Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
Professores: Valmir C. Barbosa e Felipe M. G. França
Assistente: Alexandre H. L. Porto

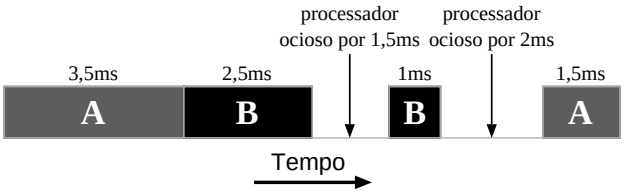
Quarto Período
Gabarito da AD1 - Segundo Semestre de 2018

Atenção: Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um, 1/3 dos pontos daquela questão.

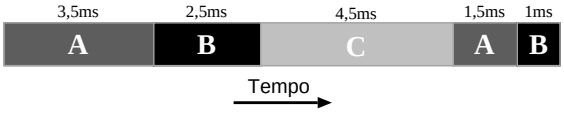
Nome -
Assinatura -

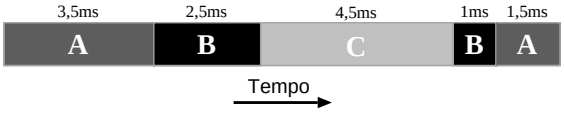
---

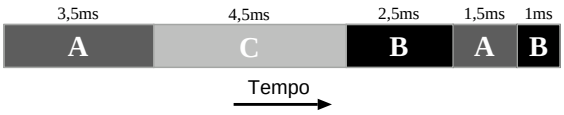
1. (1,5) Suponha que dois programas, A e B, que fazem somente uma operação de E/S cada um, tenham sido executados como na figura a seguir, em um sistema operacional em que a multiprogramação é usada somente para evitar a ociosidade do processador quando operações de E/S são feitas. Se um programa C, que não faz operações de E/S e que precisa executar por 4,5ms, passar a executar junto com A e B, que ordens de execução de A, B e C levarão à menor ociosidade possível do processador? Justifique a sua resposta.

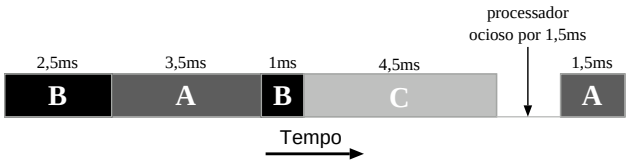


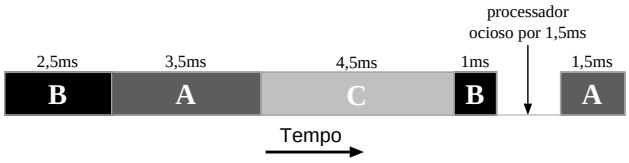
Resp.: Pela figura do enunciado, vemos que as durações das operações de E/S de A e B são, respectivamente, 7ms e 1,5ms. Nas figuras a seguir mostramos todas as soluções que respeitam esses dois tempos. Pelas figuras, vemos que somente as ordens de execução ABCAB, ABCBA e ACBAB geram a menor ociosidade possível (0ms).

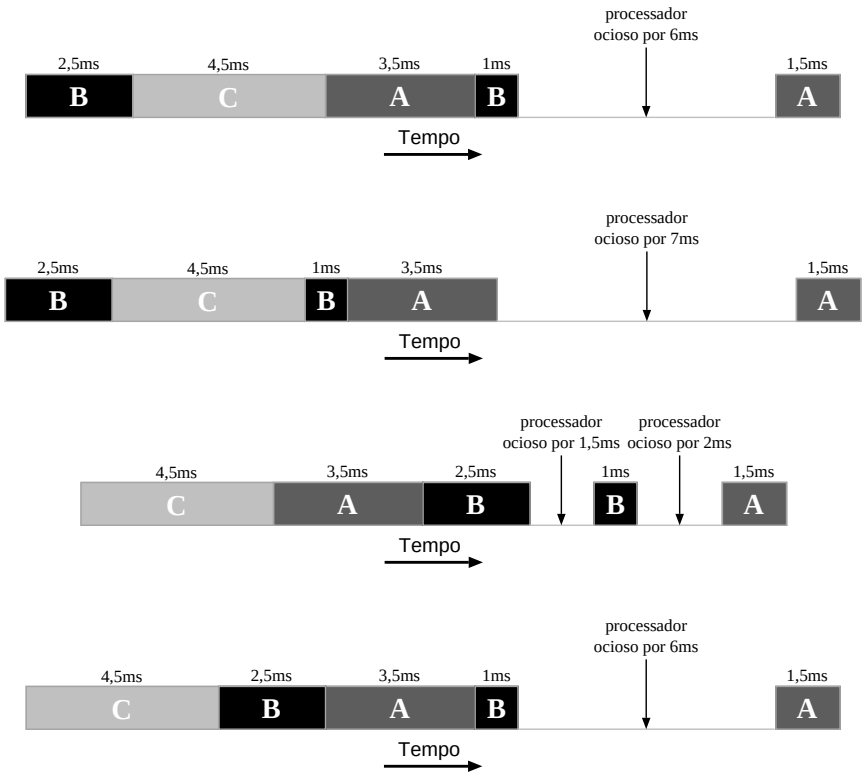




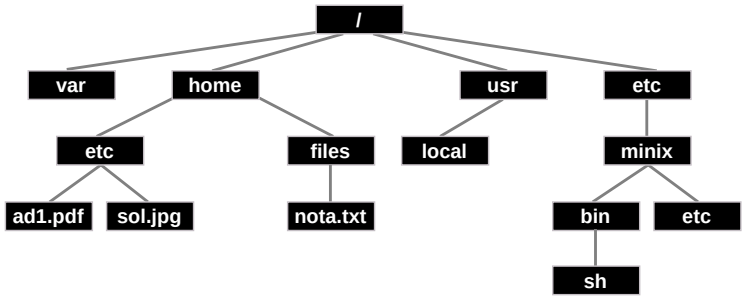








2. (1,0) Considere a hierarquia de diretórios dada a seguir, e suponha que você está no diretório /etc/minix/etc. Com relação a este diretório, existirão arquivos no diretório cujo caminho relativo é ../bin/../../../var /../usr/local/../../home/etc/../files, supondo que ".." indica o diretório pai do diretório atual durante o percurso? Qual teria sido o caminho relativo mais sucinto ao mesmo diretório final a partir do mesmo diretório inicial? E qual é o caminho absoluto para os arquivos do mesmo diretório final, se existirem?



Resp.: -Na tabela dada a seguir mostramos, passo a passo, cada um dos diretórios alcançados ao seguirmos o nome do diretório dado no enunciado da questão. Como podemos ver pela última linha da tabela, existe o arquivo nota.txt.

Nome do diretório (parcial)	Diretório atual
..	/etc/minix
../bin	/etc/minix/bin
../bin/..	/etc/minix
../bin/../..	/etc
../bin/../../..	/
../bin/../../../var	/var
../bin/../../../var/..	/
../bin/../../../var/../usr/	/usr
../bin/../../../var/../usr/local	/usr/local
../bin/../../../var/../usr/local/..	/usr
../bin/../../../var/../usr/local/../..	/
../bin/../../../var/../usr/local/../../home	/home
../bin/../../../var/../usr/local/../../home/etc	/home/etc
../bin/../../../var/../usr/local/../../home/etc/..	/home
../bin/../../../var/../usr/local/../../home/etc/../files	/home/files

-Para obter o caminho relativo mais sucinto ao diretório com o arquivo nota.txt, basta usar "..", a partir do diretório /etc/minix/etc, até atingir o primeiro diretório do qual o arquivo nota.txt seja descendente, e depois adicionar o caminho desse diretório até o arquivo. Como esse diretório é /, então o caminho relativo mais sucinto é ../../../home/files.

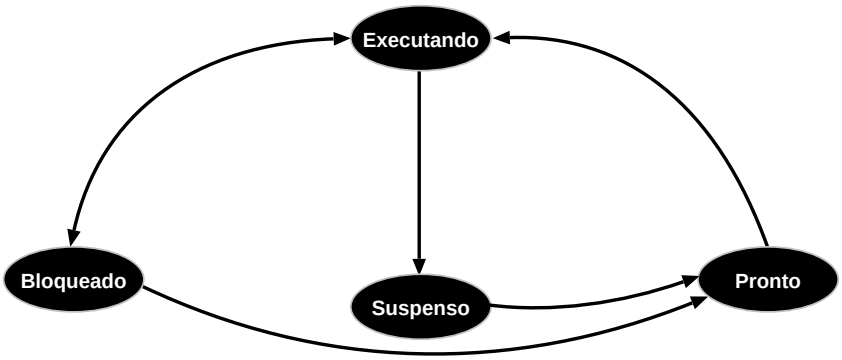
-O nome absoluto do diretório com o arquivo nota.txt é aquele a partir do diretório raiz, /, ou seja, /home/files.

3. (2,0) Suponha que o sistema operacional esteja executando diretamente sobre o hardware de um computador cujas operações de E/S demorem 1,4ms. Suponha ainda que um processo tenha executado por 15 000ms e que, durante a sua execução, tenha feito x operações de E/S. Se o sistema operacional agora executar sobre uma máquina virtual que reduza a velocidade do processador em 50% e a velocidade das operações de E/S em 30%, e se além disso forem executadas somente $x - 2\,500$

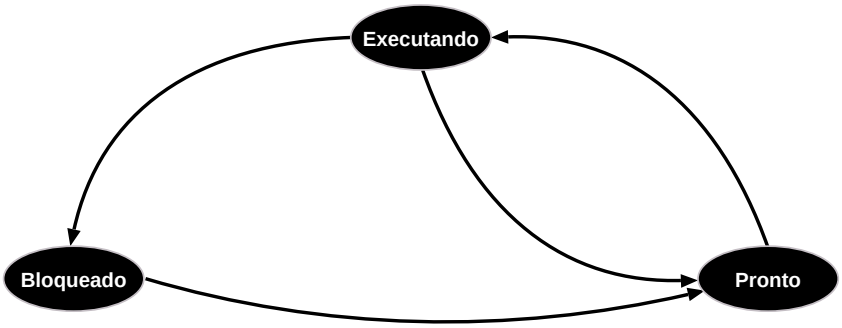
operações de E/S, para que valor de x o processo executará na máquina virtual por 18 000ms? Justifique a sua resposta.

Resp.: Como o tempo total de execução é de 15 000ms, e como o processo faz x operações de E/S com duração de $1,4$ms cada, então $1,4x$ ms dos 15 000ms são gastos com operações de E/S quando a execução ocorre sobre o hardware do computador. Logo, o processo executa no processador desse hardware por $(15\,000 - 1,4x)$ ms. Note que a velocidade do processador ser reduzida em 50% significa que a velocidade do processador virtual é 50% da velocidade do processador real, o que por sua vez significa que, durante os $(15\,000 - 1,4x)$ ms, somente 50% das instruções podem ser executadas. O tempo necessário para executar 100% das instruções sobre a máquina virtual é de $\frac{15\,000-1,4x}{0,5} = (30\,000 - 2,8x)$ ms. Agora, como o processo faz $x - 2\,500$ operações de E/S na máquina virtual, e como o novo tempo de cada operação de E/S é de $\frac{1,4}{0,7} = 2$ms (já que, similarmente à redução do tempo do processador, a redução da velocidade de cada operação de E/S em 30% significa que no mesmo tempo podemos, na máquina virtual, executar somente 70% das operações de E/S originais), então $(x-2\,500)\times 2 = (2x-5\,000)$ ms dos 18 000ms do tempo de execução do processo na máquina virtual são gastos com E/S. Logo, o tempo de execução do processo no processador virtual é de $18\,000 - (2x - 5\,000) = (23\,000 - 2x)$ ms. Usando $30\,000 - 2,8x = 23\,000 - 2x$, temos $x = 8\,750$.

4. (1,5) Suponha que um aluno tenha fornecido o diagrama de transição dado na figura a seguir, com os possíveis estados de um processo estudados na aula 4 e as possíveis transições entre esses estados. Existe algum estado adicional ou algum erro nas transições fornecidas? Justifique a sua resposta.



Resp.: A figura do aluno está errada, porque existe um estado adicional, uma aresta errada, duas arestas inexistentes associadas ao estado adicional, e uma aresta faltando. Como podemos ver no diagrama dado na aula 4, não existe o estado **Suspenso**, logo também não existem as arestas entre o estado **Executando** e o estado **Suspenso** ou entre o estado **Suspenso** e o estado **Pronto**. Além disso, a aresta entre os estados **Executando** e **Bloqueado** está errada porque não deveria ter duplo sentido, sendo somente direcionada do estado **Executando** para o estado **Bloqueado**, pois a transição somente ocorre quando o processo em execução é bloqueado devido a precisar esperar por um evento externo. Finalmente, a aresta faltando é a do estado **Executando** para o estado **Pronto**, indicando que o processo em execução foi suspenso pelo escalonador devido a este último ter decidido que um outro processo deveria executar no processador. A seguir mostramos o diagrama de transições correto:



5. (2,0) Suponha que um arquivo, que pode armazenar até n números, seja compartilhado pelos processos A, B e C. O processo A continuamente insere quatro números no arquivo, desde que este possa armazenar pelo menos quatro números adicionais. Já o processo B continuamente remove do arquivo dois números, a e b, desde que o arquivo possua pelo menos dois números, e depois insere de volta o resultado de a^b. Finalmente, o processo C continuamente imprime todos os números armazenados no arquivo se ele não estiver vazio, ou uma mensagem de alerta, caso contrário. Como três semáforos, um binário e dois de contagem, podem ser usados para garantir a correta execução de A, B e C, sem a ocorrência de condições de corrida? Justifique a sua resposta.

   Resp.: Como não foi definido o total inicial de números no arquivo, vamos supor que ele tem inicialmente x números, $0 \leq x \leq n$. O semáforo binário *acesso* é usado para garantir o acesso exclusivo ao arquivo, e é inicializado com 1 porque inicialmente nenhum processo está acessando o arquivo. O semáforo *ocupados* conta o total de números no arquivo, é inicializado com x, e é usado para bloquear B quando o arquivo não tem pelo menos 2 números. Finalmente, o semáforo *livres* conta o total de números que ainda podem ser inseridos no arquivo, sendo inicializado portanto com $n - x$, e sendo usado para bloquear A quando o arquivo não tem espaço para pelo menos 4 números adicionais. A seguir mostramos os pseudocódigos para os processos A, B e C, usando os semáforos descritos.

```
void ProcessoA(void)
{
  while(1)
  {
    // Usa a operação P sobre livres para garantir que pelo menos quatro
    // números podem ser inseridos no arquivo.
    P(livres);
    P(livres);
    P(livres);
    P(livres);
    // Garante o acesso exclusivo ao arquivo.
    P(acesso);
    // Código para gerar os quatro números e depois inseri-los no arquivo.
    // Libera o acesso exclusivo ao arquivo.
    V(acesso);
    // Usa a operação V sobre ocupados para registrar que quatro números foram
    // inseridos no arquivo.
    V(ocupados);
    V(ocupados);
    V(ocupados);
    V(ocupados);
  }
}
```

```
void ProcessoB(void)
{
  while(1)
  {
    // Usa a operação P sobre ocupados para garantir que pelo menos dois
    // números existem no arquivo.
    P(ocupados);
    P(ocupados);
    // Garante o acesso exclusivo ao arquivo.
    P(acesso);
    // Código para remover dois números do arquivo e armazená-los em a e b.
    // Código para inserir o número a^b no arquivo.
    // Libera o acesso exclusivo ao arquivo.
    V(acesso);
    // Usa a operação V sobre livres para registrar que um número foi
    // removido do arquivo (pois inserimos um número após remover dois
    // números do arquivo).
    V(livres);
    // Usa a operação V sobre ocupados para registrar que a^b foi inserido no
    // arquivo.
    V(ocupados);
  }
}

void ProcessoC(void)
{
  while(1)
  {
    // Garante o acesso exclusivo ao arquivo.
    P(acesso);
    // Código para imprimir a mensagem de alerta, se o arquivo estiver vazio, ou
    // todos os números do arquivo, em caso contrário.
    // Libera o acesso exclusivo ao arquivo.
    V(acesso);
  }
}
```

6. (2,0) Suponha que os processos A, B e C, que não fazem operações de E/S, tenham executado como na tabela a seguir, quando o escalonador por *round robin* foi usado pelo sistema operacional. Suponha agora que um escalonador por prioridades passe a ser usado pelo sistema operacional, sendo que a prioridade do processo em execução é reduzida em 3 unidades a cada 2ms, e que cada processo executa até existir um processo com prioridade maior. Como será a variação do tempo decorrido

entre o início e o término de cada processo, em relação aos mesmos tempos derivados da tabela, se a prioridade inicial de cada processo for igual a, respectivamente, 3, 11 e 7? Justifique a sua resposta.

0	2	4	6	8	10	12	14	15	17	19	21	23	25	26
B	A	C	B	A	C	B	A	C	B	C	B	C	C	-

Resp.: Primeiramente, pela tabela, vemos que os tempos de execução de A, B e C são de, respectivamente, 5ms, 10ms e 11ms, e que os tempos decorridos entre o início e o término de A, B e C são de, respectivamente, 13ms, 23ms e 22ms. Usando os tempos de execução obtidos da tabela do enunciado e o modo de executar o algoritmo de prioridades descrito, vemos que a ordem de execução dos processos é como dada na tabela a seguir. Na tabela, mostramos como os processos são escolhidos pelo algoritmo, sendo que cada coluna refere-se à execução de um processo dando o tempo de início, o processo correspondente, e a prioridade cujo valor definiu que o processo deveria executar no processador. Pela tabela, vemos que os tempos decorridos entre o início e o término de A, B e C agora são de, respectivamente, 13ms, 20ms e 22ms, implicando que os tempos de A e de C não mudaram e que o tempo de B foi reduzido em 3ms.

0	2	4	6	8	10	12	14	16	18	20	22	23	25	26
B	B	C	B	C	A	B	C	A	B	C	A	C	C	-
11	8	7	5	4	3	2	1	0	-1	-2	-3	-5	-8	-