

Sistemas Operacionais I

Introdução aos SOs

Prof. Robson Siscoutto

e-mail: robson@unoeste.br

Conteúdo Programático

- **Conceitos Básicos**
 - Evolução histórica
 - Funções e objetivos de um sistema Operacional
 - Tipos de sistemas operacionais
 - Arquiteturas de sistema operacionais
- **Processos**
 - Conceitos
 - Classificação
 - Estados do processo
 - Processos no Unix/Windows
 - Criação e término
 - Hierarquia
 - Chamadas de sistema POSIX para gerenciamento de processos
 - Implementação de aplicações multiprocessos

Conteúdo Programático

- **Escalonamento de Processos**
 - Conceitos
 - Classes de escalonamento
 - Algoritmos de escalonamento
 - Não preemptivos
 - Preemptivos
 - Transições de estados
- **Concorrência**
 - Exclusão mútua
 - Condições de disputa
 - Região crítica
 - Exclusão mútua com espera ociosa
 - { Dormir e acordar
 - Semáforos
 - Mutexes
 - Monitores
 - Troca de mensagens
 - Deadlocks
 - Ocorrência
 - Prevenção
 - Detecção
 - Recuperação

Conteúdo Programático

- Threads
 - Introdução e conceitos
 - Conteúdo e funcionamento
 - Tipos de threads
 - Prática de Threads
 - Criação
 - Controle de concorrência
 - Troca de informações
 - Implementação de aplicações multithreaded

Sistemas Operacionais

Introdução

Histórico dos Sistemas Operacionais

Sistemas Operacionais

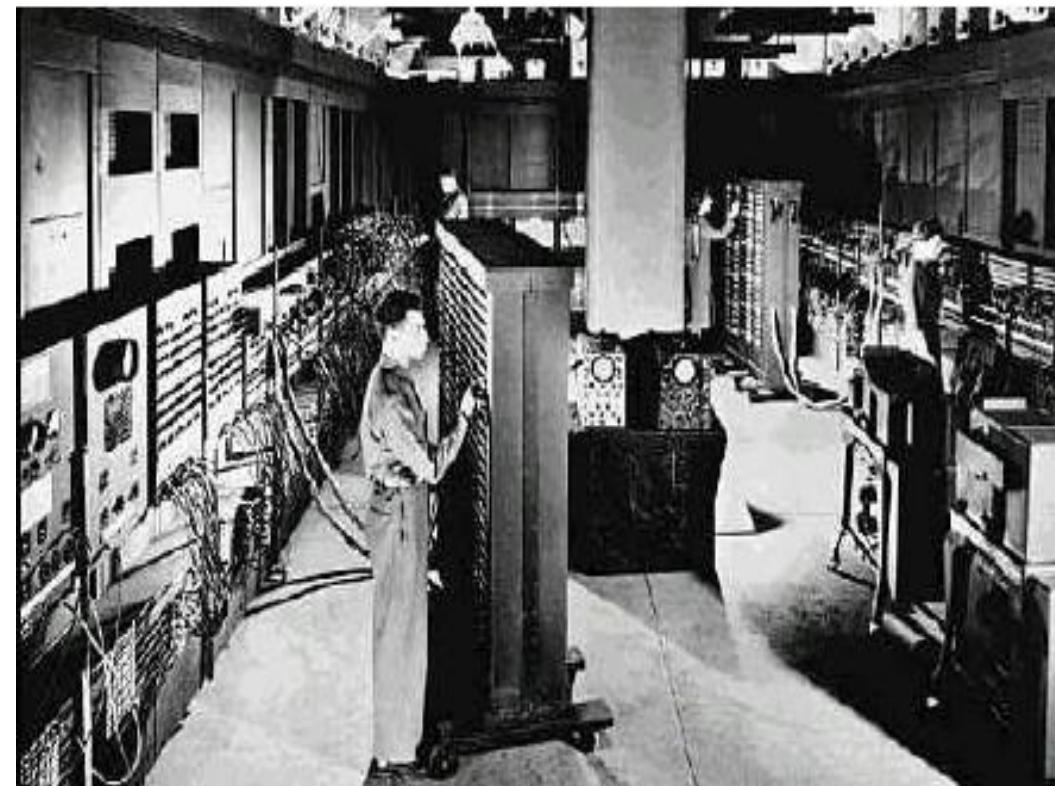
Introdução

- Primeira Geração (1945 – 1955): Válvulas e Painéis
 - Computadores baseados em válvulas (+- 20.000);
 - Ocupavam enormes áreas;
 - Funcionamento lento e duvidoso;
 - Surge o primeiro computador digital (ENIAC)
 - Realizava cálculos balísticos, possuía 18 mil válvulas, 10 mil capacitores, 70 mil resistores e pesava 30 toneladas;
 - Dificuldades:
 - Conhecer profundamente o hardware;
 - Programação feita através de painéis – fios;
 - Utilizava-se linguagem de máquina;
 - Primeiro S.O. – desenvolvido pela GM Lab. Para IBM 701(Lote)

Sistemas Operacionais

Introdução

- Primeira Geração (1945 – 1955): Válvulas e Painéis

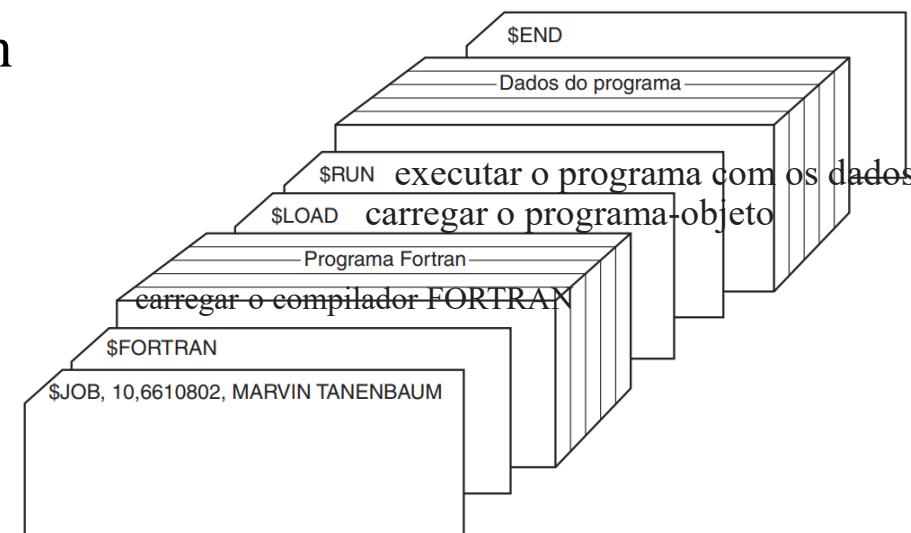


Sistemas Operacionais

Introdução

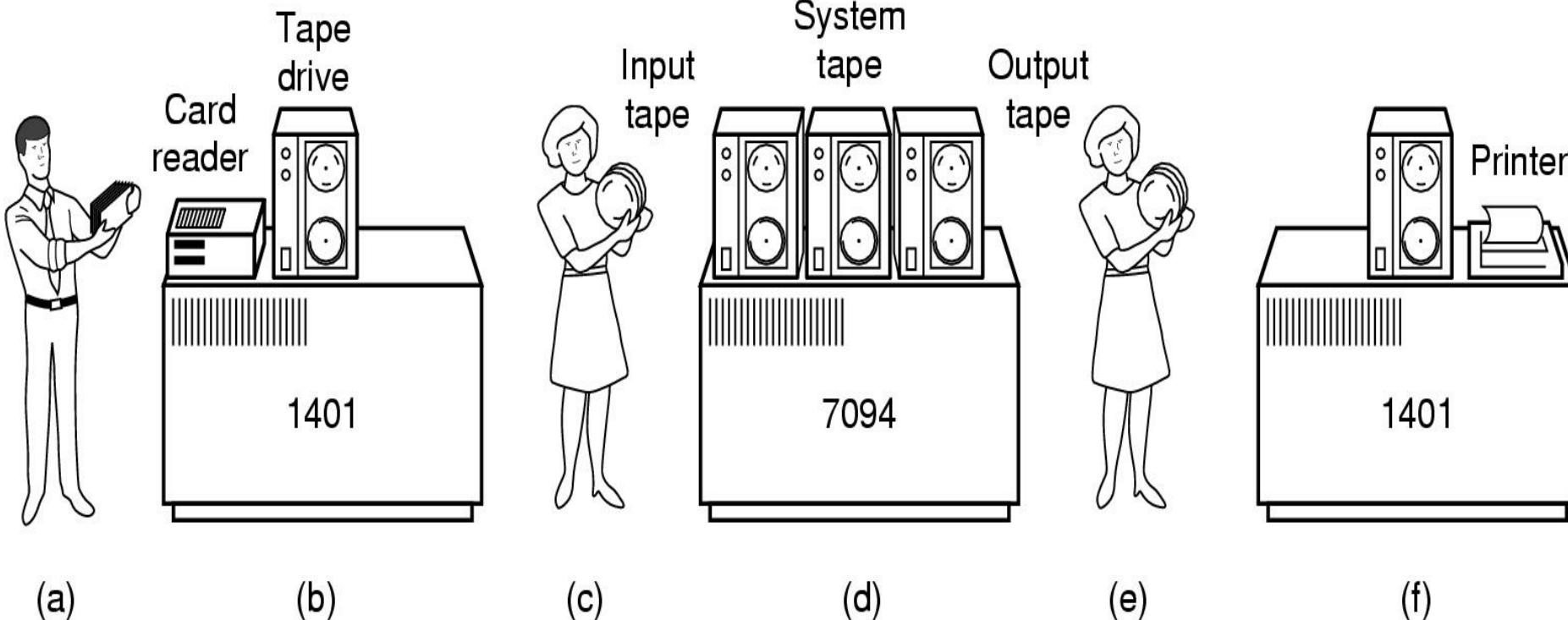
- Segunda Geração (1955 - 1965): transistores e Sist. Lote

- Surgimento dos transistores
 - Aumento da velocidade e mais confiabilidade no processamento;
 - Surgimento das Memórias Magnéticas
 - Acesso mais rápido aos dados, maior capacidade de armazenamento e computadores menores;
 - Surgimento das ling. de programação Assembly e Fortran
 - Surgimento dos Cartões perfurados
 - Utilizados para escrever os programas
 - Um programa por vez processado
 - Surgimento do conceito de Sistema em Lote (Batch)
 - Eram lidos mais que um programa por vez;
 - Um programa controlava os intervalos entre cada programa;
 - Sistemas operacionais típicos eram o FMS (o Fortran Monitor System) e o IBSYS, sistema operacional da IBM para o 7094



Sistemas Operacionais

Introdução



Sistemas Operacionais

Introdução

- **Terceira Geração (1966 - 1980): CI's e Multiprog.**

- Circuitos integrados e microprocessadores
 - Viabilizou e difundiu o uso de sistemas computacionais com menor custos;
 - Aumento do poder de processamento e diminuição no tamanho dos equipa;
- Multiprogramação
 - Executar outro programa no processador quando o que estava executando ficar bloqueado;
 - Dividir a memória em partições, onde cada programa ocupava uma partição
 - Alterar a ordem de execução das tarefas, até então sequencial;
- Surgimento dos Monitores de vídeo e teclados;
- Conceito de Tempo compartilhado (Time Sharing)
 - Programas utilizam o processador em pequenos intervalos de tempo;
- Surgimento do primeiro computador de oito bits (Em 1974, a Intel apareceu com o microprocessador 8080)
 - Apple (11 de abril de 1976);
- S.O. UNIX (versão monousuário simplificada do MULTICS - Ken Thompson - minicomputador PDP-7)

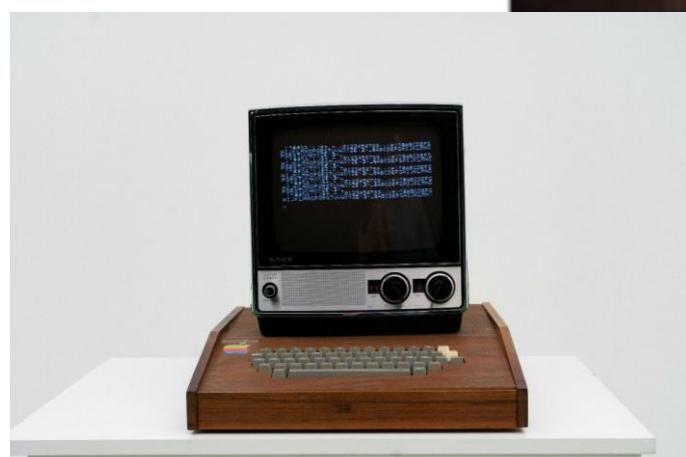
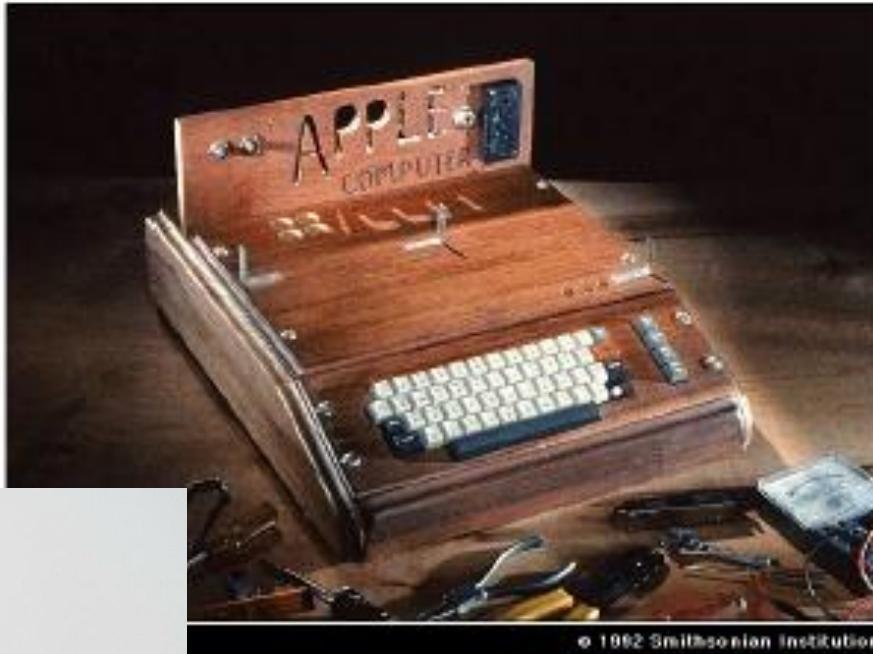


Sistemas Operacionais

Introdução

- Apple 1

Baseado no microprocessador MOS 6502 de 8 bits com 1 MHz de potência. A memória padrão possui 4 KB, mas o usuário ainda pode fazer um upgrade para 8 KB ou 48 KB.

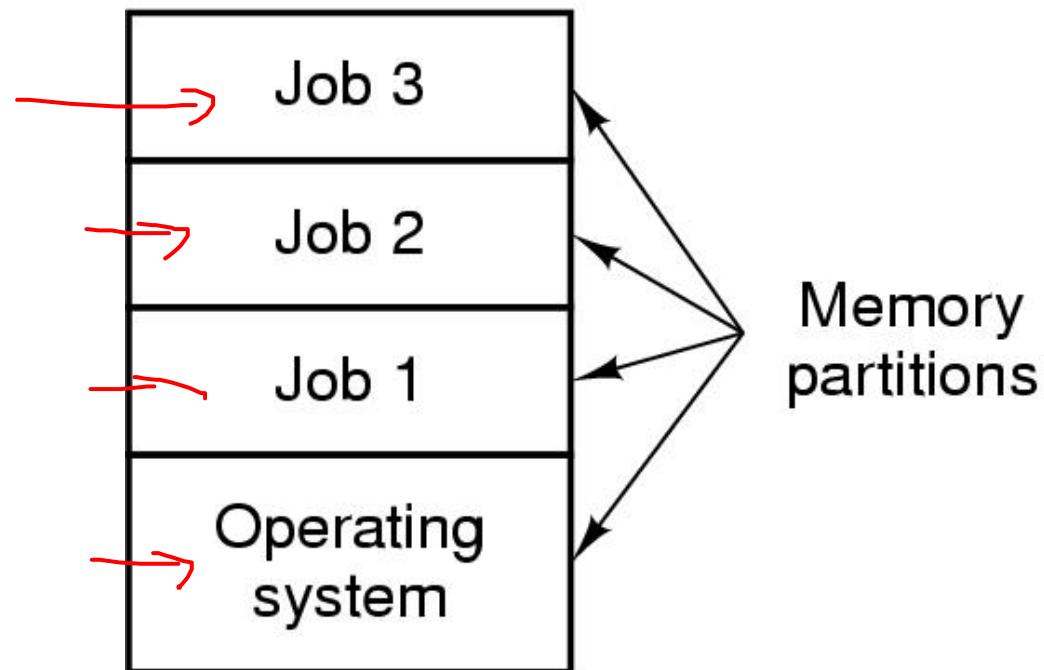


placa-mãe, com cerca de 60 chips,
totalmente montada;
Foi montado pelo engenheiro eletrônico
Steve Wozniak

Sistemas Operacionais

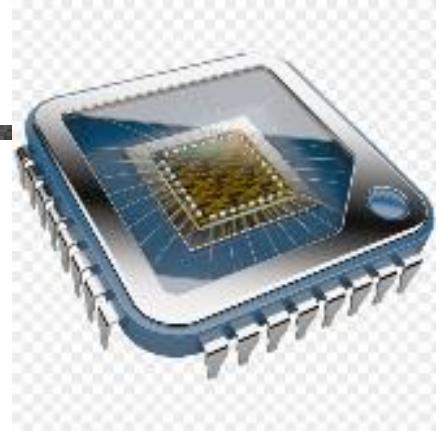
Introdução

- Sistema Multiprogramado
 - Três JOBS na Memória



Sistemas Operacionais

Introdução



- **Quarta Geração (1981 - 1990): Computadores Pessoais**

- Desenvolvimento da LSI (Integração de circuitos em grande escala) e VLSI
- Surgimento dos Chips
 - Milhares de transistores encapsulados em 1 centímetro de silício;
- sistema operacional CP/M (Control Program for Microcomputers), de uma empresa chamada Digital Research;
- Surgimento do primeiro PC de 16 bits – IBM - IBM PC utilizando o 8088 da Intel;
- Surgimento do S.O. DOS (Disk Operating System);
- O CP/M, o MS-DOS e o Apple DOS eram todos sistemas de linha de comando;
 - Doug Engelbart, do Stanford Research Institute, tinha inventado a **GUI (Graphical User Interface)**
 - Macintosh da Apple foi anunciado no início de 1984 ;
 - 2001, a Apple lança o **Mac OS X** (nova versão da GUI Macintosh sobre o UNIX)
- Surgimento de equipamentos com múltiplos processadores;
- Surgimento das redes de computadores;
 - Cliente/Servidor e Sistemas distribuídos;

Sistemas Operacionais

Introdução

- Quinta Geração (1991 -)
 - Grandes avanços em termos de Hardware, software e telecomunicações;
 - Devido a necessidade de armazenamento e processamento das aplicações;
 - Arquiteturas paralelas e processamento distribuídos;
 - Novas linguagens específicas para este tipo de arquit;
 - » Uso de inteligência artificial e CAD (Computer-Aided Design);
 - » Processamento de imagens (3D, Realidade Virtual, CAVE);
 - Evolução da microeletrônica:
 - VLSI (Very Large Scale Integration) p/ ULSI (Ultra Large Scale Integration);
 - Consolidação das interfaces gráficas para S.O.
 - Novos tipos de interfaces estão surgindo:
 - Imagens, sons, interfaces dinâmicas etc..

Sistemas Operacionais

Introdução

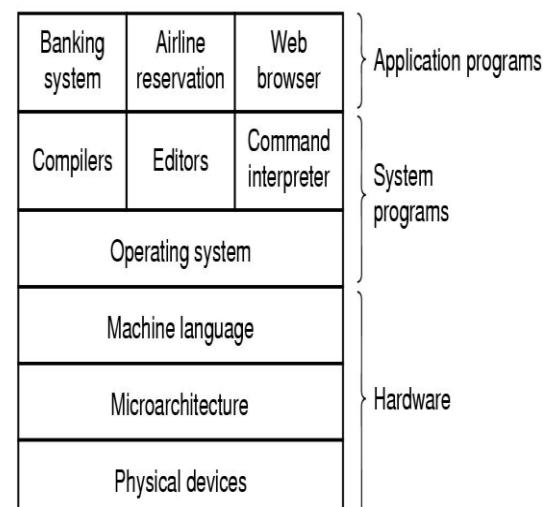
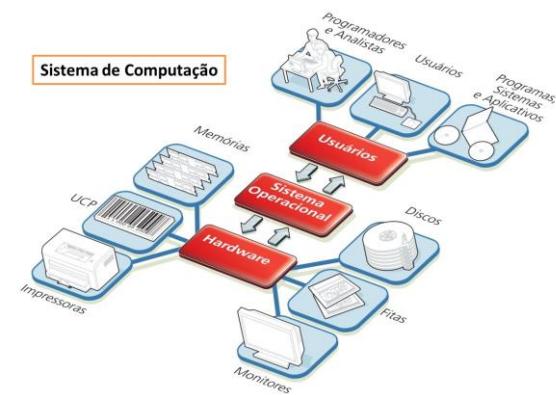
- **O que é um Sistema Operacional?**
 - Conjunto de programas que atuam como um intermediário entre o usuário e o hardware;
 - Seu propósito é prover um ambiente na qual o usuário possa executar seus programas, de maneira fácil e eficiente;
- **Funções Principais:**
 - Apresentação ao usuário de uma máquina mais flexível e adequada para se programar do que aquela que o hardware lhe apresenta;
 - Possibilitar o uso eficiente e controlado dos vários componentes de hardware e software do sistema;
 - Possibilitar a diversos usuários o uso compartilhado e protegido dos diversos componentes do sistema;

Introdução

Um sistema de computação se divide em:

- **Programas de aplicação** (criado pelo usuário);
 - Jogos, Sistemas Bancários etc
- **Programas do Sistema (Camada Intermediária)**;
 - Compiladores, editores, interpretadores de comandos, etc.
 - Sistema Operacional (Núcleo)
- **Hardware**;
 - Linguagem de Máquina (50 a 300 instruções);
 - Dispositivos de E/S são controlados através de registradores;
 - Microprograma (software localizado em memória ROM)
 - Interpretador que traduz linguagem de máquina (ADD, MOVE) para sinais de controle para o hardware;
 - Dispositivos físicos (parte elétrica - engenheiros elétricos)

Gerenciador de Recursos:
Controlar quem está usando qual recurso, garantir os pedidos de recursos, medir a utilização e mediar pedidos conflitantes de diferentes programas e usuários.

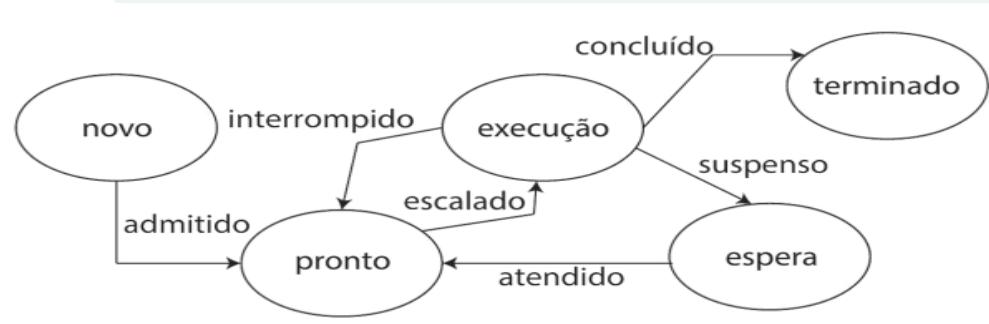


CONCEITOS DE SISTEMA OPERACIONAL

• Tarefas do sistema operacional:

Gerenciamento de Processos:

Círculo de Vida e estados de um Processo:



Gerenciamento de arquivos

Organização das estruturas de arquivos

- GNU/Linux e macOS X:
 - derivados do Unix a partir de um diretório raiz
- Windows: unidades de armazenamentos são lidas como letras (A:\, B:\, C:\, D:\, E:\, etc.)

Gerenciamento de memória Memoria Virtual

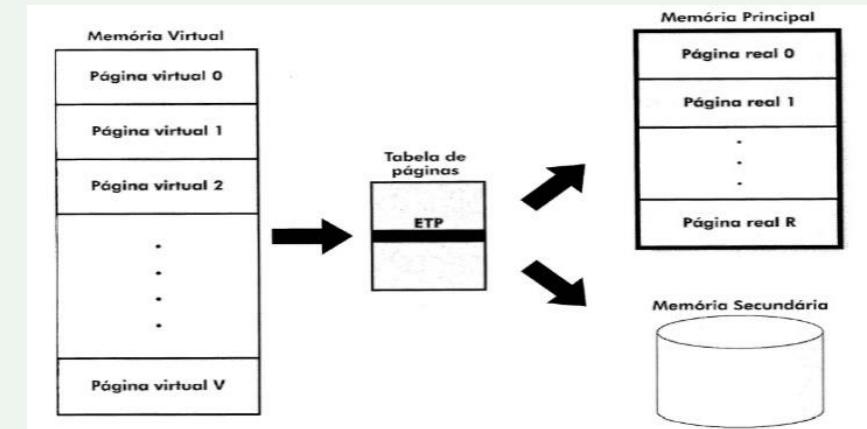


Figura 6: Tabela de páginas

Gerenciamento de E/S

- gerenciar os dispositivos de Entrada e Saída (E/S) ligados ao computador

Sistemas Operacionais

Introdução

Conceitos Básicos sobre Sistemas Operacionais

Sistemas Operacionais

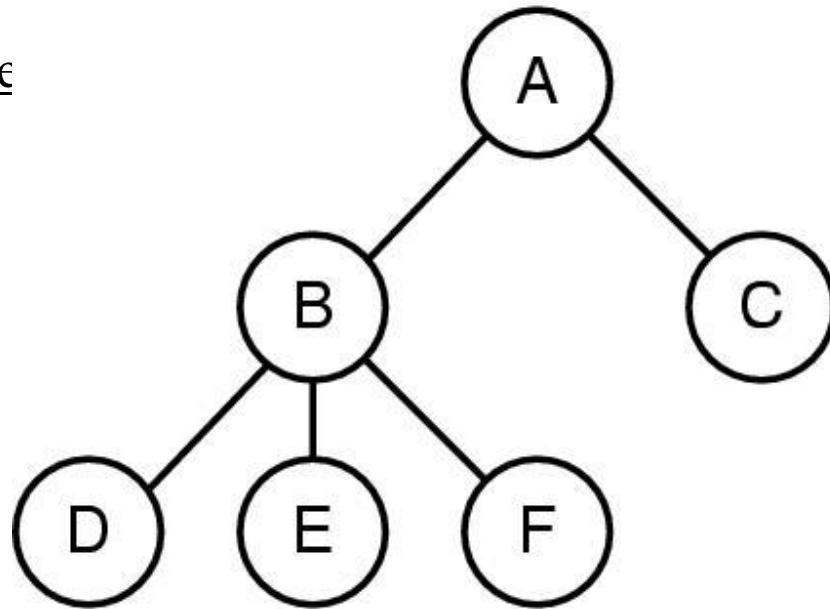
Introdução

- **Processos:**
 - Programa em execução, sendo constituído de:
 - Código executável (área de texto – código do programa)
 - Dados referentes ao código (área de dados - variáveis alocadas)
 - Pilha de execução,
 - Contador de programas (PC - registrador),
 - Valor do apontador de pilha (SP - registrador),
 - Valores dos demais registradores do hardware,
 - Conjunto de outras informações necessárias a execução do programa.
 - Conceito de processo é visível principalmente em sistemas de tempo compartilhado (troca de processos devido a término do tempo no proc.);
 - Conceito de Tabela de Processos ou PCB (Bloco de Controle de Processo)
 - Estrutura que contem os valores dos registradores do processo que deverão ser utilizados quando o processo for restaurado;

Sistemas Operacionais

Introdução

- **Processos: (Cont.)**
 - Conceito de árvore de processos



- Uma árvore de processos:
 - A cria dois processos filhos, B e C
 - B cria três processos filhos, D, E e F

Sistemas Operacionais

Introdução

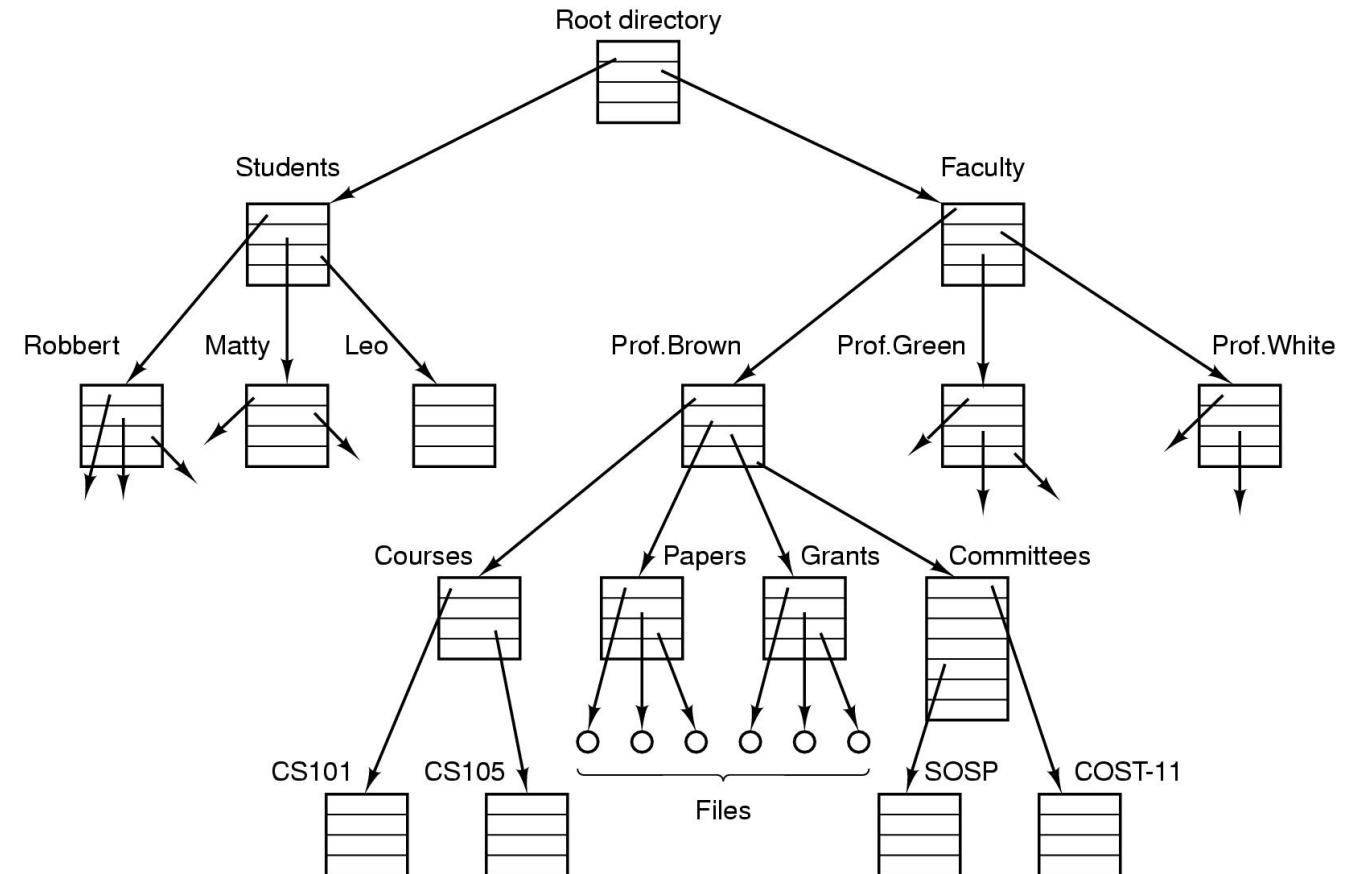
- **Processos: (Cont.)**

- Comunicação entre processos:
 - Através de primitivas de comunicação p.ex: send e receive;
 - Utilização de Sockets, RPC etc;
- Em sistemas operacionais onde há vários usuários ao mesmo tempo trabalhando, (p.ex: Linux ou Unix):
 - Cada processo possui um identificador único chamado de **UID (identificação Usuário dono)** de 16 ou 32 bits;
 - Quando os usuários são divididos em grupos, o **identificador de grupo** é conhecido como **GID**;
 - **PID** (Processo Identification) – Identificador do Processo
 - UID e GID úteis para proteção no sistema.

Sistemas Operacionais

Introdução

- Arquivos? e Diretórios?



Arquivo: conjunto de dados relacionados que podem ou não ter uma estrutura interna definida;

Diretório: unidade organizacional

Sistemas Operacionais

Introdução

- Arquivos: (Cont.)

- Proteção de Diretórios ou Arquivos

- » Linux: 10 bits ((L D -) RWX RWX RWX);

- » Windows: Controle de acesso através de grupos e compartilhamento;

```
robson@RAS-NOTEG5:~$ ls -l
total 696
drwxr-xr-x 2 robson robson 4096 Nov 25 20:48 A0
-rw-r--r-- 1 robson robson 139 Nov 11 20:46 ao.txt
-rwxr-xr-x 1 robson robson 281479 Mar  8 2021 CódigoSlides.zip
```

Nomes de grupo ou de usuário:

- usuários autenticados
- SISTEMA
- Administradores (RAS-NOTEG5\Administradores)
- usuários (RAS-NOTEG5\Usuários)

Para alterar permissões, clique em Editar.

| Permissões para Usuários autenticados | Permitir |
|---------------------------------------|----------|
| Controle total | ✓ |
| Modificar | ✓ |
| Ler & executar | ✓ |
| Listar conteúdo da pasta | ✓ |
| Leitura | ✓ |
| Gravar | ✓ |

- Descriptor de arquivos (fd)

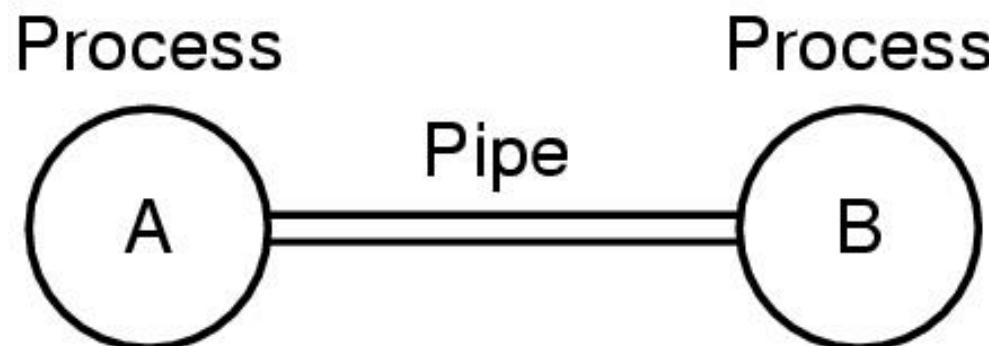
- Sempre que uma arquivo é aberto o S.O. retorna um valor inteiro que deve ser utilizado para manipular o arquivo;

```
5 int fd, sz;
6 char *c = (char *) calloc(100, sizeof(char));
7
8 fd = open("foo.txt", O_RDONLY);
9 if (fd < 0) { perror("r1"); exit(1); }
```

Sistemas Operacionais

Introdução

- Arquivos: (Cont.)
 - Conceito de PIPE
 - Pseudo-arquivo que pode ser usado para conectar dois processos;
 - Funcionamento:
 - » Processo A escreve no PIPE (como se fosse um arquivo de saída);
 - » Processo B lê do PIPE (como se fosse um arquivo de entrada);



Sistemas Operacionais

Introdução

- **Chamadas de Sistema:**
 - Interface entre um processo e o sistema operacional;
 - Essas chamadas são geralmente disponíveis em instruções ling. assembly
 - Algumas linguagens de alto nível disponibilizam tais chamadas:
 - C, C++, PERL, BCPL, PL etc.
 - Exemplo de chamada de sistema (UNIX)
 - Count = Read (file, buffer, nbytes);
 - Chamadas de sistemas são agrupadas em 4 categorias:
 - Controle de processos;
 - Manipulação de arquivos;
 - Manutenção de arquivos e
 - Comunicações;

Sistemas Operacionais

Introdução

Process management

| Call | Description |
|---------------------------------------|------------------------------------------------|
| pid = fork() | Create a child process identical to the parent |
| pid = waitpid(pid, &statloc, options) | Wait for a child to terminate |
| s = execve(name, argv, environp) | Replace a process' core image |
| exit(status) | Terminate process execution and return status |

File management

| Call | Description |
|--------------------------------------|------------------------------------------|
| fd = open(file, how, ...) | Open a file for reading, writing or both |
| s = close(fd) | Close an open file |
| n = read(fd, buffer, nbytes) | Read data from a file into a buffer |
| n = write(fd, buffer, nbytes) | Write data from a buffer into a file |
| position = lseek(fd, offset, whence) | Move the file pointer |
| s = stat(name, &buf) | Get a file's status information |

Sistemas Operacionais

Introdução

Directory and file system management

| Call | Description |
|--------------------------------|----------------------------------------------|
| s = mkdir(name, mode) | Create a new directory |
| s = rmdir(name) | Remove an empty directory |
| s = link(name1, name2) | Create a new entry, name2, pointing to name1 |
| s = unlink(name) | Remove a directory entry |
| s = mount(special, name, flag) | Mount a file system |
| s = umount(special) | Unmount a file system |

Miscellaneous

| Call | Description |
|--------------------------|-----------------------------------------|
| s = chdir(dirname) | Change the working directory |
| s = chmod(name, mode) | Change a file's protection bits |
| s = kill(pid, signal) | Send a signal to a process |
| seconds = time(&seconds) | Get the elapsed time since Jan. 1, 1970 |

Sistemas Operacionais

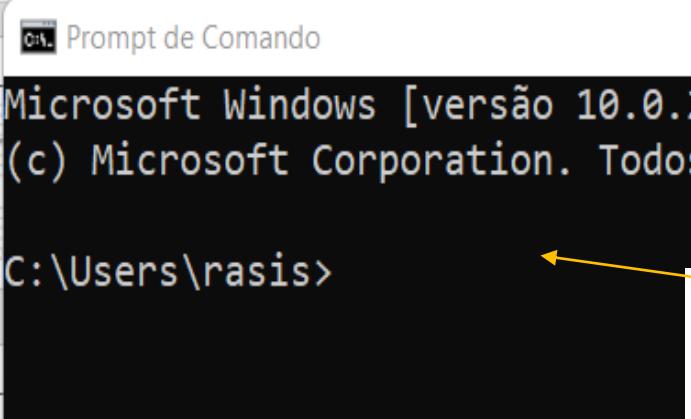
Introdução

- Algumas chamadas API Win32

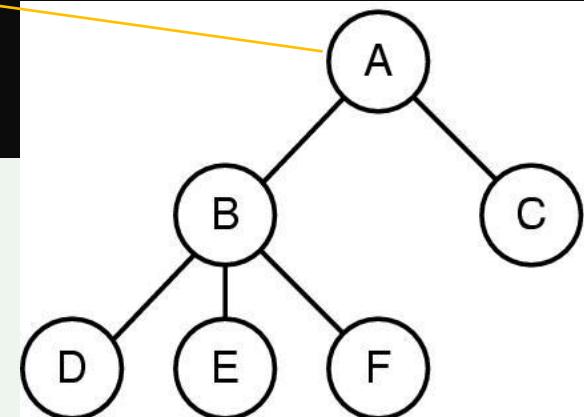
| UNIX | Win32 | Description |
|---------|---------------------|----------------------------------------------------|
| fork | CreateProcess | Create a new process |
| waitpid | WaitForSingleObject | Can wait for a process to exit |
| execve | (none) | CreateProcess = fork + execve |
| exit | ExitProcess | Terminate execution |
| open | CreateFile | Create a file or open an existing file |
| close | CloseHandle | Close a file |
| read | ReadFile | Read data from a file |
| write | WriteFile | Write data to a file |
| lseek | SetFilePointer | Move the file pointer |
| stat | GetFileAttributesEx | Get various file attributes |
| mkdir | CreateDirectory | Create a new directory |
| rmdir | RemoveDirectory | Remove an empty directory |
| link | (none) | Win32 does not support links |
| unlink | DeleteFile | Destroy an existing file |
| mount | (none) | Win32 does not support mount |
| umount | (none) | Win32 does not support mount |
| chdir | SetCurrentDirectory | Change the current working directory |
| chmod | (none) | Win32 does not support security (although NT does) |
| kill | (none) | Win32 does not support signals |
| time | GetLocalTime | Get the current time |

Introdução

- **Interpretador de Comandos (Shell):**
 - Criado sempre que o usuário abre uma sessão de trabalho (**sinal de prompt**);
 - Funcionamento:
 - Sempre que usuário digita um comando,
 - a shell (processo Pai) cria um processo filho e executa o comando sobre a estrutura do processo filho;
 - Algumas **shell** permitem utilizar o comando &
 - Coloca o processo executando em background;



```
Prompt de Comando  
Microsoft Windows [versão 10.0.22000.493]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\rasis>
```



- **Um simples Shell**

```
while (TRUE) {  
    type_prompt();  
    read_command(command, parameters)  
  
    if (fork() != 0) {  
        /* Código do PAI */  
        waitpid(-1, &status, 0);  
    } else {  
        /* Código do filho */  
        execve(command, parameters, 0);  
    }  
}
```

/* loop infinito */
/* mostra prompt */
/* entrada via terminal */

/* cria um processo filho */
/* espera o filho terminar p/ fim*/

/* executa comando no filho */



Sistemas Operacionais

Introdução

Tipos de Sistemas Operacionais

Sistemas Operacionais

Introdução

- **Monoprogramado ou Monotarefa:**
 - Único programa (job) utiliza o processador, a memória e os periféricos;
 - Único usuário utiliza o sistema por vez
 - Relacionado ao surgimento dos Mainframes;
 - Quando um programa aguarda um evento:
 - O processador fica ocioso;
 - A memória é sub-utilizada, quando não totalmente preenchida;
 - Não há necessidade de proteção do sistema
 - Existe apenas um usuário;

Sistemas Operacionais

Introdução

- **Multiprogramáveis ou Multitarefa:**
 - Vários programas compartilham os mesmos recursos;
 - Memória, processador e periféricos;
 - Vantagens:
 - Aumento da produtividade e redução dos custos (compartilhamento);
 - Mais de um usuário pode utilizar o sistema;
 - » Monousuário ou multiusuário
 - Podem ser classificados em:
 1. Sistema Batch;
 2. Sistemas de Tempo Compartilhado (Time-Sharing);
 3. Sistema de Tempo Real;

Sistemas Operacionais

Introdução

- **Multiprogramáveis ou Multitarefa:**

1. **Sistema Batch:**

- Caracterizam por terem:
 - » seus programas (JOBS) armazenados em disco ou fitas magnéticas
 - » Que esperam ser executados seqüencialmente;
- Não existem interação com os usuários
 - » Exemplos: compilações, linkedições, sorts, backups
- Vantagem quando bem protegido:
 - » Melhor utilização do processador;
- Desvantagem:
 - » Tempo de resposta longos (processamento sequencial) e
 - » Variação dos seus tempos de execução;

Sistemas Operacionais

Introdução

- **Multiprogramáveis ou Multitarefa:**

2. **Sistema de Tempo Compartilhado (Time-Sharing):**

- Permitem a interação do usuário com o sistema (teclado, vídeo e mouse);
- Conceito de fatia de tempo (Time-Slice) do processador
 - » Cada programa do usuário utiliza o processador durante sua fatia de tempo;
 - » Fica esperando nova fatia;
- Tanto a memória quanto periféricos também são compartilhados;
- Vantagem:
 - » Aumento da produtividade dos usuários
 - » Redução dos custos de utilização do sistema.

Sistemas Operacionais

Introdução

- **Multiprogramáveis ou Multitarefa:**

- 3. Sistema de Tempo Real:

- Semelhante aos sistemas de tempo compartilhado;
 - Maior diferença: Tempo de resposta exigido;
 - Deve estar dentro de tempos rígidos (ex: 30 frames por segundo);
 - Não existe a idéia de fatia de tempo de processador;
 - Programa executa até aparecer outro de maior prioridade;
 - Exemplos:
 - Monitoramento de refinarias de petróleo;
 - Controle de tráfego aéreo;
 - Aplicações multimídia;

Sistemas Operacionais

Introdução

- **Multiprocessados:**
 - Possuem mais que um processador;
 - Cada processador pode compartilhar a mesma memória ou cada um pode ter a sua própria;
 - Conceito de multiprogramação é mantida:
 - Sobre os diversos processadores;
 - Permite a re-configuração e o balanceamento do sistema;
 - Re-configuração : continua a processar mesmo se um processador falhar;
 - Balanceamento : balanceamento de carga nos processadores e de E/S;
 - Vários programas podem ser executados ao mesmo tempo;
 - Um programa pode ser dividido e cada parte executado em um processador;
 - Exemplos: aero-espacial, simulação e processamento de imagens

Sistemas Operacionais

Introdução

- **Multiprocessados : (Cont.)**
 - Dividido em:
 - Sistemas Fracamente Acoplados;
 - Sistemas Fortemente Acoplados;
 - Assimétricos e Simétricos
 - Sistemas Assimétricos;
 - Sistemas Simétricos;
 - Multiprocessamento:
 - Processamento vetorial e paralelo;

Sistemas Operacionais

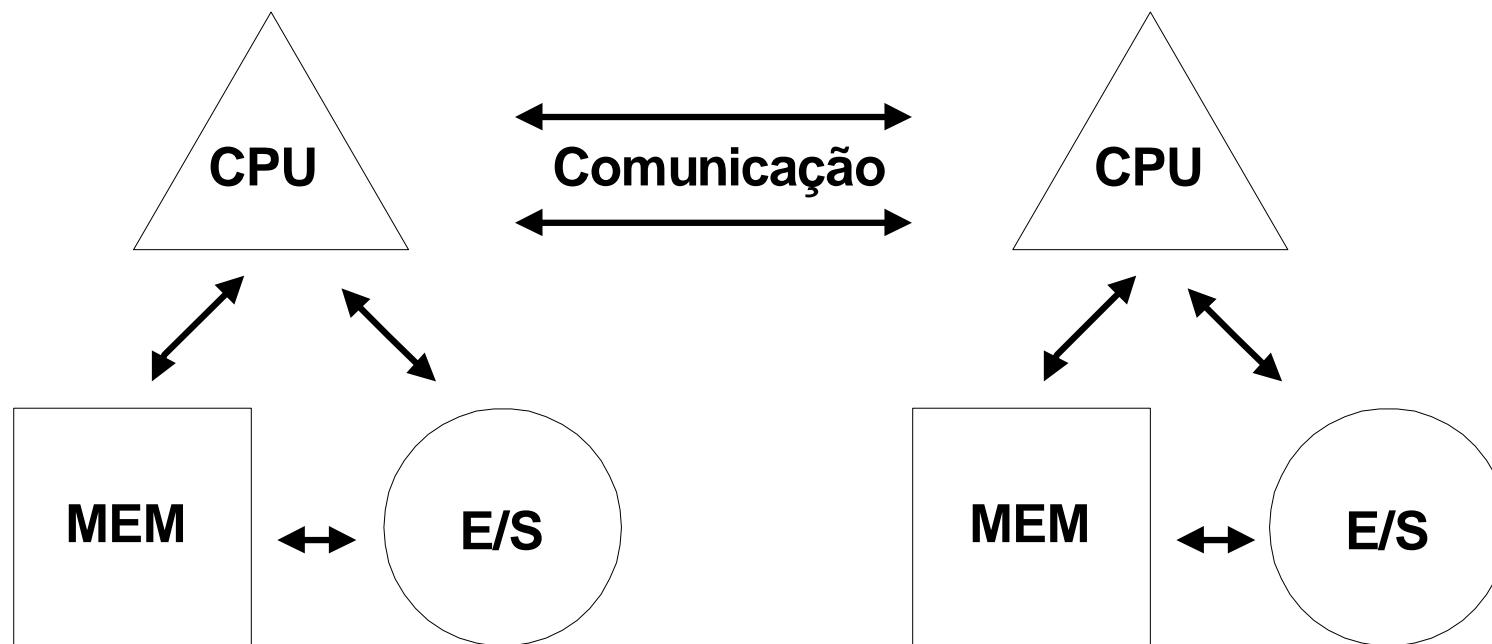
Introdução

- **Multiprocessados : (Cont.)**
 - Sistemas Fracamente Acoplados:
 - Possuem dois ou mais sistemas de computação;
 - Conectados através de uma linha de comunicação;
 - Cada sistema tem seu próprio processador, memória e dispositivos E/S e Sistema Operacional
 - Sistemas de computação podem funcionar independente ou não;
 - Comunicação

Sistemas Operacionais

Introdução

- Sistemas Fracamente Acoplados:



Sistemas Operacionais

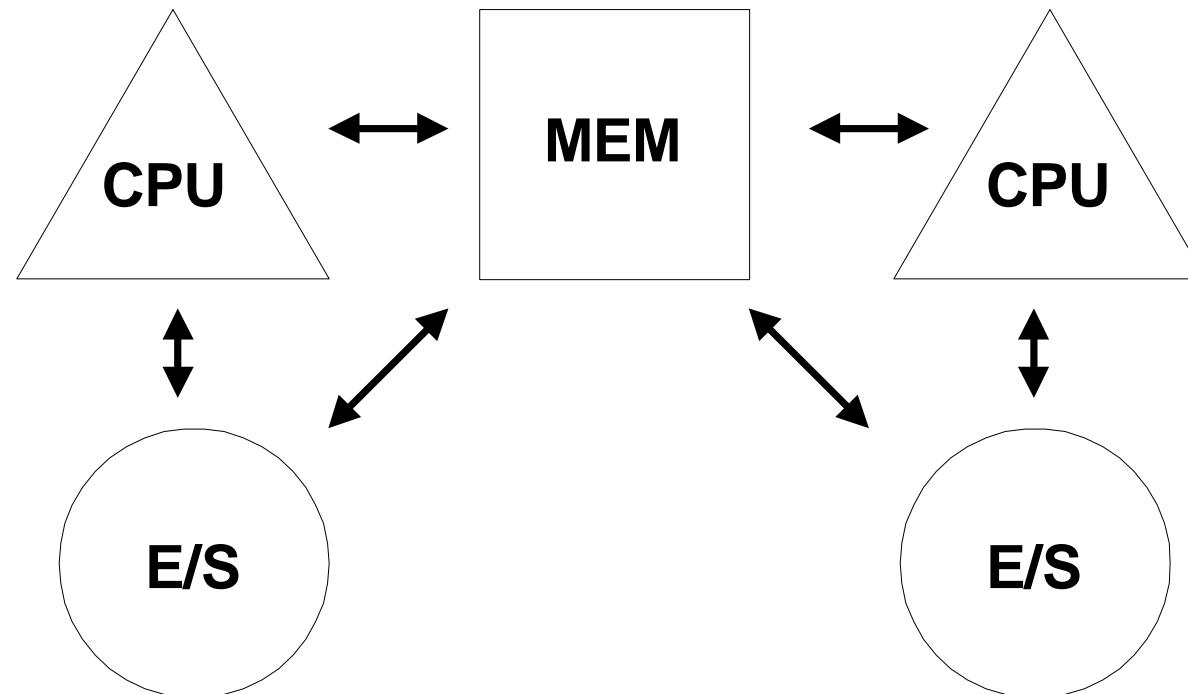
Introdução

- **Multiprocessados : (Cont.)**
 - Sistemas Fortemente Acoplados
 - Existem vários processadores;
 - Compartilham uma única memória;
 - Controlados por um único sistema operacional;
 - Exemplo: redes locais.

Sistemas Operacionais

Introdução

- Sistemas Fortemente Acoplados



Sistemas Operacionais

Introdução

- **Multiprocessados : (Cont.)**
 - Sistemas Fortemente Acoplados (Cont.)
 - Sistemas Fortemente Acoplados Assimétricos (mestre/escravo)
 - Possuem um processador primário que controla os demais processadores (secundários) e pela execução do Sistema Operacional;
 - Os processadores secundários processam programas do usuário e solicitam serviços ao processador primário.
 - Problema: falha do processador mestre (re-configurar um dos secundários)

Sistemas Operacionais

Introdução

- **Multiprocessados : (Cont.)**
 - Sistemas Fortemente Acoplados (Cont.)
 - Sistemas Fortemente Acoplados Simétricos
 - Todos os processadores tem a mesma função;
 - Pode ser executados Sistema operacional independentes em cada um deles;
 - Permite o balanceamento do processamento;

Sistemas Operacionais

Introdução

- **Sistemas Operacionais em Rede**
 - Cada nó da rede possui seu próprio:
 - S.O;
 - Hardware e software
 - Acesso a outros sistema da rede (compartilhar recursos);
 - Sistemas independentes em caso de falha;
 - Permite:
 - Cópia remota de arquivos,
 - Emulação de terminal,
 - Impressão remota,
 - Exemplo clássico : redes locais;

Sistemas Operacionais

Introdução

- **Sistemas Operacionais Distribuídos**
 - Cada componente da rede possui:
 - Seu próprio S.O, memória, processador e dispositivos de E/S;
 - O usuário tem a sensação de um sistema centralizado;
 - Vantagem:
 - Balanceamento de carga;
 - Compartilhamento de impressoras, discos e fitas
 - Independentes do sistema em que a aplicação esteja sendo executada.
 - Redundância do sistema
 - Em caso de falha outro pode assumir o papel;

Sistemas Operacionais

Introdução

- **Sistemas Operacionais Distribuídos**
 - Considerado a evolução dos sistemas fortemente acoplados
 - Uma aplicação pode ser executado por qualquer processador;
 - Aplicações podem ser divididas em diferentes partes que se comunicam através de uma linha de comunicação
 - Alguns padrões para implementação de S.D.:
 - DCE (Distributed Computing Environment)
 - CORBA (Common Object Broker Architecture)
 - OLE (Object Linking and Embedding)

Sistemas Operacionais

Introdução

- **Sistemas operacionais embarcados (Embedded)**
 - PDAs e computadores embarcados
 - Muitas vezes possuem características de tempo real
 - Possuem restrições de tamanho, memória, consumo de energia, etc
 - Ex: Windows Mobile, Palm OS, Symbian, Android...

Sistemas Operacionais

Introdução

- **SOs de cartões Inteligentes**
 - SMARTCARDS
 - Podem conter uma JVM

Sistemas Operacionais

Introdução

- **Sistemas Operacionais de computadores de grande porte**
 - Gerenciam computadores com grande capacidade de E/S
 - Oferecem (normalmente) três tipos de serviços:
 - Em lote (batch) -> jobs (processos em segundo plano)
 - Processamento de transações (CICS)
 - Tempo compartilhado (Time Sharing)
 - Ex: zOS, VM, ICS, Linux

Sistemas Operacionais

Introdução

Arquiteturas de Sistema Operacionais

Sistemas Operacionais

Arquiteturas de Sistema Operacionais

X

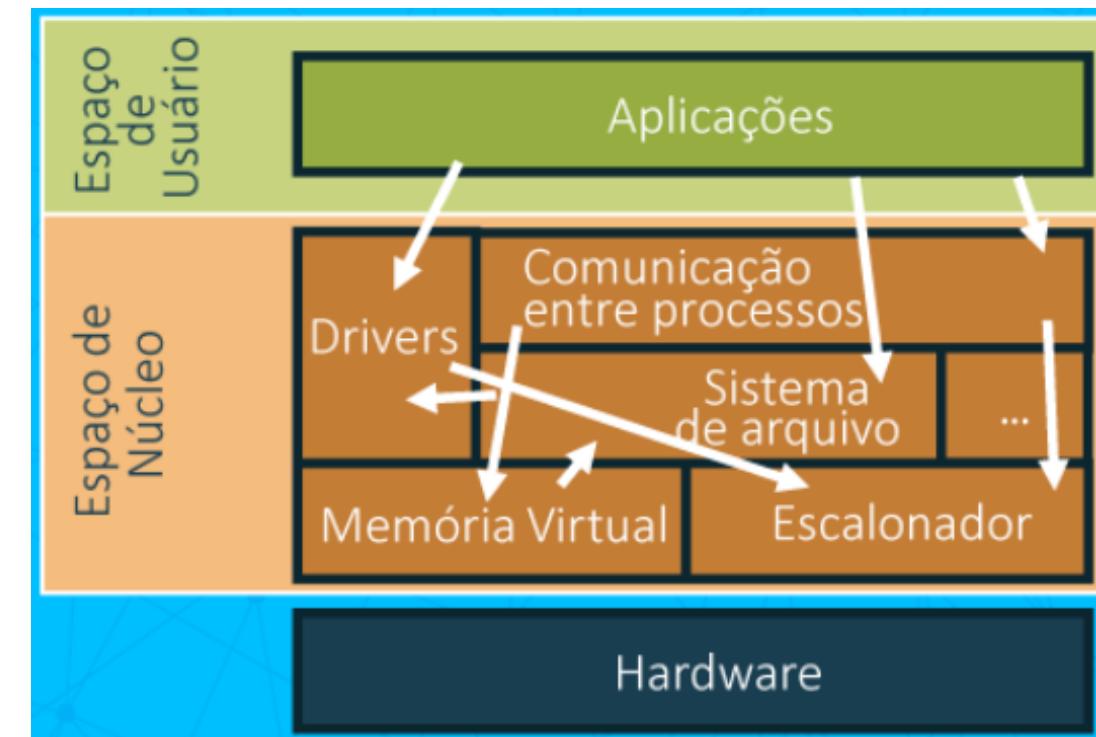
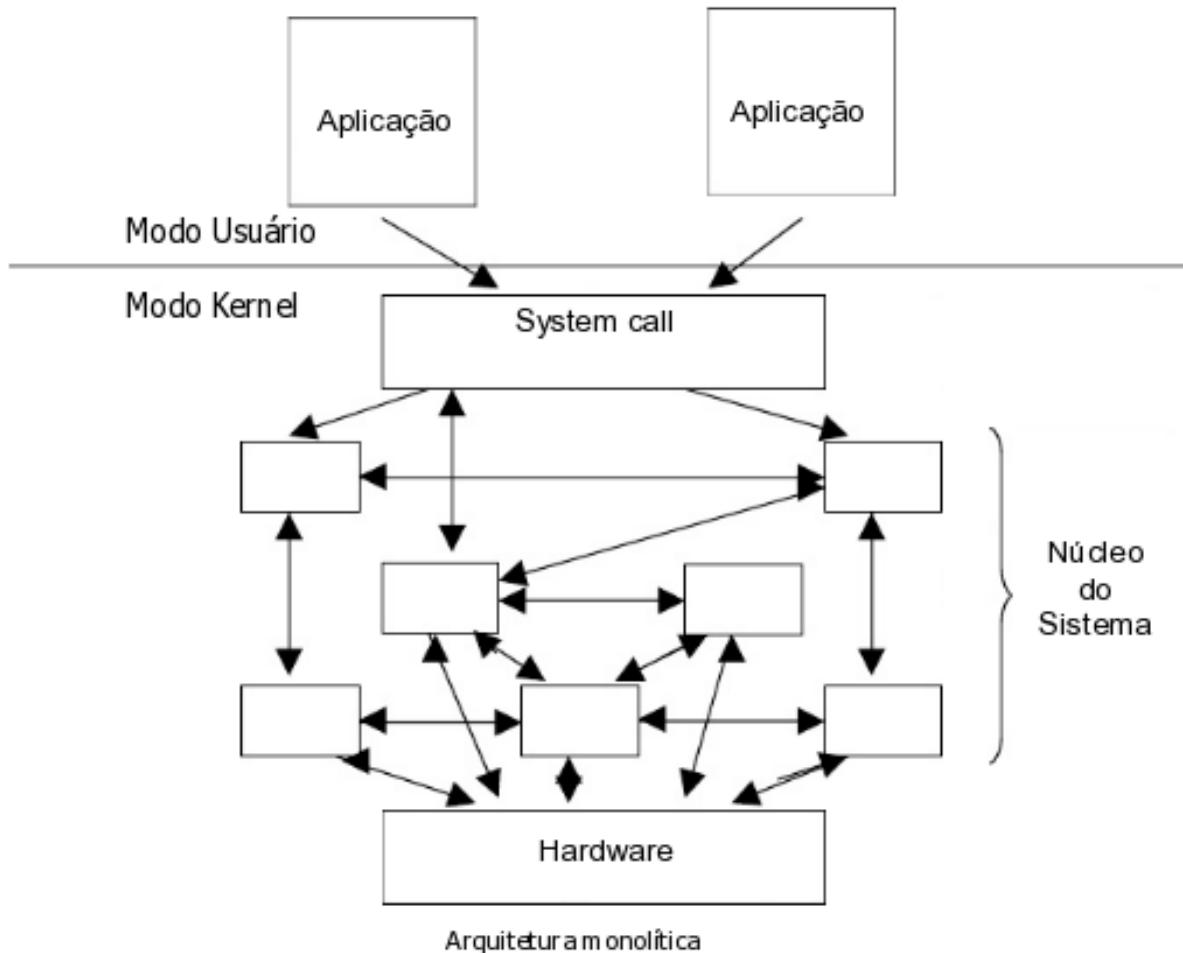
- **Arquitetura Monolítica:**

- É caracterizada por possuir seus **módulos compilados separadamente mas linkados** formando um único e enorme programa executável.
- Os módulos podem interagir livremente.
- Os primeiros sistemas operacionais foram desenvolvidos com base nesta arquitetura, o que tornava seu **desenvolvimento** e, principalmente, sua **manutenção muito difíceis**.
- Como **vantagens** desta arquitetura podemos citar a **rapidez de execução e simplicidade** de implementação.
- Como **desvantagens**,
 - a limitação quanto a inovações futuras e a dificuldade de manutenção;
 - Falha pode parar o núcleo;

Sistemas Operacionais

Arquiteturas de Sistema Operacionais

- Arquitetura Monolítica



Sistemas Operacionais

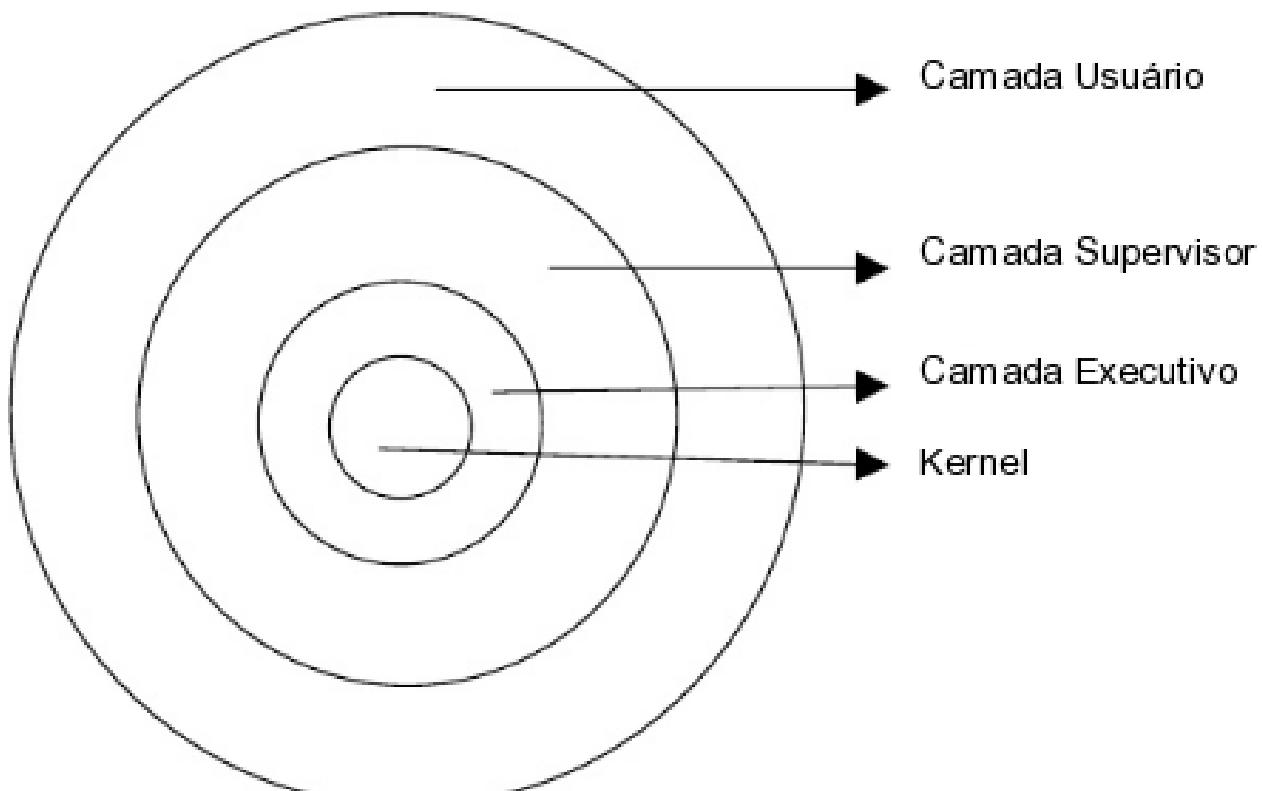
Arquiteturas de Sistema Operacionais

- **Arquitetura em Camadas:**
 - Com o aumento do tamanho do código dos sistemas operacionais, técnicas de programação estruturada e modular foram incorporadas em seu projeto.
 - Na arquitetura em camadas:
 - o sistema é dividido em **níveis sobrepostos**.
 - Cada **camada oferece um conjunto de funções** que podem ser utilizadas somente pelas camadas superiores.
 - Neste tipo de implementação as **camadas mais internas são mais privilegiadas que as camadas mais externas**.
 - Vantagem
 - **isolamento das funções** do sistema, facilitando sua manutenção.
 - Desvantagem
 - **é o desempenho**, comprometido devido às várias mudanças de estado do processador provocado pela mudança de camadas.

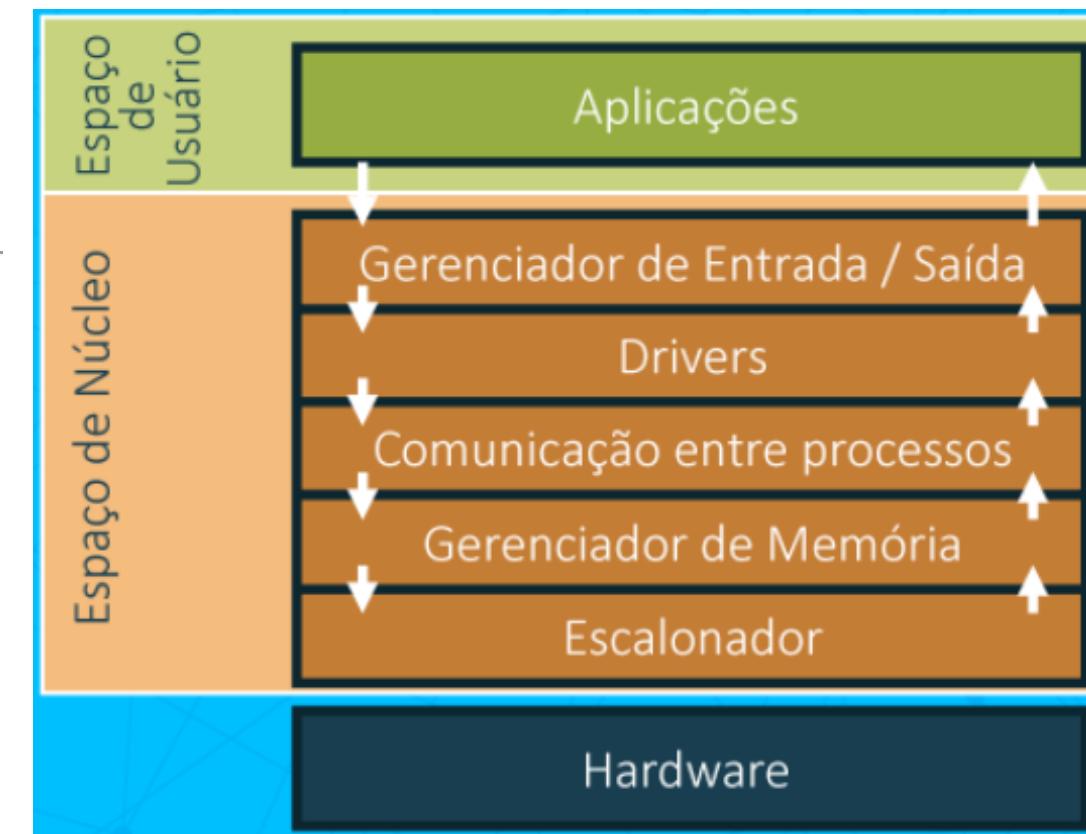
Sistemas Operacionais

Arquiteturas de Sistema Operacionais

- Arquitetura em Camadas:



Arquitetura em camadas



Sistemas Operacionais

Arquiteturas de Sistema Operacionais

- **Arquitetura Microkernel (Modelo cliente/servidor):**
 - uma tendência nos sistemas operacionais modernos é **tornar o núcleo do sistema o menor e o mais simples possível.**
 - Para implementar esta idéia:
 - os **serviços** do sistema são disponibilizados **através de processos**;
 - cada **processo** é responsável por oferecer um **conjunto específico de funções**, como gerência de arquivos, gerência de processos, gerência de memória e escalonamento.
 - Sempre que uma aplicação deseja algum serviço é realizada uma solicitação ao processo responsável.
 - Neste caso, a aplicação que está solicitando o serviço é chamada de cliente, enquanto o processo que responde à solicitação é chamado de servidor
 - Um **cliente**, solicita um serviço **enviando uma mensagem para o servidor**.
 - O servidor responde ao cliente através de uma outra mensagem.
 - A principal **função do núcleo** é realizar a **comunicação**, ou seja, a troca de mensagens entre o cliente e o servidor.

Sistemas Operacionais

Arquiteturas de Sistema Operacionais

- A utilização deste modelo **permite que os servidores operem em modo usuário**, não tendo acesso direto a certos componentes do sistema.
- Apenas o **núcleo do sistema é responsável pela comunicação entre clientes e servidores**, executa em modo kernel.
- Como consequência, se ocorrer um **erro em algum servidor**, este poderá parar, mas **o sistema não ficará inteiramente comprometido**, aumentando assim a sua disponibilidade.
- Como os **servidores se comunicam através de trocas de mensagens**, não importa se os clientes e servidores processam em um sistema com um único processador, com vários processadores ou ainda em um ambiente de sistema distribuído.

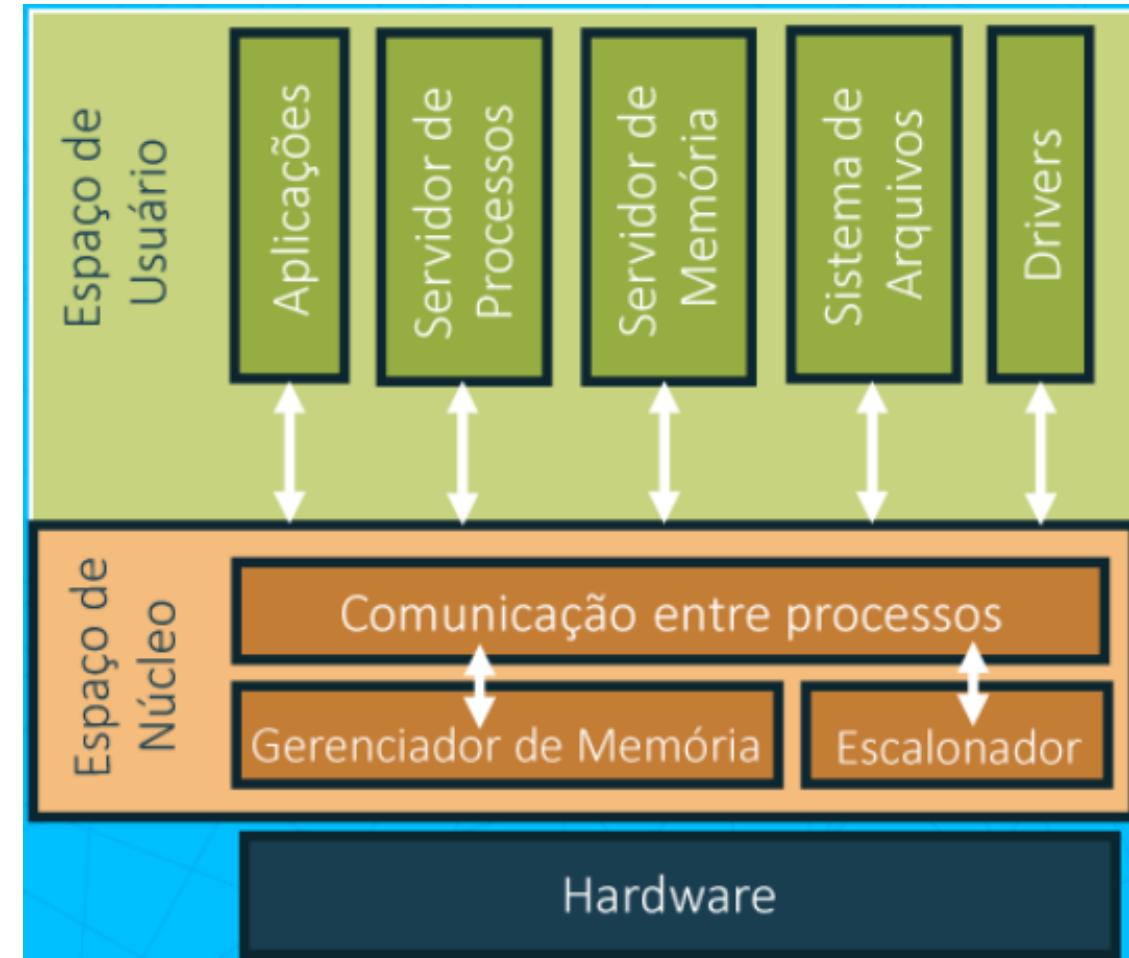
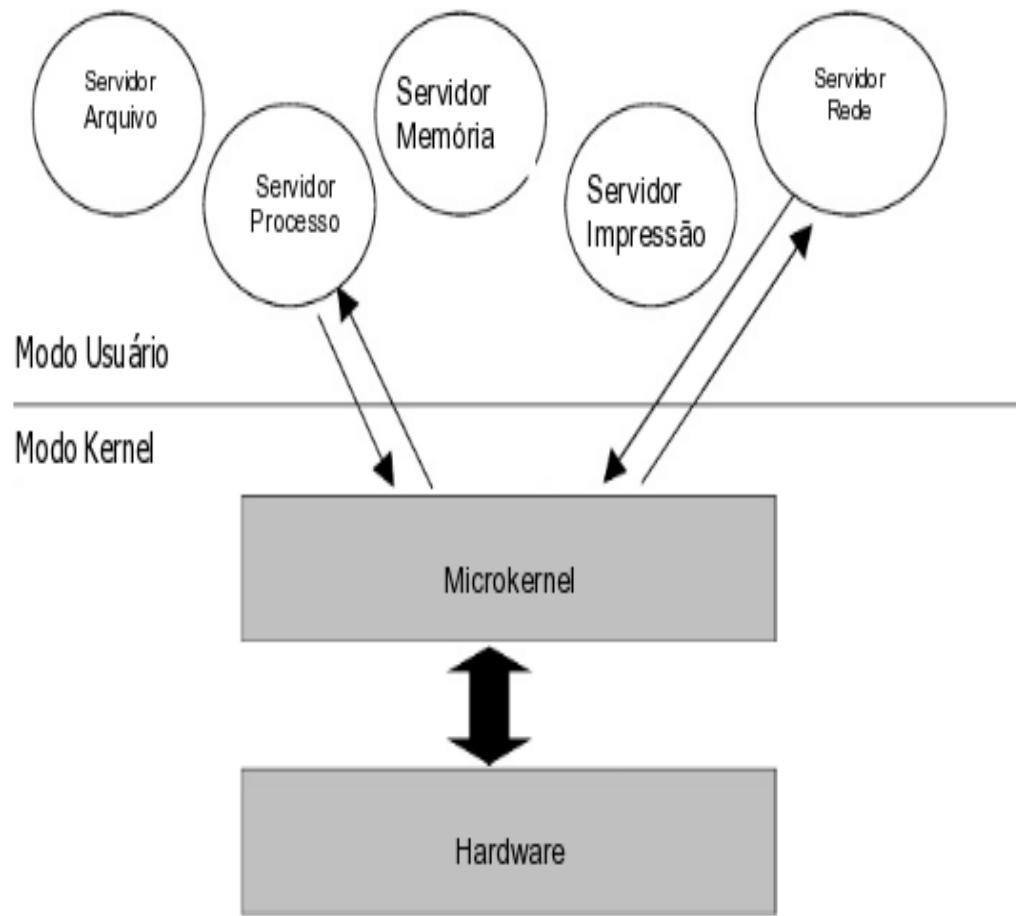
Sistemas Operacionais

Arquiteturas de Sistema Operacionais

- A arquitetura **microkernel** permite isolar as funções do sistema operacional por **diversos servidores pequenos e dedicados a serviços específicos**, tornando o núcleo menor, mais fácil de depurar (procurar e solucionar erros) e, com isso, aumentando a sua confiabilidade.
- Apesar de todas as vantagens deste modelo, sua **implementação é muito difícil**:
 - A começar pelo **desempenho, comprometido** devido às várias trocas de modo de acesso a cada troca de mensagens entre cliente e servidor.
 - Outro problema é que certas funções do sistema operacional, como operações de E/S, exigem acesso direto ao hardware.
 - Assim, o núcleo do sistema, além de promover a comunicação entre clientes e servidores, passa a incorpora funções críticas do sistema, como escalonamento, tratamento de interrupções e gerência de dispositivos;

Sistemas Operacionais

Arquiteturas de Sistema Operacionais



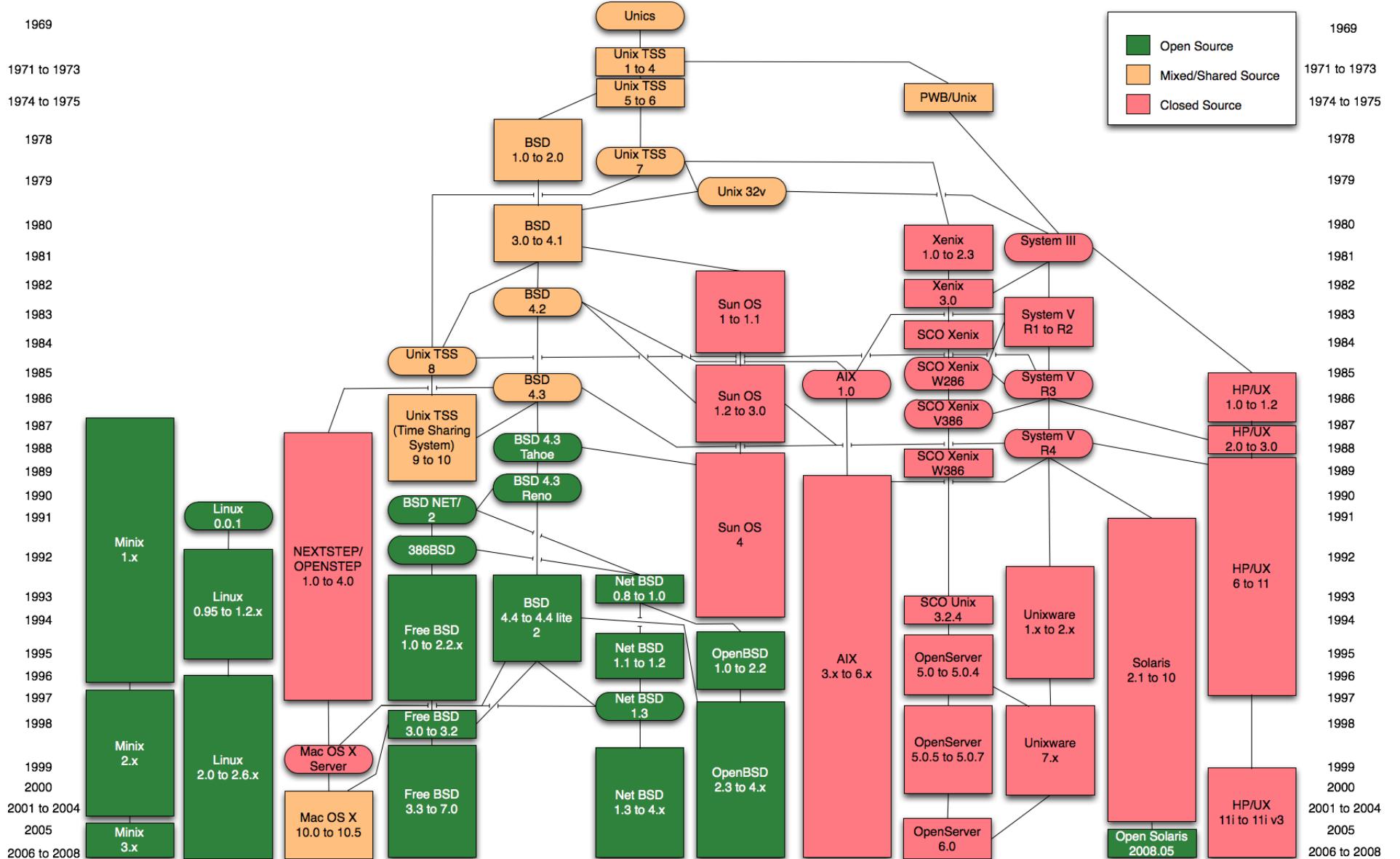
Sistemas Operacionais

Arquiteturas de Sistema Operacionais

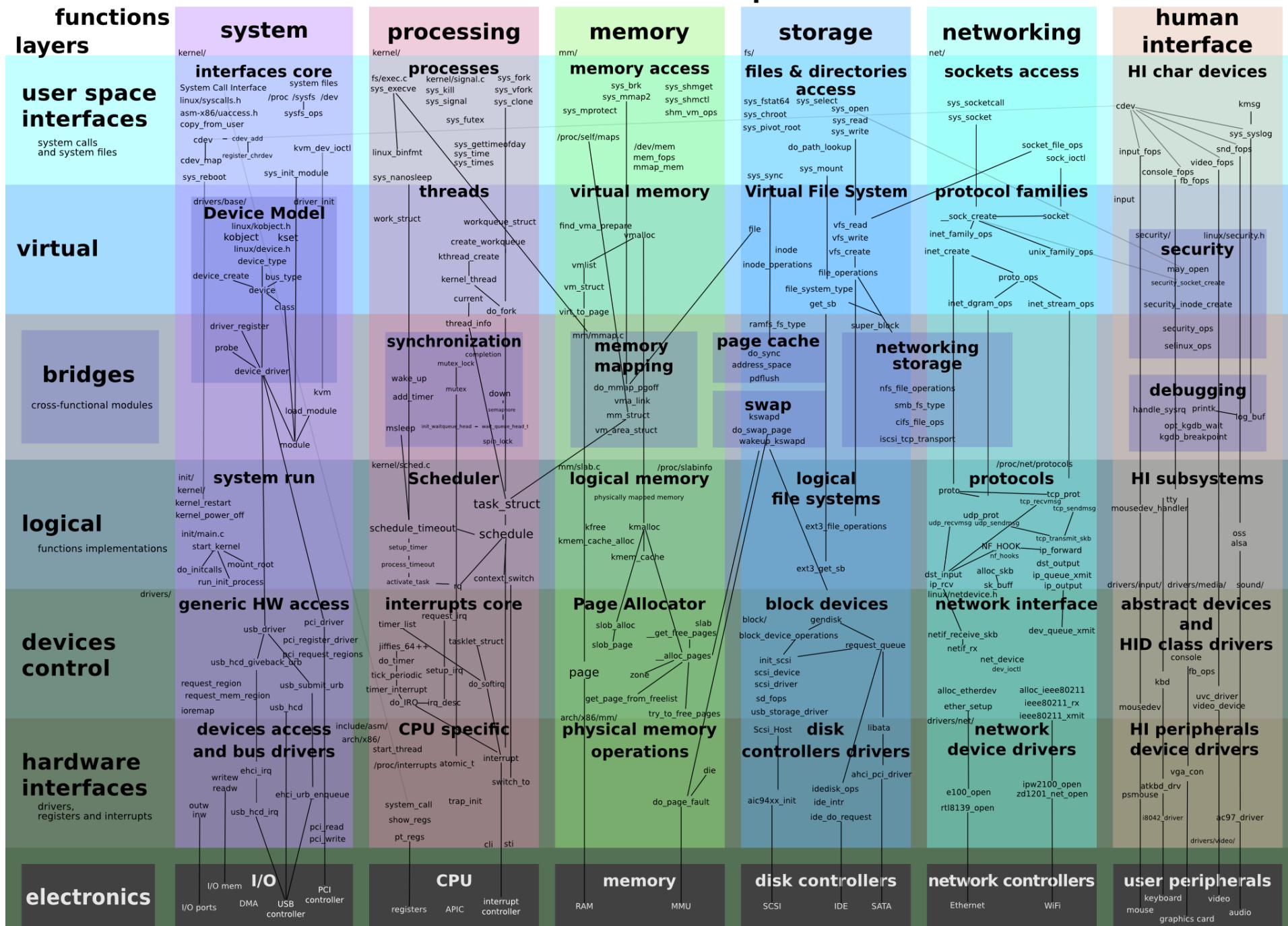
- Arquitetura Híbrida
 - A **arquitetura monolítica** é confusa, muito suscetível a bugs críticos e pode ser complexa para evoluir o SO, mas possibilita boa performance e oferece liberdade para o desenvolvimento;
 - A **arquitetuta de MicroKernel** evita que erros travem o sistema, pois pouco código roda com privilégio, mas pode ter má performance pelo necessidade extra de comunicação entre processo e chamada ao Espaço do Núcleo.
 - O **modelo híbrido**, que use as melhores características de cada uma dessas arquiteturas, é uma abordagem viavel e interessante, usada por grandes sistemas operacionais, como o windows 10;
 - Nesse modelo, é possível que a maior parte dos drivers e outros softwares de sistema sejam executados em espaço do Usuário, como em um sistema de MicroKernel, evitando travamentos. Mas, algumas partes frequentemente acessadas por aplicações, como o sistema de arquivos, ainda são mantidas em espaço do Nucleo, garantindo a performance.

Sistemas Operacionais

Hierarquia Unix



Linux kernel map



Sistemas Operacionais

Introdução

- Referências Utilizadas:
 - Livro do Tanenbaum
 - Sistemas Operacionais Modernos
 - www.cs.vu.nl/~ast
 - Livro do Silberschatz
 - Operating System Concepts
 - www.bell-labs.com/topic/books/aos-book/
 - Livro do Machado e Maia
 - Arquitetura de Sistemas Operacionais.