

Sistema Operacional

É um sistema que possibilita o uso eficiente e controla os componentes e hardware (memória, dispositivo de entrada e saída, unidade de processamento), implementando políticas e estruturas de software de forma a gerar um melhor desempenho do sistema de computação.

Organização de um Sistema Operacional

Núcleo (kernel)

- Responsável pela gerência do processador, tratamento de interrupções, comunicação e sincronização entre processos.

Gerente de Memória

- Responsável pelo controle e alocação de memória dos processos ativos.

Sistema de Arquivos

- Responsável pelo acesso e integridade dos dados residentes na memória secundária.

Sistema de E/S

- Responsável pelo controle e execução de operações de E/S e otimização do uso dos periféricos.

Processador de Comandos (shell) / Interface com o Usuário

- Responsável pela interface conversacional com o usuário.

Tipos de Sistemas Operacionais

Sistemas de Lotes (Batch)

Foi o primeiro tipo de SO, possuía um modelo de ser compilado e executado, acompanhando os seus dados em forma de Jobs (Processos) semelhantes que eram agrupados em lotes para reduzir o tempo de setup.

Possuía um processamento sequencial e inexistência de computação interativa (comunicação com o mundo externo, ex: pessoa) e como as tarefas eram processadas em conjunto (Jobs), o número por unidade de tempo era alta (porque não havia interrupções) e possuíam um tempo de resposta muito alto, pois poderia haver jobs grandes a frente de jobs pequenos.

Sistemas de Tempo Compartilhado

Esse sistema possibilita a manipulação de múltiplos jobs, onde a capacidade e tempo de processamento da máquina era dividida entre múltiplos usuários e os comandos de usuários eram interpretados e executados de forma online (via terminal).

Possuíam um tempo de resposta e o número de tarefas processadas por unidade de tempo baixa e arquiteturas mais complexas de propósito geral.

Lotes x Tempo Compartilhado (Time Sharing)

	Sistemas Batch	Tempo Compartilhado
Principal objetivo	Maximizar o uso do processador (throughput)	Minimizar o tempo de resposta (response time)
Fonte de diretivas para o sistema operacional	Job control language Comandos providos com o job	Comandos via terminal

Sistemas de Tempo Real

Eram sistemas dedicados a aplicações específicas (por ex: controle e medição), e possuíam um tempo de resposta crítico com hardware e software em Real-Time.

Mono x Multiprogramação (com time sharing)

Monoprogramação

Na monoprogramação, os recursos eram dedicados em apenas uma tarefa, e enquanto aguardava algum evento externo (entrada de texto/leitura de disco) a CPU ficava ociosa, não havia preocupações com a proteção de memória (A memória principal é subutilizada caso o programa não a preencha totalmente) e possuía complexidade baixa em sua implementação.

Por ser dedicada a uma única tarefa, possui periféricos dedicados a um único usuário.

Multiprogramação

O objetivo é manter mais de um programa em execução simultaneamente, para dar ilusão de que cada programa teria a máquina dedicada para si, e tirar proveito da ociosidade da CPU, enquanto ela ficaria ociosa outros programas poderiam ser processados para tirar o máximo de proveito do processador.

A multiprogramação teria um suporte mais adequado para o hardware, como proteção de memória e mecanismos de interrupção, e discos magnéticos para o acesso de forma randômica para os programas, tendo maior desempenho em operações E/S.

Algumas perguntas referente a aula:

1. Defina o que é um Sistema Operacional, descrevendo suas principais funções:

Resposta:

É um conjunto de programas (sistema) que possibilita o uso eficiente e controla os componentes e hardware (memória, dispositivo de entrada e saída, unidade de processamento), implementando políticas e estruturas de software de forma a gerar um melhor desempenho do sistema de computação.

Suas principais funções se resumem em ser responsável pelo gerenciamento de todo o hardware do computador, garantindo que um programa não interfira no outro e abstraindo a complexidade do hardware para facilitar o uso pelo usuário. Em suas funções básicas o SO interpreta os comandos do usuário, controlam os periféricos e organizam os arquivos em disco.

2. Quais as principais dificuldades que um programador teria no desenvolvimento de uma aplicação em um ambiente sem um sistema operacional?

Resposta:

A maior dificuldade que o programador teria seria em programar em linguagem de máquina, já que o hardware não possui ambiente programável. O programador teria que programar todo os componentes de hardware: vídeo, teclado, mouse, placa de rede, disco rígido. Cada tarefa que a aplicação fosse fazer teria que ser detalhada num nível muito baixo para que o computador funcione.

3. O que é multiprogramação? Cite duas razões para se ter multiprogramação? Cite exemplos de problemas de segurança que podem ocorrer em sistemas multiprogramados.

Resposta:

Os sistemas multiprogramáveis ou multitarefa são uma evolução dos sistemas monoprogramáveis. Neste tipo de sistema, por exemplo, enquanto um programa espera por uma operação de leitura ou gravação em disco, outros programas podem estar sendo processados neste mesmo intervalo de tempo. Nesse caso, podemos observar o compartilhamento da memória e do processador.

As vantagens do uso de sistemas multiprogramáveis são: a redução do tempo de resposta das aplicações processadas no ambiente e de custos, a partir do compartilhamento dos diversos recursos do sistema entre as diferentes aplicações.

Os sistemas multiprogramáveis, apesar de mais eficientes que os monoprogramáveis, são de implementação muito mais complexa. Um problema é a segurança. Por exemplo, uma vez que diferentes processos estão em memória e com isso, poderiam acessar inadvertidamente a memória de outro processo.

4. Porque sistemas distribuídos são desejáveis?

Resposta:

Um sistema distribuído é uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente. Pois beneficia o trabalho em rede, não precisando necessariamente todos os processos envolvidos estarem em um único sistema. São sistemas mais tolerantes a falhas: falhas em partes do sistema não comprometem o sistema como um todo