

Introdução

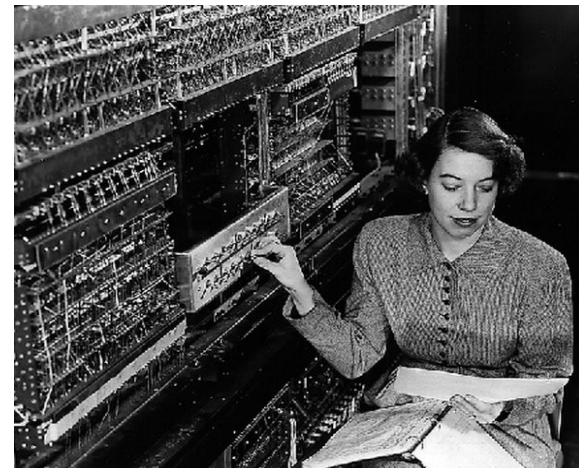
Fonte:
Tanenbaum cap 1

- Evolução dos SOs
- Tipos de Sistemas Operacionais
- Características dos sistemas operacionais



1ª Geração (+- 1945 – 1955)

- Primeiros computadores: **válvulas**
 - ENIAC (Electronic Numerical Integrator and Computer)
 - cálculos balísticos
 - EDVAC (Electronic Discrete Variable Automatic Computer)
 - Uso: universidades e órgãos militares

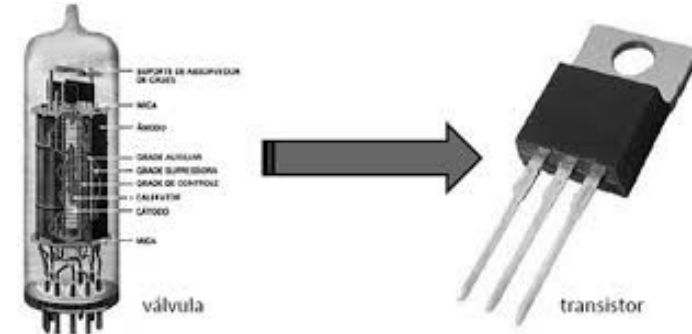


1ª Geração

- Usuário é o programador e o operador da máquina
 - SOs **não** existiam
 - Programador interagia diretamente com o HW
 - Processamento serial – 1 usuário e 1 programa por vez
 - Uso do computador precisava ser agendado (planilhas)
 - Tempo de configuração: carregar compilador, código fonte, ...
 - Definição de categorias de programas similares (filas de jobs) = **processamento em lote (batch)**
 - Operador manipulava o HW p/ executar os jobs, controlava a atribuição do HW e o andamento dos jobs
-

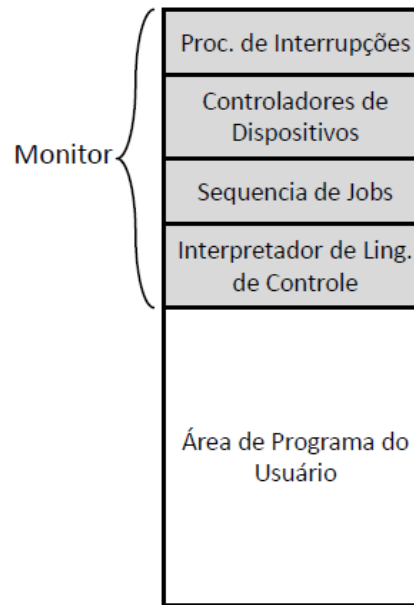
2ª Geração (+- 1956 – 1965)

- Criação dos **Transistores**
 - Velocidade de processamento
- Criação das memórias magnéticas
 - Acesso mais rápido
- Primeiras LPs
 - Assembly e Fortran
- Cartão perfurado



2ª Geração

- Processamento em lote (batch)
 - Evolução → automatizar o andamento dos lotes de Jobs
 - Monitor residente
 - Interpretador de JCL
 - 1º “SO”
 - Sequenciamento automático de Jobs



JCL (Job Control Language)

- Proteção de memória
- Temporização
- Instruções privilegiadas
- Interrupções

```
$JOB
$FTN
.
. } instruções FORTRAN
.
$LOAD
$RUN
.
. } dados
.
$END
```

3ª Geração (+- 1966 – 1980)

- Circuitos integrados (CIs)
 - Menor custo e dimensão
 - Desempenho
- Surgimento do SO Unix (em C)
- Sequenciamento automático de Jobs
 - Execução de apenas um job por vez!
 - Ideia: reduzir o desperdício de CPU devido ao I/O
 - Multiprogramação

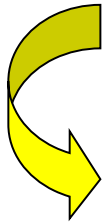


Multiprogramação (1)

- fitas magnéticas → discos



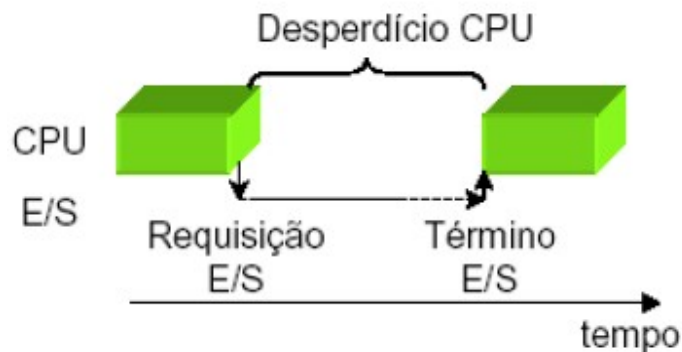
{ job queue
job scheduling
mecanismo de interrupção



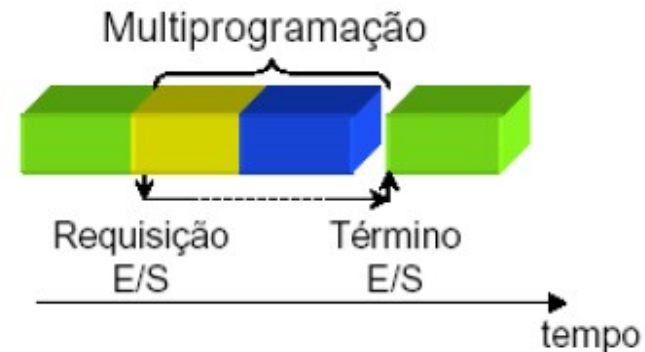
multiprogramação

Multiprogramação (2)

- Objetivo: maximizar o uso da CPU
 - Sistema **multiprogramado**: Aproveita desperdício de tempo de CPU com operações de E/S
- Memória é particionada e vários jobs são carregados ao mesmo tempo



monoprogramado



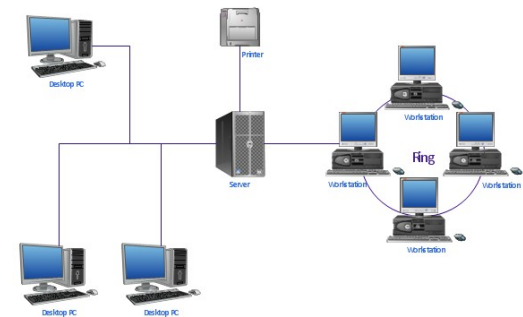
multiprogramado

Sistema time-sharing

- Multiprogramação → time-sharing
- Usuário com ilusão de possuir máquina dedicada
 - Divisão do tempo de processamento entre os usuários (MUX do tempo de CPU)
 - Alternância
 - Time slice
 - Tempo de resposta é importante

4ª Geração (+- 1981 - 1990)

- Aperfeiçoamento dos CIs (LSI, VLSI, ...)
- Microcomputadores (PCs): família Intel
- Microsoft e DOS (Disk Operating System)
- Estações de trabalho (monousuárias)
 - Multitarefa
- Multiprocessadores
- SO de rede e distribuídos



5ª Geração (+- 1995(?) - ...)

- Arquitetura cliente-servidor
- Processamento distribuído
 - Multiprocessadores não convencionais
- Linguagem natural
- Segurança, gerência e desempenho do SO e da rede
- Consolidação dos sistemas de interfaces gráficas
 - Interação com usuários mais flexível



Computadores portáteis



Data-centers/ cloud

5ª - 6ª Geração: ??

- Sistemas embarcados, vestíveis, implantados, interconectados



Car Computer



watch Computer



Wearable Comp.



Data Centers

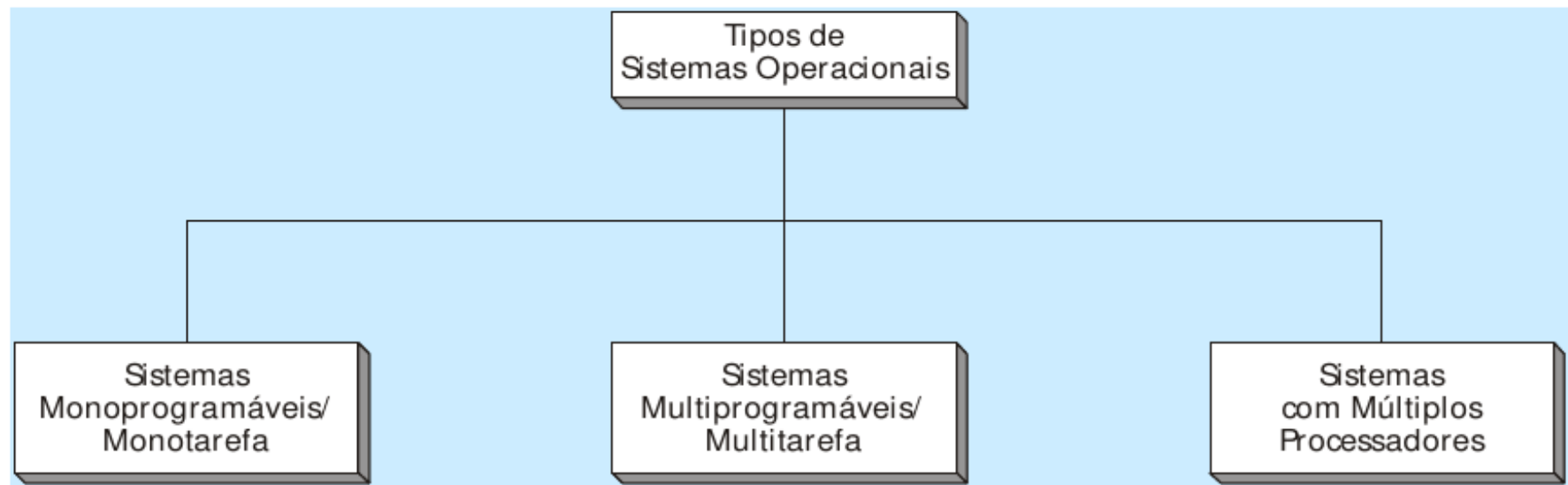


Table Computing

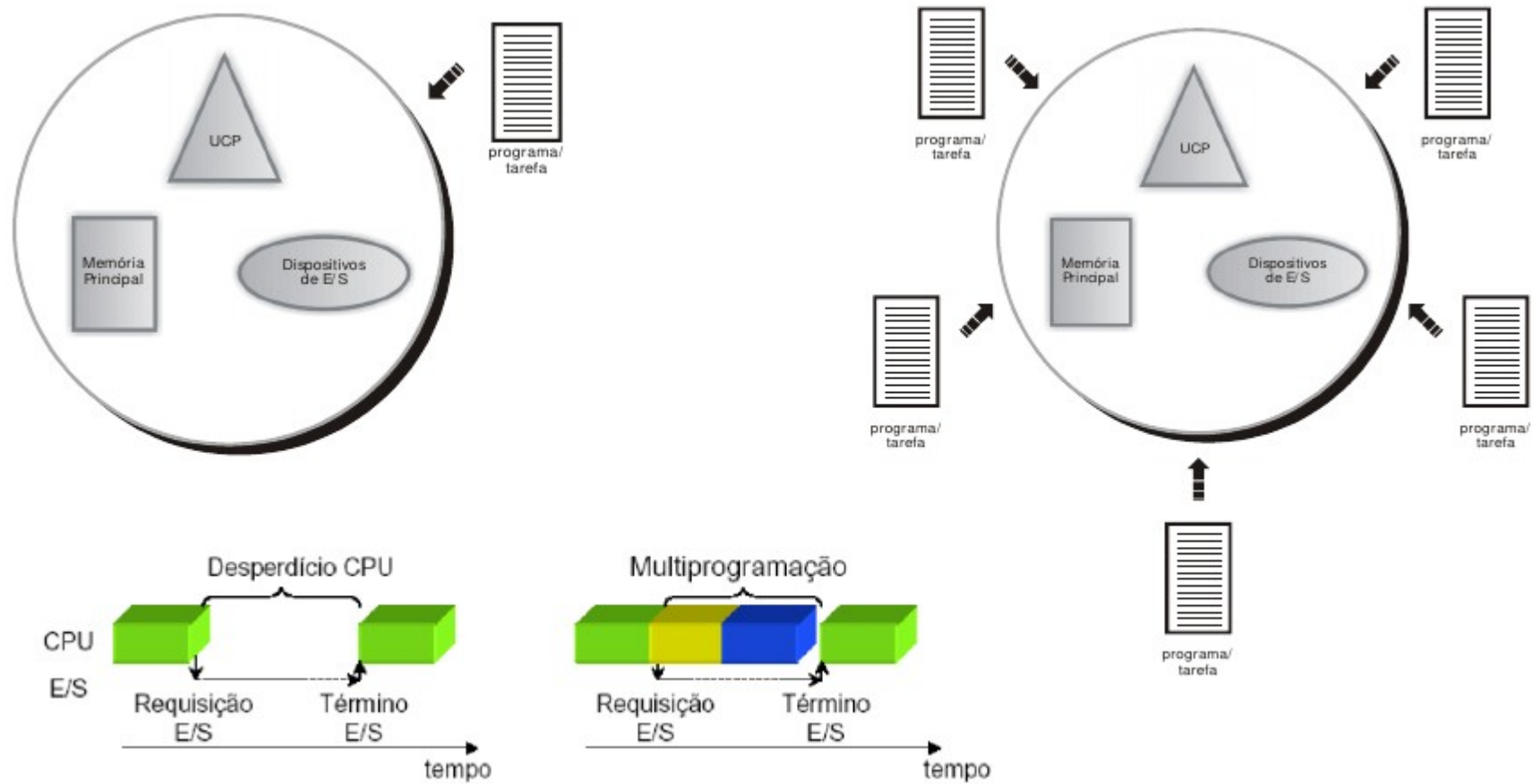


Embedded Computing

Tipos de sistemas operacionais

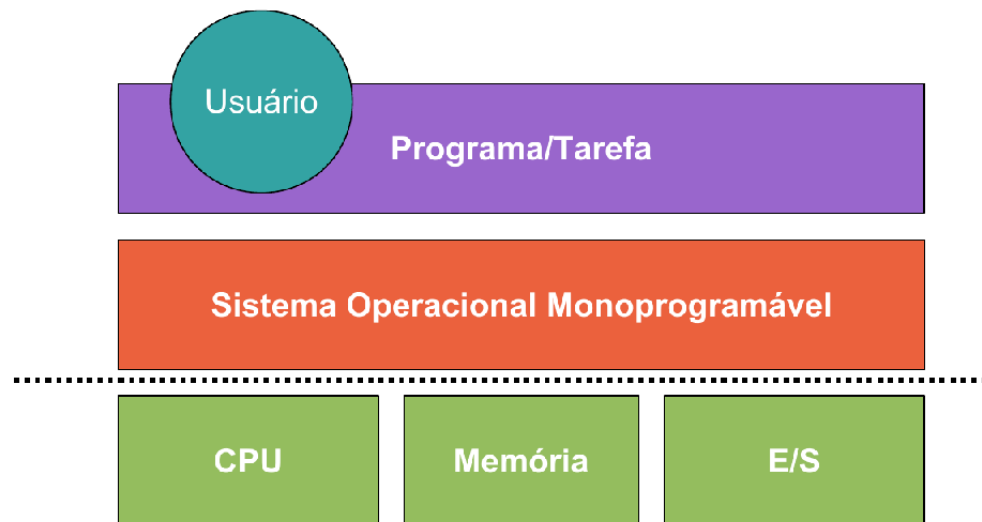


Mono x Multiprogramação



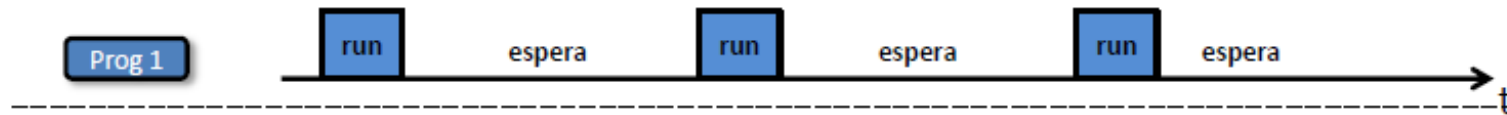
Sistema Monoprogramável/Monotarefa (1)

- Execução exclusiva de **um único** programa por vez
 - recursos dedicados
 - Execução sequencial
 - subutilização (memória, CPU, dispositivos de E/S)
 - CPU ociosa – custo!

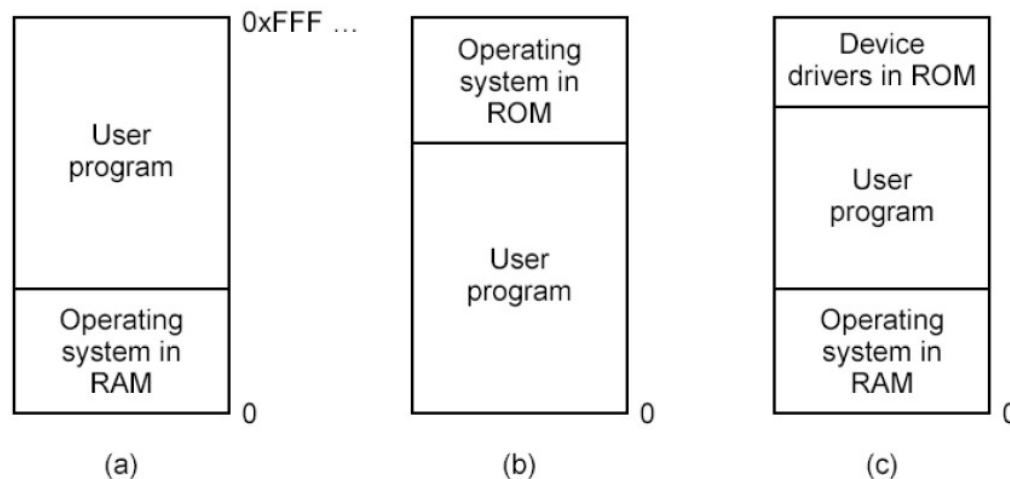


Sistema Monoprogramável/Monotarefa (2)

Sem Multiprogramação

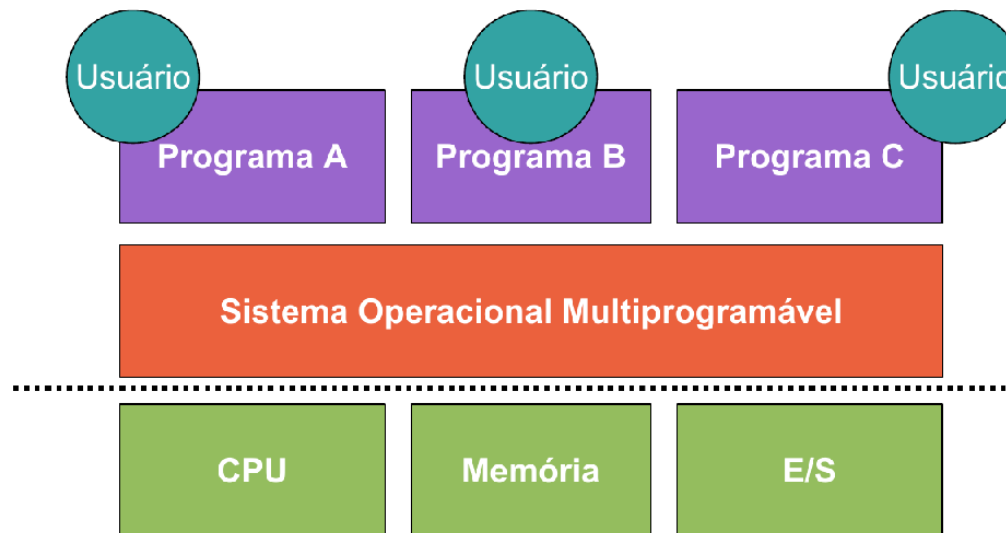


- Baixa complexidade de implementação
 - Não exige proteção de memória (apenas separa SO e usr)



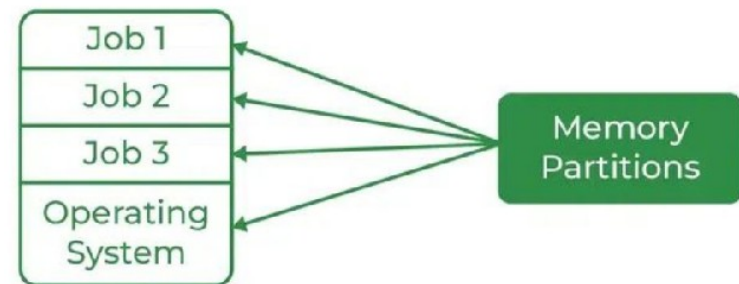
Sistema Multiprogramável/Multitarefa (1)

- Objetivo: manter **mais de um** programa em execução “simultaneamente”
 - ilusão de que cada programa possui uma máquina dedicada



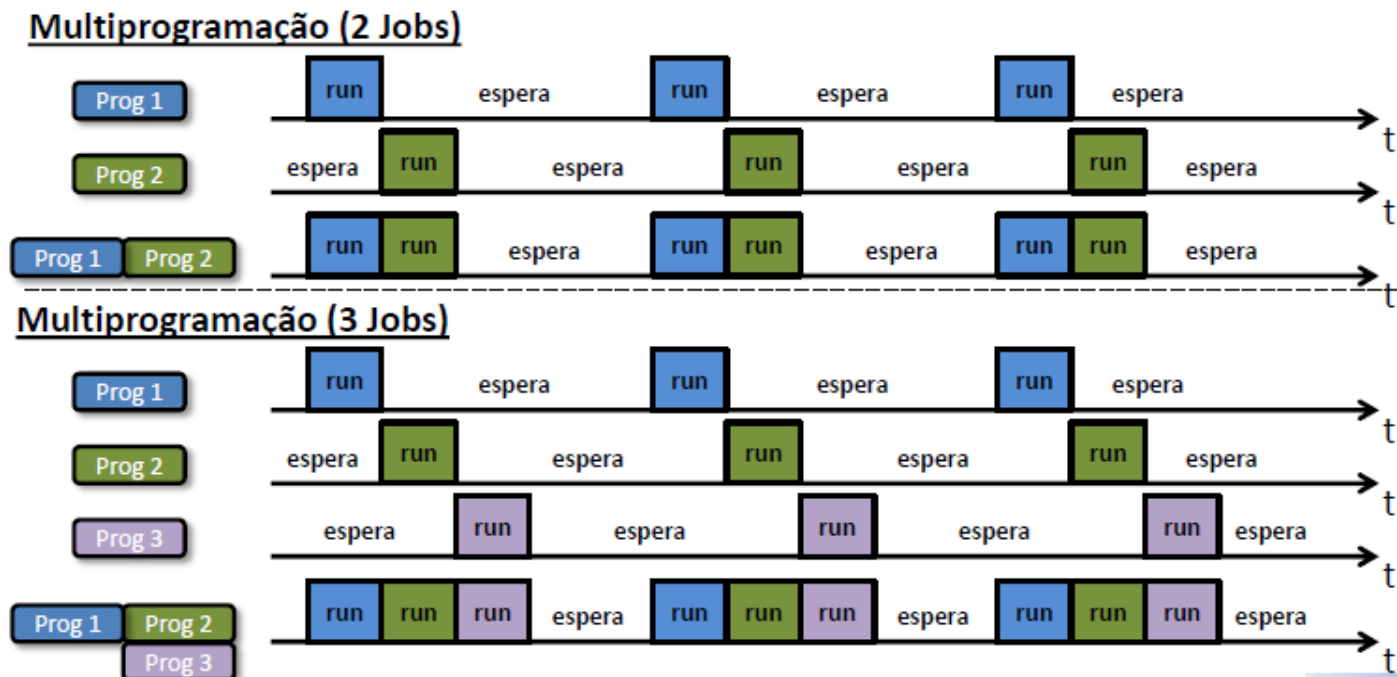
Sistema Multiprogramável/Multitarefa (2)

- Ideia: aproveitar o tempo ocioso da CPU durante a E/S
 - Maximização do uso do processador e da memória
 - Maior taxa de utilização do sistema como um todo
- Compartilhamento de recursos - acesso concorrente
- Exige:
 - proteção de memória
 - mecanismo de interrupção

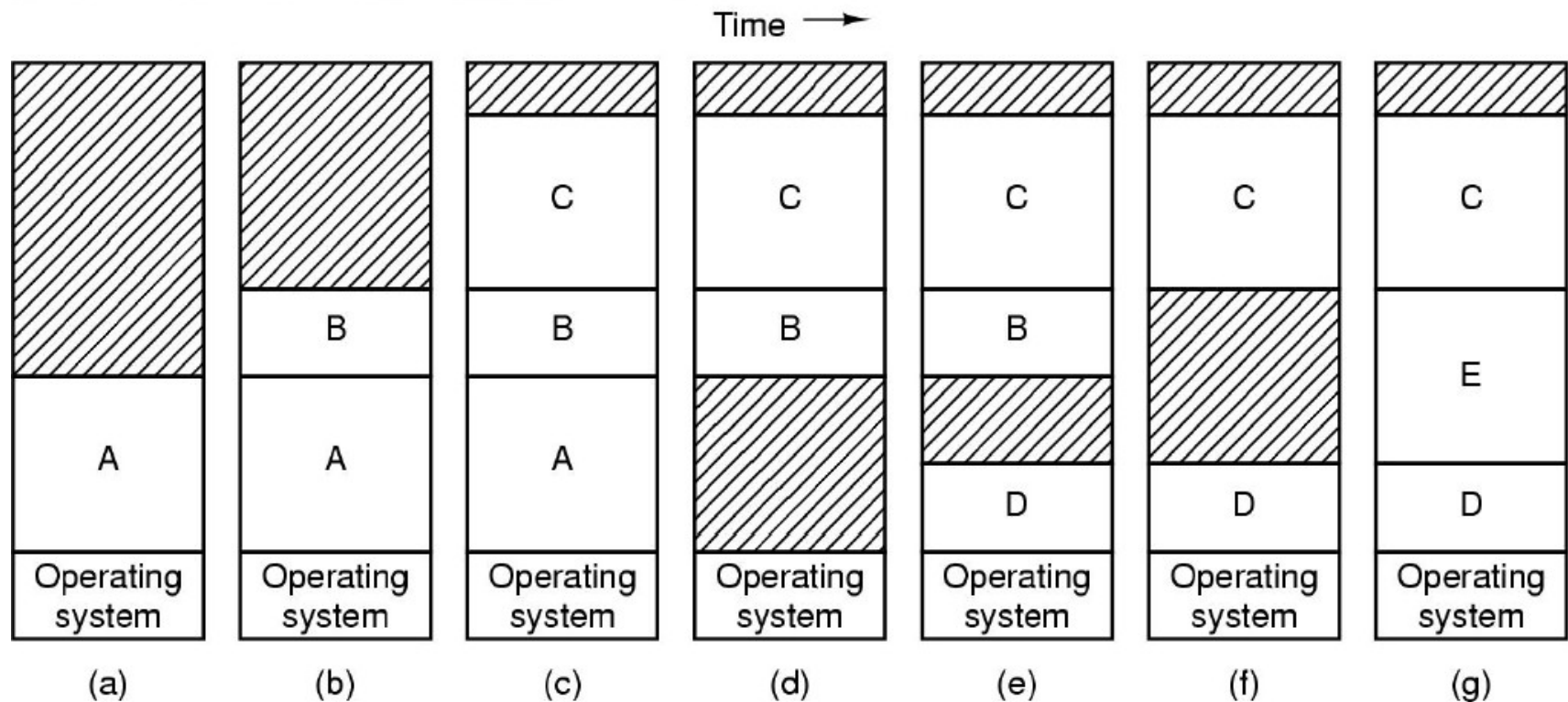


Sistema Multiprogramável/Multitarefa (3)

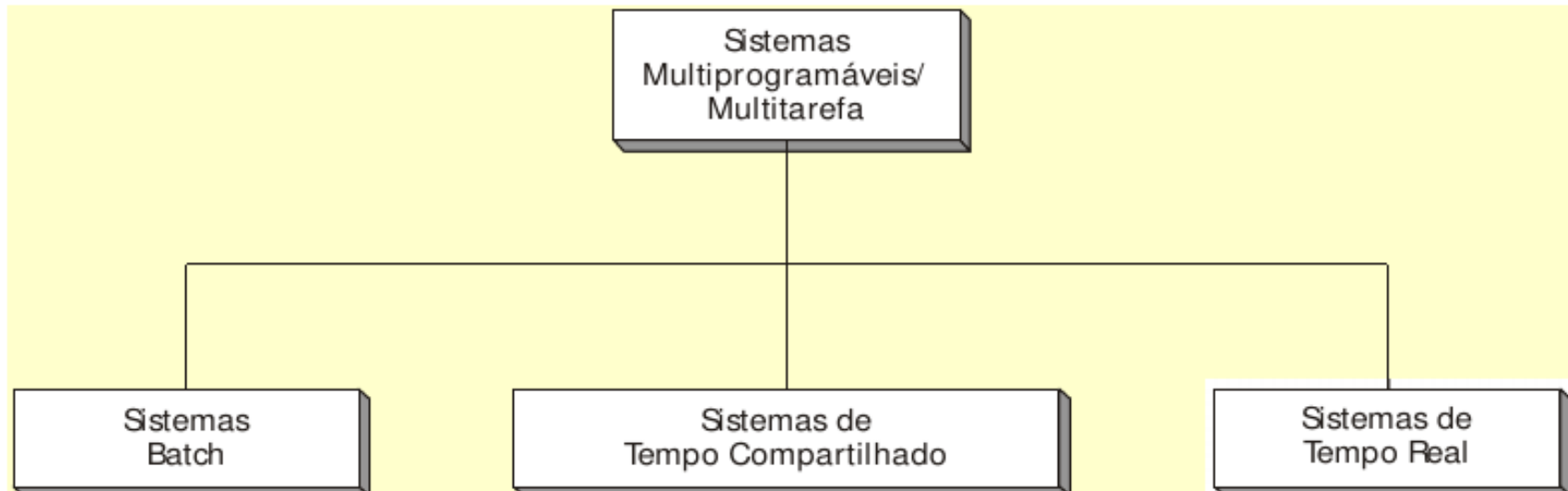
- Vários programas **competindo** pelos recursos do sistema



Sistema Multiprogramável/Multitarefa (4)

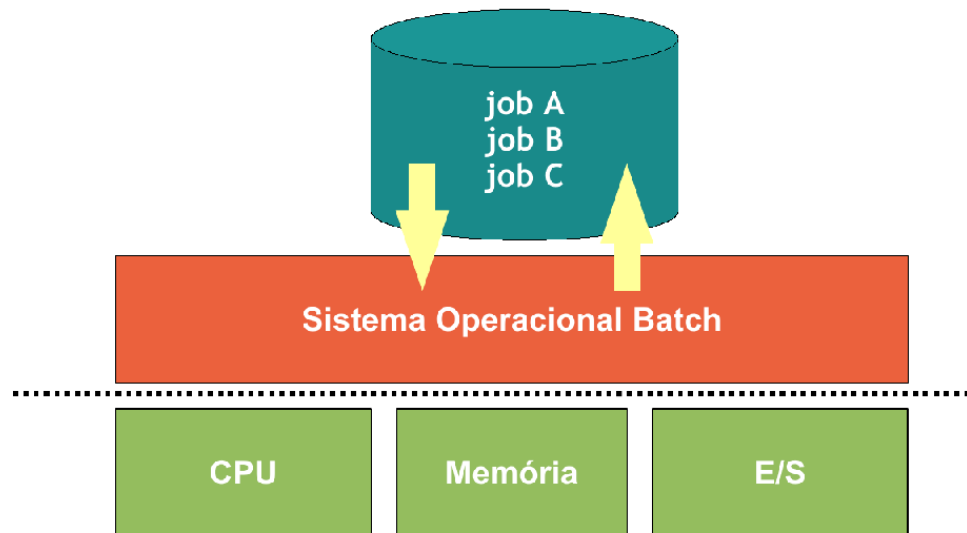


Sistemas multiprogramáveis/multitarefa: gerência das aplicações



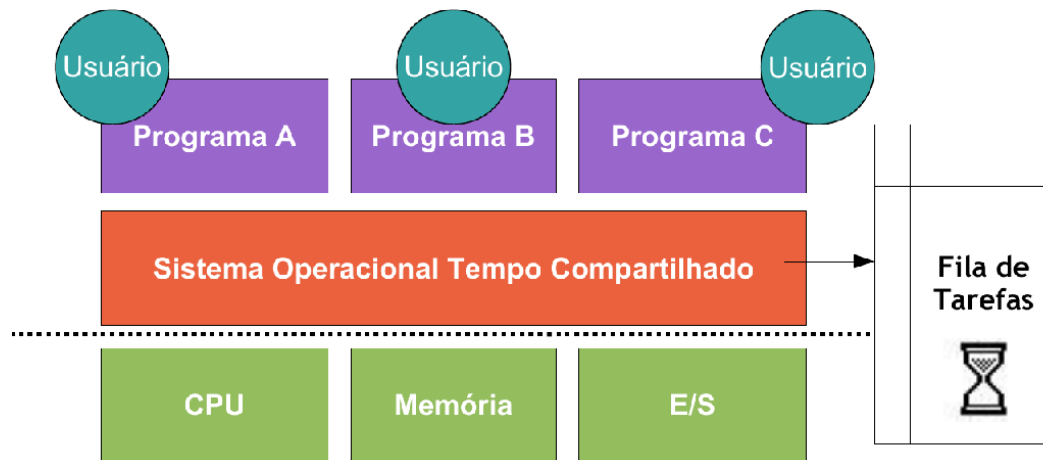
Sistemas em Batch

- Processamento em lote
 - Execução sequencial de jobs
- Sem interação com o usuário



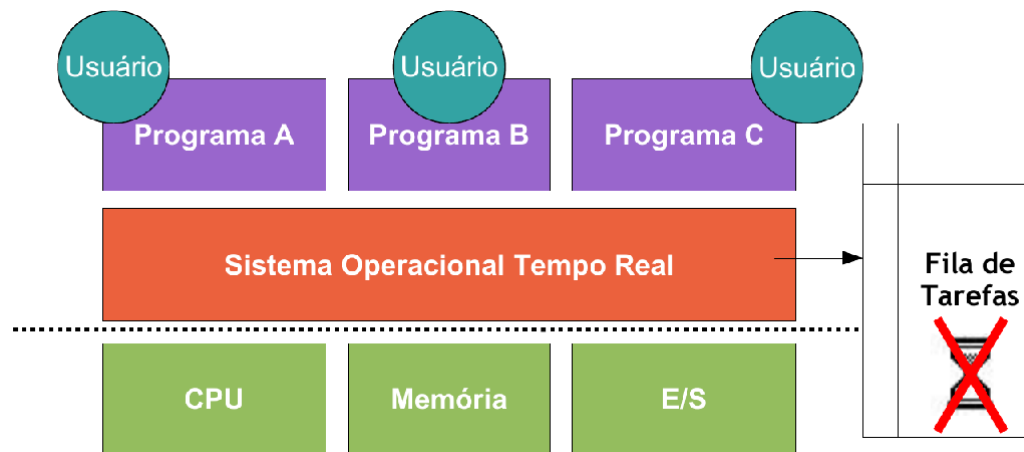
Sistemas de Tempo Compartilhado

- Time slice
- Escalonamento + multiprogramação
- Modelos preemptivos e colaborativos



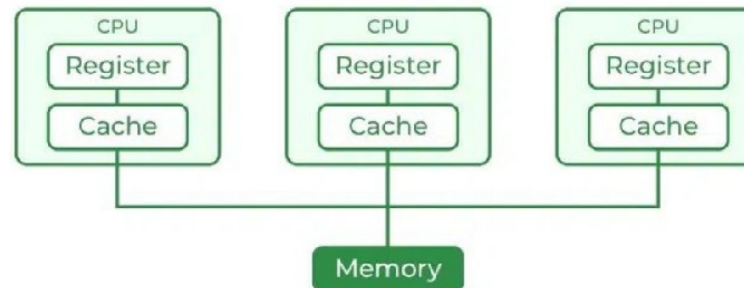
Sistemas de Tempo Real

- Restrições temporais
 - Sem time slice
 - Aplicações controlam o tempo de uso
- Hard real time x soft real time



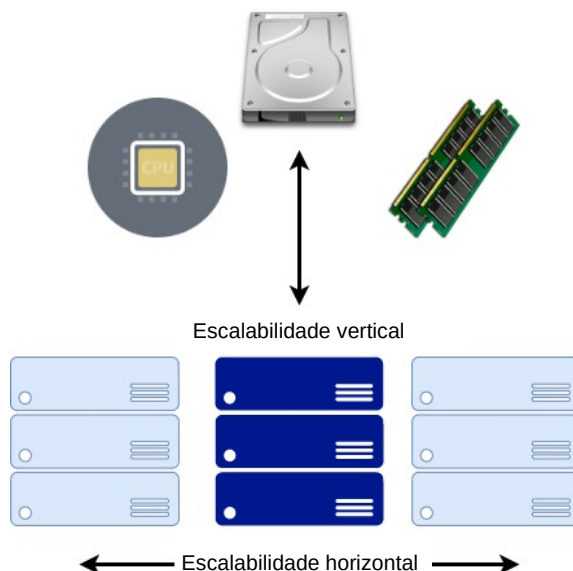
Sistemas com Múltiplos Processadores (1)

- Sistemas com **mais de uma CPU** interligada
- Execução **simultânea** de programas
- Supre a dificuldade no desenvolvimento de processadores mais rápidos
- Ideal para sistemas que necessitam uso intensivo de CPU (*CPU-bound*)
 - ex. processamento científico

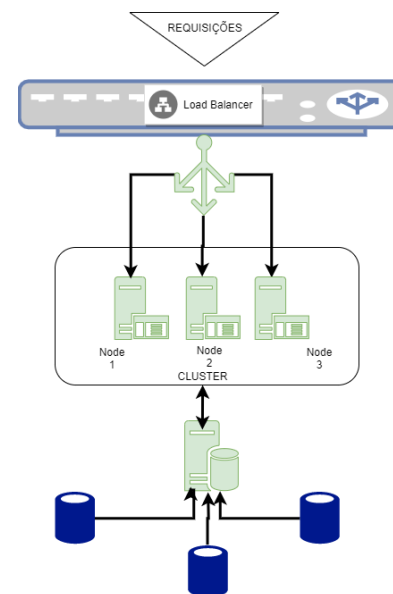


Sistemas com Múltiplos Processadores: características (2)

Escalabilidade
aumento da
capacidade
computacional



Balanceamento
distribuição de
carga



Tolerância a falhas
aumento da
disponibilidade

Sistemas com Múltiplos Processadores: classificação (3)

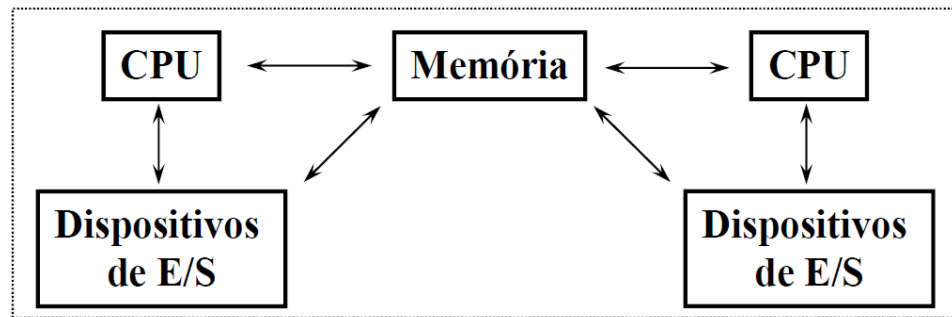
- Classificação em função:
 - da forma de comunicação entre CPUs
 - do grau de compartilhamento de memória e E/S

Sistemas fortemente acoplados x fracamente acoplados



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

- 2 ou mais processadores **compartilhando memória**



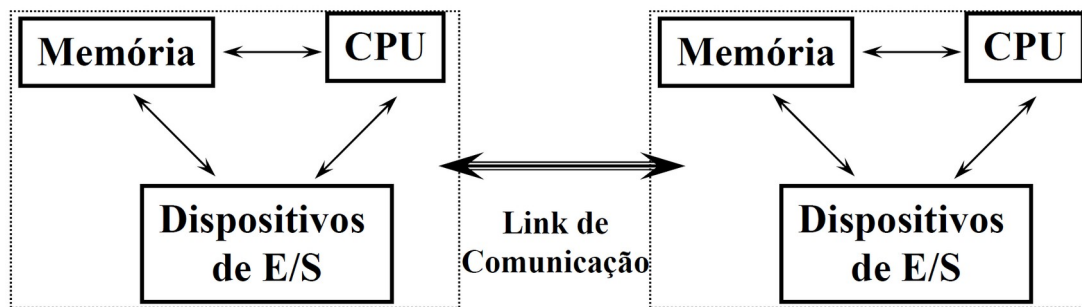
- controle: um único SO
- usados em sistemas ***CPU-bound*** para processamento de solução de um único problema

Sistemas fortemente acoplados (*tightly coupled*) (2)

- bom desempenho
 - execução de tarefas em paralelo
- custo
 - compartilhamento – vantagens e desvantagens
- Processadores podem ou não executar as mesmas tarefas
 - Assimétrico
 - Modelo master-slave
 - Cada processador recebe uma tarefa específica
 - Simétrico
 - 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, sem compartilhamento de memória ou clock



- comunicação: **troca de mensagens**
- agrega confiabilidade (redundância, autonomia)

Sistemas fracamente acoplados (*loosely coupled*) (2)

- acelera a computação
 - “subcomputações” podem rodar concorrentemente
 - sobrecarga de jobs → balanceamento de carga
- Processamento distribuído
 - SO de rede
 - usuários conhecem a localização dos recursos que estão utilizando; acessam máquinas remotas
 - cada máquina roda o seu próprio SO local
 - SO distribuído
 - usuários não sabem onde os seus programas estão executando, nem onde seus arquivos estão localizados
 - tudo é gerenciado pelo SO

Tipos de sistemas operacionais e suas características (1)

- **Batch**
 - Execução de tarefas sequenciais
- **De rede**
 - Acesso recursos em outros computadores
- **Distribuído**
 - Acesso transparente a recursos
- **Multiusuário**
 - Cada recurso possui um dono e regras de acesso
- **Servidor**
 - Gerência eficiente de recursos

Tipos de sistemas operacionais e suas características (2)

- Desktop
 - GUI e suporte à interatividade
- Móvel
 - Gestão de energia, conectividade e interação com sensores
- Embarcado
 - restrições quanto ao espaço de memória e consumo de energia
- Tempo real
 - restrições temporais; classificados como soft ou hard real-time