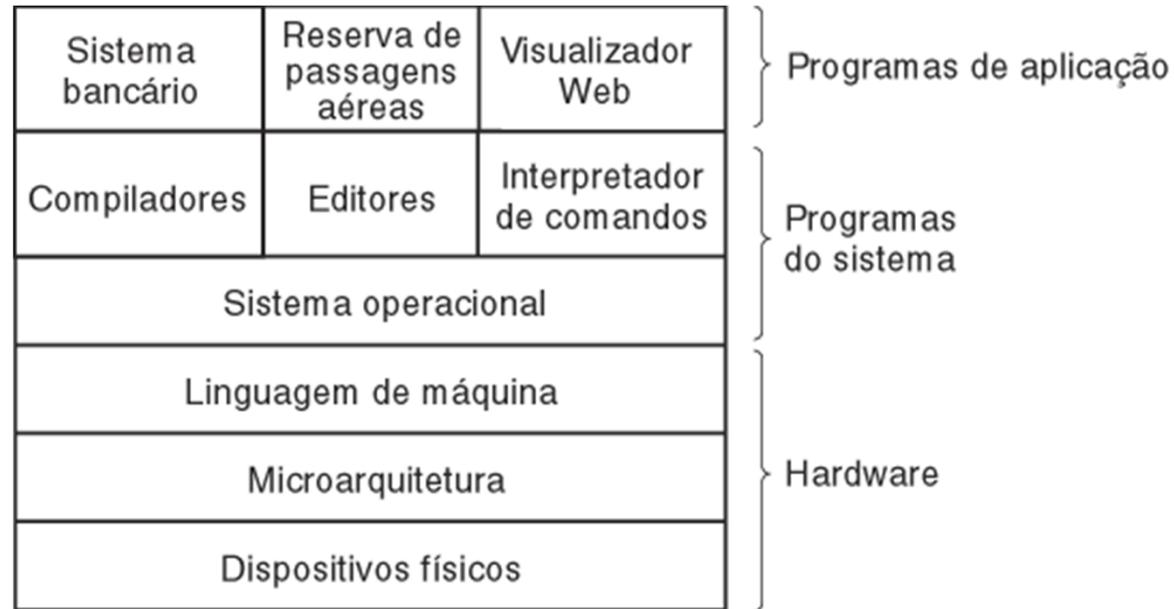
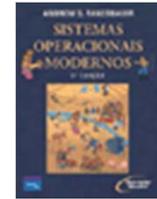


# Capítulo 1

## Introdução

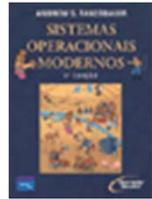
- 1.1 O que é um sistema operacional
- 1.2 História dos sistemas operacionais
- 1.3 O zoológico de sistemas operacionais
- 1.4 Revisão sobre hardware de computadores
- 1.5 Conceitos sobre sistemas operacionais
- 1.6 Chamadas ao sistema
- 1.7 Estrutura de sistemas operacionais

# Introdução

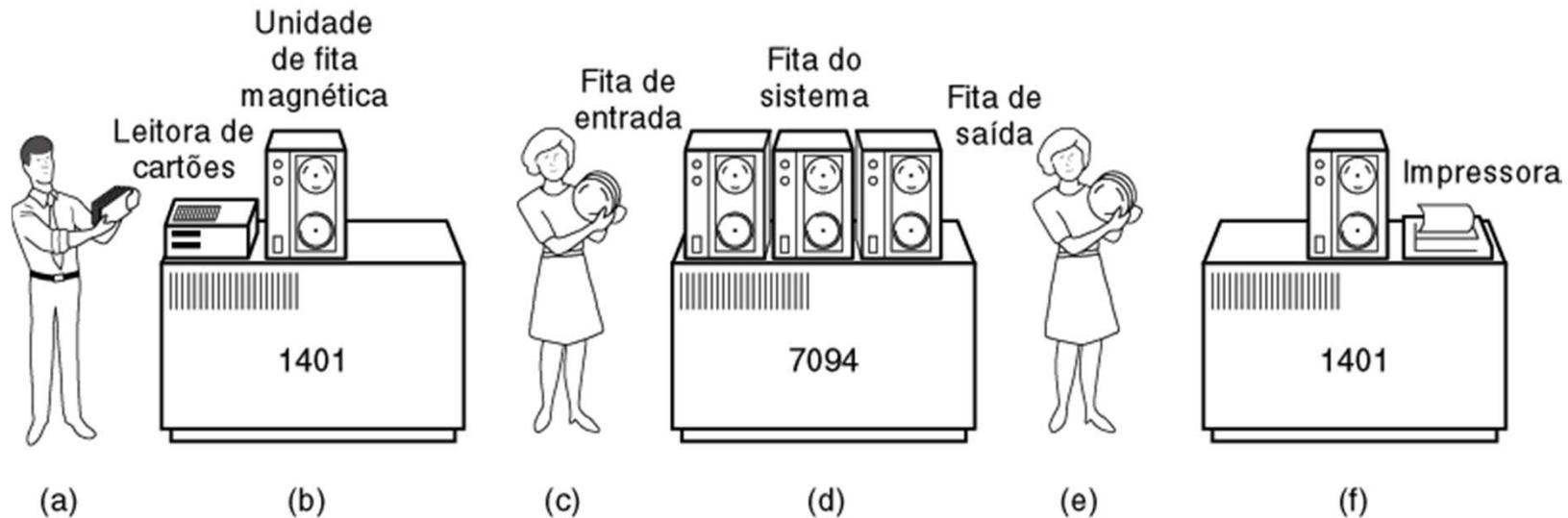
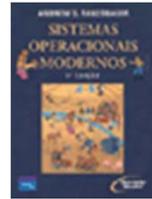


- Um sistema computacional consiste em
  - hardware
  - programas do sistema
  - programas de aplicação

# O que é um Sistema Operacional

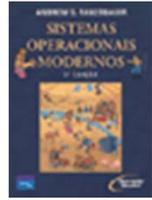


- É uma máquina estendida
  - Oculta os detalhes complicados que têm que ser executados
  - Apresenta ao usuário uma máquina virtual, mais fácil de usar
- É um gerenciador de recurso
  - Cada programa tem um tempo com o recurso
  - Cada programa tem um espaço no recurso



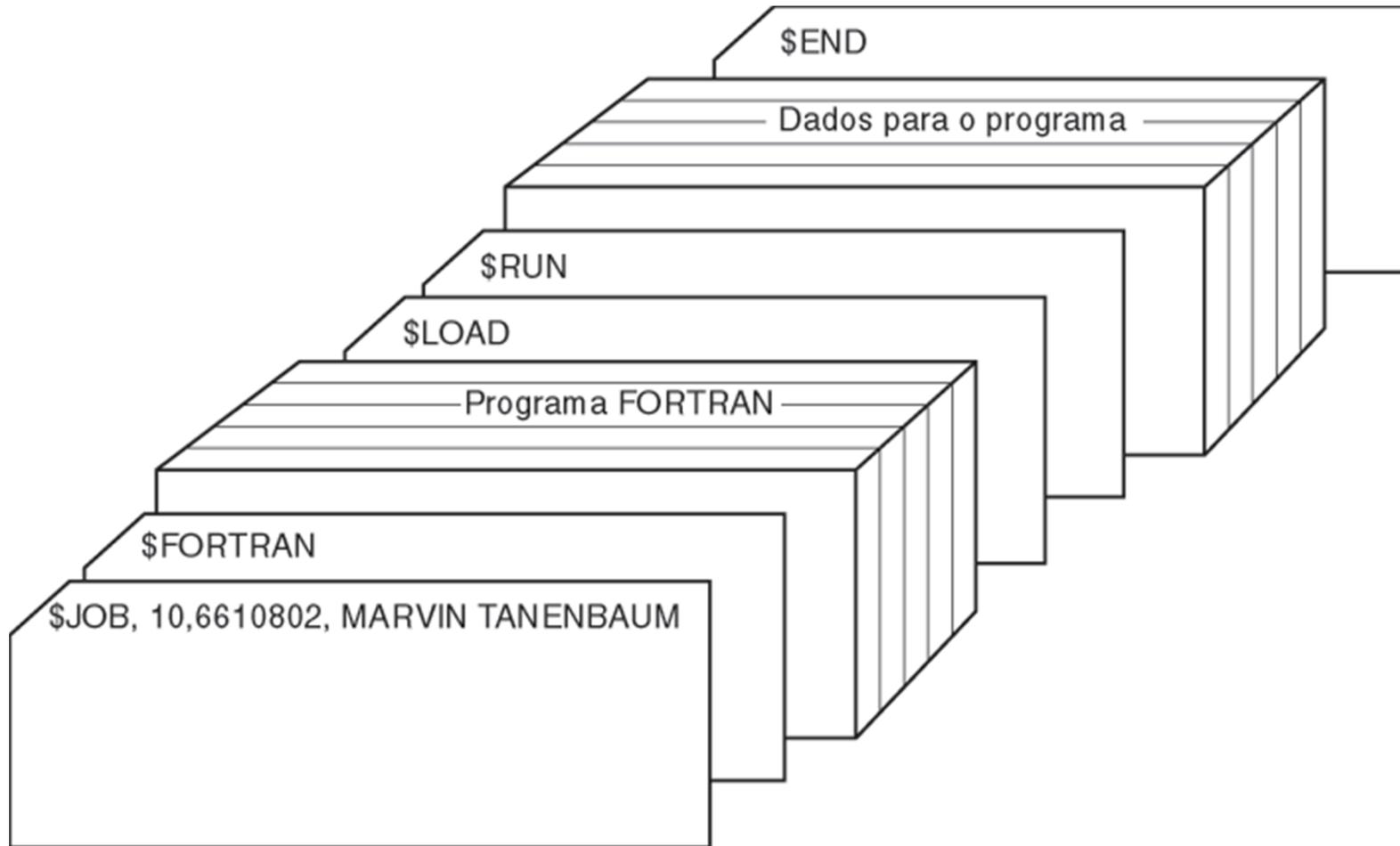
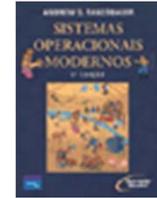
## Antigo sistema em lote

- traz os cartões para o 1401
- lê os cartões para a fita
- coloca a fita no 7094 que executa o processamento
- coloca a fita no 1401 que imprime a saída

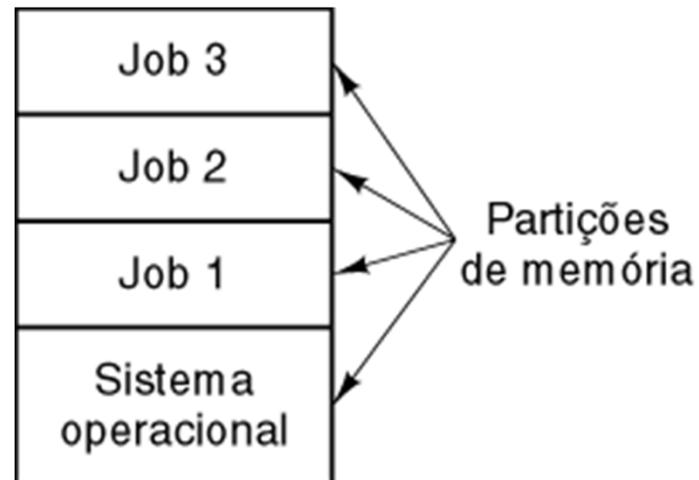
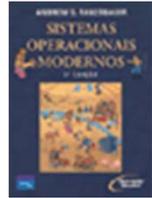


- Primeira geração 1945 - 1955
  - Válvulas, painéis de programação
- Segunda geração 1955 - 1965
  - transistores, sistemas em lote
- Terceira geração 1965 – 1980
  - CIs e multiprogramação
- Quarta geração 1980 – presente
  - Computadores pessoais

# História dos Sistemas Operacionais (3)

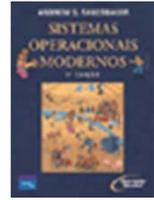


Estrutura de um job FMS típico – 2a. geração



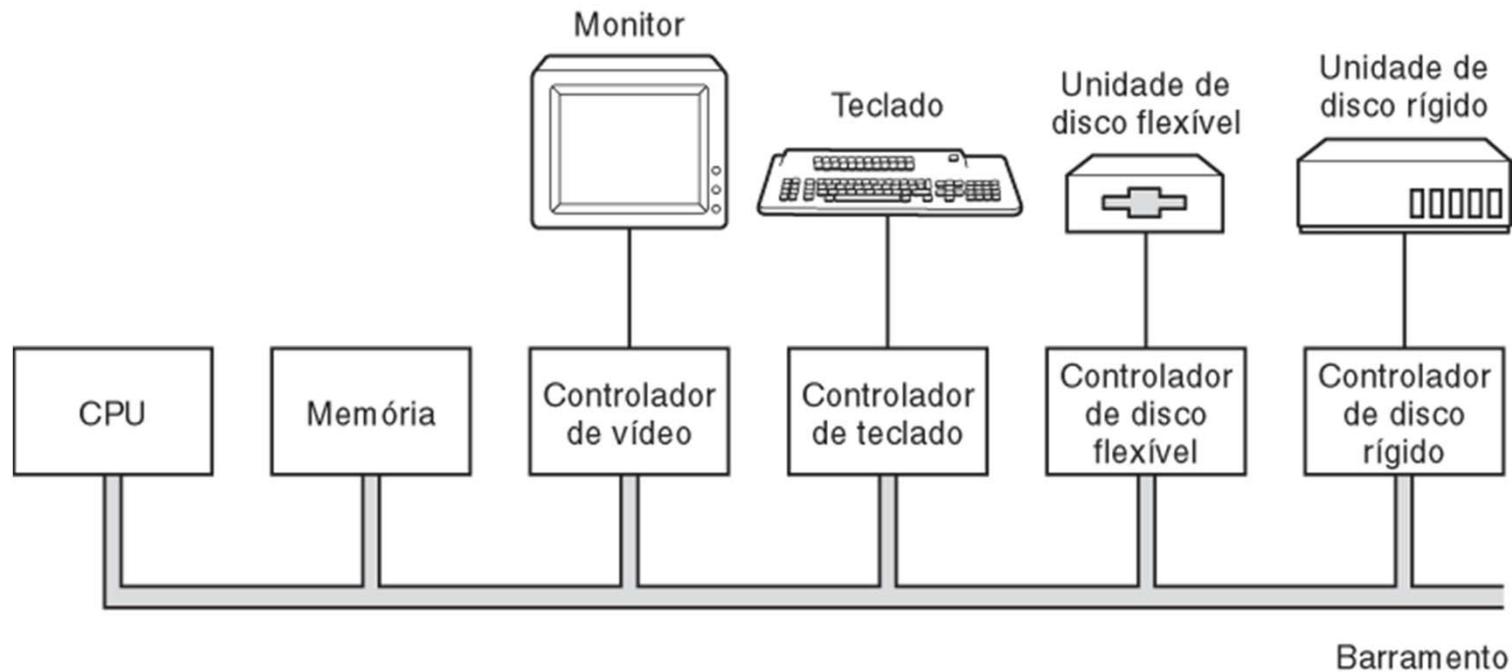
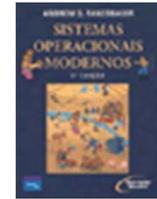
- Sistema de multiprogramação
  - Três jobs na memória – 3a. geração

# O Zoológico de Sistemas Operacionais



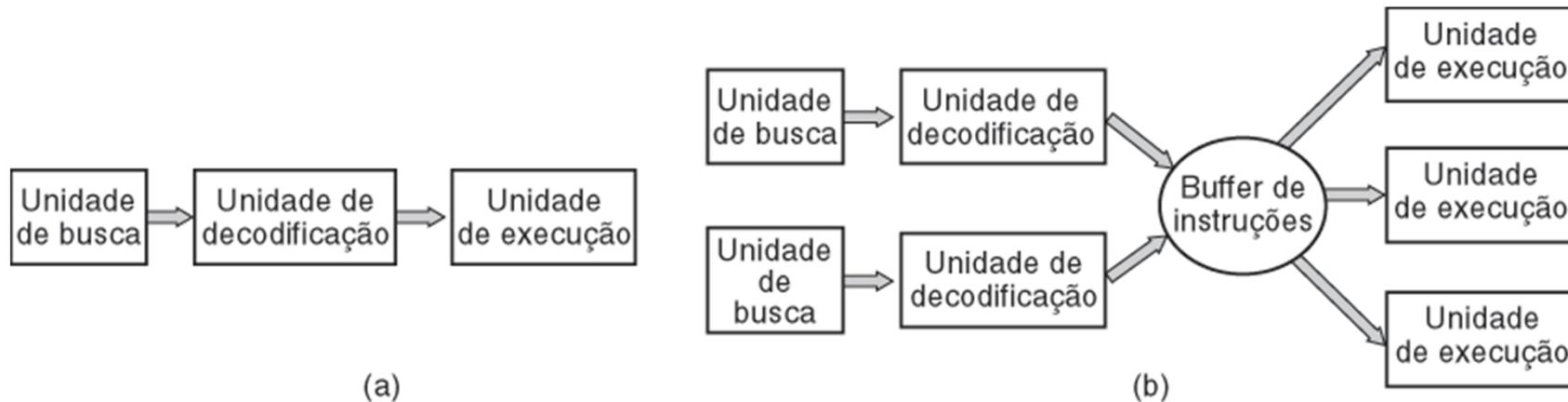
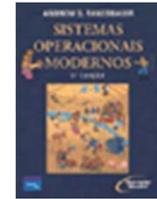
- Sistemas operacionais de computadores de grande porte (OS/390)
- Sistemas operacionais de servidores
- Sistemas operacionais de multiprocessadores
- Sistemas operacionais de computadores pessoais
- Sistemas operacionais de tempo-real (QXN)
- Sistemas operacionais embarcados (Palm OS)
- Sistemas operacionais de cartões inteligentes

# Revisão sobre hardware de computadores (1)



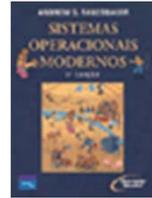
Componentes de um computador pessoal simples

# Revisão sobre hardware de computadores (2)



- (a) Um pipeline de três estágios
- (b) Uma CPU superescalar

# Revisão sobre hardware de computadores (3)



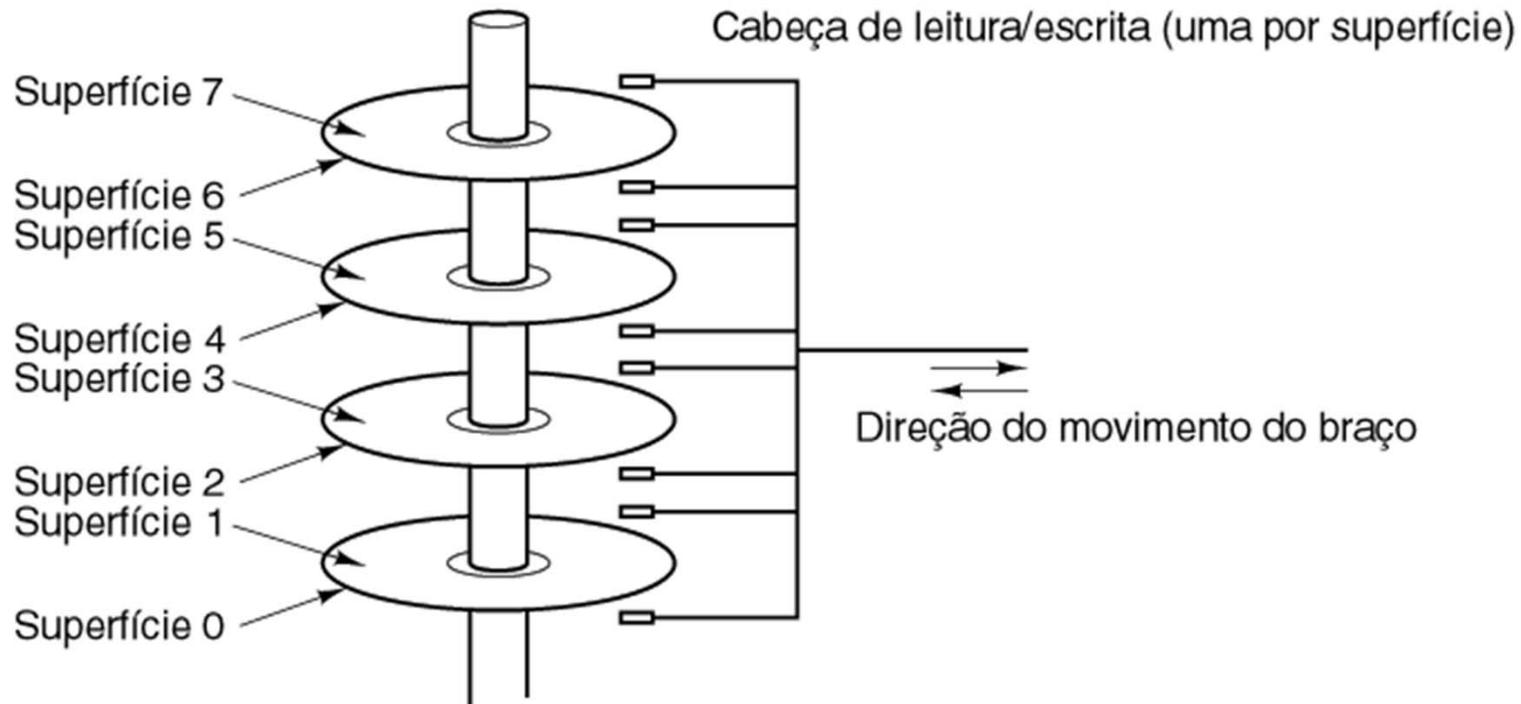
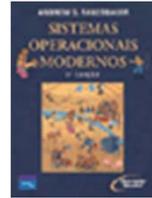
Tempo de acesso típico

Capacidade típica



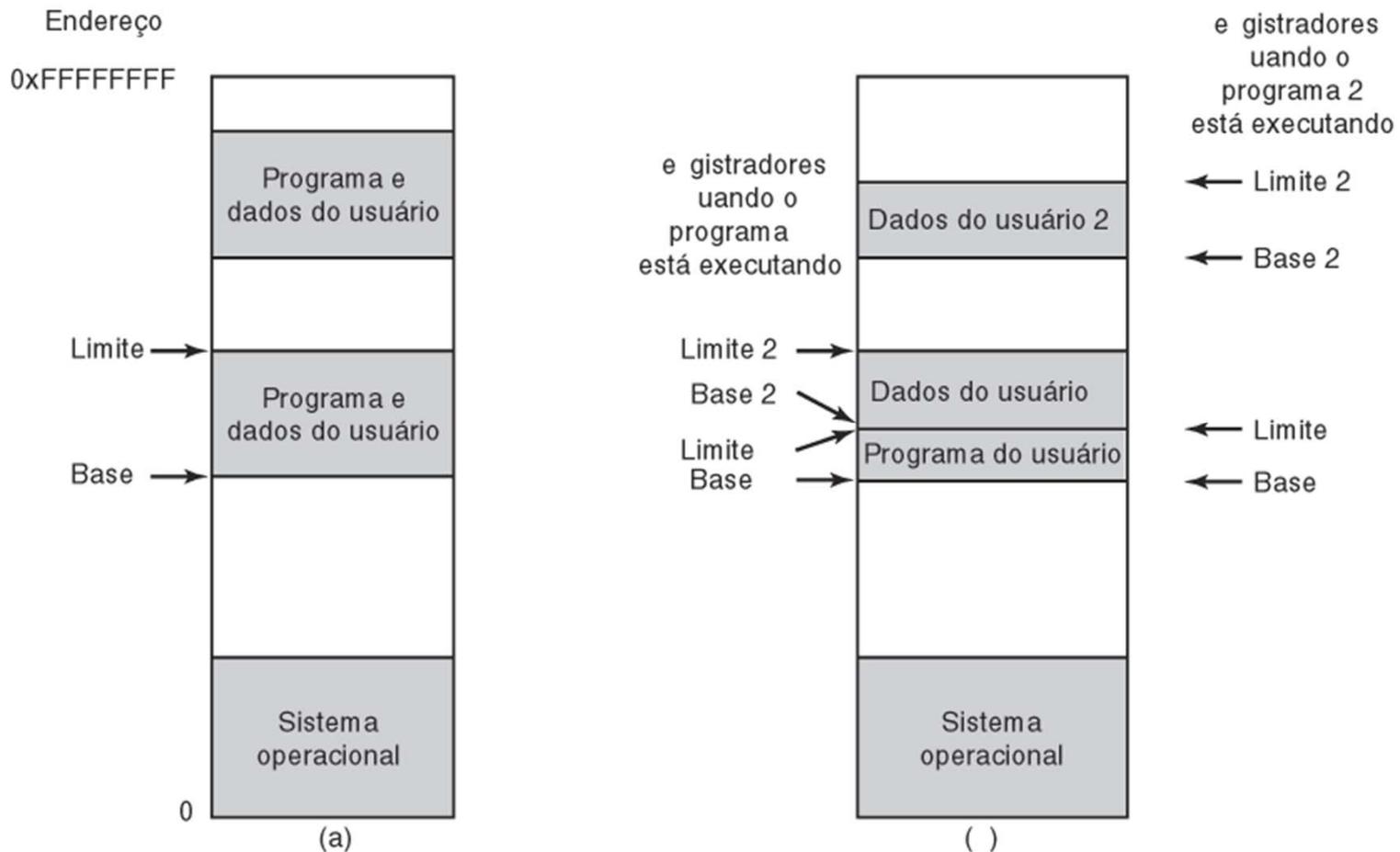
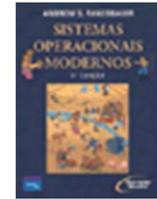
- Típica hierarquia de memória
  - números mostrados são apenas aproximações

# Revisão sobre hardware de computadores(4)



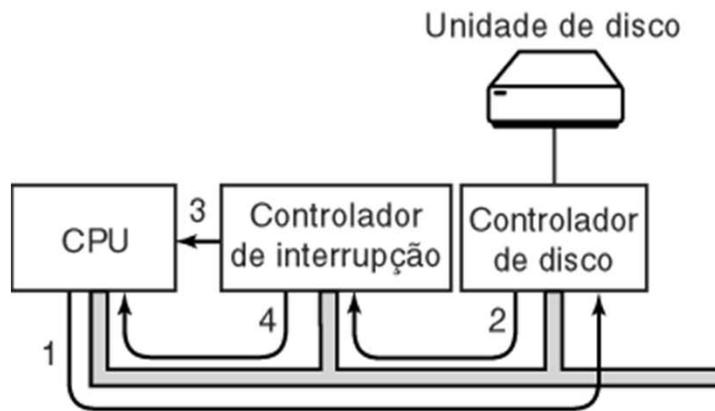
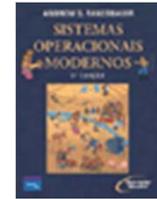
## Estrutura de uma unidade de disco

# Revisão sobre hardware de computadores (5)

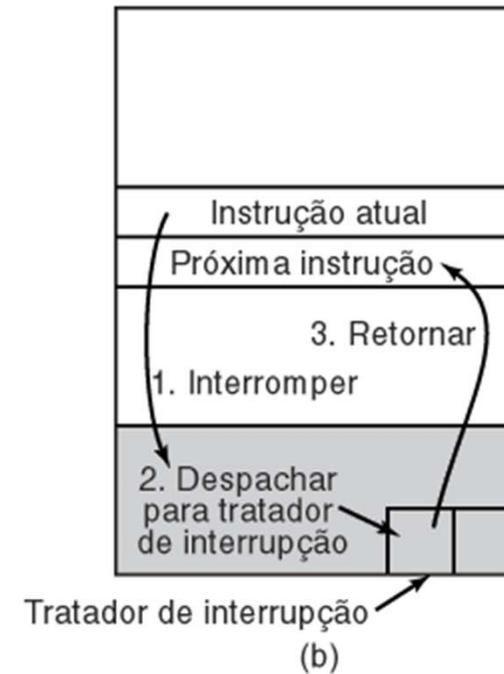


Um par base-limite e dois pares base-limite

# Revisão sobre hardware de computadores (6)



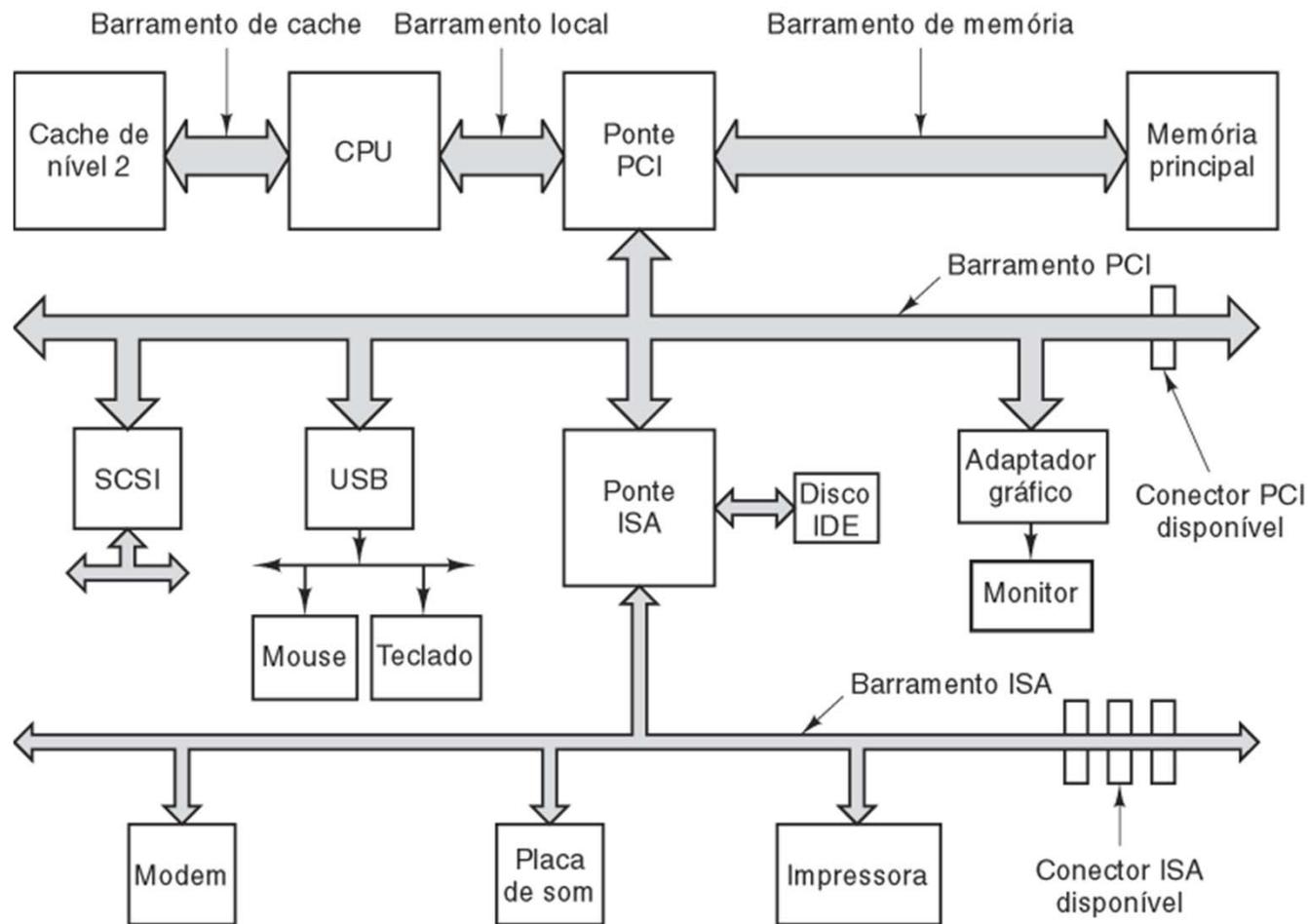
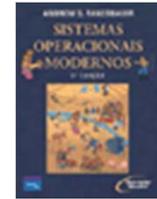
(a)



(b)

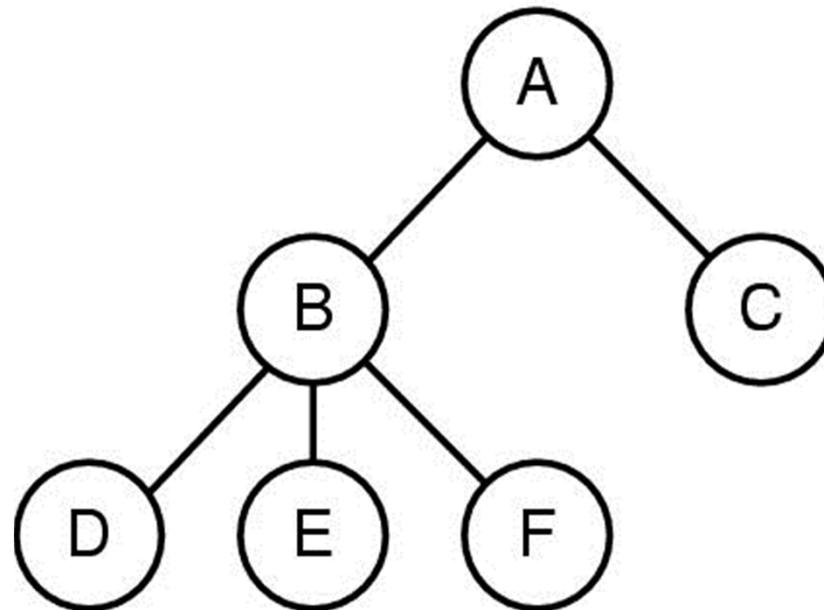
- (a) Passos para iniciar um dispositivo de E/S e obter uma interrupção
- (b) Como a CPU é interrompida

# Revisão sobre hardware de computadores(7)



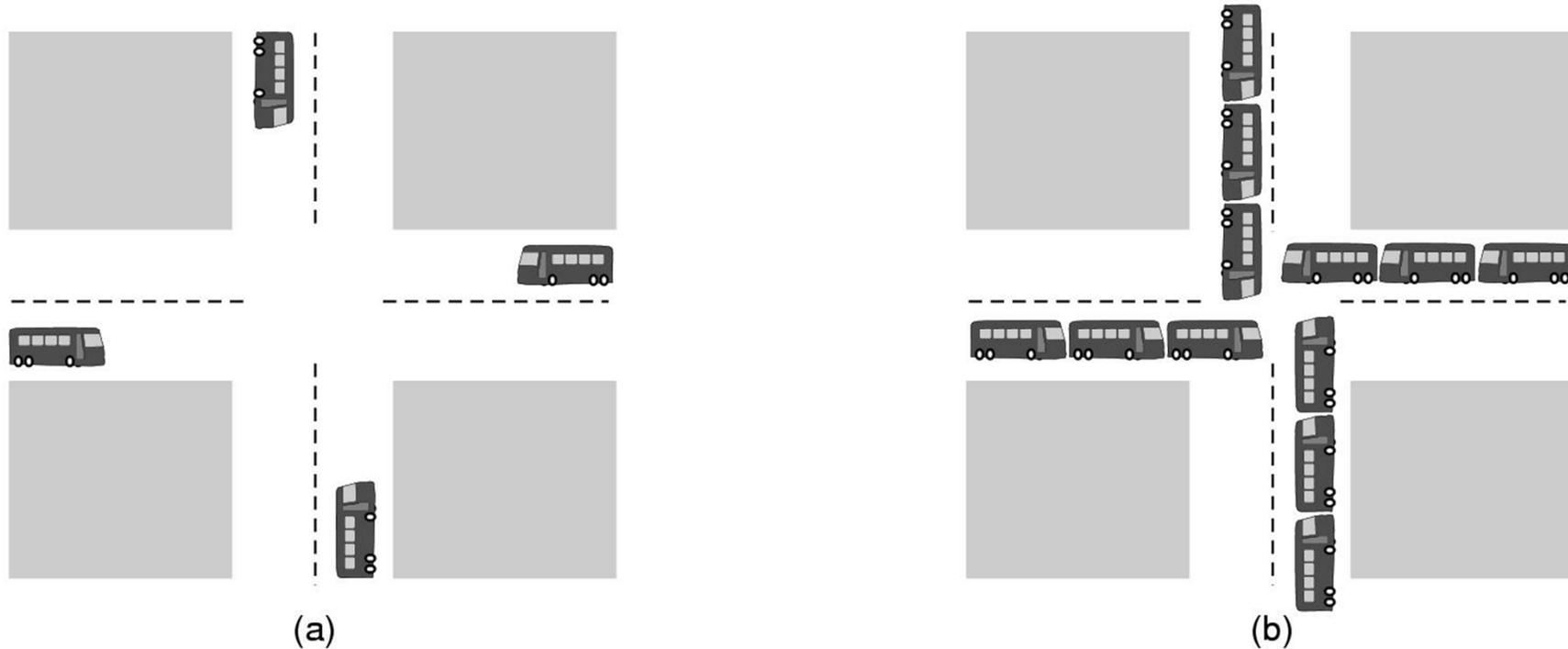
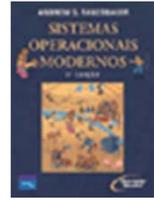
## Estrutura de um sistema Pentium grande

# Conceitos sobre Sistemas Operacionais (1)



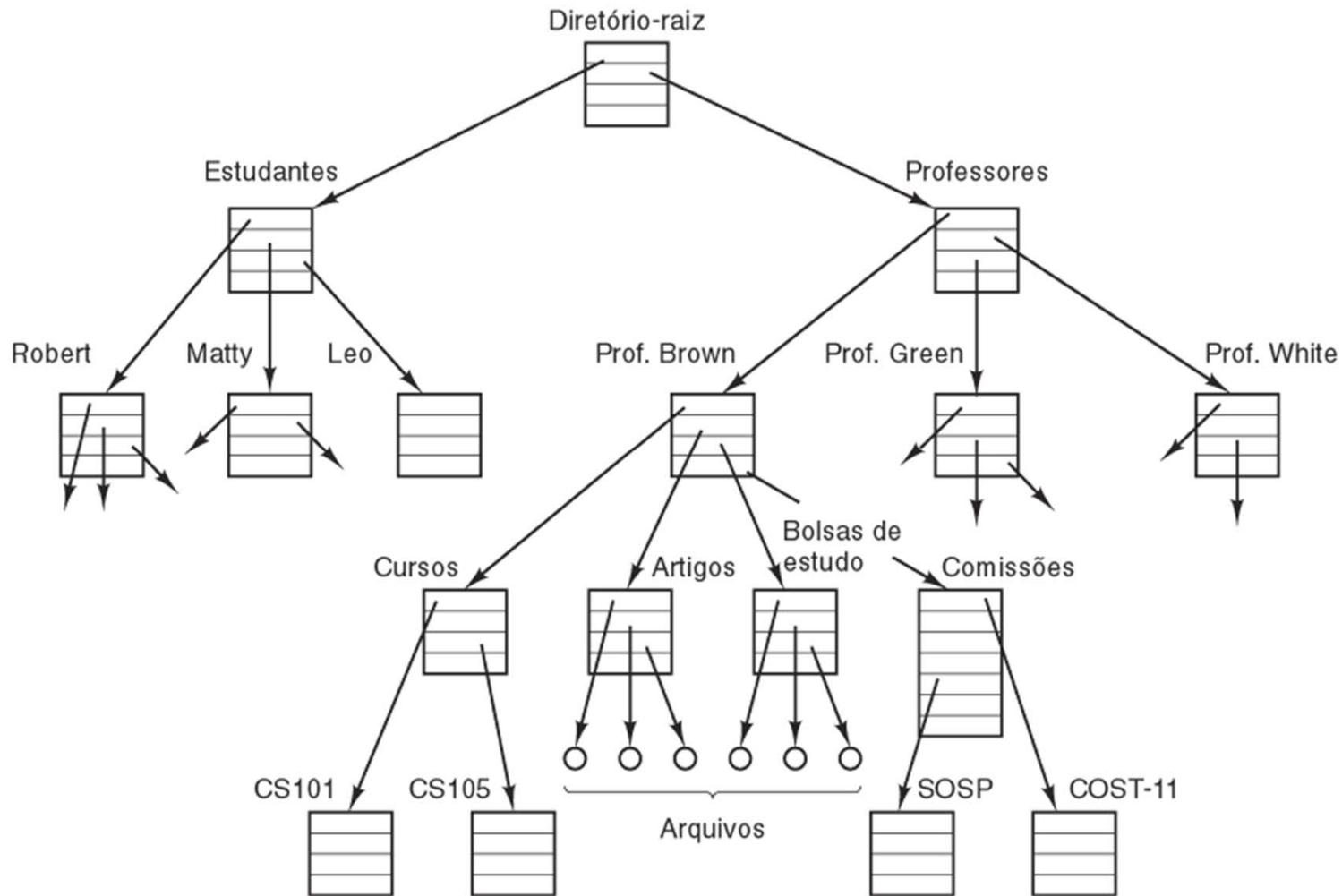
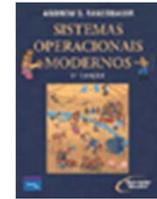
- Uma árvore de processos
  - A criou dois processos filhos: B e C
  - B criou três processos filhos: D, E, e F

# Conceitos sobre Sistemas Operacionais (2)



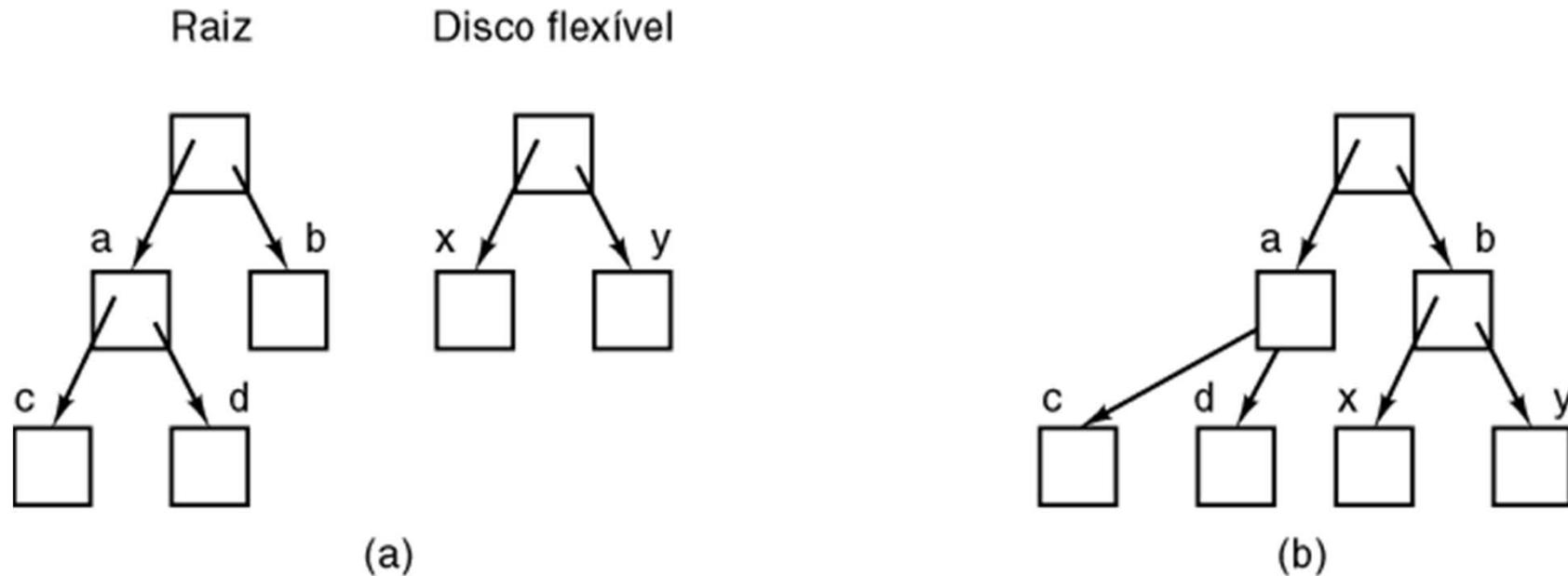
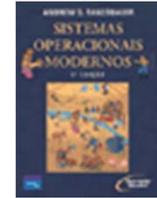
(a) Um deadlock potencial. (b) um deadlock real.

# Conceitos sobre Sistemas Operacionais (3)



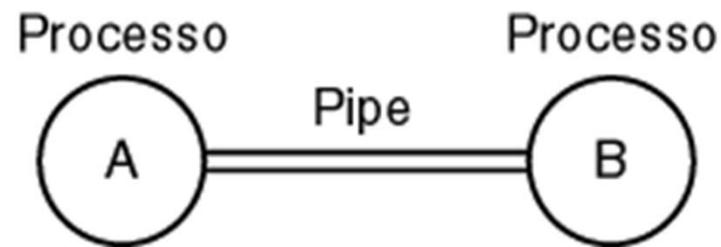
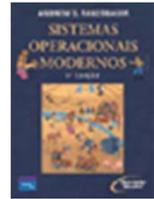
Sistema de arquivos de um departamento universitário

# Conceitos sobre Sistemas Operacionais (4)



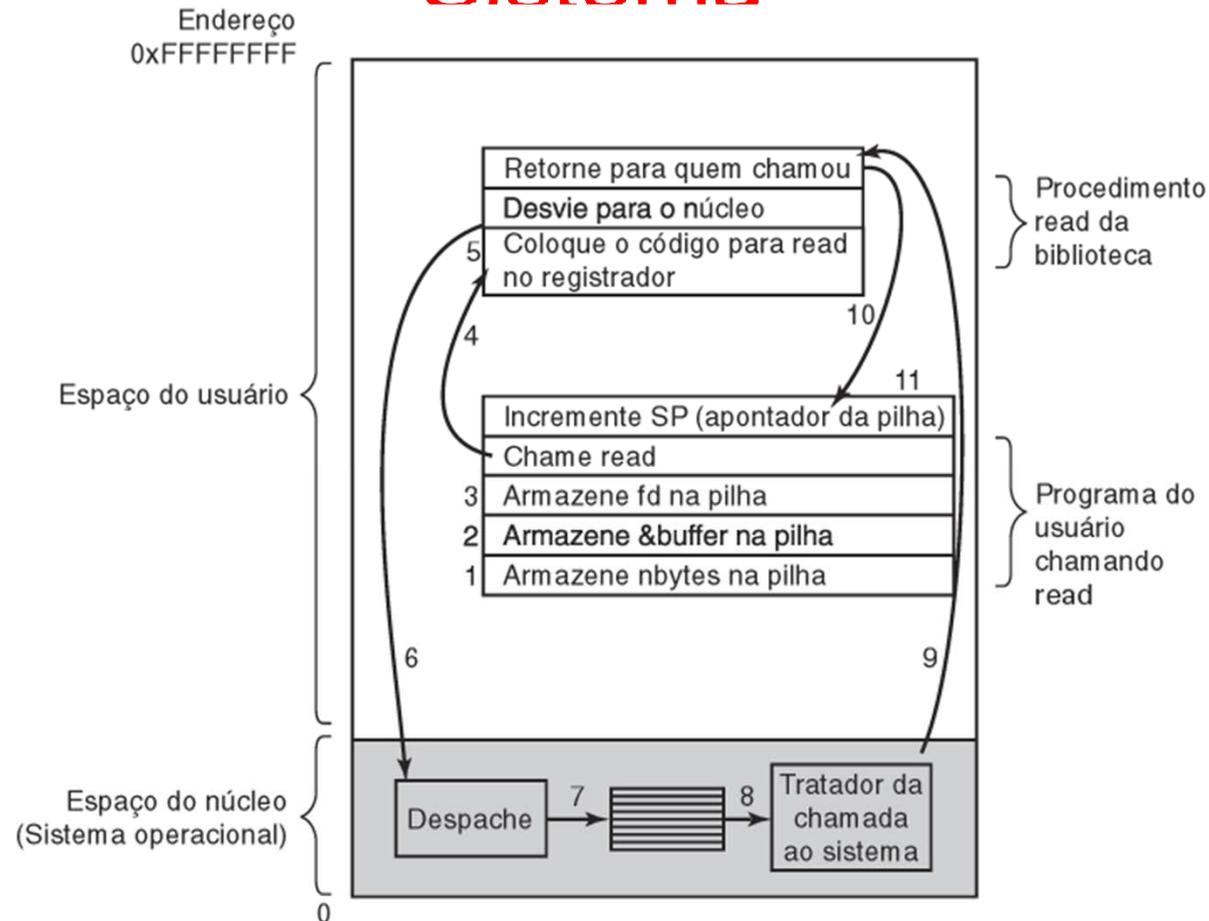
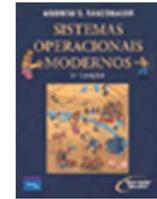
- Antes da montagem,
  - os arquivos do disco flexível são inacessíveis
- Depois da montagem do disco flexível em b,
  - os arquivos do disco fazem parte da hierarquia de arquivos

# Conceitos sobre Sistemas Operacionais (5)



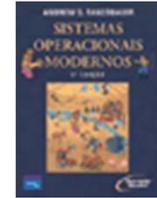
Dois processos conectados por um pipe

# Os Passos de uma Chamada ao Sistema



Os 11 passos para fazer uma chamada ao sistema read (fd, buffer, nbytes)

# Algumas Chamadas ao Sistema para Gerenciamento de Processos



## Gerenciamento de processos

Chamada	Descrição
<code>pid = fork( )</code>	Crie um processo filho idêntico ao processo pai
<code>pid = waitpid(pid, &amp;statloc, options)</code>	Aguarde um processo filho terminar
<code>s = execve(name, argv, environp)</code>	Substitua o espaço de endereçamento do processo
<code>exit(status)</code>	Termine a execução do processo e retorne o estado

# Algumas Chamadas ao Sistema para Gerenciamento de Arquivos



## Gerenciamento de arquivos

Chamada	Descrição
<code>fd = open(file, how, ...)</code>	Abra um arquivo para leitura, escrita ou ambas
<code>s = close(fd)</code>	Feche um arquivo aberto
<code>n = read(fd, buffer, nbytes)</code>	Leia dados de um arquivo para um buffer
<code>n = write(fd, buffer, nbytes)</code>	Escreva dados de um buffer para um arquivo
<code>position = lseek(fd, offset, whence)</code>	Mova o ponteiro de posição do arquivo
<code>s = stat(name, &amp;buf)</code>	Obtenha a informação de estado do arquivo

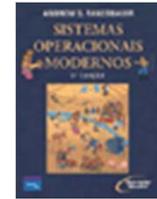
# Algumas Chamadas ao Sistema para Gerenciamento de Diretório



## Gerenciamento do sistema de diretório e arquivo

Chamada	Descrição
s = mkdir(name, mode)	Crie um novo diretório
s = rmdir(name)	Remova um diretório vazio
s = link(name1, name2)	Crie uma nova entrada, name2, apontando para name1
s = unlink(name)	Remova uma entrada de diretório
s = mount(special, name, flag)	Monte um sistema de arquivo
s = umount(special)	Desmonte um sistema de arquivo

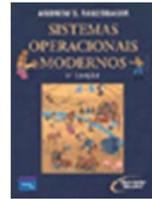
# Algumas Chamadas ao Sistema para Tarefas Diversas



## Diversas

Chamada	Descrição
<code>s = chdir(dirname)</code>	Altere o diretório de trabalho
<code>s = chmod(name, mode)</code>	Altere os bits de proteção do arquivo
<code>s = kill(pid, signal)</code>	Envie um sinal a um processo
<code>seconds = time(&amp;seconds)</code>	Obtenha o tempo decorrido desde 1º de janeiro de 1970

# Chamadas ao Sistema (1)



- O interior de um shell:

```
#define TRUE 1

while (TRUE) {
    type_prompt( );
    read_command(command, parameters);

    if (fork( ) !=0) {
        /* Parent code. */
        waitpid(-1, *status, 0);
    } else {
        /* Child code. */
        execve(command, parameters, 0);
    }
}

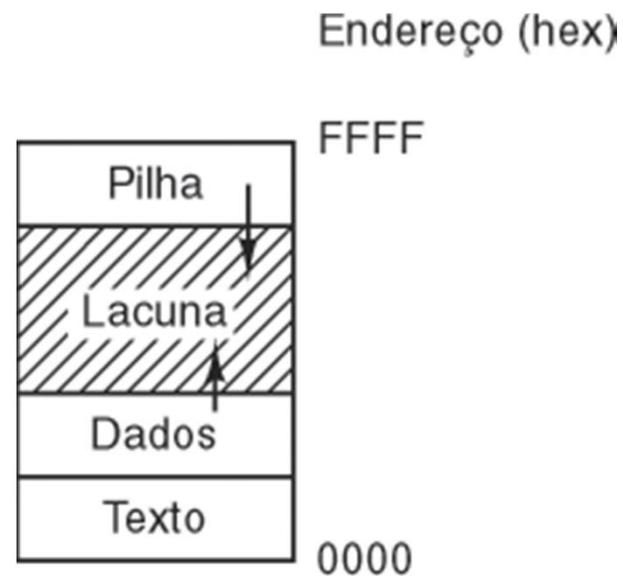
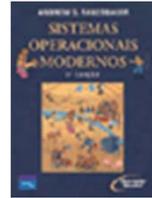
/* repita para sempre */
/* mostra prompt na tela */
/* lê entrada do terminal */

/* cria processo filho */

/* aguarda o processo filho acabar */

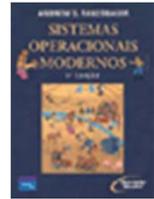
/* executa o comando */
```

# Chamadas ao Sistema (2)



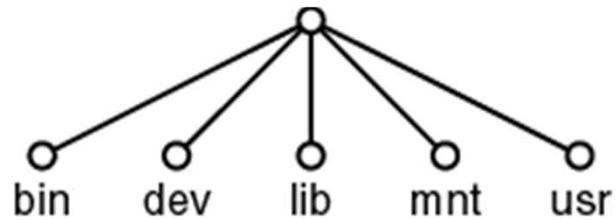
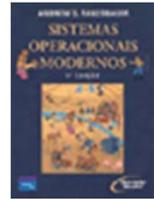
Os processos têm três segmentos:  
texto, dados e pilha

# Chamadas ao Sistema (3)

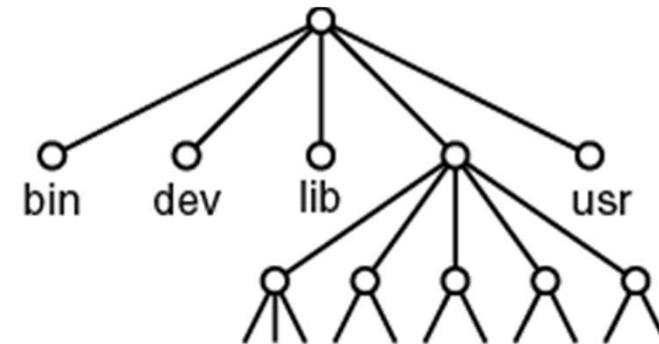


- (a) Dois diretórios antes da ligação de */usr/jim/memo* ao diretório *ast*
- (b) Os mesmos diretórios depois dessa ligação

# Chamadas ao Sistema (4)



(a)



(b)

(a) Sistema de arquivos antes da montagem

(b) Sistema de arquivos depois da montagem

# Chamadas ao Sistema (5)



Unix	Win32	Descrição
fork	CreateProcess	Crie um novo processo
waitpid	WaitForSingleObject	Pode esperar um processo sair
execve	(none)	CrieProcesso = fork + execve
exit	ExitProcess	Termine a execução
open	CreateFile	Crie um arquivo ou abra um arquivo existente
close	CloseHandle	Feche um arquivo
read	ReadFile	Leia dados de um arquivo
write	WriteFile	Escreva dados para um arquivo
lseek	SetFilePointer	Mova o ponteiro de posição do arquivo
stat	GetFileAttributesEx	Obtenha os atributos do arquivo
mkdir	CreateDirectory	Crie um novo diretório
rmdir	RemoveDirectory	Remova um diretório vazio
link	(none)	Win32 não suporta ligações (link)
unlink	DeleteFile	Destrua um arquivo existente
mount	(none)	Win32 não suporta mount
umount	(none)	Win32 não suporta mount
chdir	SetCurrentDirectory	Altere o diretório de trabalho atual
chmod	(none)	Win32 não suporta segurança (embora NT suporte)
kill	(none)	Win32 não suporta sinais
time	GetLocalTime	Obtenha o horário atual

## Algumas chamadas da interface API Win32