

1. Abstração de recursos e Gerência de recursos
2. A abstração de recursos simplifica a construção de programas aplicativos, torna os aplicativos independentes do hardware e através de suas abstrações, o sistema operacional permite aos aplicativos usar a mesma interface para dispositivos diversos.
3. Os computadores normalmente possuem menos processadores que o número de tarefas em execução. Por isso, o uso desses processadores deve ser distribuído entre os aplicativos presentes no sistema, de forma que cada um deles possa executar na velocidade adequada para cumprir suas funções sem prejudicar os demais. Entretanto, em um sistema com várias atividades simultâneas, podem surgir conflitos no uso do hardware, quando dois ou mais aplicativos precisam dos mesmos recursos para poder executar. Cabe ao sistema operacional definir políticas para gerenciar o uso dos recursos de hardware pelos aplicativos, e resolver eventuais disputas e conflitos.
4. Um sistema operacional de tempo real tem 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.

Existem duas classificações de sistemas de tempo real: soft real-time systems, nos quais a perda de prazos implica na degradação do serviço prestado. Um exemplo seria o suporte à gravação de CDs ou à reprodução de músicas. Caso o sistema se atrase, pode ocorrer a perda da mídia em gravação ou falhas na música que está sendo tocada. Por outro lado, nos hard real-time systems a perda de prazos pelo sistema pode perturbar o objeto controlado, com graves consequências humanas, econômicas ou ambientais. Exemplos desse tipo de sistema seriam o controle de funcionamento de uma turbina de avião a jato ou de uma caldeira industrial.

5. O núcleo é o coração do sistema operacional, responsável pela gerência dos recursos do hardware usados pelas aplicações. Ele também implementa as principais abstrações utilizadas pelos programas aplicativos.
6. Não, para assegurar a integridade do sistema operacional, é essencial garantir que as aplicações não consigam acessar o hardware

diretamente, mas sempre através de pedidos ao sistema operacional, que avalia e intermedeia todos os acessos ao hardware

7. Sim, se com divisão de processamento entre os níveis podem ser usados para oferecer uma política de segurança.
8. As interrupções são eventos causados por dispositivos externos ao processador, exceções são eventos gerados pelo próprio processador, traps são eventos causados por software.
9. O processador perde tempo para varrer todos os dispositivos do sistema para verificar se há eventos a serem tratados ou não
10. É uma função da biblioteca padrão da linguagem C, ela abre o arquivo cujo nome é indicado por filename.
11. Em um sistema monolítico, todos os componentes do núcleo operam em modo núcleo e se inter-relacionam conforme suas necessidades, sem restrições de acesso entre si, pois o código no nível núcleo tem acesso pleno a todos os recursos e áreas de memória. A grande vantagem dessa arquitetura é seu desempenho: qualquer componente do núcleo pode acessar os demais componentes, toda a memória ou mesmo dispositivos periféricos diretamente, pois não há barreiras impedindo esse acesso. A interação direta entre componentes também leva a sistemas mais compactos. Todavia, a arquitetura monolítica pode pagar um preço elevado por seu desempenho: a robustez e a facilidade de desenvolvimento. Caso um componente do núcleo perca o controle devido a algum erro, esse problema pode se alastrar rapidamente por todo o núcleo, levando o sistema ao colapso (travamento, reinicialização ou funcionamento. Uma forma mais elegante de estruturar um sistema operacional faz uso da noção de camadas: a camada mais baixa realiza a interface com o hardware, enquanto as camadas intermediárias proveem níveis de abstração e gerência cada vez mais sofisticados. Por fim, a camada superior define a interface do núcleo para as aplicações (as chamadas de sistema). Essa abordagem de estruturação de software fez muito sucesso no domínio das redes de computadores, através do modelo de referência OSI (Open Systems Interconnection) [Day, 1983], e também seria de se esperar sua adoção no domínio dos sistemas operacionais.

No entanto, alguns inconvenientes limitam sua aceitação nesse contexto:

- O empilhamento de várias camadas de software faz com que cada pedido de uma aplicação demore mais tempo para chegar até o

dispositivo periférico ou recurso a ser acessado, prejudicando o desempenho do sistema.

- Não é óbvio como dividir as funcionalidades de um núcleo de sistema operacional em camadas horizontais de abstração crescente, pois essas funcionalidades são interdependentes, embora tratem muitas vezes de recursos distintos. Em decorrência desses inconvenientes, a estruturação em camadas é apenas parcialmente adotada hoje em dia. Muitos sistemas implementam uma camada inferior de abstração do hardware para interagir com os dispositivos (a camada HAL – Hardware Abstraction Layer, implementada no Windows NT e seus sucessores), e também organizam em camadas alguns subsistemas como a gerência de arquivos e o suporte de rede (seguindo o modelo OSI). Como exemplos de sistemas fortemente estruturados em camadas podem ser citados o IBM OS/2 e o MULTICS [Corbató and Vyssotsky, 1965]

12.

T

S

E

D

M

E

K

S

D

E

13. C - No nível usuário o hardware restringe o uso da memória, permitindo o acesso somente a áreas previamente definidas.

14. D - Os sistemas operacionais definem chamadas de sistema para todas as operações envolvendo o acesso a recursos de baixo nível (periféricos, arquivos, alocação de memória, etc.)

15.

5

9

1

4

8

2

7

3

6

16. As afirmações III e IV estão erradas. III está errado pois é uma característica de sistemas distribuídos e não de rede; e IV está errado pois é uma característica de sistemas desktop e não de tempo real.
17. As afirmações II e V estão corretas. I está errada pois uma máquina virtual de sistema é construída para suportar sistemas operacionais convidados completos; III está errada pois isso ocorre em sistemas monolíticos e não micronúcleos; IV está errada pois sistemas monolíticos não tem uma manutenção fácil e sim complexa.