

Sistema Operacional

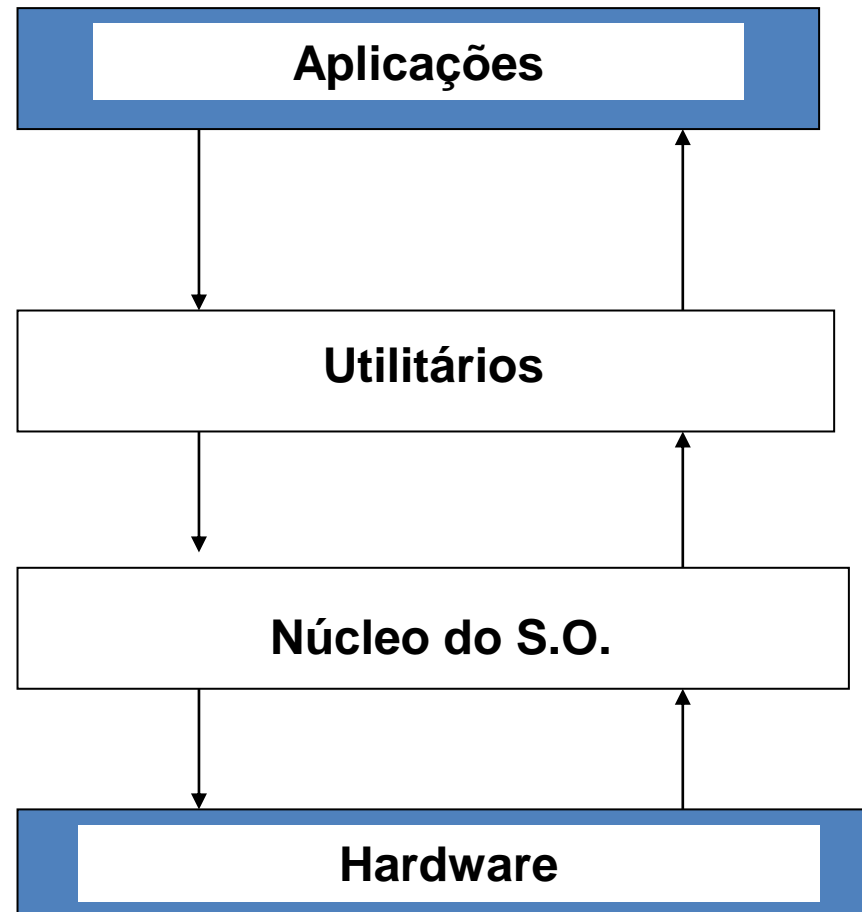
Estrutura

Introdução

- O sistema operacional é formado por um conjunto de rotinas que oferecem serviços aos usuários, às suas aplicações e também ao próprio sistema. Esse conjunto de rotinas é denominado “núcleo do sistema” ou “kernel”.

Introdução

- Sistema Computacional



Introdução

- Principais funções do núcleo:
 - Tratamento de interrupções e exceções
 - Criação e eliminação de processos e threads
 - Sincronização e comunicação entre processos e threads
 - Gerência de memória
 - Gerência do sistema de arquivos
 - Gerência de dispositivos de E/S
 - Suporte a redes locais e distribuídos
 - Contabilização do uso do sistema
 - Auditoria e segurança do sistema

System Calls

- Pode ser entendida como uma porta de entrada para o acesso ao núcleo do sistema operacional e a seus serviços.

Aplicação → System Call → Núcleo do S.O. → Hardware

System Calls

- Funções:
 - Gerência de processos e threads
 - Gerência de memória
 - Gerência do sistema de arquivos
 - Gerência de dispositivos

Modos de Acesso

- Instruções privilegiadas – têm o poder de comprometer o sistema
- Instruções não-privilegiadas – são as que não oferecem risco ao sistema
- Para que uma aplicação possa executar uma instrução privilegiada, é necessário que no processador seja implementado o mecanismo de proteção conhecido como “modo de acesso”.

Modos de Acesso

- Existem, basicamente, dois modos de acesso implementado pelos processadores: modo usuário e modo kernel.
- Modo Usuário – uma aplicação só pode executar instruções não-privilegiadas, tendo acesso a um número reduzido de instruções.
- Modo Kernel – a aplicação pode ter acesso ao conjunto total de instruções do processador.

Modos de Acesso

- A melhor maneira de controlar o acesso às instruções privilegiadas é permitir que apenas o SO tenha acesso a elas.
- Caso a aplicação tente executar uma instrução privilegiada diretamente em modo usuário, o processador sinalizará um erro, uma exceção é gerada e a execução do programa é interrompida.

Arquitetura Monolítica

- Pode ser comparada com uma aplicação formada por vários módulos que são compilados separadamente e depois linkados, formando um grande e único programa executável, onde os módulos podem interagir livremente.

Arquitetura Monolítica

- Devido a sua simplicidade e bom desempenho, foi adotado no projeto do MS-DOS e nos primeiros sistemas UNIX.

Arquitetura de Camadas

- O sistema é dividido em níveis sobrepostos, cada camada oferece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores.
- A vantagem da estruturação em camadas é isolar as funções do sistema operacional, facilitando sua manutenção e depuração, além de criar uma hierarquia de níveis de acesso, protegendo as camadas mais internas.

Arquitetura de Camadas

- Sua desvantagem é o desempenho. Cada nova camada implica uma mudança no modo de acesso.
- Atualmente, a maioria dos sistemas operacionais utiliza o modelo de duas camadas, onde existem os modos de acesso usuário (não privilegiado) e kernel (privilegiado).

Máquina Virtual

- O modelo de “máquina virtual” cria um nível intermediário entre o hardware e o SO, denominado gerência de máquinas virtuais.
- Este nível cria diversas máquinas virtuais independentes, onde cada uma oferece uma cópia virtual do hardware, incluindo os modos de acesso, interrupções, dispositivos de E/S, etc.

Máquina Virtual

- Permite a convivência de SO diferentes no mesmo computador, e cria um isolamento total entre cada máquina virtual, oferecendo grande segurança para cada máquina virtual. Se uma máquina virtual executar uma aplicação que comprometa seu funcionamento, as demais não sofrerão qualquer problema.

Máquina Virtual

- A desvantagem desta arquitetura é a sua grande complexidade, devido à necessidade de se compartilhar e gerenciar os recursos do hardware entre as diversas máquinas virtuais.

Máquina Virtual

- Um exemplo de utilização desta arquitetura ocorre na linguagem Java. Para se executar um programa em Java é necessário uma máquina virtual Java (Java Virtual Machine – JVM).
- Qualquer SO pode suportar uma aplicação Java, desde que exista um JVM desenvolvida para ele.
- Desta forma, a aplicação não precisa ser recompilada para sistema computacional.

Máquina Virtual

- A desvantagem deste modelo é o seu menor desempenho se comparado a uma aplicação compilada e executada diretamente em uma arquitetura específica.

Arquitetura Microkernel

- Uma tendência nos SO modernos é tornar o núcleo do SO menor e mais simples possível.

Projeto do Sistema

- Muito complexo e atender aos seguintes requisitos:
 - Confiabilidade
 - Portabilidade
 - Manutenibilidade
 - Flexibilidade
 - Desempenho

Projeto do Sistema

- Dependo da arquitetura do hardware
- Tipo de sistema
- Windows 2000 possui mais de 40 milhões de linhas de código, em grande parte escrita em C/C++.