# Sistemas Operacionais

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
kalinka@icmc.usp.br

Apresentação baseada nos slides
do Prof. Dr. Antônio Carlos Sementille e nas transparências fornecidas no site de compra do livro "Sistemas Operacionais Modernos"

# Mini Flash Back e Novidades

2

## Histórico dos Computadores

A evolução dos sistemas operacionais está relacionada ao desenvolvimento de equipamentos cada vez mais velozes, compactos e baratos, e à necessidade de aproveitamento e controle desses recursos.

- Geração Zero: 1642 - 1945
- 1ª Geração: 1945-1955
- 2 ª Geração: 1955-1965
- 3ªGeração: 1965-1980
- 4 ª Geração: 1980-1990
- 5 ª Geração: 1990

3

## Histórico dos Computadores

- Geração Zero: 1962 -1945
  - Blaise Pascal – Primeira máquina de calcular
  - Leibniz – Máquina que permitia realizar multiplicação e divisão
  - Computadores mecânicos (máquina diferencial, máquina analítica)
  - Máquinas baseadas em relés

4

## Histórico dos Computadores

- Geração Zero: 1962 -1945
  - Sem sistemas operacionais....

5

## Histórico dos Computadores

- 1ª Geração: 1945-1955
  - Válvulas
  - Único grupo
    - Projeto, construção, operação e manutenção
  - Utilizado para cálculos numéricos repetitivos
  - Máquina de Von Neumann

6

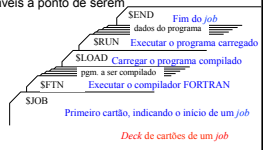
# Histórico dos Computadores

- 1ª Geração: 1945-1955
  - Nenhum SO
  - Instruções por fios e válvulas
  - Operador = Programador
  - Ex. ENIAC (***E**lectronic **N**umerical **I**ntegrator **A**nd **C**omputer*)

7

# Histórico dos Computadores

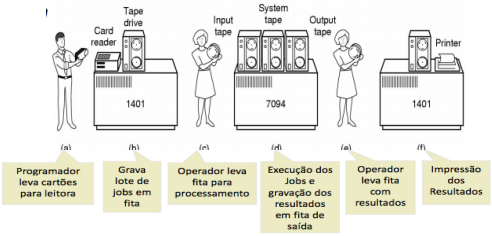
- 2 ª Geração: 1955-1965
  - Transistores
    - Os computadores tornaram-se confiáveis a ponto de serem comercializados
  - Alto custo
  - Batch
    - cartões perfurados
    - Reduzia o tempo desperdiçado
  - Fortran ou linguagem de montagem
  - Cálculos científicos




8

# Histórico dos Computadores

- 2 ª Geração: 1955-1965
  - Sistema em Batch (lote)




# Histórico dos Computadores

- 3ªGeração: 1965-1980
  - Diminuição dos custos de aquisição e utilização ***Processamento científico (palavra) X Processamento comercial (caractere);***
  - ***Aparecimento de máquinas mais compactas com compatibilidade de software (Sistema 360) e circuitos integrados , ainda caras com lentidão de operação (produtividade RH);***
  - ***Melhor utilização do hardware, ainda com dificuldade de depuração dos erros, utilização dos conceitos de gerenciamento e proteção de memória, interrupções, canais de E/S, buffers, spool, multiprogramação, Time-sharing;***

10

# Histórico dos Computadores

- 3ª Geração: 1965-1980

  ***Conceito de performance, jobs com execução rápida e estabelecimento de concorrência, com os Sistemas Operacionais rodando em várias máquinas, um terminal para cada usuário, dados on-line e com tempo de resposta razoável;***

11

# Histórico dos Computadores

- 3ªGeração: 1965-1980
  - CIs - Viabilização do uso dos sistemas computacionais
    - Diminuição dos custos de aquisição e utilização
  - Multiprogramação
    - A memória foi dividida para permitir que enquanto um programa esperasse por uma operação de leitura/gravação o processador executasse outro programa
    - Spooling (*Simultaneous Peripheral Operation On Line*)
      - **Capacidade de ler jobs de cartão direto para o disco**
      - **Alteração da ordem de execução das tarefas, até então sequencial**
  - *Time-sharing*
    - Cada programa utiliza o processador em pequenos intervalos de tempo

12

## Histórico dos Computadores

- 3ªGeração: 1965-1980
  - Multiprogramação
    - Aplicações que eram CPU-bound não tinham problema com relação ao tempo que precisava esperar para realizar E/S
    - Aplicações que eram IO-bound gastavam de 80 a 90% do tempo realizando E/S
      - Enquanto isso, a CPU ficava esperando parada
    - A multiprogramação divide a memória em diversas partes e aloca a cada uma dessas um *job*. Com isso mantem-se na memória, simultaneamente uma quantidade de *jobs* suficientes para ocupar 100% do tempo do processador, diminuindo a ociosidade. (Hardware é que protegia cada um dos *jobs* contra acessos indevidos de outros *jobs*).
  - Problema
    - Máquinas separadas para E/S e processamento e operadores andando entre as máquinas
  - Solução
    - *Spooling*
      - Assim que um *job* terminava, o SO já alocava o novo à uma partição livre da memória, direto do disco.

13

## Histórico dos Computadores

- 3ªGeração: 1965-1980
  - *Time Sharing*
    - Mesmo com o surgimento de novas tecnologias, o tempo de processamento ainda era crítico. Para corrigir um erro de programação, por exemplo, o programador poderia levar horas pois cada *job* era tratado dentro de um lote.
    - Com o *time sharing* cada usuário tem um terminal *on-line* à disposição e tem a sensação de possuir um computador apenas para ele.

14

## Histórico dos Computadores

- 4 ª Geração: 1980-1990
  - VLSI – *Very Large Scale Integration*
  - Primeiros computadores pessoais
  - PC, DOS, UNIX
  - Microcomputadores
  - Redes de computadores

15

## Histórico dos Computadores

- 4 ª Geração: 1980-1990
  - Cara máquina possuía um Sistema Operacional diferente – Sistemas Operacionais de Propósito geral
    - MULTICS
    - UNICS
    - DOS
    - MAC OS
    - WINDOWS

16

## Histórico dos Computadores

- 5 ª Geração: 1990
  - Processamento distribuído
    - Funções espalhadas por vários processadores através de redes de computadores
  - Novas interfaces
  - Linguagens
  - Comunicação

17

## Histórico dos Computadores

- 5 ª Geração: 1990
  - Sistemas Operacionais Distribuídos
    - Apresentam-se como um sistema operacional centralizado, mas que, na realidade, tem suas funções executadas por um conjunto de máquinas independentes.
  - Sistemas Operacionais de Redes
    - Usuários conhecem a localização dos recursos que estão utilizando e não têm uma visão de um sistema centralizado.

18

## Histórico dos Computadores

- 5 ª Geração: 1990
  - Sistemas Operacionais de Tempo Real
    - Gerenciamento de tempo.
    - Gerenciamento de processos críticos (aviões, caldeiras, carros, etc)
    - RTLinux (Real Time Linux);
      - http://www.fsmlabs.com/
  - Sistemas Operacionais para Sistemas Embarcados
    - Telefones, aparelhos eletrodomésticos, PDAs, Carros, ...

19

## Histórico dos Computadores

- Atualidades
  - Sistemas Operacionais Orientados a Objetos
    - Reúso
    - Interface orientada a objetos
  - JavaOS
    - Portabilidade;

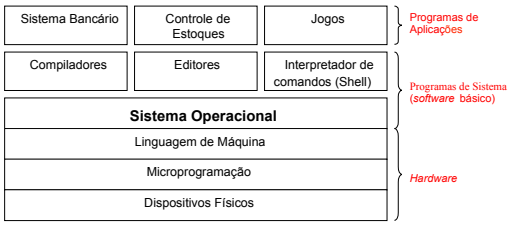
20

## O Que é um Sistema Operacional?

- Um programa que atua como um intermediário entre os usuários e o hardware [Silberschatz 1997].
- Um conjunto de programas que controla os recursos do computador e provê a base sobre a qual as aplicações são escritas [Tanenbaum 1992].
- Software responsável por gerenciar dispositivos que compõem um sistema computacional e realizar a interação entre o usuário e esses dispositivos.

21

## O Que é um Sistema Operacional?




Um sistema de computador composto de hardware, programas de sistemas e programas de aplicação

22

## Componentes de um Sistema

- **Dispositivos físicos**
  - São os chips de circuito integrado, fios, fontes de alimentação, tubos de raios catódicos, e assim por diante;
- **Micro arquitetura**
  - Dispositivos físicos agrupados para formar unidades funcionais
  - CPU – Processamento
  - ULA (Unidade Lógica Aritmética) – Operações aritméticas. Essas operações podem ser controladas por software (migro programas) ou por circuitos de hardware.

23

## Componentes de um Sistema

- **Microcódigo**
  - Software bastante primitivo, composto por um conjunto de microprogramas, usualmente gravados numa memória do tipo *read-only.*
  - É um interpretador que busca as instruções de máquina na memória principal (ADD, MOVE, JUMP ...), gerando o conjunto de sinais de controle necessários à execução de tais instruções de hardware;

24

## Componentes de um Sistema

- **Linguagem de máquina**
  - Conjunto de instruções que o microcódigo interpreta
    - Composta por um conjunto de 50 a 300 instruções
    - A maioria das quais movimenta dados dentro da máquina
    - Realiza operações aritméticas e compara valores
    - Baixo nível de abstração
    - Ex. Assembly
- **Sistema Operacional**
  - Esconde a complexidade do hardware
    - Oferece ao programador um conjunto de instruções mais convenientes para o desenvolvimento de seu trabalho.
  - É a porção de software que roda no modo *kernel* ou modo supervisor
    - Protege o hardware da ação direta do usuário final da máquina;

25

## Componentes de um Sistema

- **Programas do Sistema**
  - Ferramentas utilizadas para o desenvolvimento dos programas de Aplicação do Sistema
- **Programas de Aplicações**
  - Programas escritos pelos usuários para resolver problemas específicos
    - Processamento comercial de informações
    - Cálculo científicos
    - Jogos eletrônicos

26

## Sistema operacional x Programas do Sistema

- **Sistema operacional**
  - Roda em modo kernel (ou supervisor)
    - O programa tem acesso ao *hardware*
- **Programas do Sistema ou da Aplicação**
  - Rodam em modo usuário.
    - Tem acesso somente a determinadas regiões de memória
    - Não podem acessar dispositivos diretamente
    - Precisam pedir para o SO quando necessitam de alguma tarefa especial.
    - Garante que os programas dos usuários, não acabem por invadir áreas de memória do sistema operacional, e acabem por "travar" o sistema.
    - Possibilita que programas de diferentes usuários estejam rodando na mesma máquina, de forma que um usuário não consiga interferir nos programas de outro.

27

## Objetivos do SO

- Executar programas de usuário para solucionar seus problemas mais facilmente
- Tornar o computador conveniente ao uso
- Utilizar o hardware de maneira eficiente
- Compartilhar os recursos de um sistema computacional entre os vários usuários

28

## Serviços oferecidos pelo SO

- Facilidade para criação de programas
  - Editores, compiladores, depuradores e linkeditores
- Execução de programas
  - Carga e inicialização de arquivos e E/S
- Acesso controlado a arquivos
  - Formas e formatos específicos de E/S
- Acesso ao sistema
  - Proteção de acesso aos recursos e dados
  - Contenção no uso de recursos compartilhados
- Contabilização de uso
  - Estatísticas de uso e parâmetros de desempenho

29

## Serviços oferecidos pelo SO

- Detecção de erros
  - Erro de hardware
    - Erro de memória
    - Falha de dispositivo
  - Erro de software
    - Estouro aritmético
    - Acesso não permitido
  - Inabilidade do SO em garantir resposta
- Tratamento do erro
  - Simplesmente reporta o erro ao aplicativo
  - Repete a operação
  - Aborta a operação

30

# Vamos ao que interessa!!!

31

## Introdução

### Sistema Computacional

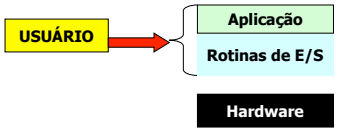
- **Consiste de:**
  - Um ou mais processadores
  - Memória principal
  - Discos, impressoras, teclado, monitor, interfaces de redes e outros dispositivos de entrada e saída

32

## Introdução

### A Importância do Sistema Operacional (S.0.)

- **Sistema sem S.O.**
  - Gasto maior de tempo de programação
  - Aumento da dificuldade
  - Usuário preocupado com detalhes de hardware

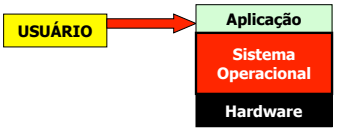



33

## Introdução

### A Importância do Sistema Operacional (S.0.)

- **Sistema com S.O.**
  - Maior racionalidade
  - Maior dedicação aos problemas de alto nível
  - Maior portabilidade




34

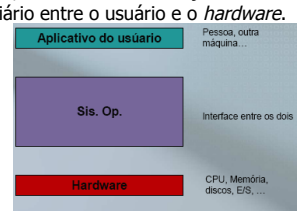
## Máquinas Multinível

35

## Introdução

### Definição de Sistema Operacional

Um sistema operacional é um programa, ou conjunto de programas, interrelacionados cuja finalidade é agir como intermediário entre o usuário e o *hardware*.




36

## Introdução

- O Sistema Operacional é uma interface HW/SW aplicativo
- Duas formas de vê-lo:
  - É um "fiscal" que controla os usuários
  - É um "juiz" que aloca os recursos entre os usuários
- Objetivos contraditórios:
  - Conveniência
  - Eficiência
  - Facilidade de evolução
  - A melhor escolha sempre DEPENDE de alguma coisa…

37

## Introdução

- Como "Juiz"
  - Como **máquina estendida** (top-down): tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário.
- Como "Fiscal"
  - Como **gerenciador de recursos** (bottom-up): gerenciar os dispositivos que compõem o computador.

38

## Introdução

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

39

## Introdução

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

40

## Introdução

- Possui várias funções, entre elas:
  - apresentar uma máquina mais flexível;
  - permitir o uso eficiente e controlado dos componentes de *hardware*;
  - permitir o uso compartilhado e protegido dos diversos componentes de *hardware* e *software*, por diversos usuários.

41

## Introdução

- O Sis. Op. deve fornecer uma interface aos programas do usuário
  - Quais recursos de HW?
  - Qual seu uso?
  - Tem algum problema? (Segurança, falha…?)
  - É preciso manutenção?
  - Chegou um email?
  - Entre outros…
  - Chamadas de sistema – programas de sistema

42

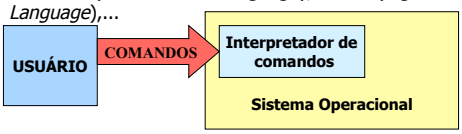
# Introdução

## Interação com o Sistema Operacional

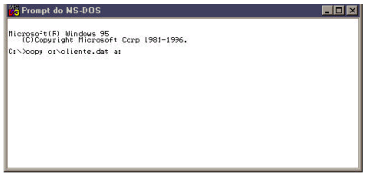
**O USUÁRIO**

Interage com o S.O. de maneira direta, através de comandos pertencentes à uma linguagem de comunicação especial, chamada "linguagem de comando".

Ex: JCL (*Job Control Language*), DCL (*Digital Control Language*),...

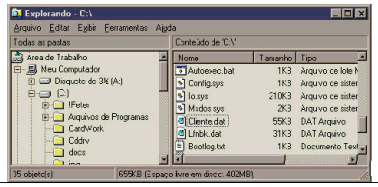



43



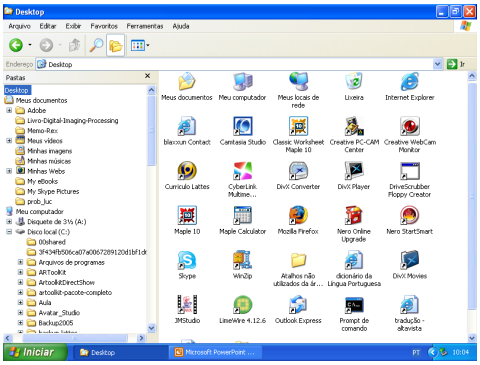

Interface Texto

Interface em Modo Texto (Linha de Comando)



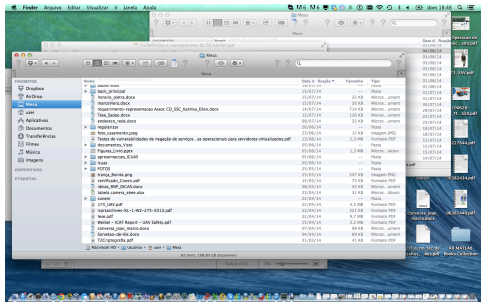

Interface Gráfica (GUI)

44




Windows XP

45



Mac

46

# Introdução

## Interação com o Sistema Operacional

**OS PROGRAMAS DE USUÁRIO**

Invocam os serviços do S.O. por meio das "chamadas ao sistema operacional".

47

# Introdução

## A Evolução dos Sistemas Operacionais

- O alcance e a extensão dos serviços de um S.O: dependem das necessidades e características do ambiente que deve suportar.
- Um S.O. pode processar sua carga de trabalho (*workload*) de duas formas:
  - **Serial:** neste, os recursos são dedicados à um único programa, até o seu término.
  - **Concorrente:** os recursos são dinamicamente re-associados entre uma coleção de programas ativos, em diferentes estágios de execução.

48

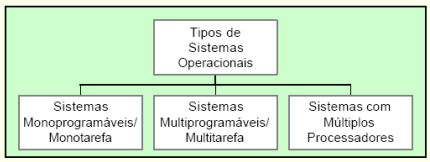
# Introdução

## Tipos de Sistemas Operacionais

- Classificação quanto ao compartilhamento de hardware
  - **Sistemas Operacionais Monoprogramados**
    - Só permite um programa ativo em um dado período de tempo, o qual permanece na memória até seu término
    - Ex: DOS
  - **Sistemas Operacionais Multiprogramados**
    - Mantém mais de um programa simultaneamente na memória principal, para permitir o compartilhamento efetivo do tempo de UCP e demais recursos
    - EX: Unix, VMS, Windows NT, etc.

49

# Introdução




- **SOs Monoprogramáveis ou Monotarefa**
  - *Se caracterizam por permitir que o processador, a memória e os periféricos permaneçam exclusivamente dedicados à execução de um único programa. Recursos são mal utilizados, entretanto é fácil de ser implementado.*

50

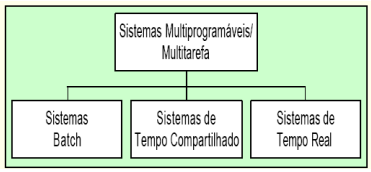
# Introdução

- **SOs Multiprogramáveis ou Multitarefa**
- *Nestes SOs vários programas dividem os recursos do sistema. As vantagens do uso destes sistemas são o aumento da produtividade dos seus usuários e a redução de custos, a partir do compartilhamento dos diversos recursos do sistema.*
- *Podem ser* **Multiusuário** *(mainframes, mini e microcomputadores) ou* **Monousuário** *(PCs e estações de trabalho). É possível que ele execute diversas tarefas concorrentemente ou mesmo simultaneamente (***Multiprocessamento***) o que caracterizou o surgimento dos SOs* ***Multitarefa****.*

51

# Introdução

- *Os SOs* **Multiprogramáveis/Multitarefa** *podem ser classificados pela forma com que suas aplicações são gerenciadas, podendo ser divididos conforme mostra o gráfico.*




52

# Introdução

- **Classificação quanto a interação permitida**
  - fator determinante ⇨ Tempo de resposta

53

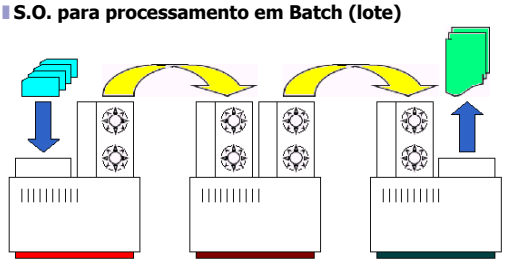
# Introdução

- **Classificação quanto a interação permitida**
  - fator determinante ⇨ Tempo de resposta
  - S.O. para processamento em Batch (lote)
    - Os *jobs* dos usuários são submetidos em ordem sequencial para a execução
    - Não existe interação entre o usuário e o *job* durante sua execução

54

# Introdução

- **S.O. para processamento em Batch (lote)**

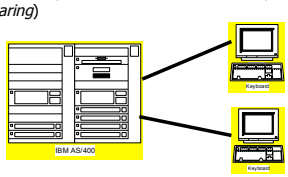


Sistema Batch (processamento em lote)

55

# Introdução

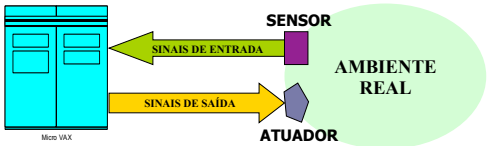
- **S.O. Interativo**
  - O sistema permite que os usuários interajam com suas computações na forma de diálogo
  - Podem ser projetados como sistemas mono-usuários ou multi-usuários (usando conceitos de multiprogramação e *time-sharing*)




56

# Introdução

- **S.O. de Tempo Real**
  - Usados para servir aplicações que atendem processos externos, e que possuem tempos de resposta limitados
  - Geralmente sinais de interrupções comandam a atenção do sistema
  - Geralmente são projetados para uma aplicação específica




57

# Introdução

- **Classificação segundo o Porte (Tanenbaum, 2003)**
  - S.O.s de Computadores de grande porte
  - S.O.s de Servidores
  - S.O.s de Multiprocessadores
  - S.O.s de Computadores Pessoais
  - S.O.s de Tempo Real
  - S.O.s embarcados
  - S.O.s de cartões inteligentes

58

# Introdução

## Diferentes Visões de um S.O.

- **Visão do Usuário da Linguagem de Comando**
  - As linguagens de comando são específicas de cada sistema

| Classe Funcional | Operações Típicas |
|---|---|
| Ativação de Programa e Controle | Carregar (Load)<br>Executar (Run)<br>Abortar (abort)<br>Destruir processo (kill) |
| Gerência de Arquivos | Copiar (Copy, cp,...)<br>Renomear (Ren)<br>Listar diretório (Dir, ls,...) |
| ... | ... |

59

# Introdução

## Diferentes Visões de um S.O.

- **Visão do Usuário das Chamadas do Sistema**
  - Permitem um controle mais eficiente sobre as operações do sistema e um acesso mais direto sobre as operações de hardware (especialmente a E/S).

| Tipos Principais de Chamadas |
|---|
| Iniciação de dispositivos<br>Execução e controle de programas<br>Serviços de alocação e reserva de recursos do sistema (ex: memória)<br>Comunicação com dispositivos de E/S, etc. |

60

# Introdução

## Estrutura de Sistemas Operacionais

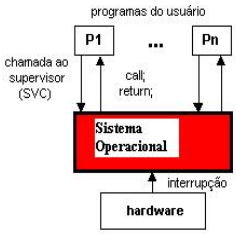
- Como os sistemas operacionais são normalmente grandes e complexas coleções de rotinas de software, os projetistas devem dar grande ênfase à sua organização interna e estrutura

61

# Introdução

**Estrutura Monolítica**

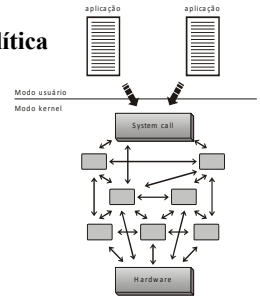
- É a forma mais primitiva de S.O.
- Consiste de um conjunto de programas que executam sobre o hardware, como se fosse um único programa.
- Os programas de usuário podem ser vistos como sub-rotinas, invocadas pelo S.O., quando este não está executando nenhuma das funções do sistema




62

# Introdução

**Estrutura do Monolítica**




63

# Introdução

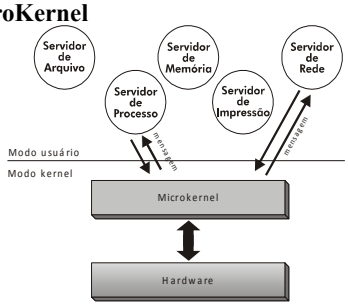
**Estrutura do MicroKernel**

- MicroNúcleo (microkernel): incorpora somente as funções de baixo nível mais vitais
- O microkernel fornece uma base sobre a qual é construído o resto do S.O.
- A maioria destes sistemas são construídos como coleções de processos concorrentes
- Fornece serviços de alocação de UCP e de comunicação aos processos

64

# Introdução

**Estrutura do MicroKernel**




65

# Introdução

**Sistemas de Camadas -**
**Estrutura Hierárquica de Níveis de Abstração**

Os princípios utilizados nesta abordagem são:

- **Modularização: divisão de um programa complexo em módulos de menor complexidade. Os módulos interagem através de interfaces bem definidas.**
- **Conceito de "Informação Escondida": os detalhes das estruturas de dados e algoritmos são confinados em módulos. Externamente, um módulo é conhecido por executar uma função específica sobre objetos de determinado tipo.**

66

# Introdução

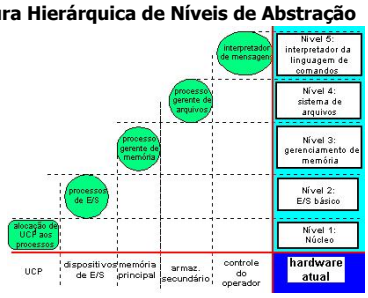
**Estrutura Hierárquica de Níveis de Abstração**

- A ideia básica é criar um S.O. como uma hierarquia de níveis de abstração, de modo que, a cada nível, os detalhes de operação dos níveis inferiores possam ser ignorados. Assim, cada nível pode confiar nos objetos e operações fornecidas pelos níveis inferiores.

67

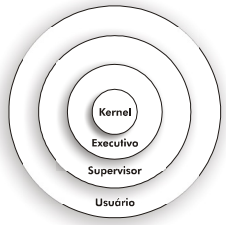
# Introdução

**Estrutura Hierárquica de Níveis de Abstração**




68

- Ex: Arquitetura do OpenVMS



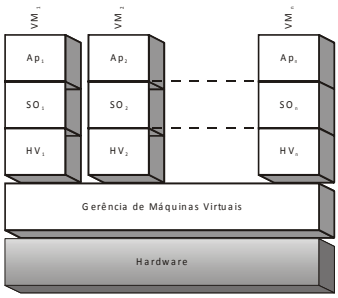

69

## Máquina virtual

- O Modelo de Máquina Virtual ou *Virtual Machine* (VM), cria um nível intermediário entre o hardware e o S.O., denominado Gerência de Máquinas Virtuais.
- Este nível cria diversas máquinas virtuais independentes, onde cada uma oferece uma cópia virtual do hardware, incluindo modos de acesso, interrupções, dispositivos de E/S, etc.
- Como cada VM é independente das demais, é possível que tenha seu próprio S.O.

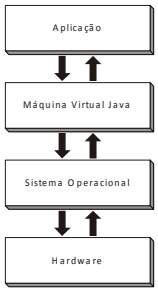
70

# Máquina virtual




71

Um outro exemplo de utilização desta estrutura ocorre na linguagem Java. Para executar um programa Java é necessário uma máquina virtual Java (*Java Virtual Machine* – JVM)




72