

Sistemas Operacionais

Aula 06 – Estrutura do Sistema Operacional
SCC5854

Prof. Dr. Jonathan Ramos
jonathan@unir.br

Departamento Acadêmico de Ciências de Computação – DACC
Núcleo de Tecnologia – NT

25/10/2022

Sumário

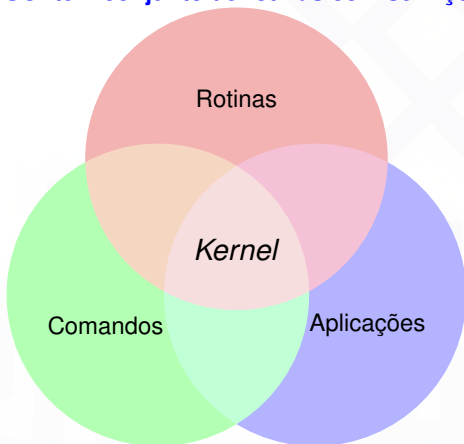
- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Introdução

SO: tem conjunto de rotinas com serviços aos usuários/aplicações:



Núcleo ou *Kernel*

Conjunto de rotinas que oferece **serviços aos usuários e às suas aplicações**.

- Acompanhada de **utilitários e linguagem de comandos**, que são ferramentas de apoio ao usuário, **porém não são parte do núcleo do sistema**.

Introdução

Modelo de camadas: estrutura do sistema operacional e suas três interfaces:

Aplicações

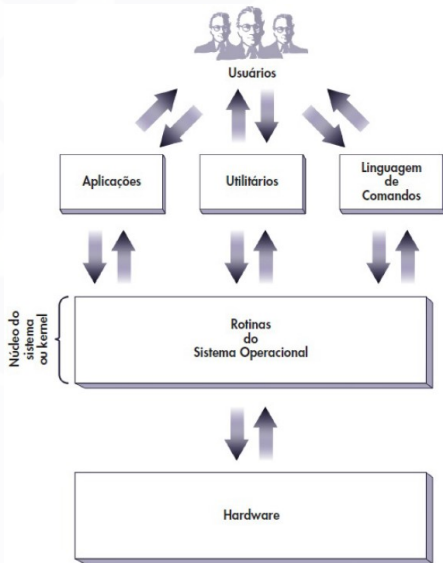
Ferramentas de apoio ao usuário.

Utilitários

Compiladores, editores de texto.

Comandos

Particular de cada Sistema (estrutura e sintaxe própria). Por exemplo, cmd.



Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Funções do Núcleo

A compreensão da estrutura e do funcionamento de um sistema operacional não é simples:

- Rotinas **não possuem começo, meio e fim**;
- Executadas **concorrentemente**;
- **Sem ordem predefinida**;
- Depende de **eventos assíncronos** (dissociado do tempo);
- Eventos **estão associados a hardware e tarefas internas do SO**.

Funções do Núcleo

As principais funções do núcleo encontradas nos SOs são:

- 1 Tratamento de interrupções e exceções;
- 2 Criação e eliminação de processos e *threads*;
- 3 Sincronização e comunicação entre processos e *threads* (**Grupo Bytes Negativos**);
- 4 Escalonamento e controle dos processos e *threads* (**Grupo Kali**);
- 5 Gerência de memória (**Grupos Espaguete Software e Winx**);
- 6 Gerência do sistema de arquivos;
- 7 Gerência de dispositivos de E/S;
- 8 Suporte a redes locais e distribuídas;
- 9 Contabilização do uso do sistema;
- 10 Auditoria e segurança do sistema.

Funções do Núcleo

Devido a complexidade, surgem alguns problemas:

Segurança no inter-relacionamento dos diversos subsistemas existentes

- Diversos usuários **compartilham os mesmos recursos**:
 - Memória;
 - Processador;
 - Dispositivos de E/S.
- SO deve garantir a **confiabilidade na execução concorrente**:
 - Programas;
 - Dados dos usuários;
- Garantir integridade do próprio SO;

Multiprogramação multiusuário

- SO deve controlar a UCP, de forma que nenhum programa monopolize o seu uso inadequadamente.
- Cada usuário deve possuir uma área reservada onde seus dados e código são armazenados.
- Mecanismos de proteção (violação de acesso);
- Acesso sincronizado e controlado aos recursos (inconsistência);

Funções do Núcleo

Para solucionar esses **diversos problemas** originados pelo ambiente **multiprogramável/multiusuário**



o SO deve implementar mecanismos de proteção que **controlem o acesso concorrente** aos diversos recursos do sistema.

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso**
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Modo de acesso

Mecanismo de acesso/proteção ao núcleo/serviços

**Operação que
altera integri-
dade do núcleo**

**Todo sistema
pode ficar
comprometido
e inoperante**

**Modo de
Acesso**

**Mecanismo
presente no
hardware dos
processadores**

Modo de acesso

Modos de Acesso

**Modo
usuário**

só pode executar instruções conhecidas como não privilegiadas, acesso a um número reduzido de instruções

**Modo
Kernel**

Pode ter acesso ao conjunto total de instruções do processador

Modo de acesso

Instruções privilegiadas:

- Não devem ser utilizadas de maneira indiscriminada pelas aplicações;
- Poderia ocasionar sérios problemas à integridade do sistema;
- Exemplo: acesso a disco, recurso compartilhado (gerenciado unicamente pelo sistema operacional)
 - O programa, por si só, não deve especificar diretamente as instruções que acessam seus dados no disco
- só podem ser executadas quando o modo de acesso do processador encontra-se em *kernel*, caso contrário o hardware irá impedir a execução da instrução.

Instruções não privilegiadas

- Não oferecem risco ao sistema e podem ser executadas em modo não privilegiado, ou seja, modo usuário

Modo de acesso

Acesso ao próprio núcleo do SO:

Residente na memória principal, acesso a área específica de forma mal-intencionada pode acarretar na violação do SO.

- Só pode ser alterada no modo **kernel**.

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls***
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Rotinas do SO e *System Calls*

Rotinas do sistema operacional

Compõem o núcleo do sistema, oferecendo serviços aos usuários e suas aplicações

Modo privilegiado:

Todas as funções do núcleo executam instruções privilegiadas e o processador deve estar no modo kernel

Segurança

Exige a implementação de mecanismos de proteção para garantir a confiabilidade do sistema

Controle de Execução

- As rotinas são controladas pelas ***System Calls***.
- Toda vez que uma aplicação desejar chamar uma rotina do sistema operacional, o mecanismo de system call é ativado

Rotinas do SO e *System Calls*

Passo a passo

- 1 O sistema operacional verificará se a aplicação possui privilégios necessários para executar a rotina desejada:
 - Em caso negativo, o sistema operacional impedirá o desvio para a rotina do sistema
 - sinalizando ao programa chamador que a operação não é possível
 - mecanismo de proteção por software
 - o sistema operacional garante que as aplicações só poderão executar rotinas do sistema que estão previamente autorizadas
- 2 O sistema operacional:
 - 1 salva o conteúdo corrente dos registradores
 - 2 troca o modo de acesso do processador de usuário para *kernel*
 - 3 realiza o desvio para a rotina alterando o registrador PC com o endereço da rotina chamada
- 3 Ao término da execução da rotina do sistema:
 - 1 o modo de acesso é alterado de *kernel* para usuário
 - 2 o contexto dos registradores restaurados para que a aplicação continue a execução a partir da instrução que chamou a rotina do sistema

Rotinas do SO e System Calls

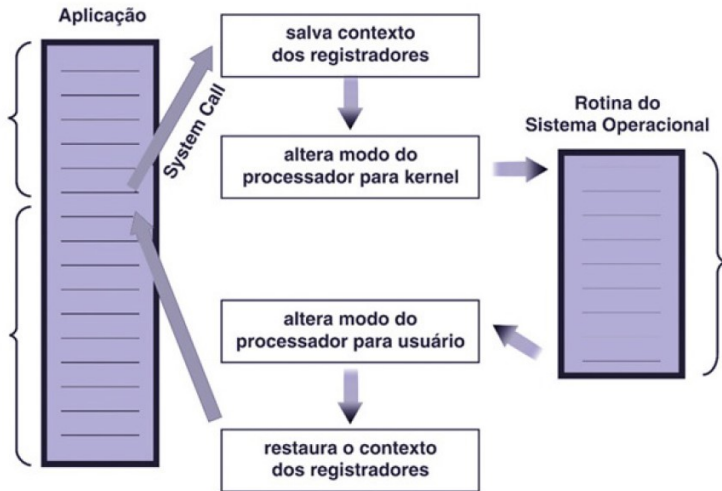


Figura: Chamada a uma rotina do sistema (a).

Rotinas do SO e *System Calls*

Uma aplicação deve executar com o processador em modo usuário:

Mecanismo de proteção por hardware garantirá a segurança do sistema:

Quando uma aplicação no modo usuário tentar executar rotinas privilegiadas:

- 1 O próprio hardware do processador sinalizará um erro;
- 2 Uma exceção é gerada;
- 3 A execução do programa é interrompida;
- 4 Protegendo desta forma o núcleo do sistema.

Mecanismos de *system call* e de proteção por hardware:

Garantem a segurança e a integridade do sistema:

- Aplicações estão impedidas de executarem instruções privilegiadas sem a autorização e a supervisão do sistema operacional

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO**
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Chamada a Rotinas do SO

Rotinas do sistema e o mecanismo de *system call*

Podem ser entendidos como uma porta de entrada para o núcleo do sistema operacional e a seus serviços

Sempre que uma aplicação desejar algum serviço do sistema:

Realiza uma chamada a uma de suas rotinas através de uma *system call*:

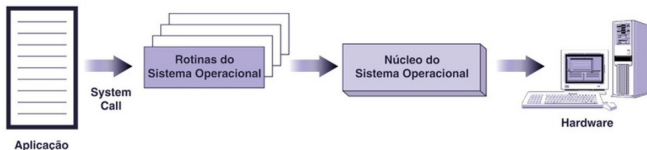


Figura: Chamada a uma rotina do sistema (b).

A solicitação é processada e uma resposta é retornada à aplicação juntamente com um estado de conclusão indicando se houve algum erro

Chamada a Rotinas do SO

As rotinas do sistema podem ser divididas por grupos de função:

Funções	System Calls
A) Gerência de processos e threads	1 – Criação e eliminação de processos e threads 2 – Alteração das características de processos e threads 3 – Sincronização e comunicação entre processos e threads 4– Obtenção de informações sobre processos e threads
B) Gerência de memória	5 –Alocação e desalocação de memória
C) Gerência do sistema de arquivos	6 – Criação e eliminação de arquivos e diretórios 7 – Alteração das características de arquivos e diretórios 8– Abertura e fechamento de arquivos 9 – Leitura e gravação em arquivos 10 – Obtenção de informações sobre arquivos e diretórios
D) Gerência de dispositivos	11 – Alocação e desalocação de dispositivos 12 – Operações de entrada/saída em dispositivos 13 – Obtenção de informações sobre dispositivos

De forma simplificada:

O comando da linguagem de alto nível é convertido pelo compilador **para uma chamada a uma rotina específica** que, quando executada, **verifica a ocorrência de erros e retorna os dados ao programa de forma transparente ao usuário**

Chamada a Rotinas do SO

Existem 2 maneiras de chamada a uma rotina do sistema operacional:

Implícita.

Realizada por intermédio de um comando da própria linguagem de programação

Explícita

No código do programa há uma função explicitando a chamada a rotina do sistema com passagem de parâmetro

No chamada implícita:

Quando este comando é traduzido para uma instrução de mais baixo nível, há uma conversão do comando em uma chamada da função.

Cada sistema operacional:

- Possui seu próprio conjunto de rotinas, com nomes, parâmetros e formas de ativação específicos;
- Uma aplicação desenvolvida utilizando serviços de um determinado sistema operacional não pode ser portada diretamente para um outro sistema, exigindo algumas correções no código-fonte
- POSIX (*Portable Operating System Interface for Unix*): biblioteca com chamadas padronizadas, sendo suportada pela maioria dos SOs atualmente

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos**
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Linguagem de Comandos

Linguagem de comandos, ou linguagem de controle

- Permite que o usuário se comunique de uma forma simples com o sistema operacional;
- O usuário dispõe de uma interface direta com o sistema operacional;
- Capacitando-o a executar diversas tarefas específicas do sistema como criar, ler ou eliminar arquivos, consultar diretórios ou verificar a data e a hora armazenadas;
- Cada sistema operacional possui a sua linguagem de comandos

Comando	Descrição
dir	Lista o conteúdo de um diretório
cd	Altera o diretório <i>default</i>
type	Exibe o conteúdo de um arquivo
del	Elimina arquivos
mkdir	Cria um diretório
ver	Mostra a versão do Windows

Tabela: Exemplos de comandos do MS Windows

Linguagem de Comandos

Cada comando, depois de digitado pelo usuário, é interpretado pelo shell ou interpretador de comandos:

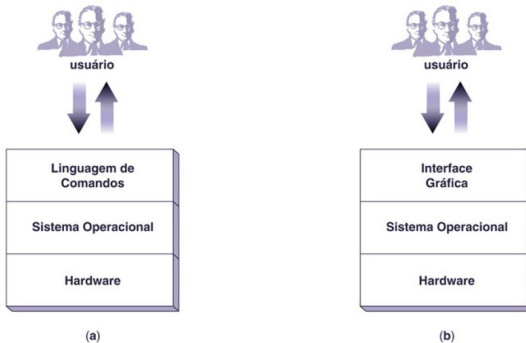


Figura: Interface do usuário com o sistema operacional.

Verifica a sintaxe do comando, faz chamadas a rotinas do sistema e apresenta um resultado ou uma mensagem informativa

Linguagem de Comandos

Na maioria dos sistemas operacionais, as linguagens de comandos evoluíram

- Permitem a interação mais amigável com os usuários;
- Utiliza interfaces gráficas como janelas e ícones, a exemplo dos sistemas MS Windows.
- Na maioria dos casos, a interface gráfica é apenas mais um nível de abstração entre o usuário e os serviços do sistema operacional

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema**
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Ativação/Desativação do Sistema

Inicialmente, quando um computador é ligado não há sistema operacional carregado na memória da máquina:

o sistema operacional reside em um disco rígido, podendo também estar armazenado em outros dispositivos de memória secundária, como CD ou DVD

Ativação do sistema ou *boot*

Carregados para a memória principal toda vez que o computador:

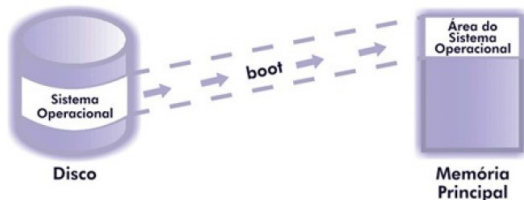


Figura: Ativação do sistema.

Ativação/Desativação do Sistema

Passos para a ativação do sistema:

- 1 Inicialização do boot loader (endereço fixo de uma memória ROM);
- 2 boot loader chama o POST (Power-on Self test): identifica problemas de hardware;
- 3 O procedimento de ativação verifica se há no sistema computacional algum dispositivo de armazenamento onde haja um sistema operacional residente;
- 4 Caso nenhum dispositivo seja encontrado, uma mensagem de erro é apresentada e o procedimento de ativação é interrompido.
- 5 Se um dispositivo com o sistema operacional é encontrado, um conjunto de instruções é carregado para memória e localizado em um bloco específico do dispositivo conhecido como setor de boot (boot sector);
- 6 O SO é finalmente carregado para a memória principal;
- 7 a ativação do sistema também consiste na execução de arquivos de inicialização onde são especificados procedimentos de customização e configuração de hardware e software específicos para cada ambiente

Desativação ou *shutdown*

Permite que as aplicações e componentes do sistema operacional sejam desativados ordenadamente, garantindo, desta forma, sua integridade

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo**
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Arquiteturas do Núcleo

- Os primeiros sistemas operacionais foram desenvolvidos integralmente em assembly, e o código possuía cerca de um milhão de instruções (IBM OS/360).
- Com a evolução dos sistemas e o aumento do número de linhas de código para algo perto de 20 milhões (MULTICS),
- Nos sistemas operacionais atuais, o número de linhas de código pode chegar a mais de 40 milhões (Windows 2000),
- grande parte do código é escrita em Linguagem C/C++.

Arquiteturas do Núcleo

A estrutura do núcleo do sistema operacional

a maneira como o código do sistema é organizado e o inter-relacionamento de seus diversos componentes:

- 1 arquitetura monolítica;
- 2 arquitetura de camadas;
- 3 máquina virtual e
- 4 arquitetura microkernel.

Arquitetura Monolítica

Monolítico:

Vários módulos compilados separadamente e *linkados*, como um grande e único programa executável.

Desvantagem

Desenvolvimento e manutenção mais difícil

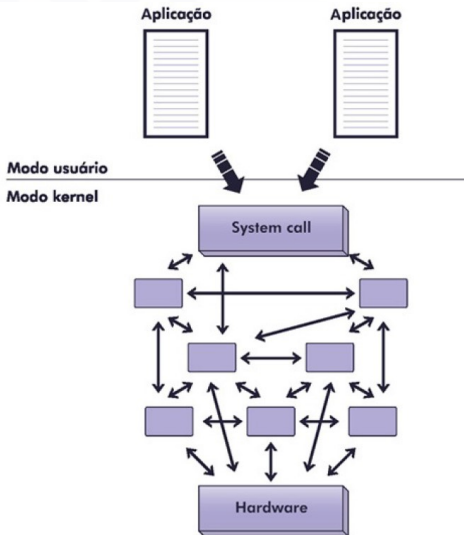


Figura: Arquitetura monolítica.

Arquitetura de Camadas



Figura: Arquitetura em camadas do OpenVMS.

Desvantagem

Desempenho.

Atualmente, a maioria dos sistemas comerciais utiliza o modelo de duas camadas, onde existem os modos de acesso usuário (não privilegiado) e kernel (privilegiado)

Camadas:

Dividido em níveis sobrepostos: Cada camada oferece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores:

- as camadas mais internas são mais privilegiadas que as mais externas.

Vantagem

Isolar Funções do SO, facilita manutenção e depuração, hierarquia de níveis de modo de acesso

Máquina Virtual - *Virtual Machine* (VM)

Nível intermediário entre o hardware e o sistema operacional:

Diversas máquinas virtuais independentes

Cada uma oferece uma cópia virtual do hardware, incluindo os modos de acesso, interrupções, dispositivos de E/S etc.

- É possível que cada VM tenha seus SO.
- Aplicações executam como se todo o computador estivesse dedicado a eles.

Vantagens

- 1 Portabilidade de código;
- 2 Consolidação de servidores;
- 3 Aumenta da disponibilidade;
- 4 facilidade de escalabilidade;
- 5 Facilidade do desenvolvimento de Software;

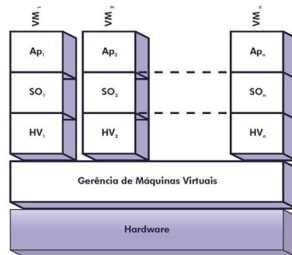


Figura: Máquina virtual: gerência de máquinas virtuais.

Arquitetura Microkernel: **compacto**

Uma aplicação solicita um serviço (cliente) ao processo responsável:

Os serviços do sistema são disponibilizados através de processos

Cada um é responsável por oferecer um conjunto específico de funções:

- Gerência de processos, gerência de memória e escalonamento

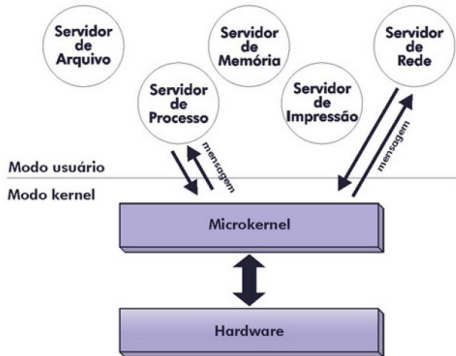


Figura: Arquitetura microkernel.

Servidor:

Processo que responde à solicitação do cliente.

- Cliente: pode ser uma aplicação de um usuário ou um outro componente do SO.
- Servidor: recebe a mensagem do cliente e responde-o através de uma outra mensagem.

A principal função do núcleo é realizar a comunicação (a troca de mensagens entre cliente/servidor)

Arquitetura Microkernel: **compacto**

Vantagem:

- permite isolar as funções do sistema operacional
- tornando o núcleo menor
- mais fácil de depurar e, conseqüentemente, aumentando sua confiabilidade

Desvantagem:

- Difícil implementação;
- problema de desempenho, devido à necessidade de mudança de modo de acesso a cada comunicação entre clientes e servidores
- certas funções do sistema operacional exigem acesso direto ao hardware, como operações de E/S;

Sumário

- 1 Introdução
- 2 Funções do Núcleo
- 3 Modo de acesso
- 4 Rotinas do SO e *System Calls*
- 5 Chamada a Rotinas do SO
- 6 Linguagem de Comandos
- 7 Ativação/Desativação do Sistema
- 8 Arquiteturas do Núcleo
 - Arquitetura Monolítica
 - Arquitetura de Camadas
 - Máquina Virtual
 - Arquitetura Microkernel
- 9 Exercícios

Exercícios I

- 1 O que é o núcleo do sistema e quais são suas principais funções?
- 2 O que são instruções privilegiadas e não privilegiadas? Qual a relação dessas instruções com os modos de acesso?
- 3 Explique como funciona a mudança de modos de acesso e dê um exemplo de como um programa faz uso desse mecanismo.
- 4 Como o *kernel* do sistema operacional pode ser protegido pelo mecanismo de modos de acesso?
- 5 Por que as rotinas do sistema operacional possuem instruções privilegiadas?
- 6 O que é uma *system call* e qual sua importância para a segurança do sistema? Como as *system calls* são utilizadas por um programa?
- 7 Quais das instruções a seguir devem ser executadas apenas em modo kernel?
Desabilitar todas as interrupções, consultar a data e a hora do sistema, alterar a data e a hora do sistema, alterar informações residentes no núcleo do sistema, somar duas variáveis declaradas dentro do programa, realizar um desvio para uma instrução dentro do próprio programa e acessar diretamente posições no disco.
- 8 Pesquise comandos disponíveis em linguagens de controle de sistemas operacionais.
- 9 Explique o processo de ativação (*boot*) do sistema operacional.
- 10 Compare as arquiteturas monolítica e de camadas. Quais as vantagens e desvantagens de cada arquitetura?

Exercícios II

- 11 Quais as vantagens do modelo de máquina virtual?
- 12 Como funciona o modelo cliente-servidor na arquitetura *microkernel*? Quais as vantagens e desvantagens dessa arquitetura?
- 13 Por que a utilização da programação orientada a objetos é um caminho natural para o projeto de sistemas operacionais?

FIM!

`jonathan@unir.br`