UNIVERSIDADE DE SÃO PAULO - ICMC
DEPARTAMENTO DE SISTEMAS DE COMPUTAÇÃO
SSC-0541 – SISTEMAS OPERACIONAIS I
Turma: Bacharelado em Sistemas de Informação
Profa: Kalinka Regina Lucas Jaquie Castelo Branco

Primeira Lista de Exercícios Para Passar o Tempo
GABARITO

1. **(Petrobras 2005)** Uma *system call* pode ser entendida como uma porta de entrada para o acesso ao núcleo do sistema operacional e a seus serviços. Isto significa que:

(A) uma aplicação desenvolvida utilizando serviços de um determinado sistema operacional pode ser portada diretamente para outro sistema operacional.

(B) a maioria dos programadores conhece os detalhes da implementação de uma system call, podendo implementá-lo no seu próprio programa.

(C) para cada serviço disponível existe uma system call associada e cada sistema operacional tem o seu próprio conjunto de chamadas, com nomes, parâmetros e formas de ativação específicos.

(D) o modo de acesso de uma system call é o mesmo de um programa de usuário.

(E) os utilitários de um SO, como compiladores e editores de texto, são considerados system calls.

2. **(Petrobras 2005)** Um device driver, ou somente *driver*, tem como função implementar a comunicação do subsistema de E/S com os dispositivos, através de controladores. Sendo assim, um device driver:

(A) manipula diretamente os dispositivos de E/S.

(B) permite a comunicação entre o sistema computacional e o mundo externo.

(C) recebe comandos gerais sobre acessos aos dispositivos, traduzindo-os para comandos específicos, que poderão ser executados pelos controladores.

(D) realiza as funções comuns a todos os tipos de dispositivos.

(E) torna as operações de E/S mais simples para o usuário bem como suas aplicações.

3. **(CVM 2003)** Os dispositivos do computador, para requisitarem atenção do processador, utilizam um sinal para provocar uma operação de

(A) sobreposição.

(B) concorrência.

(C) transferência.

(D) interrupção.

(E) *deadlock*.

4. **(TCU 2002)** A técnica de spooling foi introduzida para aumentar o grau de concorrência e a eficiência dos sistemas operacionais. Como a velocidade de operação dos dispositivos de E/S é muito menor que a do processador, era comum que a CPU permanecesse ociosa à espera de programas e dados de entrada ou pelo término de uma impressão. Com relação a esta técnica são feitas as afirmativas abaixo.

(A) A técnica de spooling utiliza uma área em disco como se fosse um grande buffer.

(B) O uso do spooling vincula o programa ao dispositivo de impressão, reservando assim o dispositivo para uso exclusivo.

(C) Com o surgimento de dispositivos de acesso direto, como discos, foi possível tornar o spooling mais eficiente, possibilitando principalmente o processamento não sequencial de jobs.

Está(ão) correta(s) a(s) afirmativa(s):

(I) A, apenas.
(II) A e B, apenas.
(III) A e C, apenas.
(IV) B e C, apenas.
(V) A, B e C.

5. **(TCU 2002)** Analise as seguintes afirmações relativas a conceitos gerais de informática:

I. Uma API é um conjunto de rotinas que um programa aplicativo usa para solicitar e realizar serviços de baixo nível executados por um outro componente, como, por exemplo, o sistema operacional do computador ou um serviço que está sendo executado em rede.

II. Uma interrupção pode ser definida como uma condição assíncrona do sistema operacional que interrompe a execução normal e transfere o controle a um gerenciador de interrupção. As interrupções podem ser enviadas tanto por dispositivos de hardware como de software que necessitem de serviços do processador.

III. Uma memória expandida é um tipo de memória de até 512 Kbytes que pode ser acrescentada apenas em computadores com processador 8086 ou 8088. O gerenciamento da memória expandida é permitido apenas no sistema operacional MS-DOS para execução de programas com mais de 64 Kbytes.

IV. OLE é uma interface de programação de aplicativos que permite o acesso a dados de uma variedade de fontes de dados existentes.

Indique a opção que contenha todas as afirmações verdadeiras.

a) I e II
b) I e III
c) II e III
d) II e IV
e) III e IV

6) Em linha geral, quais são as 2 principais funcionalidades de um Sistema Operacional ?

Resposta: Gerenciar Recursos e Máquina Virtual

7) Um sistema operacional pode ser encarado sob dois aspectos: máquina virtual e gerenciador de recursos. Explique cada um deles.

Gerenciar Recursos

O sistema operacional deve gerenciar a utilização dos recursos fornecidos pelo hardware, como processadores, memória, dispositivos de E/S, de modo que mantenha o controle sobre qual usuário/programa utiliza qual recurso, compartilhando os recursos entre os usuários/programas de modo seguro e sem conflitos.

e Máquina Virtual

O sistema operacional deve oferecer ao usuário uma maneira mais acessível de programar/utilizar o hardware do que as próprias instruções que este oferece. Por exemplo: o usuário não precisa saber qual a trilha e o setor do disco se deseja gravar alguma informação, apenas faz uma chamada ao sistema que estende as instruções de E/S, disponibilizando instruções mais amigáveis para estas e outras tarefas. Ou seja, o sistema operacional atua como uma interface entre o hardware e o ambiente de software.

8) Descreva os tipos de sistemas operacionais que estudamos na segunda aula.

Resposta: olhar nos slides

9) Explique como os computadores modernos conseguem executar várias operações ao mesmo tempo sendo que eles têm apenas um processador, ou seja, como é possível ouvir música e digitar um texto ao mesmo tempo nos PCs atuais ?

Resposta: O sistema é multiprogramado, ou seja, ele consegue manter vários programas na memória da máquina e, além disso, compartilha o tempo de uso dos programas por meio do 'time-sharing', mas esse tempo é tão rápido que se torna imperceptível pelo usuário.

10) Defina:

a. Sistemas multiprogramados;

Resposta: Mantém mais de um programa simultaneamente na memória principal, para permitir o compartilhamento efetivo do tempo de CPU e demais recursos.

b. Sistemas multiprocessados;

Resposta: Executam várias tarefas simultaneamente, por ter mais de um processador.

11) Todo sistema multiprogramado é um sistema multiprocessado? Por que?

Resposta: Sim, pois realiza a mesma tarefa do multiprocessado, sendo diferente, apenas o número de processadores.

12) Todo sistema multiprocessado deve ser multiprogramado ? Por que?

Resposta: Não necessariamente, pois se for monoprogramado, usará apenas um processador, porém para se obter uma vantagem em usar o multiprocessado, ele deve ser multiprogramado.

13) Sobre processos:

a. O que é um processo?

Resposta: É um programa em execução, incluindo os valores correntes do contador de programa, registradores, e variáveis. O conceito de processo é dinâmico, em contraposição ao conceito de programa, que é estático.

b. Quais são os estados que eles podem assumir?

Resposta: Indefinido, bloqueado, pronto para execução e em execução.

c. Quais são as três partes essenciais de que um processo é composto?

Resposta: contexto de hardware, de software e espaço de endereçamento, que juntos mantêm informações necessárias a execução de um programa.

14) Qual a relação entre programa e processo?

Resposta: um processo é um programa em execução

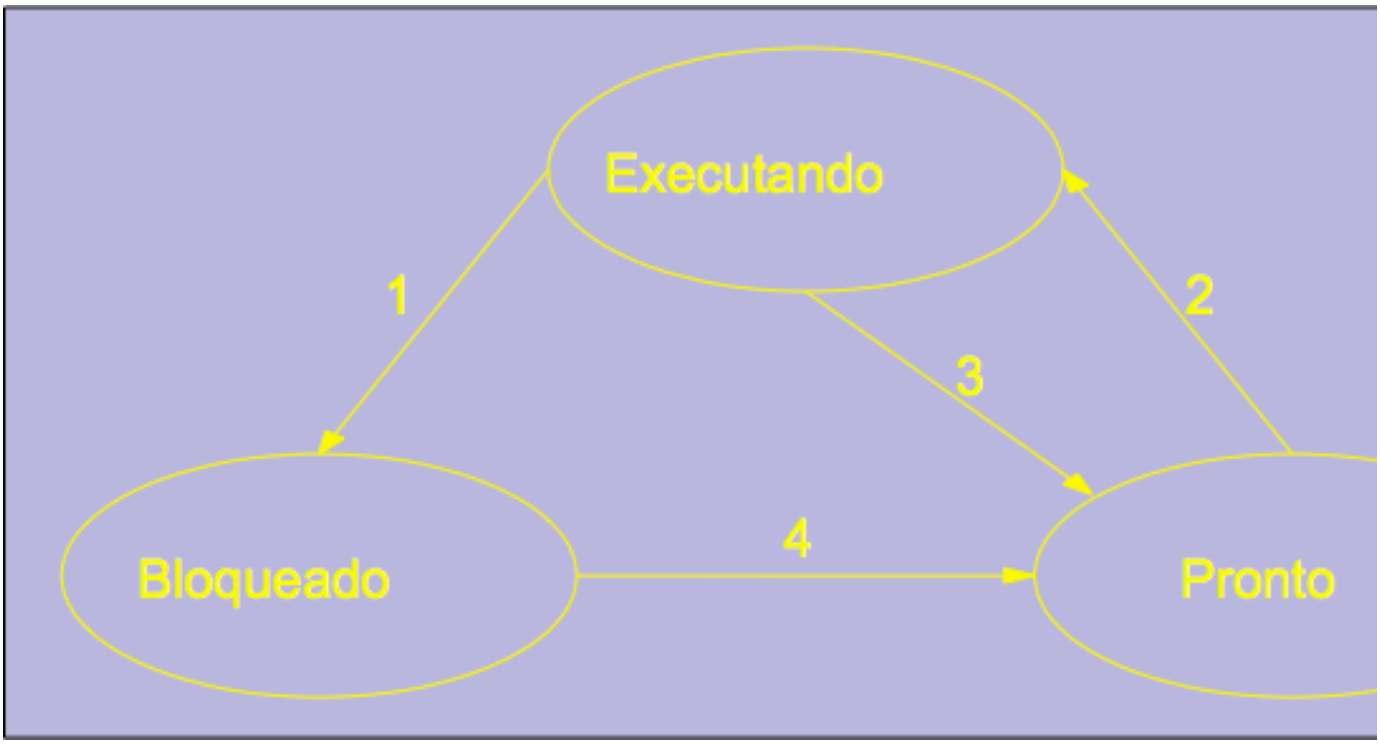
15) O que é o BCP? Qual é o seu conteúdo típico?

Resposta: É o Bloco de Controle de um Processo. Ele reside na memória principal e mantém todas as informações sobre contexto de hardware, software e espaço de endereçamento de cada processo (ponteiros, estado do processo, nome do processo, prioridade do processo, registradores, limites de memória, lista de arquivos abertos, etc).

16) Explique como funciona o algoritmo de escalonamento de processos com Prioridade e simule sua execução com no mínimo 04 filas de processos.

Resposta: olhar nos slides

17) Considerando os três possíveis estados de um processo (executando, bloqueado e pronto), desenhe um diagrama com as possíveis transições de um estado para outro e explique quando essas transições ocorrem.



Olhar os Slides de 22 a 25

18) Explique como **Monitores** tratam a Exclusão Mútua entre processos que competem por recursos compartilhados.

Resposta: É um mecanismo de sincronização de processos/threads que provê programação concorrente estruturada que concentra a responsabilidade para correção dentro de poucos módulos. Sendo assim, seções críticas tais como alocação de dispositivos de I/O e memória, requisições de enfileiramento para I/O, e assim por diante, são centralizados em um programa privilegiado. Programas ordinários requisitam serviços que são realizados pelo monitor central. A sintaxe de monitores é baseada no encapsulamento de itens de dados e as procedures que operam sobre eles em um único módulo. A interface de um monitor consiste em um conjunto de procedures. Essas procedures operam sobre dados que são ocultos dentro do módulo monitor garantindo assim a exclusão mútua.

19) O código abaixo apresenta uma solução utilizando semáforos para solucionar o problema do Barbeiro. O problema do barbeiro consiste do seguinte:

A barbearia possui **um barbeiro**, **uma cadeira de barbeiro**, e **n (CHAIRS)** cadeiras para os clientes que estão esperando para serem atendidos. Se não existem clientes para serem atendidos, o barbeiro senta na sua cadeira e dorme. Quando um cliente chega, ele acorda o barbeiro para ser atendido. Se outros clientes chegarem enquanto o barbeiro estiver ocupado, ou eles sentam em alguma cadeira para clientes que esteja livre ou vão embora se a barbearia estiver cheia.

Considerando esse problema, responda as seguintes questões:

a. Descreva como funcionam os semáforos nesta solução.
b. Qual é a região crítica?

c. O que aconteceria se os seguintes comandos fossem retirados:
void barber -> down(&mutex);
void customer -> up (&mutex);

```
# define CHAIRS 5 /*quantidade de cadeiras para clientes*/

typedef int semaphore;

semaphore customers = 0; /*clientes esperando pelo serviço*/
semaphore barbers = 0;  /* barbeiros esperando*/
semaphore mutex = 1; /*para exclusão mútua*/
int waiting = 0; /* quantidade de clientes*/

void barber (void)
{
    while (TRUE)
    {
down(&custumers):
down(&mutex);
waiting = waiting – 1;
up(&barbers);
up(&mutex);
cut_hair( );
    }
}
```

```
void customer (void)
{
    down(&mutex);
    if (waiting < CHAIRS)
    {
waiting = waiting + 1;
up(&customers);
up(&mutex);
down(&barbers);
get_haircut( );
    }
    else
        {up(&mutex);}
}
```

DICA: a região crítica **NÃO** é a cadeira do barbeiro.

Resposta: Problema do Barbeiro com semáforos.

a)
mutex: semáforo binário; protege região crítica;
customers: semáforo inteiro; sincronismo, 0: barbeiro dorme, 1: barbeiro acordado e clientes esperando.
barbers: semáforo inteiro; sincronismo, 0: barbeiro ocupado, 1: barbeiro esperando.

b) Região Crítica: waiting

c)
i- a variável waiting (região crítica) poderá ser acessada por customers e barbers sem restrição; pode haver mais clientes na barbearia do que o número de cadeiras.
ii- Barbeiro e cliente ficarão dormindo. O Barbeiro ficará dormindo em down (&mutex) e o cliente ficará dormindo em down (&barbers). Não haverá cortes.

20) Explique qual o significado do termo ***espera ocupada***? Que outros tipos de espera existem em um sistema operacional? A espera ocupada pode ser completamente evitada? Explique sua resposta.

Resposta: Espera ocupada é um modelo de programação paralela caracterizado por testes repetidos de um condição que impedem o progresso de um processo e que só pode ser alterada por outro processo. Ela pode ser evitada por meio de primitivas como semáforos, block e wake-up, monitores, troca de mensagens.

21) Qual a finalidade das chamadas do sistema? Que chamadas do sistema precisam ser executadas por um interpretador de comandos ou *shell* a fim de iniciar um novo processo?

Resposta: As chamadas de sistema permitem que processos de nível de usuário solicitem serviços do sistema operacional. Em um sistema UNIX tem-se que executar uma chamada FORK, seguida por uma chamada de sistema exec. precisa ser executada para iniciar um processo novo. A chamada FORK clona o processo em execução corrente, enquanto a exec substitui o processo que fez a chamada por um novo processo, com um executável diferente.

22) Qual a relação entre programa e processo?

Resposta: Um processo é um programa em execução.