempo de sobrecarga.

C) Sim, pois isso permitiria ao SO conseguir trocar o contexto de hardware mais rapidamente, já que não seria necessário executar a transição de salvar o contexto atual e de trazer o de outro processo e vice-versa a todo momento, podendo economizar tempo com isso.
3. Analisando a tabela I e a informação de uso da CPU/IO nos processos CPU-bound e I/O-bound, temos que cada processo apresenta um uso CPU/IO de aproximadamente:
 - App1: 20 ms de CPU
 - App2: 40 ms de CPU
 - App3: 60 ms de CPU
 - App4: 15 ms de CPU | 35 ms de IO

- App5: 30 ms de CPU | 70 ms de IO
- App6: 45 ms de CPU | 105 ms de IO

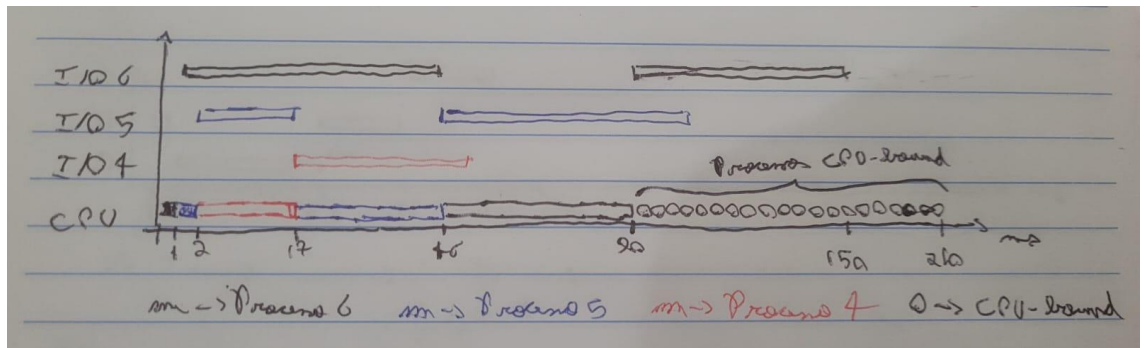
E considerando que:

- A melhor maneira de otimizar a execução de um grupo de processos seja utilizar o tempo em que um processo esteja bloqueado para executar o processamento de outro, de forma que durante a execução do maior processo de I/O os outros processos são capazes de serem executados, ou seja, ao executar o processo com maior I/O primeiro, temos a possibilidade de otimizar a execução do grupo rodando os demais processos durante o bloqueio desse.
- O tempo calculado a partir da tabela I esteja correto e que ocorra em todas as execuções simuladas.
- A ordem de execução do grupo não siga a ordem definida na tabela II.
- As execuções de I/O não são bloqueadas entre si.
- A pior possibilidade em relação ao tempo de execução é a situação em que se utiliza muito pouco ou quase nada da CPU enquanto os processos estão bloqueados, de forma que todos os processos precisam que os seus I/O rodem após quase todo os processamentos serem executados, onde o processo com maior uso de I/O é executado por último, prolongando a execução do grupo.

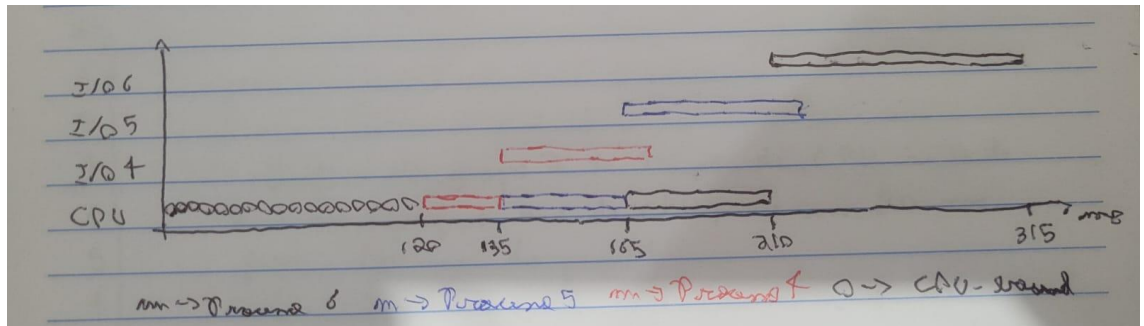
Temos que:

A) Verdadeira, já que ao simular cenários otimizados e mal otimizados, temos os seguintes gráficos de execução:

Cenário otimizado:



Cenário mal otimizado:

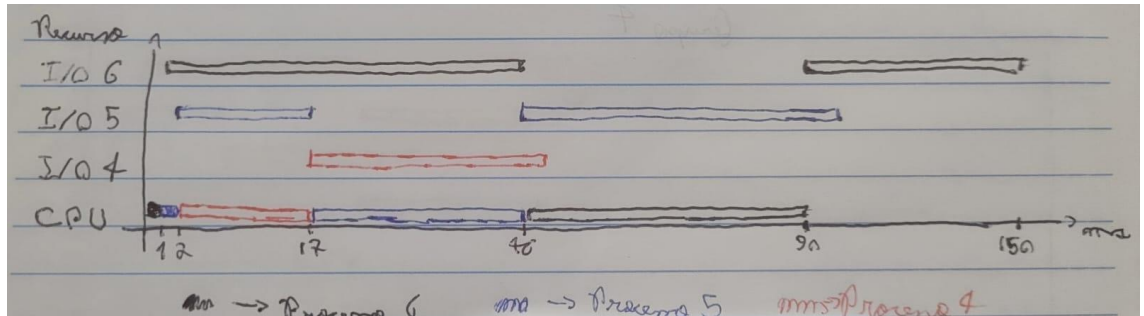


De forma que temos um tempo total mínimo de 210 ms e máximo de 315 ms, possibilitando os 306 ms da alternativa.

B) Falsa, pois tendo em vista que todos os processos do grupo são CPU-bound e que temos apenas um processador, todos os processos teriam que ser executados um após

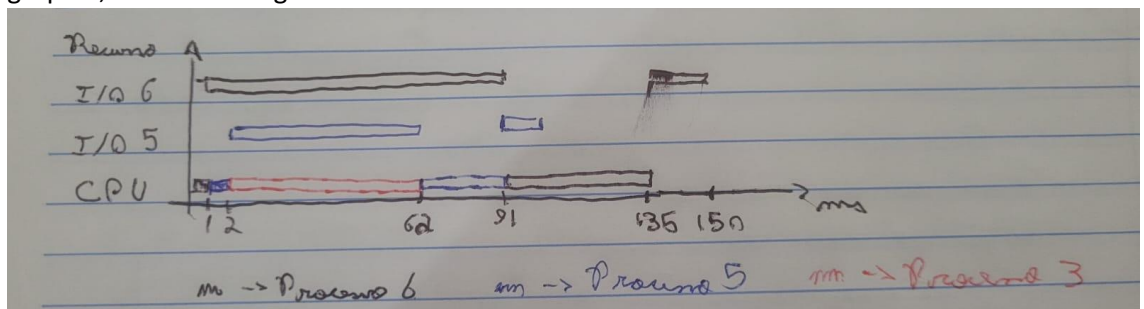
o outro, sem a possibilidade de economizar o tempo de execução, ou seja, o tempo mínimo para a sua execução seria $20 + 40 + 60$, resultando em 120 ms.

C) Falsa, pois considerando que todos os processos apresentam a execução de I/O, existe a possibilidade de que os processos tenham sido executados de uma forma eficiente, utilizando o tempo de I/O de um processo para otimizar a execução de CPU de outros, de forma que podemos obter o seguinte gráfico imaginando um cenário otimizado executando os processos 6, 5 e 4, nessa ordem:



Portanto, temos que o tempo mínimo de execução do grupo 2 seria 150 ms.

D) Falsa, pois ao desenhar o gráfico de execução em um cenário otimizado para o grupo 3, obtemos o seguinte:



De forma que o tempo mínimo de execução do grupo 3 seria 150 ms, invalidando a informação de que o tempo total deveria estar acima de 175 ms.