



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  
AP3 - Segundo Semestre de 2008

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. (2.0) Qual é a diferença entre a multiprogramação do processador do hardware e a definição de máquinas virtuais que são cópias exatas deste hardware?

**Resp.:** A diferença está em quais dispositivos do hardware são compartilhados, e no conhecimento deste compartilhamento pelas entidades que compartilham estes dispositivos. Na multiprogramação, o processador do hardware é compartilhado entre os diversos processos em execução no sistema operacional. Com isso, cada processo executa em um processador virtual que é mapeado, de tempos em tempos, no processador do hardware, através das trocas de contexto conforme os processos são escolhidos pelo escalonador. Além disso, cada processo sabe da existência dos outros processos do sistema, e do compartilhamento do processador com estes processos. Agora, as máquinas virtuais, que são cópias exatas da máquina real, compartilham entre si todos os dispositivos do hardware. Neste caso, ao contrário do que ocorre com os processos na multiprogramação, os programas em execução em cada uma das máquinas virtuais (em geral, sistemas operacionais) não sabem da existência uns dos outros. Além disso, estes programas também não sabem que estão compartilhando todos os dispositivos do hardware entre si, pois acham que estão executando exclusivamente no hardware.

2. (2.0) Suponha que dois processos, A e B, compartilham uma região de memória  $R$ . Suponha ainda que A executa por 3 unidades de tempo e que B executa por 2 unidades de tempo. Existirá uma condição de corrida, se o sistema operacional usar:
  - (a) (1.0) O algoritmo por *round robin*, sendo que cada quantum equivale a 2 unidades de tempo?

**Resp.:** Sim, quando A começar a execução antes de B. Neste caso, A executará por um quantum e não terminará a sua execução, pois precisa executar por 3 unidades de tempo. Depois disso, B executará por um quantum e terminará a sua execução ao final do quantum, pois precisa executar por 2 unidades de tempo. Como A ainda não terminou, a execução de B após A poderá gerar condições de corrida. Finalmente, A executará por mais um

quantum, usando somente a sua metade, pois precisa executar por somente 1 unidade de tempo. Note que não existirão condições de corrida se B executar antes de A, pois B executará por um quantum e terminará a sua execução ao final deste quantum, antes de A começar a executar, pois B precisa executar por somente 2 unidades de tempo.

- (b) (1.0) O algoritmo por prioridades, sendo que a prioridade é reduzida a cada unidade de tempo, e que a prioridade de A é 9 e a de B é 3?

**Resp.:** Neste caso, não existirão condições de corrida, porque B somente executará depois de A terminar a sua execução após 3 unidades de tempo. Isso ocorrerá porque, como a prioridade de A é 9 e a de B é 3, então A poderá executar por até 5 unidades de tempo antes de B ser executado pela primeira vez (no caso de o processo executar até que a sua prioridade deixe de ser a maior).

3. (2.0) Por que o conceito de independência de dispositivos foi introduzido nos sistemas operacionais? Este conceito é essencial para o projeto do Software de E/S em camadas?

**Resp.:** -O conceito foi inventado para que as operações executadas sobre qualquer dispositivo físico de um dado tipo (por exemplo, as diversas marcas de um disco rígido) por um processo, sejam independentes do funcionamento interno deste dispositivo. Para isso, o sistema operacional define um dispositivo abstrato com operações genéricas aplicáveis a todos os dispositivos deste tipo. Além disso, o sistema trata, de modo transparente, dos detalhes do funcionamento de cada um destes dispositivos, mapeando as operações do dispositivo abstrato para as operações correspondentes deste dispositivo.

-Sim, pois o conceito de independência de dispositivo é usado ao implementarmos as camadas superiores no projeto do Software de E/S em camadas, porque as operações sobre os dispositivos abstratos fazem parte da interface simples e fácil de usar fornecida por estas camadas.

4. (2.0) Qual é a principal vantagem de usarmos uma tabela de páginas multinível? E qual é a principal desvantagem? Esta desvantagem pode ser atenuada se usarmos uma TLB?

**Resp.:** -A principal vantagem é a de que reduzimos o espaço gasto na memória para armazenar a tabela de páginas de cada um dos processos do sistema operacional. Isso ocorrerá porque o objetivo de usarmos uma tabela de páginas multinível, que divide a tabela de um processo em várias partes, é o de manter, na memória, somente aquelas partes associadas às páginas deste processo que estão sendo usadas. As outras partes da tabela são salvas no disco. Porém, se fosse usada uma tabela de páginas normal, então a tabela de cada processo deveria ser totalmente armazenada na memória ou no disco.

-A principal desvantagem é a de que o tempo de conversão de um endereço virtual para um endereço físico será maior. Como vimos, o endereço virtual é dividido em dois campos, o número da página virtual e o deslocamento dentro desta página. Ao usarmos uma tabela de páginas multinível, o campo número da página será adicionalmente dividido em um número de campos igual ao número de níveis da tabela. Logo, o tempo de conversão será maior porque deveremos usar cada campo, em seqüência, para acessar a entrada da tabela do nível correspondente. Esta entrada indicará ou a tabela do próximo nível, caso o campo não seja o último da seqüência, ou as informações da página, no caso contrário. Note que o tempo será ainda maior para as tabelas acessadas que não estão na memória, pois precisaremos copiá-las do disco antes de acessarmos a sua entrada. Agora, se usarmos uma tabela de páginas normal, somente precisaremos acessar uma entrada de uma tabela, apontada pelo campo número da página e, se necessário, carregar somente uma tabela do disco.

-Sim, pois como vimos na Aula 8, a TLB é usada para armazenar o mapeamento das páginas mais acessadas pelo processo em execução. Logo, o processo descrito na resposta anterior, necessário para obtermos as informações da página acessada, somente será executado mais de uma vez para as páginas menos acessadas pelo processo. Note que as informações de cada página mais acessada sempre estarão na TLB depois de terem sido obtidas quando esta página foi acessada pela primeira vez.

5. (2.0) Suponha que o sistema operacional usa uma cache com os blocos mais referenciados do disco. Neste caso, o que o sistema fará ao ler um bloco do disco? E ao escrever um bloco no disco?

**Resp.:** -Quando usamos uma cache com os blocos mais referenciados do disco e desejamos ler um bloco do disco, primeiramente verificamos se este bloco está na cache. Se o bloco estiver na cache, então passaremos, a quem pediu este bloco, o conteúdo dele armazenado na cache. Porém, se o bloco não estiver na cache, leremos este bloco do disco e, para que futuras referências ao bloco sejam satisfeitas pela cache, copiaremos o seu conteúdo na cache. Se a cache estiver cheia, um dos blocos armazenados nela deverá ser removido, e também salvo no disco caso tenha sido alterado e ainda não salvo (isso ocorrerá se o sistema não usar uma cache de gravação intermediária).

-Quando usamos uma cache com os blocos mais referenciados do disco e desejamos escrever um bloco no disco então, assim como na leitura, inicialmente verificamos se este bloco está na cache. Caso ele não esteja na cache, ele é escrito no disco, e depois o copiamos na cache (removendo, se necessário, um dos blocos da cache), para que futuras referências a ele sejam satisfeitas pela cache. Como a alteração de blocos da cache pode gerar inconsistências no sistema de arquivos se eles não forem imediatamente salvos no disco, precisamos lidar com o que será feito neste caso. Se a cache for a de gravação intermediária, então o bloco será imediatamente salvo no disco e, com isso, inconsistências não ocorrerão, pois a cache nunca terá blocos alterados que não foram salvos no disco. Em caso contrário, um bloco alterado somente será salvo quando for removido da cache e, devido a isso, poderão ocorrer inconsistências no sistema de arquivos. Note que alguns sistemas que usam este tipo de cache salvam, de tempos em tempos, os blocos alterados no disco, para tentar minimizar a possibilidade destas inconsistências ocorrerem.