

Introdução a sistemas operacionais

Felipe Barbosa Nicolau Fernandes

1. Quais os dois principais objetivos dos sistemas operacionais?

Os dois principais objetivos dos sistemas operacionais (SO) são de abstração de recursos 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?

Porque o desenvolvedor não precisa se preocupar com os requisitos de hardware na construção de sua aplicação. A abstração ajuda no desenvolvimento do próprio SO, pois sua implementação se torna mais estruturada com o uso de interfaces homogêneas.

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.

A principal vantagem é diminuição de tempo ocioso do processador, ou seja, ele sempre estará executando instruções de aplicações distintas, melhorando o desempenho para o usuário. A desvantagem é que a implementação é complexa e é necessário um conjunto de políticas adotadas para definir esse paralelismo, a disputa pelo uso do processador.

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

Um SO de tempo real é aquele que trabalha de forma contínua (sem interrupções) e seus resultados são transmitidos com o mínimo de espera possível. Eles são classificados em *Soft*, onde a perda de prazos na transmissão de dados pode ocasionar em falhas no serviço prestado, e *Hard*, onde esse tipo de falha pode ocasionar consequências mais graves, do tipo humana, econômica ou ambiental

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

Ele é considerado o coração do sistema operacional, onde ele vai gerenciar os recursos de hardware solicitados pelas aplicações, além disso implementa as principais abstrações utilizadas pelos programas aplicativos.

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

Não, porque aplicações poderiam ter acesso à instruções de nível de hardware, podendo modificar o comportamento do sistema operacional.

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?

Sim, poderia fazer subníveis no nível de usuário, dividindo-os por aplicação.

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

Interrupções são instruções vindas do hardware que entram como prioridade na fila de execução do processador através do controle. Exceções são eventos disparados para tratamento quando o processador executa instruções ilegais (divisão por zero,

por exemplo). Trap é um instrução especial que os processadores possuem para dar um nível privilegiado de forma intencional.

- 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?**

Pode deixar o sistema vulnerável, pois está acessando a seção crítica, e interrupções de entrada/saída não serão atendidas. O sistema pode entrar em loop infinito, travando consecutivamente. O usuário não ter acesso a região crítica.

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

Uma função da biblioteca. Essa função a nível de usuário que irá fazer uma chamada de sistema para o SO.

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

- 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, S, E, D, M, E, K, S, K, E

- 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?**

F - Mascarar interrupções não é disponível para o usuário, pois está relacionado a região crítica.

- 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:**

- 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).**

> > 2 > 3 > > > 1 > >

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

C

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

- 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?**

- 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 item anterior?