

Aluno: Leandro Bernardes

1º) Qual a finalidade das chamadas de sistema?

Chamadas de sistema são funções que os programas dos usuários podem usar para acessar os serviços do núcleo, e tem como finalidade de trazer comandos da aplicação e transformá-los em comandos no quais são interpretados pelo núcleo do sistema Operacional e assim passando para o hardware.

2º) Quais as principais atividades de um sistema operacional relacionadas com o gerenciamento de processos?

Um sistema operacional é quem gerencia os recursos computacionais assim evitando conflitos entre recursos, dentre tais recursos estão:

Algoritmos e estruturas de dados para implementar processos e abstrações de recursos

O escalonamento de Processos

Sincronização de processos

Estratégia para tratamento de bloqueios perpétuos

Parte dos mecanismos de proteção e segurança

3º) Quais as principais atividades de um sistema operacional relacionadas com o gerenciamento de memória?

Manter os registros das partes da memória que estão sendo utilizados, decidir qual processo deve ser alocado na memória quando houver espaço disponível

Alocar e deslocar espaço em memória

4º) Quais as principais atividades de um sistema operacional relacionadas com o gerenciamento de memória secundária?

Gerenciamento de espaço livre, alocação de espaço e escalonamento de disco.

5º) Qual a finalidade do interpretador de comandos? Por que geralmente ele é separado do kernel?

Tradutor que funciona em tempo de execução. Estes programas traduzem programas codificados em linguagem de alto nível para um código intermediário e o coloca em execução. O interpretador fica fora do Kernel pois ele é um utilitário do SO no qual se comunicam por System calls.

6º) Qual a principal vantagem da abordagem em camadas para o projeto de sistemas? Quais as desvantagens?

Por ser separados, tem uma maior facilidade de modificação do código, contudo para utilizar outras camadas deve se utilizar várias chamadas de funções.

7º) Quais são os dois modelos de comunicação entre os processos? Cite as vantagens e desvantagens destas duas abordagens.

Memória compartilhada: Vantagem é a segurança – não funciona em sistemas distribuídos

Troca de mensagens: Implementação simples – Programar a comunicação em cada ponto

8º) Qual é o relacionamento entre um sistema operacional convidado e um sistema operacional host como em um sistema utilizando VMware? Que fatores tem que ser considerados na seleção do sistema operacional host?

O sistema

9º) De que maneiras a abordagem de um kernel do tipo microkernel é semelhante à abordagem em camadas? De que maneira ele difere da abordagem em camadas?

A abordagem do kernel é semelhante ao microkernel no sentido de que seus serviços são separados um do outro, ou seja, são vários módulos. O que difere é que em camadas ainda depende de seus precedentes.

10º) Por que um compilador just-in-time é útil na execução de programas em java?

Compiladores Just-In-time são compiladores que traduziam enquanto o programa estava sendo executado. Esses compiladores nada mais fazem do que traçar o perfil do programa em execução para descobrir onde estão os métodos principais do programa, e depois os compilam para o conjunto de instruções nativo em que a máquina virtual está executando. A parte compilada é salva para a próxima vez em que o programa for executado, de modo que a execução possa ser feita mais rapidamente da próxima vez em que ele for executado.

11º) Qual a principal vantagem da abordagem microkernel para o projeto de sistemas? Como os programas de usuário e serviços interagem em uma arquitetura de microkernel? E desvantagens desta abordagem?

Deixa os processos mais protegidos. No qual tem que acessar o núcleo que está dividido as tarefas o qual estes precisam alternando as chaves dependendo da complexidade da tarefa. Sua implementação é mais complexa.

12º) Explique o que vem a ser o modelo de kernel monolítico e qual a sua importância nos atuais modelos de sistemas operacionais.

A principal característica do kernel monolítico é permitir que funções como rede, vídeo e acesso a outros periféricos sejam possíveis através do *kernel-space*. Isso é possível através do uso de módulos. O que significa que um

módulo, apesar de não estar no mesmo código do kernel, é executado no espaço de memória do kernel. Sendo assim, apesar de modular, o kernel monolítico continua sendo único e centralizado