

1. Quais os dois principais objetivos dos sistemas operacionais?

Abstração e gerência de recursos.

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

É importante porque tira a necessidade de toda aplicação ter "conhecimento" de linguagem de baixo nível, ou de vários tipos de hardware ou, ainda, comunicação com tecnologias distintas, tudo isso é papel do sistema operacional. Logo ela, a abstração de recursos, tem mais utilidade aos desenvolvedores de sistemas operacionais, pois através dela ele consegue gerenciar os recursos do S.O. resolvendo conflitos de softwares, etc.

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.

Como já citado no enunciado, uma das principais vantagens é a simultaneidade no uso de aplicações sem causar conflitos no hardware, com isso o maior desafio em implementá-la é o uso "democrático" do poder de processamento.

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

Caracteriza-se por um comportamento temporal previsível, onde seu tempo de resposta é conhecido no melhor e pior caso de operação. Suas duas classificações são: Soft real-time systems e Hard real-time systems. A principal diferença entre eles é que, no caso de perda de prazo de uma determinada operação as consequências serão na degradação do serviço prestado (Soft real-time) ou em graves consequências humanas, ambientais e econômicas (Hard real-time).

5. O que diferencia o núcleo do restante do sistema operacional (SO)?

Enquanto maior parte do SO trabalha de forma "abstrata", o núcleo, é a parte que está em contato direto com o hardware da máquina.

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

Não, pois causaria lentidão e conflitos nos processos levando assim a desestruturação de todo o SO

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 002 e 112). Haveria alguma utilidade para os níveis intermediários?

Certamente, toda vez que houver processamento entre níveis.

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

O primeiro, pode-se dizer que são eventos causados por dispositivos externos ao processador, o segundo é causado pelo próprio processador e o último é gerado por softwares.

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?

Mascarar interrupções permite softwares ignorar toda interrupção de hardware externo ao mesmo tempo que é definido, esta máscara pode oferecer acesso mais rápido do que acessar o registro da máscara de interrupção (IMR) em um PIC, ou desabilitar as interrupções do próprio dispositivo. Em alguns casos, como a da arquitetura x86, desabilitar e habilitar as interrupções no próprio processador, como uma barreira de memória, ele pode ser de fato mais lento e em casos mais graves pode haver o que é denominado de tempestade de interrupção (interrupt storm). Isso tudo pode ser evitado adicionando ao circuito um controlador de interrupção.

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

É uma função de biblioteca pois ela se utiliza de primitivas (chamadas de sistema) como por exemplo a open().

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

Arquitetura	Benefícios	Deficiências
Sistemas Monolíticos	Desempenho	Mal funcionamento de uma aplicação do núcleo pode levar a travamento e instabilidade; Manutenção complexa; Evolução complexa.
Sistemas em Camadas	Separação de código; Mudança de implementação entre camadas se afetá-las; Camadas trabalham com diferentes versões de outras camadas;	Aumento do número de classes no sistema.
Sistemas Micronúcleo	Robustez e flexibilidade.	Custos associados as trocas de mensagem entre os componentes do sistema bastante elevados reduz desempenho.
Maquinas Virtuais	Aperfeiçoamentos e testes de novos Sistemas Operacionais; Executar diferentes SO's simultaneamente em um mesmo hardware; Simulações de configurações e situações do mundo real em SO's; Diminuir custos com hardware.	Custo adicional de execução dos processos em comparação com a real (isso não ocorre com mainframes.)

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):
- [D] Deve ter um comportamento temporal previsível, com prazos de resposta claramente definidos.
 - [M] Sistema operacional usado por uma empresa para executar seu banco de dados corporativo.
 - [E] São tipicamente usados em telefones celulares e sistemas eletrônicos dedicados.
 - [S] Neste tipo de sistema, a localização física dos recursos do sistema computacional é transparente para os usuários.
 - [D] Todos os recursos do sistema têm proprietários e existem regras controlando o acesso aos mesmos pelos usuários.
 - [T] 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.
 - [M] Construído para gerenciar de forma eficiente, grandes volumes de recursos.
 - [T] 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) **Mascarar uma ou mais interrupções.**
Por que isso é uma atribuição do kernel
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.**
Deveria ser feita através de chamada de sistema (primitiva) pois é uma operação que requer privilégios (função do kernel).
 - (e) Remover um arquivo do disco.
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).
- [4] A rotina de tratamento da interrupção de software é ativada dentro do núcleo.
 - [7] A função `printf` finaliza sua execução e devolve o controle ao código do processo.
 - [2] A função de biblioteca `printf` recebe e processa os parâmetros de entrada (a string "Hello world").
 - [1] A função de biblioteca `printf` prepara os registradores para solicitar a chamada de sistema `write()`
 - [5] O disco rígido gera uma interrupção indicando a conclusão da operação.
 - [3] O escalonador escolhe o processo mais prioritário para execução.
 - [6] Uma interrupção de software é acionada.

[0] O processo chama a função printf da biblioteca C.

[9] A operação de escrita no terminal é efetuada ou agendada pela rotina de tratamento da interrupção.

[8] 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 multiusuários associa um proprietário a cada recurso do sistema e gerenciar 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.

III está errada porque é uma característica de Sistema Operacional Embarcado;

IV está errada porque é uma característica de Sistema Desktop.

(d) As afirmações III, IV e V estão erradas.

(e) Todas as afirmações estão erradas.

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

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 ...):

I está errada porque uma máquina virtual de sistema é construída para suportar sistemas operacionais convidados completos;

III está errada porque isso ocorre em sistemas monolíticos e não micro-núcleos;

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?

Para o carregamento de bibliotecas compartilhadas, mapeamento da memória e, no final do rastreo, a emissão das informações sobre a data para a saída padrão.

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?

O utilitário `date` chama uma biblioteca com o fim de comunicar com o núcleo. Esta biblioteca manipula os detalhes de baixo nível relacionados com a passagem de informação para o núcleo e com a chamada da rotina privilegiada propriamente dita, nomeadamente a conversão de convenções de chamadas. As chamadas de sistema são oferecidas para as aplicações em modo usuário através da `system library` que prepara os parâmetros, invoca a interrupção de software e retorna à aplicação os resultados obtidos.