



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 AP1 - Primeiro Semestre de 2013

Nome -
Assinatura -

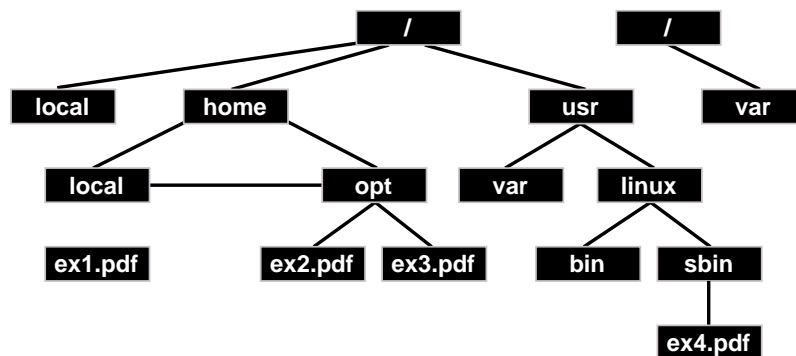
Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (1,5) Suponha que dois processos, A e B, estejam em execução no mesmo processador de tal forma que o sistema operacional somente troca um processo pelo outro quando o primeiro executa uma operação de E/S. Se A e B fazem, cada um, apenas uma operação de E/S, e se A é o primeiro a ser executado, explique as condições para que o processador não fique ocioso antes do término de B.

Resp.: Como somente os processos A e B estão em execução no processador então, para evitar a ociosidade do processador quando a operação de E/S de A for feita, o tempo de execução no processador de B antes de ele executar a sua operação de E/S deve ser maior do que o tempo da operação de E/S executada por A. Agora, para que o processador não fique ocioso quando B executar a sua operação de E/S, o tempo restante da execução de A, ou seja, o tempo que ele ainda precisa executar no processador até terminar após a operação de E/S feita por ele terminar, deve ser maior do que o tempo da operação de E/S de B.

2. (1,5) Um aluno de sistemas operacionais disse que a figura a seguir representa corretamente a estrutura hierárquica de um possível sistema de arquivos. Se você acha que o aluno está correto, basta responder que sim mas, se você acha que o aluno está errado, aponte os erros que existem na figura.



Resp.: O aluno está incorreto, pois existem três erros na sua hierarquia. O primeiro erro é que existem duas raízes no sistema de arquivos quando deveria existir apenas uma. O segundo erro é que existe uma

relação de parentesco entre o arquivo (ou diretório) local com o diretório `opt`, o que não é possível devido aos dois estarem no mesmo nível da hierarquia (dentro do diretório `/home`). Finalmente, o terceiro erro é o arquivo `ex1.pdf` no quarto nível da hierarquia não possui uma relação de parentesco com nenhum diretório do terceiro nível.

3. (2,5) Diga se as seguintes afirmativas são falsas ou verdadeiras. Para responder, escreva apenas F ou V para cada item em seu caderno de respostas.

- (a) (0,5) A multiprogramação não é visível ao usuário do sistema operacional quando ela é gerenciada pelo sistema.

Resp.: F (Falsa), pois a multiprogramação somente não é visível ao usuário se ela for gerenciada por uma extensão do sistema operacional, como ocorre ao usarmos as máquinas virtuais.

- (b) (0,5) As máquinas virtuais são criadas e gerenciadas pelo monitor de máquina virtual, o qual também mapeia todos os acessos das máquinas virtuais aos dispositivos virtuais nos dispositivos físicos correspondentes do hardware.

Resp.: V (Verdadeira).

- (c) (0,5) No sistema operacional em camadas, apesar de o núcleo ser igual ao do sistema monolítico e não apresentar uma estruturação, as chamadas ao sistema operacional são organizadas em camadas definidas de acordo com as suas funcionalidades.

Resp.: F (Falsa), pois é o núcleo do sistema operacional que é organizado em camadas, divididas de acordo com as diversas tarefas de gerenciamento executadas pelo núcleo.

- (d) (0,5) A hierarquia em anéis, além de fornecer uma estruturação para o núcleo do sistema baseada em anéis concêntricos, obriga

que a abstração definida pelos anéis internos não seja violada pelos anéis mais externos.

Resp.: V (Verdadeira).

- (e) (0,5) No modelo cliente-servidor, o micronúcleo é o responsável pelo gerenciamento dos processos servidores do sistema.

Resp.: F (Falsa), pois o micronúcleo é o responsável pelo acesso direto aos dispositivos físicos (o gerenciamento deles é feito por processos executando no modo usuário) e pela troca de mensagens entre os diversos processos em execução no modo usuário.

4. (1,5) Diga a quais conceitos vistos em aula se referem as seguintes definições:

- (a) (0,5) Tipo de paralelismo que ocorre quando um sistema operacional multiprogramado executa em um computador com um único processador de somente um núcleo.

Resp.: Pseudoparalelismo.

- (b) (0,5) Estado no qual um processo se encontra quando ele está sendo executado em um dos processadores do computador.

Resp.: Executando ou Rodando.

- (c) (0,5) Estrutura de dados do sistema operacional que armazena, para cada processo, todas as informações necessárias para que ele seja executado corretamente.

Resp.: Tabela de processos.

5. (1,5) Suponha que um contador armazenado em um endereço da memória seja compartilhado por dois processos, A e B. O processo A

repetidamente espera até que o contador seja maior do que 0 e, quando isso ocorre, decrementa-o em 2 unidades. Já o processo B repetidamente incrementa o contador em 1 unidade. Explique como executar A e B fazendo uso de semáforos binários para garantir o correto compartilhamento do contador. Justifique a sua resposta.

Resp.: Vamos precisar de dois semáforos binários, o semáforo *acesso* que é usado para garantir o acesso exclusivo ao contador, e o semáforo *bloquearA* que é usado para bloquear A enquanto o valor do contador não for maior do que 0. Como inicialmente o contador não está sendo usado, o valor inicial do semáforo *acesso* é 1. Já o valor inicial do semáforo *bloquearA* depende do valor inicial do contador, e deve ser 0 se o contador for menor do que 1 ou igual a 1 em caso contrário. A seguir mostramos os possíveis códigos para os processos A e B, escritos na linguagem C. Nos códigos, supomos que a variável *contador* aponta para o endereço de memória com o contador compartilhado pelos dois processos.

```
void ProcessoA(void)
{
    while (1)
    {
        P(acesso);
        if (contador < 1)
        {
            V(acesso)
            P(bloquearA)
            P(acesso)
        }
        contador = contador - 2;
        V(acesso);
    }
}
```

```
void ProcessoB(void)
{
```

```

while (1)
{
    P(acesso);
    contador = contador + 1;
    if (contador > 0)
        V(bloquearA)
    V(acesso);
}
}

```

6. (1,5) Suponha que um processo A tenha sido executado com exclusividade no processador por t unidades de tempo. Suponha agora que outros $n - 1$ processos venham a compartilhar o processador com A e que eles precisem das mesmas t unidades de tempo para terminarem as suas execuções. Se A for o último processo a executar, qual será o tempo decorrido até o seu término se o sistema operacional usar o algoritmo por *round robin* com um quantum de t/u unidades de tempo, sendo u um número positivo qualquer? Justifique a sua resposta.

Resp.: Como o quantum tem tamanho de t/u unidades de tempo, então cada processo precisa executar exatamente por u quanta no processador, pois o tempo de execução de todos os processos é igual a t . Agora, como A é o último a executar, então ele é o último a terminar, significando que o tempo decorrido até A terminar é igual a todos os quanta executados por todos os processos multiplicado pelo tamanho de um quanta, ou seja, o tempo decorrido é de $n \times u \times \frac{t}{u} = nt$ unidades de tempo.