BC1518 - Sistemas Operacionais

Aula 2: Estruturas de Sistemas de Computação

Tópicos desta aula

- Revisão de hardware
  - Processador
  - Memória
  - Discos
  - Dispositivos de E/S
- Interrupção
- Proteção de *Hardware*
  - Operação modo dual
  - Proteção de E/S
  - Proteção de memória

Um Sistema de Computação

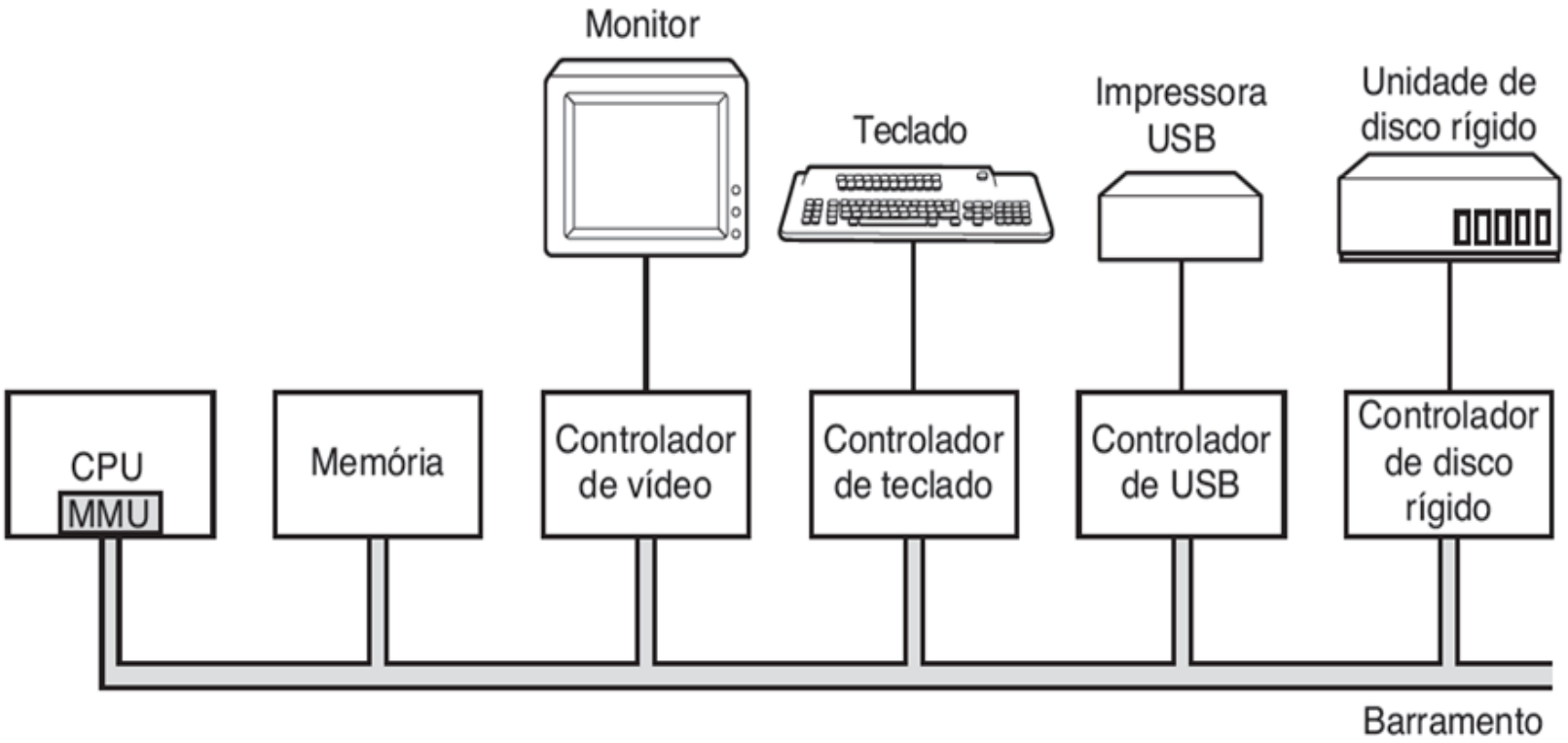


Figura 1.6 Alguns dos componentes de um computador pessoal simples.

Um Sistema de Computação de uso geral consiste em uma ou mais CPUs e em uma série de controladores de dispositivos, conectados através de um barramento comum

Processador

- O processador (CPU) é o componente vital de um sistema de computação, é o "cérebro" do computador
- É responsável pela realização de operações de processamento e controle durante a execução de um programa
- Para que o processador execute um programa, este deve ser traduzido em instruções (em linguagem de máquina)
- Essas instruções devem estar armazenadas em posições sucessivas da memória principal
- A execução de um programa é feita sequencialmente, uma instrução de cada vez
- Todo processador suporta um conjunto específico de instruções de máquina
- Esse conjunto de instruções é especificado pela arquitetura do processador
  - Por exemplo, um processador Pentium não executa programas voltados para um processador SPARC e vice-versa

Instrução de máquina

- O conjunto de instruções contém as operações que o processador é capaz de executar:
  - movimentação de dados (memória <--> registrador)
  - matemáticas (aritméticas, lógicas, ...)
  - entrada-saída (leitura e escrita em dispositivos de E/S)
  - controle (desvio da sequência de execução, parada, etc...)

Alguns exemplos de instruções* (processador hipotético):

0000	Load	Carregar no acumulador
0001	Store	Salvar na memória
0010	Add	Somar
0011	Sub	Subtrair

Exemplo:

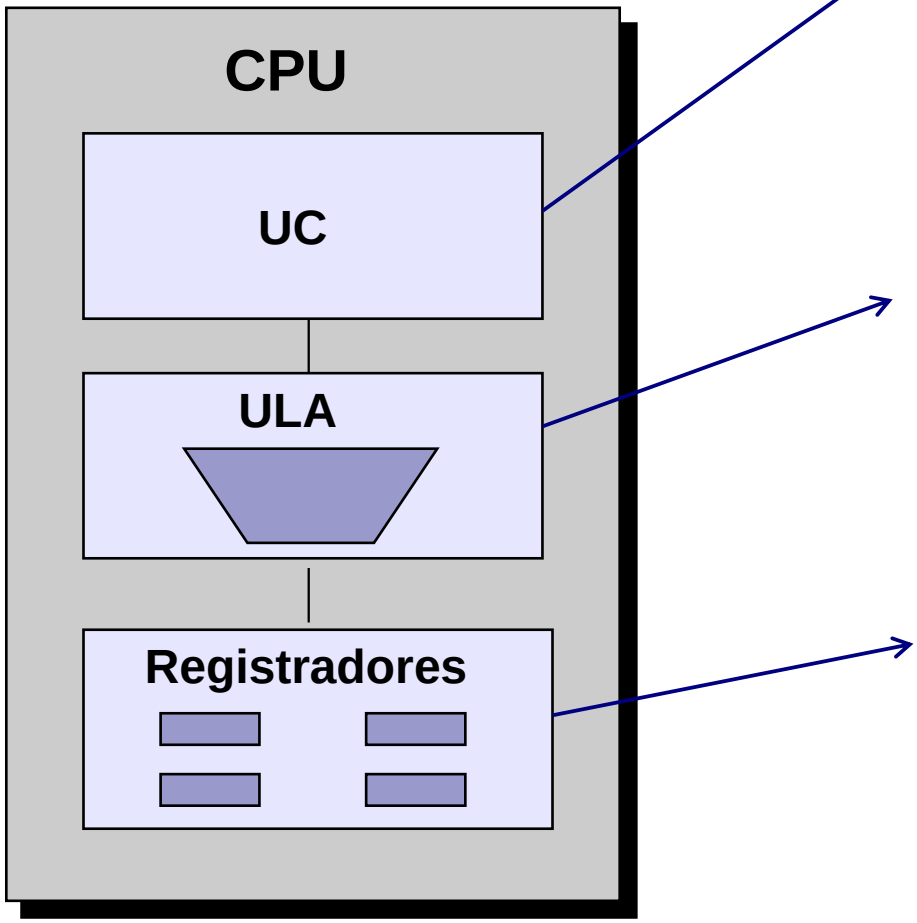
Add A B C (Soma o conteúdo de A e B e armazena o resultado em C)

A mesma instrução em linguagem de máquina:

0010 11110000 11110001 1111011

Endereço (local onde está armazenado o dado)

Componentes da CPU



Unidade de Controle (UC)

- Controla a operação da CPU
- Busca e decodifica instruções

Unidade Lógica e Aritmética (ULA)

- Processamento dos dados
- Executa operações lógicas e aritméti

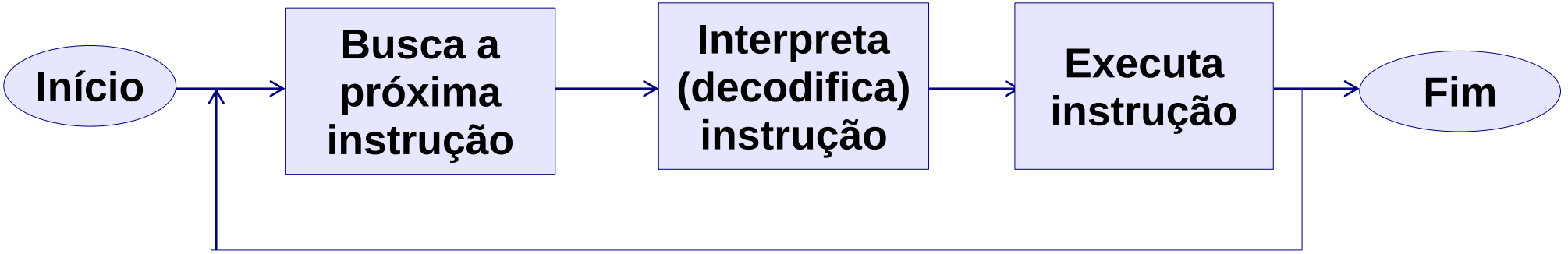
Registradores

- Armazenamento interno de dados pa

Registradores

- Os registradores são utilizados para o armazenamento temporário de informações durante a execução de um programa
  - Armazena dados que estão sendo manipulados e os resultados temporário das operações
  - São utilizados pois o tempo de acesso para buscar instruções (ou operandos) na memória é muito maior do que o tempo para processá-las
- Alguns registradores são de propósito específico:
  - Contador de Programa (PC – *Program Counter*): armazena o endereço (na memória) da próxima instrução a ser buscada
  - Registrador de Instrução (RI): armazena a instrução a ser executada
  - Ponteiro de pilha: aponta para o topo da pilha na memória
  - PSW (*Program Status Word*): contém informações de controle (por exemplo, modo de funcionamento da CPU – núcleo ou

Ciclo básico de execução da CPU



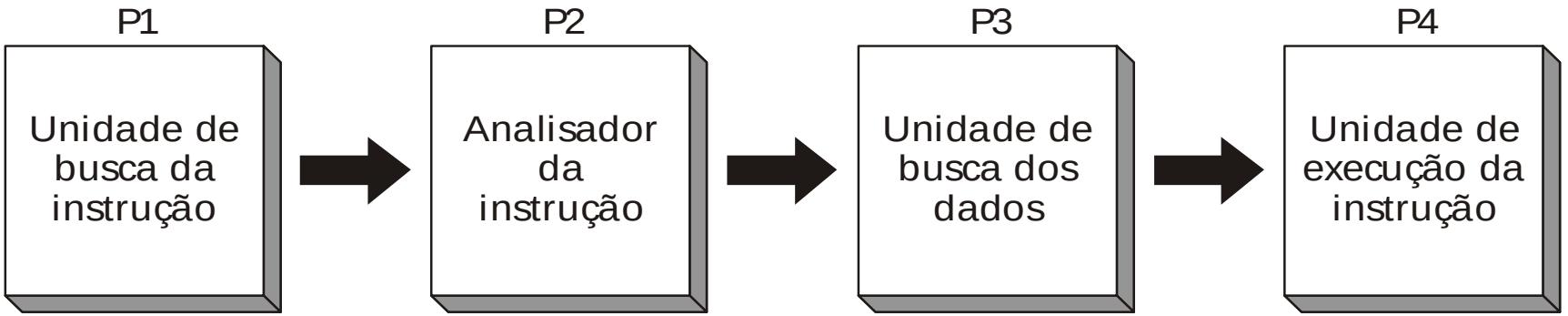
Pipeline

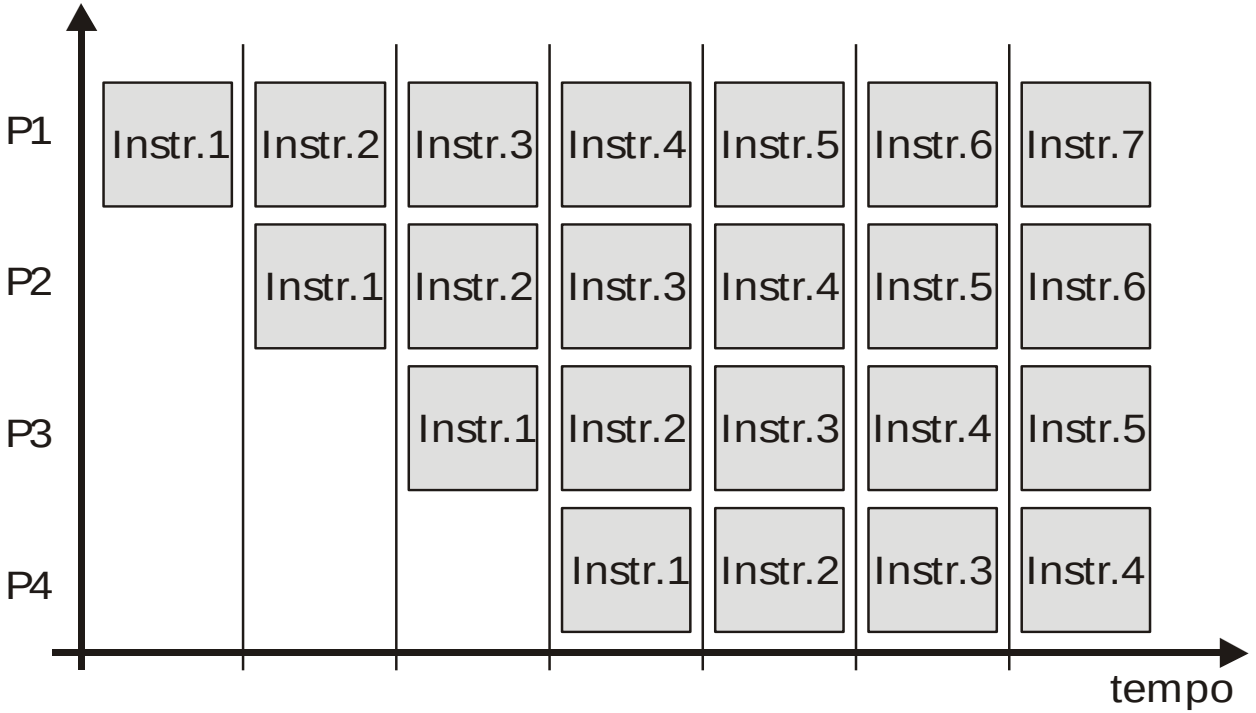
➢Para melhorar o desempenho, em vez do ciclo básico (busca, decodificação e execução), foram introduzidas técnicas como o *Pipeline*

- A principal ideia é dividir o ciclo básico de execução em estágios (busca, decodificação e execução) de modo que possam ser executados paralelamente
- Através da utilização de múltiplas unidades funcionais na CPU

- Exemplo: uma CPU pode ter unidades separadas de busca, decodificação e execução, de modo que possa executar os três estágios de maneira independente e paralelamente

Pipeline





Arquitetura pipeline com 4 estágios [Machado]

Processadores pipeline e superescalar

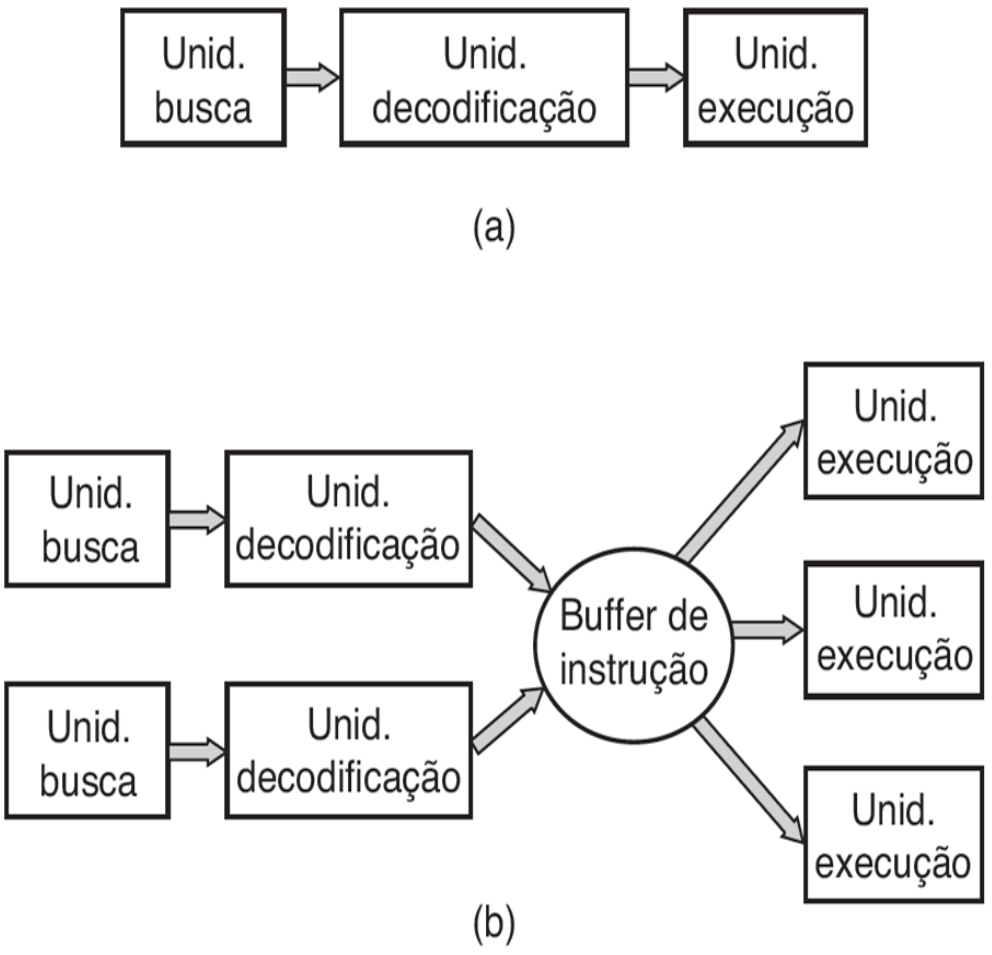


Figura 1.7 (a) Um processador com pipeline de três estágios. (b) Uma CPU superescalar. [Tanenbaum]

- Processador superescalar:
  - Possui vários estágios de pipeline
  - Pode ter múltiplas unidades de busca, decodificação
  - E também múltiplas unidades de execução (uma unidade aritmética para inteiros, outra para ponto flutuante e outra para operações lógicas
  - Um *buffer* de instrução para armazenamento temporário das instruções decodificadas e prontas para execução
  - Quando uma unidade de execução terminar, é buscada uma instrução que possa executar no *buffer* (e remove

UFABC

Armazenamento

Hierarquia de Armazenamento

- Sistemas de armazenamento são organizados em hierarquia
  - **Velocidade** (geralmente, quanto mais rápida, mais cara a memória)
  - **Custo**
  - **Volatilidade**

- Quanto mais alto estiver na hierarquia, maior a velocidade e custo e menor a capacidade

Hierarquia de Dispositivos de Armazenamento

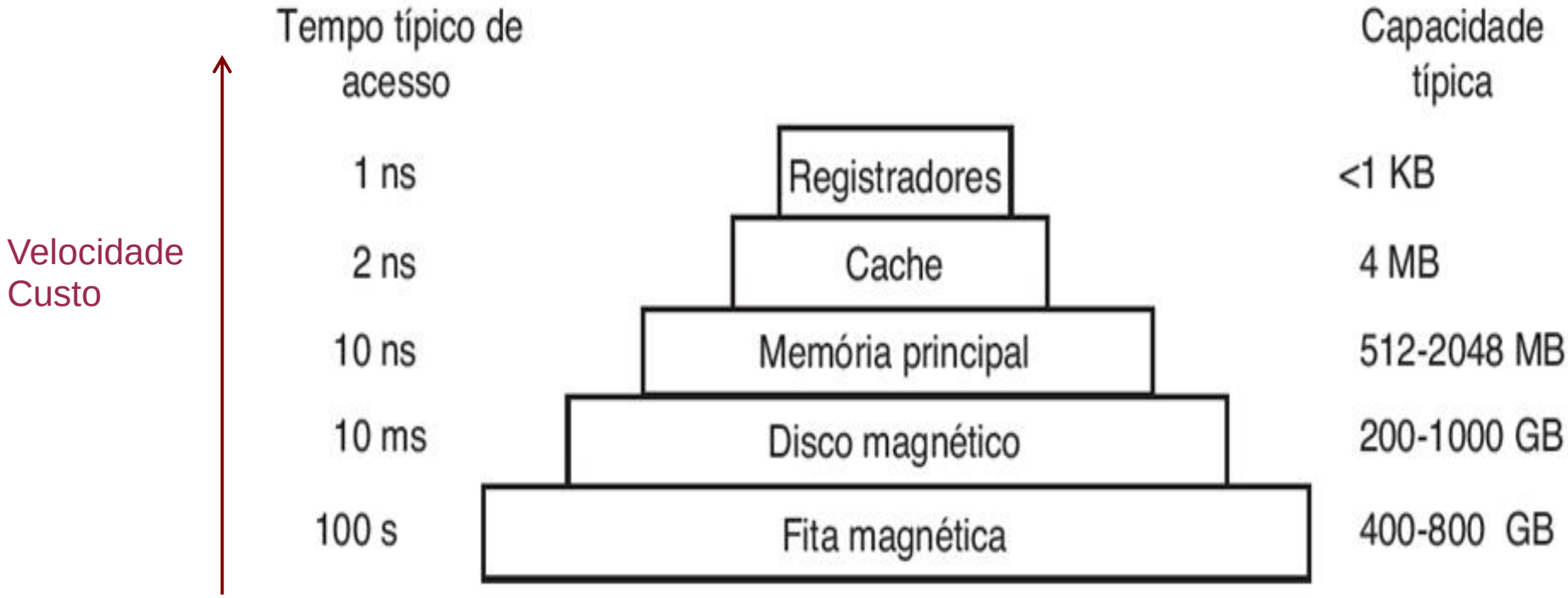


Figura 1.9 Hierarquia de memória típica. Os números são aproximações. [Tanenbaum]

Registradores, cache e memória principal são voláteis, enquanto que discos e fitas são nã

Memória cache

- Memória de alta velocidade, localizada dentro ou bem próxima da CPU
- Possui tamanho limitado, devido ao seu alto custo
- A principal vantagem no uso de cache é possibilitar a redução no tempo de acesso a um dado (obtenção do dado) pela CPU
  - Dados frequentemente usados são armazenados na memória cache
  - Quando a CPU precisa ler um dado, antes de buscar na memória principal, verifica se o dado está na cache
  - Caso esteja presente na cache (*cache hit*), não há a necessidade de acessar a memória principal (através do barramento)
  - Porém se o dado não estiver na cache (*cache miss*) é feita a requisição na memória principal

Memória cache

- Muitas máquinas possuem 2 até 3 níveis de cache
  - Cache L1: está integrada à CPU, em geral possui capacidade de dezenas de kilobytes
  - Cache L2: capaz de armazenar centenas de megabytes a dezenas ou centenas de gigabytes
  - Cache L3: menos comum, maior capacidade de armazenamento
  - L1 é mais rápida que L2, que é mais rápida que L3 e que por sua vez é mais rápida que a memória principal

- Em muitos processadores atuais L1 e L2 são incorporados à CPU, utilizando conexões de alta velocidade da mesma

Memória cache em processadores multicore

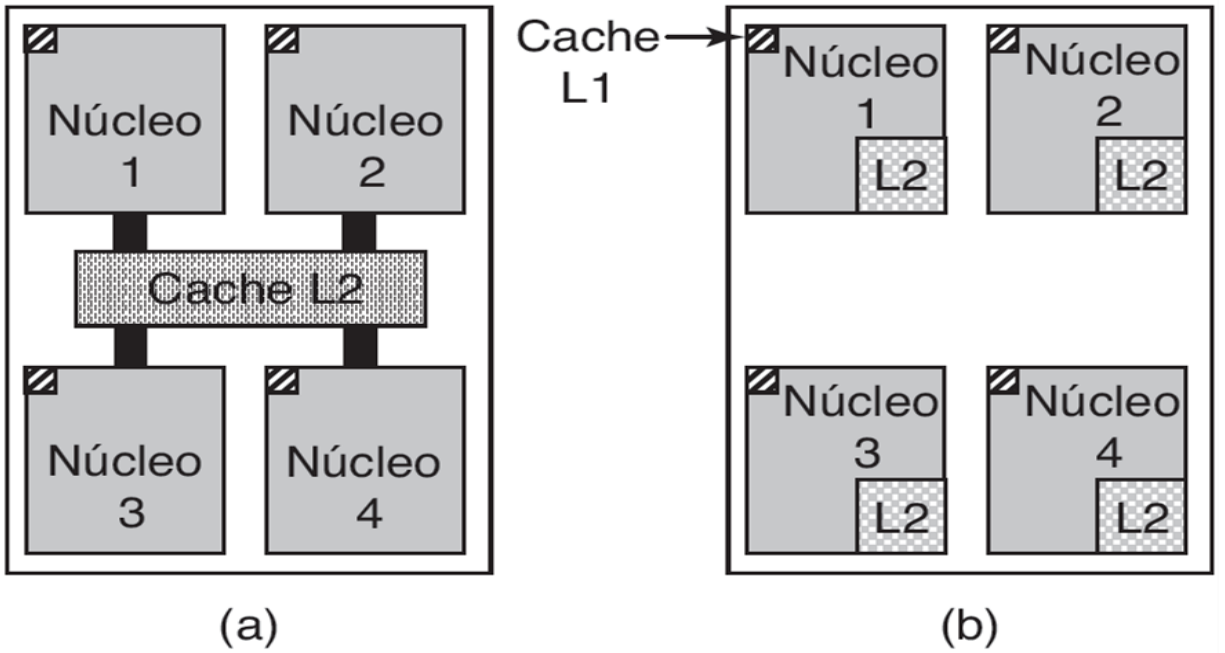


Figura 1.8 (a) Chip quad-core com uma cache L2 compartilhada. (b) Um chip quad-core com caches L2 separadas. [Tanenbaum]

- Em (a) cache L2 compartilhada pelos núcleos (chips da Intel)
  - Requer um mecanismo para controle de concorrência no acesso à cache
- Em (b) Cada núcleo tem sua própria cache L2 (chips da AMD)
  - É necessário manter a consistência das caches

Caching

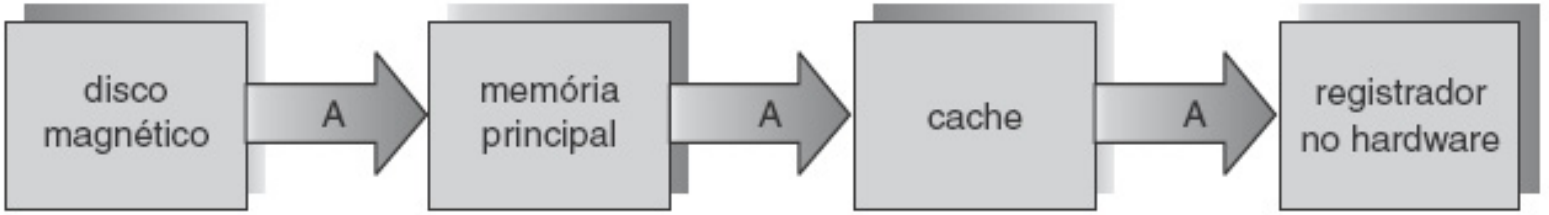
➢Uso da memória de alta velocidade para armazenar **dados recentemente acessados** a fim de melhorar o desempenho

➢Requer uma política de ***gerenciamento de cache.*** Questões ao lidar com cache:

- Quando colocar um novo item em um novo cache
- Em qual linha de cache colocar um novo item
- Que item remover do cache quando for preciso espaço
- Onde colocar um item desalojado recentemente na memória mais ampla
- Escolha do tamanho do cache e a forma de substituição dos itens do cache pode causar um grande aumento no desempenho

➢O *caching* introduz outro nível na hierarquia de armazenamento

- Isso exige que os **dados** armazenados simultaneamente em mais de um nível sejam ***consistentes***



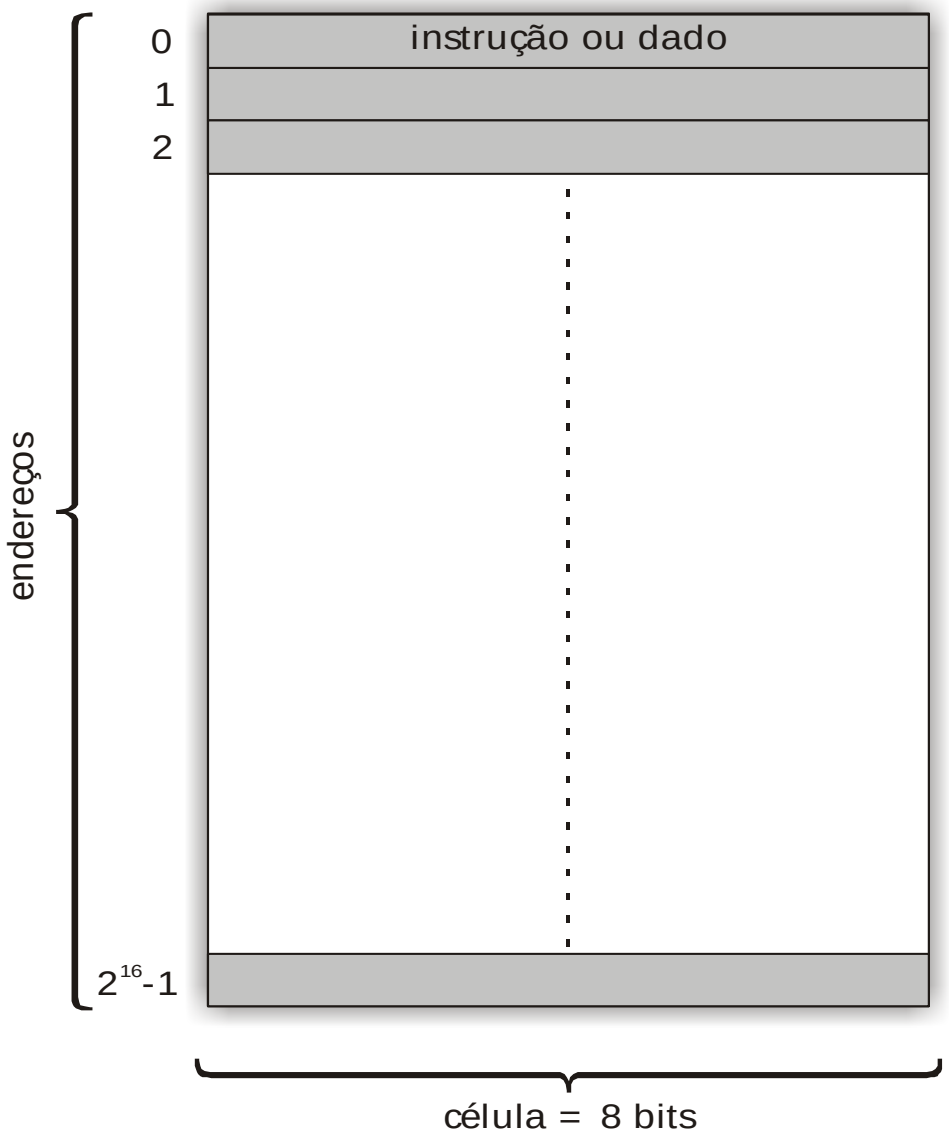
Migração do Inteiro A do Disco para o Registrador [Silberschatz]

- Em uma estrutura de armazenamento hierárquica, um mesmo dado pode estar em diferentes níveis
- Exemplo: Incrementar em 1 o valor de um inteiro A que está no arquivo B e este arquivo B está armazenado no disco
  - O bloco correspondente ao inteiro A é copiado do disco para a memória principal, em seguida, o mesmo é copiado da memória principal para a cache e finalmente copiada da cache para o registrador
  - Se o resultado da operação de incremento de A for armazenado somente no registrador, os valores de A ficarão diferentes nos dispositivos
  - É preciso garantir que o valor de A acessado (por outros processos)

Memória principal

- A memória principal ou memória de acesso aleatório (RAM – *Random Access Memory*) é uma memória essencial em um computador, onde as instruções e dados são armazenados
- Características:
  - Volatilidade: o seu conteúdo é perdido ao desligar o computador ou em uma interrupção no fornecimento de energia
  - Acesso aleatório aos dados: em vez do acesso sequencial como em fitas magnéticas => o tempo de acesso a qualquer endereço de memória é constante
  - Suporte a leitura e gravação de dados
- A memória armazena apenas bits (0 ou 1)
  - O bit é a unidade básica de memória
- A memória é estruturada em sequências de bits, chamadas de células (em geral, as células possuem 8 bits (1 byte))
- O tamanho da célula depende, dentre outros fatores, do propósito de utilização do computador

Memória principal



Memória principal (64 KB) [Machado]

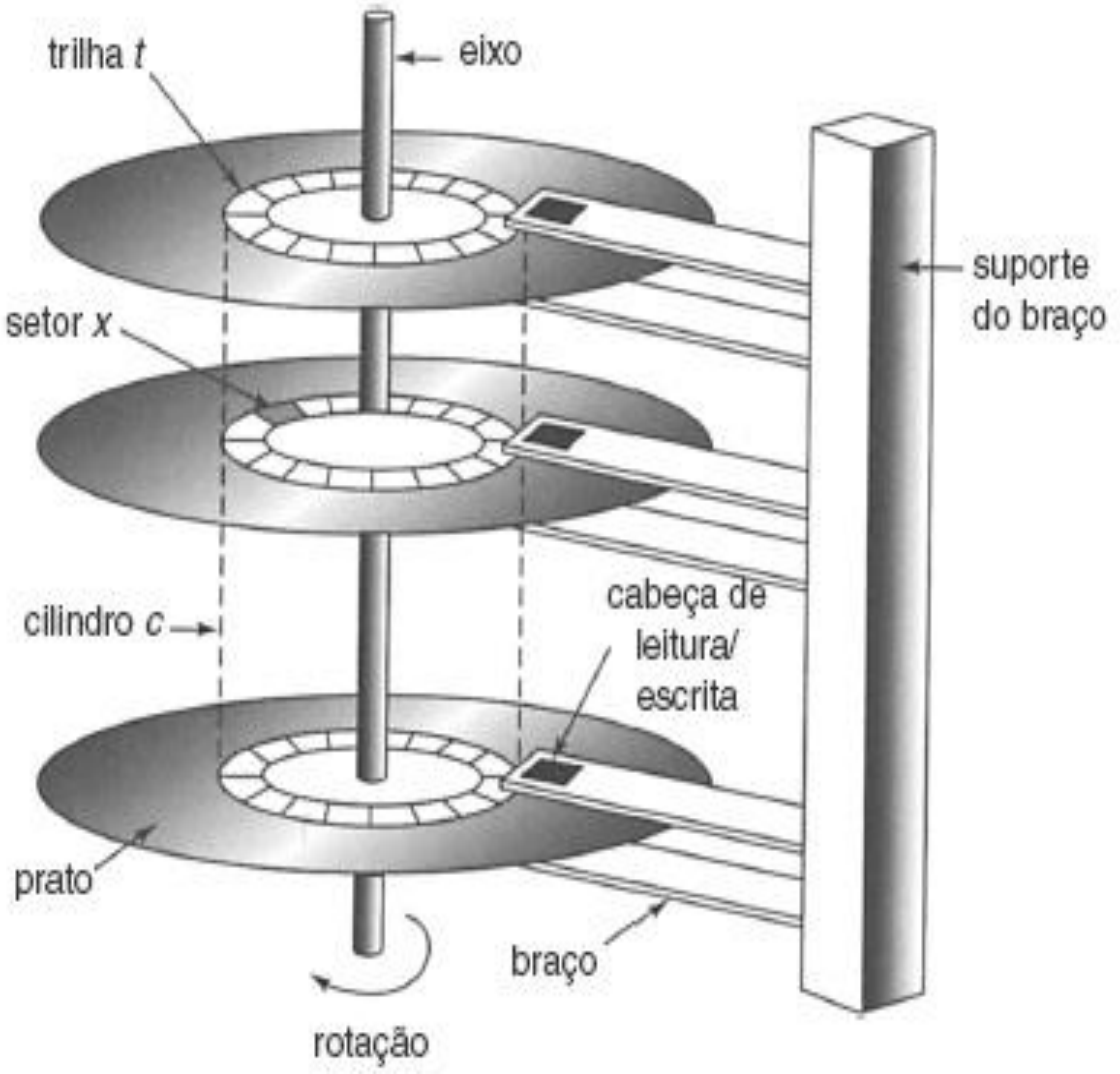
- Em geral, cada célula tem 8 bits
  - É possível armazenar 2^k combinações de bits
- Cada célula esta associada a um endereço
  - Se a memória possui n células, os endereços das células são numerados de 0 a n-1
  - São endereços fixos representados por números binários
- Capacidade da memória: número de células que pode endereçar vezes o número de bits da célula

Armazenamento secundário

- Extensão da memória principal que fornece grande capacidade de armazenamento não-volátil

- O dispositivo de memória secundária mais comum são os
- ***Discos magnéticos*** – discos de metal ou plástico cobertos com material de gravação magnético
  - A superfície do disco é dividida logicamente em **trilhas**, que são subdivididas em **setores**
  - O controlador de disco determina a interação lógica entre o dispositivo e o computador
  - O disco é um dispositivo mecânico, logo o tempo de acesso aos dados é lento (comparado à memória RAM)

Mecanismo de Movimentação da cabeça do HD



- Para acessar dados do disco é preciso mover o cabeçote de leitura/escrita para encontrar a trilha, aguardar até encontrar o setor correto (depende da velocidade de rotação do dispositivo acionador)
- Além disso, os dados devem ser transferidos para a memória principal para serem lidos pela CPU

Mecanismo de disco com cabeça móvel [Silberschatz]

Memória virtual

- A técnica de memória virtual é utilizada por muitos computadores, possibilita a execução de programas maiores que a memória principal colocando parte deles em disco e a memória principal é utilizada como um cache
- As partes mais utilizadas ficam na memória principal e o restante em disco
- Requer um mecanismo para converter endereços de memória física da RAM para a localização no disco

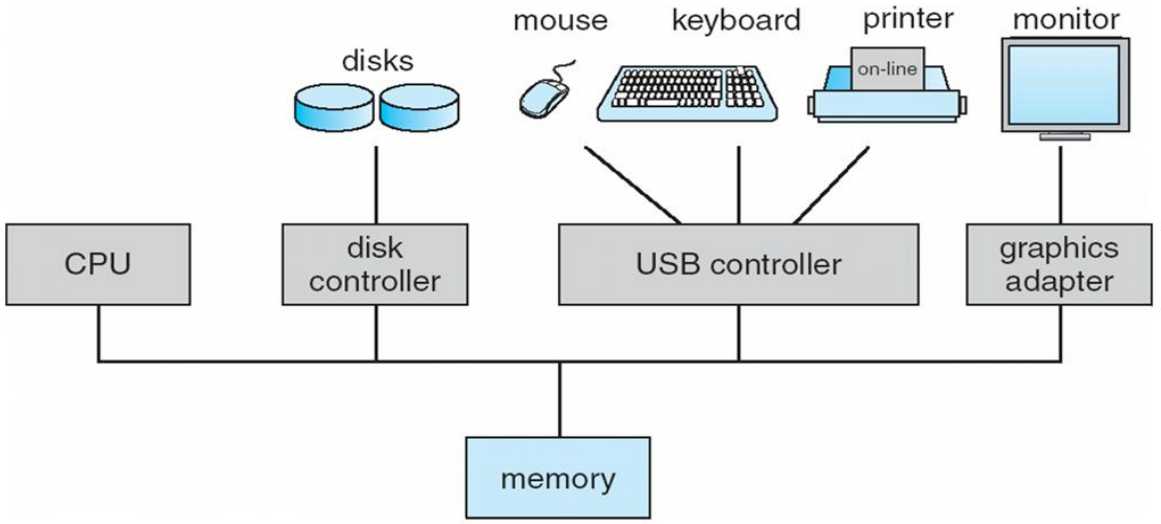
- (Memória virtual será visto mais para a frente)

UFABC

Dispositivos de E/S

Dispositivos de E/S

- Os dispositivos de E/S (Entrada/Saída) são constituídos por duas partes: o controlador do dispositivo e o dispositivo em si
- Cada controlador de dispositivo é responsável por um tipo de dispositivo específico (controla fisicamente o dispositivo) e interage com o SO
- O controle do dispositivo real é na maioria das vezes muito complicado, com muitos detalhes
- Em geral, os SOs possuem um *driver* de dispositivo para cada controlador que oferece uma interface uniforme para acessar o



Dispositivos de E/S

- Os controladores de dispositivo possuem um *buffer* local para armazenamento e alguns registradores específicos
- Por exemplo, o controlador de disco deve ter registradores para especificar endereços de memória, contador de setores (do disco) e indicador de direção (leitura/escrita)
- Para iniciar uma operação de E/S:
  - O *driver* recebe o comando, traduz o comando em valores apropriados e carrega os registradores
  - O controlador de dispositivo examina o conteúdo desses registradores para determinar a ação que deve ser realizada (por ex. “Ler um bloco de dados do disco”)
  - O controlador começa a transferir dados do dispositivo para seu *buffer* local
  - Quando a transferência de dados é concluída o controlador de dispositivo informa ao driver que terminou a operação
  - O término de uma operação de E/S pode ser através de:

- *Polling* – consulta periódica pela CPU

Operação de E/S e interrupção

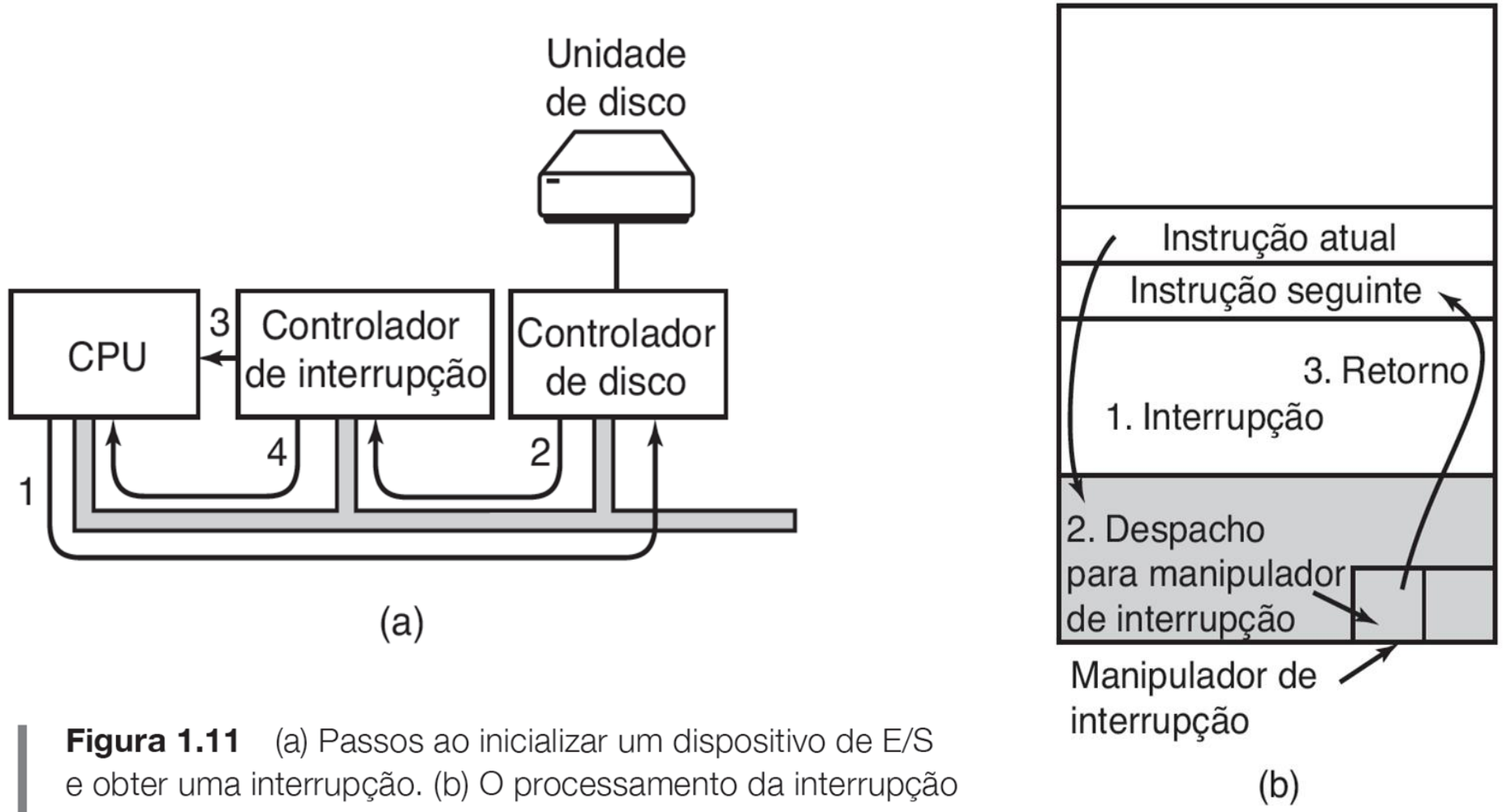
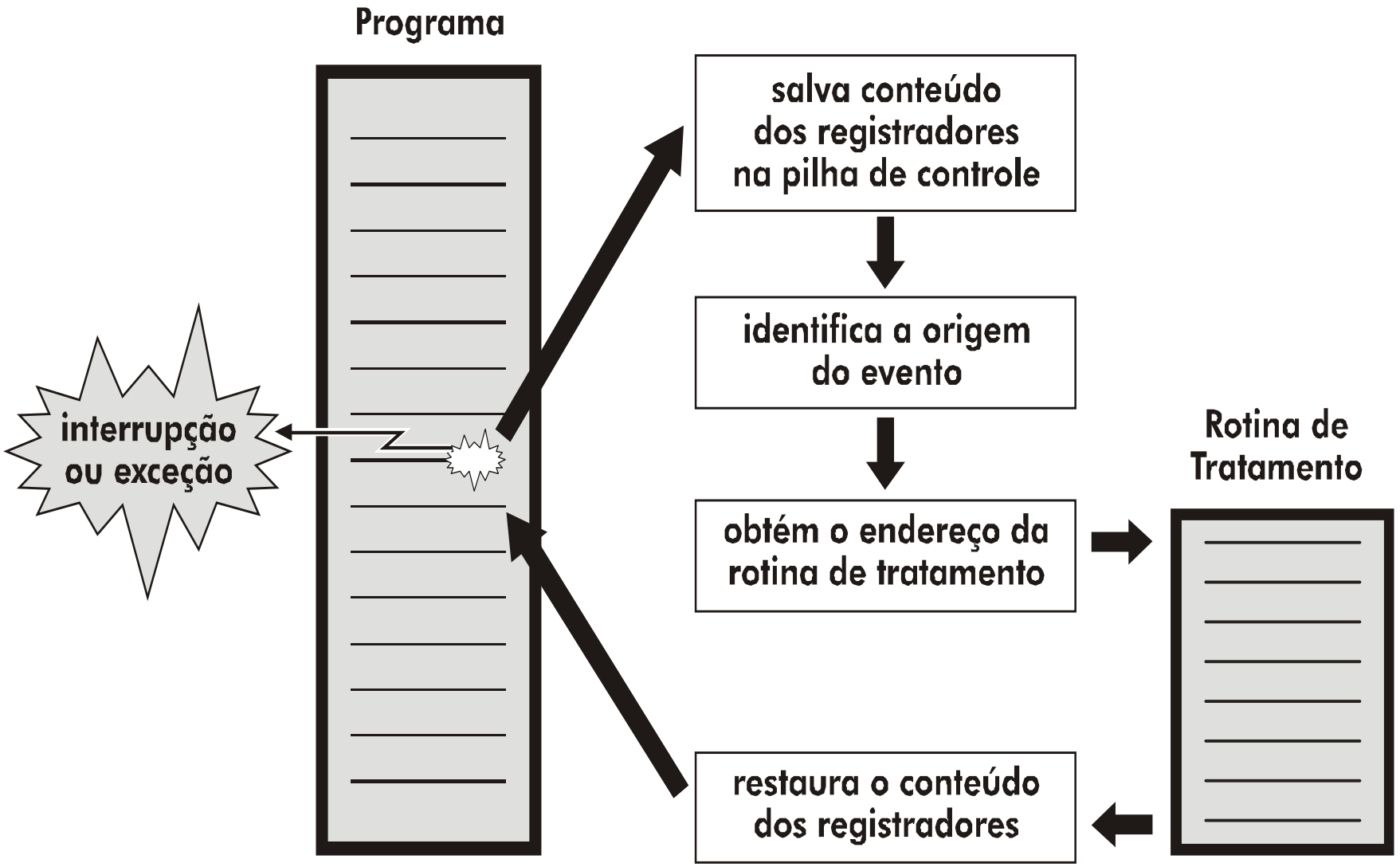


Figura 1.11 (a) Passos ao inicializar um dispositivo de E/S e obter uma interrupção. (b) O processamento da interrupção envolve fazer a interrupção, executar o manipulador de interrupção e retornar ao programa de usuário. [Tanenbaum]

Mecanismo de interrupção



Mecanismo de interrupção/exceção [Machado]

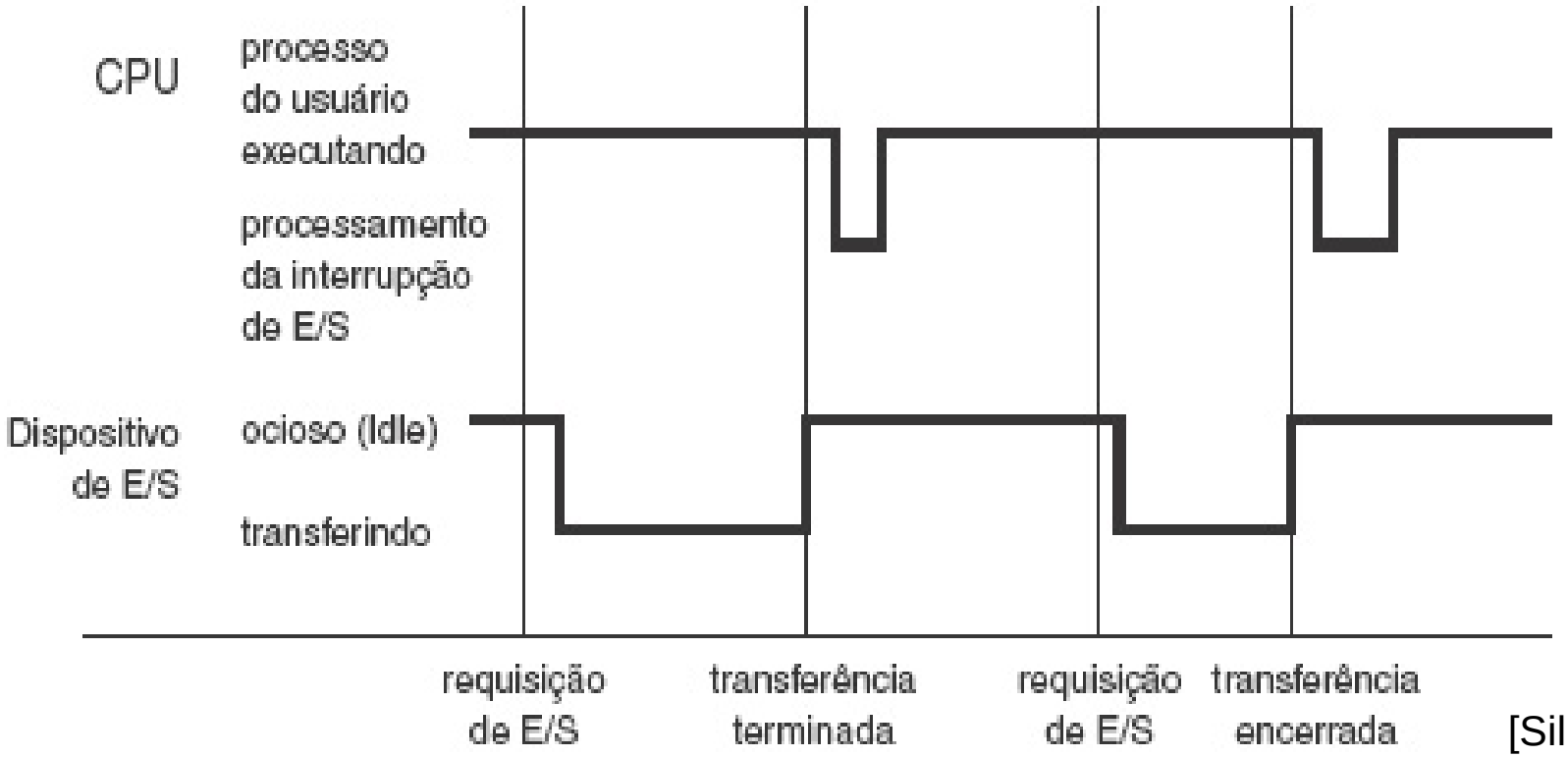
Funções comuns de Interrupções

- Para cada tipo de interrupção existe uma rotina de tratamento associada, para a qual o fluxo de execução deve ser desviado
- O tipo de evento ocorrido é fundamental para determinar o endereço de tratamento da rotina de tratamento
- A interrupção geralmente transfere o controle para o serviço de interrupção através do vetor de interrupção, que contém o endereço de todas as rotinas de serviço (tratadores de interrupção)
- O Sistema Operacional preserva o estado da CPU, armazenando o conteúdo de registradores e do contador de programa (o endereço de retorno é armazenado na pilha do sistema)

- Uma exceção (*trap*) é uma **interrupção gerada por software**, causada por um erro ou por uma requisição do usuário
  - Ex.: divisão por zero ou requisição de E/S pelo programa do usuário

- **Sistemas Operacionais modernos** são baseados em **interrupções**

Diagrama de Tempo de Interrupção para um único processo gerando saída



[Silberschatz]

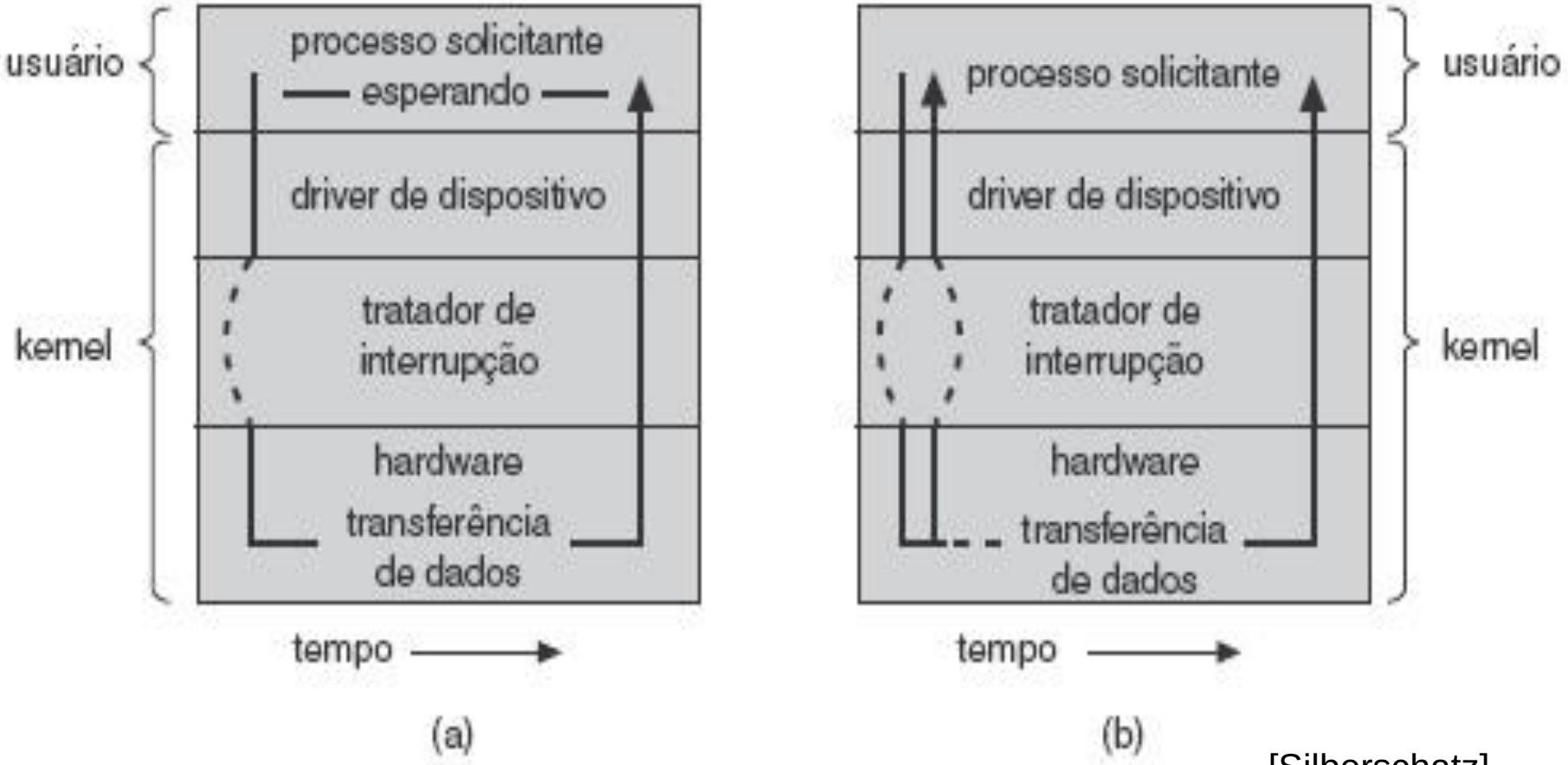
Quando um dispositivo de E/S termina a operação atribuída a ele, a CPU é informada através de uma interrupção. Então a CPU momentaneamente pára o que está sendo feito, trata a interrupção e coleta o resultado da operação de I/O

Estrutura de E/S

- E/S síncrona: Após o início da E/S, o controle retorna ao programa do usuário somente quando a E/S é finalizada
  - Apenas uma requisição de E/S pode estar pendente de cada vez, sem processamento de E/S concorrente

- E/S assíncrona: Após o início da E/S, o controle retorna ao SO ou ao programa do usuário sem esperar o término da E/S
  - É necessária uma requisição ao SO para permitir que o contexto seja salvo e o programa do usuário aguarde pelo término da E/S
  - Como vários dispositivos de E/S podem ser solicitados, há uma tabela de *status* de dispositivo contendo uma entrada para cada dispositivo de E/S indicando seu tipo, endereço e estado
  - O SO examina a tabela de dispositivo de E/S para determinar o status do dispositivo e para modificar a entrada da tabela a fim de incluir a requisição

Dois métodos de E/S

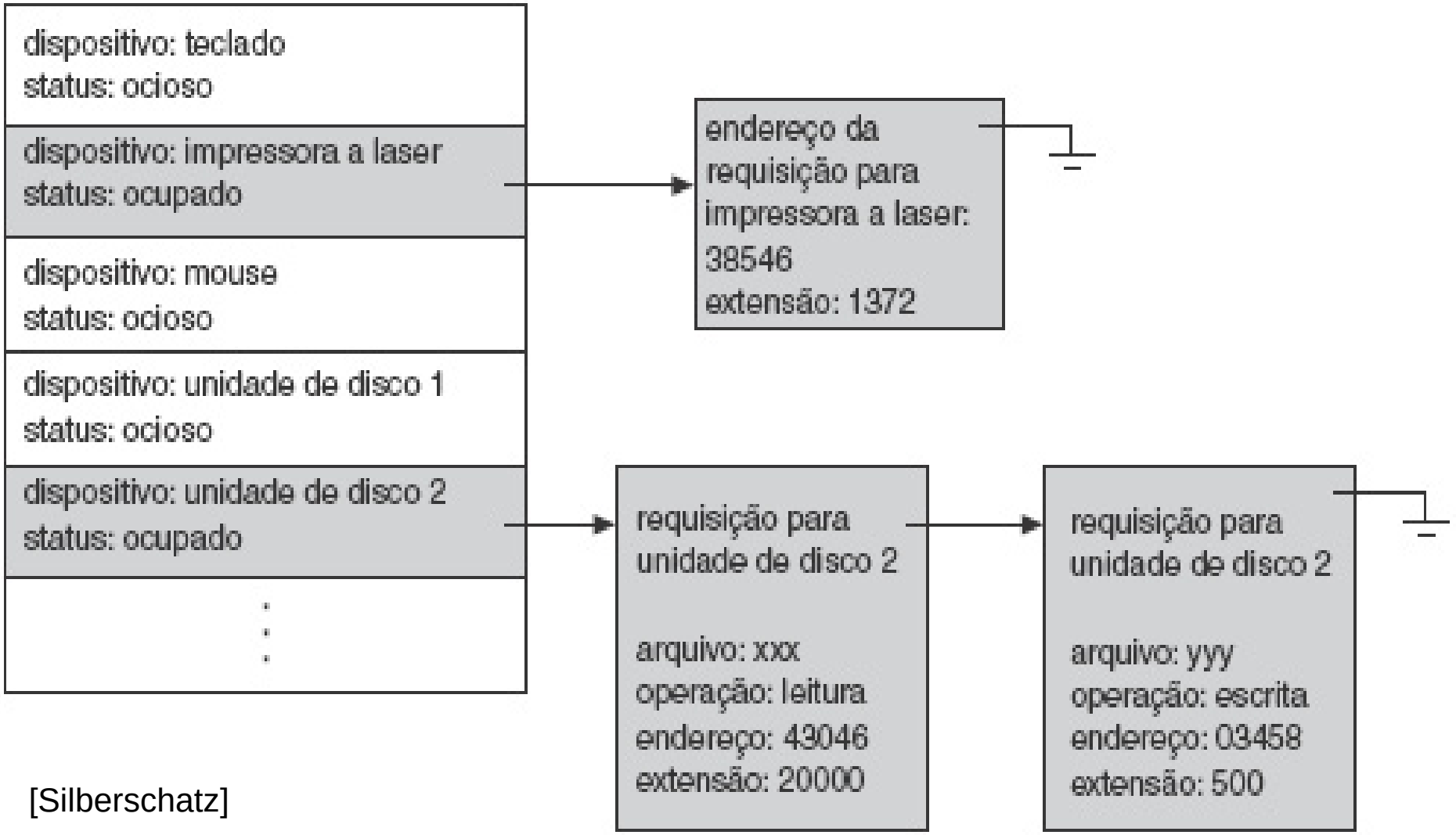


[Silberschatz]

(a) E/S síncrona; (b) E/S assíncrona

A principal vantagem da E/S assíncrona é maior eficiência do sistema: enquanto ocorre a operação de E/S, a CPU pode ser usada para outra tarefa

Tabela de *status* de dispositivos



[Silberschatz]

Quando é utilizado E/S assíncrona, é necessário armazenar as várias requisições de E/S simultâneas

Estrutura de acesso direto à memória (DMA – Direct Memory Access)

➢Usado para **dispositivos de E/S de alta velocidade** capazes de transmitir informações próximas às velocidades da memória

➢O DMA é um chip que controla a transferência de bits entre o controlador de algum dispositivo e a memória sem intervenção da CPU

➢Através do DMA, o controlador de dispositivo transfere **blocos de dados** diretamente do armazenamento em *buffer* para a memória principal

➢Apenas **uma interrupção é gerada por bloco**, em vez de uma interrupção por *byte*

Proteção de Hardware

- Operação em **Modo Dual**
- Proteção de I/O
- Proteção de Memória

Operação em Modo Dual

➢A fim de proteger o sistema e garantir a sua execução apropriada, é utilizado um mecanismo de proteção para evitar que usuários modifiquem o código do sistema ou acessem seus serviços indevidamente

➢Esse mecanismo (presente no hardware dos processadores) suporta dois modos de operação:

- **Modo monitor** (também chamado *modo supervisor* ou *modo do sistema* ou ***modo kernel***) – execução feita em nome do SO
  - Permite o acesso ao conjunto total de instruções, possibilitando o acesso irrestrito ao hardware
- **Modo usuário –** execução feita em nome de um usuário
  - Permite executar apenas um número reduzido de instruções, conhecidas como não-privilegiadas

➢As intruções privilegiadas são instruções que podem ser acessadas somente pelo sistema operacional ou, por intermédio dele, evitando assim problemas de segurança e integridade do sistema

Operação em Modo Dual (cont.)

➢O modo de operação é determinado pelo bit de modo no *hardware* do computador (no registrador de *status* da CPU) e indica o modo de operação corrente

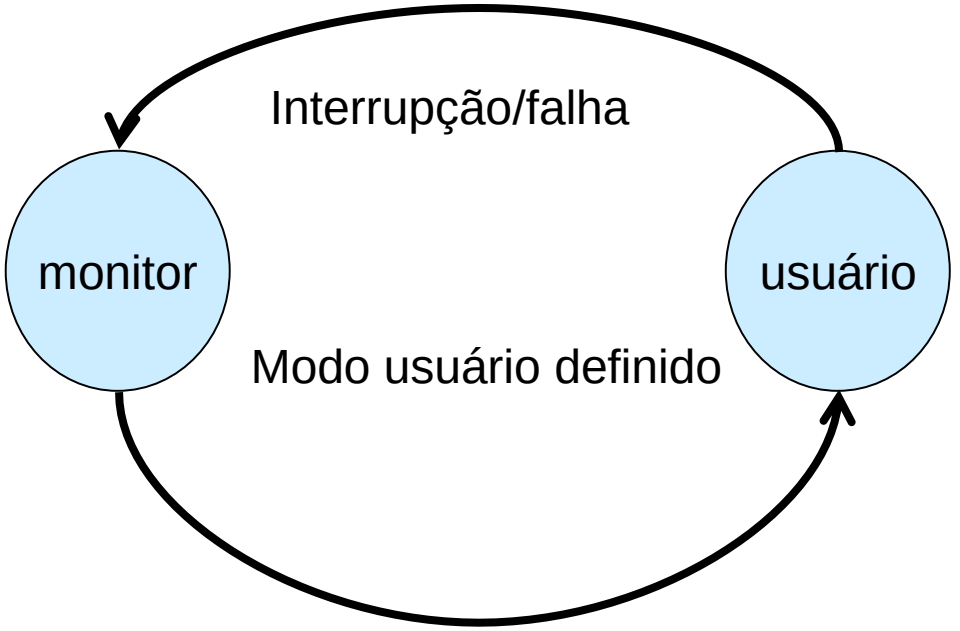
➢São utilizados os valores:

- **0 – modo monitor** ou
- **1 – modo usuário**

➢Através do bit de modo é avaliado se a instrução corrente pode ou não ser executada

Operação em Modo Dual (cont.)

➢Quando ocorre uma interrupção ou erro, o bit de modo é alterado para o modo monitor, de modo que o sistema operacional possa executar a rotina de tratamento correspondente



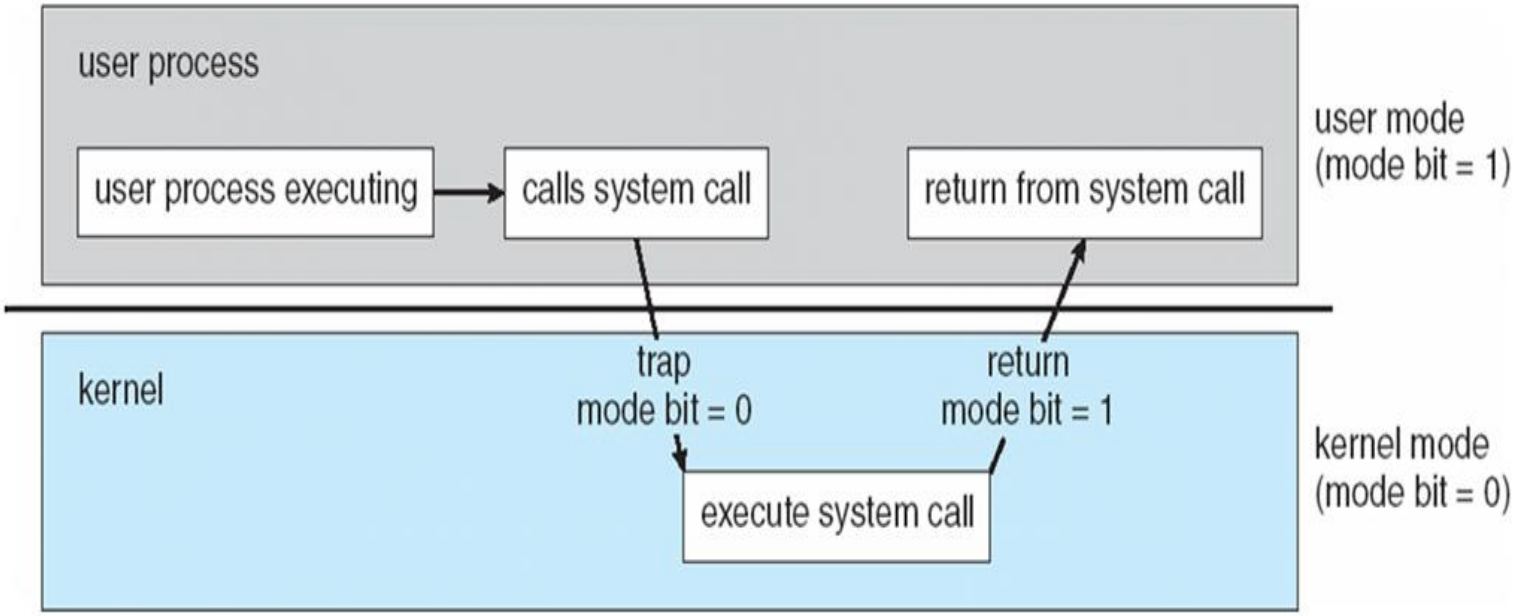
Proteção de E/S

- Todas as **instruções de E/S são instruções privilegiadas**

- É preciso assegurar que um **programa do usuário nunca possa obter o controle do computador no modo monitor** (por exemplo, um programa do usuário que, como parte de sua execução, consiga colocar um novo endereço no **vetor de interrupção**; se esse novo endereço apontar para a área do usuário, este pode assumir o controle do sistema no **modo monitor**!)

Chamadas de sistema

- Sabendo-se que as **instruções de E/S são privilegiadas**, como o programa do usuário efetua operações de E/S?
  - Através de **Chamada ao Sistema** (*system call*), que é o método usado para requisitar uma ação do Sistema Operacional
  - Por exemplo, se um programa deseja realizar uma operação de leitura de disco, é feita uma chamada ao sistema. Essa chamada é tratada pelo SO, que avalia a requisição e se a requisição for válida, executa-a em seu nome (muda o bit de modo para 0)

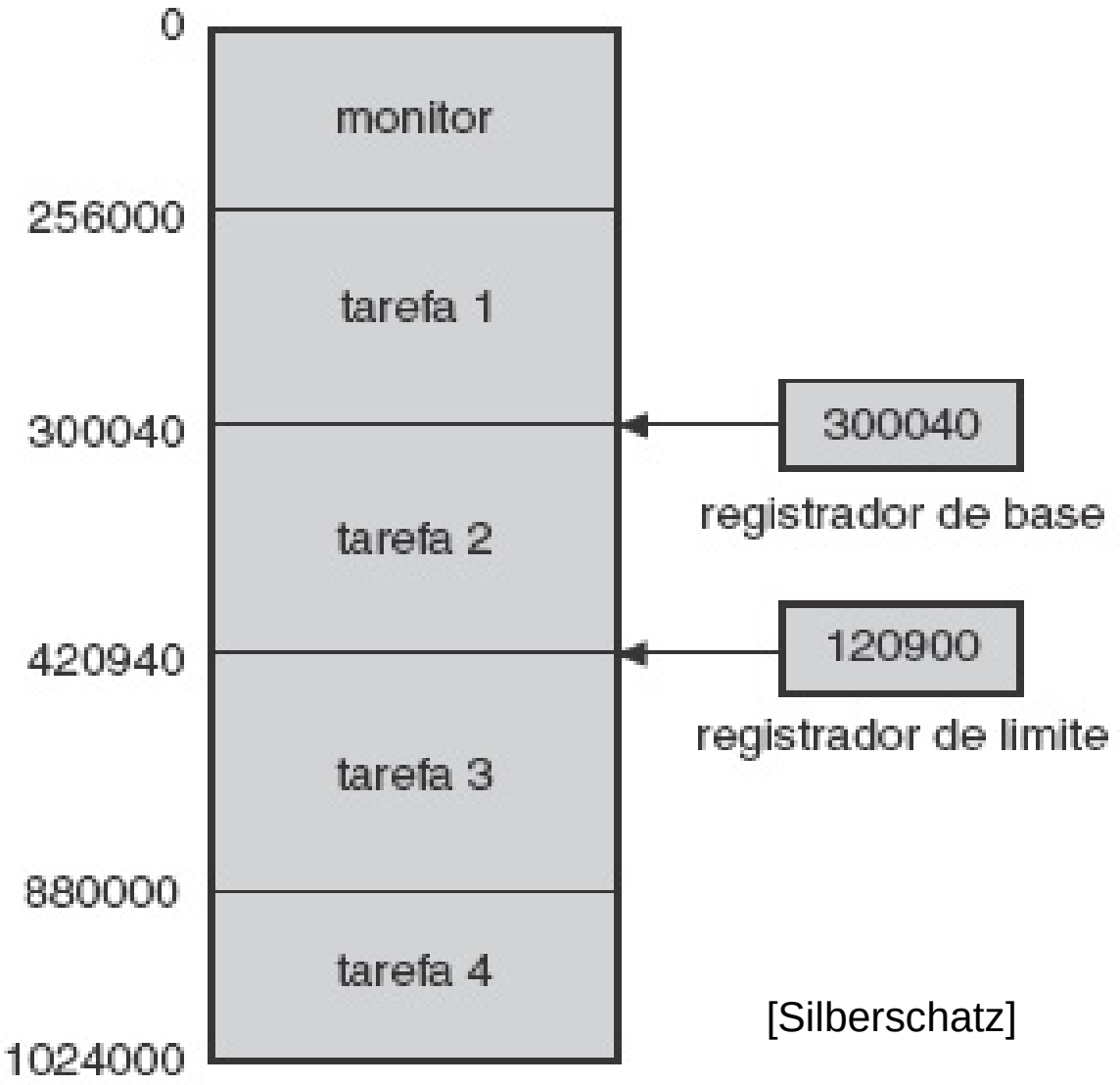


Transição do modo usuário para modo kernel [Silberschatz]

Proteção de Memória

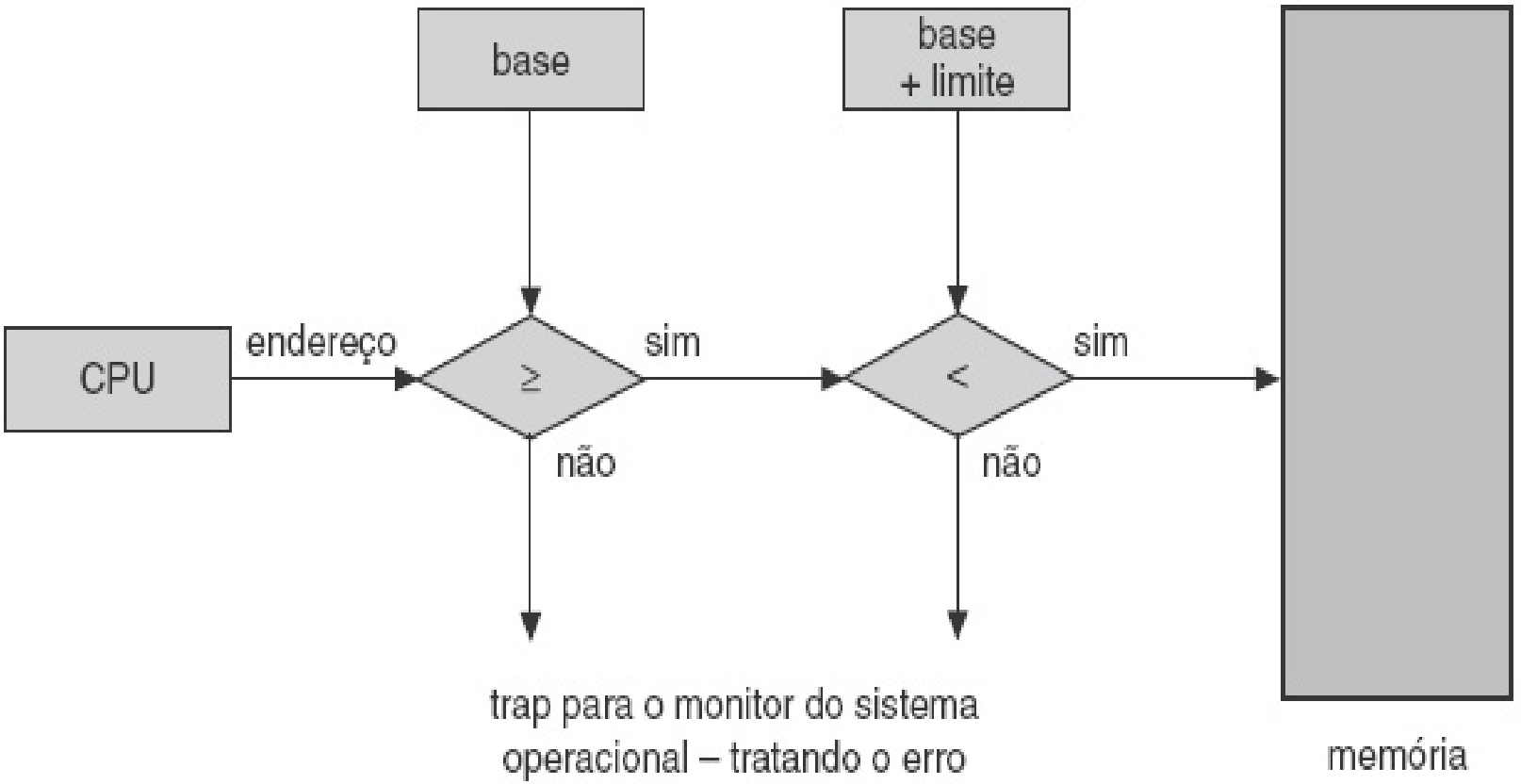
- É necessário proteger a memória de programas que tentem acessar um espaço que não foi lhes foi designado (por exemplo, a memória do SO e memória de outros usuários)
- Essa proteção de memória é realizada por meio da especificação de limites na memória que determinam o espaço alocado para um determinado programa
- Esses limites são implementados através de dois registradores no processador e só podem ser modificados por instruções privilegiadas
- Esses dois registradores que determinam o intervalo de endereços válidos que um programa pode acessar são:
  - **Registrador de base:** contém o menor endereço de memória física válido
  - **Registrador de limite:** contém o tamanho do intervalo
- A memória fora do intervalo definido é protegida
- Quando executado no **modo monitor**, o sistema operacional

Uso de registradores de base e de limite



[Silberschatz]

Proteção de endereços via hardware



[Silberschatz]

Referências

- [Silberschatz] SILBERCHATZ, A., GALVIN, P. B. e GAGNE, G. **Sistemas Operacionais com Java**. 7ª ed., Rio de Janeiro: Elsevier, 2008.
- [Tanenbaum] TANENBAUM, A. **Sistemas Operacionais Modernos**. 3ª ed. São Paulo: Prentice Hall, 2009.
- [MACHADO] MACHADO, F. B. e MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 4ª ed., Rio de Janeiro: LTC, 2007.