



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 - Segundo Semestre de 2012

Nome -  
Assinatura -

---

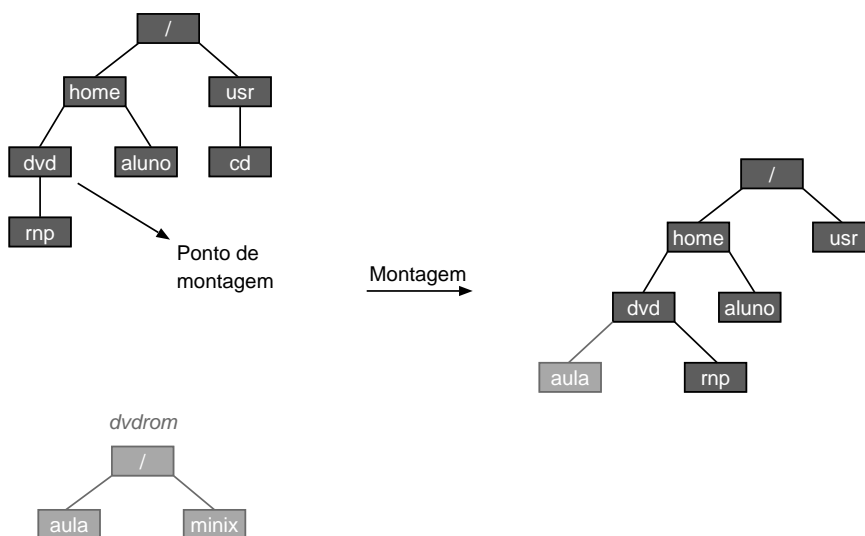
Observações:

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 um programa executando em um sistema operacional de segunda geração demore  $t$  segundos para terminar. Suponha ainda que o desperdício de tempo devido à execução de operações de E/S seja de  $x\%$ . Por quanto tempo o programa efetivamente executa no processador? Justifique a sua resposta.

**Resp.:** Como o programa executa em um sistema da segunda geração, então o tempo de E/S está incluído no tempo de  $t$  segundos que o programa leva para terminar. Agora, como  $x\%$  desse tempo é gasto com E/S, então  $(100 - x)\%$  de  $t$  é o tempo no qual o programa executa no processador. Logo, o tempo que o programa executa efetivamente no processador é de  $\left(\frac{100-x}{100}\right)t$  segundos.

2. (1,5) Um aluno de sistemas operacionais disse que a figura a seguir representa corretamente o que ocorre quando um sistema de arquivos (dvdrom) é montado em um outro 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 errado porque existem três erros na figura. O primeiro erro é que o diretório **cd** deveria estar no sistema de arquivos obtido após a montagem, porque seu pai, o diretório **usr**, não tem

relação com o ponto de montagem do dvdrom. O segundo erro é que o diretório **rnp** não poderia estar no sistema de arquivos obtido após a montagem, porque ele é subdiretório do diretório **dvd** usado como ponto de montagem. Como vimos na Aula 2, quando uma montagem é feita em um diretório, todo o seu conteúdo fica indisponível até que o sistema de arquivos montado nele seja desmontado. Finalmente, o terceiro erro é que o sistema de arquivos do dvdrom não foi totalmente montado, porque o diretório **minix** não passou a ser subdiretório do diretório **dvd**.

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) Uma máquina virtual é uma cópia da máquina estendida, sendo portanto tão fácil de ser usada quanto essa última.

**Resp.:** F (Falsa), porque cada máquina virtual é uma cópia do hardware do computador, sendo portanto tão difícil de ser usada quanto esse hardware.

- (b) (0,5) No gerenciamento da multiprogramação usando um exonúcleo, todos os recursos do hardware são divididos entre as diversas máquinas virtuais.

**Resp.:** V (Verdadeira).

- (c) (0,5) Nos sistemas monolíticos, o núcleo do sistema operacional é bem organizado, sendo dividido em módulos com funções bem definidas.

**Resp.:** F (Falsa), pelo contrário, os sistemas monolíticos são caracterizados exatamente pela falta de organização do núcleo do sistema operacional.

- (d) (0,5) Para garantir a abstração, a estruturação do núcleo em anéis é formada por um conjunto de anéis concêntricos, sendo que os anéis mais externos somente podem acessar os anéis mais internos de um modo similar a como um processo do usuário faz uma chamada ao sistema operacional.

**Resp.:** V (Verdadeira).

- (e) (0,5) Em um sistema cliente-servidor, o micronúcleo trata somente do acesso mais básico aos dispositivos físicos e da troca de mensagens entre os processos em execução no modo usuário.

**Resp.:** V (Verdadeira).

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

- (a) (0,5) Conjunto de instruções necessário para a execução de uma dada tarefa.

**Resp.:** Programa.

- (b) (0,5) Estado do processo após ter a sua execução suspensa para que um outro processo possa passar a executar no processador.

**Resp.:** Pronto.

- (c) (0,5) Algoritmo do sistema operacional que gerencia a alocação dos processadores do computador aos processos do sistema.

**Resp.:** Escalonador.

5. (1,5) Considere dois processos A e B que compartilhem uma mesma fila, sendo que A coloca elementos nela e B remove elementos dela.

A seguir mostramos códigos para esses processos usando o semáforo binário *acesso* e os semáforos de contagem *vazias* e *cheias*. O que ocorreria se A não executasse a operação **P** sobre *vazias* antes de tentar colocar um elemento na fila, supondo que os semáforos tenham sido corretamente inicializados? E se A não executasse a operação **V** sobre *cheias* após inserir um elemento na fila?

```
void ProcessoA()
{
    while (1)
    {
        P(vazias);
        P(acesso);
        InsereElementoFila();
        V(acesso);
        V(cheias);
    }
}
```

```
void ProcessoB()
{
    while (1)
    {
        P(cheias);
        P(acesso);
        RemoveElementoFila();
        V(acesso);
        V(vazias);
    }
}
```

**Resp.:** -Se a operação **P** sobre *vazias* não fosse executada antes de o processo A colocar elementos na fila, então A não iria verificar se ela está cheia antes de inserir um novo elemento. Isso certamente causaria um erro de execução em A porque, como não existiria espaço na fila, ele poderia abortar por ter acessado uma posição de memória inválida não pertencente à fila, ou perder o novo elemento criado devido a não

ter como salvá-lo, ou até mesmo sobre-escrever um dos elementos da fila.

-Se a operação **V** sobre *cheias* não fosse executada pelo processo A depois de inserir um elemento na fila, então os elementos inseridos por A na fila não seriam contabilizados. Portanto, após remover todos os elementos inseridos inicialmente na fila antes da execução dos processos A e B, B ficaria bloqueado para sempre no semáforo *cheias* esperando incorretamente por novos elementos. Além disso, depois de B ficar bloqueado, A também ficaria bloqueado para sempre depois de preencher todas as entradas vazias da fila. Logo, após A preencher essas entradas, um impasse passaria a existir envolvendo os processos A e B.

6. (1,5) Suponha que três processos, A, B e C, sejam executados no processador na ordem AACBCBAC. Suponha também que todo processo em execução seja interrompido a cada 2 unidades de tempo para uma decisão do escalonador. Ou seja, na sequência AACBCBAC, cada ocorrência de um processo corresponde a no máximo 2 unidades de tempo. Se o tempo total de execução de B for um inteiro ímpar e os tempos totais de execução de A e C forem inteiros pares, qual seria a sequência de execução se o sistema operacional passasse a usar o algoritmo *round robin* com um quantum de 1 unidade de tempo? Justifique a sua resposta.

**Resp.:** Pelo enunciado, os processos A e C executam por 2 unidades de tempo em todas as suas ocorrências na ordem de execução AACBCBAC, porque seus tempos de execução são inteiros pares. Já o processo B executa por 2 unidades de tempo na primeira ocorrência e somente por 1 unidade de tempo na última ocorrência, porque o tempo de execução de B é um inteiro ímpar. Com isso, os tempos de execução dos processos A, B e C são de, respectivamente, 6, 3 e 6 unidades de tempo. Agora, no algoritmo *round robin*, cada processo executa por um quantum de 1 unidade de tempo e somente volta a executar após todos os outros processos que ainda não terminaram terem executado também por um quantum. Logo, teríamos uma das seguintes ordens de execução: ABCABCABCACACAC, ACBACBACBACACAC, BACBACBACACACAC, BCABCABCACACACA, CABCABCABCACACA ou CBACBACBACACACA. Para a sua resposta ser

considerada como correta, basta ter informado uma dessas ordens.