

Sistemas Operacionais

Prof. Leonardo Grando

2025

Objetivo do curso

- ▶ Fundamentos de Sistemas Operacionais
- ▶ Estruturas de Computadores e de Sistemas Operacionais
- ▶ Gerenciamento de Processos
- ▶ Escalonamento de Processador
- ▶ Sincronização de Processos Concorrentes
- ▶ Deadlocks
- ▶ Gerenciamento de Memória
- ▶ Memória Virtual

Tópicos adicionais

- ▶ Sistema de Arquivos
- ▶ Sistema de Entrada e Saída
- ▶ Armazenamento Secundário
- ▶ Estudo da Estrutura de Sistemas Operacionais Existentes

Sistema Computacional

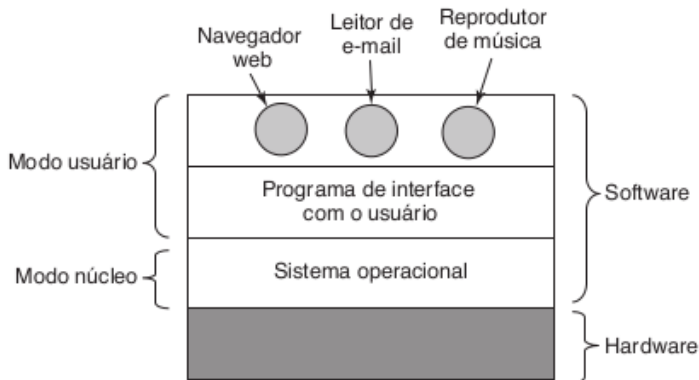


Figure 1: Localização de um SO num sistema computacional

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Interação com o Sistema Operacional

O programa pelo qual os usuários interagem pode ser pelo shell (TUI) ou interface gráfica (GUI).

(TANENBAUM, 2015)

Modo Núcleo vs Modo Usuário

- ▶ **Modo núcleo (supervisor):** Acesso total ao hardware
- ▶ **Modo usuário:** Acesso restrito a um conjunto limitado de instruções, sem permissão para realizar E/S diretamente.

(TANENBAUM, 2015)

O que é um Sistema Operacional?

- ▶ Conjunto de programas que gerenciam os recursos do computador
- ▶ Facilita a interação entre o hardware e os aplicativos ou o usuário
- ▶ Atua como uma camada intermediária entre usuário e hardware

(TANENBAUM, 2015)

Funções principais de um Sistema Operacional

- ▶ **Gerenciamento de processos:** Controla a execução de programas e recursos
- ▶ **Gerenciamento de memória:** Organiza a memória RAM e a alocação de espaço

(TANENBAUM, 2015)

Outras funções principais

- ▶ **Gerenciamento de dispositivos:** Comunicação com periféricos como teclados e impressoras
- ▶ **Gerenciamento de arquivos:** Organiza e controla o acesso a arquivos
- ▶ **Interface com o usuário:** Interface gráfica ou de linha de comando

(TANENBAUM, 2015)

Exemplos de Sistemas Operacionais

- ▶ Windows
- ▶ macOS
- ▶ Linux
- ▶ Android
- ▶ iOS

(TANENBAUM, 2015)

História dos Computadores

- ▶ **1945-1955:** Geração 1: Válvulas, painéis de programação
- ▶ **1955-1965:** Geração 2: Transistores, sistemas em lote
- ▶ **1965-1980:** Geração 3: Circuitos integrados, multiprogramação
- ▶ **1980-presente:** Geração 4: Computadores pessoais
- ▶ **1990-presente:** Geração 5: Computadores móveis

(TANENBAUM, 2015)

Tipos de Sistemas Operacionais

- ▶ SOs para computadores de grande porte
- ▶ SOs de servidores
- ▶ SOs de multiprocessadores
- ▶ SOs de computadores pessoais
- ▶ SOs de tempo real
- ▶ SOs embarcados
- ▶ SOs de cartões inteligentes

(TANENBAUM, 2015)

Hardware

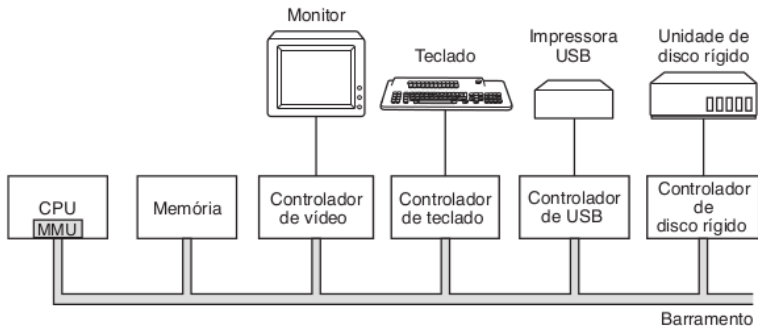


Figure 2: Componentes de um computador pessoal

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Pipeline de execução CPU

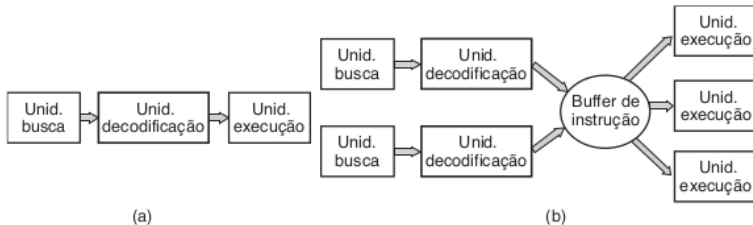


Figure 3: Pipeline de 3 estágios e CPU superescalar

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Multithreading e Multicore

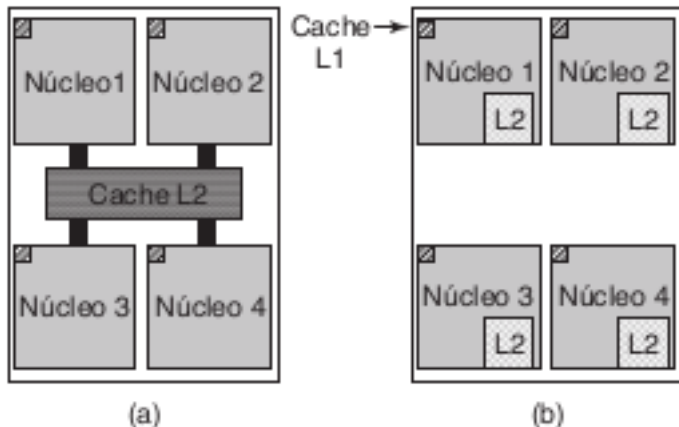


Figure 4: Chip Quadcore e cache L2 compartilhada

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

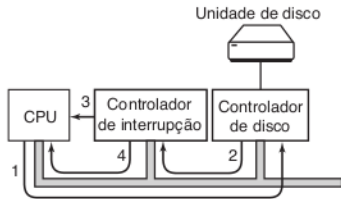
Hierarquia de Memória



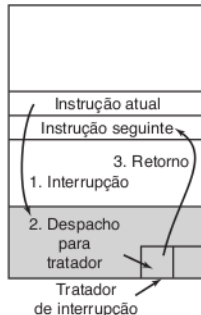
Figure 5: Hierarquia de memória

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

E/S e Barramentos



(a)



(b)

Figure 6: Passos para iniciar um dispositivo E/S e como a CPU é interrompida

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Processos

Processos são programas em execução, associados a um espaço de endereçamento.

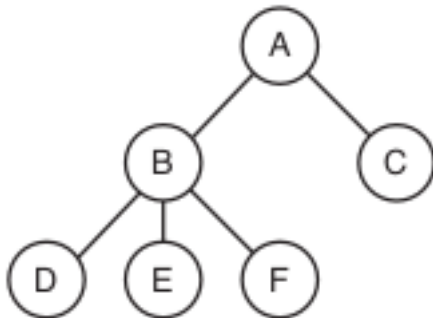
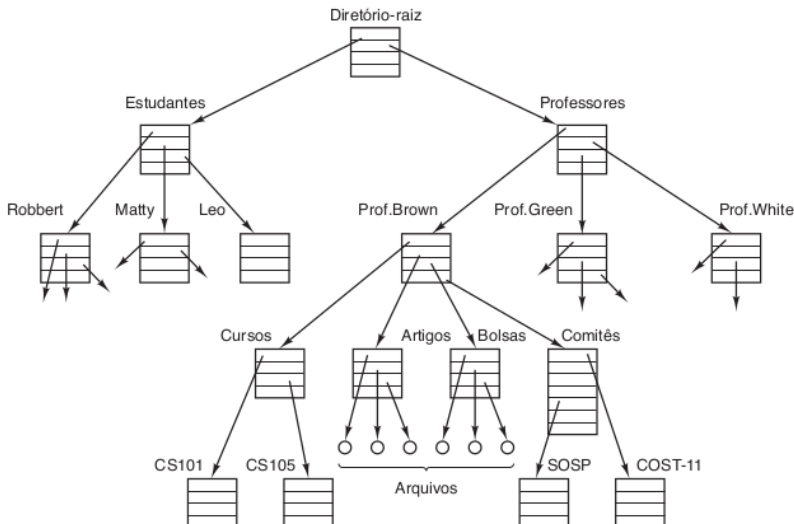


Figure 7: Árvore de processos

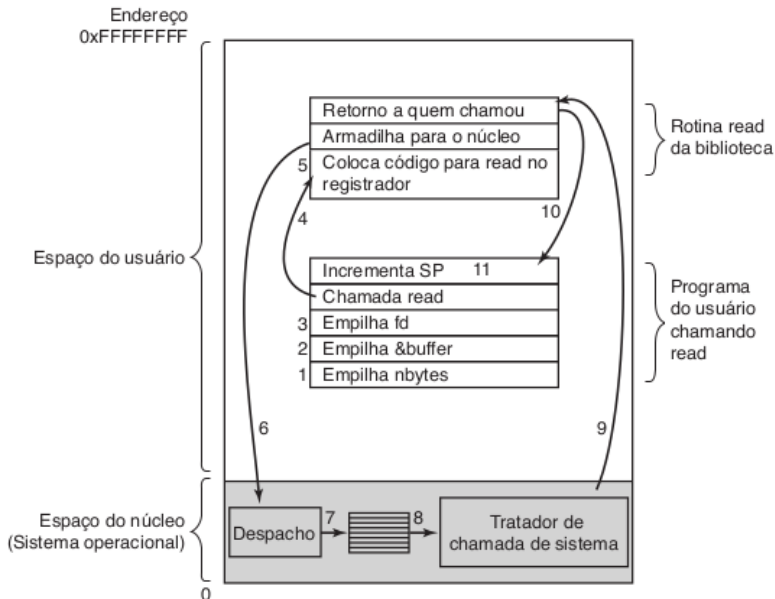
Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Sistemas de Arquivos

- ▶ Diretório: maneira de agrupar arquivos
- ▶ Caminho: ~/Documentos/Git/PaginaWEB - ProfLgrando/proflgrando.github.io



Chamadas de Sistema



Gerenciamento de Arquivos e Diretórios

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

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

Figure 10: Gerenciamento de arquivos e diretórios

Fonte: TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015.

Modelos de Arquitetura de Sistemas Operacionais

1. **Monolíticos**
2. **Camadas**
3. **Micronúcleos**
4. **Clientes-Servidor**
5. **Máquinas Virtuais**
6. **Exonúcleos**

Referências

TANENBAUM, Andrew S. *Sistemas operacionais modernos*. São Paulo, SP: Pearson, 2015. Disponível em:
https://biblioteca.sophia.com.br/9198/index.asp?codigo_sophia=809213.
Acesso em: 2 fev. 2025.