

Sistemas operacionais

AP1 2009-1

01. Para os usuários do computador, qual é a diferença essencial entre os sistemas de lote e os sistemas de compartilhamento de tempo?

Resp.: A diferença é que em um sistema de lote, os usuários submetem os seus programas para execução no computador. Posteriormente, os programas de diversos usuários são organizados em lotes pelos operadores, e são colocados, por estes, para executarem no computador. Após as execuções, os resultados dos programas são coletados pelos operadores (por exemplo, eles são impressos), e são disponibilizados aos usuários que originalmente submeteram os programas. Em essência, não existe nenhuma interação do usuário com o computador, somente do usuário com os operadores. Já em um sistema de compartilhamento de tempo, os usuários acessam o computador a partir de terminais, portanto interagindo diretamente com o computador. Além disso, os próprios usuários, que têm a ilusão de estarem usando exclusivamente o computador, colocam os programas para executarem no computador e coletam os resultados destas execuções após o término dos programas.

02. Qual é a principal vantagem de uma hierarquia de diretórios em relação a uma abordagem alternativa em que todos os arquivos do sistema operacional sejam armazenados em um mesmo diretório?

Resp.: Quando todos os arquivos são armazenados em um mesmo diretório, além de não poderem existir dois ou mais arquivos com mesmo nome, também não se podem organizar os arquivos, em categorias, de acordo com os seus conteúdos. Agora, se o sistema operacional possui uma hierarquia de diretórios, podem existir dois ou mais arquivos com o mesmo nome, desde que sejam armazenados em diretórios diferentes. Além disso, a própria hierarquia, em que os diretórios de um nível, diferente do primeiro, são subdiretórios do diretório do nível imediatamente anterior (o diretório do primeiro nível, o diretório raiz, não é subdiretório de nenhum diretório), permite naturalmente organizar os arquivos de acordo com os seus conteúdos. Por exemplo, em uma hierarquia de diretórios, cada usuário do sistema operacional pode ter o seu próprio diretório, e pode criar, por exemplo, um subdiretório para figuras, um para textos e um para músicas. Se existisse somente um diretório, não somente todos os arquivos de todos os usuários deveriam ser armazenados nele, como um usuário não poderia criar subdiretórios e dividir os seus arquivos de acordo com os seus conteúdos.

03. A multiprogramação precisa ser visível ao usuário do computador? Justifique a sua resposta.

Resp.: Não, a multiprogramação não precisa ser visível ao usuário do computador. Se o sistema operacional executar no hardware e implementar a multiprogramação, então ela será visível ao usuário do computador. Isso ocorrerá porque ele saberá que cada programa é executado por um processo diferente do sistema operacional, e que estes processos compartilham o(s) processador(es) do hardware. Agora, se o sistema operacional executar em uma máquina virtual, o usuário do computador não saberá que o processador do hardware está sendo multiprogramado, pois terá a impressão de que o sistema operacional está executando exclusivamente no computador. Quem será o responsável por multiprogramar o processador, neste caso, será o monitor de máquina virtual, que irá mapear, de tempos em tempos, o processador de cada máquina virtual no processador do hardware. Note que o usuário saberá, porém, da multiprogramação do processador da máquina virtual, se o sistema operacional for multiprogramado.

04. Como o modelo de processos facilita o entendimento e a implementação da multiprogramação?

Resp.: No modelo de processos, o sistema operacional é visto como um conjunto de processos sequenciais. Para cada processo, as informações necessárias à sua execução, incluindo os registradores do processador, são salvas no seu contexto quando ele deixa de executar no processador, e são restauradas quando ele voltar a executar no processador. Logo, podemos imaginar que cada um destes processos está executando em um processador virtual cujo conteúdo dos registradores são exatamente

aqueles que foram salvos no contexto do processo. Portanto, o conceito facilitará o entendimento da multiprogramação, pois ela ocorrerá naturalmente quando alternarmos o uso do processador do hardware entre estes processadores virtuais. O conceito também facilitará a implementação da multiprogramação, pois bastará salvarmos os registradores do processador no contexto do processo quando ele deixar de executar no processador, e restaurar os registradores do contexto imediatamente antes de o processo voltar a executar no processador.

05. Descreva como os semáforos são implementados no sistema operacional, destacando o modo de funcionamento das operações P e V.

Resp.: Cada semáforo do sistema operacional é composto por um valor inteiro e por uma fila que armazena as informações sobre cada processo bloqueado por causa deste semáforo. As operações P e V são implementadas no núcleo do sistema operacional de modo atômico, isto é, os códigos que as implementam são executados do início até o fim sem interrupções, para garantir a exclusão mútua ao acessar o valor e a fila. A ação executada pela operação P dependerá do valor associado ao semáforo. Se este valor for maior do que 0, a operação simplesmente o decrementará em uma unidade. Porém, se este valor for igual a 0, o processo que executou a operação será bloqueado, e as informações que o identificam serão inseridas na fila do semáforo. A ação executada pela operação V também dependerá do valor do semáforo. Se o valor for maior do que 0, ele será incrementado em uma unidade se o semáforo for de contagem (pois o valor de um semáforo Bin somente pode ser 0 ou 1). Se o valor for 0 e a fila não estiver vazia, um dos processos cujas informações estão na fila será escolhido aleatoriamente, as suas informações serão removidas da fila, e ele será desbloqueado e colocado no estado pronto, para depois poder ser escolhido pelo escalonador. Finalmente, se o valor for 0 e a fila estiver vazia, ele também será incrementado em uma unidade.

06. O algoritmo de escalonamento por round Robin precisa ser primitivo, ou pode ser também não-preemptivo? Justifique a sua resposta.

Resp.: O algoritmo precisa ser preemptivo, como veremos a seguir. Como vimos na Aula 6, no algoritmo de escalonamento por round Robin, cada processo executa no processador por no máximo um quantum (um intervalo fixo de tempo definido pelo sistema operacional). Após executar por um quantum, o processo somente executará novamente, por mais um quantum, após todos os outros processos no estado pronto executarem também por um quantum. Para garantir o funcionamento do algoritmo, precisamos executar o escalonador em intervalos fixos de tempo (iguais ao quantum), para que ele possa alternar o uso do processador entre os processos no estado pronto. Isso não poderá ser feito se o algoritmo for não-preemptivo porque, neste caso, o escalonador somente será executado se o processo atualmente em execução fizer alguma operação de E/S ou terminar a sua execução. Já um algoritmo preemptivo, além de executar o escalonador nestes casos, o executa também em intervalos fixos de tempo, usando o temporizador do hardware para gerar uma interrupção entre cada um destes intervalos. Logo, para poder alternar o uso do processador entre os processos em intervalos fixos de tempo iguais ao quantum, precisaremos que o algoritmo seja preemptivo.

01. Defina as multiplexações dos recursos gerenciados pelo sistema operacional: (i) por tempo e (ii) por espaço.

Resp.: A multiplexação dos recursos por tempo é usada pelo sistema operacional para gerenciar os recursos que devem ser usados, de modo exclusivo, por um dado processo (ou programa) durante um dado intervalo de tempo. Já a multiplexação dos recursos por espaço é usada pelo sistema operacional para gerenciar os recursos que podem ser compartilhados por mais de um processo (ou programa) ao mesmo tempo, sendo que somente uma parte do recurso (e não todo o recurso) é usada por cada processo. Por exemplo, o processador do computador (se não for um processador multicore) pode somente executar um processo de cada vez por um dado intervalo de tempo e, por isso, precisa ser um recurso multiplexado por tempo. Já a memória principal do computador precisa ser um recurso multiplexado por espaço, pois ela é dividida entre os diversos processos em execução no computador.

02. 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) Em uma árvore de processos, o fato de um nó possuir filhos significa que o processo correspondente ao nó criou todos os processos correspondentes aos filhos.

Resp.: V (Verdadeira).

(b) O nome de caminho absoluto é obtido a partir do diretório de trabalho, e o nome de caminho relativo a partir do diretório raiz do sistema de arquivos.

Resp.: F (Falsa, pois as duas definições, de caminho absoluto e de caminho relativo, estão trocadas).

(c) Qualquer sistema de arquivos pode ser acessado sem ser montado, independentemente de conter ou não o sistema operacional.

Resp.: F (Falsa, pois a necessidade ou não de montarmos um sistema de arquivos depende do sistema operacional. Por exemplo, no Linux a montagem é necessária, enquanto no Windows não é).

(d) Os arquivos especiais de bloco são usados para representar os dispositivos divididos em conjuntos de blocos que podem ser acessados aleatoriamente. Os arquivos especiais de caractere são usados para representar os dispositivos modelados por fluxos de caracteres.

Resp.: V (Verdadeira).

(e) Ao executar uma chamada ao sistema operacional, não é necessário mudarmos o processador do modo usuário para o modo supervisor se usarmos uma biblioteca para fazer a chamada, porque o código desta biblioteca executa no modo supervisor.

Resp.: F (Falsa, pois as bibliotecas executam no modo usuário.)

03. Qual é a diferença entre um sistema operacional estruturado em camadas e um sistema operacional baseado em anéis? Justifique a sua resposta.

Resp.: Como vimos na Aula 3, em ambas as estruturas o núcleo do sistema operacional é dividido em diversas camadas organizadas de modo hierárquico, sendo que cada uma destas camadas gerencia uma parte do hardware do computador. Além disso, cada camada oferece, às camadas superiores a ela na hierarquia, uma versão abstrata da parte do hardware mais fácil de ser usada. A diferença está em que em um sistema operacional estruturado em camadas, uma camada não é obrigada a usar as versões abstratas das partes do hardware gerenciadas pelas camadas inferiores a ela na hierarquia, isto é, ela pode acessar diretamente as partes do hardware. Com isso, a abstração fornecida pela estruturação não é obrigatória e somente serve para organizar o código do núcleo. Já em um sistema operacional baseado em anéis, a estruturação é forçada pelo hardware do computador. Nesta última estruturação, um anel (é assim que uma camada é chamada nesta estruturação) somente pode acessar as versões abstratas

das partes do hardware gerenciadas pelos anéis inferiores da hierarquia. Isto ocorre porque o hardware sempre gera um erro se um anel tenta acessar diretamente uma dessas partes do hardware. Logo, a abstração fornecida pela estruturação deve sempre ser respeitada e, portanto, não é somente uma organização para o código do núcleo.

04. Como vimos nas aulas, existem três estados, executando, pronto e bloqueado, em que um processo pode estar em um dado momento. É possível desenvolver um sistema operacional sem um destes estados? Justifique a sua resposta.

Resp.: Sim, é possível desenvolver um sistema operacional sem o estado bloqueado. Neste caso, ao invés de suspendermos a execução de um processo quando ele precisar esperar pela ocorrência de um evento externo, podemos simplesmente colocar o processo em espera ocupada. Isto pode ser feito com uma verificação contínua da ocorrência do evento e permitindo que o processo somente saia deste laço de verificação quando o evento ocorrer. Em relação aos outros estados, eles sempre são necessários, pois mesmo que o sistema operacional não use a multiprogramação e não bloqueie os processos, ainda precisamos suspender a execução dos processos. Isto ocorre quando precisamos tratar os eventos externos prioritários, como as interrupções geradas pelos dispositivos de E/S, e não existir um processador disponível para executar o código do tratador da interrupção.

05. Descreva como os semáforos podem ser usados para simplificar o tratamento das interrupções dos dispositivos de E/S.

Resp.: Os semáforos poderão facilitar o tratamento das interrupções se associarmos, a cada um dos dispositivos de E/S do computador, um semáforo binário, o qual será usado para bloquear o processo que executa o driver deste dispositivo até que o dispositivo gere uma interrupção. Quando uma interrupção for gerada pelo dispositivo, uma operação V sobre o semáforo associado ao dispositivo será executada pelo sistema operacional, o que fará com que o seu driver seja desbloqueado e possa tratar a interrupção. Além disso, ao inicializar e sempre após tratar de uma interrupção, o driver de um dispositivo deverá executar uma operação P sobre o semáforo associado ao dispositivo. Finalmente, como um dispositivo somente gera interrupções após uma operação de E/S ser inicializada sobre ele, o valor inicial do seu semáforo deverá ser de 0, o que fará com que o seu driver seja bloqueado ao inicializar até que uma interrupção seja gerada.

06. Descreva o funcionamento do escalonamento de dois níveis, enfatizando por que o seu uso é necessário.

Resp.: Dependendo do número de processos em execução no sistema operacional, nem sempre podemos armazenar todos eles na memória, ou seja, alguns processos devem necessariamente ser armazenados no disco. Porém, como os tempos de comutação dos processos são bem maiores se a comutação é feita diretamente do disco, devido ao tempo de acesso ao disco ser bem maior do que à memória, é inviável usarmos um algoritmo de escalonamento tradicional. O algoritmo de escalonamento de dois níveis, descrito a seguir, foi criado exatamente para resolver este problema. Neste algoritmo, dois escalonadores (e portanto dois algoritmos de escalonamento não necessariamente iguais) serão usados: o escalonador de baixo nível e o escalonador de alto nível. O escalonador de baixo nível executa a tarefa do escalonador tradicional, isto é, ele é o responsável por escolher, dentre os processos prontos que estão na memória, qual deles é o próximo a ser executado no processador. Já o escalonador de alto nível escolhe qual dos processos do disco deve ser carregado na memória, no lugar de um outro processo, também escolhido, que é então salvo no disco. Note que o tempo de comutação dos processos não depende mais do tempo de acesso ao disco, porque o escalonador de baixo nível somente escolhe um processo dentre os processos que estão na memória.

01. Por que os programas de usuário não podem executar no modo supervisor? O núcleo do sistema operacional pode executar no modo usuário?

Resp.: -Um programa de usuário não pode executar no modo supervisor porque, neste modo, ele teria acesso direto a todo o hardware do computador. Devido a isso, a integridade dos dispositivos físicos poderia ser comprometida, porque cada programa de usuário poderia acessar o mesmo dispositivo de um modo diferente (por exemplo, os dados armazenados em um disco poderiam ser comprometidos). Além disso, também poderíamos ter problemas de segurança e de consistência durante a execução dos programas, porque um programa poderia acessar e/ou alterar os dados usados exclusivamente por um outro programa. Finalmente, um programa poderia executar do seu início até o seu término, mesmo que o sistema operacional usasse a multiprogramação, porque ele poderia usar exclusivamente o processador.

- O núcleo do sistema operacional não pode executar no modo usuário, porque ele precisaria ter acesso direto aos dispositivos físicos para, pelo menos, poder enviar comandos para as suas controladoras (o gerenciamento dos dispositivos pelo núcleo depende da estrutura na qual o sistema operacional foi baseado). Além disso, o sistema também precisaria gerenciar a memória alocada aos programas, para poder distribuí-la entre eles, e garantir que um programa não acesse a memória dos outros programas. Finalmente, o sistema precisaria gerenciar a alocação do processador entre os diversos programas em execução.

02. 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) Cada processo do sistema operacional possui uma identificação (UID) única. Além disso, existe um espaço de endereçamento associado a cada processo com o seu código, os seus dados, e a sua pilha de execução.

Resp.: F (Falsa, pois a identificação de um processo é denominada PID; UID se refere à identificação do usuário).

(b) Em uma árvore de processos, quando dois processos estiverem ligados por uma aresta, isso significará que um dos processos dependerá de um resultado da execução do outro processo.

Resp.: F (Falsa, pois uma aresta ligando dois processos de uma árvore de processos indica que um deles criou o outro).

(c) Normalmente, para podermos acessar o sistema de arquivos com o sistema operacional, precisaremos primeiramente montá-lo em algum diretório de um outro sistema de arquivos.

Resp.: F (Falsa, pois é exatamente o contrário, ou seja, para usarmos um sistema de arquivos que não contenha o sistema operacional, precisaremos montá-lo antes em algum diretório do sistema de arquivos com o sistema operacional).

(d) Um pipe permite a troca de dados de modo transparente entre dois processos, porque conecta a saída de um dos processos à entrada do outro processo.

Resp.: V (Verdadeira).

(e) As bibliotecas são, em geral, usadas para fazermos as chamadas ao sistema operacional, porque elas fornecem vários modos de usarmos uma mesma chamada, e porque elas facilitam o uso de uma chamada ao ocultarem os detalhes necessários à sua execução.

Resp.: V (Verdadeira).

03. Por que a velocidade de execução de um processo pode ser reduzida e o tempo de execução das operações de E/S feitas por este processo pode ser aumentado, se o processo estiver executando em um sistema operacional sobre uma máquina virtual? Justifique a sua resposta.

Resp.: A redução da velocidade de execução do processo e o aumento no tempo de execução das operações de E/S ocorre porque uma máquina virtual é uma simulação da máquina real. Em relação à velocidade de execução ser reduzida, isso ocorre porque, como parte da simulação, o monitor de máquina virtual deve alternar o uso do processador do hardware entre os processadores virtuais das diversas máquinas virtuais em execução. Além disso, alguns monitores de máquina virtual podem, ao invés de usar, simular o processador (ou seja, o seu modo de operação) e, com isso, existe um overhead adicional no tempo de execução devido a esta simulação. Já em relação à redução da velocidade das operações de E/S, isso ocorre porque o monitor de máquina virtual também precisa simular cada operação de E/S. Esta simulação mapeia cada operação de E/S, feita em uma máquina virtual sobre um dispositivo virtual, na operação de E/S correspondente sobre um dispositivo físico. Logo, existe também um aumento no tempo de execução de cada operação de E/S, por causa do overhead gerado pelo mapeamento.

04. Como os pipes podem ser usados para garantir o correto funcionamento de dois processos que cooperem para executar uma tarefa em comum, se um dos processos sempre depende dos dados gerados pelo outro processo?

Resp.: Como vimos nas aulas 2 e 4, os pipes funcionam ligando a saída de um processo, no caso o processo que gera os dados, com a entrada de um outro processo, no caso o processo que depende destes dados. Ao ligarmos estes dois processos pelo pipe, quando o processo que gera os dados escrevê-los em sua saída, eles serão copiados para o pipe na ordem em que foram escritos. Além disso, quando o processo que depende dos dados ler a sua entrada, os dados serão lidos do pipe na mesma ordem em que foram escritos, pois o pipe mantém a ordem dos dados. A garantia do correto funcionamento dos processos ocorre porque o processo que gera os dados sempre é bloqueado caso tente escrever dados em sua saída quando o pipe estiver cheio, e o processo que depende dos dados sempre é bloqueado caso tente ler dados da sua entrada quando o pipe estiver vazio. Note que os processos não precisam se preocupar com a exclusão mútua ao acessar o pipe, porque ele é um tipo especial de arquivo do sistema operacional, para o qual exclusão mútua já é automaticamente garantida.

05. Descreva como os semáforos podem ajudar a garantir a correta execução de um conjunto de processos que cooperem para executar uma tarefa em comum.

Resp.: Os semáforos podem ser usados para garantir a exclusão mútua, ou seja, que somente um processo possa acessar, em um dado momento, a sua seção crítica, e a sincronização destes processos, ou seja, que cada processo somente possa executar depois que certas condições, dependentes dos outros processos, tenham sido satisfeitas. A exclusão mútua é garantida através do uso de um semáforo binário, com valor inicial 1, sobre o qual cada processo deve executar a operação P imediatamente antes de entrar em sua seção crítica e executar a operação V imediatamente após sair da sua seção crítica. Para garantir a sincronização, precisamos usar um ou mais semáforos (binários e/ou de contagem), sendo que os objetivos destes semáforos e as suas inicializações dependem das condições que devem ser atendidas para que cada processo possa executar corretamente. Por exemplo, no problema do produtor/consumidor precisamos usar dois semáforos de contagem para a sincronização dos processos. O primeiro, inicializado com o tamanho do buffer, é usado para bloquear o processo produtor caso o buffer esteja cheio. Já o segundo, inicializado com 0, é usado para bloquear o processo consumidor caso o buffer esteja vazio. Note que, neste exemplo, o processo produtor deve executar a operação P sobre o primeiro semáforo antes de colocar um elemento no buffer, e executar a operação V sobre o segundo após colocar este elemento. Já o processo consumidor deve executar a operação P sobre o segundo antes de retirar um elemento do buffer, e executar a operação V sobre o primeiro após remover este elemento.

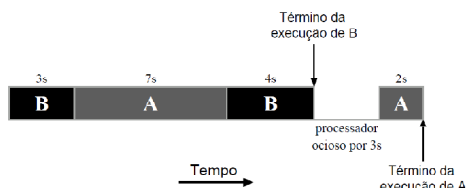
06. Qual é a diferença entre os algoritmos de escalonamento preemptivos e não-preemptivos? Dê um exemplo para cada um dos casos.

Resp.: -No algoritmo de escalonamento preemptivo, cada processo executa no processador ou por um dado intervalo de tempo ou até ser bloqueado esperando por um evento externo (por exemplo, o término de uma operação de E/S). Para poder alternar o processador entre os diversos processos, as interrupções do temporizador são usadas para chamar o algoritmo de escalonamento preemptivo, em intervalos fixos de tempo, para decidir se o processador deve ou não ser cedido a um outro processo. Note que este algoritmo também é chamado quando o processo em execução é bloqueado, para poder escolher um outro processo e evitar que o processador fique ocioso. Agora, no algoritmo de escalonamento não-preemptivo, as interrupções não são usadas e o processo continua a executar até ser bloqueado ou terminar a sua execução. Logo, o algoritmo de escalonamento não-preemptivo somente é chamado quando o processo em execução termina ou bloqueia, para poder alocar o processador para um outro processo.

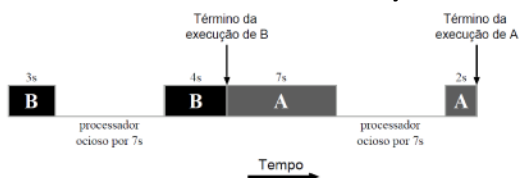
-Na aula 6 vimos diversos exemplos de algoritmos de escalonamento preemptivos. Os dois mais importantes são o algoritmo por round robin e o algoritmo por prioridades, mas também podemos citar, como exemplos, o algoritmo por sorteio e o algoritmo baseado em classes de prioridades. Finalmente, o único algoritmo não-preemptivo que vimos na aula 6 foi o algoritmo do trabalho mais curto primeiro.

AP1 2010-2

01. Descreva como os sistemas operacionais são classificados de acordo (i) com o número de usuários que usam o sistema e (ii) com o número de programas no sistema.



Resp.: (i) Em relação ao número de usuários, os sistemas operacionais podem ser classificados como monousuários ou multiusuários. Nos sistemas monousuários, somente um usuário pode acessar o sistema em um dado intervalo de tempo. Já nos sistemas multiusuários, diversos usuários podem acessar simultaneamente o sistema. (ii) Em relação ao número de programas, o sistema operacional pode ser classificado como monoprogramado ou multiprogramado. Nos sistemas monoprogramados, somente um programa (excluindo o sistema operacional) pode estar residente na memória e em execução em um dado intervalo de tempo. Já nos sistemas multiprogramados, vários programas podem estar simultaneamente residentes na memória e em execução.



02. 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) A tabela de processos é essencial para podermos reiniciar a execução de um processo que foi suspenso.

Resp.: V (Verdadeira).

(b) Dizemos que um processo é filho de um outro processo se ele criou este último.

Resp.: F (Falsa, porque um processo ser filho de um outro processo significa exatamente o contrário, ou seja, que ele foi criado por este outro processo).

(c) Os arquivos são estruturas de alto nível definidas para podermos acessar, de um modo mais transparente, um dispositivo de E/S, como o disco do computador.

Resp.: V (Verdadeira).

(d) Um ponto de montagem é um arquivo especial usado para conectar a entrada de um processo à saída de um outro processo.

Resp.: F (Falsa, porque um ponto de montagem é o diretório do sistema de arquivos com o sistema operacional usado para montar outros sistemas de arquivos).

(e) O procedimento responsável por executar uma chamada ao sistema operacional é chamado de tratador da chamada. Este procedimento precisa executar no modo supervisor porque a chamada pode requerer algum acesso privilegiado disponível somente neste modo.

Resp.: Verdadeira (com exceção dos sistemas operacionais baseados no modelo cliente-servidor porque, nestes sistemas, o tratador da chamada, que é um dos processos servidores, executará no modo usuário).

Obs.: Devido ao enunciado ambíguo, ficou decidido que todos os alunos irão receber 0,5 ponto neste item.

03. Quais são as principais características de um sistema baseado no modelo cliente-servidor em relação (i) ao tipo de núcleo; (ii) às chamadas ao sistema operacional e (iii) ao gerenciamento do hardware.

Resp.: (i) Nos sistemas baseados no modelo cliente-servidor, o núcleo do sistema operacional, chamado de micronúcleo, somente é responsável por duas tarefas: gerenciar a troca de mensagens entre os processos executando no modo usuário; e enviar os comandos às controladoras dos dispositivos do hardware dados em mensagens especiais enviadas pelos processos executando no modo usuário. (ii) Neste modelo, todo o gerenciamento do sistema operacional é feito por processos especiais em execução no modo usuário, chamados de processos servidores. Quando um processo deseja executar uma chamada ao sistema operacional, ele envia uma mensagem ao processo servidor responsável por tratar as chamadas ao sistema operacional. Este processo servidor, após fazer todas as ações necessárias à execução da chamada enviará, também usando uma mensagem, o resultado da chamada ao processo que fez a chamada. (iii) Já em relação ao gerenciamento do hardware, cada processo servidor responsável por um dispositivo do hardware fará todas as tarefas de gerenciamento no modo usuário, com exceção do acesso direto a este dispositivo. Quando o processo servidor precisar enviar comandos à controladora do dispositivo, ele enviará aquela mensagem especial ao micronúcleo que, por sua vez, repassará os comandos contidos na mensagem à controladora do dispositivo.

04. Um aluno de sistemas operacionais afirmou que o diagrama de transições de estado dado a seguir está correto. Se você acha que o aluno está correto basta responder que sim mas, se você acha que ele está errado, então redesenhe corretamente o diagrama.

Resp.: O aluno está incorreto, pois existem duas transições incorretas no diagrama. Na sua resposta, você não precisa dar as explicações dadas a seguir dos erros, mas somente desenhar o diagrama dado na figura a seguir. A primeira transição incorreta é a do estado Bloqueado para o estado Executando. Quando o evento externo que bloqueou um processo finalmente ocorre, o processo precisa passar ao estado Pronto, porque o processo em execução somente deverá ser suspenso pelo escalonador. A segunda transição incorreta é a do estado Pronto para o Bloqueado, e está incorreta porque um processo no estado Pronto jamais pode ser bloqueado devido a não estar em execução no processador. A figura a seguir, similar à dada na Aula 4, mostra o diagrama correto:

05. O que são as seções críticas?

Resp.: Quando um conjunto de processos cooperam para executar uma tarefa em comum eles, em geral, compartilham os diversos recursos necessários à execução daquela tarefa. Se alguns dos recursos

compartilhados forem dedicados então, para que a tarefa seja executada corretamente, cada um deles deverá ser acessado exclusivamente por um dos processos em um dado intervalo de tempo. Neste cenário, a seção crítica de cada processo é a parte do seu código que acessa recursos dedicados. Note que, devido à necessidade de se acessar exclusivamente o recurso, a seção crítica de um processo somente deverá ser executada se nenhuma das seções críticas dos outros processos que acessam este mesmo recurso estiverem sendo executadas.

06. Suponha que o sistema operacional realize escalonamento por round robin com um quantum de 1ms. Suponha também que há quatro processos em execução, todos requerendo o mesmo tempo de processamento. Qual é o maior valor que este tempo pode ter para que todos os processos terminem em 40 quanta?

Resp.: Como vimos na Aula 6, quando o sistema operacional usar o algoritmo por round - robin, cada processo, após executar no processador por no máximo um quantum, somente poderá executar novamente no processador após todos os outros processos não bloqueados também terem executado por no máximo um quantum. Como no cenário dado na questão existem quatro processos com o mesmo tempo de processamento, para que todos eles terminem a sua execução em 40 quanta é necessário que cada um execute por no máximo 10 quanta no processador. Agora, como cada quantum dura 1ms, então o maior valor para o tempo de execução de cada processo será de 10ms.

0	2	4	6	8	10
A	A	B	A	B	B
24	21	19	18	16	13

AP1 2011-1

01. Suponha que dois programas, A e B, tenham sido executados como mostrado na figura a seguir, em um sistema operacional que somente use a multiprogramação para evitar a ociosidade do processador durante operações de E/S. Mostre como os programas seriam executados se o sistema não usasse a multiprogramação, supondo que os tempos das operações de E/S feitas por A e B sejam idênticos.

Resp.: Como podemos ver pela figura, o programa B foi executado por 3s até ser suspenso, significando que ele fez uma operação de E/S após esses 3s, já que a multiprogramação não foi usada para dividir o uso do processador entre os programas em execução. Além disso, como o programa A executou por 7s antes de B voltar a executar por 4s, podemos também concluir que A fez uma operação de E/S após esses 7s de execução, e que o tempo da operação de E/S de B foi no máximo de 7s. Finalmente, como o processador ficou ocioso por 3s após B executar por mais 4s e antes de A voltar a executar por mais 2s, sabemos que o tempo da operação de E/S de A foi de 7s. Assim, concluímos que o tempo da operação de E/S de B também foi de 7s, pois é igual ao tempo da operação de E/S de A. Logo, a execução dos processos em um sistema operacional sem a multiprogramação seria a dada na figura a seguir, na qual podemos notar que o tempo de ociosidade do processador é de 14s.

02. 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) Uma entrada da tabela de processos armazena todas as informações de um processo, incluindo o que está no espaço de endereçamento deste processo.

Resp.: F (Falsa, porque o conteúdo do espaço de endereçamento do processo não é armazenado na sua entrada da tabela de processos).

(b) O uso de sinais permite que os processos em execução no sistema operacional tratem de eventos externos.

Resp.: V (Verdadeira).

(c) Um arquivo deve sempre ser referenciado por um dos seus possíveis nomes de caminho.

Resp.: V (Verdadeira).

(d) O interpretador de comandos sempre cria um processo especial, o qual é responsável por executar todos os programas do usuário.

Resp.: F (Falsa, pois o interpretador de comandos cria um processo diferente para cada programa executado pelo usuário).

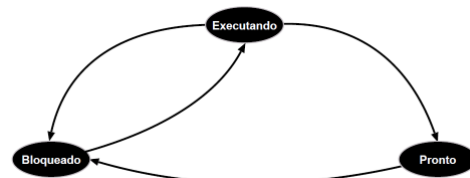
(e) As chamadas ao sistema operacional são caracterizadas por serem de fácil utilização e implementação, pois a sua execução independe da máquina na qual o sistema está executando.

Resp.: F (Falsa, pois as chamadas ao sistema operacional são de difícil implementação justamente por dependerem da máquina na qual o sistema está executando).

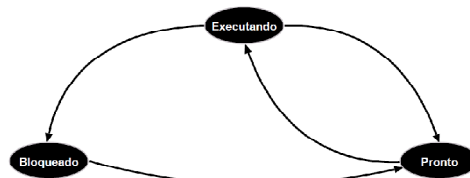
03. Descreva a diferença entre um processo executar em um sistema operacional sobre uma máquina real e sobre uma máquina virtual em relação a: (i) execução das instruções do processador e (ii) execução das operações de E/S.

Resp.: Quando um processo executa em um sistema operacional sobre uma máquina virtual, todo acesso a um componente do hardware acessa uma versão virtual dele. Isto ocorre porque a máquina virtual é uma simulação na qual o processador virtual simula o processador real e os dispositivos virtuais simulam os dispositivos físicos. Então, em relação ao item (i), é o processador virtual que executa as instruções do processo. Naturalmente, o monitor de máquina virtual usa o processador real para executar, alternadamente, a simulação do processador virtual de cada máquina virtual. Já em relação ao item (ii), cada operação é executada sobre um dispositivo virtual e é interceptada pelo monitor de máquina virtual. O monitor de máquina virtual, como parte da simulação da máquina real, então mapeia todas as operações de E/S feitas sobre um dispositivo virtual para o dispositivo físico correspondente, que em geral é compartilhado por todas as máquinas virtuais.

04. Em que circunstâncias é possível dois processos estarem simultaneamente no estado executando? Justifique a sua resposta.



Resp.: Quando o hardware no qual o sistema operacional está executando permite o paralelismo real, e não somente o pseudoparalelismo. Nesse caso, dois ou mais processos podem executar paralelamente no computador, cada um em uma das unidades de processamento do hardware. Essas unidades de processamento são processadores, núcleos (cores) de um mesmo processador, ou threads de um mesmo núcleo.



05. Um aluno de sistemas operacionais afirmou que, para evitar uma condição de corrida, um semáforo binário inicializado com o valor 0 deve ser sempre definido para garantir a exclusão mútua. A afirmação do aluno está correta? Justifique a sua resposta.

Resp.: A afirmação do aluno não está correta. Como inicialmente nenhum dos processos está executando, então o semáforo binário deve ser inicializado com o valor 1, a fim de que o primeiro processo a acessá-lo através de uma operação P consiga executar sua seção crítica. A inicialização com 0 proibiria a entrada de qualquer processo em sua seção crítica, o que configuraria um impasse.

06. Um sistema operacional usa o algoritmo de escalonamento por prioridades, sendo que a prioridade do processo em execução é reduzida em três unidades a cada duas unidades de tempo, e que um processo executa enquanto a sua prioridade é a maior. Mostre como dois processos serão executados, supondo que as suas prioridades iniciais sejam, respectivamente, 24 e 19, e que ambos os processos precisem ser executados por 6 unidades de tempo.

Resp.: Seja A o processo de prioridade 24 e B o processo de prioridade 19. A tabela dada a seguir mostra a ordem de execução de A e B no processador. Nessa tabela, a primeira linha mostra a passagem do tempo de duas em duas unidades. As duas próximas linhas mostram, respectivamente, o processo em execução durante as duas unidades de tempo dadas pela coluna e a prioridade desse processo no início dessas duas unidades. Como podemos ver pela tabela, a sequência de execução será AABABB.