**1.8 Exercícios**

**1.** Aliste quatro areas nas quais o sistema operacional é responsável pelo gerenciamento e pela prestação de serviços. Para cada uma delas, aliste três serviços que costumam ser solicitados por uma chamada ao sistema.

R:

**Processos:**

criação de processo;

destruição de processo;

alteração da prioridade do processo.

**Memória:**

solicitar memória adicional diretamente;

solicitar memória indiretamente;

liberar memória de volta para o SO.

**Dispositivos de E/S:**

abrir um dispositivo ou associá-lo a um processo;

ler dados de um dispositivo;

gravar dados em um dispositivo.

**Sistemas de arquivos:**

abrir um arquivo;

leitura de um arquivo;

gravar em um arquivo.

**2.** Qual é a diferença entre chamada ao sistema e chamada de biblioteca?

R: Quando uma chamada de sistema é feita, o usuário solicita ao sistema operacional (SO) que realize uma operação em seu nome. Uma chamada de biblioteca, por outro lado, não requer normalmente a utilização do sistema para realizar a operação desejada.

**3.** De que modo o ponto de vista do gerenciador de recursos e o do provedor de serviços estão em conflito?

R: Podemos imaginar que o ponto de vista do gerenciador de recursos representa o ponto de vista do proprietário do sistema, que deseja se certificar de que os recusros sejam

usados de forma efetiva. Por outro lado, podemos encarar as coisas do ponto de vista da aplicação. Por essa perspectiva, queremos que o sistema operacional ofereça uma ótima coleção de serviços que facilite as tarefas da aplicação.

**4.** Descreva como os projetos de microkernel são especialmente adequados para sistemas operacionais distribuidos.

R: Na teoria, a idéia é que se alguma máquina parar de funcionar,

alguma outra máquina pode pegar o trabalho dela. Na prática, existem alguns servidores em diferentes máquinas que precisam estar no ar para que o sistema como um todo funcione. Essa confiabilidade e capacidade de se adaptar aos problemas compensam o desempenho pouco abaixo dos sistemas monolíticos, fazendo os microkernels crescerem muito nas aplicações em sistemas operaconais distribuidos.

**5.** Deve-se permitir que um processo de usuário com poucos privilégios execute uma chamada ao sistema que muda o clock do sistema? Por que?

R: Não. Porque esta ação pode acarretar em erros e problemas em outras aplicações e até mesmo no próprio sistema operacional.

**6.** Quais são as vantagens e desvantagens de integrar a interface com o usuário ao sistema operacional?

R: As vantantagens incluem um núcleo de processamento central menos complicado e mais facil de manter, mais eficácia na execução de suas funções e custo de produção reduzido.

A principal desvantagem é a falta de acessibilidade quando ocorrem erros. A dificuldade de reparação desses erros é muito grande e também perigosa, pois mexe diretamente com o código do SO.

**7.** Quais são os aspectos de segurança do gerenciamento de memoria? E do gerenciamento de sistemas de arquivos?

R: Na memória, o SO precisa atender às solicitações de alocação e liberação de memória, assegurando que os processos não interfiram uns com os outros e que o espaço de memória não seja desperdiçado.

No sistema de arquivos, o SO precisa abrir, ler, gravar e fechar um arquivo mantendo a segurança e integridade da informação, além de gerenciar corretamente os metadados, local de gravação, alocação de memória em casos de tranferência e etc.

**8.** Por que são usadas instruções especiais para implementar chamadas ao sistema? Por que não usar intruções normais de chamadas de sub-rotina?

R: Porque se fosseam instruções normais, qualquer aplicação, até mesmo as com poucas permissões e privilégios, conseguiriam acesso ao kernel. Com as instruções especiais o sistema consegue filtrar quem e quando pode acessar seu núcleo.

**9.** SO de máquina virtual e sistemas como Xen costumam ser usados para executar multiplas copias do mesmo SO convidado. Quais são algumas vantagens dessa abordagem em relação à execução de uma única cópia do SO convidado diretamente no hardware?

R: A vantagem é que cada aplicação tem a ilusão de estar sendo executada sozinha na máquina. Na teoria, isso divide o custo de operação e aumenta a velocidade de processamento dos dados de um programa, e também impede que um programa influencie na operação de outro.

**10.** Por que precisamos de um procedimento especial de bootstrapping? Por que não carregar o SO como carregamos qualquer outro programa?

R: Porque o computador não saberia por onde começar, e mesmo que passasemos o SO diretamente, o codigo seria enorme e o carregamento extremamente lento e tedioso.

O procedimento de bootstrapping mostra ao computador onde iniciar o sistema em dois ou mais estágios, carregando arquivos pequenos que contém somente a informação necessária, tornando tudo mais fácil e rápido.

**2.11 Exercícios**

**1.** Por que os projetistas do CTSS projetaram o sistema para permitir que o FMS fosse executado como job normal?

**2.** Qual é a vantagem de implementar um sistema operacional em linguagem de alto nível (como PL/I no Multics) em vez de em linguagem de montagem?

3. Quais as vantagens e desvantagens de implementar bibliotecas usando uma porta de chamada (ou outros mecanismos semelhantes de chamada ao sistema) como no Multics?

**4.** Em todo este capítulo vimos que a tendência de escrever sistemas operacionais mudou de linguagem de montagem para linguagens de alto nível. Vimos também umamudança de sistemas operacionais escritos para um único tipo de máquina para aqueles portados a uma variedade de máquinas diferentes. De que maneiras essas duas tendências estão ligadas?

**5.** No RT-11, por que não se deve executar apenas o monitor de primeiro plano/ segundo plano (FB), mesmo que haja apenas um job a executar, em vezde usar um monitor de job ùnico (SJ) especial?

**6.** No VAX, o 4.3BSD e o VMS usam instruções especiais para trocar o modo do processador no mecanismo de chamadas ao sistema. No 4.3BSD, essas instruções são executadas no código normal do usuário, enquanto no VMS são executadas no kernel. O que se deve levar em conta ao escolher um ou outro esquema?

**7.** Não analisamos as chamadas ao sistema no TinyOS. Foi por esquecimento? Será possível que o TinyOS na verdade não tenha chamadas ao sistema como as estudamos? Como o sistema operacional pode trabalhar sem elas?

**8.** Que serviços de um sistema operacional típico não são fornecidos pelo Xen?

**9.** O UNIX foi criado intencionalmente para ser mais limitado do que o Multics. O Multics teve um início melhor no desenvolvimento, mas foi o UNIX que se tornou mais usado. Mencione algumas das possíveis razões para isso ter acontecido.

**10.** Pesquise e escreva um resumo da história e organização de outro sistema operacional, como o MVS.