

Aluno : Jackson Ricardo dos Santos da Silva

Turma: 20191.3.01404.1M

Respostas do capítulo 1 - Conceitos básicos

1. Quais os dois principais objetivos dos sistemas operacionais?

R: Cria um camada de abstração entre a aplicação e o hardware e gerencia os recursos de forma eficiente.

2. Por que a abstração de recursos é importante para os desenvolvedores de aplicações? Ela tem utilidade para os desenvolvedores do próprio sistema operacional?

R: Porque a abstração pode prover interfaces de acesso aos dispositivos mais simples de usar que as interfaces de baixo, nível; torna os aplicativos independentes do hardware e define interfaces de acesso homogêneas para dispositivos com tecnologias distintas. A abstração é útil, pois minimiza os conflitos no uso de hardware, por que o acesso se faz através de interfaces simples.

3. A gerência de atividades permite compartilhar o processador, executando mais de uma aplicação ao mesmo tempo. Identifique as principais vantagens trazidas por essa funcionalidade e os desafios a resolver para implementá-la.

R: A principal vantagem é poder compartilhar o uso do processador, realizando várias tarefas simultaneamente sem o surgimento de conflitos no uso do hardware. Os desafios são a distribuição do processador entre os aplicativos, fazer com que o uso da impressora seja exclusivo a um único aplicativo e impedir que um usuário monopolize todos os recursos do sistema.

4. O que caracteriza um sistema operacional de tempo real? Quais as duas classificações de sistemas operacionais de tempo real e suas diferenças?

R: Um sistema operacional de tempo real deve ter um comportamento temporal previsível, ou seja, seu tempo de resposta deve ser conhecido no melhor e no pior caso.

- **Soft real time systems: nos quais a perda de prazos implica na degradação do serviço prestado.**
- **Hard real time systems: onde a perda de prazos pelo sistema pode perturbar o objeto controlado, com graves consequências humanas, econômicas e ambientais.**

5. O que diferencia o *núcleo* do restante do sistema operacional?

R: O núcleo é o coração do sistema operacional. É responsável pela gerencia de recursos e implementar as principais abstrações utilizadas pelos aplicativos.

6. Seria possível construir um sistema operacional seguro usando um processador que não tenha níveis de privilégio? Por quê?

R: Não, porque uma aplicação poderá interferir nas áreas de memória de outras aplicações ou do núcleo. Sem os privilégios uma aplicação pode acessar a placa de rede para enviar ou receber dados.

7. O processador Pentium possui dois bits para definir o nível de privilégio, resultando em 4 níveis distintos. A maioria dos sistemas operacionais para esse processador usa somente os níveis extremos (0 e 3, ou 00_2 e 11_2). Haveria alguma utilidade para os níveis intermediários?

R: O processador Pentium possui dois bits para definir o nível de privilégio, resultando em 4 níveis distintos. A maioria dos sistemas operacionais para esse processador usa somente os níveis extremos (0 e 3, ou 00_2 e 11_2).

8. Quais as diferenças entre *interrupções*, *exceções* e *traps*?

R: Interrupção: os circuitos do processador suspendem seu fluxo de execução corrente e desviam para um endereço predefinido, onde se encontra uma rotina de tratamento de interrupção.

Exceção: eventos gerados pelo próprio processador que podem ocasionar o desvio da execução usando o mesmo mecanismo das interrupções.

Traps: instrução especial que permite acionar o mecanismo de interrupção de forma intencional, sem depender de eventos externos ou internos.

9. Quais as implicações de mascarar interrupções? O que pode ocorrer se o processador ignorar interrupções por muito tempo? O que poderia ser feito para evitar o mascaramento de interrupções?

R: O processador perde tempo para varrer todos os dispositivos do sistema para verificar se há eventos a serem tratados ou não.

10. O comando em linguagem C `fopen` é uma chamada de sistema ou uma função de biblioteca? Porquê?

R: É uma função de uma biblioteca. Porque o comando “fopen” é usada para abertura de arquivos na Biblioteca Padrão Entrada/Saída.

11. Monte uma tabela com os benefícios e deficiências mais significativos das principais arquiteturas de sistemas operacionais.

Arquiteturas	Benefícios	Deficiências
Sistemas monolíticos	Qualquer componente do núcleo pode acessar os demais componentes	Se um componente do núcleo perder o controle devido algum erro, esse problema alastrar rapidamente por todo o núcleo
Sistemas em camadas	Estruturar um sistema operacional em faz o uso da noção em camadas	O empilhamento de várias camadas de software faz com que cada pedido de uma aplicação demore mais tempo até chegar ao periférico.
Sistemas micronúcleo	Robustez e flexibilidade: caso um subsistema tenha problemas, os mecanismos de proteção de memória irão confiná-lo.	Os processos não podem se comunicar diretamente, devido às restrições impostas pelos mecanismos de proteção do hardware.
Máquinas virtuais	Construir uma camada de software que ofereça aos demais componentes serviços com outra interface.	É o custo adicional de execução dos processos na máquina virtual com a real.

12. Relacione as afirmações aos respectivos tipos de sistemas operacionais: distribuído (D), multi-usuário (M), desktop (K), servidor (S), embarcado (E) ou de tempo-real (T):

[T] Deve ter um comportamento temporal previsível, com prazos de resposta claramente definidos.

[S] Sistema operacional usado por uma empresa para executar seu banco de dados corporativo.

[E] São tipicamente usados em telefones celulares e sistemas eletrônicos dedicados.

[D] Neste tipo de sistema, a localização física dos recursos do sistema computacional é transparente para os usuários.

[M] Todos os recursos do sistema têm proprietários e existem regras controlando o acesso aos mesmos pelos usuários.

[E] A gerência de energia é muito importante neste tipo de sistema.

[K] Sistema que prioriza a gerência da interface gráfica e a interação com o usuário.

[S] Construído para gerenciar de forma eficiente grandes volumes de recursos.

[D] O MacOS X é um exemplo típico deste tipo de sistema.

[E] São sistemas operacionais compactos, construídos para executar aplicações específicas sobre plataformas com poucos recursos.

13. A operação em modo usuário permite ao processador executar somente parte das instruções disponíveis em seu conjunto de instruções. Quais das seguintes operações não deveriam ser permitidas em nível usuário? Por quê?

- (a) Ler uma porta de entrada /saída
- (b) Efetuar uma divisão inteira
- (c) Escrever um valor em uma posição de memória
- (d) Ajustar o valor do relógio do hardware
- (e) Ler o valor dos registradores do processador
- (f) Mascara uma ou mais interrupções

R: (c) Escrever um valor em uma posição de memória não deveria ser permitida no nível usuário. Porque instruções perigosas como HALT(parar o processador) e RESET(reiniciar o processador) são proibidas neste nível.

14. Considerando um processo em um sistema operacional com proteção de memória entre o núcleo e as aplicações, indique quais das seguintes ações do processo teriam de ser realizadas através de chamadas de sistema, justificando suas respostas:

- (a) Ler o relógio de tempo real do hardware.
- (b) Enviar um pacote através da rede.
- (c) Calcular uma exponenciação.
- (d) Preencher uma área de memória do processo com zeros.
- (e) Remover um arquivo do disco.

R: Os processo que teriam de ser realizadas através de chamadas de sistemas seriam: (a) Ler o relógio de tempo real do hardware e (d) Preencher uma área de memória do processo com zeros. Porque chamadas de sistema são todas as operações envolvendo o acesso a recursos de baixo nível (periféricos, arquivos, alocação de memória, etc.) ou abstrações lógicas (criação e finalização de tarefas, operadores de sincronização e comunicação, etc.).

15. Coloque na ordem correta as ações abaixo, que ocorrem durante a execução da função `printf("Hello world")` por um processo (observe que nem todas as ações indicadas fazem parte da sequência).

- [5] A rotina de tratamento da interrupção de software é ativada dentro do núcleo.
- [9] A função `printf` finaliza sua execução e devolve o controle ao código do processo.
- [1] A função de biblioteca `printf` recebe e processa os parâmetros de entrada (a string "Hello world").
- [4] A função de biblioteca `printf` prepara os registradores para solicitar a chamada de sistema `write()`
- [8] O disco rígido gera uma interrupção indicando a conclusão da operação.
- [2] O escalonador escolhe o processo mais prioritário para execução.
- [7] Uma interrupção de software é acionada.
- [3] O processo chama a função `printf` da biblioteca C.
- [6] A operação de escrita no terminal é efetuada ou agendada pela rotina de tratamento da interrupção.
- [10] O controle volta para a função `printf` em modo usuário.

16. Considere as afirmações a seguir, relativas aos diversos tipos de sistemas operacionais:

- I. Em um sistema operacional de **tempo real**, a rapidez de resposta é menos importante que a previsibilidade do tempo de resposta.
- II. Um sistema operacional **multi-usuários** associa um proprietário a cada recurso do sistema e gerencia as permissões de acesso a esses recursos.
- III. **Nos sistemas operacionais de rede a localização dos recursos é transparente para os usuários.**
- IV. **Um sistema operacional de tempo real deve priorizar as tarefas que interagem com o usuário.**
- V. Um sistema operacional **embarcado** é projetado para operar em hardware com poucos recursos.

Indique a alternativa correta:

- (a) As afirmações II e III estão corretas.
- (b) Apenas a afirmação V está correta.
- (c) As afirmações III e IV estão erradas.**
- (d) As afirmações III, IV e V estão erradas.
- (e) Todas as afirmações estão erradas.

R: A alternativa III está errada, porque é uma característica de sistemas

distribuídos e não de rede. E a alternativa também está errada, porque é uma característica de sistemas desktop e não de tempo real.

17. Considere as afirmações a seguir, relativas às diversas arquiteturas de sistemas operacionais:

- I. Uma máquina virtual de sistema é contruída para suportar uma aplicação escrita em uma linguagem de programação específica, como Java.
- II. Um hipervisor convidado executa sobre um sistema operacional hospedeiro.
- III. Em um sistema operacional micro-núcleo, os diversos componentes do sistema são construídos como módulos interconectados executando dentro do núcleo.
- IV. Núcleos monolíticos são muito utilizados devido à sua robustez e facilidade de manutenção.
- V. Em um sistema operacional micro-núcleo, as chamadas de sistema são implementadas através de trocas de mensagens.

Indique a alternativa correta:

- (a) Todas as afirmações estão erradas.
- (b) As afirmações II e III estão corretas.
- (c) As afirmações II e IV estão erradas.
- (d) Apenas a afirmação V está correta.
- (e) As afirmações II e V estão corretas.

Justifique as afirmações julgadas erradas (Assim: *VII está errada porque ...*):

R: I está errada, porque uma máquina virtual de sistema é construída para suportar sistemas operacionais convidados completos. Já a III está errada, porque isso ocorre em sistemas monolíticos e não micronúcleos. E a IV está errada, porque sistemas monolíticos não tem uma manutenção fácil e sim complexa.

18. O utilitário strace do UNIX permite observar a sequência de chamadas de sistema efetuadas por uma aplicação. Em um terminal UNIX, execute strace date para descobrir quais os arquivos abertos pela execução do utilitário date (que indica a data e hora correntes). Por que o utilitário date precisa fazer chamadas de sistema?

R: Porque o strace é uma ferramenta que monitora as chamadas de sistema (system calls) e os sinais recebidos pela aplicação. A maneira mais comum de executá-la é passando a aplicação a ser monitorada como parâmetro.

19. O utilitário ltrace do UNIX permite observar a sequência de chamadas de biblioteca efetuadas por uma aplicação. Em um terminal UNIX, execute ltrace date para descobrir as funções de biblioteca chamadas pela execução do utilitário date (que indica a data e hora correntes). Pode ser observada alguma relação entre as chamadas de biblioteca e as chamadas de sistema observadas no ítem anterior?

R: O ltrace tem as mesmas características do strace, mas ao invés de monitorar as chamadas do sistema, ele monitora as chamadas às funções das bibliotecas carregadas dinamicamente.