

Gerenciamento de memória

1

Prof. Me. Rodrigo Brito Battilana

Roteiro

- Introdução
- Espaço de endereçamento e Registradores
- Alocação de memória
- Memória nos discos
- Área de troca (swapping)
- Memória Virtual
- Paginação
- Segmentação

Roteiro

- Dispositivos de E/S
 - Tipos de Instrução
 - Dispositivos externos
 - Etapas de E/S
 - Diagrama do módulo de E/S
 - Decisões do módulo de E/S
 - Mapeamento de E/S
 - E/S controlada por interrupção
 - Processamento de interrupção simples
 - Fluxo de Controle

Roteiro

- Sistemas de Arquivos
 - Introdução
 - Arquivos
 - Diretórios
 - Esquemas do sistema de arquivos
 - Exemplos de sistemas de arquivos

Introdução

- A abstração de memória mais simples é não ter abstração alguma.
- Até 1980 os computadores não tinham abstração de memória. Cada programa apenas via a memória física, mesmo assim, várias opções eram possíveis

Três maneiras simples de organizar a memória com um sistema operacional e um processo de usuário. Também existem outras possibilidades.



Introdução

- Embora o endereçamento direto de memória física seja apenas uma memória distante nos computadores de grande porte, minicomputadores, computadores de mesa, notebooks e smartphones, a falta de uma abstração de memória ainda é comum em sistemas embarcados e de cartões inteligentes.
- Casos em que o software se endereça à memória absoluta: rádios, máquinas de lavar roupas e fornos de micro-ondas.
- Smartphones, por exemplo, possuem sistemas operacionais elaborados.

7 Espaços de endereçamento e registradores

Espaços de endereçamento

- ▶ Um **espaço de endereçamento** é o conjunto de endereços que um processo pode usar para endereçar a memória. Cada processo tem seu próprio espaço de endereçamento, independente daqueles pertencentes a outros processos (exceto em algumas circunstâncias especiais onde os processos querem compartilhar seus espaços de endereçamento).
- ▶ Exemplos:
 - ▶ - números de telefones;
 - ▶ - portas de E/S;
 - ▶ - endereços de IPv4;
 - ▶ - conjunto de domínios da internet *.com*.

Registradores base e registradores limite

- Quando esses registradores são usados, os programas são carregados em posições de memória consecutivas sempre que haja espaço e sem realocação durante o carregamento.
- Usar registradores base e limite é uma maneira fácil de dar a cada processo seu próprio espaço de endereçamento privado, pois cada endereço de memória gerado automaticamente tem o conteúdo do registrador base adicionado a ele antes de ser enviado para a memória.
- Desvantagem: necessidade de realizar uma adição e uma comparação em cada referência de memória.

Alocação de Memória

Alocação de memória

- Se os processos são criados com um tamanho fixo que nunca muda, então a alocação é simples: o sistema operacional aloca exatamente o que é necessário, nem mais nem menos. No entanto, um problema ocorre sempre que um processo tenta crescer.
- Se houver um espaço adjacente ao processo, ele poderá ser alocado e o processo será autorizado a crescer naquele espaço.
- Por outro lado, se o processo for adjacente a outro, aquele que cresce terá de ser movido para um espaço na memória grande o suficiente para ele, ou um ou mais processos terão de ser trocados para criar um espaço grande o suficiente.

Alocação de memória

- Se um processo não puder crescer em memória e a área de troca no disco estiver cheia, ele terá de ser suspenso até que algum espaço seja liberado (ou ele pode ser morto).
- Se o esperado for que a maioria dos processos cresça à medida que são executados, provavelmente seja uma boa ideia alocar um pouco de memória extra sempre que um processo for trocado ou movido, para reduzir a sobrecarga associada com a troca e movimentação dos processos que não cabem mais em sua memória alocada.

Memória nos discos

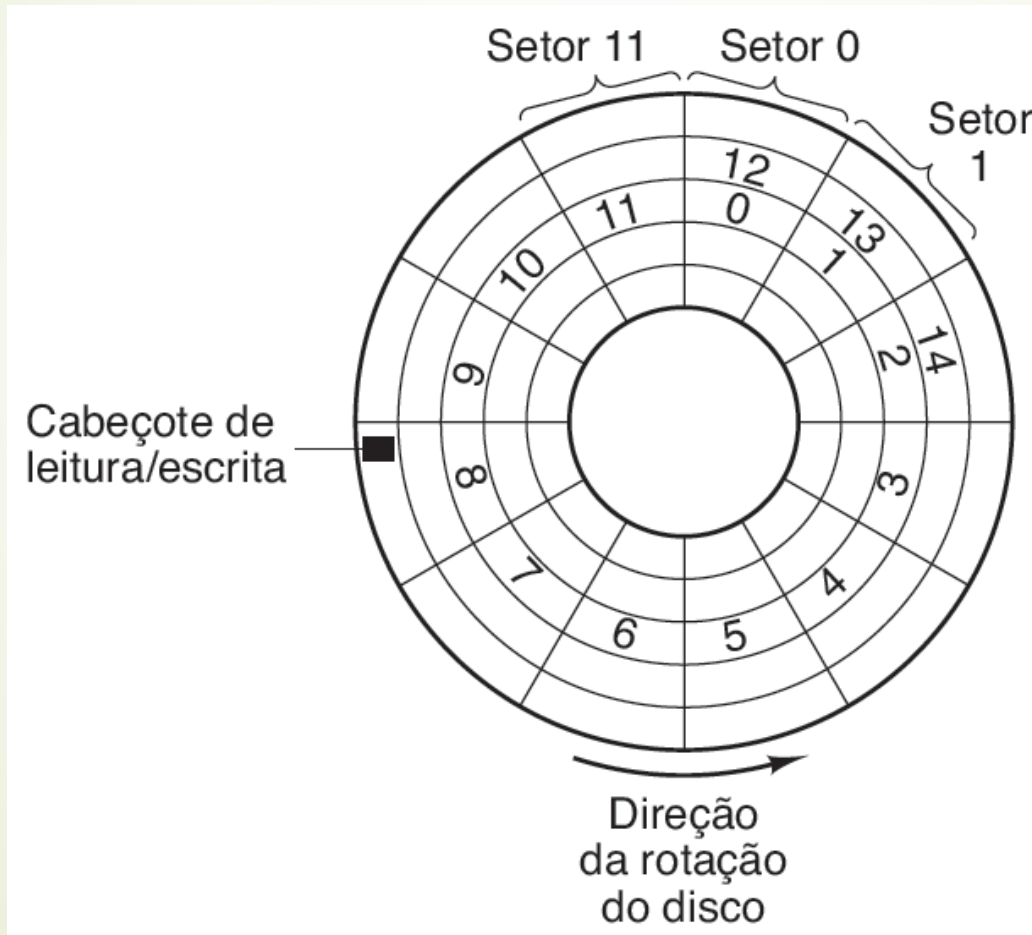
Memória nos discos

- Os sistemas monoprogramáveis permitem que o processador permaneça ocioso e que a memória seja subutilizada, enquanto um programa aguarda o término de uma operação de I/O, por exemplo.
- A multiprogramação vem resolver este problema, pois enquanto aguarda algum evento, outros processos podem ser executados pela CPU nesse intervalo de tempo.
- Nos primeiros sistemas multiprogramáveis, a memória foi dividida em pedaços de tamanho fixo, chamados partições.

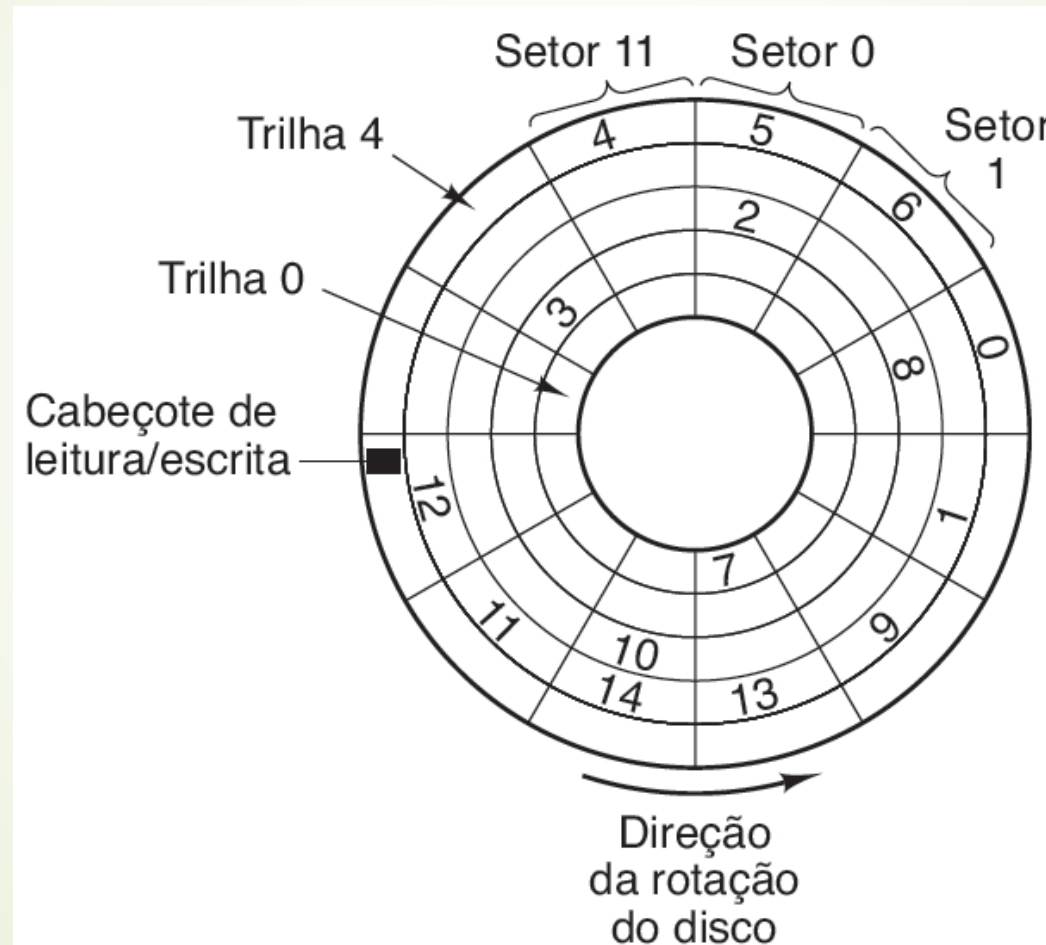
Memória nos discos

- ▶ Para entender como são realizadas instruções de E/S virtuais, é necessário examinar como os arquivos são organizados e armazenados.
- ▶ Há uma questão básica que todos os sistemas de arquivo devem encarar: a alocação de armazenamento.
- ▶ Uma questão fundamental é se um arquivo é armazenado em unidades de alocação consecutivas ou não.
- ▶ Há uma importante distinção entre a visão que um programador de aplicação tem de um arquivo e a visão que o sistema operacional tem desse arquivo.

- Estratégias de alocação de disco. Arquivo em setores consecutivos.



- Estratégias de alocação de disco. Arquivo em setores não consecutivos.



- Dois modos de monitorar setores disponíveis. Lista de livres.

Trilha	Setor	Número de setores na lacuna
0	0	5
0	6	6
1	0	10
1	11	1
2	1	1
2	3	3
2	7	5
3	0	3
3	9	3
4	3	8

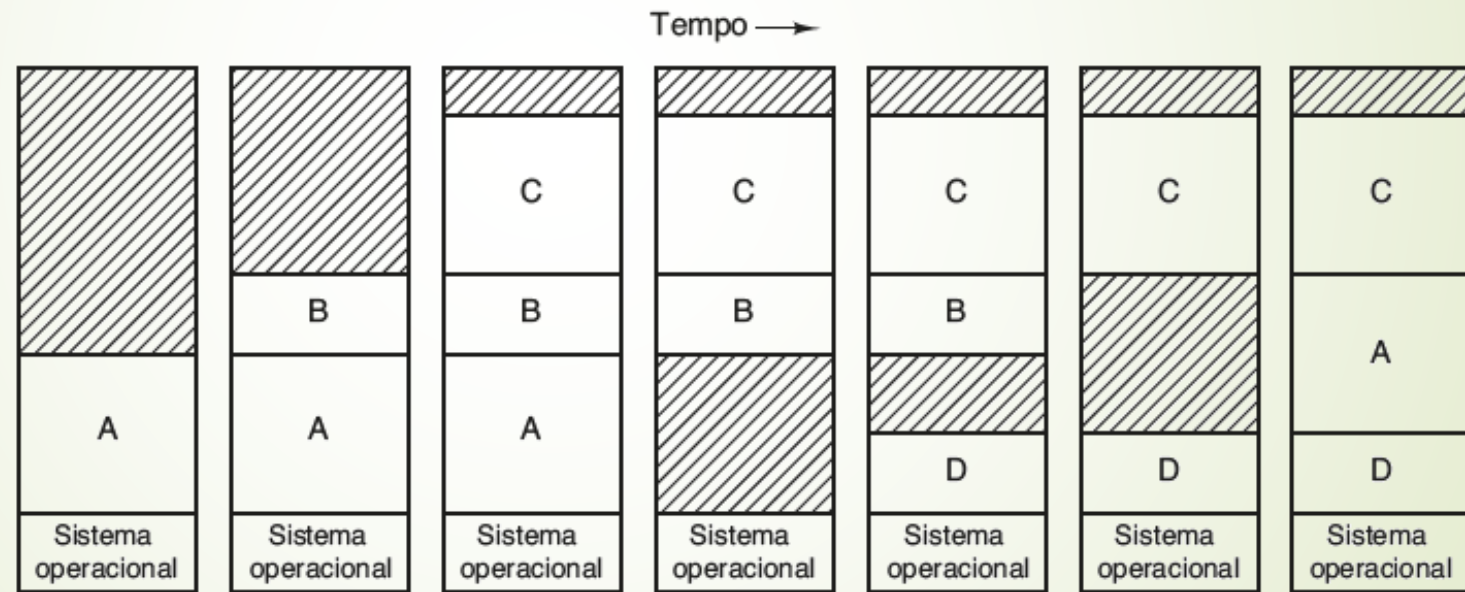
Área de Troca (Swaping)

Área de Troca (Swapping)

- Estratégia simples para lidar com a sobrecarga de memória que consiste em trazer cada processo em sua totalidade, executá-lo por um tempo e então colocá-lo de volta no disco.
- Processos ociosos estão armazenados em disco em sua maior parte, portanto não ocupam qualquer memória quando não estão sendo executados.
- Quando as trocas de processos criam múltiplos espaços na memória, é possível combiná-los em um grande espaço movendo todos os processos para baixo, o máximo possível. Essa técnica é conhecida como *compactação de memória*.

Área de Troca (Swapping)

- Mudanças na alocação de memória à medida que processos entram nela e saem dela. As regiões sombreadas são regiões não utilizadas da memória:



Sistema de Arquivos

Introdução

- Condições essenciais para armazenamento de informações por um longo prazo:
- Deve ser possível armazenar uma grande quantidade de informação.
- A informação deve sobreviver ao término do processo que esta usando a mesma.
- Múltiplos processos devem ser capaz de acessar a informação simultaneamente.

Introdução

- Todas as aplicações precisam gravar e recuperar informações.
- Alguns problemas podem ser listados:
 1. A memória é pequena demais.
 2. Manter uma informação dentro do espaço de endereçamento e quando desligada memória RAM apaga as informações.
 3. Muitos processos necessitam compartilhar as informações, muitas vezes simultaneamente.
- Para isso foram criados os sistemas de arquivos, para armazenar dados/informações de maneira persistente.

Arquivos

- Arquivos são abstrações de informação
- Muitos sistemas de arquivos não permitem nomes de arquivos acima de 255 caracteres.
- Arquivos são compostos por 2 partes:
 - Nome do arquivo
 - Extensão do arquivo

Ex: arquivo.txt

arquivo2.c

arquivo3.zip

arquivo4.jpeg

Arquivos

Exemplos de estruturas dos arquivos

Extensão	Significado
.bak	Cópia de segurança
.c	Código-fonte de programa em C
.gif	Imagem no formato <i>Graphical Interchange Format</i>
.hlp	Arquivo de ajuda
.html	Documento em HTML
.jpg	Imagem codificada segundo padrões JPEG
.mp3	Música codificada no formato MPEG (camada 3)
.mpg	Filme codificado no padrão MPEG
.o	Arquivo objeto (gerado por compilador, ainda não ligado)
.pdf	Arquivo no formato PDF (<i>Portable Document File</i>)
.ps	Arquivo PostScript
.tex	Entrada para o programa de formatação TEX
.txt	Arquivo de texto
.zip	Arquivo compactado

Tabela 4.1 Algumas extensões comuns de arquivos.

Fonte: Tanenbaum(2010)

28 Arquivos

- Existem diversos tipos de arquivos.
- A maioria deles é algum tipo de executável.
- Existem arquivos comprimidos e etc..

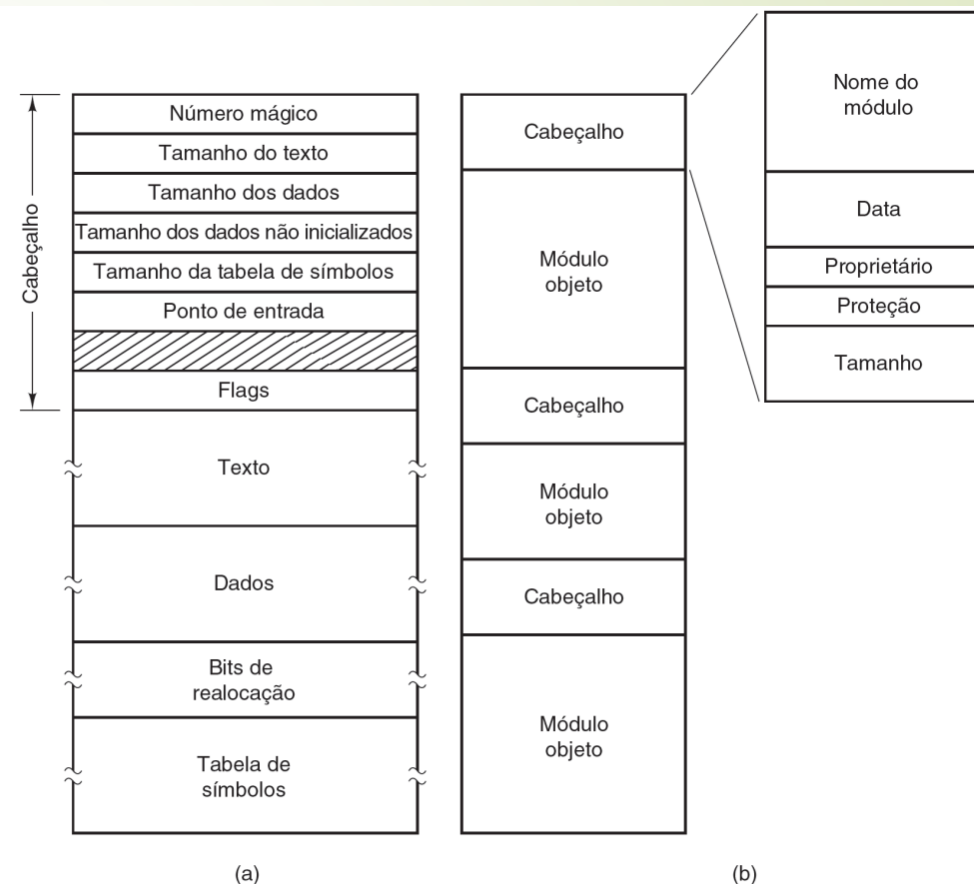


Figura 4.2 (a) Um arquivo executável. (b) Um arquivo.

29 Arquivos

➤ Os arquivos possuem atributos, o que permite realizar algumas operações tais como:

- Create (criar)
- Delete (apagar)
- Open (abrir)
- Close (fechar)
- Read (ler)
- Write (escrever)
- Append (anexar)
- Seek (procurar)
- Get Attributes (conseguir atributos)
- Set Attributes (configurar atributos)
- Rename (renomear)

Atributo	Significado
Proteção	Quem tem acesso ao arquivo e de que modo
Senha	Necessidade de senha para acesso ao arquivo
Criador	ID do criador do arquivo
Proprietário	Proprietário atual
Flag de somente leitura	0 para leitura/escrita; 1 para somente leitura
Flag de oculto	0 para normal; 1 para não exibir o arquivo
Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
Flag de arquivamento	0 para arquivos com backup; 1 para arquivos sem backup
Flag de ASCII/binário	0 para arquivos ASCII; 1 para arquivos binários
Flag de acesso aleatório	0 para acesso somente sequencial; 1 para acesso aleatório
Flag de temporário	0 para normal; 1 para apagar o arquivo ao sair do processo
Flag de travamento	0 para destravados; diferente de 0 para travados
Tamanho do registro	Número de bytes em um registro
Posição da chave	Posição da chave em cada registro
Tamanho do campo-chave	Número de bytes no campo-chave
Momento de criação	Data e hora de criação do arquivo
Momento do último acesso	Data e hora do último acesso do arquivo
Momento da última alteração	Data e hora da última modificação do arquivo
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número máximo de bytes no arquivo

■ **Tabela 4.2** Alguns atributos possíveis de arquivos.

Diretórios

- Para controlar os arquivos, o sistema de arquivos possui diretórios, ou pastas.
- Estes diretórios também são uma abstração dos dados e informações armazenadas no disco.
- Os diretórios podem ser de nível único ou hierárquico.
- Diretórios de nível único não possuem subdiretórios.
- Diretórios hierárquico possuem um ou mais subdiretórios

- Exemplo de um diretório hierárquico.

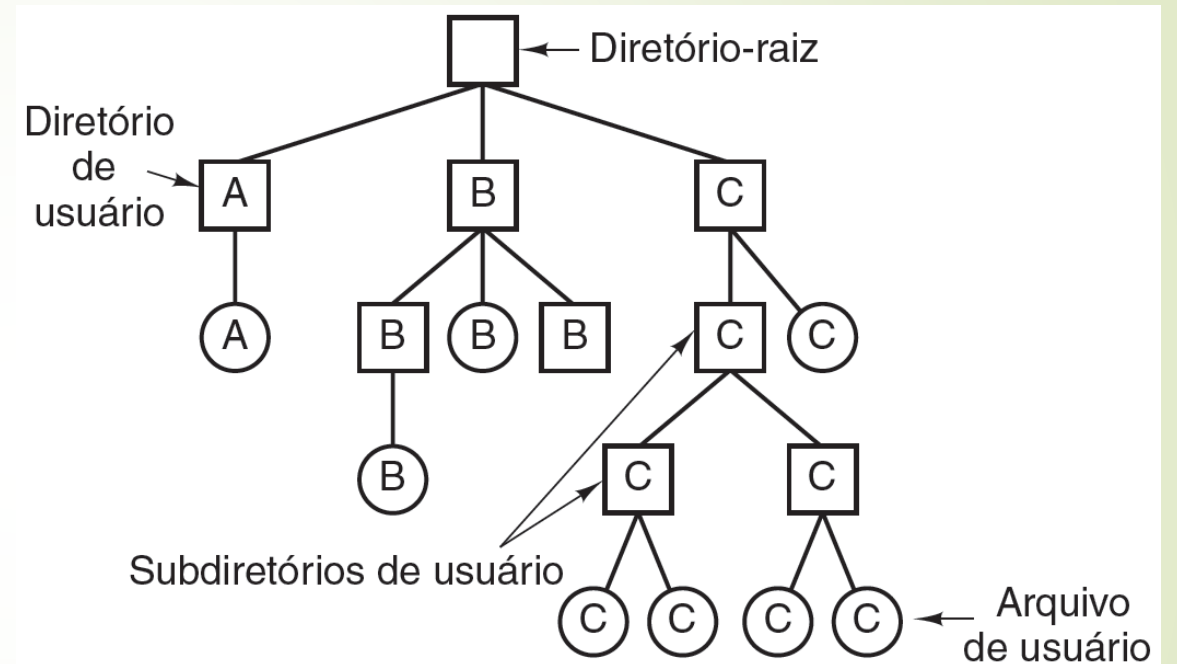


Figura 4.5 Um sistema hierárquico de diretórios.

Fonte: Tanenbaum(2010)

32 Diretórios

- ➔ Os diretórios possuem caminhos (paths) para indicar o local onde estão e também os arquivos e subpastas que ele contém.

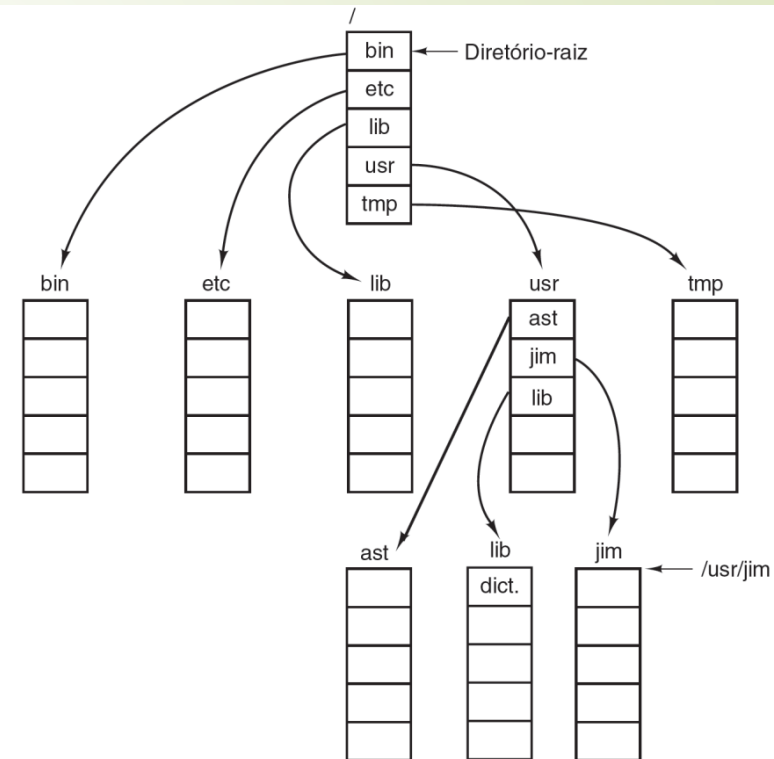


Figura 4.6 Uma árvore de diretórios UNIX.

Fonte: Tanenbaum(2010)

Diretórios

■ Algumas operações com diretórios são:

- Create (criar)
- Delete (apagar)
- Opendir (abrir diretório)
- Closedir (fechar diretório)
- Readdir (ler diretório)
- Rename (renomear)
- Link (ligar)
- Uplink

34 Esquema do sistema de arquivos

- Exemplo de divisão dos dados em disco para um Sistema de arquivos

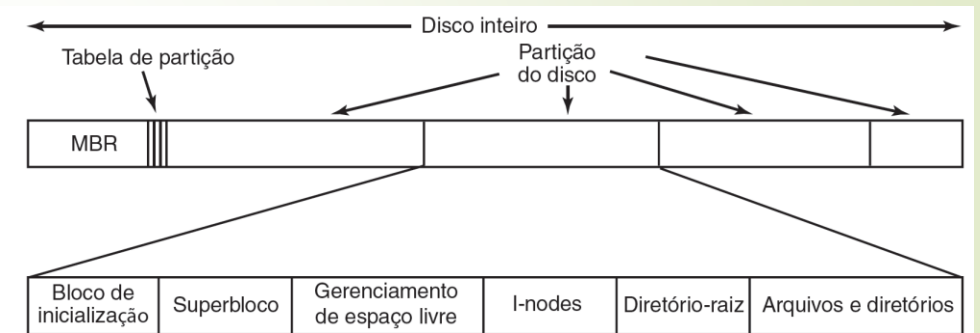


Figura 4.7 Uma organização possível para um sistema de arquivos.

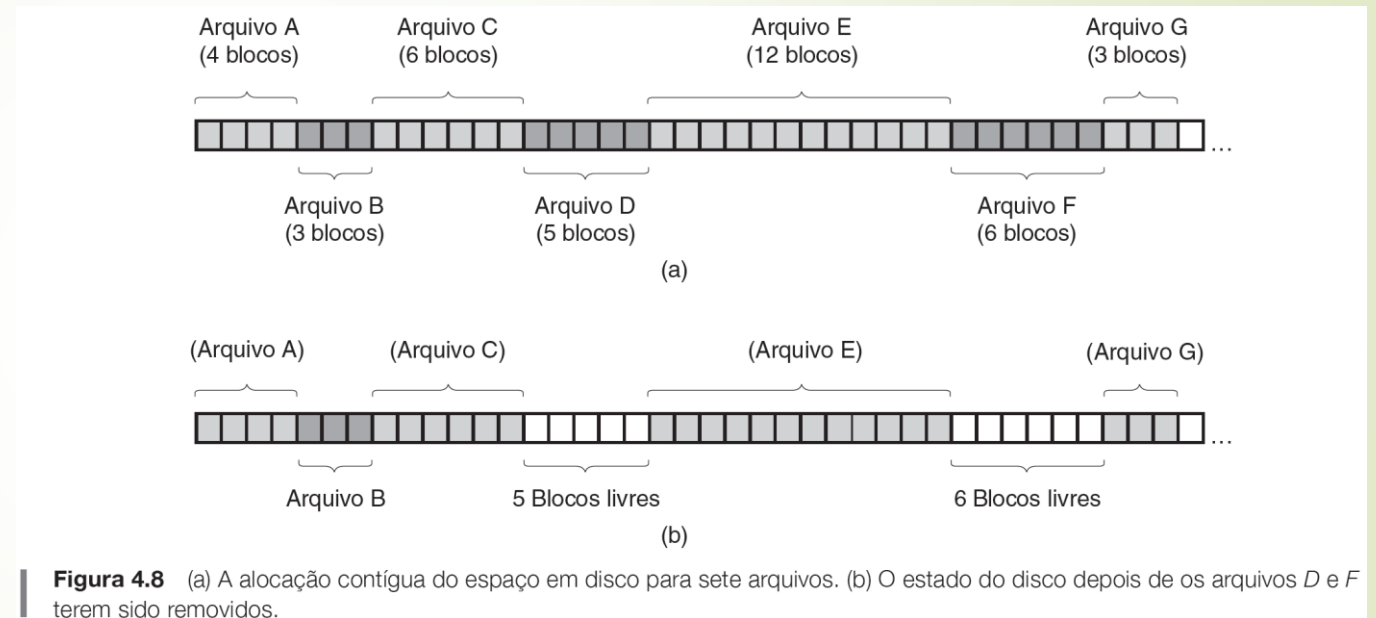
Fonte: Tanenbaum(2010)

Esquema do sistema de arquivos

- ▀ As alocações podem ser classificadas como:
 - Alocação contígua
 - Alocação encadeada
 - Alocação por I-nodes

Esquema do sistema de arquivos

Exemplo de Alocação contígua



Fonte: Tanenbaum(2010)

Esquema do sistema de arquivos

- Exemplo de alocação por lista encadeada

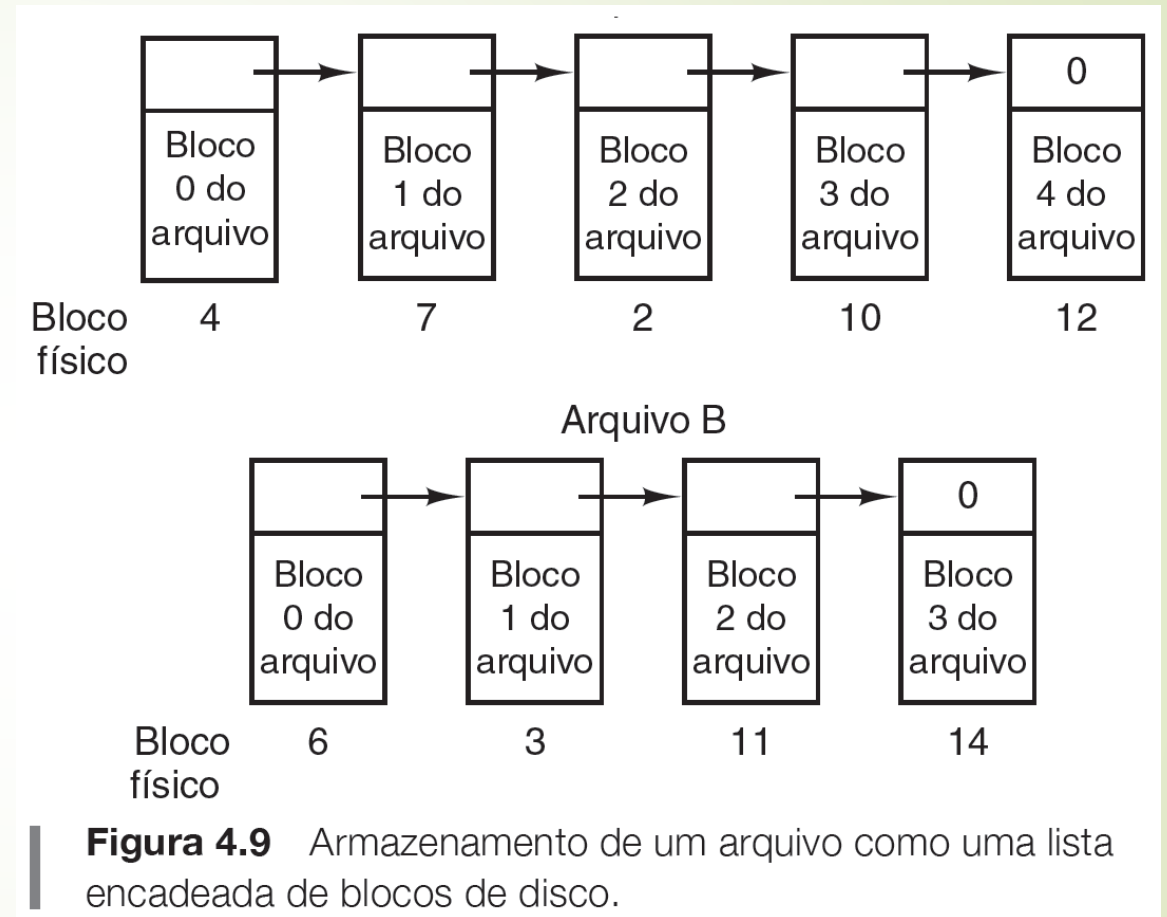
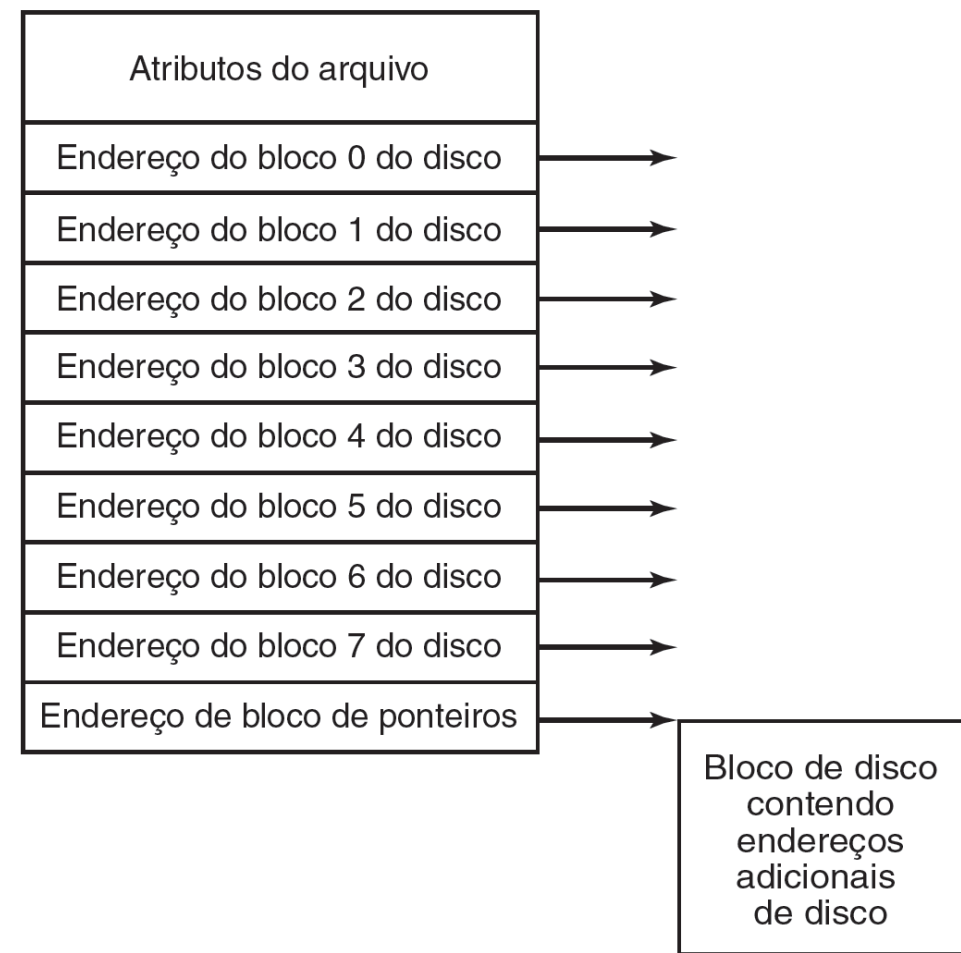


Figura 4.9 Armazenamento de um arquivo como uma lista encadeada de blocos de disco.

Fonte: Tanenbaum(2010)

38 Esquema do sistema de arquivos

➤ Exemplo de alocação por I-nodes



■ **Figura 4.11** Um exemplo de i-node.

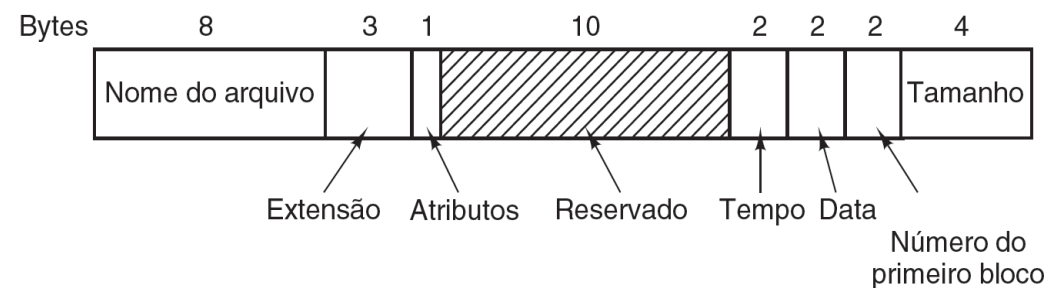
Fonte: Tanenbaum(2010)

Sistemas de arquivos Journaling

- Operações necessárias para remover um arquivo no UNIX:
- Remova o arquivo de seu diretório.
- Libere o i-node para o conjunto de i-nodes livres.
- Volte todos os blocos do disco para o conjunto de blocos livres no disco.

40 Exemplos de sistemas de arquivos

- Exemplo de Sistema de arquivos MS-DOS



■ **Figura 4.28** A entrada de diretório do MS-DOS.

Fonte: Tanenbaum(2010)

41 Exemplos de sistemas de arquivos

- Alguns exemplos de Sistemas de arquivos.
- Existem também os sistemas EXT que são utilizados pelos sistemas operacionais Unix.

Tamanho do bloco	FAT-12	FAT-16	FAT-32
0,5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Tabela 4.4 Tamanho máximo da partição para diferentes tamanhos de bloco. As células em branco representam combinações não permitidas.

Fonte: Tanenbaum(2010)

Memória virtual

Memória Virtual

- A memória virtual foi usada pela primeira vez em alguns computadores durante a década de 1960, a maioria deles associada com projetos de pesquisa na área de sistemas de computação.
- No início da década de 1970, a memória virtual já estava disponível na maioria dos computadores.
- Agora, até computadores de um só chip, incluindo o Core i7 e a CPU ARM do OMAP4430, têm sistemas de memória virtual altamente sofisticados.

Memória virtual

- Apesar de os tamanhos das memórias aumentarem depressa, os tamanhos dos softwares estão crescendo muito mais rapidamente.
- Como consequência desses desenvolvimentos, há uma necessidade de executar programas que são grandes demais para se encaixar na memória e há certamente uma necessidade de ter sistemas que possam dar suporte a múltiplos programas executando em simultâneo, cada um deles encaixando-se na memória, mas com todos coletivamente excedendo-a.

Memória virtual

- Método encontrado para passar todo o programa para o computador.
- A ideia básica é que cada programa tem seu próprio espaço de endereçamento, o qual é dividido em blocos chamados de **páginas**.
- De certa maneira, é uma generalização da ideia do registrador base e registrador limite.
- Funciona bem em um sistema de multiprogramação, com pedaços e partes de muitos programas na memória simultaneamente. Enquanto um programa está esperando que partes de si mesmo sejam lidas, a CPU pode ser dada para outro processo.

Paginação

- Técnica usada pela maioria dos sistemas de memória virtual.
- O espaço de endereçamento virtual consiste em unidades de tamanho fixo chamadas de páginas.
- As unidades correspondentes na memória física são chamadas de quadros de página.
- Transferências entre a memória RAM e o disco são sempre em páginas inteiras.
- Muitos processadores dão suporte a múltiplos tamanhos de páginas que podem ser combinados e casados como o sistema operacional preferir.

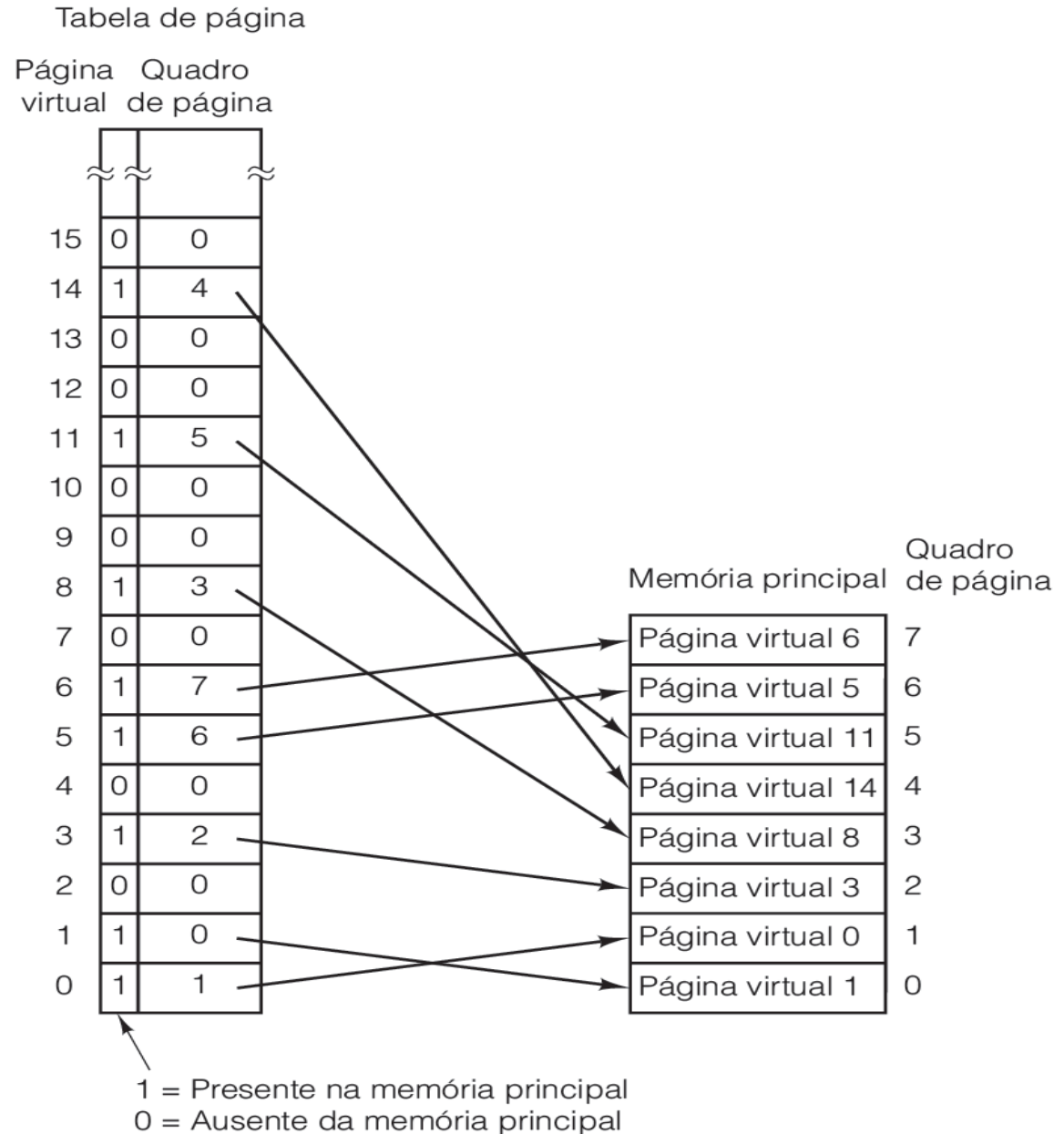
Paginação

- Memória principal de 32 KB dividida em oito quadros de página de 4 KB cada.

32 KB da parte inferior da memória principal	
Quadro de página	Endereços físicos
7	28672 – 32767
6	24576 – 28671
5	20480 – 24575
4	16384 – 20479
3	12288 – 16383
2	8192 – 12287
1	4096 – 8191
0	0 – 4095

Paginação

- Possível mapeamento das 16 primeiras páginas virtuais para uma memória principal com oito quadros de página.



Paginação

- No hardware real, um **bit Presente/ausente** controla quais páginas estão fisicamente presentes na memória.
- A interrupção é chamada de **falta de página** (*page fault*). O sistema operacional escolhe um quadro de página pouco usado e escreve seu conteúdo de volta para o disco (se já não estiver ali). Ele então carrega (também do disco) a página recém-referenciada no quadro de página recém-liberado, muda o mapa e reinicia a instrução que causou a interrupção.
- O número da página é usado como um índice para a **tabela de páginas**, resultando no número do quadro de página correspondente àquela página virtual.

Tabela de páginas

- O objetivo da tabela de páginas é mapear as páginas virtuais em quadros de páginas.
- Matematicamente falando, é uma função com o número da página virtual como argumento e o número do quadro físico como resultado. Usando o resultado dessa função, o campo da página virtual em um endereço virtual pode ser substituído por um campo de quadro de página, desse modo formando um endereço de memória física.

Segmentação

- ▶ Ajuda a lidar com estruturas de dados que podem mudar de tamanho durante a execução e simplifica a ligação e o compartilhamento.
- ▶ Facilita proporcionar proteção para diferentes segmentos.
- ▶ Às vezes a segmentação e a paginação são combinadas para proporcionar uma memória virtual bidimensional.

Segmentação x Paginação

Consideração	Paginação	Segmentação
O programador precisa saber que essa técnica está sendo usada?	Não	Sim
Há quantos espaços de endereçamento linear?	1	Muitos
O espaço de endereçamento total pode superar o tamanho da memória física?	Sim	Sim
Rotinas e dados podem ser distinguidos e protegidos separadamente?	Não	Sim
As tabelas cujo tamanho flutua podem ser facilmente acomodadas?	Não	Sim
O compartilhamento de rotinas entre os usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

Segmentação com paginação

- Se os segmentos forem grandes, talvez não seja possível mantê-los na memória principal em sua totalidade. Isso leva à ideia de realizar a paginação dos segmentos, de maneira que apenas aquelas páginas de um segmento que são realmente necessárias tenham de estar na memória.
- O **sistema MULTICS** e o **x86** de 32 bits da Intel dão suporte à segmentação e à paginação. Ainda assim, fica claro que poucos projetistas de sistemas operacionais se preocupam mesmo com a segmentação (pois são casados a um modelo de memória diferente). Em consequência, ela parece estar saindo rápido de moda. Hoje, mesmo a versão de 64 bits do x86 não dá mais suporte a uma segmentação de verdade.

Tipos de instrução

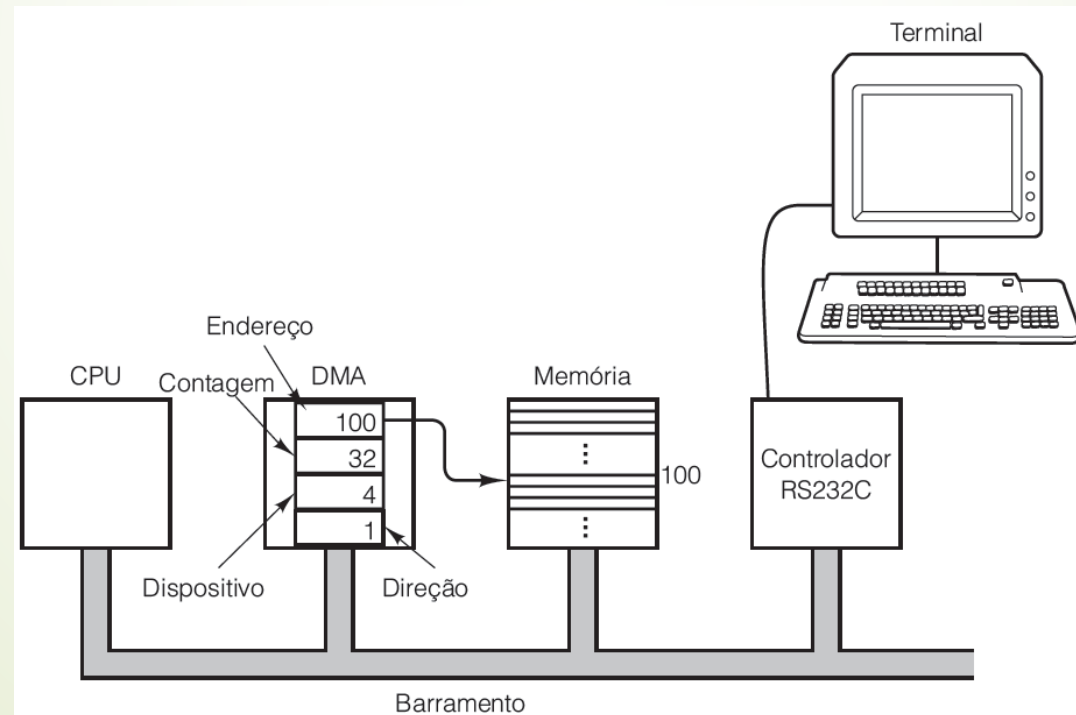
Entrada/Saída

- Embora a E/S por interrupção seja um grande passo à frente em comparação com a E/S programada, está longe de ser perfeita.
- O problema é que é requerida uma interrupção para todo caractere transmitido.
- Como processar uma interrupção é caro, precisamos de um meio de nos livrar da maioria das interrupções.

Tipos de instrução

Entrada/Saída

- A solução está em acrescentar ao sistema um novo chip, um controlador **DMA**, com acesso direto ao barramento.



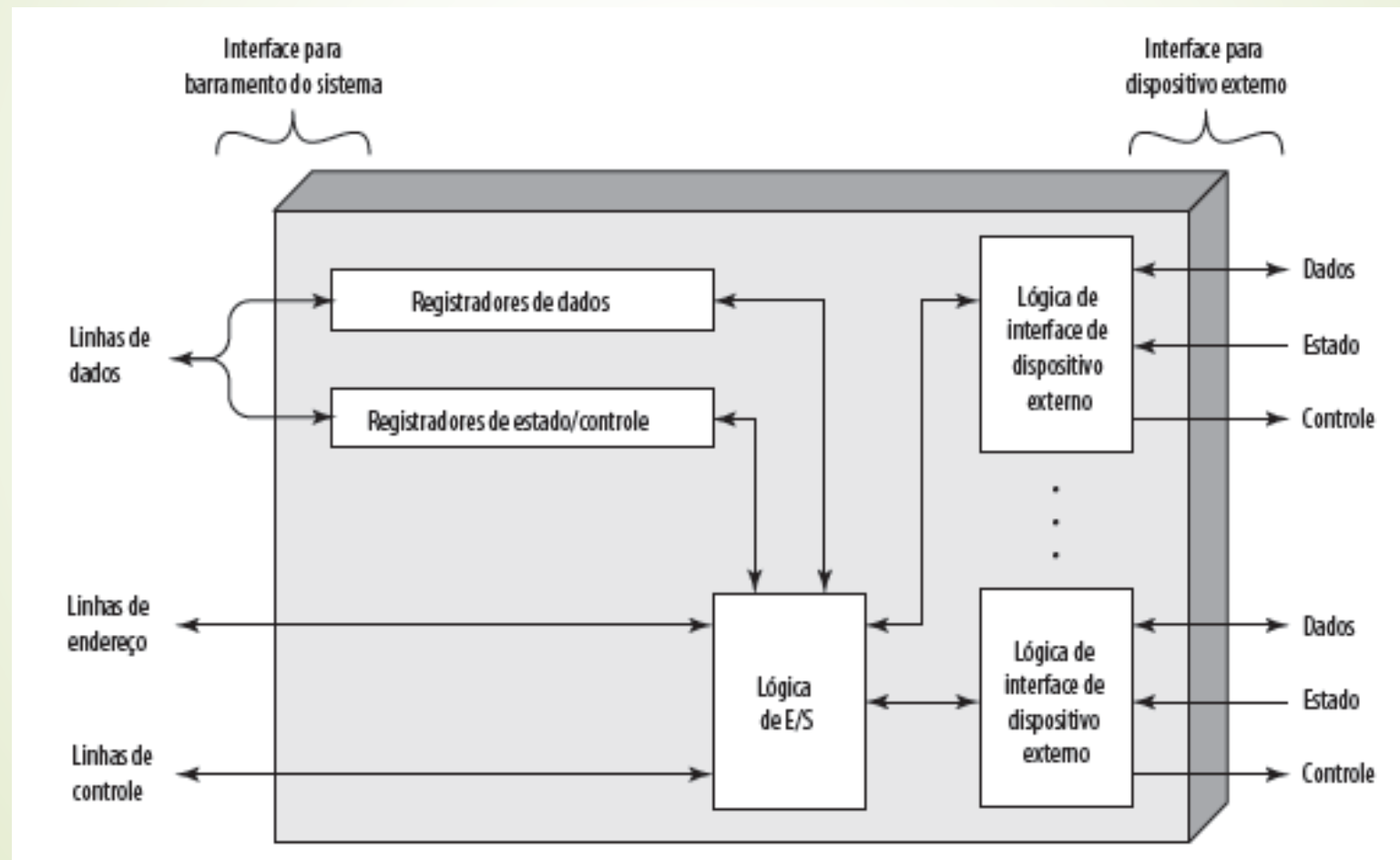
Dispositivos externos

- Legíveis ao ser humano:
 - Monitor, impressora, teclado.
- Legíveis à máquina:
 - Monitoração e controle.
- Comunicação:
 - Modem.
 - Placa de interface de rede (NIC).

Etapas da E/S

- CPU verifica estado do dispositivo do módulo de E/S.
- Módulo de E/S retorna o estado.
- Se estiver pronto, CPU solicita transferência de dados.
- Módulo de E/S recebe dados do dispositivo.
- Módulo de E/S transfere dados à CPU.
- Variações para saída, DMA etc.

Diagrama do módulo de E/S



Decisões do módulo de E/S

- Ocultar ou revelar propriedades do dispositivo à CPU.
- Admitir dispositivo múltiplo ou único.
- Controlar funções do dispositivo ou sair para CPU.
- Também decisões do SO.
 - P.e., Unix trata de tudo o que pode como arquivo.

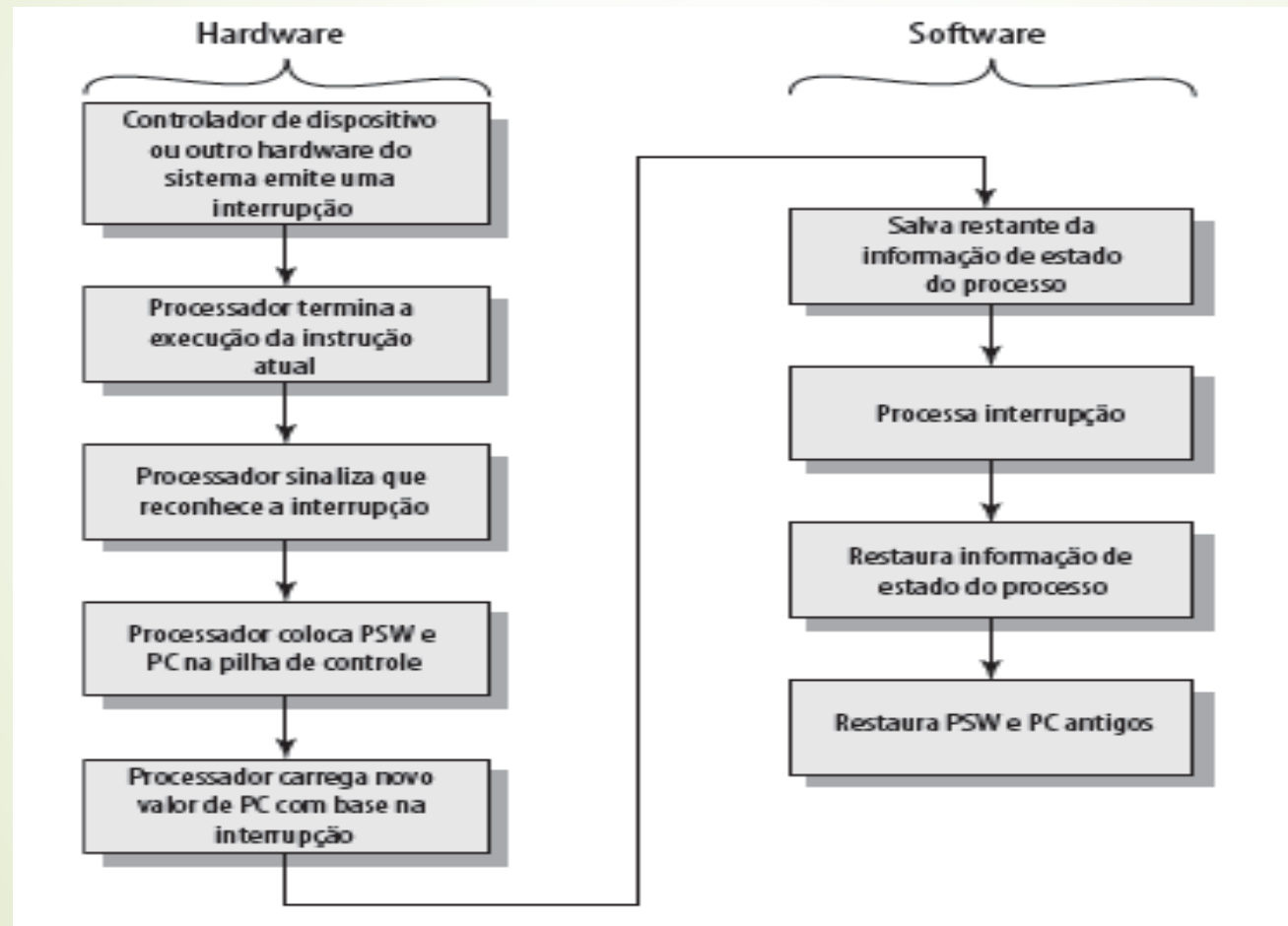
Mapeamento de E/S

- E/S mapeada na memória:
 - Dispositivos e memória compartilham um espaço de endereços comum.
 - E/S se parece com leitura/escrita na memória.
 - Nenhum comando especial para E/S.
 - Grande seleção disponível de comandos de acesso à memória.
- E/S independente:
 - Espaços de endereços separados.
 - Precisa de linhas de seleção de E/S ou memória.
 - Comandos especiais para E/S.
 - Conjunto limitado.

E/S controlada por interrupção

- Contorna problema de espera da CPU.
- Sem verificação de dispositivo repetida da CPU.
- Módulo de E/S interrompe quando estiver pronto.

Processamento de interrupção simples



Fluxo de controle

- Fluxo de controle se refere à sequência em que as instruções são executadas dinamicamente, isto é, durante a execução do programa.
- A maioria das instruções não altera o fluxo de controle.
- A ordem dinâmica na qual o processador de fato executa as instruções é a mesma em que elas aparecem na listagem do programa.
- Quando há desvios presentes, o contador de programa deixa de ser uma função monotônica crescente do tempo.

Fluxo de controle

- **Interrupções** são mudanças no fluxo de controle que não são causadas pelo programa em execução, mas por alguma outra coisa, em geral relacionada à E/S.
- A diferença essencial entre exceções e interrupções é a seguinte:
- exceções são síncronas com o programa e *interrupções* são assíncronas.
- De forma simplificada, as etapas são as seguintes:

Fluxo de controle

- **AÇÕES DO HARDWARE**

1. O controlador de dispositivo ativa uma linha de interrupção no barramento de sistema para iniciar a sequência de interrupção.
2. A CPU ativa um sinal de reconhecimento de interrupção no barramento.
3. Quando o controlador de dispositivo vê que seu sinal de interrupção foi reconhecido, coloca um inteiro pequeno nas linhas de dados para se identificar. Esse número é denominado **vetor de interrupção**.

Fluxo de controle

- **AÇÕES DO HARDWARE**

4. A CPU retira o vetor de interrupção do barramento e o salva temporariamente.
5. Então, a CPU passa o contador de programa e a PSW para a pilha.
6. Em seguida, a CPU localiza um novo contador de programa usando o vetor de interrupção como um índice para uma tabela na parte inferior da memória.

Referências

- TANENBAUM, A. S. **sistemas operacionais modernos** 4ª ed. Editora Pearson 2016.
- TANENBAUM, A. S; Austin, T. **organização estruturada de computadores** 6ª ed. editora: Pearson 2014.