

01.

Abstração e gerência de recursos.

02.

Porque ela provém interfaces de acessos aos dispositivos, o que simplifica no desenvolvimento de aplicações, já que o desenvolvedor não vai precisar acessar os recursos diretamente, apenas terá que se preocupar com o desenvolvimento em alto nível da sua aplicação. Além disso, a abstração é útil até mesmo para os desenvolvedores do próprio S.O., pois com a interface simples, o número de problemas e conflitos para acessar o hardware é diminuído.

03.

Com a gerência, não haverá conflitos para o uso do hardware, pois o S.O. define como gerenciar o acesso aos recursos de hardware. Os principais desafios são definir como o uso do processador será dividido entre as aplicações, impedir que um usuário ou uma aplicação monopolize todos os recursos disponíveis do sistema, etc.

04.

A sua característica essencial é ter um comportamento temporal previsível (ou seja, seu tempo de resposta deve ser conhecido no melhor e pior caso de operação). A estrutura interna de um sistema operacional de tempo real deve ser construída de forma a minimizar esperas, e latências imprevisíveis, como tempos

de acesso a disco e sincronizações excessivas. Há 2 tipos de S.O. de tempo real: soft real-time systems e hard real-time systems. Nos hard real-time, a perda de prazos pelo sistema pode perturbar o objeto controlado, com graves consequências humanas, econômicas ou ambientais. Já nos soft real-time, a perda de prazos implica na degradação do serviço prestado.

05-

Ele é a principal parte do S.O., o coração. O núcleo é responsável pela gerência dos recursos e pela implementação das principais abstrações utilizadas pelos aplicativos.

06-

Não, pois sem os diversos níveis de privilégio, qualquer pessoa/aplicação poderia acessar qualquer recurso, e isso seria uma vulnerabilidade à pessoas/aplicativos mal intencionados, podendo afetar tanto o sistema quanto o hardware.

07-

Sim, caso houvesse uma divisão de processamento em todos os níveis.

08-

Interrupções são eventos oriundos de periféricos, ou seja, externos ao processador. Os circuitos do processador suspendem seu fluxo de execução cor-

rante e fazem um desvio para algum endereço pré-definido, onde há um tratamento para a interrupção. Exceções são eventos que são causados pelo próprio processador. O funcionamento é semelhante ao das interrupções. Traps, assim como os anteriores, são eventos, porém, causados por softwares. São instruções especiais que permitem alterar o mecanismo de interrupções de forma proposital, sem depender de eventos internos ou externos.

09.

Pode trazer uma série de problemas. Ao desligar as interrupções, a preempção por tempo ou por recursos deixa de funcionar; caso a tarefa mal intencionada pode forçar essa situação e travar o sistema. Para evitar tais problemas, o S.O. deve programar a manipulação de interrupção de rede com o mesmo cuidado como cronogramas de execução de processos.

10.

Função de biblioteca. Quando um desenvolvedor está desenvolvendo uma aplicação, ele não utiliza chamadas de sistemas, geralmente, e sim, uma função de biblioteca, a qual irá converter a ação em várias chamadas de sistemas.

11.

Sistemas monolíticos:

• Vantagens:

- desempenho (qualquer componente do núcleo pode acessar os demais componentes, memória e periféricos). Mais compacto, devido à interação direta entre os componentes.

• Desvantagens:

- robustez, dificuldade de desenvolvimento, manutenção e evolução de núcleos mais complexos. Caso algum componente do núcleo perca o controle, o problema pode se espalhar por todo o núcleo, colapsando o sistema.

Sistema em camadas:

• Vantagens:

- É uma forma mais elegante de estruturar um S.O.. Permite níveis de abstração e gerência mais sofisticados.

• Desvantagens:

- O empilhamento de várias camadas faz com que cada pedido de aplicação demore mais tempo para chegar até o dispositivo periférico ou recurso a ser acessado, prejudicando o desempenho. Além disso, não é tão simples dividir as funcionalidades de um núcleo em camadas horizontais de abstração crescente, pois as funcionalidades são interdependentes.

Sistemas micro-núcleo:

• Vantagens:

- Robustez e flexibilidade. Se um subsistema tiver problemas, os mecanismos de proteção de memória e níveis de privilégios irão confiná-lo

impedindo que a instabilidade seja espalhada no restante do sistema. A customização do S.O. também é possível, iniciando somente os componentes necessários ou escolhendo os mais adequados às aplicações que serão executadas.

• Desvantagens:

- custo entre trocas de mensagens pode ser elevado, o que prejudica seu desempenho, diminuindo a aceitação desta abordagem.

Máquinas Virtuais:

• Vantagens:

- Contorna os problemas de compatibilidade entre os componentes de um sistema. Oferece aos demais componentes serviços com outra interface, permitindo o acoplamento entre interfaces distintas. Executa diversos sistemas operacionais sobre o mesmo hardware, simultaneamente. Aumenta a portabilidade, diminui os custos com hardware, etc.

• Desvantagens:

- custo adicional de execução dos processos na máquina virtual em comparação com a máquina real.

12.

[T] Deve ter um comportamento temporal previsível, com prazos de respostas 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.

[C] 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.

[S] 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 grande volume de recursos.

[K] O Mac OS X é um exemplo típico deste tipo de sistema.

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

13.

Escrever um valor em uma posição da memória e mascarar uma ou mais interrupções, pois isso poderia afetar o funcionamento do sistema.

14.

(b) Enviar um pacote através da rede.

(d) Preencher uma área da memória do processo com zeros.

(e) Remover um arquivo do disco.

15.

[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.

[2] A função de biblioteca printf recebe e processa os parâmetros de entrada (a string "Hello world").

[3] A função de biblioteca printf prepara os registradores para solicitar a chamada de sistema write();

[7] O disco rígido gera uma interrupção indicando a conclusão da operação.

[8] O escalonador escolhe o processo mais prioritário para a execução.

[4] Uma interrupção de software é acionada.

[1] 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-

I correto. II correto. III errado, pois os sistemas distribuídos são quem deixa tudo de forma transparente.

IV errado, são os sistemas desktops que devem priorizar as tarefas que interagem com o usuário. V correto.

17-

I falso, ela permite o acoplamento de interfaces distintas, não sendo específico à vma. Engavagem de

programação. II correto. III falso, somente as abstrações fundamentais ficam no núcleo. IV falso, isso são desvantagens deles. São usados por terem desempenho alto. II correto.

18.

Porque ele precisa acessar as rotinas do Sistema Operacional, essas chamadas de sistemas são como portas de entrada, pois elas irão verificar se a aplicação possui os privilégios necessários para executar a ação.

19.

O Itrace é uma abstração do strace em alto nível, já que o strace são as chamadas de sistemas realizadas pelas chamadas de bibliotecas.