



Universidade Federal Rural de Pernambuco  
Sistemas de Informação

Princípios de Software Básico

Lhaíslla Eduarda Cavalcanti Rodrigues da Silva

RECIFE-PE

## **Exercícios referente ao capítulo 1 do livro Sistemas Operacionais modernos Andrew . S Tanenbaum**

1. Quais são as duas principais funções de um sistema operacional?
  - Máquina estendida provendo uma interface simples com o hardware e gerenciamento de recursos.
2. Na Seção 1.4, nove tipos diferentes de sistemas operacionais são descritos. Dê uma lista das aplicações para cada um desses sistemas (uma para cada tipo de sistema operacional).
  - Sistemas operacionais de computadores de grande porte
    - Sistemas operacionais de grande porte , mainframes, utilizadas nos centros de processamento de dados de grandes corporações, como servidores da web para sites de comércio eletrônico e para transações entre empresas. Um exemplo de sistemas operacionais de grande porte é: OS/390.
  - Sistemas operacionais de servidores
    - Sistemas operacionais executados em servidores que são computadores pessoais muito grandes, em estações de trabalho ou mesmo computadores de grande porte,Provedores de acesso a internet utilizam várias máquinas para dar suporte aos cliente, e sites usam servidores para armazenar páginas e lidar com as requisições que chegam. Sistemas operacionais típicos de servidores: Solaris, FreeBSD,Linux, Windows Server 201x.
  - Sistemas operacionais de multiprocessadores
    - Para obter potência computacional é comum conectar múltiplas CPUs a um único sistema.Muitos sistemas operacionais populares como: Linux e Windows são executados em multiprocessadores.
  - Sistemas operacionais de computadores pessoais
    - Todos computadores modernos dão suporte a multiprogramação. Exemplos: Linux,FreeBSD,windows 7,windows 8 e o OS x da apple.
  - Sistemas operacionais de computadores portáteis

- É um computador pequeno que pode ser segurado na mão durante a operação. Exemplos: Smartphones e tablets, como android do google e iOS da Apple.
- Sistemas operacionais embarcados
  - Dispositivos que não aceitam software instalados pelo usuário, como: forno de micro-ondas, aparelhos de televisão, carros, aparelhos de DVD, telefones tradicionais, e MP3 players.
- Sistemas operacionais de nós sensores (sensor-node)
  - Redes de nós sensores minúsculos que se comunicam entre si e com uma estação base utilizando comunicação sem fio. Exemplo: São usados para proteger os perímetros de prédios, guardar fronteiras nacionais, detectar incêndios em florestas, medir temperatura e a precipitação para a previsão do tempo, colher informação sobre a movimentação de inimigos nos campos de batalha e muito mais.
- Sistemas operacionais de tempo real
  - É o sistema de tempo real. Esses sistemas são caracterizados por ter o tempo como um parâmetro-chave. Por exemplo, em sistemas de controle de processo industrial, computadores em tempo real têm de coletar dados a respeito do processo de produção e usá-los para controlar máquinas na fábrica. Muitas vezes há prazos rígidos a serem cumpridos. Por exemplo, se um carro está seguindo pela linha de montagem, determinadas ações têm de ocorrer em dados instantes
- Sistemas operacionais de cartões inteligentes (smartcard) .
  - Menores sistemas operacionais, executados em cartões inteligentes. Exemplo: Alguns cartões inteligentes são orientados a Java. Isso significa que o ROM no cartão inteligente contém um interpretador para a Java Virtual Machine (JVM — Máquina virtual Java). Os aplicativos pequenos (applets) Java são baixados para o cartão e são interpretados pelo JVM. Alguns desses cartões podem lidar com múltiplos applets Java ao mesmo tempo, levando à multiprogramação e à necessidade de escaloná-los.

3. Qual é a diferença entre sistemas de compartilhamento de tempo e de multiprogramação?

- Os sistemas de compartilhamento de tempo, também chamados de time-sharing, possibilitam que vários programas sejam executados através de uma divisão de tempo do processador em intervalos curtos. Os sistemas de multiprogramação ou multitarefa são uma evolução dos programas de monoprogramação. Eles possibilitam, por exemplo, que enquanto um programa esteja em operação de leitura outros possam ser executados.
4. Para usar a memória de cache, a memória principal é dividida em linhas de cache, em geral de 32 a 64 bytes de comprimento. Uma linha inteira é capturada em cache de uma só vez. Qual é a vantagem de fazer isso com uma linha inteira em vez de um único byte ou palavra de cada vez?
- As linhas de cache mais utilizadas são mantidas em uma cache de alta velocidade localizada dentro ou muito próximo da CPU. Quando o programa precisa ler uma palavra de memória, o hardware de cache confere se a linha requisitada está na cache. Se ela estiver presente na cache (cache hit), a requisição é atendida e nenhuma requisição de memória é feita para a memória principal sobre o barramento. Se a linha requisitada estiver ausente da cache (cache miss), uma requisição adicional é feita à memória, com uma penalidade de tempo substancial. A memória da cache é limitada em tamanho por causa do alto custo. Sempre que um recurso pode ser dividido em partes, algumas das quais são usadas com muito mais frequência que as outras, o caching é muitas vezes utilizado para melhorar o desempenho.
5. Nos primeiros computadores, cada byte de dados lido ou escrito era executado pela CPU (isto é, não havia DMA). Quais implicações isso tem para a multiprogramação?
- Pode ocasionar perda de desempenho ao sistema. Com a multiprogramação vários processos estão sendo executados sequencialmente com alternância rápida e constante, se todas operações de dados envolverem E/S, sempre pelo menos um dos processos estiver executando E/S todos os outros processos tem de esperar pelo término da operação.

6. Instruções relacionadas ao acesso a dispositivos de E/S são tipicamente instruções privilegiadas, isto é, podem ser executadas em modo núcleo, mas não em modo usuário. Dê uma razão de por que essas instruções são privilegiadas.
  - Instruções privilegiadas devem ser executadas pelo sistema operacional ou com supervisão do administrador, o que impede a ocorrência de problemas de segurança e integridade do sistema. Instruções não-privilegiadas não oferecem risco ao sistema., quando o processador trabalha no modo usuário, uma aplicação só pode executar instruções não-privilegiadas, tendo acesso a um número reduzido de instruções, enquanto no modo kernel ou supervisor a aplicação pode ter acesso ao conjunto total de instruções do processador.
7. A ideia de família de computadores foi introduzida na década de 1960 com os computadores de grande porte System/360 da IBM. Essa ideia está ultrapassada ou ainda é válida?
  - A ideia ainda vive, pois atualmente a maioria dos fabricantes lançam suas máquinas em famílias, incrementando novas funcionalidades aos modelos.
8. Uma razão para a adoção inicialmente lenta das GUIs era o custo do hardware necessário para dar suporte a elas. Quanta RAM de vídeo é necessária para dar suporte a uma tela de texto monocromo de 25 linhas 80 colunas de caracteres? E para um bitmap colorido de 24 bits de 1.200 900 pixels? Qual era o custo desta RAM em preços de 1980 (US\$ 5/KB)? Quanto é agora?
  - Uma tela de texto monocromático de  $25 \times 80$  caracteres requer um buffer de 2000 bytes, supondo que cada carácter ocupa 8 bits, temos que para o primeiro se faz necessário  $25 \times 80 \times 8 = 16\text{KB}$ . Já para a segunda situação temos  $1024 \times 768 \times 24 = 18432\text{KB}$ . No ano de 1980 o custo da primeira seria de 80 dólares e da segunda de 92162 dólares. Hoje em dia a primeira situação, preço de 0,000026 dólares/KB a primeira solução custa 0,00042 e a segunda 0,48 dólares.
9. Há várias metas de projeto na construção de um sistema operacional, por exemplo, utilização de recursos, oportunidade, robustez e assim por diante. Dê um exemplo de duas metas de projeto que podem contradizer uma à outra

- Podemos tomar como exemplo, de conflito entre metas, aquelas que exigem agilidade (tempo/prazo de entrega) e perfeição (qualidade), sem dispor de recursos financeiros e humanos para atender às demandas.
10. Qual é a diferença entre modo núcleo e modo usuário? Explique como ter dois modos distintos ajuda no projeto de um sistema operacional.
- No modo kernel(núcleo) é possível o acesso irrestrito ao conjunto de instruções da máquina. Um processo deixa de executar em modo usuário e passa a executar em modo kernel quando executa uma chamada de sistema. Ter estes dois modos nos ajuda a garantir uma segurança maior do sistema operacional, pois somente se ele permitir esse modo kernel algum programa pode fazer alguma modificação que traria risco ao sistema.
11. Um disco de 255 GB tem 65.536 cilindros com 255 setores por faixa e 512 bytes por setor. Quantos pratos e cabeças esse disco tem? Presumindo um tempo de busca de cilindro médio de 11 ms, atraso rotacional médio de 7 ms e taxa de leitura de 100 MB/s, calcule o tempo médio que será necessário para ler 400 KB de um setor.
- $7\text{ms} \times 400 \text{ GB}(\text{taxa de transferência} * \text{o espaço})/100(\text{dividido pela taxa de leitura}) = 2800/100 = 28 \text{ ms}$
12. Quais das instruções a seguir devem ser deixadas somente em modo núcleo?
- (a) Desabilitar todas as interrupções.
  - (b) Ler o relógio da hora do dia.
  - (c) Configurar o relógio da hora do dia.
  - (d) Mudar o mapa de memória
- As alternativas (a), (c) e (d), pois só devem ser autorizadas no modo núcleo.
13. Considere um sistema que tem duas CPUs, cada uma tendo duas threads (hiper-threading). Suponha que três programas, P0, P1 e P2, sejam iniciados com tempos de execução de 5, 10 e 20 ms, respectivamente. Quanto tempo levará para completar a execução desses programas? Presuma que todos os três programas sejam 100% ligados à CPU, não bloqueiem durante a execução e não mudem de CPUs uma vez escolhidos.
- $P0 \ 0,75\text{ms} + P1 \ 1,5\text{ms} + P2 \ 3\text{ms} = 1,75 \text{ ms}/2 \text{ CPUs} = 0,875 \text{ ms}$ . Sendo a taxa de eficácia do hyperthreading de 15 %.

14. Um computador tem um pipeline com quatro estágios. Cada estágio leva um tempo para fazer seu trabalho, a saber, 1 ns. Quantas instruções por segundo essa máquina consegue executar?
- 4ns para cada instrução ser realizada e m 1.250000000 segundo s.
15. Considere um sistema de computador que tem uma memória de cache, memória principal (RAM) e disco, e um sistema operacional que usa memória virtual. É necessário 1 ns para acessar uma palavra da cache, 10 ns para acessar uma palavra da RAM e 10 ms para acessar uma palavra do disco. Se o índice de acerto da cache é 95% e o índice de acerto da memória principal (após um erro de cache) 99%, qual é o tempo médio para acessar uma palavra?
- considerando uma média entre o cache de memória RAM e disco a média da velocidade devido todas essas memórias se interagem fica em torno de 1 ns.
16. Quando um programa de usuário faz uma chamada de sistema para ler ou escrever um arquivo de disco, ele fornece uma indicação de qual arquivo ele quer, um ponteiro para o buffer de dados e o contador. O controle é então transferido para o sistema operacional, que chama o driver apropriado. Suponha que o driver começa o disco e termina quando ocorre uma interrupção. No caso da leitura do disco, obviamente quem chamou terá de ser bloqueado (pois não há dados para ele). E quanto a escrever para o disco? Quem chamou precisa ser bloqueado esperando o término da transferência de disco?
- Caso se queira garantir que a escrita foi feita antes de continuar a execução do processo, sim, quem chama não precisa ser bloqueado.
17. O que é uma instrução? Explique o uso em sistemas operacionais
- Instrução é uma expressão ou código que especifica uma operação a ser executada pelo microprocessador e, se necessário, identificar os operandos. Em sistemas operacionais está presente no ciclo básico de toda CPU, onde é buscado a primeira instrução da memória, decodificada para determinar o seu tipo e operandos, executada, e então buscar, decodificar e executar as instruções subsequentes. O ciclo é repetido até o programa terminar. É dessa maneira que os programas são executados. Cada CPU tem um conjunto específico de instruções que ela consegue executar.

18. Por que a tabela de processos é necessária em um sistema de compartilhamento de tempo? Ela também é necessária em sistemas de computadores pessoais executando UNIX ou Windows com um único usuário?
- A tabela de processos é necessária para manter o estado dos processos que estão atualmente suspensos (em estado de espera ou bloqueados). No caso de SO's com apenas um processo, esta tabela não é necessária, uma vez que o único processo nunca entra em estado suspenso.
19. Existe alguma razão para que você quisesse montar um sistema de arquivos em um diretório não vazio? Se a resposta for sim, por quê?
- Sim, em casos usados para impossibilitar o uso de arquivos que estavam no diretório antes do novo sistema ser montado.
20. Para cada uma das chamadas de sistema a seguir, dê uma condição que a faça falhar: fork, exec e unlink.
- fork - falta de memória para criação do processo filho na memória.
  - exec - número de parâmetros inválidos.
  - unlink de arquivo inexistente.
21. Qual tipo de multiplexação (tempo, espaço ou ambos) pode ser usado para compartilhar os seguintes recursos: CPU, memória, disco, placa de rede, impressora, teclado e monitor?
- Quando um recurso é multiplexado no tempo, diferentes programas ou usuários se revezam usando-o. Primeiro, um deles usa o recurso, então outro e assim por diante. Por exemplo, com apenas uma CPU e múltiplos programas querendo ser executados nela, o sistema operacional primeiro aloca a CPU para um programa, então, após ele ter sido executado por tempo suficiente, outro programa passa a fazer uso da CPU, então outro, e finalmente o primeiro de novo. Determinar como o recurso é multiplexado no tempo — quem vai em seguida e por quanto tempo — é a tarefa do sistema operacional. Outro exemplo da multiplexação no tempo é o compartilhamento da impressora. O outro tipo é a multiplexação de espaço. Em vez de os clientes se revezarem, cada um tem direito a uma parte do recurso. A memória principal é normalmente dividida entre vários programas sendo executados, de modo que



cada um pode ser residente ao mesmo tempo (por exemplo, a fim de se revezar usando a CPU). Presumindo que há memória suficiente para manter múltiplos programas, é mais eficiente manter vários programas na memória ao mesmo tempo do que dar a um deles toda ela, especialmente se o programa precisa apenas de uma pequena fração do total. Outro recurso que é multiplexado no espaço é o disco. Em muitos sistemas um único disco pode conter arquivos de muitos usuários ao mesmo tempo. Alocar espaço de disco e controlar quem está usando quais blocos do disco é uma tarefa típica do sistema operacional. Desta forma ambos os tipos de multiplexação são usados para compartilhar os recursos.

22. A chamada `count = write(fd, buffer, nbytes);` pode retornar qualquer valor em `count` fora `nbytes`? Se a resposta for sim, por quê?
- Sim. Caso não seja possível executar a chamada `write` será retornado -1, valor referente a variável global **erro** que é retornada sempre que há erro em alguma chamada. Além disso, a chamada `write` irá retornar um menor do que `nbytes` se caso for inserido o caractere fim de arquivo (EOF) no final da escrita.

23. Um arquivo cujo descritor é `fd` contém a sequência de bytes: 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5. As chamadas de sistema a seguir são feitas:

```
lseek(fd, 3, SEEK_SET);  
read(fd, &buffer, 4);
```

onde a chamada `lseek` faz uma busca para o byte 3 do arquivo. O que o `buffer` contém após a leitura ter sido feita?

- Como o `lseek` está no 3, o `read` faz a leitura de 4 bytes a frente, logo, ele contém os bytes 1, 5, 9 e 2.
24. Suponha que um arquivo de 10 MB esteja armazenado em um disco na mesma faixa (faixa 50) em setores consecutivos. O braço do disco está atualmente situado sobre o número da faixa 100. Quanto tempo ele levará para retirar esse arquivo do disco?

Presuma que ele leve em torno de 1 ms para mover o braço de um cilindro para o próximo e em torno de 5 ms para o setor onde o início do arquivo está armazenado para girar sob a cabeça. Também, presume que a leitura ocorra a uma taxa de 200 MB/s.

- Tempo de retirar arquivo = tempo para mover o braço na faixa + tempo para o primeiro setor girar sob a cabeça + ler 10mb tempo de retirar o arquivo =  $(1 \times 50) + 5 + (10/200 \times 1000)$  tempo de retirar o arquivo = 105ms

25. Qual é a diferença essencial entre um arquivo especial de bloco e um arquivo especial de caractere?

- Um arquivo especial de bloco é organizado em blocos aleatoriamente numerados e podem ser individualmente endereçados. Caso você deseje acessar o bloco 98, não será necessário navegar do bloco 0 ao 97. São bastante usados em discos. Um arquivo especial de caractere é utilizado para modelar impressoras e outros dispositivos que aceitam ou enviam fluxo de caracteres. A diferença essencial entre eles é o armazenamento em buffer. Enquanto os arquivos especiais de caractere fornecem um acesso direto ao dispositivo de hardware, os arquivos especiais de bloco fornecem acesso em memória temporária.

26. No exemplo dado na Figura 1.17, a rotina de biblioteca é chamada read e a chamada de sistema em si é chamada read. É fundamental que ambas tenham o mesmo nome? Se não, qual é a mais importante?

- Não necessariamente. A nomenclatura é definida para melhorar a compreensão do usuário sobre o que está acontecendo. A rotina de biblioteca provê apenas uma interface entre a linguagem e a funcionalidade do SO. O mais importante, no meu entendimento, é a chamada da biblioteca em si uma vez que por trás dos panos ela tem o poder de “traduzir” esta chamada para uma chamada de sistema compatível um sistema UNIX ou uma rotina da API Win 32 (já que o Windows não fornece acesso às suas chamadas de sistema).

27. Sistemas operacionais modernos desacoplam o espaço de endereçamento do processo da memória física da máquina. Liste duas vantagens desse projeto.

- Um programa é carregado em locais diferentes da memória da máquina e em diferentes execuções. O tamanho do programa pode exceder o tamanho da memória da máquina.
28. Para um programador, uma chamada de sistema parece com qualquer outra chamada para uma rotina de biblioteca. É importante que um programador saiba quais rotinas de biblioteca resultam em chamadas de sistema? Em quais circunstâncias e por quê?
- Isso dependerá do sistema que ele estiver construindo. Caso ele necessite de um sistema otimizado, com um custo computacional mais baixo, pode ser do interesse dele entender o que cada rotina faz e qual a chamada de rotina resultante (caso haja uma).
29. A Figura 1.23 mostra que uma série de chamadas de sistema UNIX não possuem equivalentes na API Win32. Para cada uma das chamadas listadas como não tendo um equivalente Win32, quais são as consequências para um programador de converter um programa UNIX para ser executado sob o Windows?
- Incompatibilidade do sistema por falta do reconhecimento das instruções de execução do programa com o interpretador de comandos e a comunicação entre processos.
30. Um sistema operacional portátil é um sistema que pode ser levado de uma arquitetura de sistema para outra sem nenhuma modificação. Explique por que é impraticável construir um sistema operacional que seja completamente portátil. Descreva duas camadas de alto nível que você terá ao projetar um sistema operacional que seja altamente portátil.
- Um sistema operacional é um conjunto de programas especialmente feitos para a execução de várias tarefas, entre as quais servir de intermediário entre o utilizador e o computador. Um sistema operacional, tem também como função, gerir todos os periféricos de um computador. Nas camadas de alto nível é necessário projetar a interface para o usuário, além do gerenciamento dos recursos.
31. Explique como a separação da política e mecanismo ajuda na construção de sistemas operacionais baseados em micronúcleos

- Arquivos em UNIX são protegidos designando-se a cada arquivo um código de proteção binário de 9 bits. O código de proteção consiste de três campos de 3 bits, um para o proprietário, um para os outros membros do grupo do proprietário (usuários são divididos em grupos pelo administrador do sistema) e um para todos os demais usuários. Cada campo tem um bit de permissão de leitura, um bit de permissão de escrita e um bit de permissão de execução. Esses 3 bits são conhecidos como os bits rwx. Por exemplo, o código de proteção rwxr-x--x significa que o proprietário pode ler (read), escrever (write), ou executar (execute) o arquivo, que outros membros do grupo podem ler ou executar (mas não escrever) o arquivo e que todos os demais podem executar (mas não ler ou escrever) o arquivo. Para um diretório, x indica permissão de busca. Um traço significa que a permissão correspondente está ausente.

32. Máquinas virtuais tornaram-se muito populares por uma série de razões. Não obstante, elas têm alguns problemas. Cite um.

- Sobrecarga de tarefas: Caso ligue várias máquinas virtuais ao mesmo tempo, pode ocorrer uma sobrecarga de tarefas e todas podem parar de funcionar. Por isso o ideal é instalar e executar um número de máquinas que o computador suporte.

33. A seguir algumas questões para praticar conversões de unidades

- Quantos segundos há em um nanoséculo?

- $3,154 \times 10^7$  segundos

- Micrômetros são muitas vezes chamados de microns. Qual o comprimento de um megamícron?

- $10^6$  metros

- Quantos bytes existem em uma memória de 1 PB?
  - $10^{15}$  **bytes** = 1000terabytes.
- A massa da Terra é 6.000 yottagramas. Quanto é isso em quilogramas?
  - $6 \times 10^{24}$  quilogramas