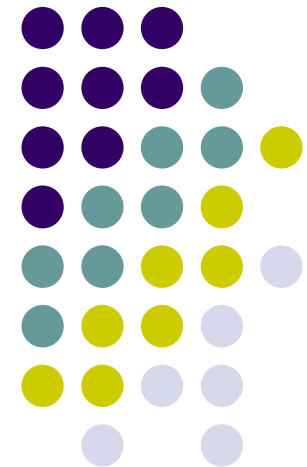
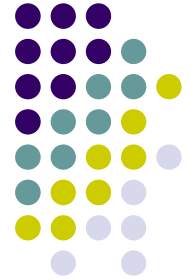


Sistemas Operacionais

Introdução





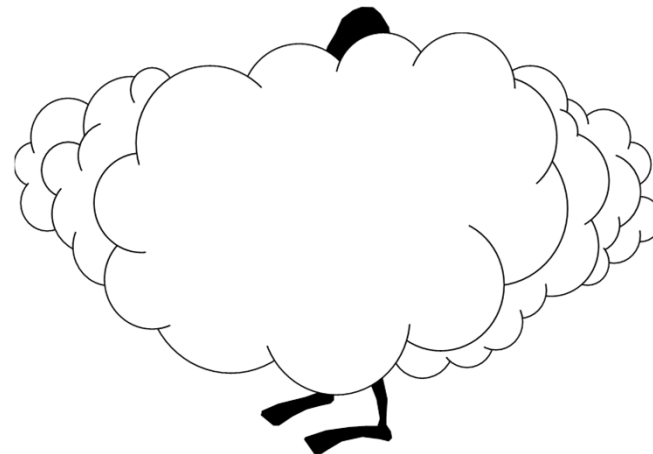
Introdução

- Sistema computacional
 - *Hardware*
 - *Software*
 - *Software Básico (Programas de Sistema)*
 - Gerenciam a operação do computador
 - *Software Aplicativo (Programas Aplicativos)*
 - Auxiliam o usuário em suas atividades específicas
 - Usuários

O que é um Sistema Operacional?



- É uma máquina estendida ou virtual
 - Esconde os detalhes que precisam ser executados
 - Apresenta uma interface com os usuário, mais amigável e mais fácil de usar.



Definição de Sistema Operacional



- É um programa cuja tarefa principal é controlar o funcionamento do computador, como um gerente dos diversos recursos disponíveis no sistema.
- Funções:
 - Facilidade de acesso aos recursos do sistema;
 - Compartilhamento de recursos de forma organizada e protegida.



O que eles controlam?

- Um computador típico

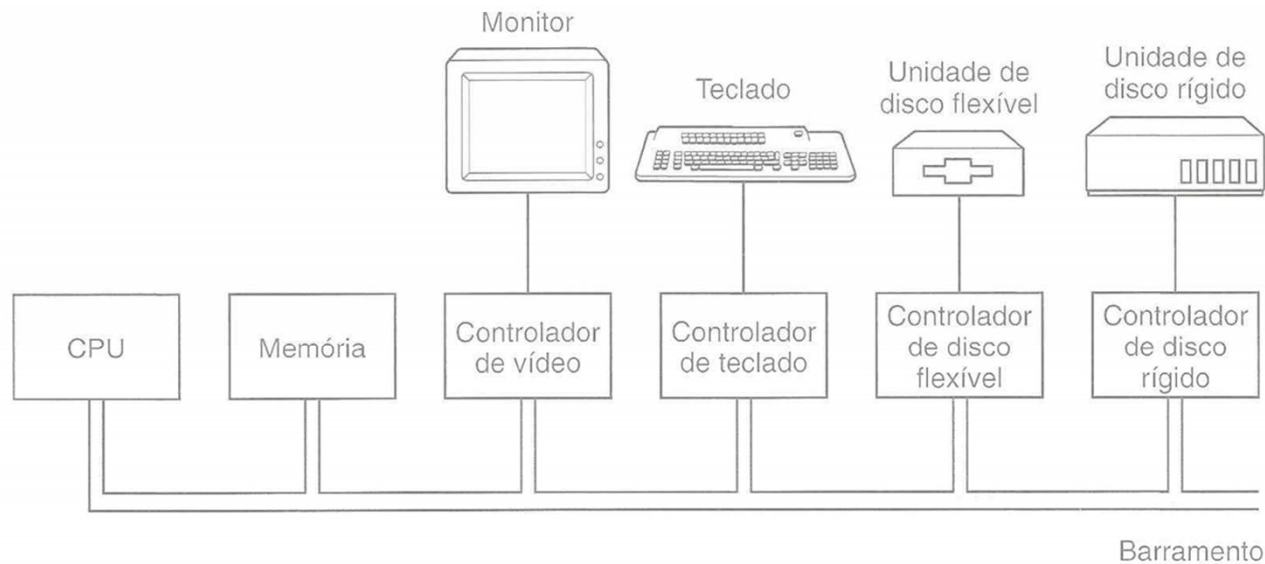


Figura 1.5 Alguns dos componentes de um computador pessoal simples.



O que eles controlam?

- Processadores (CPU)
 - Os processadores são recursos complexos, formados por unidade de controle, registradores, unidade aritmética e lógica que executam instruções armazenadas em memória.
 - Como algumas instruções dependem de componentes mais lentos, o processador fica ocioso, e pode ser compartilhado entre programas.



O que mais eles controlam?

- **Mémória**

- A memória também pode ser compartilhada por diversos processos, e tem que ser administrada.
- Quando um processo é eleito para iniciar, ele tem que ter memória para suas instruções e dados.
- Muitas vezes, um processo é retirado de memória para dar lugar a outro... seus dados são, então, armazenados em disco.
- Da memória principal para cima o controle cabe ao sistema operacional – é ele quem mapeia o uso, decide fazer cópias para disco, remover processos da memória, fazer *backups*, etc.

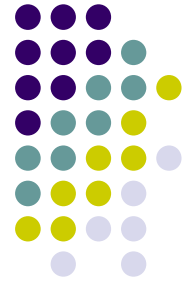


O que mais eles controlam?

- Disco

- A transformação da representação vista pelo usuário em número de cilindro, trilha, bloco, registro é feita pelo sistema operacional, de forma precisa.
- Para isso, o S.O. deve ter um sistema de arquivos e diferentes métodos de acesso aos dados.
- Quando um arquivo é criado, é necessário ser criada uma entrada para ele no diretório e deve ser reservado um espaço inicial em disco para o mesmo.
- Conforme o arquivo cresce, pode ser necessário reservar mais espaço em disco – e ir guardando essa informação em um “mapa” do arquivo.

O que mais eles controlam?



- Os SOs têm que controlar o acesso a diversos componentes, tais como discos, impressoras, CDs/DVDs, e outros dispositivos.
- Nos dispositivos de E/S, o nível baixo é controlado pelo hardware, e o controle de uso por usuários e processos é feito pelo SO.
- É um trabalho totalmente integrado.

O que é um Sistema Operacional?



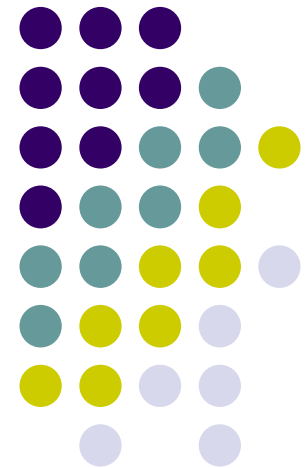
- É um gerenciador de recursos
 - Para todos os recursos da máquina, o sistema operacional deve:
 - manter informações sobre o recurso
 - decidir quem pode acessar o recurso
 - alocar o recurso
 - liberar o recurso
 - Quanto ao controle de utilização de recursos, o SO deve:
 - ser eficiente (maximizar a utilização dos recursos)
 - possuir um tempo de resposta aceitável

O que é um Sistema Operacional?



- O SO visto a partir do:
 - Usuário
 - Deve oferecer facilidade e transparência na utilização;
 - Deve garantir o bom desempenho da máquina.
 - Sistema
 - É o gerente de recursos de *hardware* e de *software*;
 - Deve lidar com os detalhes do *hardware*.

Serviços oferecidos pelo sistema operacional



Serviços oferecidos pelo Sistema Operacional



- Detecção e controle de erros
 - De *hardware*
 - Internos – memória
 - Externos – dispositivos
 - De *software*
 - *Overflow*
 - Problemas de acesso à memória
- O sistema operacional não se responsabiliza pelos detalhes das aplicações

Serviços oferecidos pelo Sistema Operacional



- Informações diversas
 - Coleta e processamento de estatísticas
 - Monitoramento de desempenho
 - Previsão e tratamento de falhas
 - Controle de acesso de usuários
 - Aos recursos
 - Limitação de uso dos recursos

Serviços oferecidos pelo Sistema Operacional



- Para oferecer esses serviços os sistemas operacionais se valem de certas abstrações em torno das quais ele se organiza.
- A principal dessas abstrações é o **processo**.
- Um processo é uma entidade “viva” dentro de um computador. Todas as ações são tomadas a partir de um processo.



Ele - O Kernel

- É também conhecido como “núcleo” ou “coração” do sistema operacional
- É a porção do sistema operacional que permanece na memória principal
- Contém as funções utilizadas mais freqüentemente





Ele – O Kernel

- O Kernel não é, entretanto, todo o sistema operacional.
- Várias das funções do SO são feitas por processos que são acionados pelo Kernel.
- O Kernel deve ser extremamente conciso e suas entradas atuam, principalmente, na manipulação de variáveis críticas que controlam os recursos do sistema.



Ele – O Kernel

- Os processos solicitam serviços direta ou indiretamente ao Kernel por meio de chamadas ao sistema (*system calls*).
- O Kernel atende a tais chamadas e, geralmente, aciona processos apropriados para resolvê-las.
- Existem diversos tipos de chamadas ao sistema.



Ele - O Kernel

- Chamadas para gerenciamento de processos
 - criar (fork), esperar (waitpid), terminar (exit)
- Chamadas para gerenciamento de arquivos
 - open, close, read...
- Chamadas para gerenciamento de diretório
 - mkdir, rmdir ...
- Chamadas diversas
 - chdir, chmod, kill,...

Definições Básicas...



O SO é formado por um conjunto de rotinas (procedimentos) que oferecem serviços aos usuários do sistema e suas aplicações, bem como a outras rotinas do próprio sistema;

Esse conjunto de rotinas é chamado de *núcleo do sistema* ou *kernel*;

Definições Básicas...



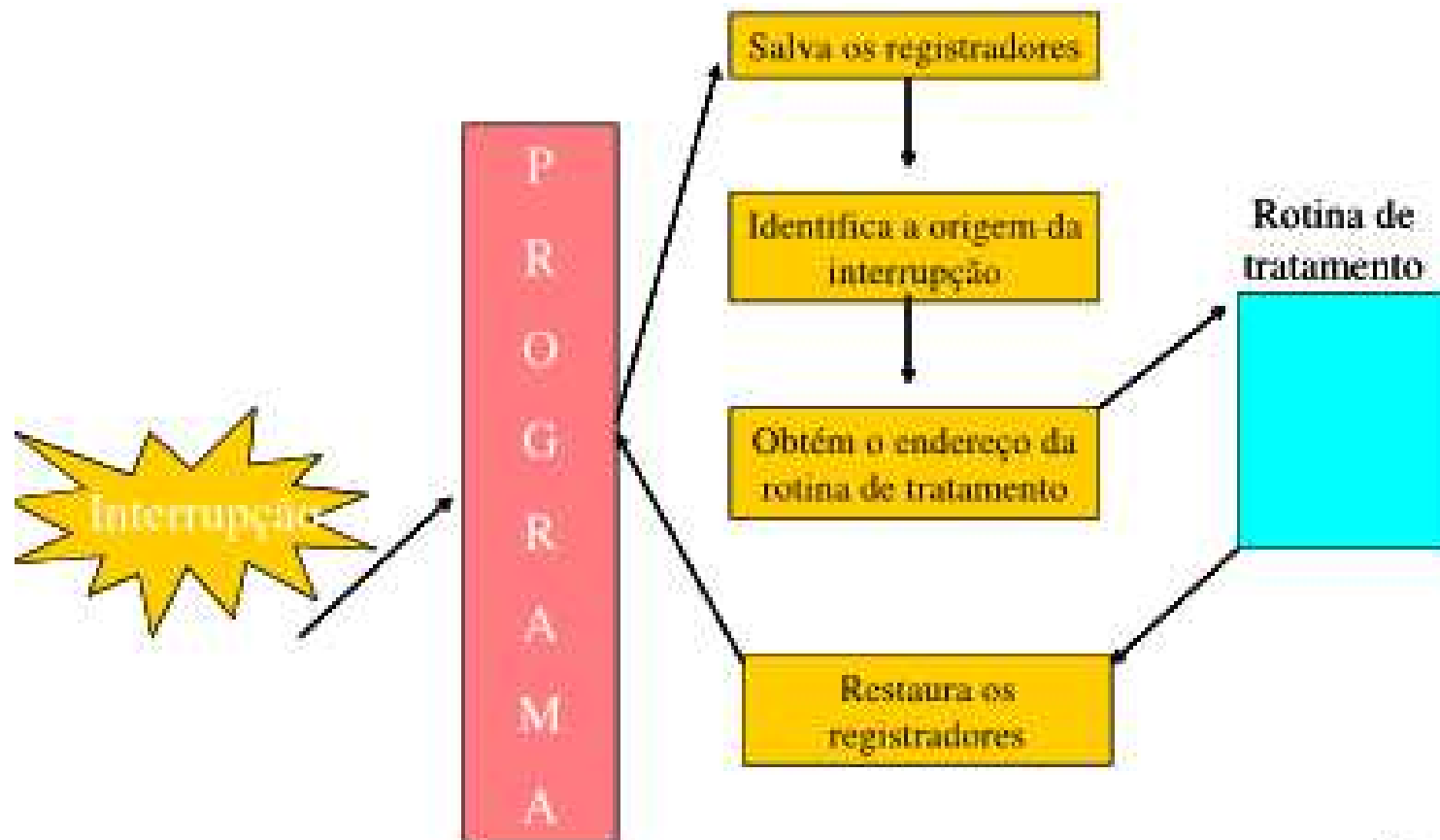
- As principais funções do núcleo são:
 - tratamento de interrupções;
 - criação e eliminação de processos;
 - sincronização e comunicação entre processos;
 - gerência de memória;
 - gerência do sistema de arquivo.

Definições Básicas...



- Os programas dos usuários solicitam serviços do SO por meio da execução de **chamadas de sistema**;
- A cada chamada corresponde um procedimento de uma biblioteca de procedimentos que o programa do usuário pode chamar;
- Tal procedimento coloca os parâmetros da chamada de sistema em locais específicos, tais como registradores da máquina e emite uma instrução de **TRAP**;
- Quando o SO é chamado após o TRAP, ele examina os parâmetros para ver se eles são válidos;
- Caso sejam, o SO realiza a função requerida pelo usuário;
- Ao terminar ele coloca um código de estado em um determinado registrador, informando se a operação teve sucesso ou falhou.

Definições Básicas...



25

Definições Básicas

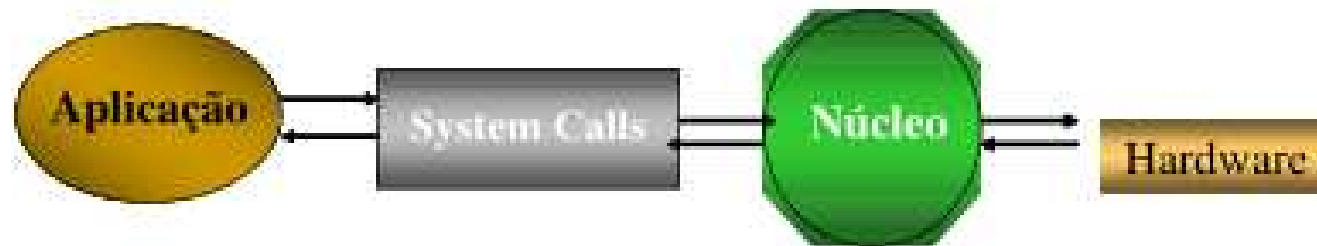


- Não existe apenas um único tipo de interrupção e sim diferentes tipos que devem ser atendidas por diversas rotinas de tratamento.
- No momento que uma interrupção acontece, a CPU deve saber para qual rotina de tratamento deverá ser desviado o fluxo de execução.
- Essa informação está em uma estrutura do sistema chamada *Vetor de Interrupção*, que contém a relação de todas as rotinas de tratamento existentes.

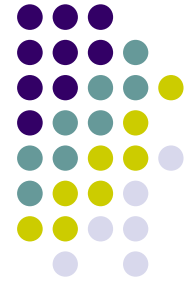
Chamadas de Sistema



- Chamadas de Sistema (*System Calls*) são a porta de entrada para ter acesso ao núcleo do sistema operacional.
- Para cada serviço existe uma *system call* associada, e cada SO tem o seu próprio conjunto (biblioteca) de chamadas, com nomes, parâmetros e formas de ativação específicos.
- As *system calls* podem ser divididas em grupos de função: gerência de processos, gerência de memória, gerência de entrada/saída e gerência de arquivos.



Chamadas de Sistema



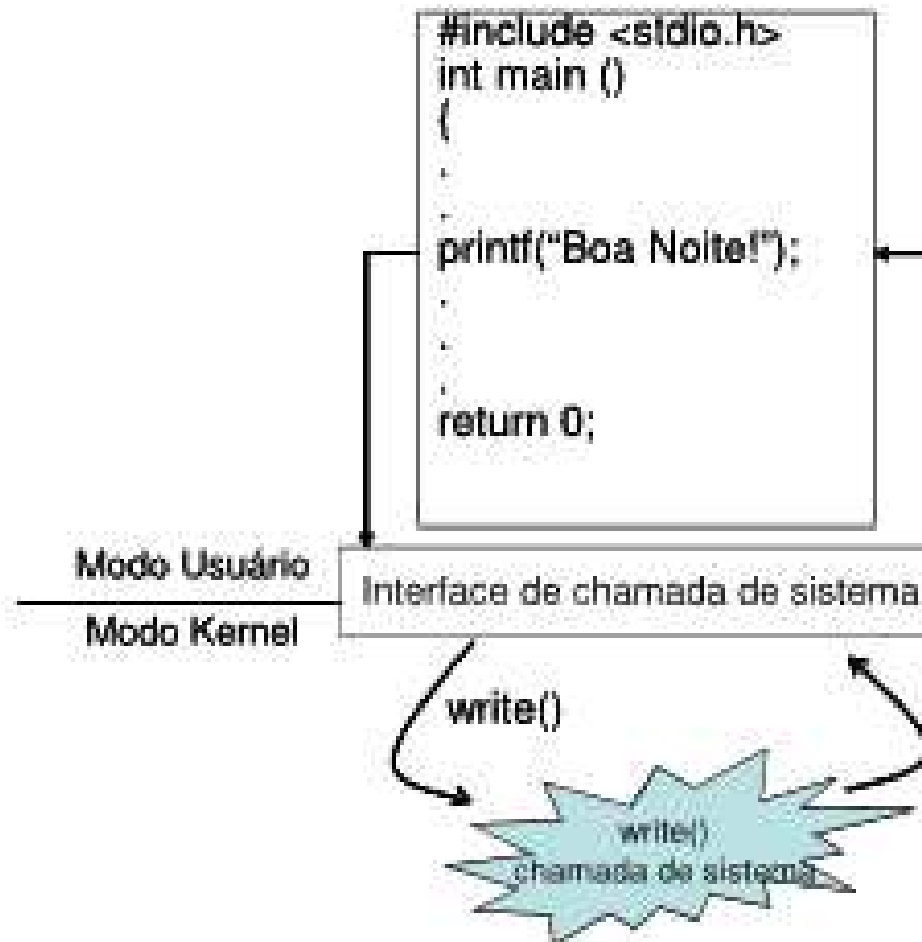
- Para impedir a ocorrência de problemas de segurança e mesmo violação do sistema, as instruções que têm o poder de comprometer o sistema são conhecidas como ***instruções privilegiadas***.
- Enquanto as instruções ***não-privilegiadas*** são as que não oferecem perigo ao sistema.
- Para que uma aplicação possa executar uma instrução privilegiada, o processador implementa o mecanismo de ***modos de acesso***.

Chamadas de Sistema



- Há dois modos de acesso implementados no processador:
 - Modo Usuário
 - Modo Kernel
- Quando o processador trabalha no *modo usuário*, uma aplicação só pode executar instruções não privilegiadas, tendo acesso a um número reduzido de instruções.
- No *modo kernel* a aplicação pode ter acesso ao conjunto total de instruções do processador.

Chamadas de Sistema



Chamadas de Sistema

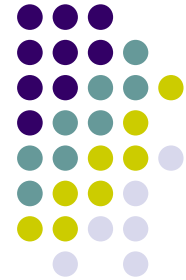


- Para cada serviço existe uma chamada de sistema associada, e cada SO tem o seu próprio conjunto (biblioteca) de chamadas, com diferentes nomes e parâmetros;
- As chamadas de sistema podem ser divididas em grupos.

Gerenciamento de processos

| Chamada | Descrição |
|--|---|
| <code>pid = fork()</code> | Cria um processo filho idêntico ao pai |
| <code>pid = waitpid(pid, &statloc, options)</code> | Espera que um processo filho seja concluído |
| <code>s = execve(name, argv, environp)</code> | Substitui a imagem do núcleo de um processo |
| <code>exit(status)</code> | Conclui a execução do processo e devolve status |

Chamadas de Sistema



Gerenciamento de arquivos

| Chamada | Descrição |
|---|---|
| <code>fd = open(file, how, ...)</code> | Abre um arquivo para leitura, escrita ou ambos |
| <code>s = close(fd)</code> | Fecha um arquivo aberto |
| <code>n = read(fd, buffer, nbytes)</code> | Lê dados a partir de um arquivo em um buffer |
| <code>n = write(fd, buffer, nbytes)</code> | Escreve dados a partir de um buffer em um arquivo |
| <code>position = lseek(fd, offset, whence)</code> | Move o ponteiro do arquivo |
| <code>s = stat(name, &buf)</code> | Obtém informações sobre um arquivo |

Gerenciamento do sistema de diretório e arquivo

| Chamada | Descrição |
|---|--|
| <code>s = mkdir(name, mode)</code> | Cria um novo diretório |
| <code>s = rmdir(name)</code> | Remove um diretório vazio |
| <code>s = link(name1, name2)</code> | Cria uma nova entrada, name2, apontando para name1 |
| <code>s = unlink(name)</code> | Remove uma entrada de diretório |
| <code>s = mount(special, name, flag)</code> | Monta um sistema de arquivos |
| <code>s = umount(special)</code> | Desmonta um sistema de arquivos |

Chamadas de Sistema



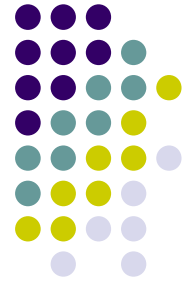
| Diversas | |
|---|---|
| Chamada | Descrição |
| <code>s = chdir(dirname)</code> | Altera o diretório de trabalho |
| <code>s = chmod(name, mode)</code> | Altera os bits de proteção de um arquivo |
| <code>s = kill(pid, signal)</code> | Envia um sinal para um processo |
| <code>seconds = time(&seconds)</code> | Obtém o tempo decorrido desde 1º de janeiro de 1970 |

Chamadas de Sistema – Unix x Win32



| UNIX | Win32 | Descrição |
|---------|---------------------|--|
| fork | CreateProcess | Cria um novo processo |
| waitpid | WaitForSingleObject | Pode esperar que um processo saia |
| execve | (nenhuma) | CreateProcess = fork + execve |
| exit | ExitProcess | Conclui a execução |
| open | CreateFile | Cria um arquivo ou abre um arquivo existente |
| close | CloseHandle | Fecha um arquivo |
| read | ReadFile | Lê dados a partir de um arquivo |
| write | WriteFile | Escreve dados em um arquivo |
| lseek | SetFilePointer | Move o ponteiro do arquivo |
| stat | GetFileAttributesEx | Obtém vários atributos do arquivo |
| mkdir | CreateDirectory | Cria um novo diretório |
| rmdir | RemoveDirectory | Remove um diretório vazio |
| link | (nenhuma) | Win32 não dá suporte a links |
| unlink | DeleteFile | Destroi um arquivo existente |
| mount | (nenhuma) | Win32 não dá suporte a mount |
| umount | (nenhuma) | Win32 não dá suporte a mount |
| chdir | SetCurrentDirectory | Altera o diretório de trabalho atual |
| chmod | (nenhuma) | Win32 não dá suporte a segurança (embora o NT suporte) |
| kill | (nenhuma) | Win32 não dá suporte a sinais |
| time | GetLocalTime | Obtém o tempo atual |

Quais são os tipos de Sistemas Operacionais?



- Existem sistemas operacionais para:
 - Sistemas de grande porte
 - Sistemas para servidores
 - Computadores pessoais
 - Sistemas de tempo real
 - Sistemas embarcados

Estruturas de um Sistema Operacional



Estrutura do Sistema Operacional



- Agora que já tivemos uma visão externa do SO, é importante entender a sua estrutura interna;
- As principais são:
 - Sistemas monolíticos;
 - Sistemas em camadas;
 - Micronúcleo;
 - Sistemas cliente-servidor.

Estrutura do Sistema Operacional



- Sistemas monolíticos:
 - É a organização mais comum, na qual o SO inteiro é executado como um único programa no modo núcleo;
 - O S.O. é escrito como uma coleção de procedimentos, sendo que cada um pode chamar um dos demais sempre que necessário
 - Cada procedimento possui uma interface bem definida.
 - Não há proteção entre os procedimentos.
 - Quanto ao tempo de resposta, é uma organização eficaz.

Estrutura do Sistema Operacional

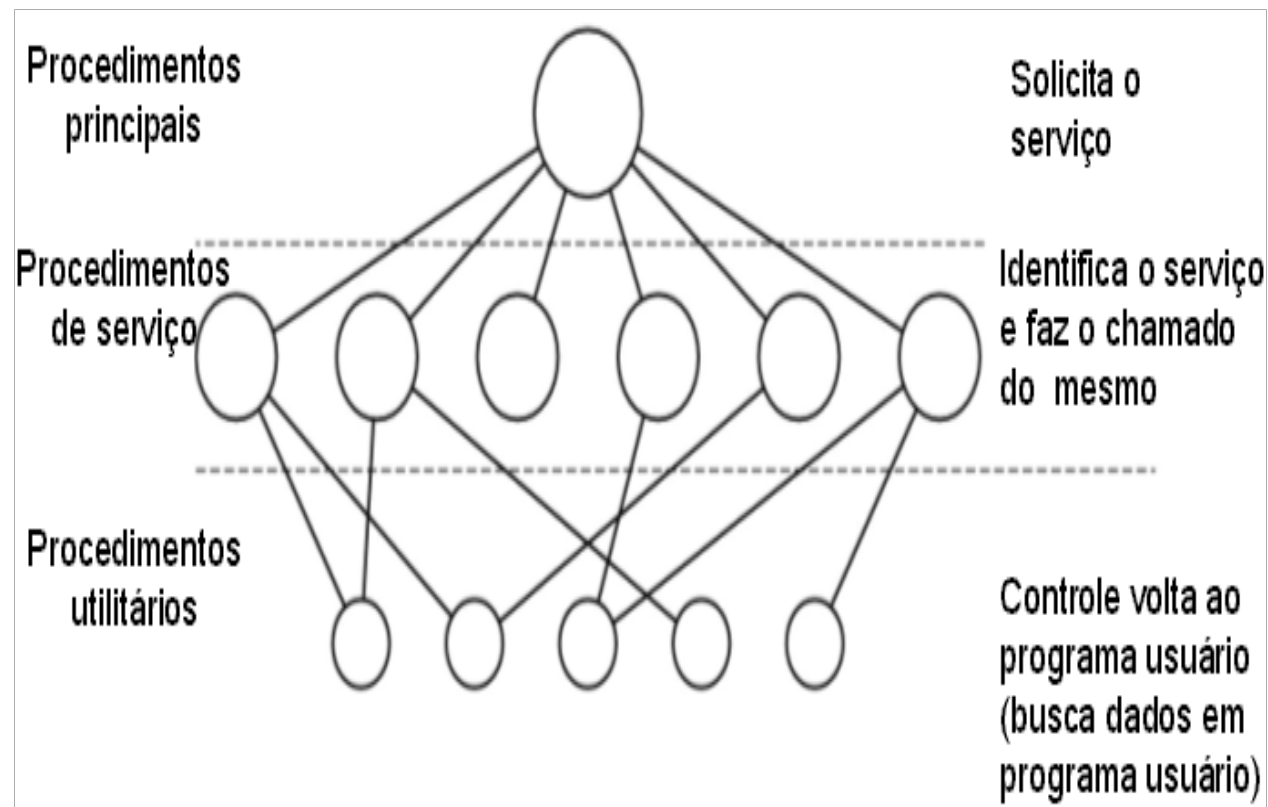


- Sistemas monolíticos:
 - O sistema operacional roda em modo kernel, enquanto os demais programas rodam em modo usuário.
 - O sistema operacional monolítico possui, geralmente, três “níveis”:
 - um procedimento principal que chama os procedimentos de serviço;
 - um conjunto de procedimentos de serviço que tratam as chamadas ao sistema; e
 - um conjunto de procedimentos que ajudam os procedimentos de serviço.

Estrutura do Sistema Operacional



- Sistemas monolíticos:



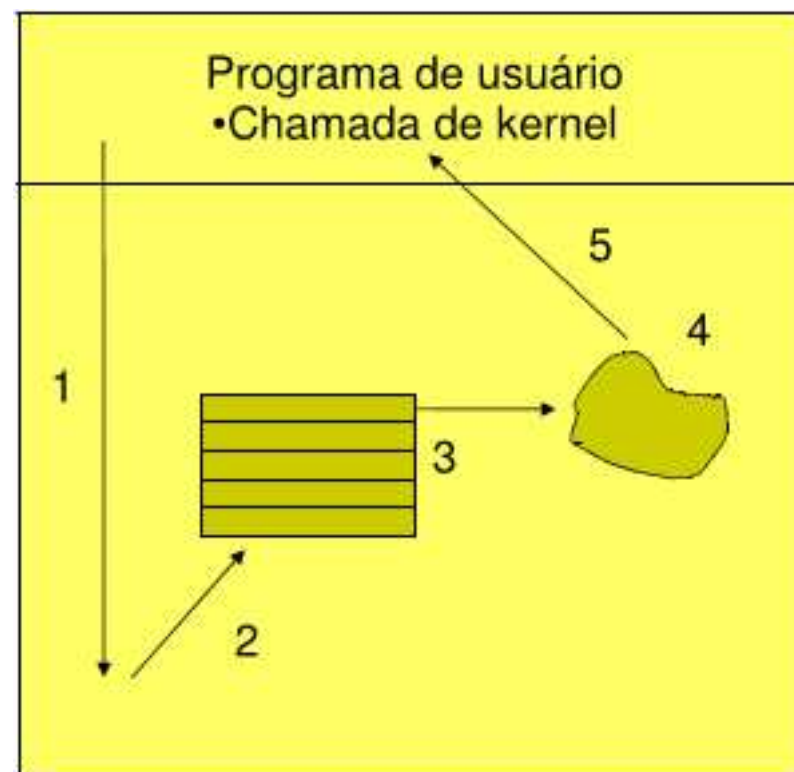
Fonte: Tanenbaum [1]

Estrutura do Sistema Operacional



- Sistemas Monolíticos:

1. Programa faz chamada ao sistema e transfere controle para o S.O.;
2. S.O. verifica qual chamada de sistema deve ser executada;
3. S.O. busca o procedimento que deverá ser executado;
4. O procedimento é executado;
5. O controle retorna para o programa de usuário.



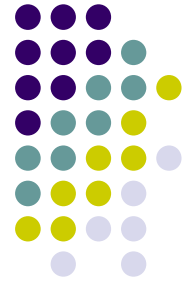
Fonte: Tanenbaum [1]

Estrutura do Sistema Operacional



- Sistemas em Camadas:
 - Sistemas que apresentam uma hierarquia de camadas;
 - Sistema dividido em níveis de complexidade, sendo que os níveis inferiores prestam serviços para os superiores;
 - As diversas camadas do sistema são dependentes, e possuem funções específicas;
 - O primeiro sistema operacional construído em camadas foi o THE (Technische Hogeschool Eindhoven), na Holanda em 1968.

Estrutura do Sistema Operacional



- Sistemas em Camadas:
 - Exemplo THE

| Camada | Função |
|--------|--|
| 5 | Operador |
| 4 | Programa usuário |
| 3 | Gerenciamento de E/S |
| 2 | Comunicação operador-processo |
| 1 | Gerenciamento de memória |
| 0 | Alocação do processador e multiprogramação |

Estrutura do Sistema Operacional

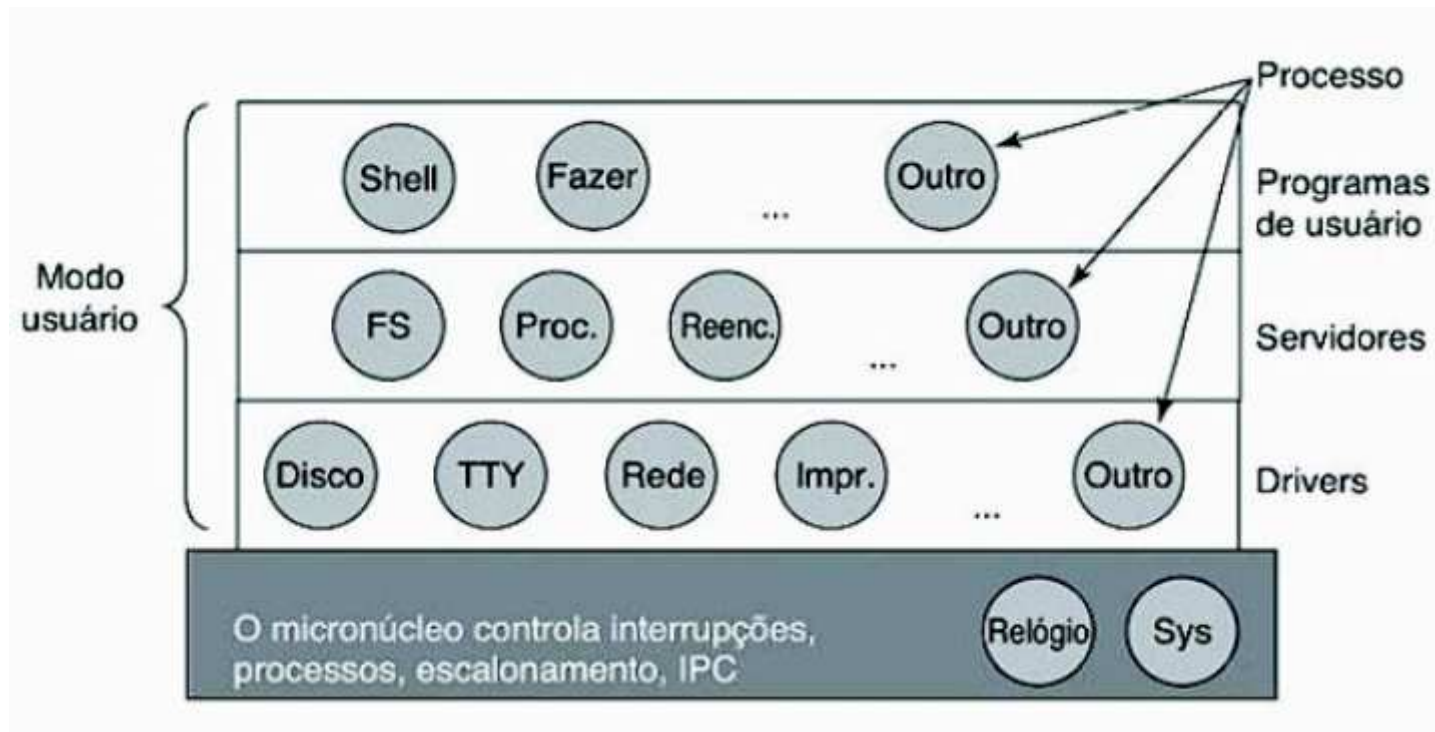


- Micronúcleo:
 - A idéia é colocar o mínimo possível do SO no modo núcleo, pois:
 - Um erro no núcleo pode ser fatal;
 - Aproximadamente 10 erros por 1000 linhas de código;
 - Um SO monolítico de 5 milhões de linhas contém aproximadamente 50 mil erros no núcleo.
 - Alcançar alta confiabilidade por meio da divisão do sistema em módulos pequenos, bem definidos, e apenas um módulo destes, o micronúcleo, é executado em modo núcleo.
 - Por exemplo, um erro em um *driver* de som não derruba todo o sistema.

Estrutura do Sistema Operacional



- Micronúcleo:



Estrutura do Sistema Operacional



- Sistemas Cliente-servidor:
 - O sistema operacional é dividido em partes, sendo cada uma responsável por oferecer um conjunto de serviços, como serviços de arquivo, serviços de criação de processos, serviços de memória e serviços de escalonamento, etc.
 - Sempre que uma aplicação deseja algum serviço ela solicita ao processo responsável.
 - A aplicação que solicita o serviço é chamada de cliente, enquanto o processo que responde à aplicação é chamado de servidor.

Estrutura do Sistema Operacional

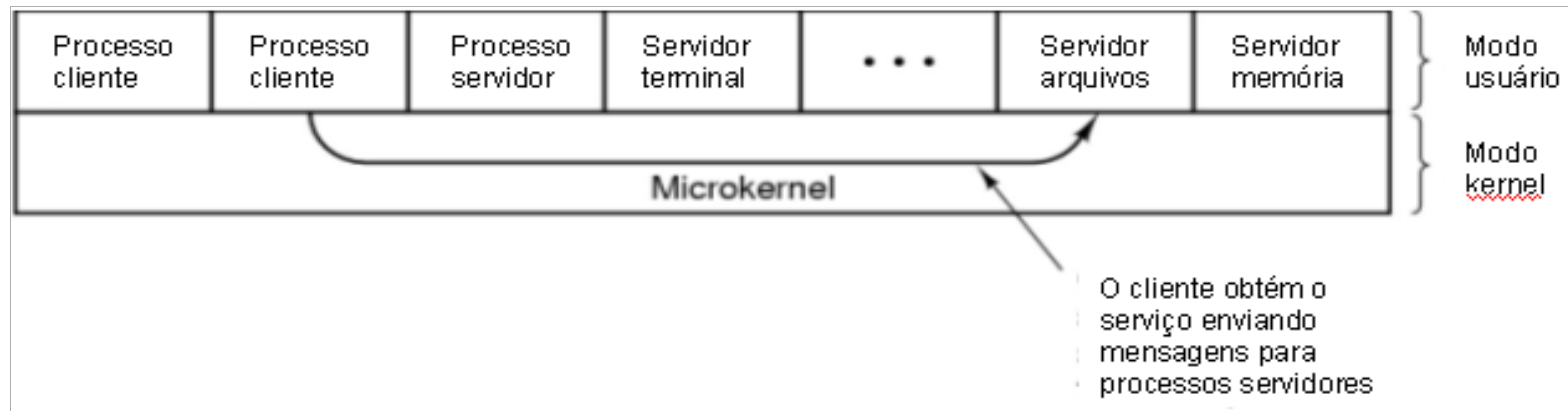


- Sistemas Cliente-servidor:
 - Processos:
 - Cliente: solicita o serviço
 - Servidor: executa em modo usuário, não tem acesso direto ao hardware
 - Na ocorrência de problemas o sistema não é completamente derrubado
 - Kernel: gerencia a comunicação (troca de mensagens) entre clientes e servidores
 - Apenas o núcleo do sistema executa no modo Kernel
 - Este modelo permite que os servidores executem em modo usuário, ou seja, não tenham acesso direto a certos componentes do sistema.

Estrutura do Sistema Operacional



- Sistemas Cliente-servidor



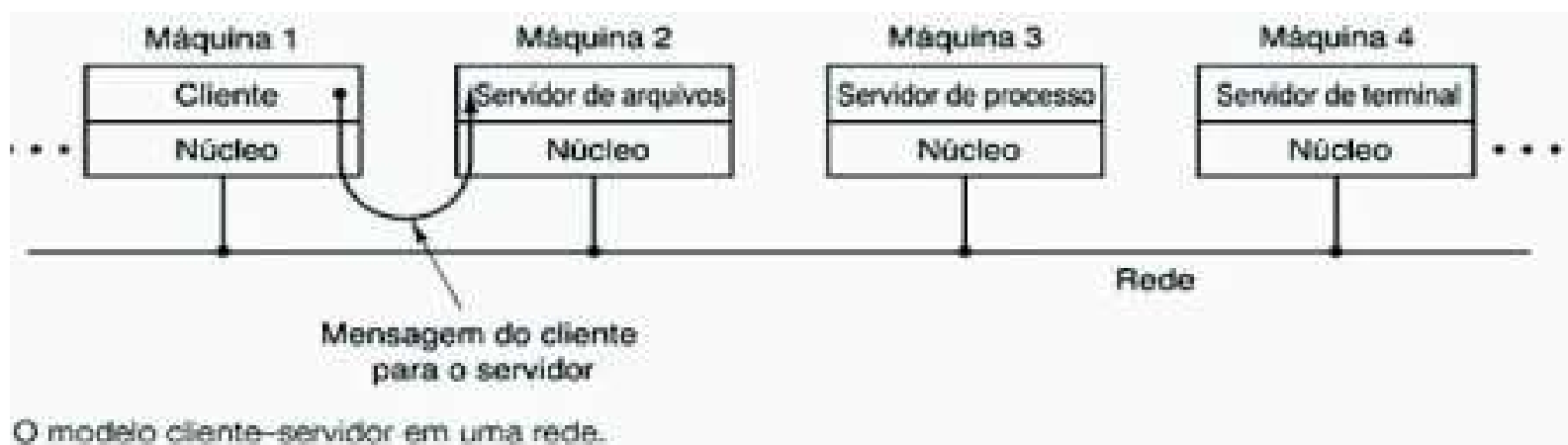
Fonte: Tanenbaum [1]

Estrutura do Sistema Operacional

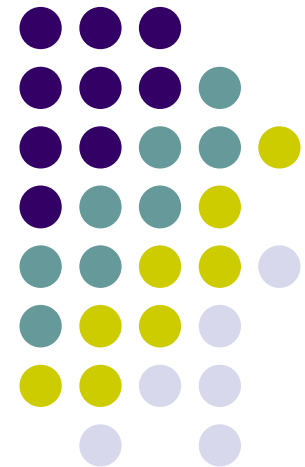


- **Sistemas Cliente-servidor**

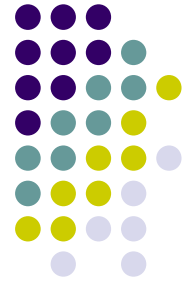
- Como os servidores se comunicam através de troca de mensagens, não importa se os clientes e servidores estão sendo processados em um sistema com um único processador, com múltiplos processadores (fortemente acoplados ou fracamente acoplados).



Um Pouco de História



Evolução dos Sistemas Operacionais



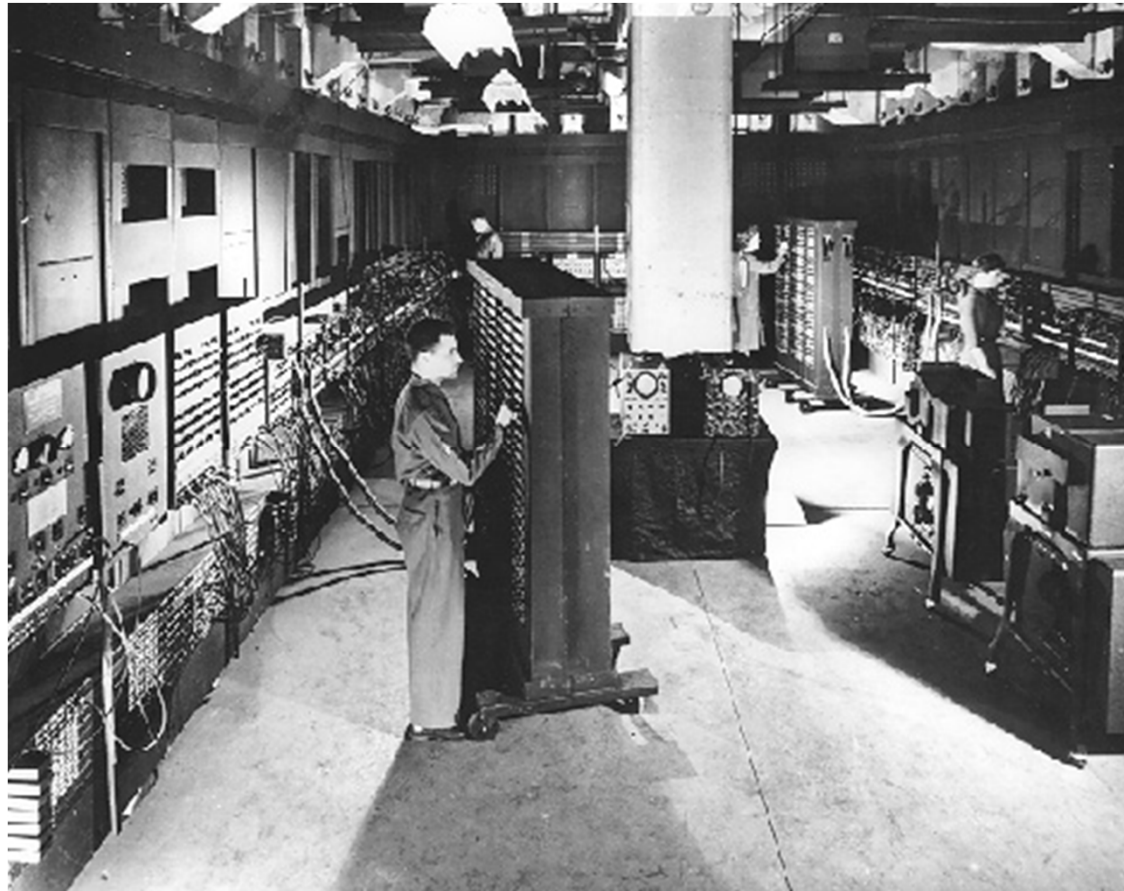
- Acontece basicamente devido a:
 - Evolução do hardware existente
 - Surgimento de novo hardware
 - Oferta de novos serviços
 - Interface gráfica
 - Ferramentas de gerenciamento de recursos
 - Necessidade de reajustes
 - Um S.O. pode conter falhas...
- Está relacionada diretamente com a arquitetura dos computadores

História dos Sistemas Operacionais



- Primeira Geração (1945 – 1955): Válvulas e Painéis
 - Máquinas enormes, com milhares de válvulas.
 - Programação em código absoluto (válvulas desligadas ou ligadas).
 - No início dos anos 50 aparecem os cartões perfurados e as impressoras de linha.
 - Não existe S.O.
 - O programador utiliza diretamente a console da máquina para executar seu programa.
 - Sistema mono-usuário.
 - Um único grupo de pessoas concebia, construía, programava, utilizava e fazia a manutenção das máquinas.

História dos Sistemas Operacionais



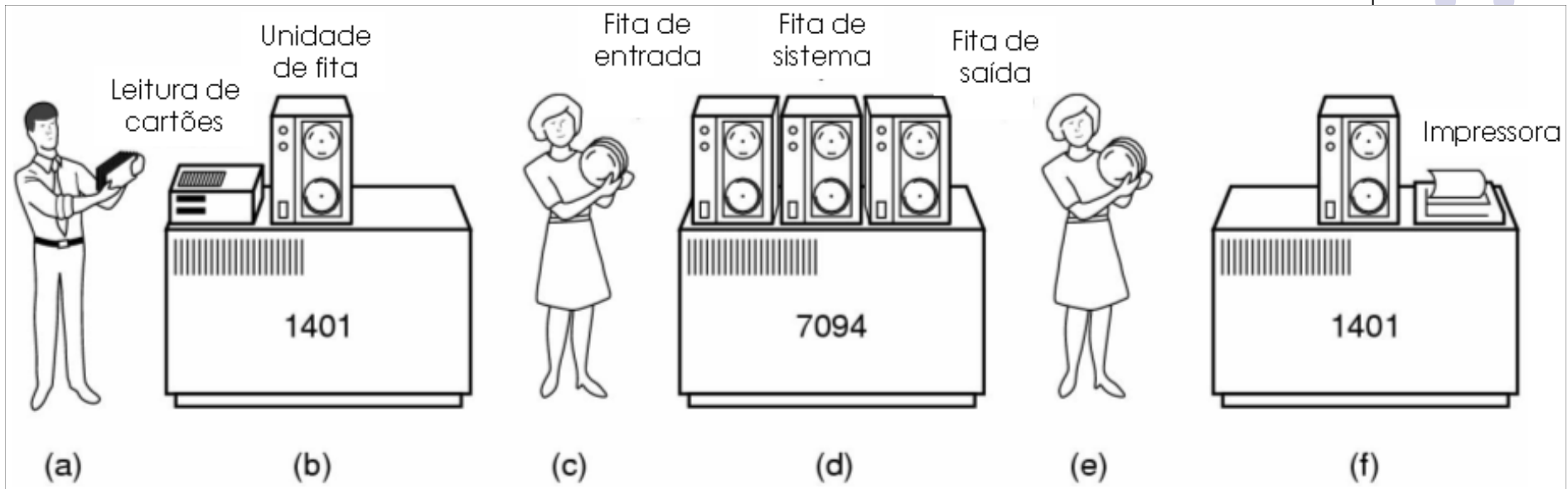
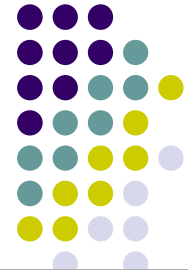
Eniac, 1946, Universidade de Pensilvânia, EUA

História dos Sistemas Operacionais



- Segunda Geração (1955 – 1965): – Transistores e Sistemas *Batch*
 - Aumento de velocidade de processamento, graças à troca de válvulas por transistores (semicondutores);
 - Criação das memórias magnéticas, o que permitiu o acesso mais rápido aos dados e maior capacidade de armazenamento.
 - Existe um pseudo-SO que lê os cartões, executa o programa e escreve os resultados na impressora. Sistema mono-usuário.
 - Aparecimento de Linguagens de Programação como Assembly e FORTRAN.
 - Início do domínio IBM (*mainframes*).
 - Processamento de *Jobs* era feito em Lote (*batch*).

História dos Sistemas Operacionais



Fonte: Tanenbaum [1]

- Segunda Geração 1955 - 1965
 - a. Trazer cartões para 1401
 - b. Ler os programas e gravar o lote na fita
 - c. Operador insere fita na unidade
 - d. Passar a fita para 7094 – processamento
 - e. Operador colocar a fita em 1401 para impressão

História dos Sistemas Operacionais

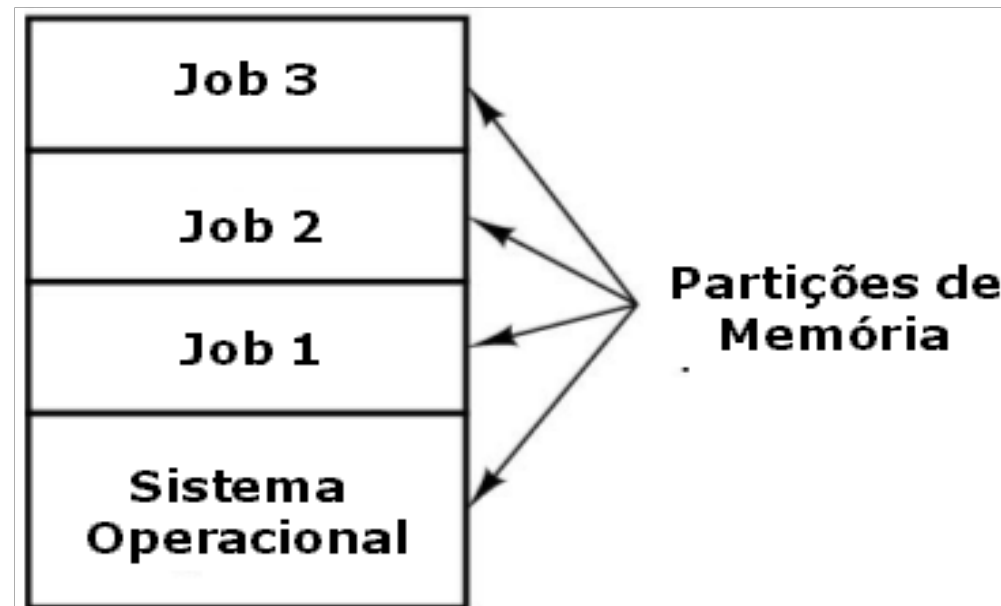


- Terceira Geração (1965 – 1980): Circuitos Integrados e Multiprogramação
 - Surgem os discos
 - Surgem duas linhas diferentes de máquina: comercial e cálculo científico.
 - IBM lança máquinas de menor porte que os *mainframes* (série IBM 360)
 - Família 360 foi a primeira série a usar tecnologia SSI (*Short Scale Integration*).
 - Surgimento de SOs mais voltados ao usuário (como o OS/360 da IBM)
 - SPOOL (*Simultaneous Peripheral Operation On Line*): consiste da leitura dos cartões direto para o disco, visando a execução desse job, tão logo o atual liberasse o processador.

História dos Sistemas Operacionais



- Terceira Geração (1965 – 1980): Circuitos Integrados e Multiprogramação
 - Criação da técnica de **Multiprogramação**

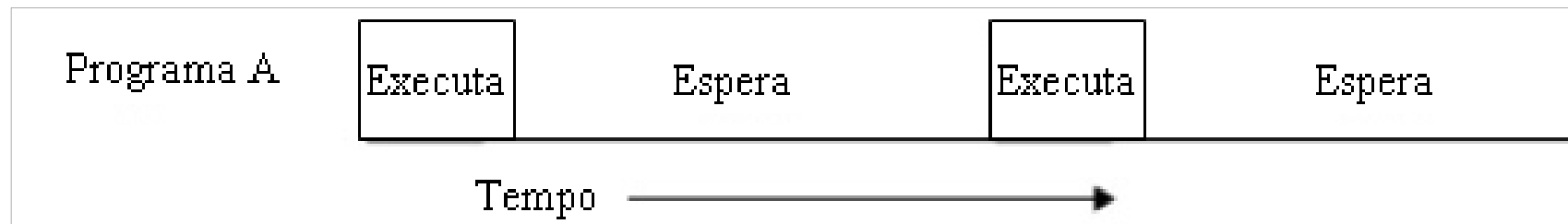


Fonte: Tanenbaum [1]

História dos Sistemas Operacionais



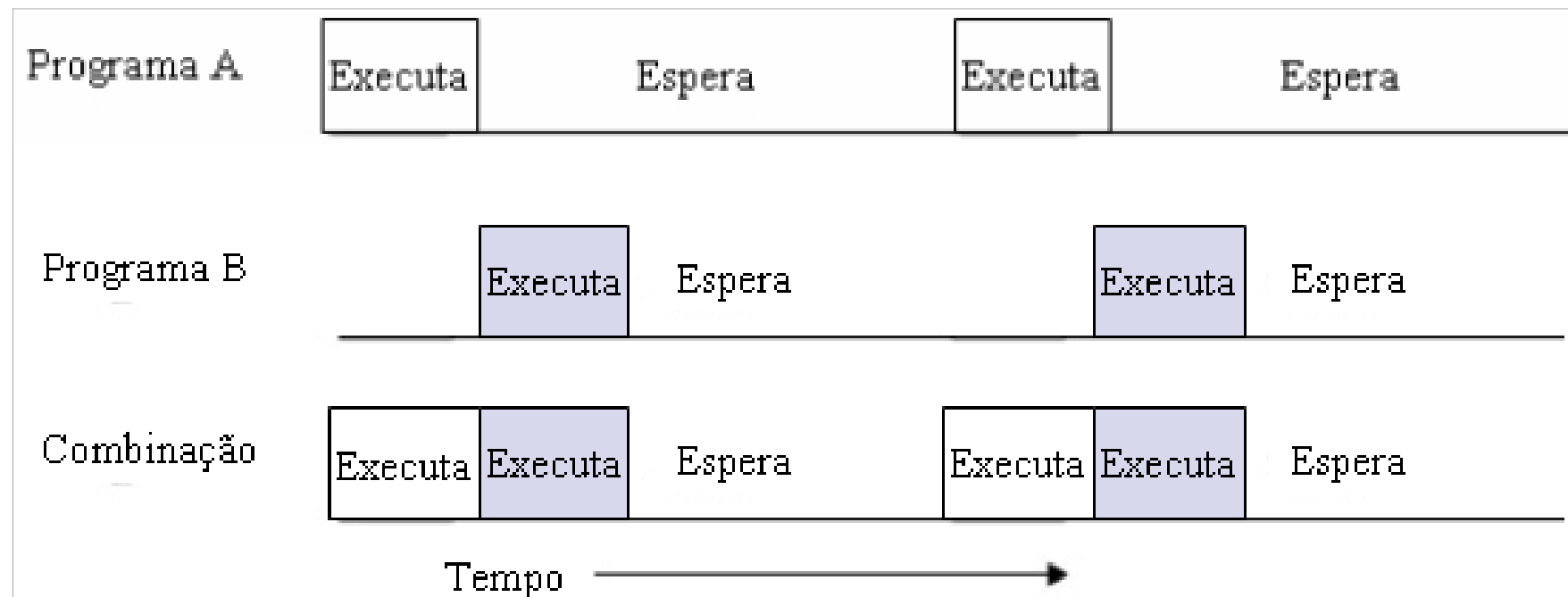
- Terceira Geração (1965 – 1980)
 - Monoprogramação
 - O processador aguarda a utilização do recurso de E/S para completar o procedimento.



História dos Sistemas Operacionais



- Terceira Geração (1965 – 1980)
 - Multiprogramação
 - O processador manipula vários *jobs* ao mesmo tempo.



História dos Sistemas Operacionais



- Terceira Geração (1965 – 1980)
 - Através da criação do terminal de vídeo e teclado foi possível a interação *on-line*;
 - Surgimento dos sistemas de **Compartilhamento de tempo** (*Timesharing*)
 - Tem por objetivo oferecer ao usuário um tempo de resposta menor
 - Criação do UNIX por Ken Thompson, do Bell Labs: um sistema multiusuário, multiprogramado e extremamente sofisticado para sua época.
 - Gerência de memória: para executar um programa que não cabe inteiramente na memória são propostas duas soluções:
 - Overlay; e
 - Memória virtual.

História dos Sistemas Operacionais



- Quarta Geração (A partir de 1980):
Computadores Pessoais
 - Circuitos LSI (*Large Scale Integration*), com milhares de transistores encapsulados em um *chip*;
 - Nasce a idéia do computador pessoal;
 - O custo do computador caiu bastante;
 - Houve a disseminação dos microcomputadores;
 - Sistemas Operacionais e Programas de Aplicação voltados ao usuário final (*user-friendly*);
 - Aparecimento da arquitetura RISC.

História dos Sistemas Operacionais



- Quarta Geração (A partir de 1980):
 - Disseminação de ambientes distribuídos: Redes de Computadores e Sistemas Distribuídos;
 - Criação de Sistemas Operacionais de Redes e Sistemas Operacionais Distribuídos (p.ex., Amoeba);
 - Aparecimento das Máquinas Paralelas, como o SP2 da IBM;
 - Surgimento de arquiteturas híbridas CISC/RISC, como o Pentium Pro e seus sucessores;
 - Redes de computadores:
 - Os usuários estão conscientes da existência de um conjunto de máquinas conectadas à rede.
 - Sistemas distribuídos:
 - As diferentes máquinas que compõem o sistema são percebidas pelo usuário como uma única máquina virtual.

História dos Sistemas Operacionais



- Quarta Geração (A partir de 1980):
 - Sistemas multiprocessados
 - São conhecidos também como sistemas paralelos;
 - Possuem mais de um processador;
 - Compartilham barramento, *clock*, e em algumas situações, memória e dispositivos de entrada/saída;
 - São sistemas fortemente acoplados (multiprocessadores)
 - Vantagens:
 - Aumento do *throughput*;
 - Aumento da confiabilidade (tolerância a falha);
 - ...

História dos Sistemas Operacionais



- Quinta Geração ??? – ???
 - ????
 - ????
 - ????



Referências Bibliográficas

- [1] TANENBAUM, A. S. *Sistemas Operacionais Modernos*, São Paulo: Pearson, 2ª Edição, 2003.
- [2] SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G.; *Sistemas Operacionais com Java*, Rio de Janeiro: Campus, 7ª Edição, 2008.
- [3] Stallings, William. **Operating Systems: internals and Design Principles**. 4ª ed. New Jersey: Prentice Hall. 2001
- DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R.; *Sistemas Operacionais*, São Paulo: Pearson, 3ª Edição, 2005.
- [4] The www.ieee.org
- [5] Machado, F. B., Maia, L. P. **Arquitetura de Sistemas Operacionais**. 3ª ed. LTC Editora. Rio de Janeiro. 2002.