

Introdução

- SO: por que é necessário ?
- O que é um SO ?
- Evolução dos SOs
- Arquitetura de SO
- Funções dos SOs
- Tipos de Sistemas Operacionais

Por que ?

- Sistemas de computadores modernos são compostos por diversos dispositivos:

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



Gerenciamento
Manipulação



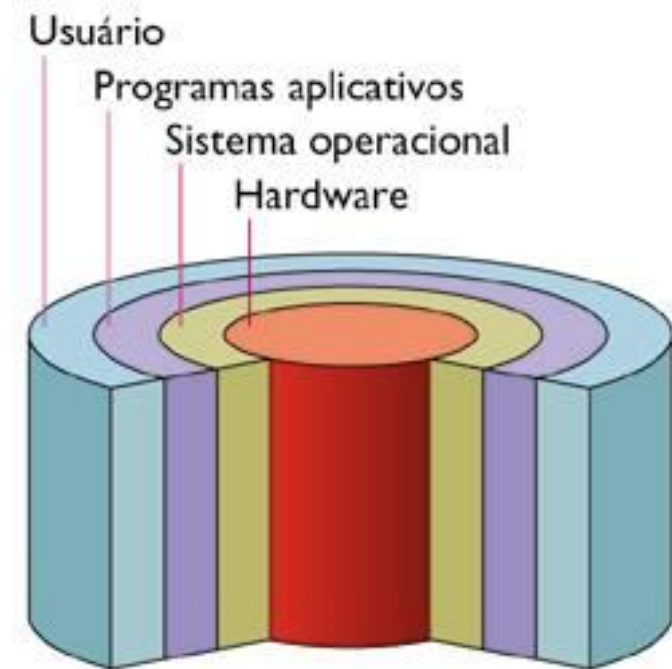
Alta Complexidade



Sistemas Operacionais

O que é um SO ?

- Programa que age como intermediário entre o usuário e o hw do computador
- Hardware
 - Processador
 - Memória Principal
 - Dispositivos de E/S
- Software
 - Programas de Aplicação
 - Programas do Sistema



Evolução dos Sistemas Operacionais

- Primeira Geração ... Atual

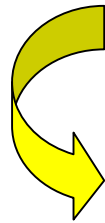


1ª - 2ª Geração

- batch → monitor residente
- fitas magnéticas → discos (SPOOLING)



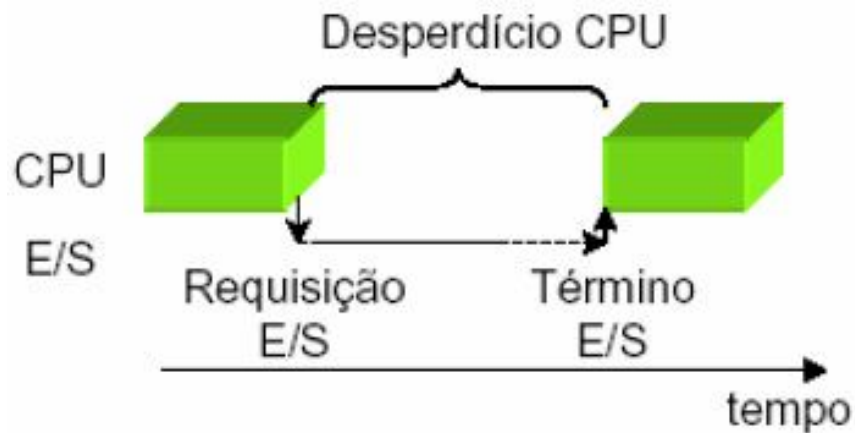
{ job queue
job pool
job scheduling



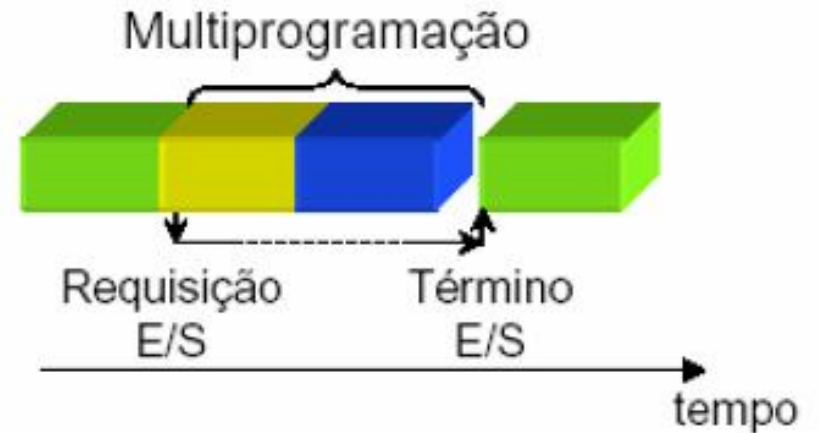
multiprogramação

2ª - 3ª Geração (1)

- multiprogramação



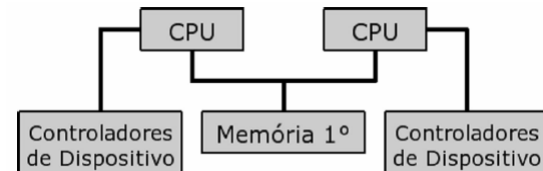
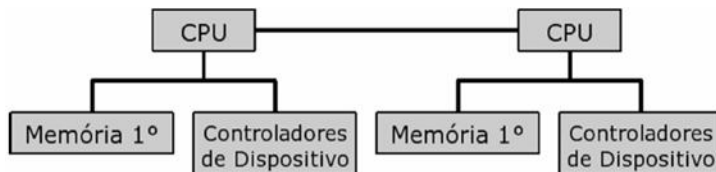
monoprogramado



multiprogramado

2ª - 3ª Geração (2)

- time-sharing
 - CPU scheduling + multiprogramação
 - MUX do tempo de CPU
 - alternância
 - time-slice
- multiprocessamento



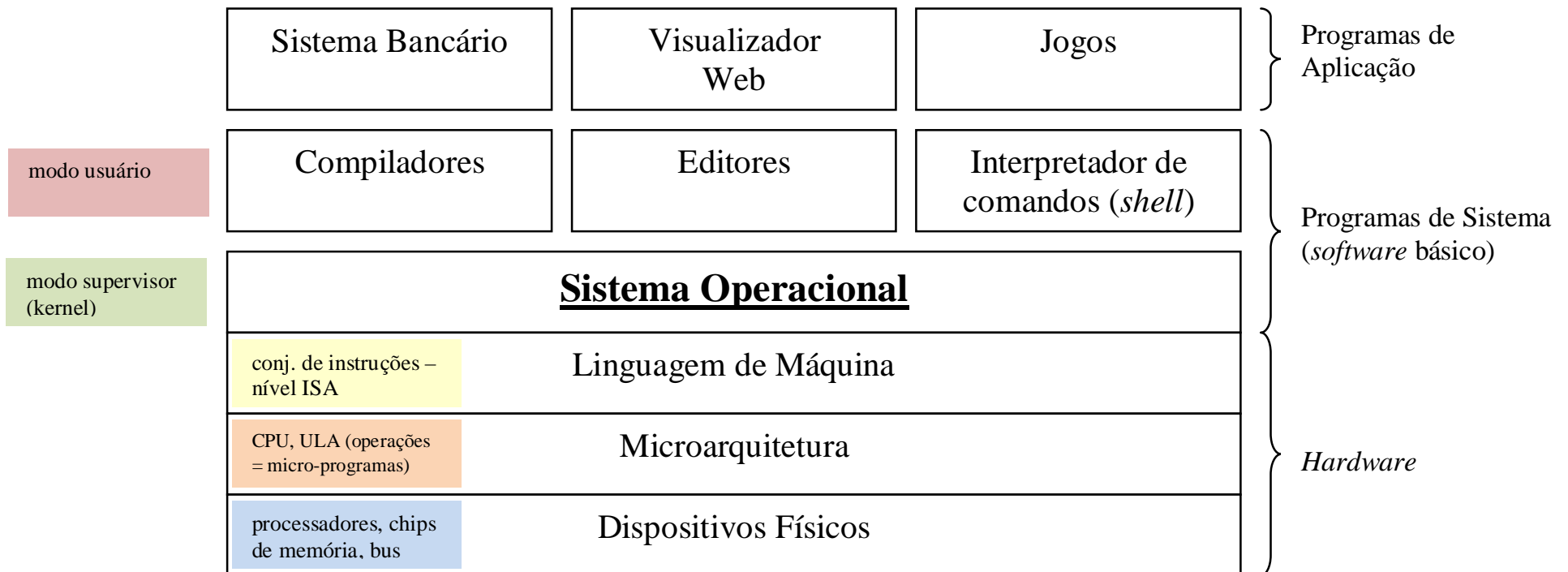
3ª - 4ª Geração

- problema: overhead do SO
 - Engenharia de SW
 - regras p/ construção de sistemas
- 4ª Geração
 - SO de redes LANs e WANs
 - Modelo C/S
 - Processamento distribuído

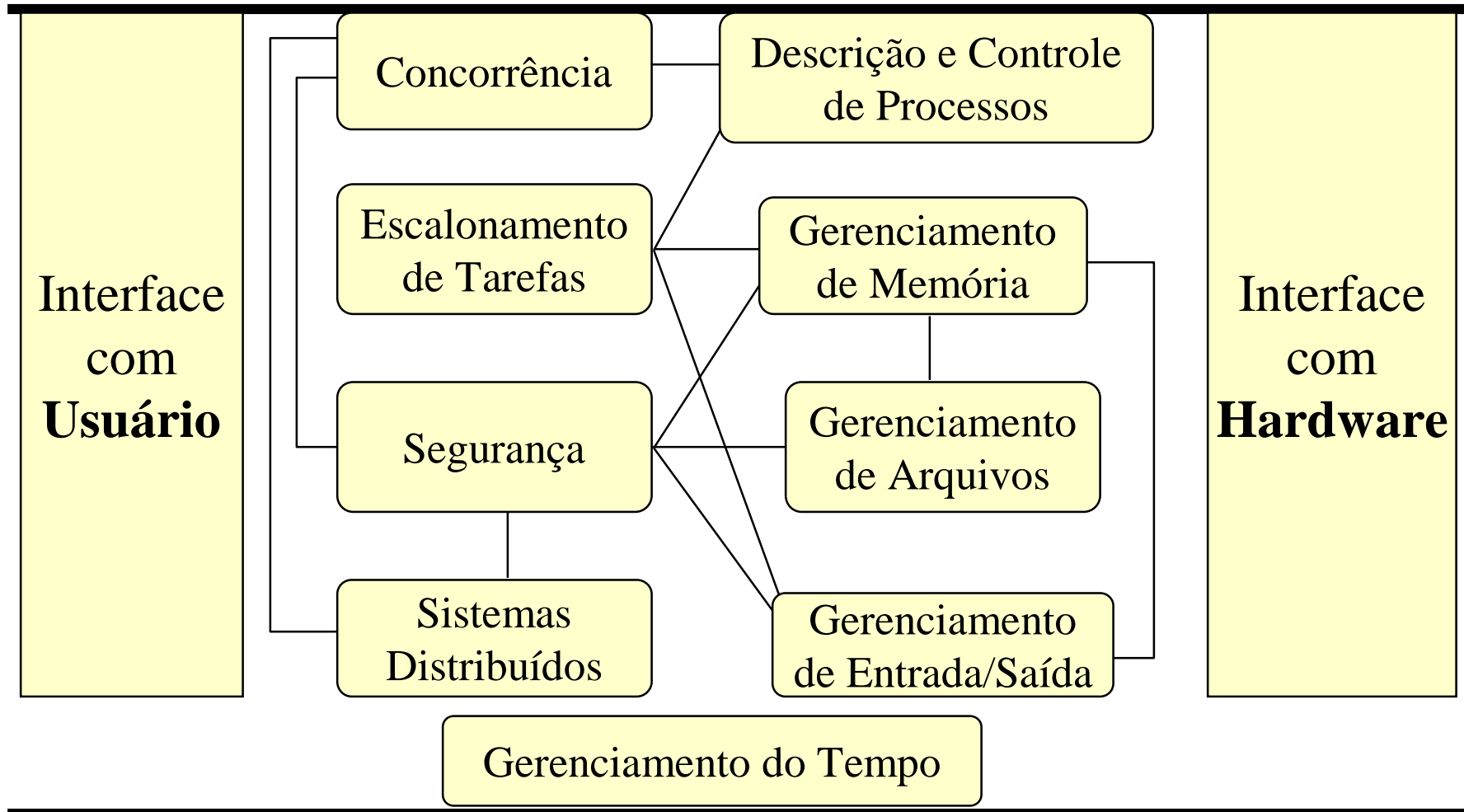
5ª - 6ª Geração ??

- SO's distribuídos
 - subcomputações
 - compartilhamento de recursos físicos e lógicos
- Computação móvel e ubíqua
- Cloud computing, IoT, ...

Arquitetura



“Des” Organização de um SO



SO

- Formas de “ver”:
 - “fiscal” que controla os usuários
 - “juiz” que aloca recursos entre os usuários
 - Formas de atuação:
 - Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário
 - Como gerente de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador
-

SO como máquina estendida

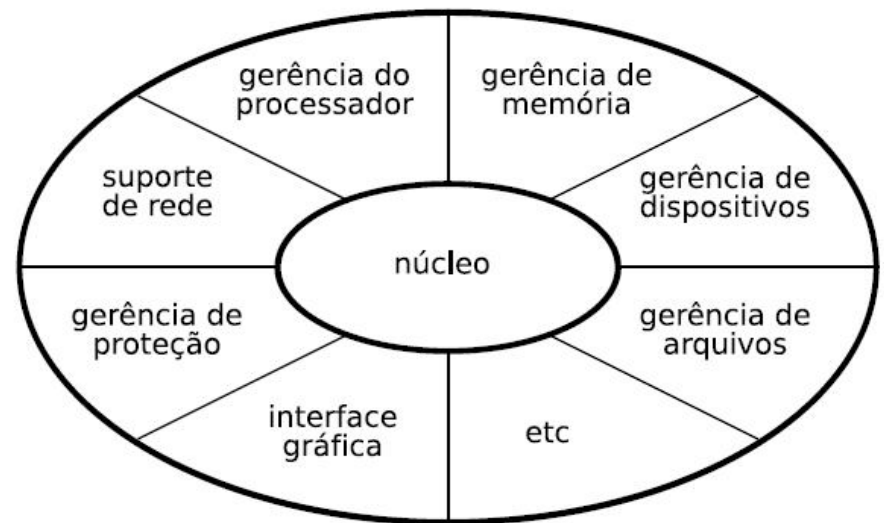
- Ex.: como é feita a entrada/saída de um disco flexível – tarefa: Leitura e Escrita
 - SO: baixo nível de detalhes
 - Número de parâmetros
 - Endereço de bloco a ser lido
 - Número de setores por trilha
 - Modo de gravação
 - Usuário: alto nível – abstração simples
 - Visualização do arquivo a ser lido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

SO como gerente de recursos

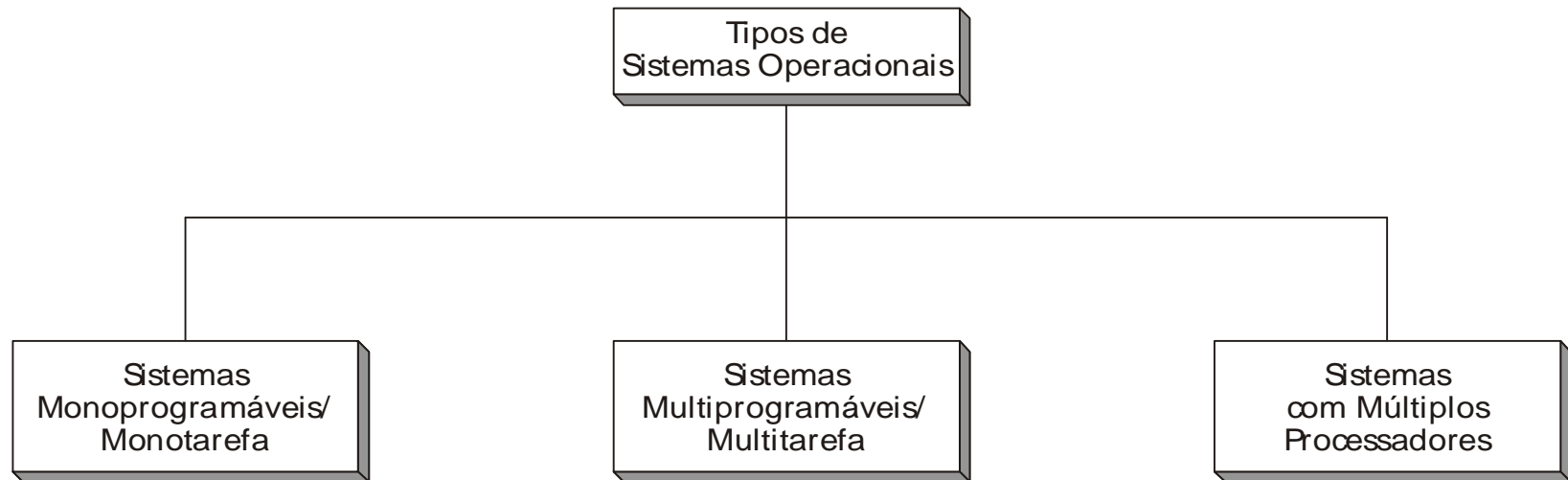
- Gerencia todos os dispositivos e recursos disponíveis no computador
 - Ex.: se 2 processos querem acessar um mesmo recurso, p. ex. uma impressora, o SO é responsável por estabelecer uma ordem p/ que ambos possam realizar sua tarefa
 - Uso do HD
 - Uso da memória
- Coordena a alocação controlada e ordenada dos recursos

Funções do Sistema Operacional

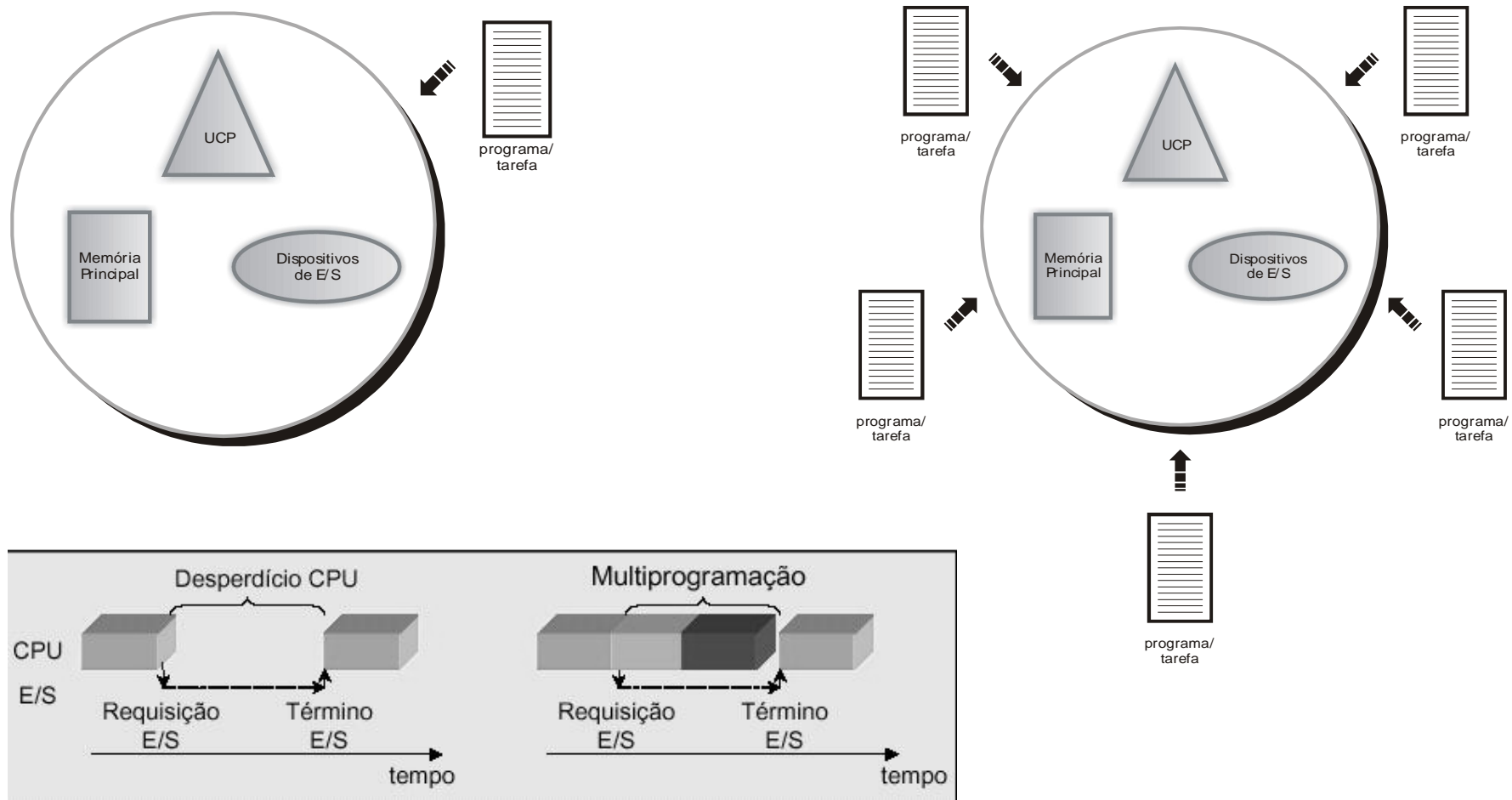
- Gerenciamento de Processos
- Gerenciamento de Memória
- Sistema de Arquivos
- Entrada e saída de dados



Tipos de sistemas operacionais

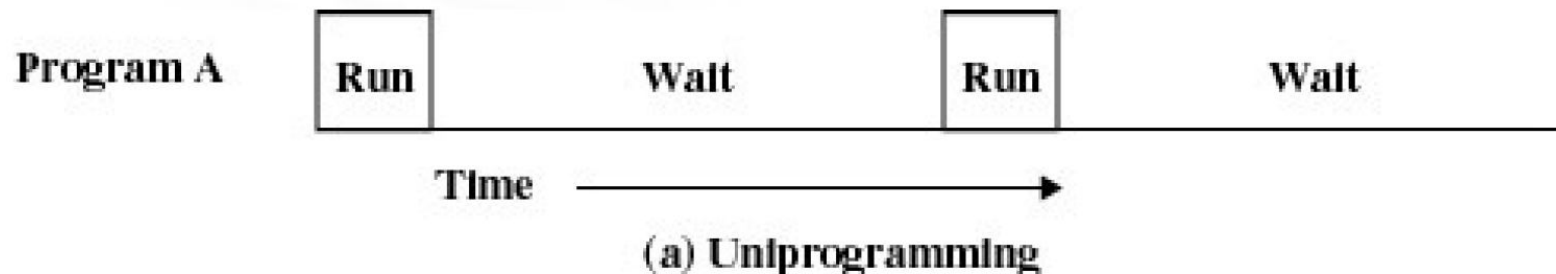


Mono x Multiprogramação



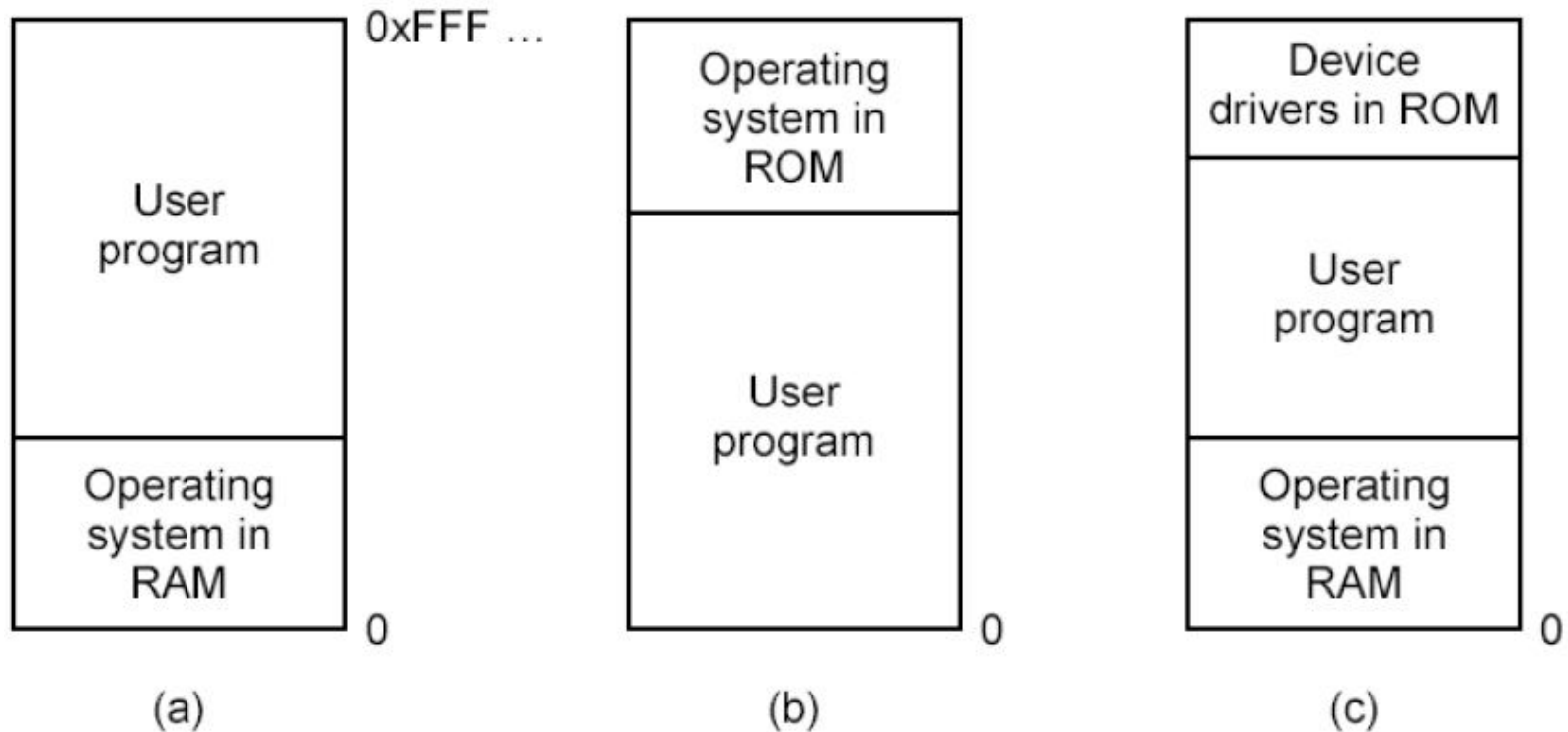
Monoprogramação (1)

- execução exclusiva de um único programa por vez
 - recursos dedicados
 - subutilização (memória, CPU, ...)
 - CPU ociosa
- não exige proteção de memória
- baixa complexidade de implementação



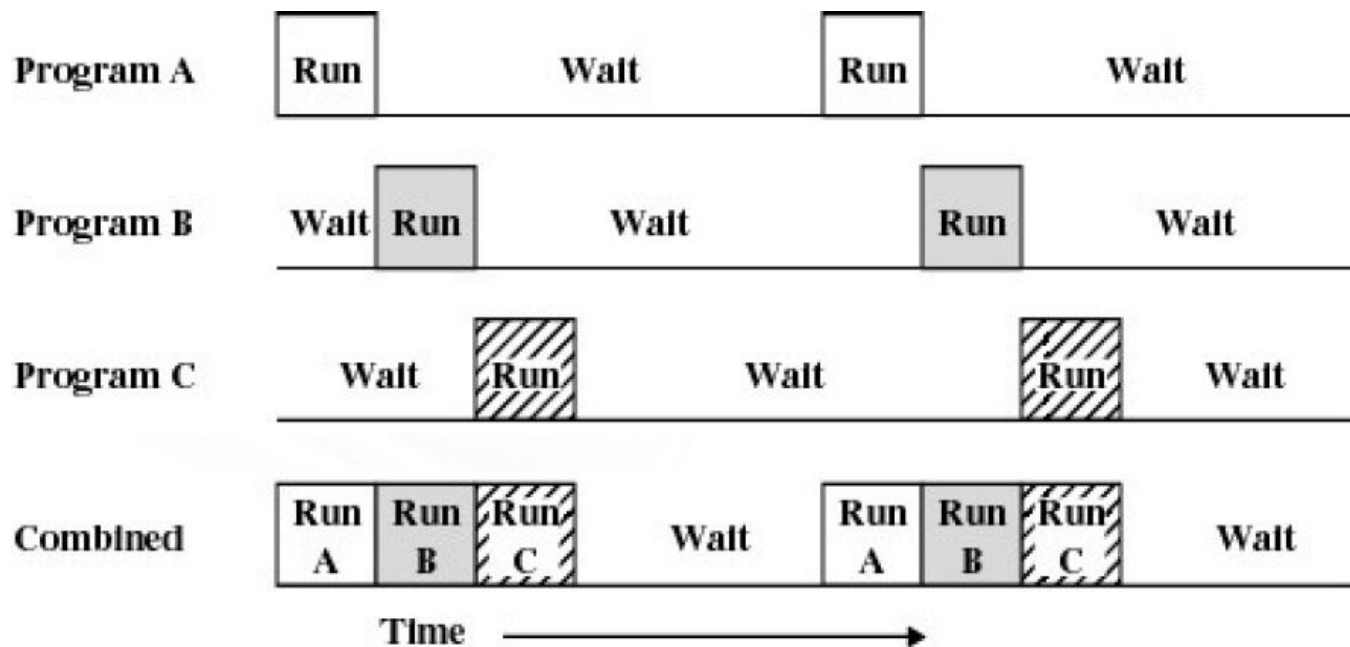
Monoprogramação (2)

- Organização da memória



Multiprogramação (1)

- Vários programas competindo pelos recursos do sistema

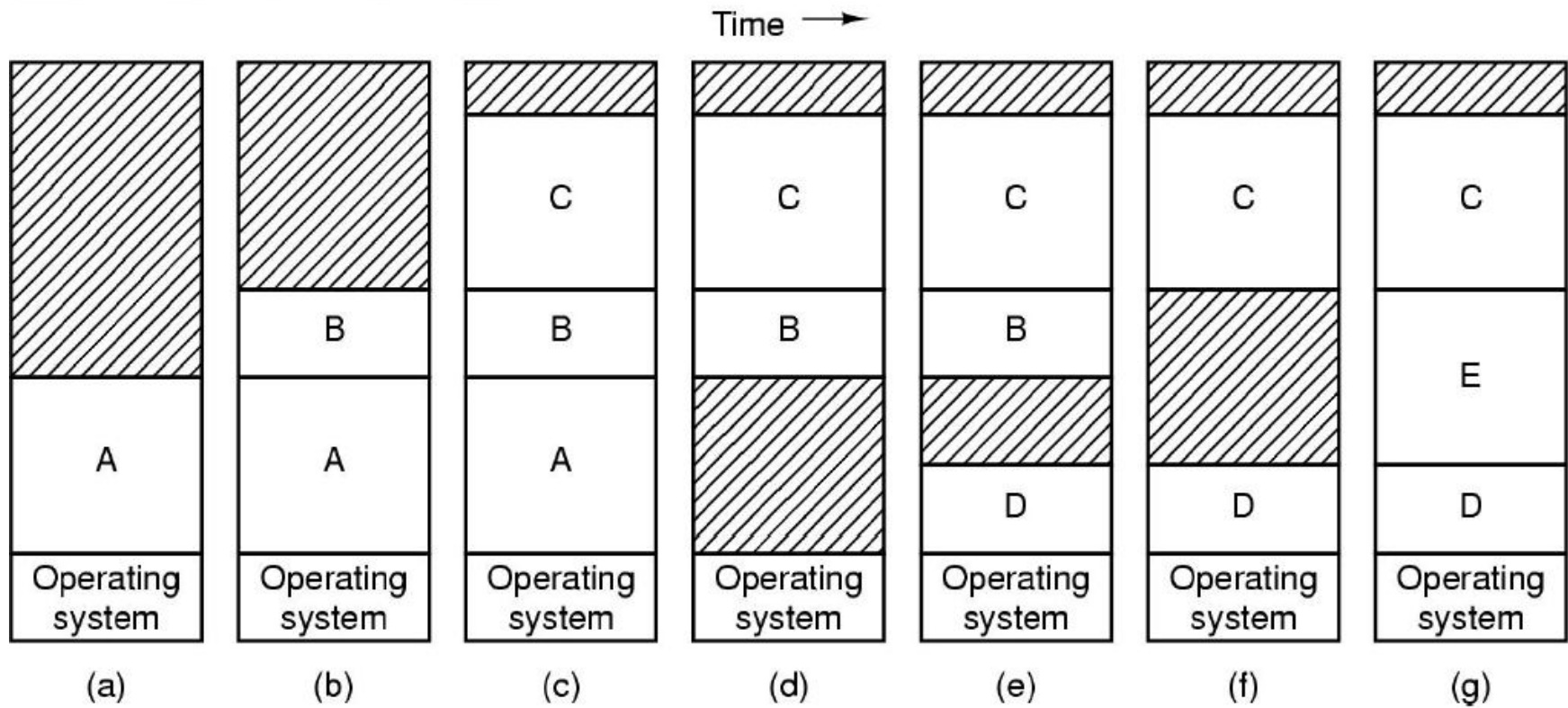


(c) Multiprogramming with three programs

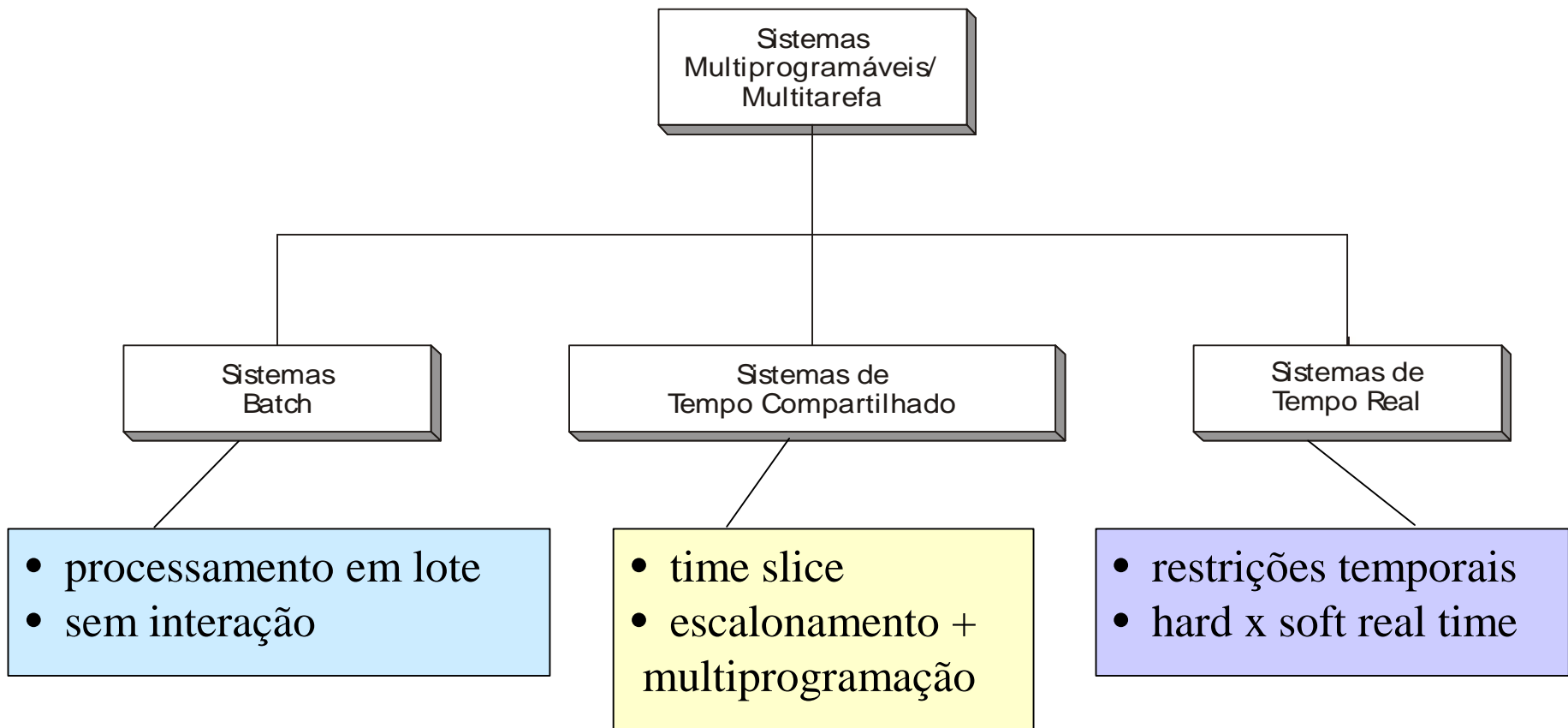
Multiprogramação (2)

- Objetivo: manter mais de um programa em execução “simultaneamente”
 - ilusão de que cada programa possui uma máquina dedicada
- Ideia: aproveitar o tempo ocioso da CPU durante a E/S
 - Maximização do uso de processador e memória
 - Maior taxa de utilização do sistema como um todo
- exige:
 - proteção de memória
 - mecanismo de interrupção

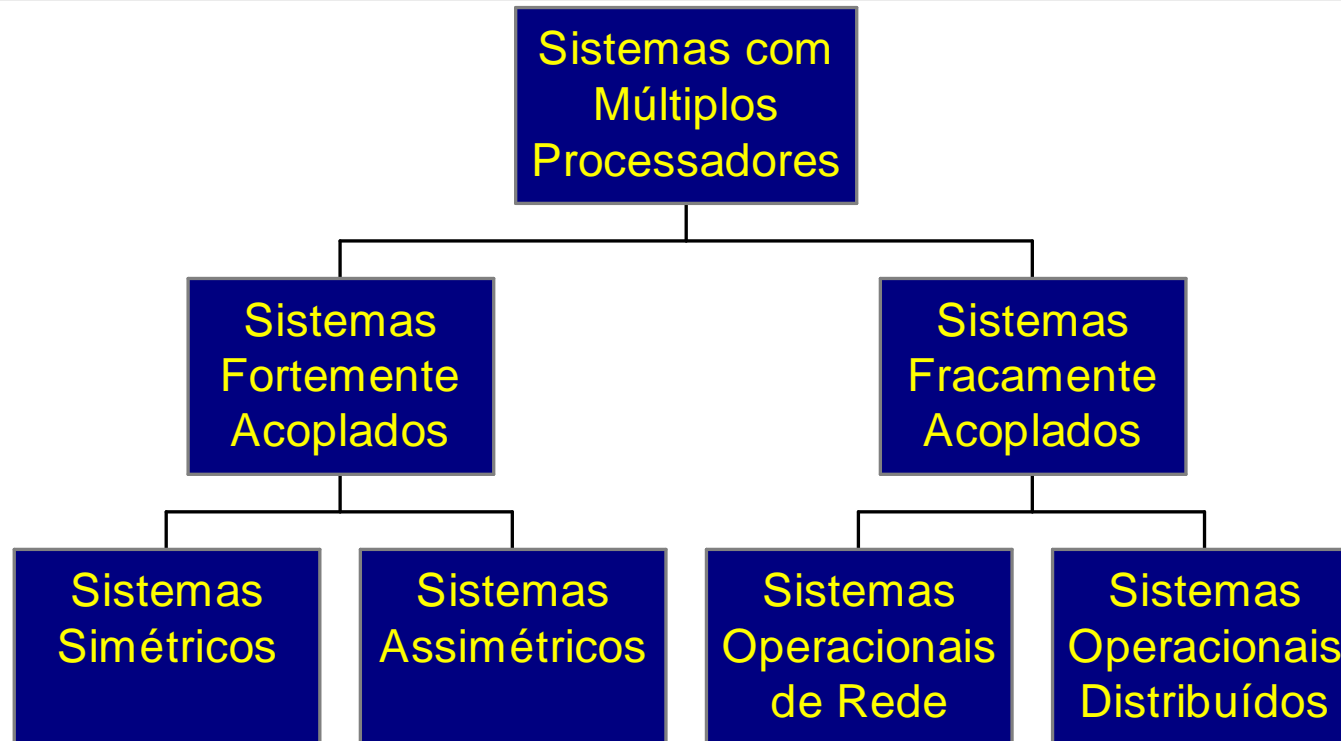
Multiprogramação (3)



Sistemas multiprogramáveis/multitarefa: gerência das aplicações



Sistemas com Múltiplos Processadores



- Forma de comunicação entre CPUs e grau de compartilhamento de memória e dispositivos de E/S

Sistemas fortemente acoplados (*tightly coupled*) (1)

- 2 ou mais processadores compartilhando memória
- controle: um único SO
- usados em sistemas **CPU-bound** para processamento de solução de um único problema

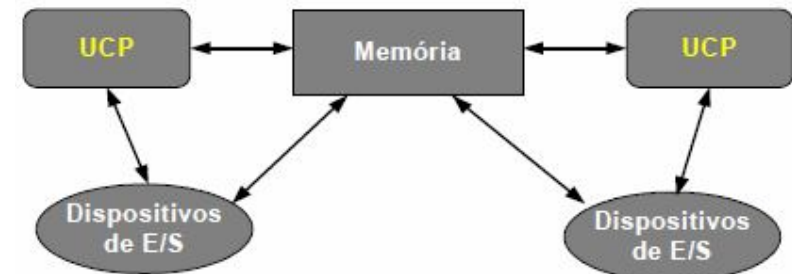
😊 desempenho

- execução de tarefas em paralelo

😊😞 custo

- Compartilhamento

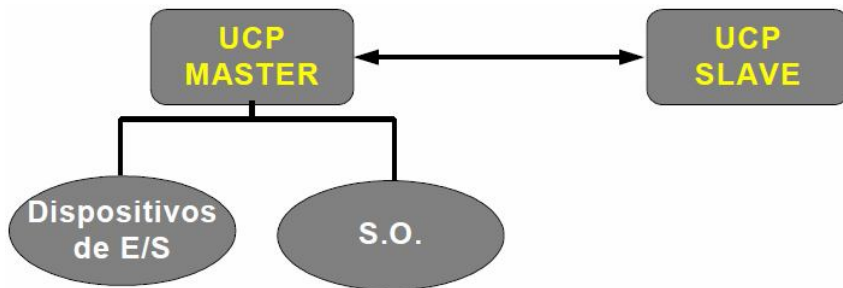
- classificação (conforme simetria entre processadores)
 - processadores podem ou não executar as mesmas tarefas
 - Assimétricos x Simétricos



Sistemas fortemente acoplados (2)

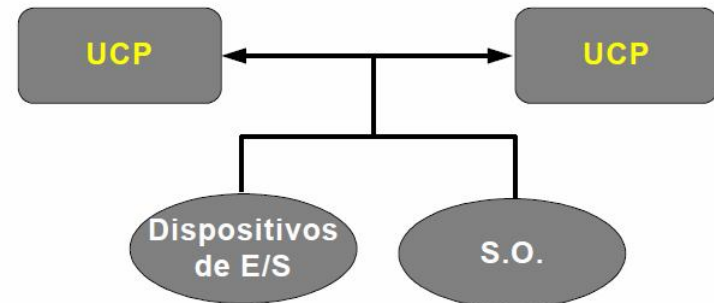
Assimétricos

- modelo master/slave
- cada processador recebe uma tarefa específica



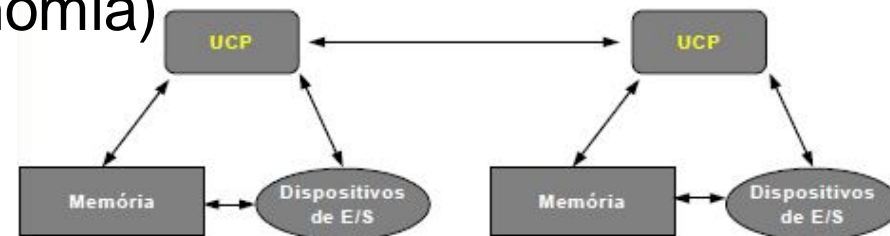
Simétricos

- cada processador roda cópia idêntica do SO
- balanceamento de carga
- implementação mais complexa



Sistemas fracamente acoplados (*loosely coupled*) (1)

- 2 ou mais sistemas de computação conectados por linhas de comunicação
 - sistemas independentes, s/ compartilhamento de memória ou clock
- comunicação: troca de mensagens
- ☺ aceleração da computação
 - “subcomputações” podem rodar concorrentemente
 - sobrecarga de jobs → balanceamento de carga
- ☺ confiabilidade (redundância, autonomia)
- classificação
 - SO de rede e SO distribuído



Sistemas fracamente acoplados (2)

- *Sistema operacional de rede*
 - usuários conhecem a localização dos recursos que estão utilizando; acessam máquinas remotas
 - cada máquina roda o seu próprio SO local
- *Sistema operacional distribuído*
 - usuários não sabem onde os seus programas estão executando, nem onde seus arquivos estão localizados
 - tudo é gerenciado pelo SO

Outros SOs (1)

- SO de embarcados
 - restrições quanto ao espaço de **memória** e consumo de **energia**
 - em geral possuem restrições temporais
 - apropriados para computadores de mão
 - também executam em dispositivos que não são computadores (fornos de microondas, celulares, ...)
 - Ex: VxWorks

Outros SOs (2)

- SO de cartões inteligentes (smart cards)
 - menores SOs existentes
 - severas restrições quanto a memória e energia
 - em geral são orientados a Java (ROM do cartão possui interpretador Java)
 - Ex.: JavaOS