

**INSTITUTO
FEDERAL**
Ceará

Sistemas Operacionais



Linux

MAC OS

Windows

Prof. Sávio Felipe
saviofelipe@ifce.edu.br

O que aprenderemos

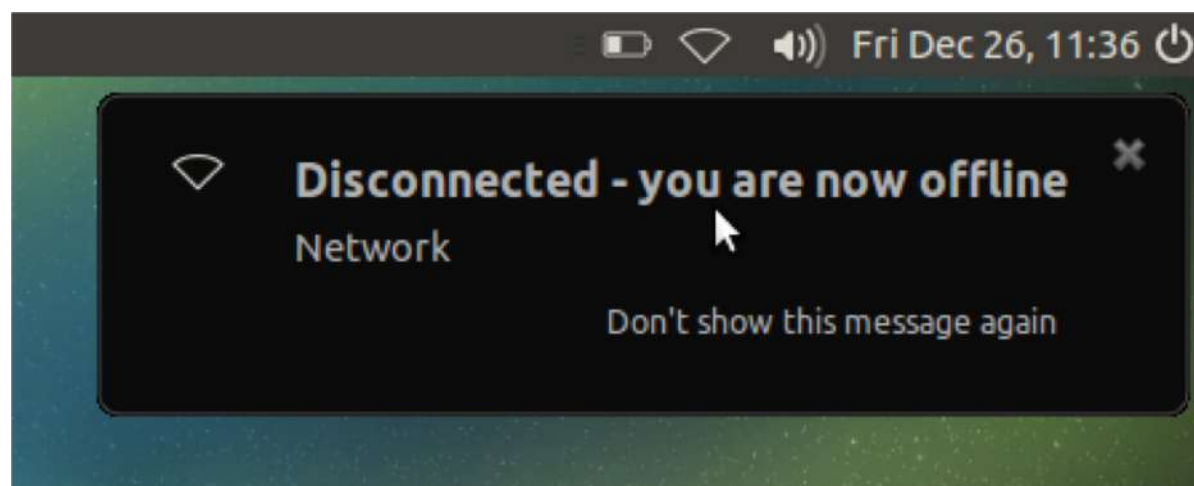
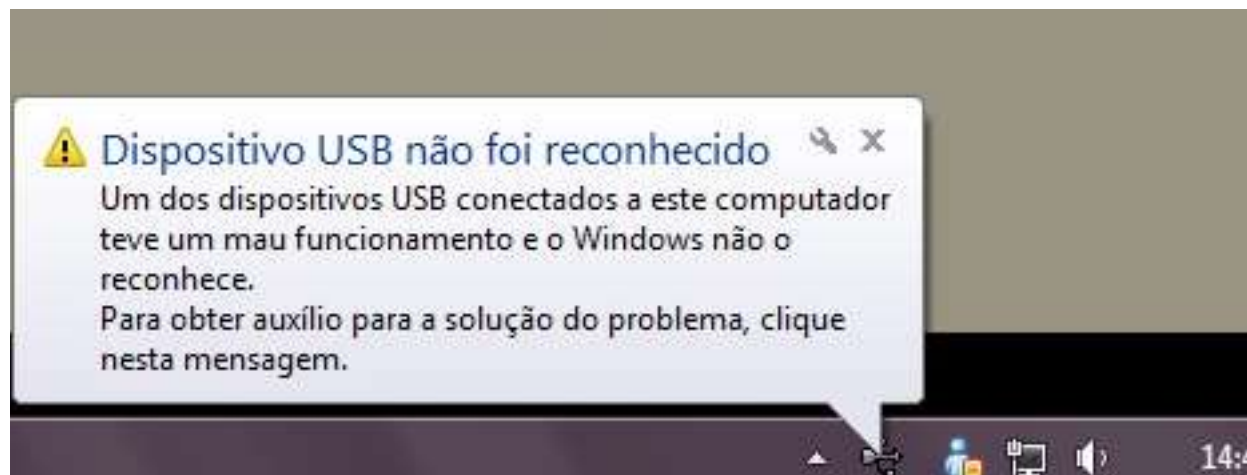


INSTITUTO FEDERAL
Ceará
Campus Fortaleza



O que é um Sistema Operacional?

O que está
relacionado ao S.O?



Start





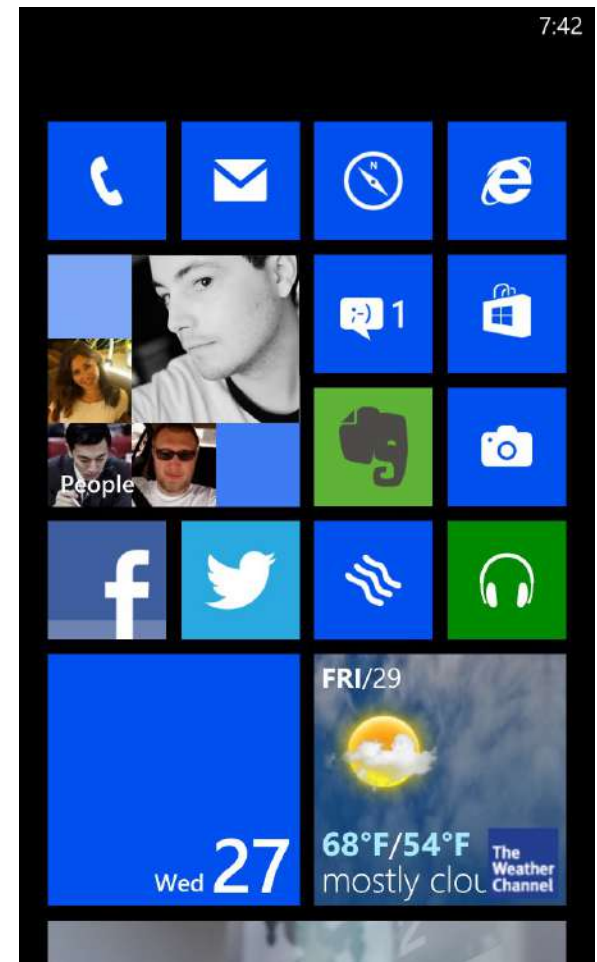


Finder File Edit View Go Window Help



Mon 4:46 AM Ryan





Antes de falar do S.O ...

De que ele precisa?

O computador

- Um computador moderno consiste em:
 - ✓ Um ou mais processadores.
 - ✓ Memória principal.
 - ✓ Discos.
 - ✓ Impressoras.
 - ✓ Diversos dispositivos de entrada e saída



Uma visão estendida do S.O

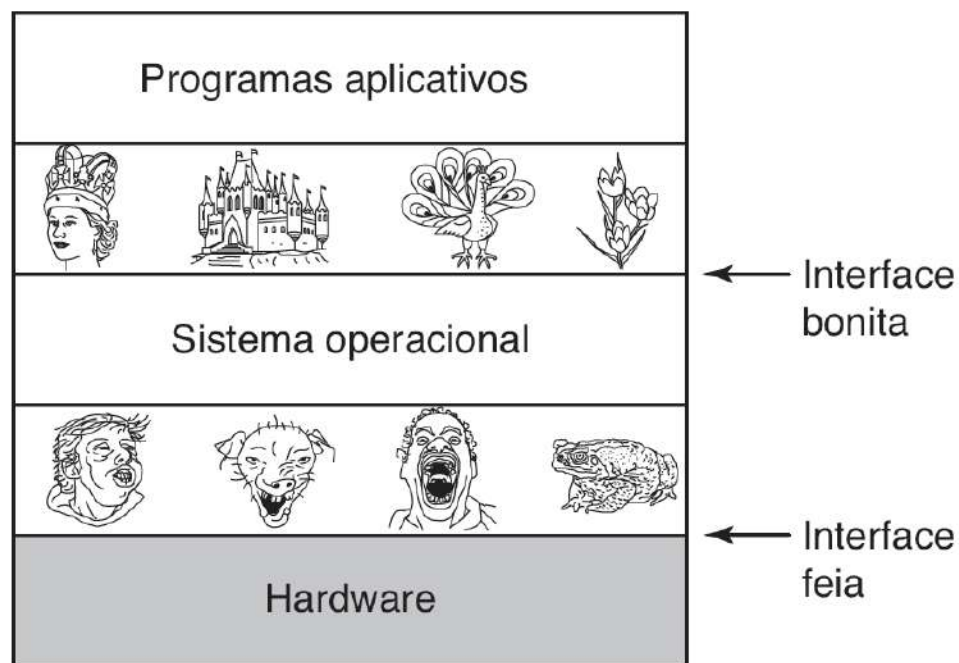


Figura 1.2 Sistemas operacionais transformam hardware feio em abstrações bonitas.

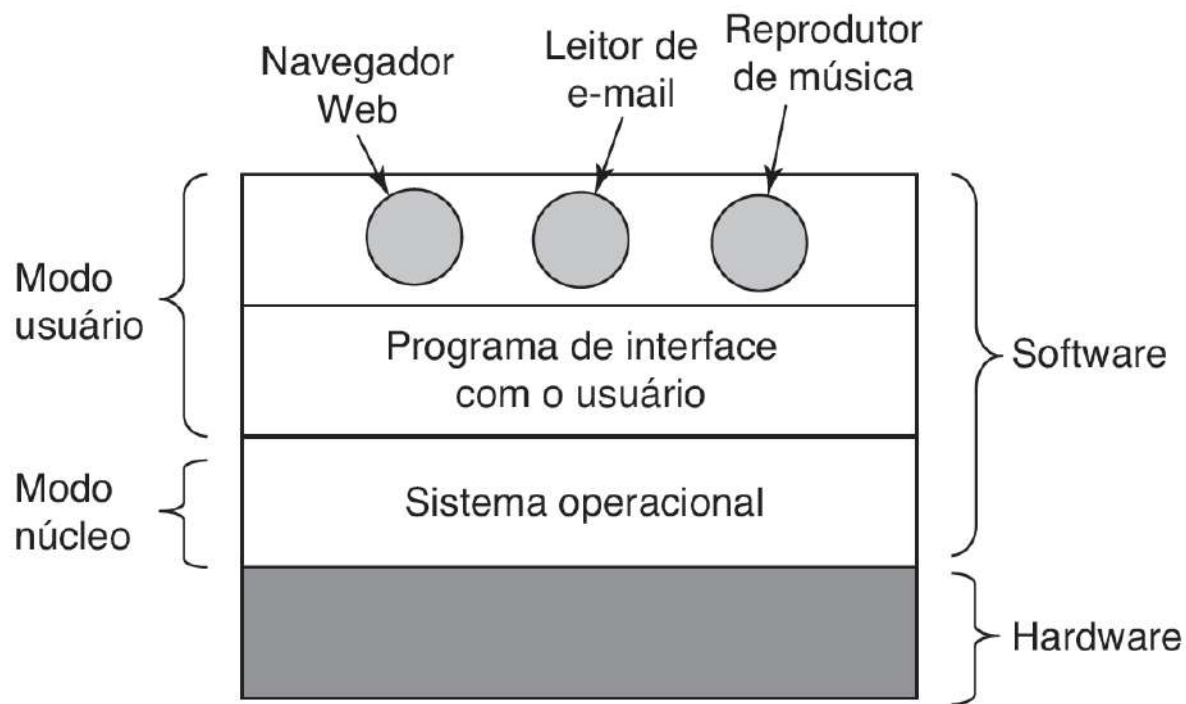


Figura 1.1 Onde o sistema operacional se encaixa.

S.O na função de gerenciador de recursos

- Permite que múltiplos programas sejam executados ao “mesmo tempo”
- Gerencia e protege a memória, os dispositivos de entrada e saída e outros recursos
- Inclui a multiplexação (partilha) de recursos de duas maneiras diferentes:
 - No tempo
 - No espaço

Evolução computacional

- Gerações:
 - (1945–55) Válvulas (eletromecânicos)
 - (1955–65) Transistores e sistemas em lotes (batch)
 - (1965–1980) CIs e multiprogramação
 - (1980–Presente) Computadores pessoais, tablets, smartphones...

CI e Multiprogramação

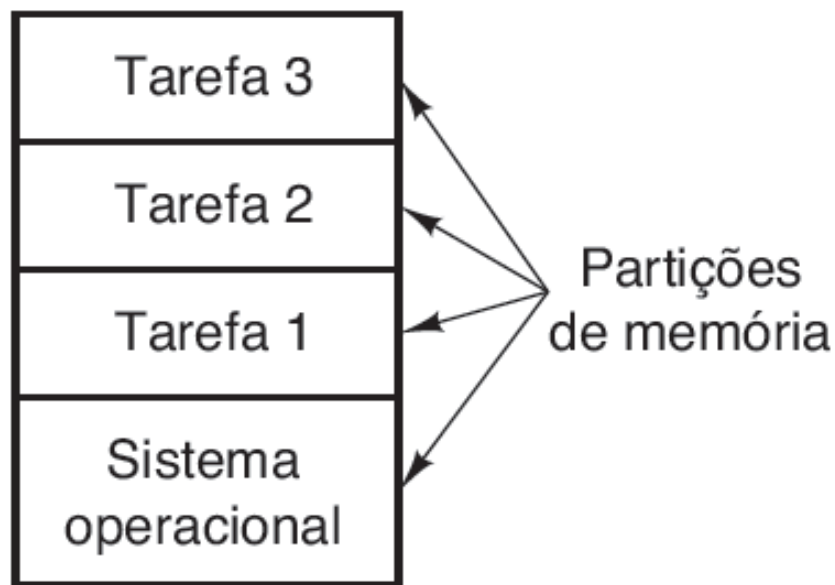
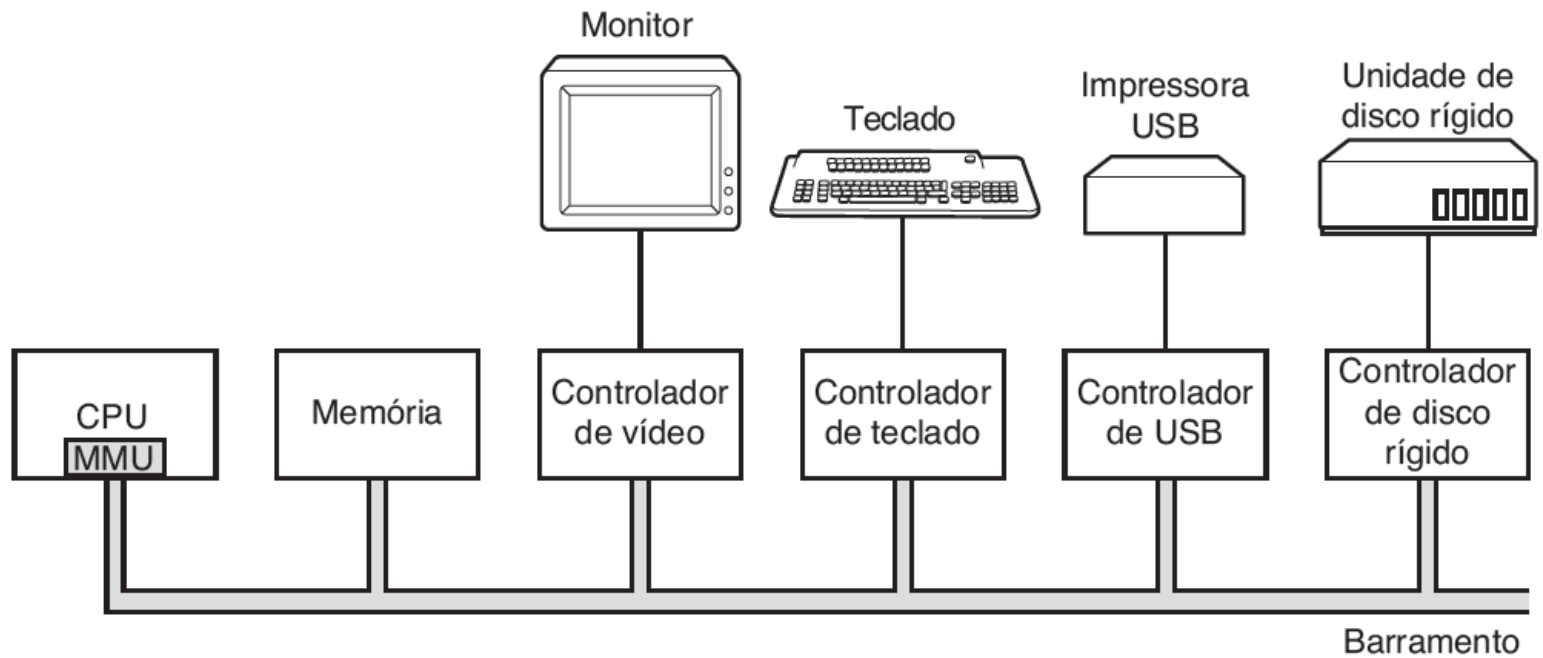


Figura 1.5 Um sistema de multiprogramação com três tarefas na memória.

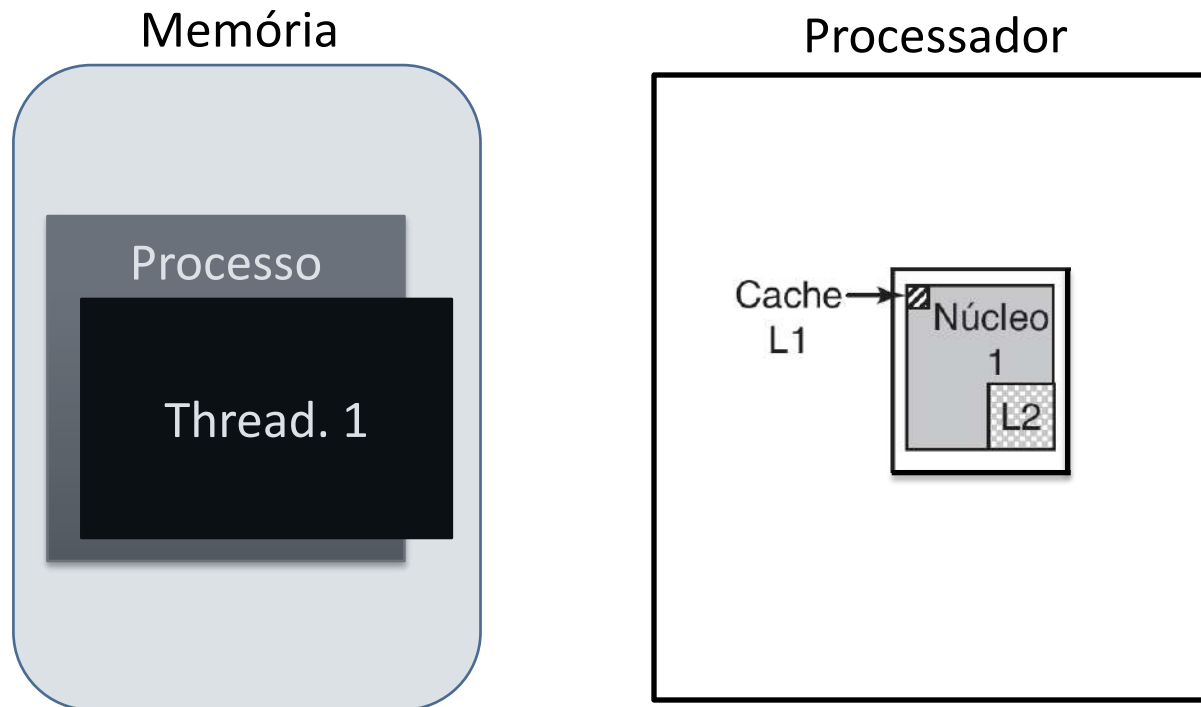
O que mudou?

Hardware e arquitetura (noções)



■ **Figura 1.6** Alguns dos componentes de um computador pessoal simples.

Chip moncore



Chips multicore/multinúcleo

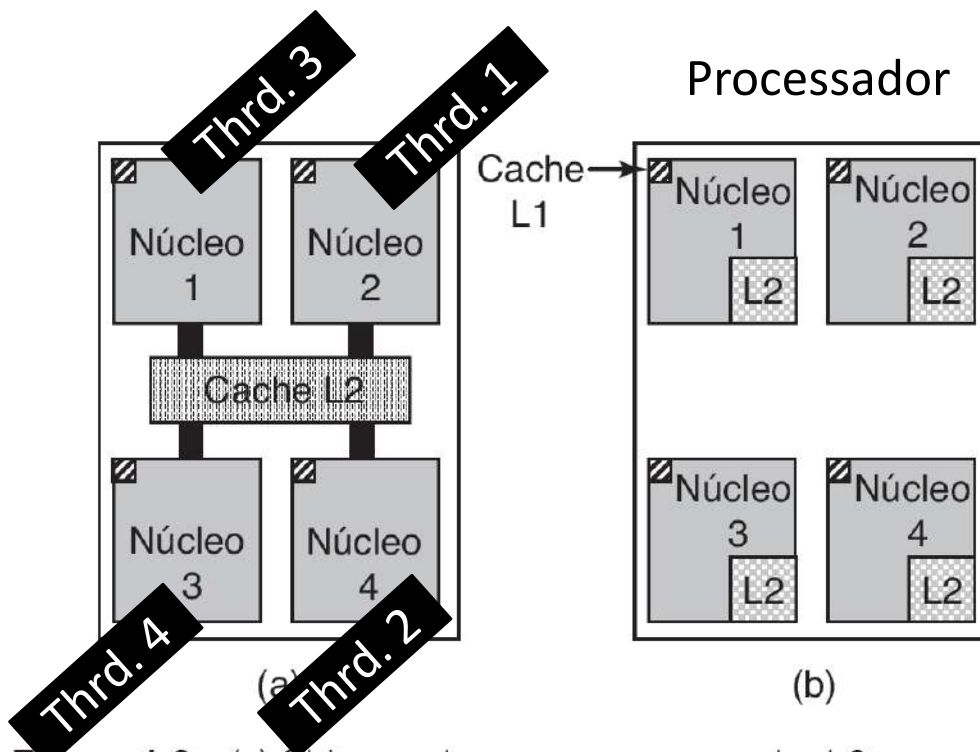
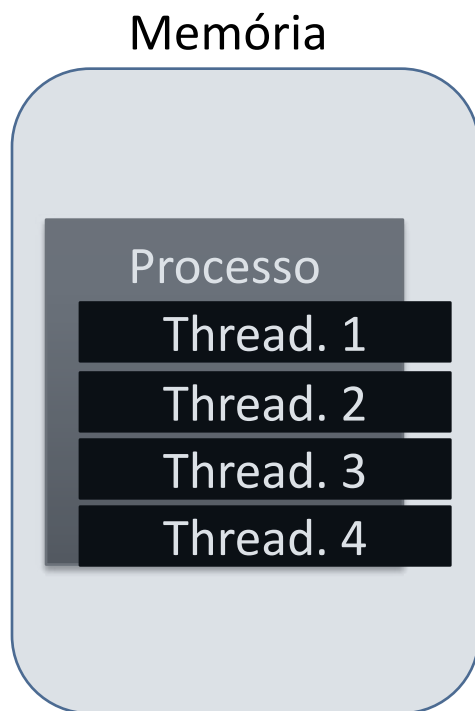
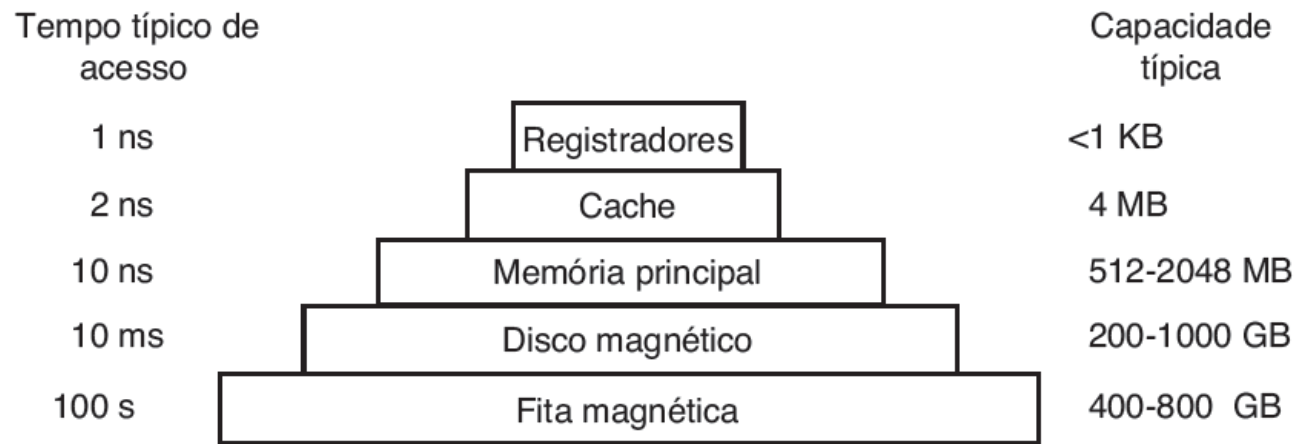


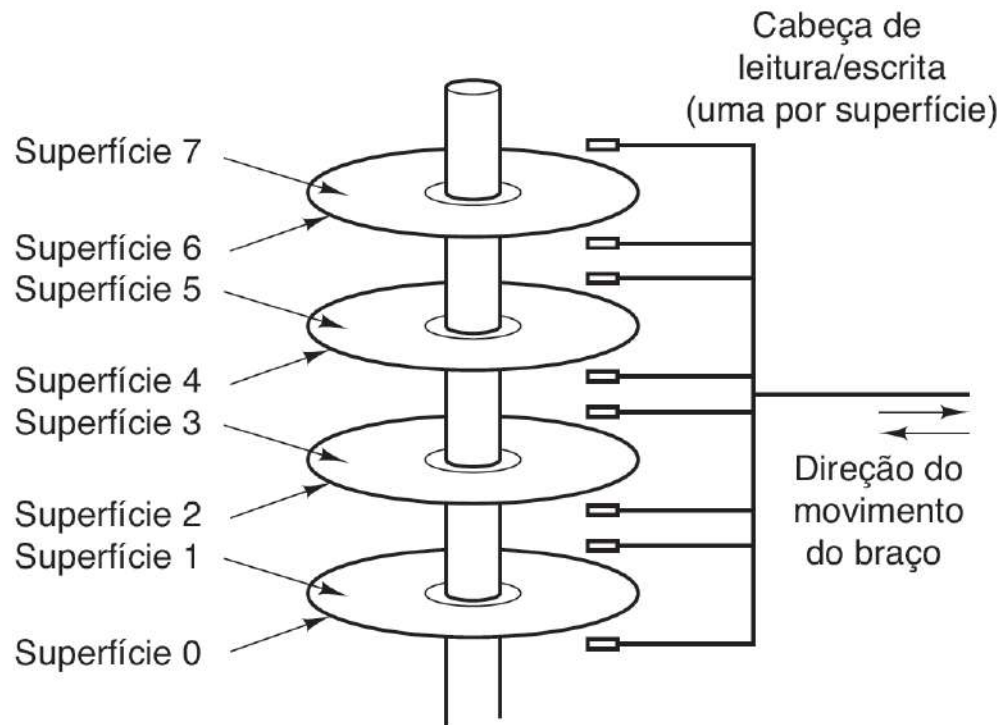
Figura 1.8 (a) Chip quad-core com uma cache L2 compartilhada. (b) Um chip quad-core com caches L2 separadas.

Memória



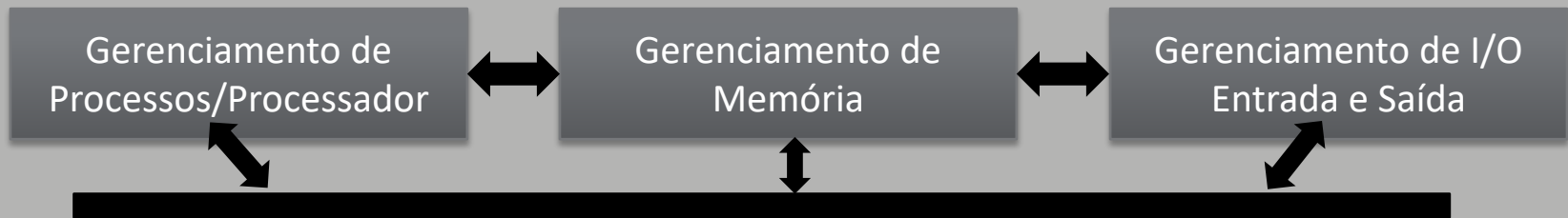
■ **Figura 1.9** Hierarquia de memória típica. Os números são aproximações.

Discos



■ **Figura 1.10** Estrutura de uma unidade de disco.

Como é feita a divisão de tarefas que o S.O realiza?



Tipos de sistemas operacionais

- S.O de computadores de grande porte
- S.O de servidores
- S.O de multiprocessadores
- S.O de computadores pessoais
- S.O de computadores portáteis
- S.O embarcados
- S.O de nós sensores (RSSF)...

Quais as diferenças
entre esses tipos?

Interrupções

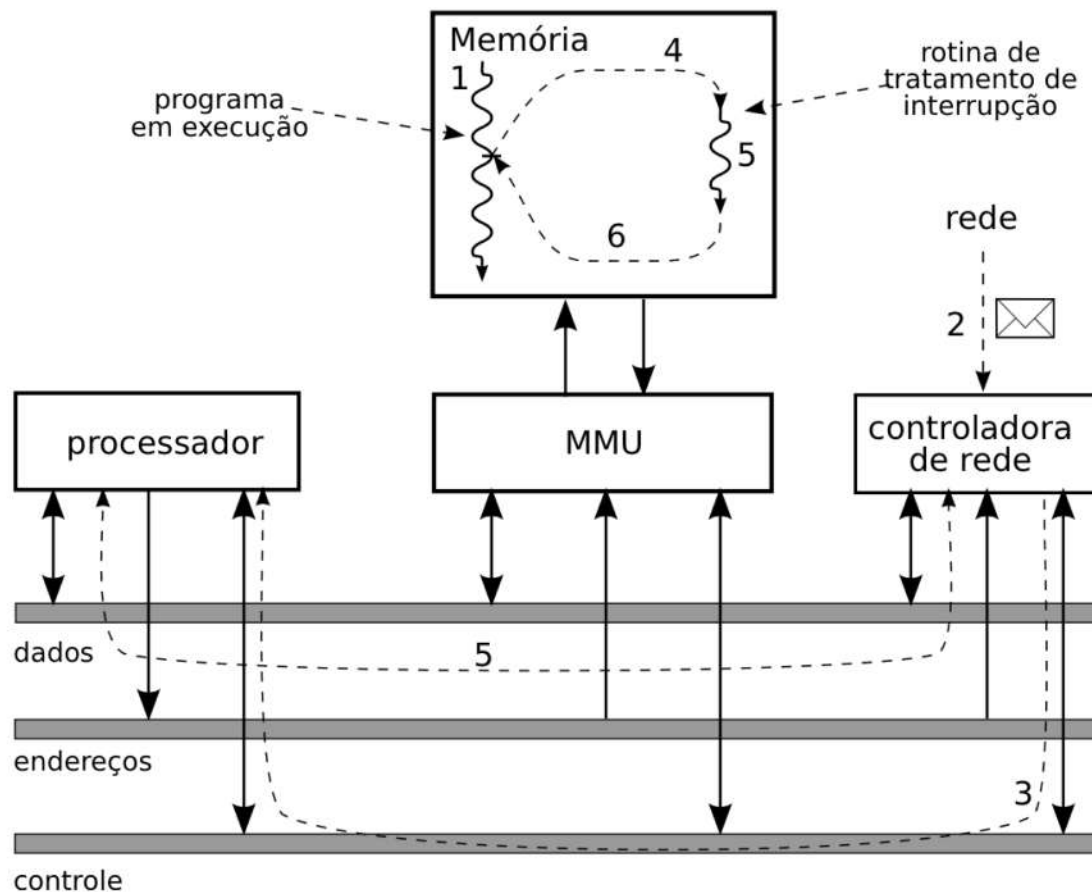


Figura 5: Roteiro típico de um tratamento de interrupção

Chamadas de sistemas *system calls*

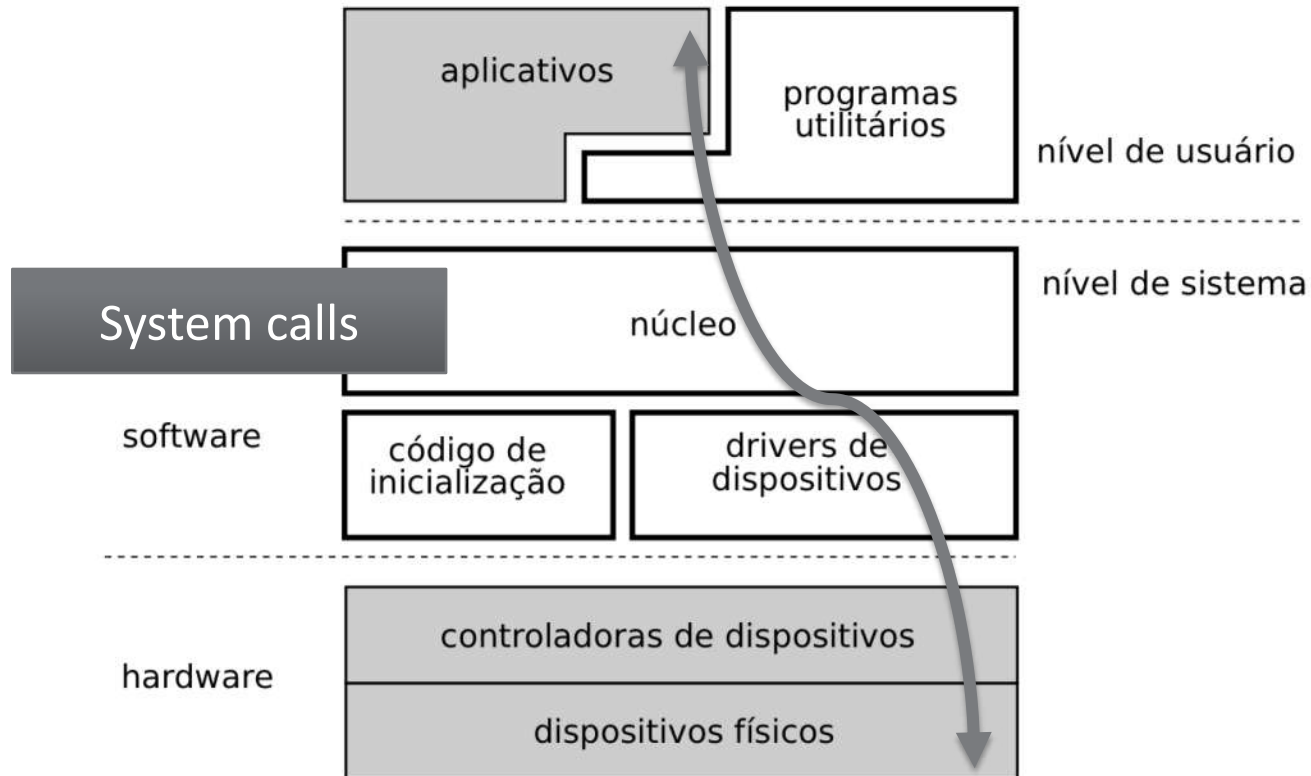


Figura 3: Estrutura de um sistema operacional

Comparativo de Chamadas de Sistemas

API Windows Win32 x Unix

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

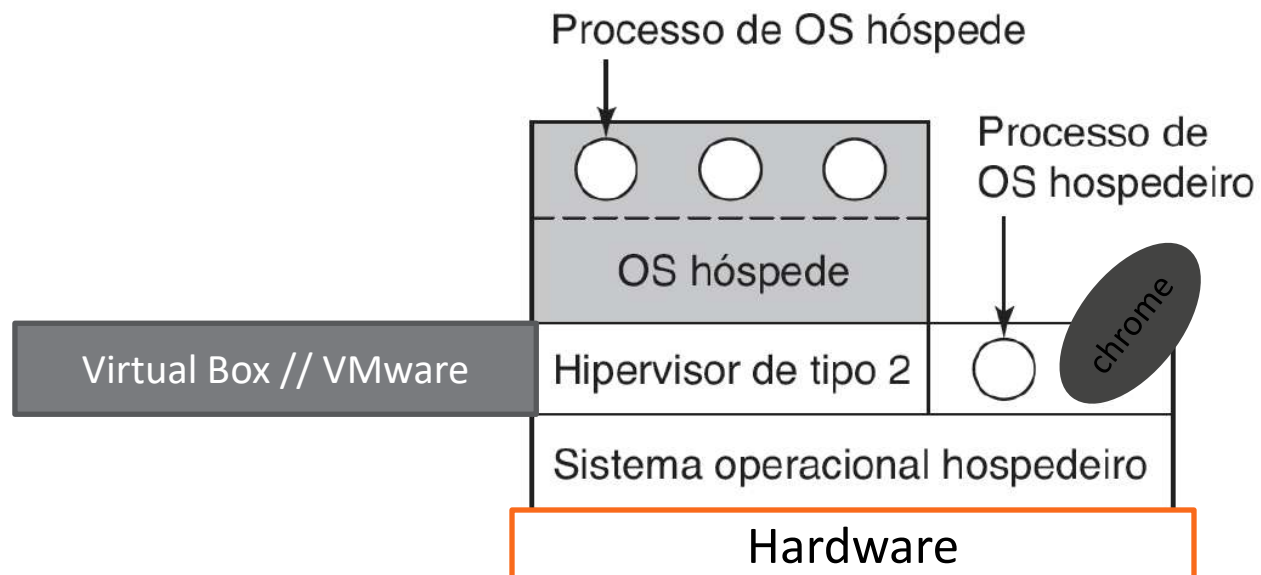
Comparativo

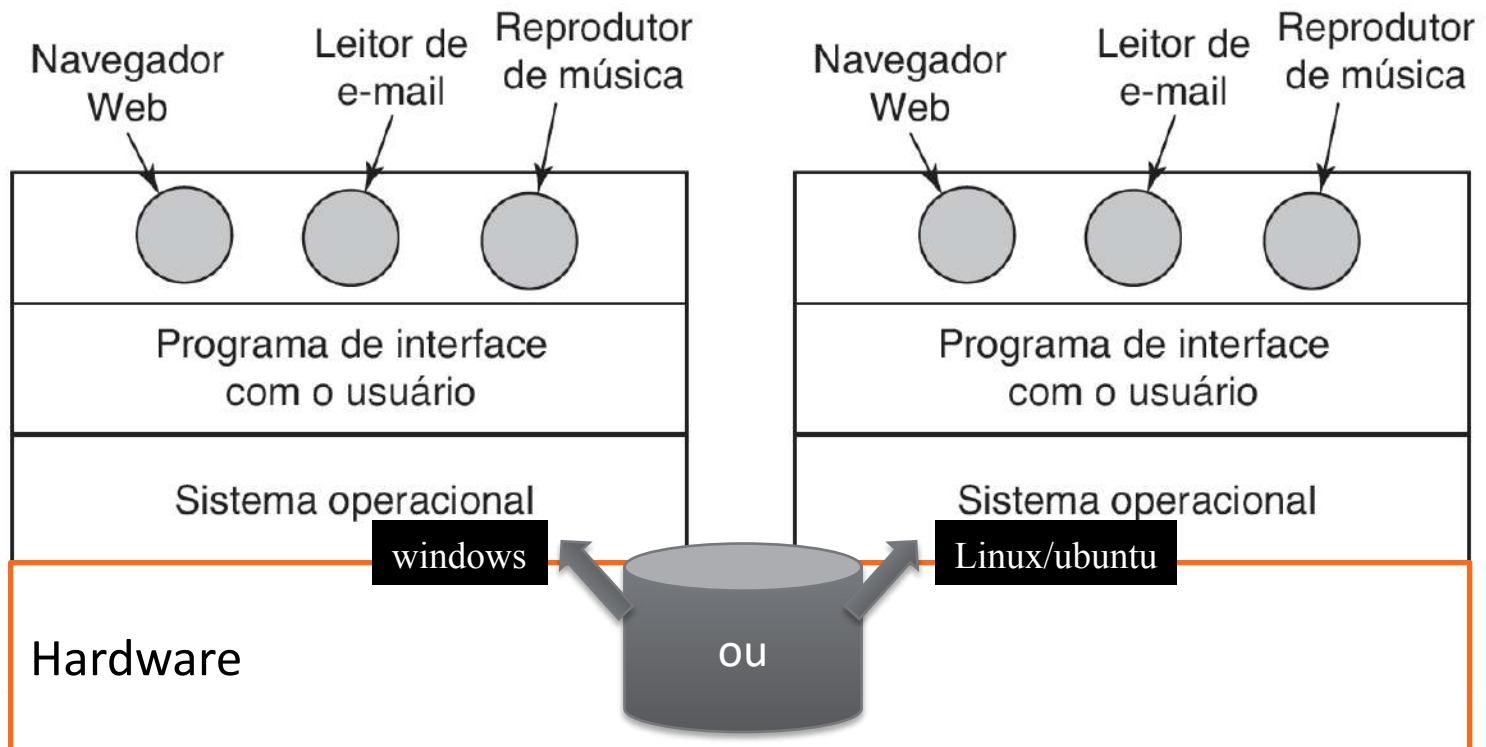
API Windows Win32

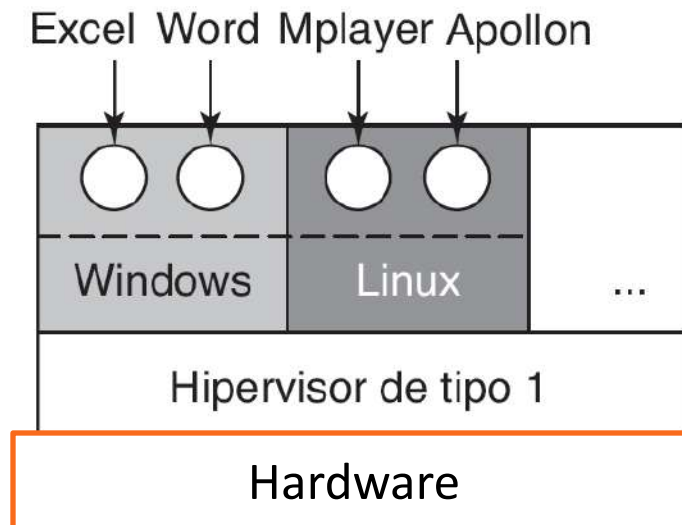
mkdir	CreateDirectory	Cria um novo diretório
rmdir	RemoveDirectory	Remove um diretório vazio
link	(nenhuma)	Win32 não dá suporte a links
unlink	DeleteFile	Destrói um arquivo existente
mount	(nenhuma)	Win32 não dá suporte a mount
umount	(nenhuma)	Win32 não dá suporte a mount
chdir	SetCurrentDirectory	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

Tabela 1.2 As chamadas da API Win32 que correspondem aproximadamente às chamadas do UNIX da Tabela 1.1.

É possível utilizar mais
de um S.O “sobre” um
mesmo hardware?







Estrutura de um S.O

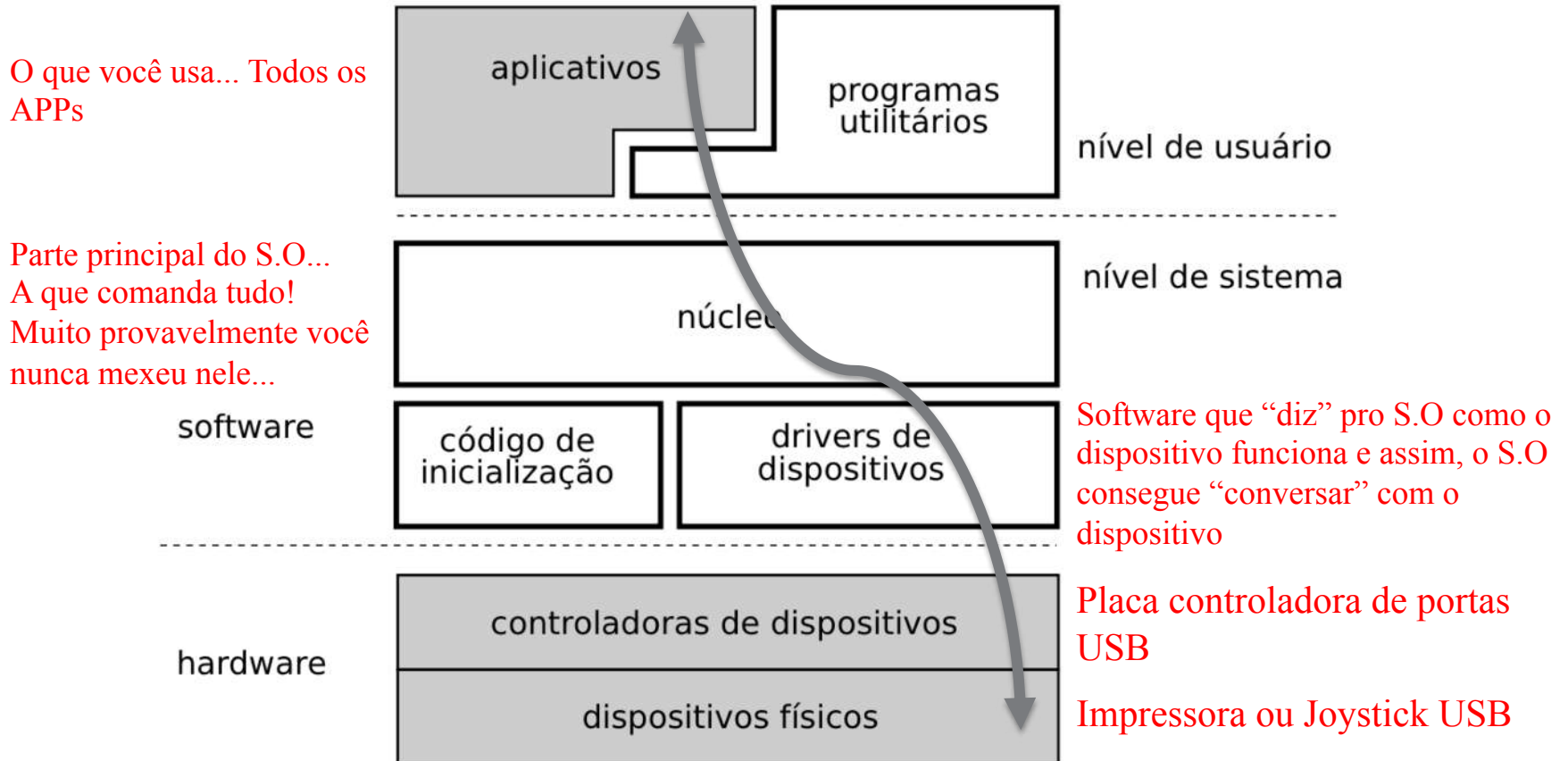
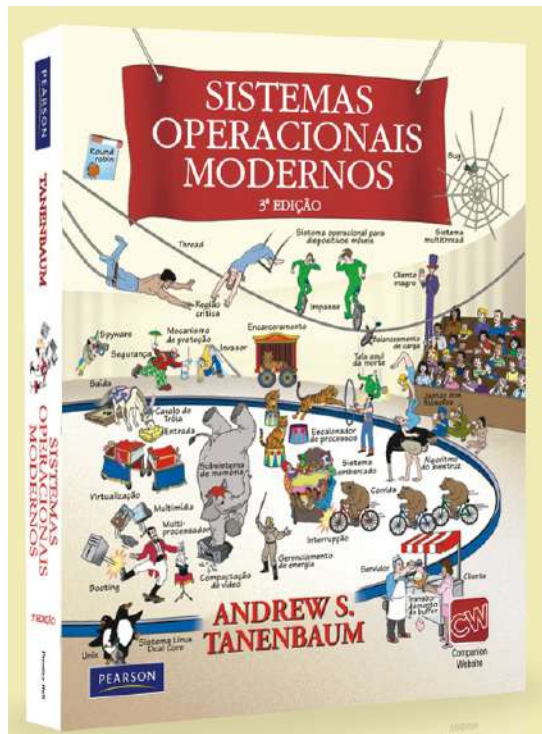
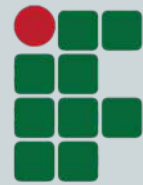


Figura 3: Estrutura de um sistema operacional



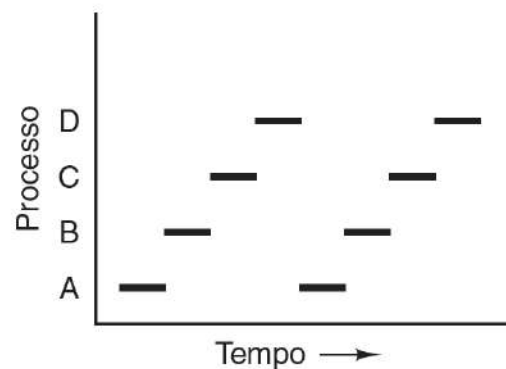
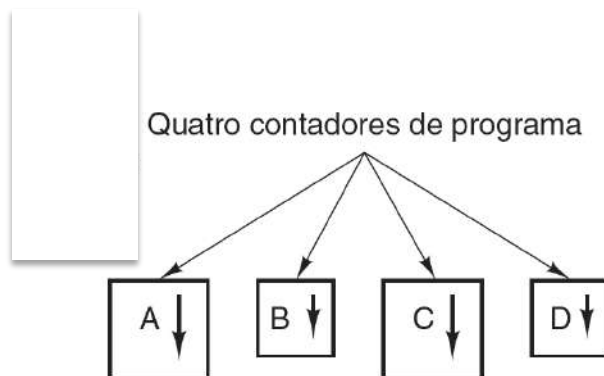
A matriz deste conjunto de slides é cedida pela editora e pelo autor do livro Sistemas Operacionais Modernos. Alguns ajustes foram realizados. As ilustrações que não tem explicitada a fonte, pertencem ao autor do livro.



**INSTITUTO
FEDERAL**
Ceará

Gerenciamento de Processos/*tasks* e threads

O Modelo de processo



Adaptado de Tanenbaum (2009)

Execução de tarefas no S.O

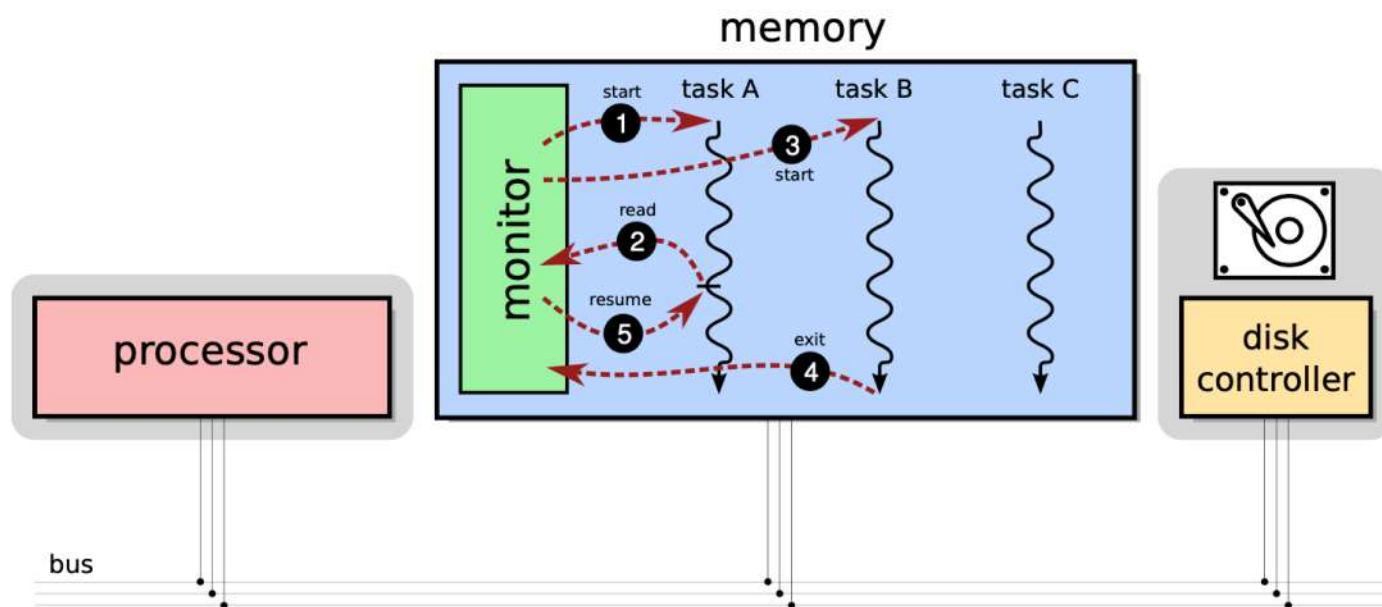


Figura 4.4: Execução de tarefas em um sistema multitarefas.

(Maziero, 2019)

Trocas de contexto

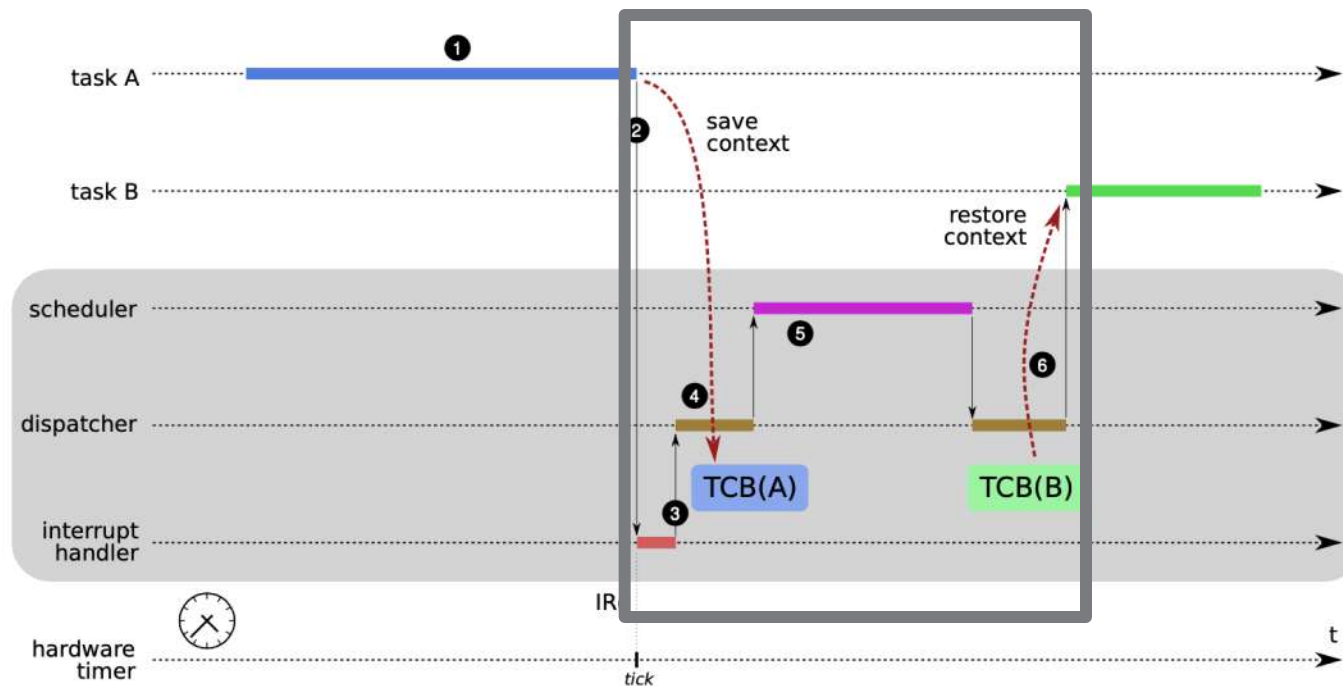


Figura 5.1: Passos de uma troca de contexto.

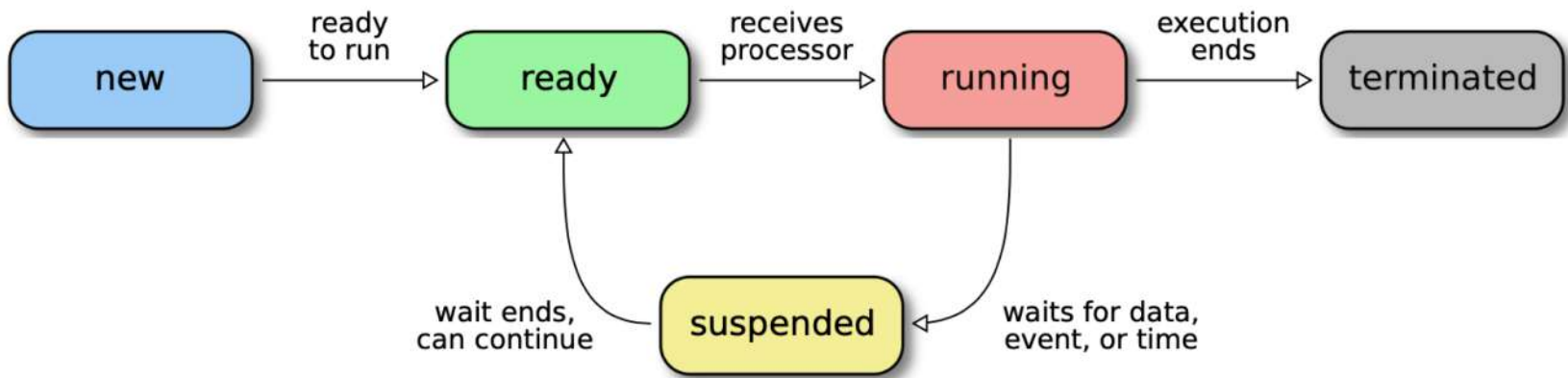
(Maziero, 2019)

Process Control Block (PCB)

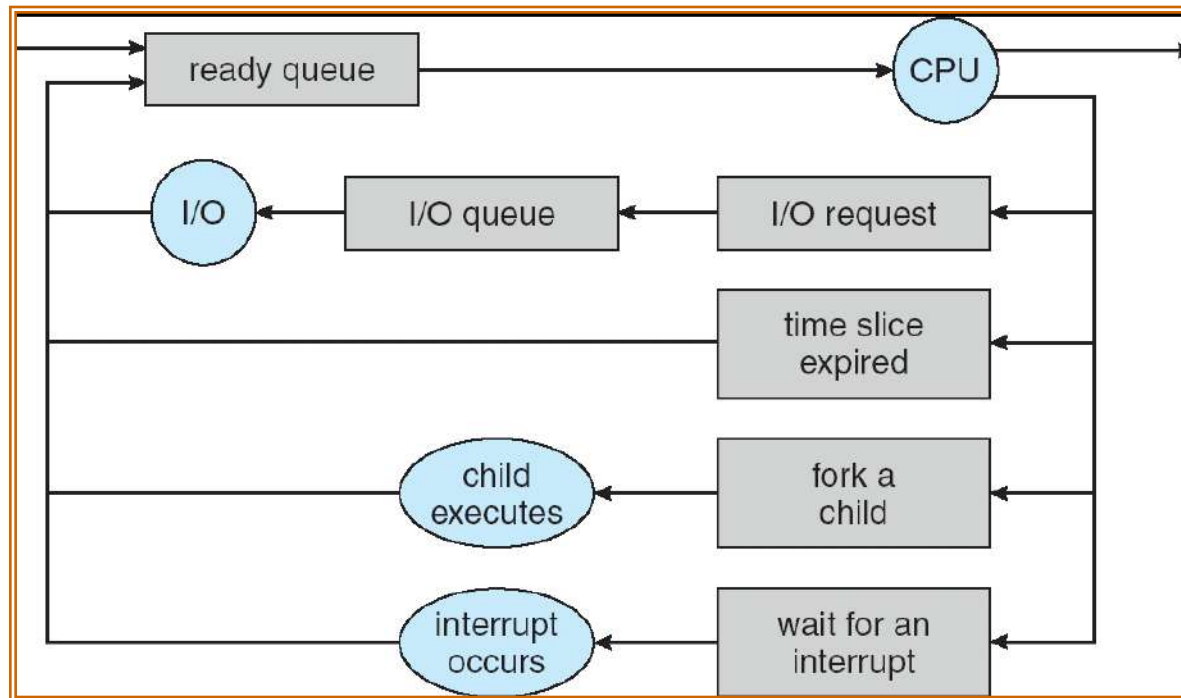
Informações associadas a cada processo

- Estado do processo
- Contador de programa
- Registradores da CPU
- Informação de escalonamento da CPU
- Informação de gerenciamento de memória
- Informação de contabilidade
- Informação de status de E/S

Estados dos processos



Representação do escalonamento de processos



A Thread

Fluxo de execução de instruções independene no processo

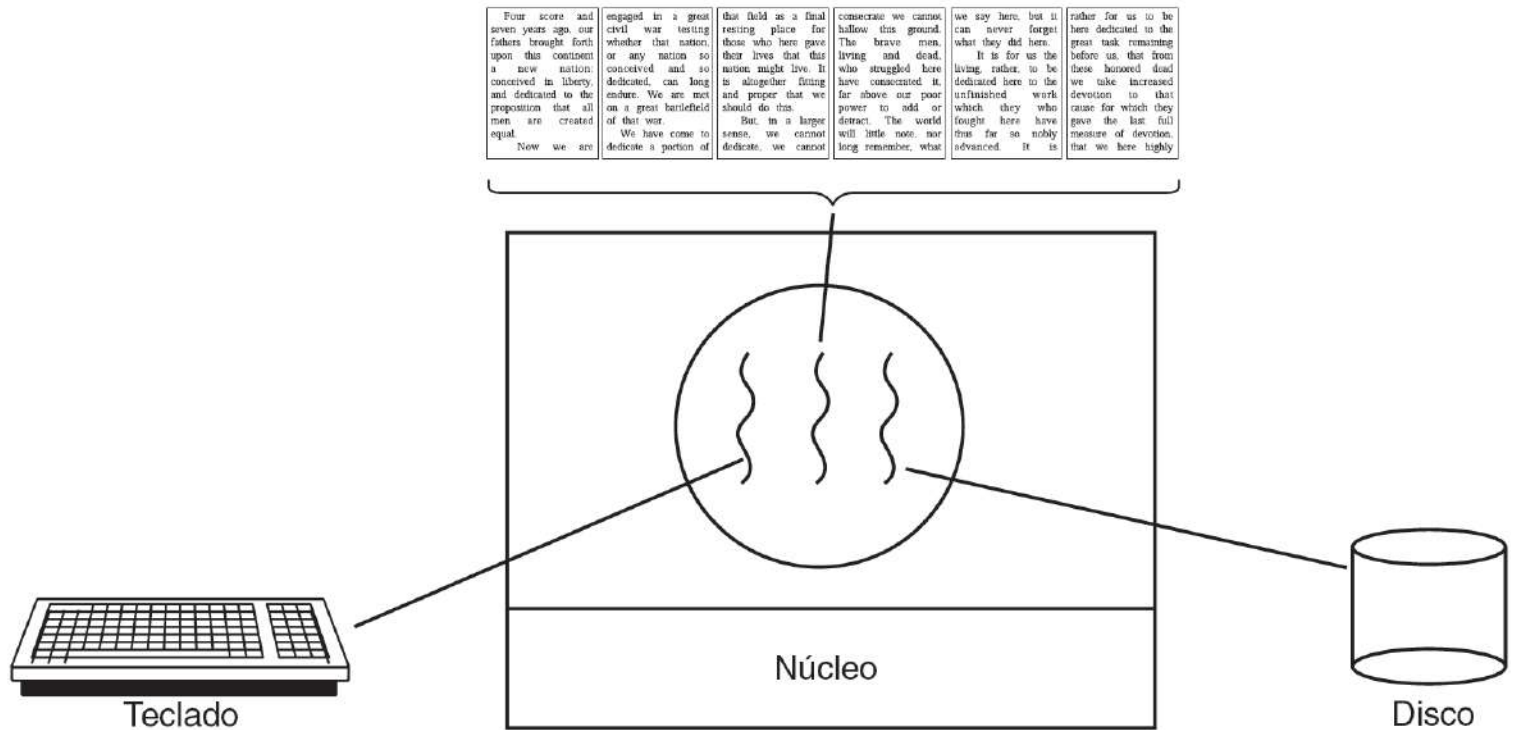


Figura 2.5 Um processador de textos com três threads.

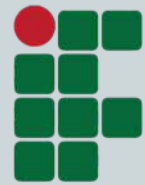
Itens compartilhados x específicos

Itens por processo	Itens por thread
Espaço de endereçamento	Contador de programa
Variáveis globais	Registradores
Arquivos abertos	Pilha
Processos filhos	Estado
Alarmes pendentes	
Sinais e manipuladores de sinais	
Informação de contabilidade	

Tabela 2.4 A primeira coluna lista alguns itens compartilhados por todos os threads em um processo. A segunda lista alguns itens específicos a cada thread.

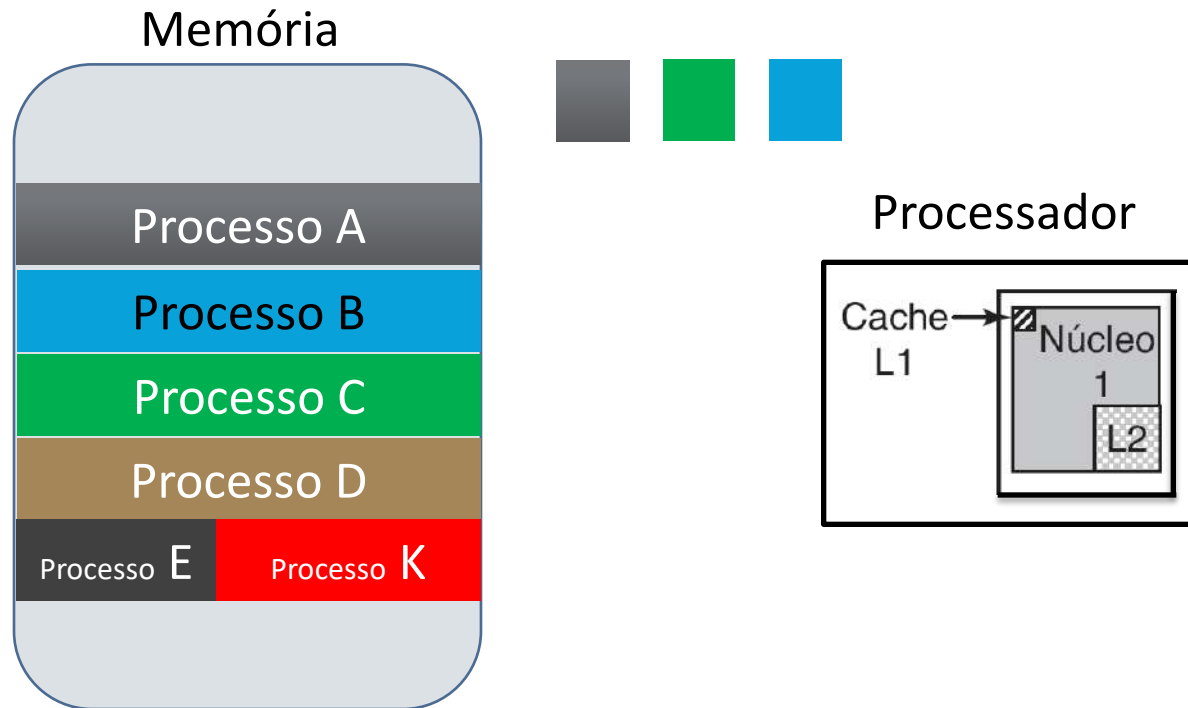
Cada thread executa seu código e compartilha recursos (arquivos abertos, conexões de rede, etc) e espaço de endereçamento

**Como é feita a alternância
de execução entre
processos/threads?**



**INSTITUTO
FEDERAL**
Ceará

Alocação de CPU/ Escalonamento de Processos



1. Um processo pode ser simplesmente mais importante que outro?
2. Quem chegou primeiro é mais importante?
3. O que consome menos tempo de processamento deve ser executado antes dos outros?
4. Será que um processo pode “pegar o processador” por tempo indeterminado?
5. O que acontece com um processo após ficar bloqueado? Pra qual parte da fila vai?
6. Em algum momento específico um processo pode tornar-se mais importante do que outro?

Critérios de escalonamento

- **Utilização de CPU (%)**– mantenha a CPU a mais ocupada possível
- **Throughput** – # de processos que completam sua execução por unidade de tempo
- **Tempo de turnaround**– quantidade de tempo para executar um processo em particular
- **Tempo de espera** – tempo em que um processo esteve esperando na fila de prontos
- **Tempo de resposta** – tempo desde quando uma solicitação foi submetida até a primeira resposta ser produzida, não a saída (para ambiente de tempo compartilhado)

Objetivos

Todos os sistemas

Justiça — dar a cada processo uma porção justa da CPU

Aplicação da política — verificar se a política estabelecida é cumprida

Equilíbrio — manter ocupadas todas as partes do sistema

Sistemas em lote

Vazão (*throughput*) — maximizar o número de tarefas por hora

Tempo de retorno — minimizar o tempo entre a submissão e o término

Utilização de CPU — manter a CPU ocupada o tempo todo

Sistemas interativos

Tempo de resposta — responder rapidamente às requisições

Proporcionalidade — satisfazer às expectativas dos usuários

Sistemas de tempo real

Cumprimento dos prazos — evitar a perda de dados

Previsibilidade — evitar a degradação da qualidade em sistemas multimídia

■ **Tabela 2.8** Alguns objetivos do algoritmo de escalonamento sob diferentes circunstâncias.

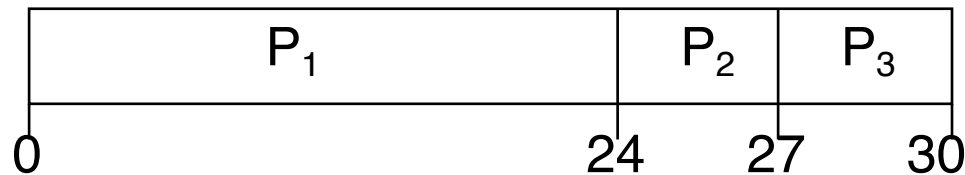
Escalonamento em sistemas em lote

- Primeiro a chegar, primeiro a ser servido (FCFS – *First come First Served*).
- Tarefa mais curta primeiro (SJF – *Shortest Job First*).
- Próximo de menor tempo restante. (SRTF)

Escalonamento First-Come, First-Served (FCFS)

PROCESSO	TEMPO DE BURST
P1	24
P2	3
P3	3

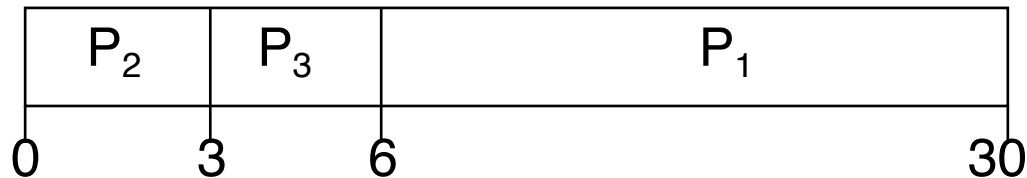
Suponha que os processos cheguem nesta ordem: P_1 , P_2 , P_3
O Gráfico de Gantt para o schedule é:



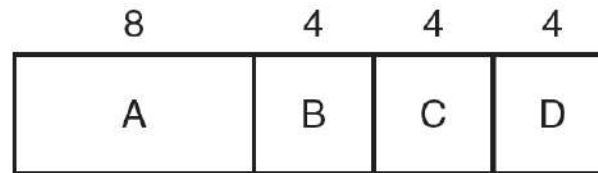
Escalonamento FCFS (cont.)

Suponha que os processos cheguem nesta ordem:
 P_2, P_3, P_1

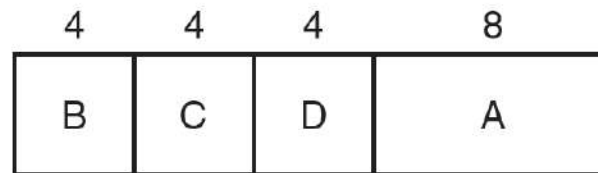
- O Gráfico de Gantt para o schedule é :



SJF (tarefa mais curta primeiro)



(a)

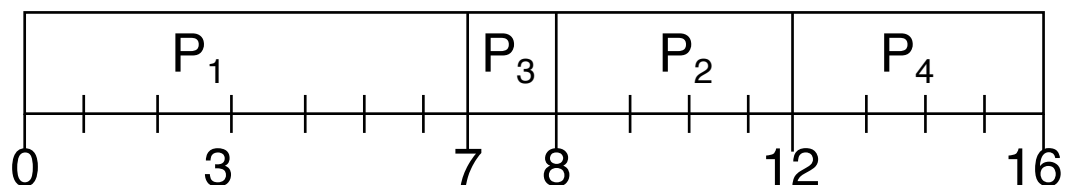


(b)

Figura 2.32 Um exemplo do escalonamento *tarefa mais curta primeiro*. (a) Execução de quatro tarefas na ordem original. (b) Execução na ordem *tarefa mais curta primeiro*.

Exemplo de SJF não preemptivo

PROCESS O	Chegada	TEMPO DE BURST
P1	0,0	7
P2	2,0	4
P3	4,0	1
P4	5,0	4



Escalonamento por prioridades

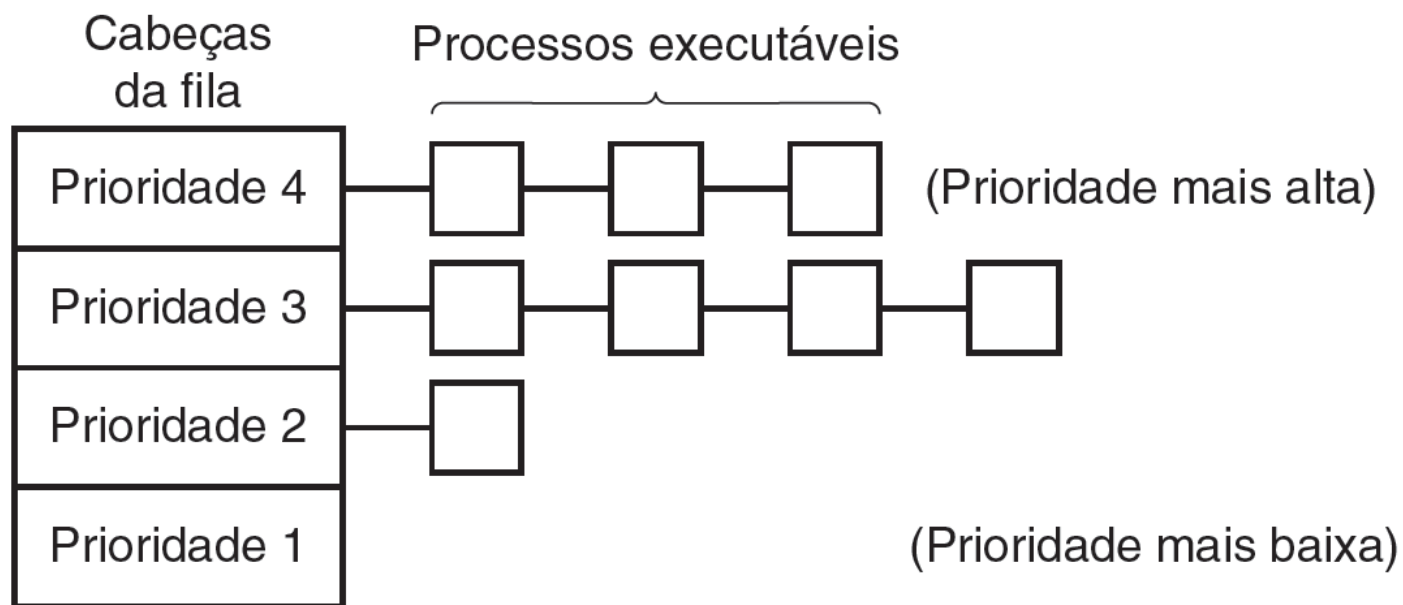


Figura 2.34 Um algoritmo de escalonamento com quatro classes de prioridade.

Round-Robin / Chaveamento Circular

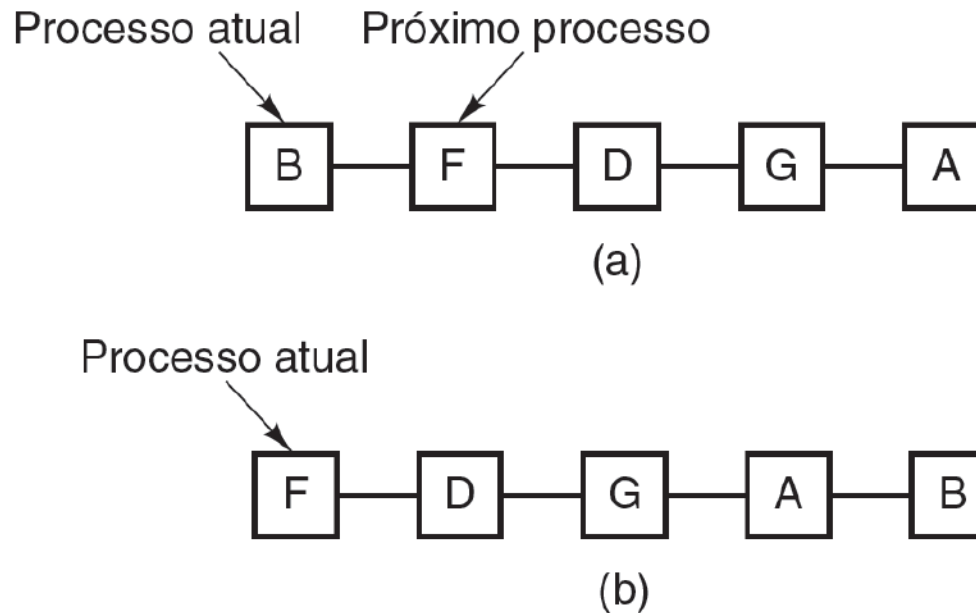


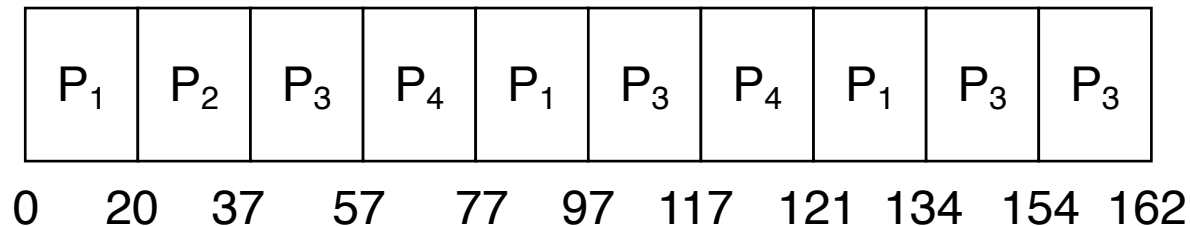
Figura 2.33 Escalonamento circular (*round robin*). (a) Lista de processos executáveis. (b) Lista de processos executáveis depois que *B* usou todo o seu quantum.

Exemplo de RR com quantum de tempo = 20

*processos chegam na sequência P1,P2,P3,P4

PROCESSO	TEMPO DE BURST
P1	53
P2	17
P3	68
P4	24

O gráfico de Gantt é:



Quantum de tempo e tempo de trocas de contexto

