

## SSC 140 - SISTEMAS OPERACIONAIS I

Turmas A e B

### Aula 3 – Conceitos Básicos

Profa. Sarita Mazzini Bruschi

Slides de autoria de  
Luciana A. F. Martimiano baseados nos livros  
*Sistemas Operacionais Modernos* de A. Tanenbaum e  
*Arquitetura de Sistemas Operacionais* de Machado e Maia

## Roteiro

- Por que Sistemas Operacionais;
- Conceitos Básicos:
  - Processos;
  - Chamadas de Sistema – *system calls*;
  - Estrutura de Sistemas Operacionais;

2

## Por que?

- Sistemas de computadores modernos são compostos por diversos dispositivos:

- Processadores;
- Memória;
- Controladoras;
- Monitor;
- Teclado;
- Mouse;
- Impressoras;
- Etc...



Alta Complexidade

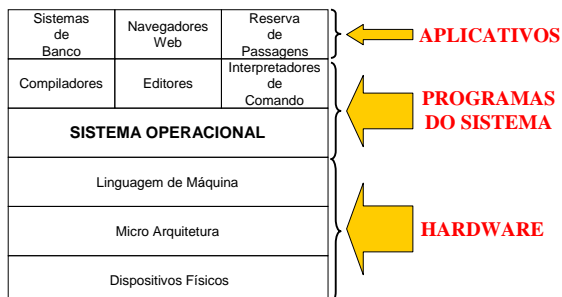
3

## Por que?

- Com tantos dispositivos, surge a necessidade de gerenciamento e manipulação desses diversos dispositivos;
- **Sistema Operacional:** Software responsável por gerenciar dispositivos que compõem um sistema computacional e realizar a interação entre o usuário e esses dispositivos;

4

## Arquitetura do Sistema



5

## Arquitetura do Sistema

- **Hardware:** Diversas camadas
  - Dispositivos físicos:
    - Circuitos (*chips*)
    - Cabos
    - Transistores
    - Capacitores
    - Memória
    - Disco rígido
    - etc...

6

## Arquitetura do Sistema

- ❑ **Micro Arquitetura:** dispositivos físicos são agrupados para formar unidades funcionais
  - CPU – processamento;
  - ULA (Unidade Lógica Aritmética) – operações aritméticas. Essas operações podem ser controladas por software (micro programas) ou por circuitos de hardware;

7

## Arquitetura do Sistema

- ❑ **Linguagem de Máquina:** conjunto de instruções interpretadas pelos dispositivos que compõem a micro arquitetura;
  - Possui entre 50 e 300 instruções;
  - Realiza operações por meio de registradores;
  - Baixo nível de abstração;
  - Ex.: **Assembler**.

8

## Sistema Operacional

- ❑ Pode atuar de duas maneiras diferentes:
  - Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário;
  - Como gerenciador de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador;

9

## Sistema Operacional como Máquina Estendida

- ❑ Ex.: como é feita a entrada/saída de um disco flexível – tarefa: Leitura e Escrita
  - SO: baixo nível de detalhes
    - ❑ Número de parâmetros;
    - ❑ Endereço de bloco a ser lido;
    - ❑ Número de setores por trilha;
    - ❑ Modo de gravação;
  - Usuário: alto nível – abstração simples
    - ❑ Visualização do arquivo a ser lido e escrito;
    - ❑ Arquivo é lido e escrito;
    - ❑ Arquivo é fechado.

10

## Sistema Operacional como Gerenciador de Recursos

- ❑ Gerenciar todos os dispositivos e recursos disponíveis no computador
  - Ex.: se dois processos querem acessar um mesmo recurso, por exemplo, uma impressora, o SO é responsável por estabelecer uma ordem para que ambos os processos possam realizar sua tarefa de utilizar a impressora.
  - Uso do HD;
  - Uso da memória;
- ❑ Coordena a alocação controlada e ordenada dos recursos;

11

## Classes de Sistemas Operacionais

Classe	Período	Principal Preocupação	Conceitos chave
Processamento Batch	1960s	Tempo de CPU parada	Transição automática entre os jobs
Multiprogramação	1960s	Utilização de recursos	Prioridade dos programas, preempção
Tempo compartilhado	1970s	Bom tempo de resposta	<i>Time slice</i> , round-robin, escalonamento
Tempo real	1980s	Atingir restrições de tempo	Escalonamento de tempo real
Distribuído	1990s	Compartilhamento de recursos	Controle distribuído, transparência

12

## Conceitos Básicos

### Processos

- ❑ Processo: chave do SO;
  - Caracterizado por programas em execução;
  - Cada processo possui:
    - ❑ Programa (instruções que serão executadas);
    - ❑ Um espaço de endereço de memória (mínimo, máximo);
    - ❑ Contextos de hardware: informações de registradores;
    - ❑ Contextos de software: atributos;
- ❑ O Sistema Operacional gerencia todos os processos → bloco de controle de processo (BCP) e tabela de processos;

13

## Conceitos Básicos

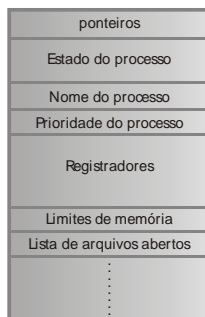
### Processos - Contextos



14

## Conceitos Básicos

### Processos - BCP



**Bloco de Controle de Processo: contém informações sobre o estado do processo**

15

## Conceitos Básicos

### Processos

- ❑ Basicamente, um processo possui três segmentos:
  - **Texto:** código do(s) programa(s);
  - **Dados:** as variáveis;
  - **Pilha de Execução:** controla a execução do processo;
- ❑ Um processo possui três estados básicos: executando, bloqueado e pronto;

16

## Conceitos Básicos

### Processos

- ❑ Um processo pode resultar na execução de outros processos, chamados de processos-filhos:
  - Características para a hierarquia de processos:
    - ❑ Comunicação (Interação) e Sincronização;
    - ❑ Segurança e proteção;
    - ❑ Uma árvore de no máximo três níveis;
- ❑ Escalonadores de processos – processo que escolhe qual será o próximo processo a ser executado;
  - Diversas técnicas para escalonamento de processos;
- ❑ Comunicação e sincronismo entre processos;

17

## Conceitos Básicos

### Chamadas de Sistema

- ❑ **Modos de Acesso:**
  - Modo usuário;
  - Modo *kernel* ou Supervisor ou Núcleo;
  - São determinados por um conjunto de bits localizados no registrador de status do processador: PSW (*program status word*);
    - ❑ Por meio desse registrador, o hardware verifica se a instrução pode ou não ser executada pela aplicação;
  - Protege o próprio *kernel* do Sistema Operacional na RAM contra acessos indevidos;

18

## Conceitos Básicos

### Chamadas de Sistema

#### ❑ Modo usuário:

- Aplicações não têm acesso direto aos recursos da máquina, ou seja, ao hardware;
- Quando o processador trabalha no modo usuário, a aplicação só pode executar **instruções sem privilégios, com um acesso reduzido de instruções**;
- Por que? Para garantir a **segurança** e a **integridade do sistema**;

19

## Conceitos Básicos

### Chamadas de Sistema

#### ❑ Modo *Kernel*:

- Aplicações têm acesso direto aos recursos da máquina, ou seja, ao hardware;
- **Operações com privilégios**;
- Quando o processador trabalha no modo *kernel*, a aplicação tem **acesso ao conjunto total de instruções**;
- Apenas o SO tem acesso às instruções privilegiadas;

20

## Conceitos Básicos

### Chamadas de Sistema

- ❑ Se uma aplicação precisa realizar alguma instrução privilegiada, ela realiza uma **chamada de sistema (system call)**, que altera do modo usuário para o modo *kernel*;
- ❑ Chamadas de sistemas são a **porta de entrada** para o modo *Kernel*;
  - São a interface entre os programas do usuário no modo usuário e o Sistema Operacional no modo *kernel*;
  - As chamadas diferem de SO para SO, no entanto, os conceitos relacionados às chamadas são similares independentemente do SO;

21

## Conceitos Básicos

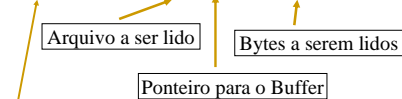
### Chamadas de Sistema

- ❑ **TRAP**: instrução que permite o acesso ao modo *kernel*;

#### ❑ Exemplo:

- Instrução do UNIX:

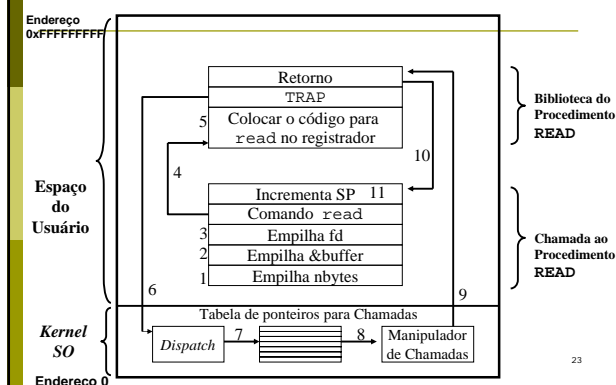
```
count = read(fd,buffer,nbytes);
```



O programa sempre deve checar o retorno da chamada de sistema para saber se algum erro ocorreu!!!

22

## Chamadas de Sistema



23

## Conceitos Básicos

### Chamadas de Sistema

#### ❑ Exemplos de chamadas de sistema:

- Chamadas para gerenciamento de processos:
  - ❑ Fork (CreateProcess - WIN32) – cria um processo;
- Chamadas para gerenciamento de diretórios:
  - ❑ Mount – monta um diretório;
- Chamadas para gerenciamento de arquivos:
  - ❑ Close (CloseHandle - WIN32) – fechar um arquivo;
- Outros tipos de chamadas:
  - ❑ Chmod: modifica permissões;

24

## Estrutura dos Sistemas Operacionais

### Principais tipos de estruturas:

- Monolíticos;
- Em camadas;
- Máquinas Virtuais;
- Arquitetura *Micro-kernel*;
- Cliente-Servidor;

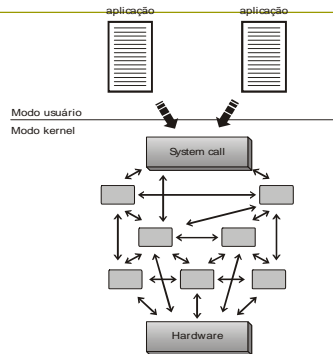
25

## Estrutura dos Sistemas Operacionais - Monolítico

- Todos os módulos do sistema são compilados individualmente e depois ligados uns aos outros em um único **arquivo-objeto**;
- O Sistema Operacional é um conjunto de processos que podem interagir entre si a qualquer momento sempre que necessário;
- Cada processo possui uma interface bem definida com relação aos parâmetros e resultados para facilitar a comunicação com os outros processos;
- Simples;
- Primeiros sistemas UNIX e MS-DOS;

26

## Estrutura dos Sistemas Operacionais - Monolítico



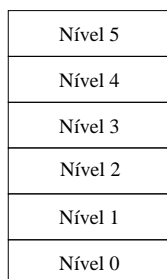
27

## Estrutura dos Sistemas Operacionais – Em camadas

- Possui uma hierarquia de níveis;
- Primeiro sistema em camadas: THE (idealizado por E.W. Dijkstra);
  - Possuía 6 camadas, cada qual com uma função diferente;
  - Sistema em *batch* simples;
- Vantagem: isolar as funções do sistema operacional, facilitando manutenção e depuração
- Desvantagem: cada nova camada implica uma mudança no modo de acesso
- Atualmente: modelo de 2 camadas

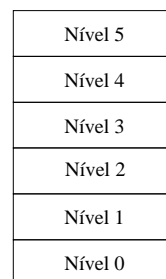
28

## Estrutura dos Sistemas Operacionais – Em camadas



29

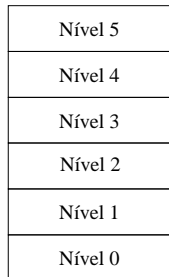
## Estrutura dos Sistemas Operacionais – Em camadas



- Alocação do processador;
- Chaveamento entre os processos em execução – multiprogramação;

30

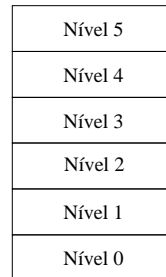
## Estrutura dos Sistemas Operacionais – Em camadas



- Gerenciamento da memória;
- Alocação de espaço para processos na memória e no disco:
  - Processo dividido em partes (páginas) para ficarem no disco;

31

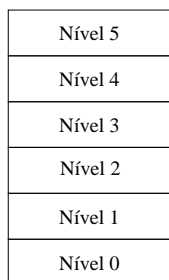
## Estrutura dos Sistemas Operacionais – Em camadas



- Comunicação entre os processos;

32

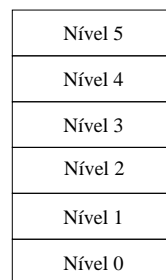
## Estrutura dos Sistemas Operacionais – Em camadas



- Gerenciamento dos dispositivos de entrada/saída – armazenamento de informações de/para tais dispositivos;

33

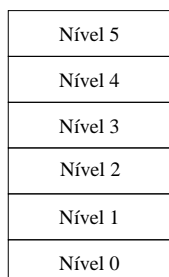
## Estrutura dos Sistemas Operacionais – Em camadas



- Programas dos usuários;
- Alto nível de abstração;

34

## Estrutura dos Sistemas Operacionais – Em camadas



- Processo do operador do sistema;

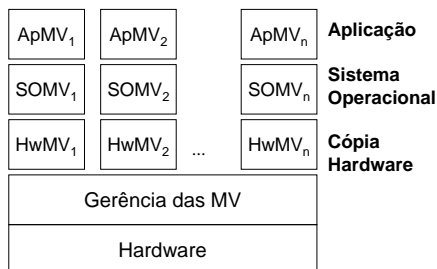
35

## Estrutura dos Sistemas Operacionais – Máquina Virtual

- ▣ Idéia em 1960 com a IBM → VM/370;
- ▣ Modelo de máquina virtual cria um nível intermediário entre o SO e o Hardware;
- ▣ Esse nível cria diversas **máquinas virtuais independentes e isoladas**, onde cada máquina oferece uma cópia virtual do hardware, incluindo modos de acesso, interrupções, dispositivos de E/S, etc.;
- ▣ Cada máquina virtual pode ter seu próprio SO;

36

## Estrutura dos Sistemas Operacionais – Máquina Virtual



37

## Estrutura dos Sistemas Operacionais – Máquina Virtual

Atualmente, a idéia de máquina virtual é utilizada em contextos diferentes:

- Programas MS-DOS: rodam em computadores 32bits;
  - As chamadas feitas pelo MS-DOS ao Sistema Operacional são realizadas e monitoradas pelo monitor da máquina virtual (VMM);
    - Virtual 8086;
- Programas JAVA (Máquina Virtual Java-JVM): o compilador Java produz código para a JVM (*bytecode*). Esse código é executado pelo interpretador Java;
  - Programas Java rodam em qualquer plataforma, independentemente do Sistema Operacional;

38

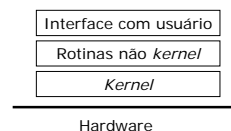
## Estrutura dos Sistemas Operacionais – Máquina Virtual

- Vantagens
  - Flexibilidade;
- Desvantagem:
  - Simular diversas máquinas virtuais não é uma tarefa simples → sobrecarga;

39

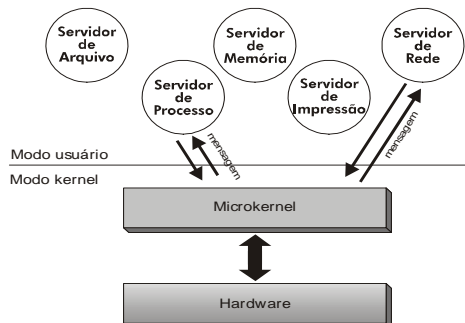
## Estrutura dos Sistemas Operacionais – Baseados em *Kernel* (núcleo)

- *Kernel* é o núcleo do Sistema Operacional
- Provê um conjunto de funcionalidades e serviços que suportam várias outras funcionalidades do SO
- O restante do SO é organizado em um conjunto de *rotinas não-kernel*



40

## Estrutura dos Sistemas Operacionais – *Micro-Kernel*



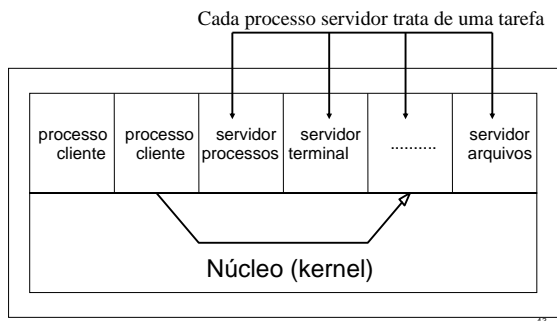
41

## Estrutura dos Sistemas Operacionais – Cliente/Servidor

- Reduzir o Sistema Operacional a um nível mais simples:
  - **Kernel**: implementa a comunicação entre processos clientes e processos servidores → Núcleo mínimo;
  - Maior parte do Sistema Operacional está implementado como processos de usuários (nível mais alto de abstração);
  - Sistemas Operacionais Modernos;

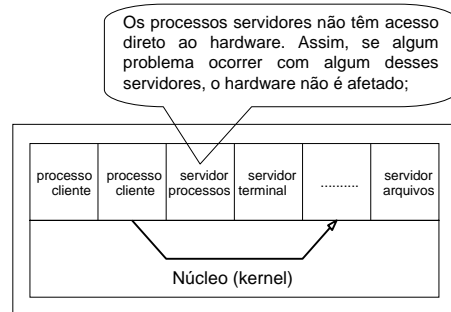
42

## Estrutura dos Sistemas Operacionais – Cliente/Servidor



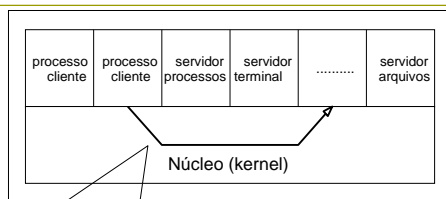
43

## Estrutura dos Sistemas Operacionais – Cliente/Servidor



44

## Estrutura dos Sistemas Operacionais – Cliente/Servidor

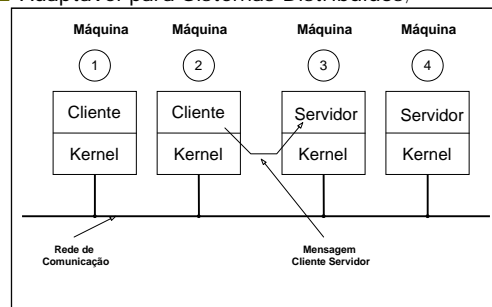


O mesmo não se aplica aos serviços que controlam os dispositivos de E/S, pois essa é uma tarefa difícil de ser realizada no modo usuário devido à limitação de endereçamento. Sendo assim, essa tarefa ainda é feita no *kernel*.

45

## Estrutura dos Sistemas Operacionais – Cliente/Servidor

### Adaptável para Sistemas Distribuídos;



46