

Sistemas Operacionais

Prof. Jó Ueyama

Apresentação baseada nos slides da Profa. Dra. Kalinka Castelo Branco, do Prof. Dr. Antônio Carlos Sementille e da Profa. Dra. Luciana A. F. Martimiano e nas transparências fornecidas no site de compra do livro “Sistemas Operacionais Modernos”

Dispositivos de Entrada e Saída

- ★ SO pode atuar de duas maneiras diferentes:
 - Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário;
 - Como gerenciador de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador;

Dispositivos de Entrada e Saída

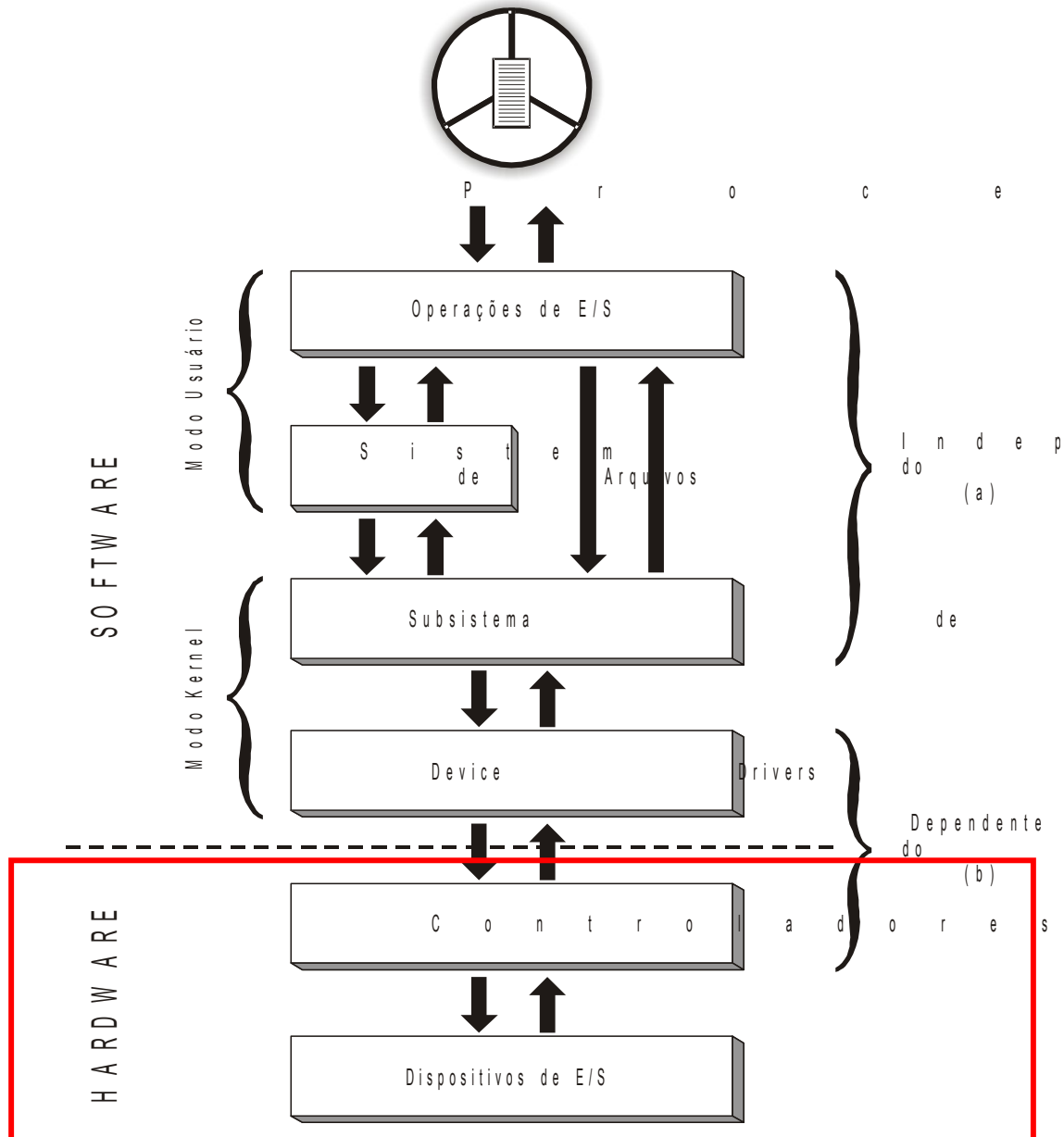
★ Funções específicas:

- Enviar sinais para os dispositivos;
- Atender interrupções;
- Gerenciar comandos aceitos e funcionalidades (serviços prestados);
- Tratar possíveis erros;
- Prover interface entre os dispositivos e o sistema;

★ Princípios:

- Hardware;
- Software;

Dispositivos de Entrada e Saída



Gerência de E/S

GERENCIAMENTO DE ENTRADA E SAÍDA

Uma das funções principais de um Sistema Operacional é **controlar todos os dispositivos de entrada/saída** do computador. Ele deve:

- enviar comandos aos dispositivos;

- atender interrupções;

- fornecer uma interface entre os dispositivos e o resto do sistema que seja simples e fácil de usar.

Geralmente, o código para tratamento da entrada e saída representa uma fração significativa do sistema operacional total

Gerência de E/S

Módulos de E/S: Controladores de Dispositivos

Princípios do Hardware de E/S

As Unidades de E/S são geralmente compostas de dois componentes principais:

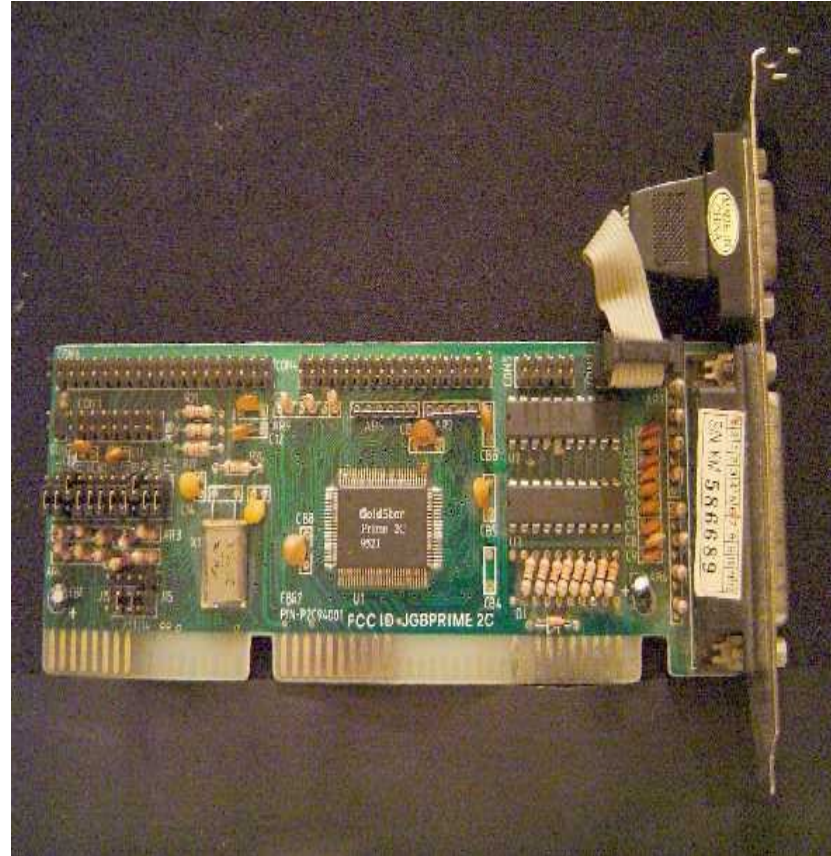
Controlador de dispositivo: parte programável (Nos PCs é normalmente uma placa de circuito impresso);

Componente Mecânico

Muitos controladores podem controlar vários dispositivos idênticos

Órgãos de padronização: IEEE, ISO, ANSI, etc.

Gerência de E/S



Gerência de E/S

- | O S.O. sempre trata com o controlador, não com os dispositivos.**
- | A Comunicação entre UCP e controladores é feita através de barramentos comuns (interface de alto nível)**
- | Interface entre controlador e dispositivo: baixo nível**
- | Mainframes: múltiplos barramentos e processadores especializados em E/S (canais de E/S).**

Gerência de E/S

- | **Controlador de disco:** converte o fluxo serial de bits em um bloco de bytes, executando qualquer correção necessária.
- | Cada controlador possui registradores para a comunicação com a UCP.
- | Em alguns computadores: estes registradores podem fazer parte do espaço de endereçamento da memória principal.

Comando, controlador e interrupção

- I O S.O.:** executa E/S escrevendo comandos (e seus parâmetros, se existirem) nos registradores dos controladores.
- I** Quando um comando é aceito, a UCP pode deixar que o controlador trabalhe sozinho, indo executar outra tarefa.
- I** Quando o dispositivo termina, avisa a UCP através de uma interrupção.

Gerência de E/S

! Operação dos Módulos de E/S

! Os módulos de E/S podem operar de 3 maneiras básicas:

- ! E/S programada
- ! E/S via Interrupções
- ! E/S via Acesso Direto à Memória

! O que distingue as três formas: a participação da UCP e a utilização das interrupções

Gerência de E/S

! E/S Programada (CPU realiza todo o trabalho)

! Na E/S programada: os dados são trocados entre a UCP e o Módulo de E/S

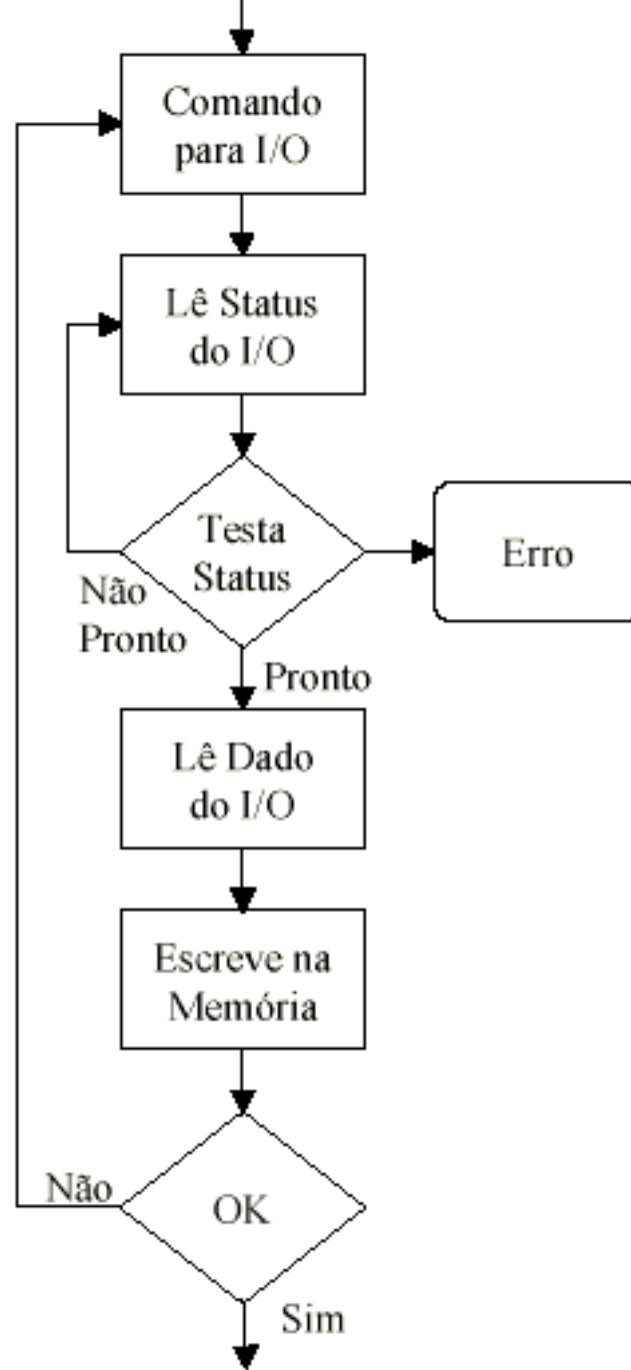
! A UCP executa um programa que:

- ! verifica o estado do módulo de E/S, preparando-o para a operação;

- ! se necessário, enviando o comando que deve ser executado; e

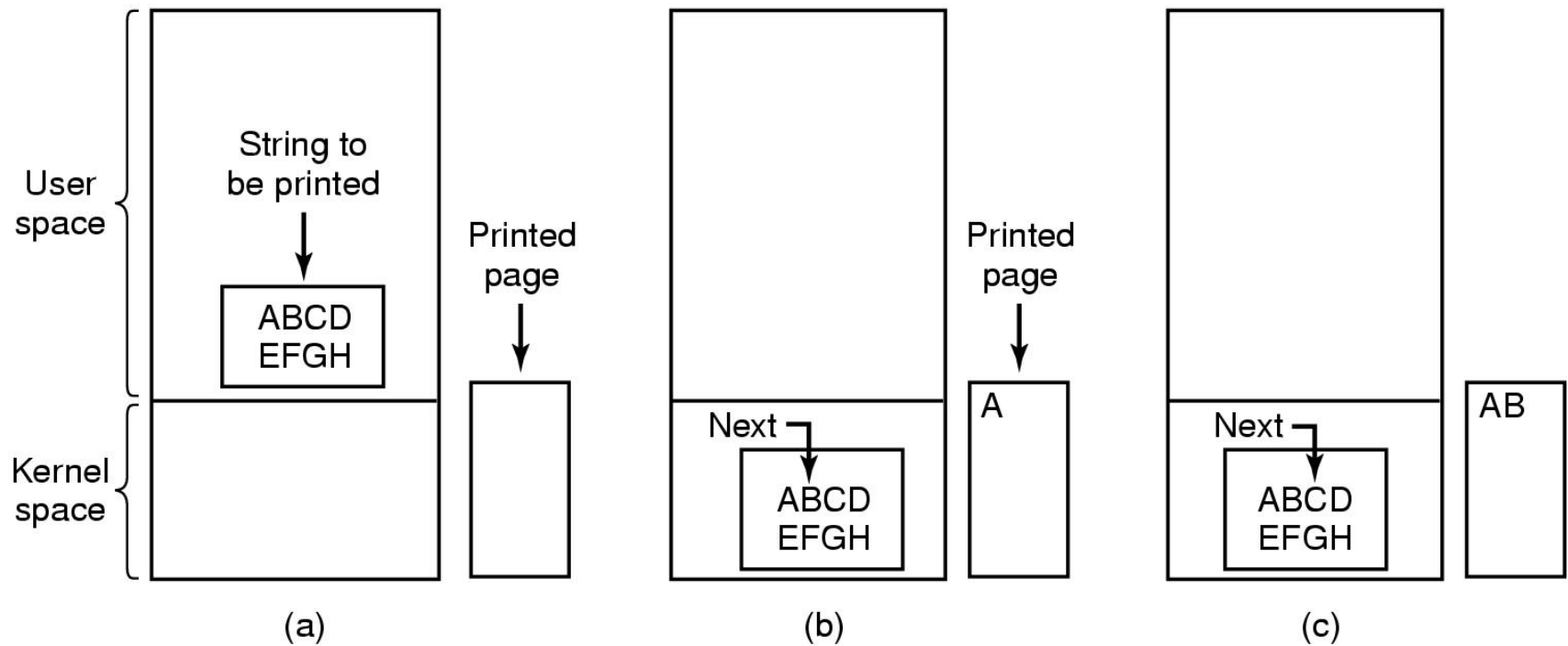
- ! **aguardando** o resultado do comando, para então, efetuar a transferência entre o módulo de E/S e algum registrador da UCP.

I E/S Programada



E/S programada

- * E/S programada: passos para impressão de uma cadeia de caracteres (laço até que toda a cadeia tenha sido impressa);
- * Impressoras modernas requerem uma ou várias páginas a serem impressas em vez de caracteres



Dispositivos de E/S

E/S programada

- ★ E/S programada:

- Desvantagem:

- ★ CPU é ocupada o tempo todo até que a E/S seja feita;
 - ★ CPU continuamente verifica se o dispositivo está pronto para aceitar outro caracter → espera ocupada;
 - ★ CPU deveria estar ocupado com trabalhos mais “nobres”

Gerência de E/S

I E/S via Interrupção

I Na E/S via interrupção: o mecanismo de interrupções é utilizado para superar o problema da espera da UCP por operações nos periféricos

I A interrupção permite que uma unidade ganhe a atenção imediata de outra, de forma que a primeira possa finalizar sua tarefa

I A UCP:

I envia um comando para o módulo de E/S e passa a executar outra tarefa;

I quando a operação for concluída, o módulo de E/S interrompe a UCP; e

I a UCP executa a troca de dados, liberando o módulo de E/S e retomando o processamento anterior.

Gerência de E/S

E/S via Interrupção

I Usualmente: são assinalados números para as interrupções, onde o menor número tem prioridade sobre o maior

Gerência de E/S

E/S via Interrupção

Exemplo de mapeamento das interrupções em um sistema IBM compatível

Int	Dispositivo	Int	Dispositivo
0	Cronômetro do sistema	9	Porta de comunicação COM3
1	Teclado	10	Porta de comunicação COM2
2	Controlador de interrupção	11	Ponte PCI (*)
4	Porta de comunicação COM1	12	Mouse porta PS/2 (*)
5	Placa de som (*)	13	Coprocessador numérico
6	Controlador de disco flexível	14	Controlador IDE/ESDI
7	Porta de Impressora LPT1	15	Controlador IDE/ESDI
8	CMOS/Relógio do sistema		(*) Opções não padronizadas

Mapa de Interrupções num IBM-PC compatível

Dispositivos de E/S

E/S por interrupção

★ E/S orientada à interrupção:

- No caso da impressão, a impressora pode armazenar dados a serem impressos no seu buffer ou não armazenar os caracteres;
- Em impressoras modernas páginas ou o documento inteiro pode ser enviado para o buffer da impressora;
- Quando a impressora está pronta para receber outros caracteres, gera uma interrupção;
- Processo é bloqueado;

Gerência de E/S

I E/S via Acesso Direto à Memória

I Inconvenientes das técnicas anteriores:

- I limitam a capacidade de transferência da UCP, entre o módulo de E/S e a Memória Principal
- I uso de mais de uma instrução
- I **UCP fica ocupada no gerenciamento**
- I se a quantidade de dados for grande, o desempenho do sistema será comprometido

I A solução deste problemas: permitir o acesso direto à memória

- I o método propõe o uso de uma única interrupção, para efetuar a transferência de um bloco de dados entre o periférico e a memória principal
- I **UCP tem envolvimento mínimo no gerenciamento**

Gerência de E/S

I E/S via Acesso Direto à Memória

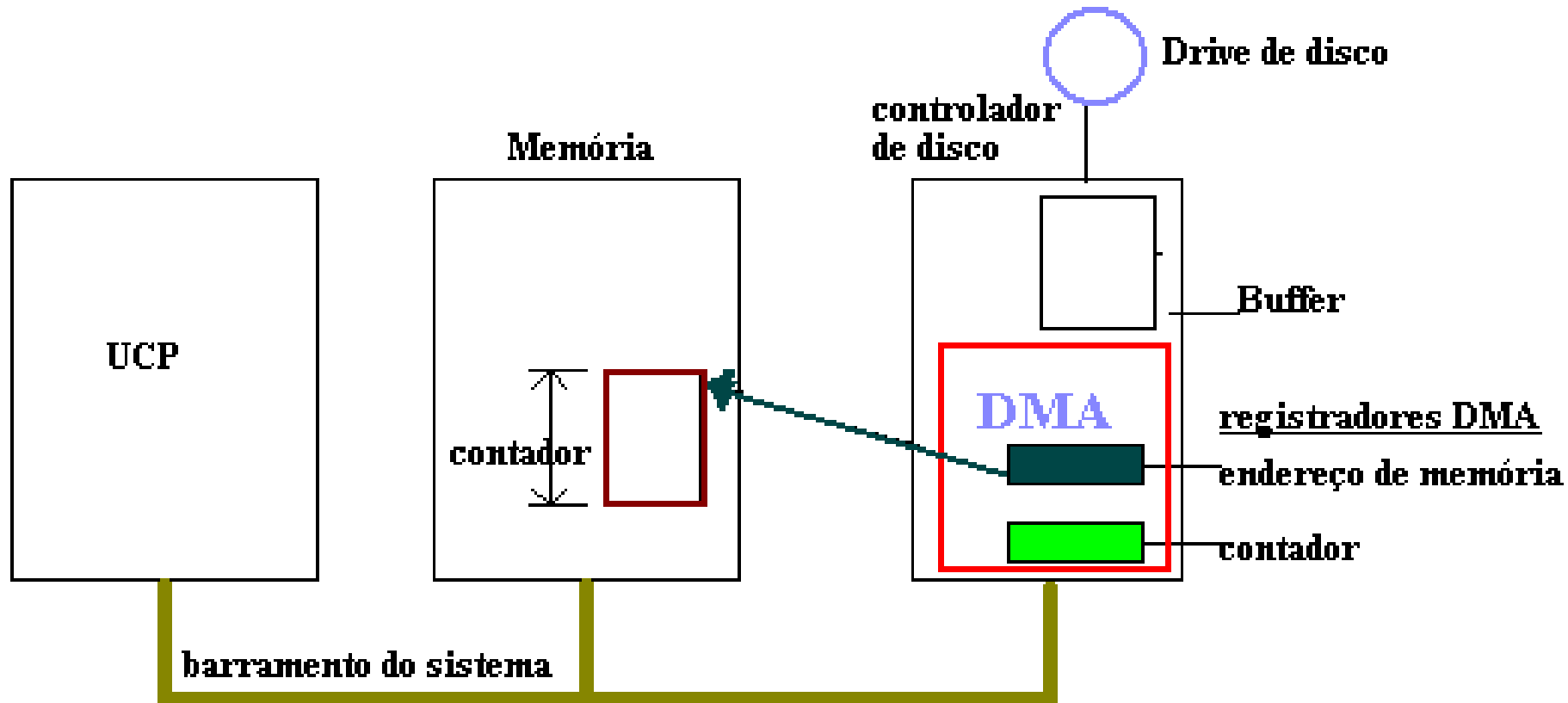
I Necessidade de um módulo adicional: o **Controlador de DMA**

I **Operação do Controlador de DMA:**

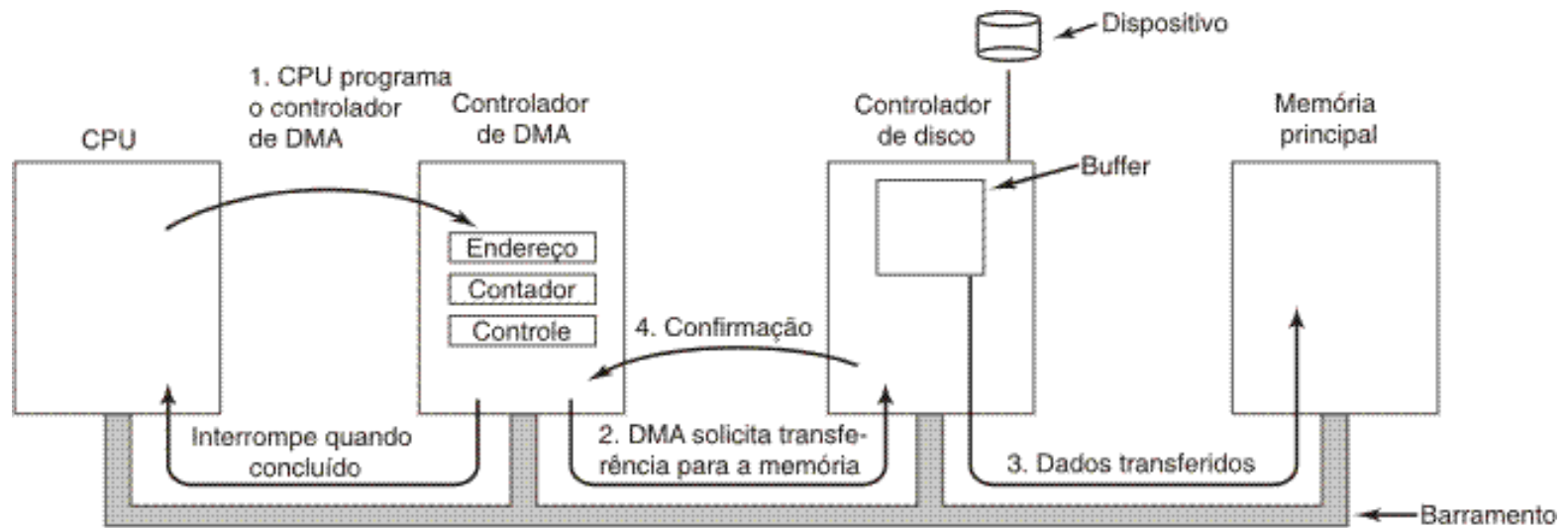
- I **UCP envia comando (leitura ou escrita) para o controlador de DMA**
- I **UCP continua seu trabalho**
- I **O controlador de DMA, para acessar memória, “rouba” ciclos da UCP, atrasando-a apenas**
- I **Ao final da operação, o controlador de DMA aciona a interrupção para sinalizar o término da operação**
- I **A UCP pode executar a rotina de tratamento da interrupção, processando os dados lidos ou produzindo novos dados para serem escritos**

Gerência de E/S

E/S via Acesso Direto à Memória (DMA)



Acesso Direto à Memória (DMA)



Operação de uma transferência com DMA

Dispositivos de E/S

Princípios de Software

* E/S com uso da DMA:

- DMA executa E/S programada → controladora de DMA faz todo o trabalho ao invés da CPU;
 - * Redução do número de interrupções;
- Desvantagem:
 - * DMA é mais lenta que a CPU;
 - * Pode deixar a CPU ociosa e o DMA trabalhando a todo “vapor”

Dispositivos de E/S

Princípios de Hardware

- ★ Sem DMA: Leitura de um bloco de dados em um disco:
 - Controladora do dispositivo lê bloco (bit a bit) a partir do endereço fornecido pela CPU;
 - Dados são armazenados no *buffer* da controladora do dispositivo;
 - **Controladora do dispositivo checa consistência** dos dados;
 - Controladora do dispositivo gera interrupção;
 - SO lê (em um *loop*) os dados do *buffer* da controladora do dispositivo e armazena no endereço de memória fornecido pela CPU;

Dispositivos de E/S

Princípios de Hardware

- ★ Com DMA: Leitura de um bloco de dados em um disco:
 - 1. Além do endereço a ser lido, a CPU fornece à controladora de DMA duas outras informações: **endereço na RAM para onde transferir os dados e o número de bytes a ser transferido**;
 - 2. Controladora de DMA envia dados para a controladora do dispositivo;
 - Controladora do dispositivo lê o bloco de dados e o armazena em seu *buffer*, verificando consistência;
 - 3. **Controladora do dispositivo copia os dados para RAM** no endereço especificado na DMA (modo direto);

Dispositivos de E/S

Princípios de Hardware

- 4. Após confirmação de leitura, a controladora de DMA **incrementa o endereço de memória na DMA e decrementa o contador da DMA com o número de bytes transferidos; por que?**
- Repete os passos de 2 a 4 até o contador da DMA chegar em 0. **Assim que o contador chegar em zero (0), a controladora de DMA gera uma interrupção avisando a CPU;**
- Quando o SO inicia o atendimento à interrupção, o bloco de dados já está na RAM;

Dispositivos de E/S

Princípios de Hardware

- ★ A DMA pode tratar múltiplas transferências simultaneamente:
 - Possuir vários conjuntos de registradores;
 - Decidir quais requisições devem ser atendidas → escalonamento (*Round-Robin* ou prioridades, por exemplo);

Gerência de E/S

I E/S via Acesso Direto à Memória

I O controlador de DMA pode suportar, tipicamente, o trabalho com vários periféricos diferentes, cada um utilizando um canal de DMA (*DMA channel*)

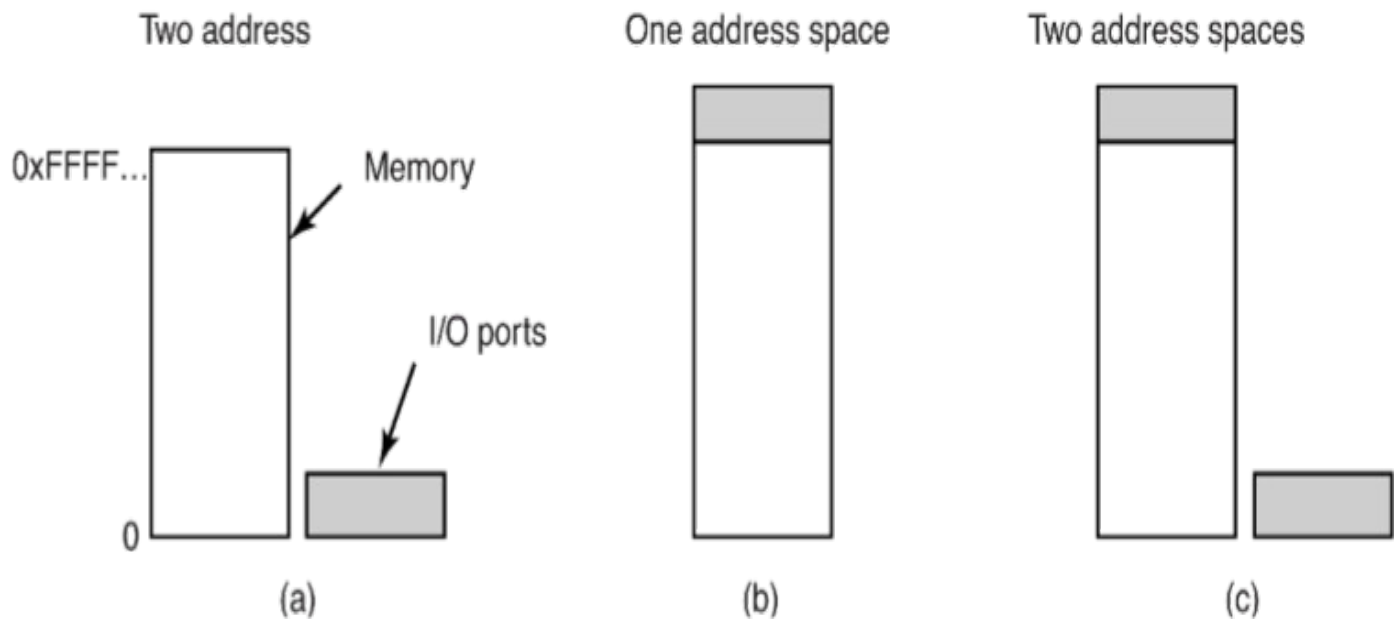
I Outra vantagem do DMA: pode ser implementada em hardware de diversas formas diferentes, conforme a quantidade de dispositivos e o desempenho pretendido

Módulo de E/S controlando múltiplos dispositivos

- ! Associação de endereços aos múltiplos dispositivos
- ! O SO pode saber o status de cada dispositivo a partir dos registradores e/ou memória
- ! Duas abordagens de endereços pelo módulo de E/S:
 - **Mapeado em Memória:** o módulo de E/S opera dentro do espaço de endereçamento de memória, usando um conjunto de endereços reservados (registradores são tratados como posições de memórias);
 - **Mapeado em E/S ou E/S isolada:** existe um espaço de endereçamento independente para os dispositivos de E/S. Uso de instruções especiais de E/S. **Desvantagens?**

Espaços de memória e E/S separados;

(b) E/S mapeada na memória; (c) Híbrido



Gerência de E/S

I Nos PCs: é utilizado um espaço de endereçamento especial para a E/S, com cada controlador alocado em certa posição da mesma.

Controlador es de E/S	Endereços de E/S
Timer (relógio do sistema)	040-043H
Teclado	060-063H
...	...

Gerência de E/S

4.1.2 Tipos de E/S

Os dispositivos de E/S podem ser classificados de forma ampla, sendo que as mais utilizadas são quanto ao:

- ▀ tipo de conexão
- ▀ tipo de transferência de dados
- ▀ tipo de compartilhamento de conexões

Quanto ao tipo de conexão:

- ▀ Leva em consideração a natureza da conexão entre o módulo de E/S e o periférico
- ▀ Do ponto de vista dos dados, as conexões são projetadas para operação:
 - ▀ Serial
 - ▀ Paralela

Gerência de E/S

4.1.2 Tipos de E/S

Os dispositivos de E/S podem ser classificados de forma ampla, sendo que as mais utilizadas são quanto ao:

- tipo de conexão

- tipo de transferência de dados

- tipo de compartilhamento de conexões

Quanto ao tipo de conexão:

Leva em consideração a natureza da conexão entre o módulo de E/S e o periférico

Do ponto de vista dos dados, as conexões são projetadas para operação:

- Serial

- Paralela

Gerência de E/S

Conexão serial:

- Uma **única linha** de sinal é utilizada para o estabelecimento de toda a **conexão, protocolo e transferência de dados**, entre o módulo de E/S e o periférico

- Características principais:

- mais barata que a paralela
- mais lenta que a paralela
- relativamente confiáveis
- usada em dispositivos mais baratos e lentos, como impressoras e terminais

Gerência de E/S

Conexão paralela:

- ▀ Várias linhas de sinais são usadas, de forma que vários bits de dados possam ser transferidos em paralelo
- ▀ É comum que existam linhas independentes para tráfego de sinais de controle
- ▀ Características principais:
 - ▀ mais complexa que a serial
 - ▀ mais cara
 - ▀ mais rápida
 - ▀ altamente confiável
 - ▀ usada em dispositivos mais velozes, como unidades de disco, fita ou impressoras rápidas

Gerência de E/S

Classificação quanto ao tipo de transferência de E/S

Podem ser divididos em 2 categorias:

Dispositivos de Bloco – armazenam informações em blocos de tamanhos fixos, cada um com seus próprios endereços.

- Os tamanhos dos blocos geralmente variam de 256 à 64K bytes.
- Independência para ler e escrever cada bloco de maneira individual
- Como exemplo, temos os discos flexíveis e discos rígidos. **Por que blocos?**

Gerência de E/S

- I Dispositivos de Caracter** – são capazes de acessar um fluxo de caracteres;
- I Não consideram qualquer estrutura de bloco**
- I Não são endereçáveis**
- I Não possuem qualquer operação de acesso aleatório (“seek operation”)**
- I Os terminais, impressoras, interfaces de redes e etc., fazem parte desta categoria.**

Gerência de E/S

- | Este esquema de classificação não é perfeito, porém é genérico o suficiente (por ex., o timer não se encaixa).**

- | O sistema de arquivos, por exemplo, trata com dispositivos de bloco abstratos.**

Gerência de E/S

Classificação quanto ao compartilhamento de conexões

Podem ser divididos em 2 categorias:

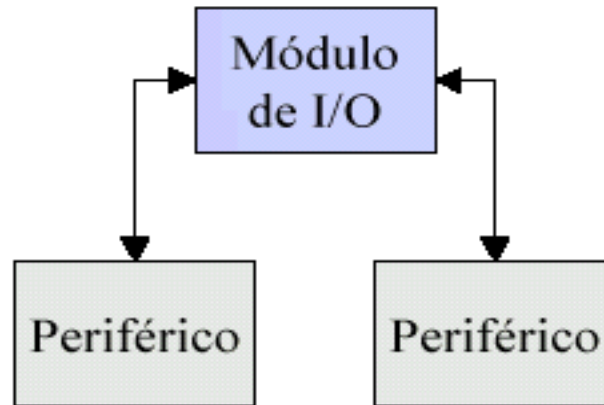
Ponto-a-Ponto – é a conexão mais simples, onde existe um conjunto de linhas **dedicadas** para a ligação entre o módulo de E/S e cada periférico.

Multiponto - neste tipo de conexão, um módulo de E/S **compartilha** um conjunto de linhas de sinais entre diversos periféricos. **Qual o problema neste tipo de conexão?**

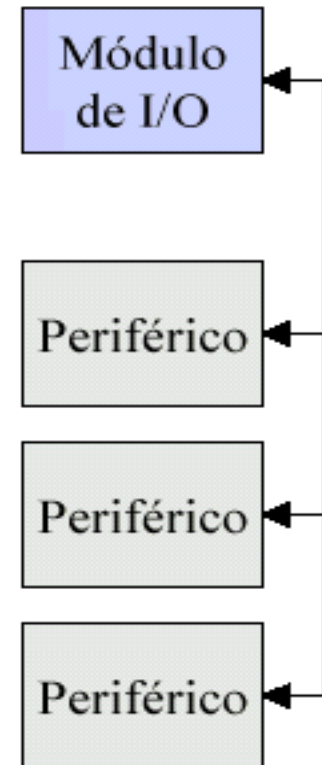
Gerência de E/S

Classificação quanto ao compartilhamento de conexões

Ponto-a-Ponto



Multiponto



Gerência de E/S

Conexões Ponto-a-Ponto:

- oferecem maior confiabilidade
 - permite a operação simultânea de diversos dispositivos
 - é usada em dispositivos mais simples, tais como modems, teclado e impressora
-
- Tem-se os seguintes exemplos de conexões ponto-a-ponto padronizadas, usados em comunicação de curta distância, usualmente na interface padrão **RS - 232C**:
 - **Protocolo RTS/CTS** (*Request to Send/Clear to Send*)
 - **Protocolo Xon/Xoff** (*Transmission On/Transmission Off*)

Dispositivos de E/S

Tratando Interrupções

- ★ Sinal (linha) de interrupção é exibido dentro de cada ciclo de instrução do processador;
- ★ Se sinal ativo → salva contexto e atende a interrupção;

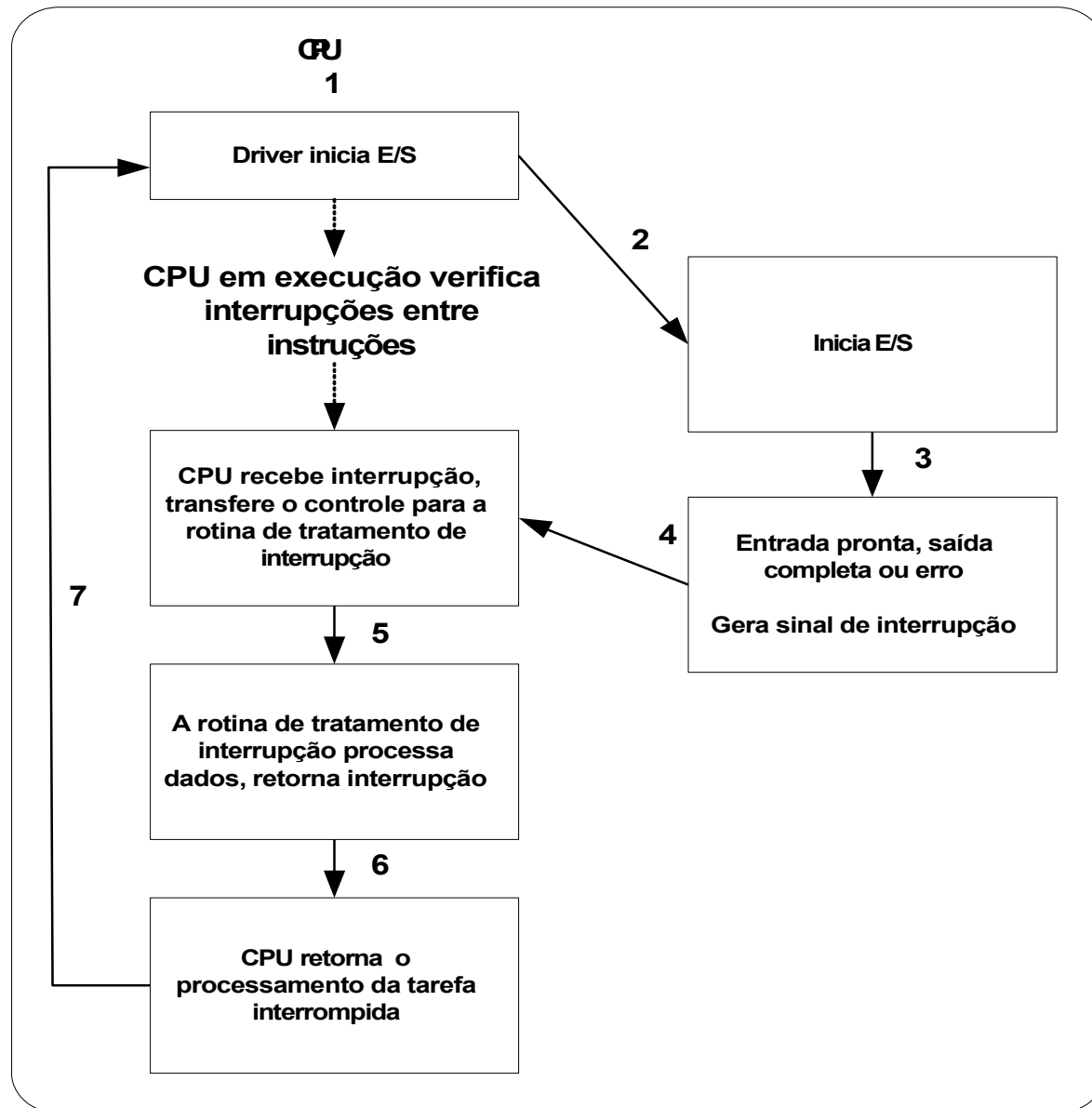
Dispositivos de E/S

Tratando Interrupções

- ★ **Ciclo de instrução com interrupção: CPU**
 - Busca; Decodificação e Execução
 - ★ Verifica se existe interrupção
 - ★ Se não → busca próxima instrução,...
 - ★ Se existe interrupção pendente:
 - Suspende a execução do programa;
 - Salva contexto;
 - Atualiza PC (*Program Counter*) → apontar para ISR (rotina de atendimento de interrupção);
 - Executa interrupção;
 - Recarrega contexto e continua processo interrompido;

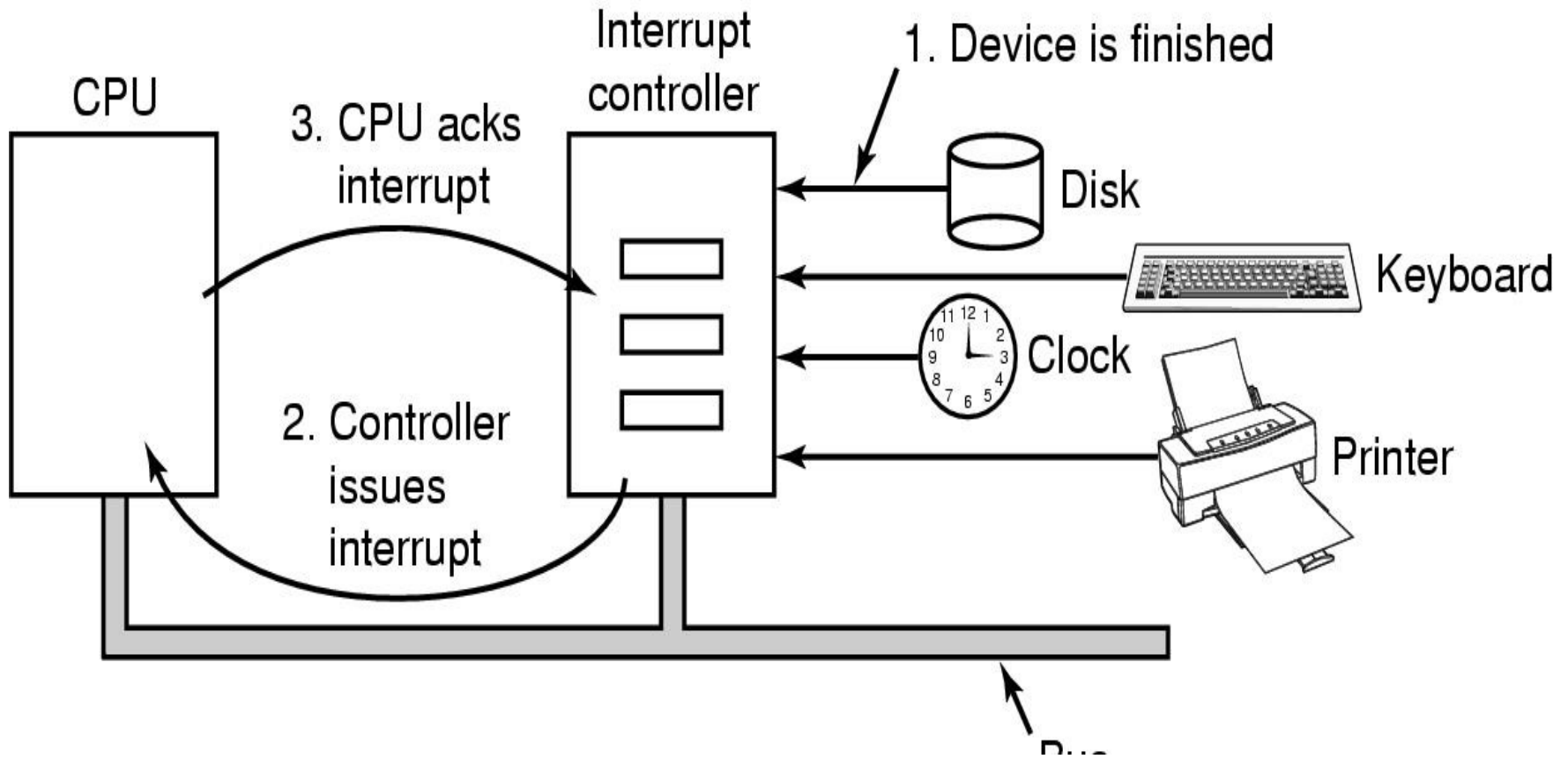
Dispositivos de E/S

Tratando Interrupções



Dispositivos de E/S interrupção

* Como uma interrupção ocorre:



Dispositivos de E/S

Princípios de Software

- ★ Organizar o software como uma série de camadas facilita a independência dos dispositivos:
- ★ Por que? Redes TCP/IP
 - Camadas mais baixas apresentam detalhes de hardware:
 - ★ Drivers e manipuladores de interrupção;
 - Camadas mais altas apresentam interface para o usuário:
 - ★ Aplicações de Usuário;
 - ★ Chamadas de Sistemas;
 - ★ Software Independente de E/S ou Subsistema de Kernel de E/S;

Dispositivos de E/S

Princípios de Software - Camadas

- ★ **Software de E/S no nível Usuário:**
 - Bibliotecas de E/S são utilizadas pelos programas dos usuários
 - ★ Chamadas ao sistema (*system calls*);

Dispositivos de E/S

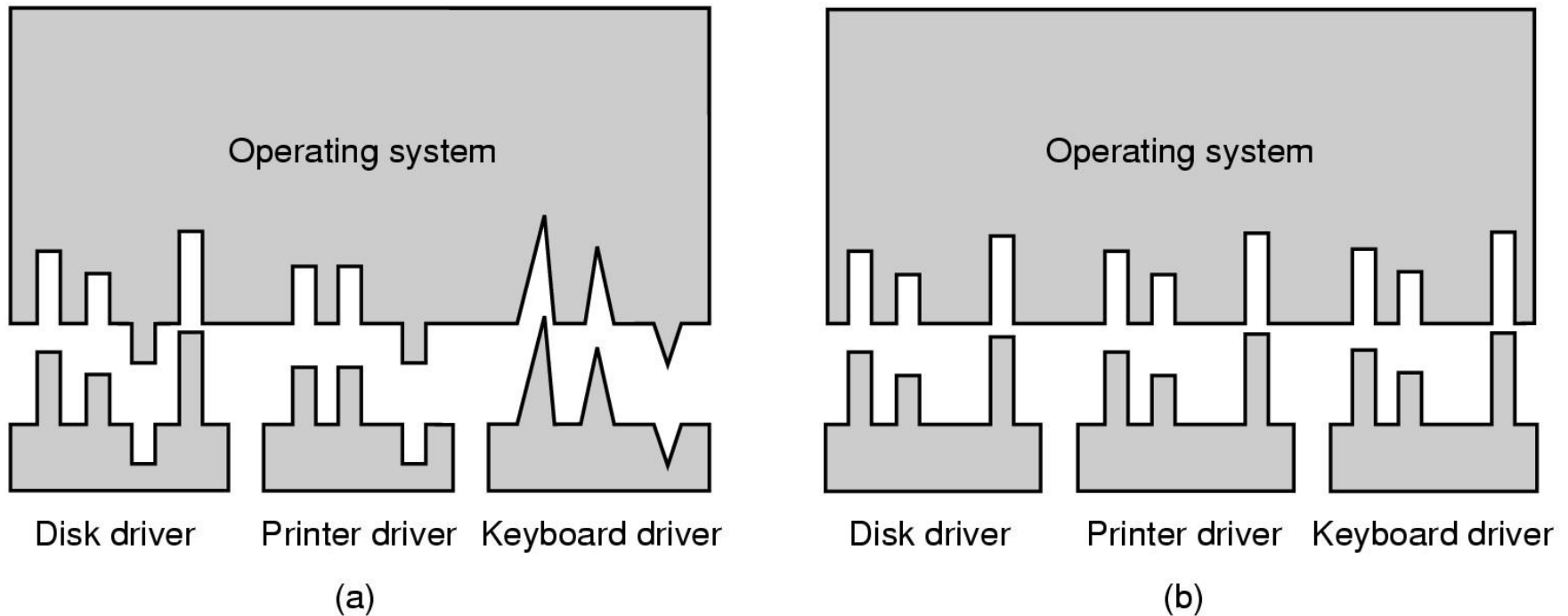
Princípios de Software - Camadas

★ **Software Independente de E/S:**

- Realizar as funções comuns a qualquer dispositivos;
- Prover uma interface uniforme para o usuário;
- Fazer o escalonamento de E/S;
- Atribuir um nome lógico a partir do qual o dispositivo é identificado;
 - ★ Ex.: UNIX → (/dev)
- Prover *buffering*: ajuste entre a velocidade e a quantidade de dados transferidos;
- *Cache* de dados: armazenar na memória um conjunto de dados freqüentemente acessados;

Dispositivos de E/S

Princípios de Software - Camadas



(a) Sem padrão de interface

(b) Com padrão de interface (uniforme)

e.g. `send()` e `receive()`

Princípios de Software - Camadas

* **Software Independente de E/S:**

- Reportar erros e proteger os dispositivos contra acessos indevidos :
 - * Programação: Ex.: tentar efetuar leitura de um dispositivo de saída (impressora, vídeo);
 - * E/S: Ex.: tentar imprimir em uma impressora desligada ou sem papel;
 - * Memória: escrita em endereços inválidos;
- Gerenciar alocação, uso e liberação dos dispositivos → acessos concorrentes;

Dispositivos de E/S

Princípios de Software

★ Software Independente de E/S:

– Transferência de dados:

- ★ **Síncrona** (bloqueante): requer bloqueio até que os dados estejam prontos para transferência;
- ★ **Assíncrona** (não-bloqueante): transferências acionadas por interrupções; mais comuns;

– Tipos de dispositivos:

- ★ **Compartilháveis**: podem ser utilizados por vários usuários ao mesmo tempo; Exemplo: disco rígido;
- ★ **Dedicados**: podem ser utilizados por apenas um usuário de cada vez; Exemplo: impressora, unidade de fita;

Dispositivos de E/S

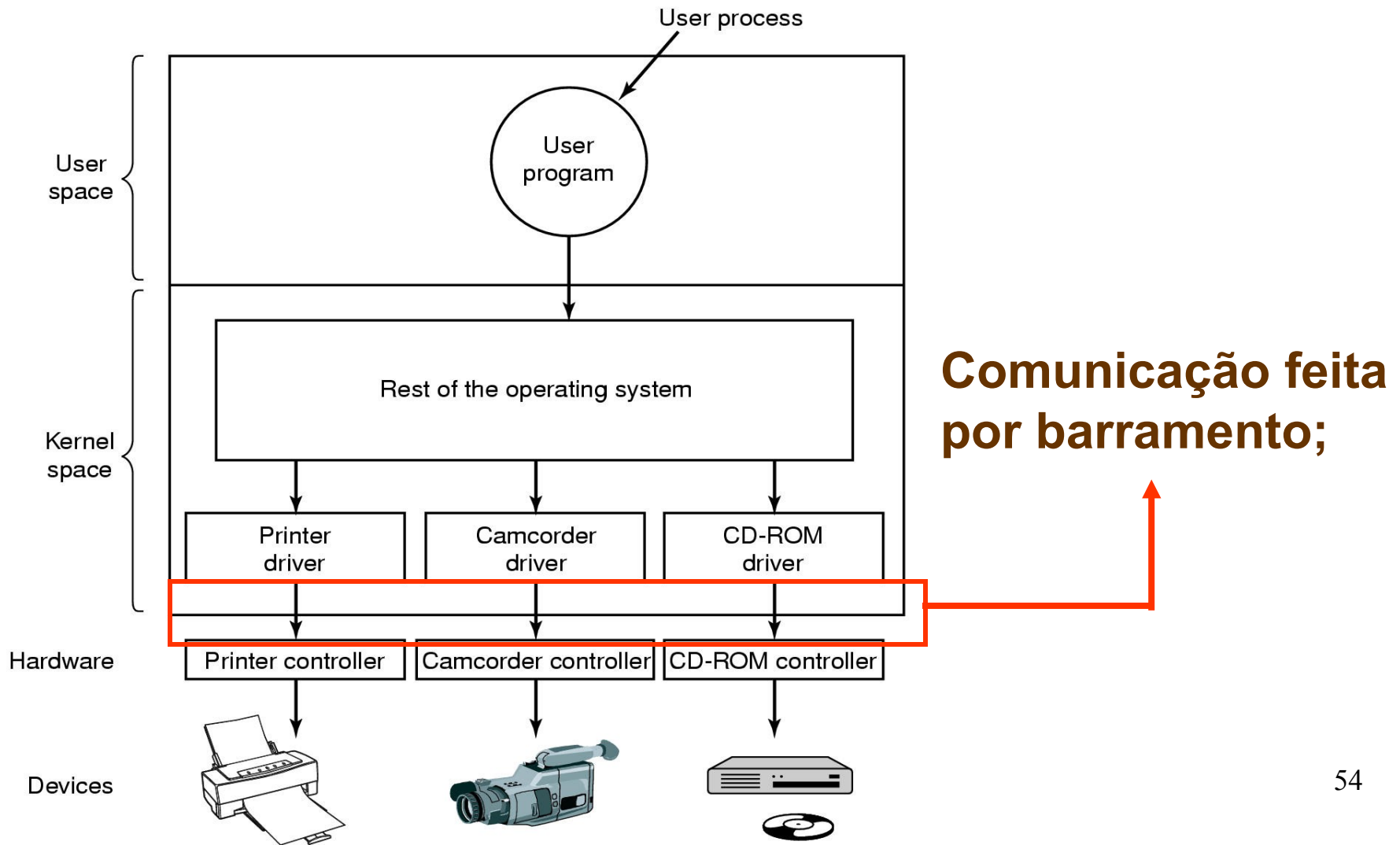
Princípios de Software - Camadas

★ **Drivers:**

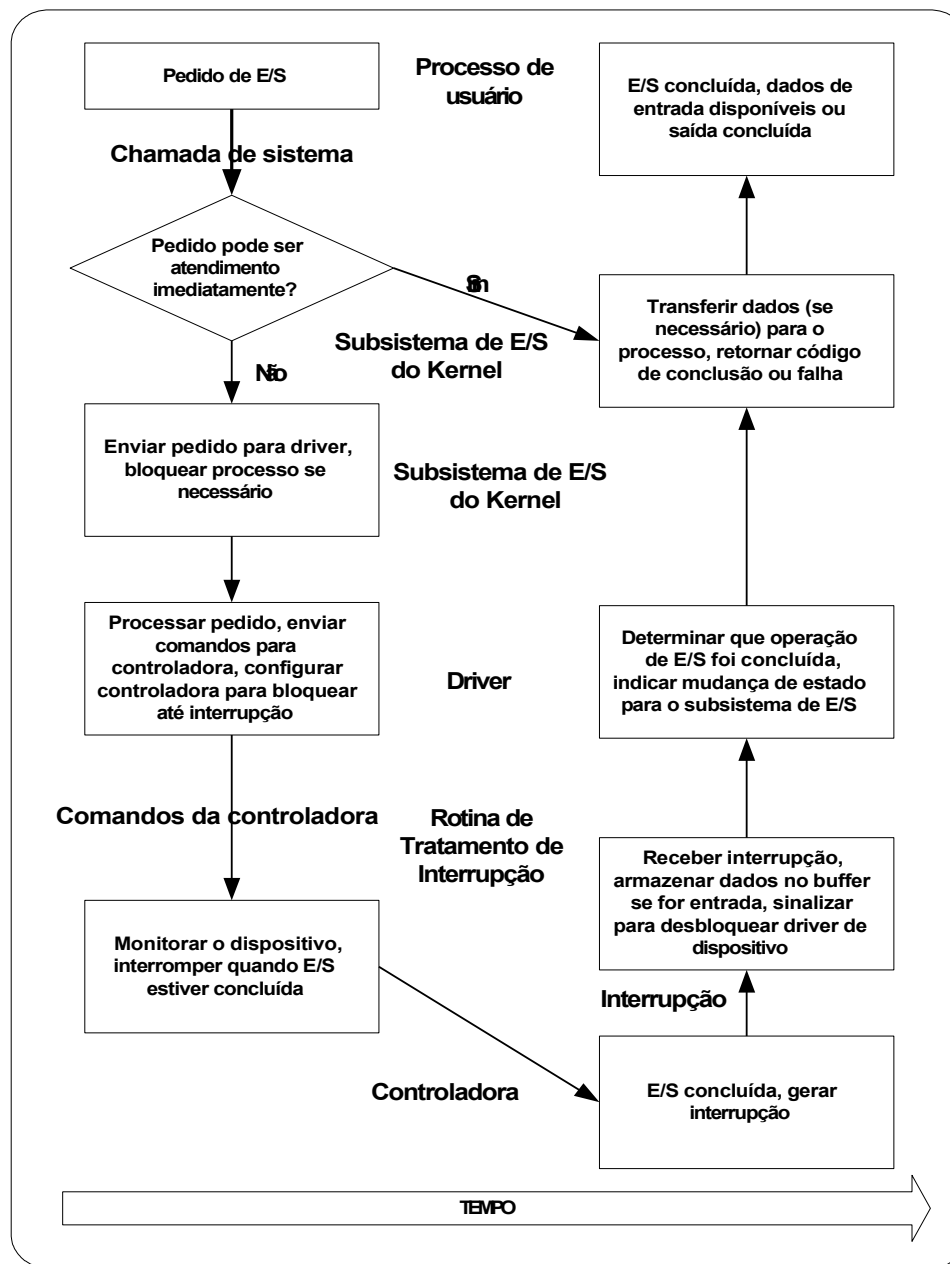
- São gerenciados pelo *kernel* do SO;
- Contêm todo o código dependente do dispositivo;
- Controlam o funcionamento dos dispositivos por meio de seqüência de comandos escritos/lidos nos/dos registradores da controladora;
- Dispositivos diferentes possuem *drivers* diferentes;
 - ★ Classes de dispositivos podem ter o mesmo *driver*;
- São dinamicamente carregados;
- *Drivers* defeituosos podem causar problemas no *kernel* do SO;

Dispositivos de E/S

Princípios de Software - Camadas



Dispositivos de E/S - Ciclo de E/S



Dispositivos de E/S - Ciclo de E/S

Seqüência da Figura anterior

- * Um processo emite uma chamada de sistema bloqueante (por exemplo: `read`) para um arquivo que já esteve aberto (`open`);
- * O código da chamada de sistema verifica os parâmetros. Se os parâmetros estiverem corretos e o arquivo já estiver no *buffer (cache)*, os dados retornam ao processo e a E/S é concluída;
- * Se os parâmetros estiverem corretos, mas o arquivo não estiver no *buffer*, a E/S precisa ser realizada;
 - E/S é escalonada;
 - Subsistema envia pedido para o *driver*;

Dispositivos de E/S - Ciclo de E/S

Seqüência da Figura anterior

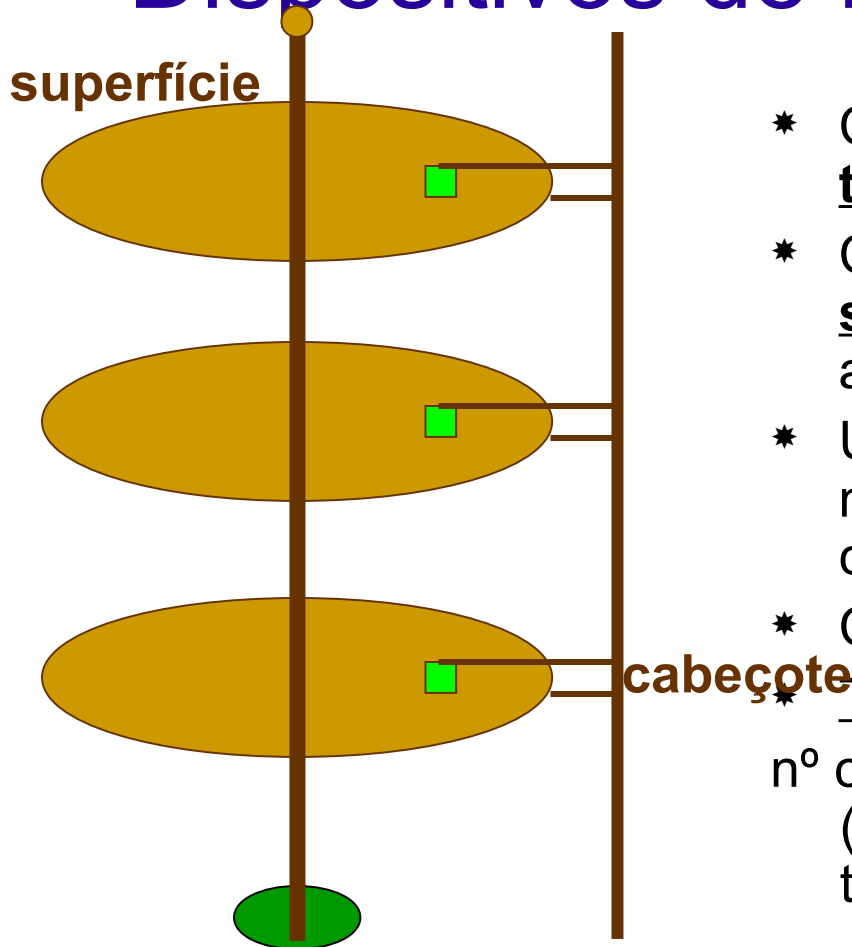
- ★ *Driver* aloca espaço de *buffer*, escalona E/S e envia comando para a controladora do dispositivo escrevendo nos seus registradores de controle;
 - *Driver* pode usar a DMA;
- ★ A controladora do dispositivo opera o hardware, ou seja, o dispositivo propriamente dito;
- ★ Após a conclusão da E/S, uma interrupção é gerada;
- ★ A rotina de tratamento de interrupções apropriada recebe a interrupção via vetor de interrupção, armazena os dados, sinaliza o *driver* e retorna da interrupção;

Dispositivos de E/S - Ciclo de E/S

Seqüência da Figura anterior

- ★ *Driver* recebe o sinal, determina qual pedido de E/S foi concluído, determina o status e sinaliza que o pedido está concluído;
- ★ Kernel transfere dados ou códigos de retorno para o espaço de endereçamento do processo que requisitou a E/S e move o processo da fila de bloqueados para a fila de prontos;
- ★ Quando o escalonador escalona o processo para a CPU, ele retoma a execução na conclusão da chamada ao sistema.

Dispositivos de E/S - Discos

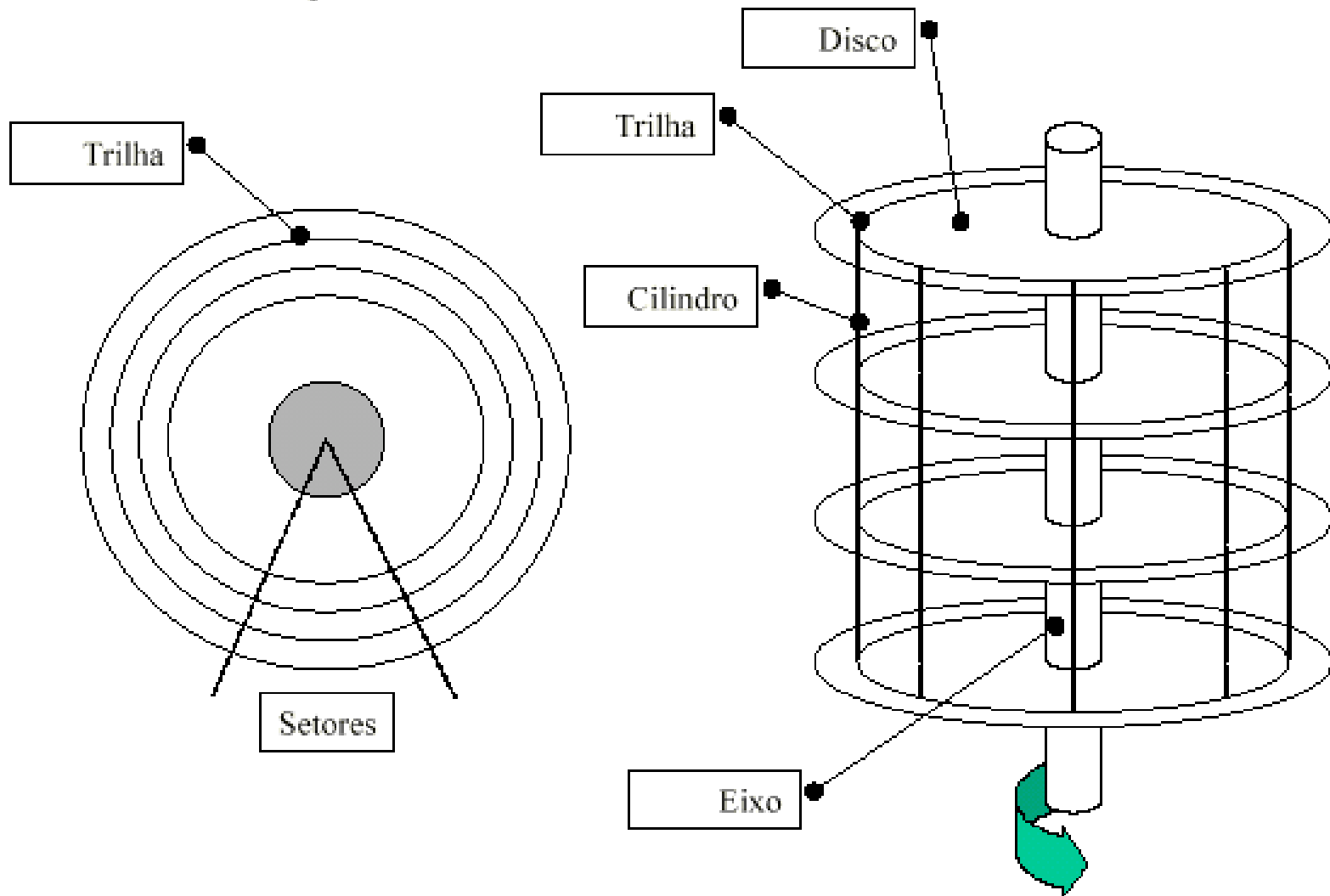


Disco Magnético

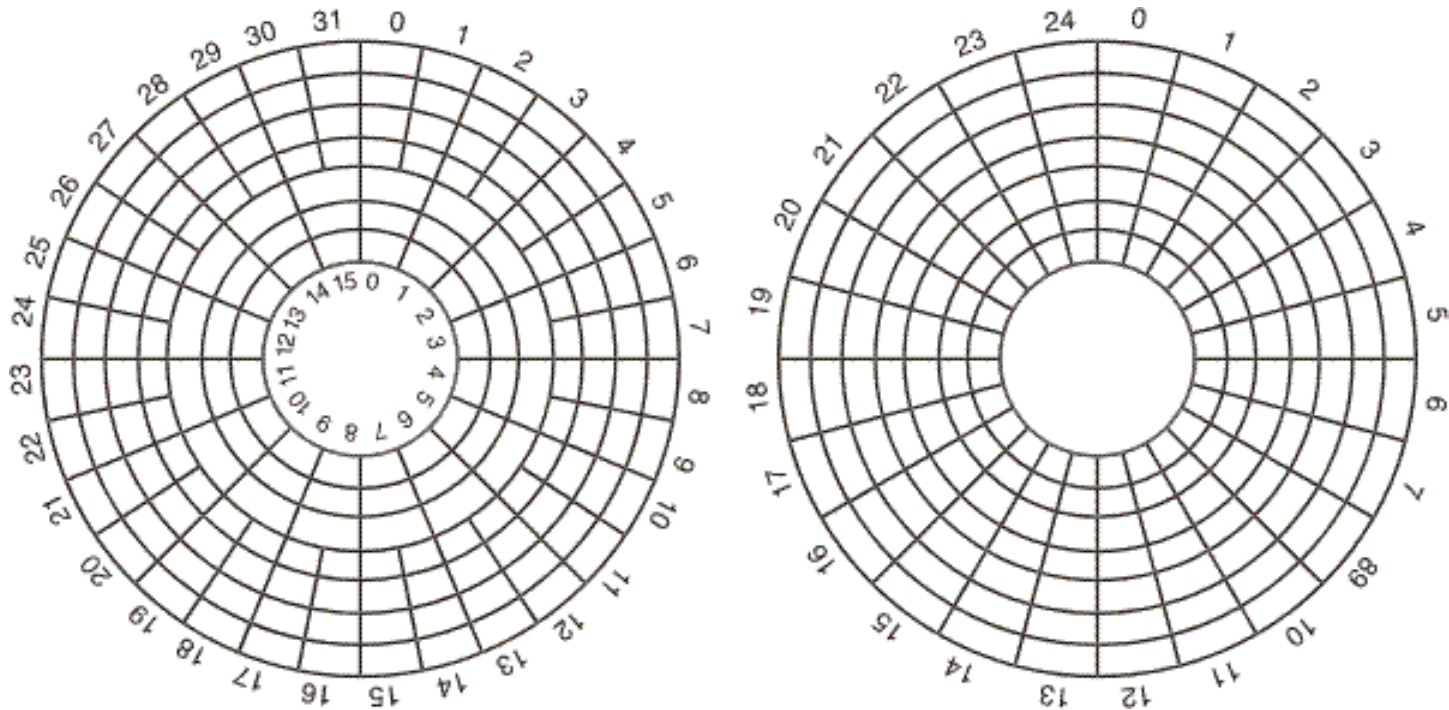
- * Cada superfície é dividida em **trilhas**;
- * Cada trilha é dividida em **setores** ou **blocos** (512 bytes a 32K);
- * Um conjunto de trilhas (com a mesma distância do eixo central) formam um **cilindro**;
- * Cabeças de leitura e gravação;
- * Tamanho do disco:
 n° cabeças (faces) x n° cilindros (trilhas) x n° setores x tamanho_setor;

Gerência de E/S

Organização dos Discos

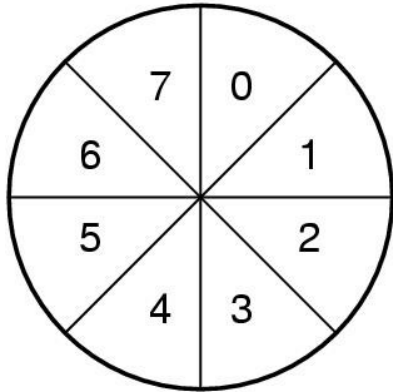


Hardware do Disco (2)

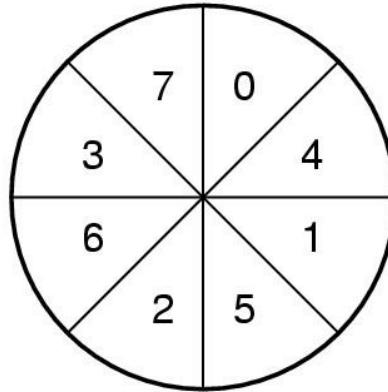


- * Geometria física de um disco com duas zonas
- * Uma possível geometria virtual para esse disco

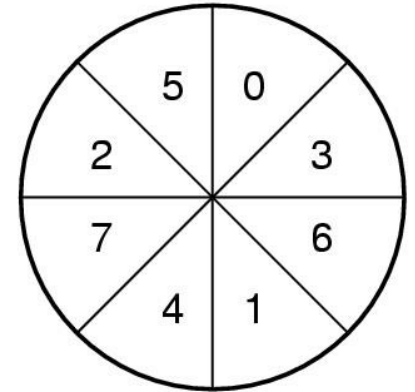
Formatação de Disco



(a)



(b)



(c)

- a) Sem entrelaçamento
- b) Entrelaçamento simples
- c) Entrelaçamento duplo

Dispositivos de E/S - Discos

* Discos Magnéticos:

– Grande evolução com relação

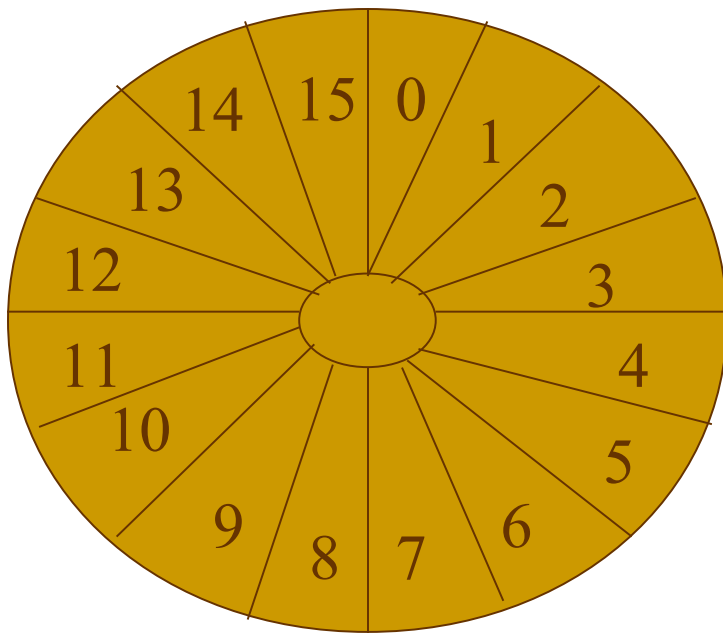
- * Velocidade de acesso (*seek*): tempo de deslocamento do cabeçote até o cilindro correspondente à trilha a ser acessada;
- * Transferências: tempo para transferência (leitura/escrita) dos dados;
- * Capacidade;
- * Preço;

Dispositivos de E/S - Discos

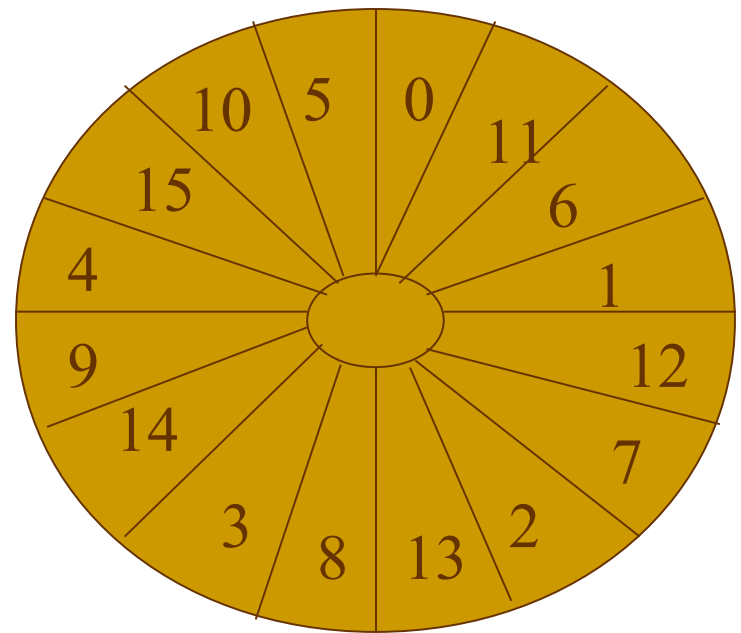
- * Técnica para reduzir o tempo de acesso: entrelaçamento (*interleaving*):
 - Setores são numerados com um espaço entre eles;
 - Entre o **setor K** e o **setor K+1** existem **n** (fator de entrelaçamento) setores;
 - * Número **n** depende da velocidade do processador, do barramento, da controladora e da velocidade de rotação do disco;

Dispositivos de E/S - Discos

Trilhas com 16 setores



Disco A
N = 0



Disco B
N = 2

Dispositivos de E/S - Discos

* *Drivers* de Disco:

– Fatores que influenciam tempo para leitura/escrita no disco:

- * Velocidade de acesso (*seek*) → tempo para o movimento do braço até o cilindro;
- * *Delay* de rotação (latência) → tempo para posicionar o setor na cabeça do disco;
- * Tempo da transferência dos dados;

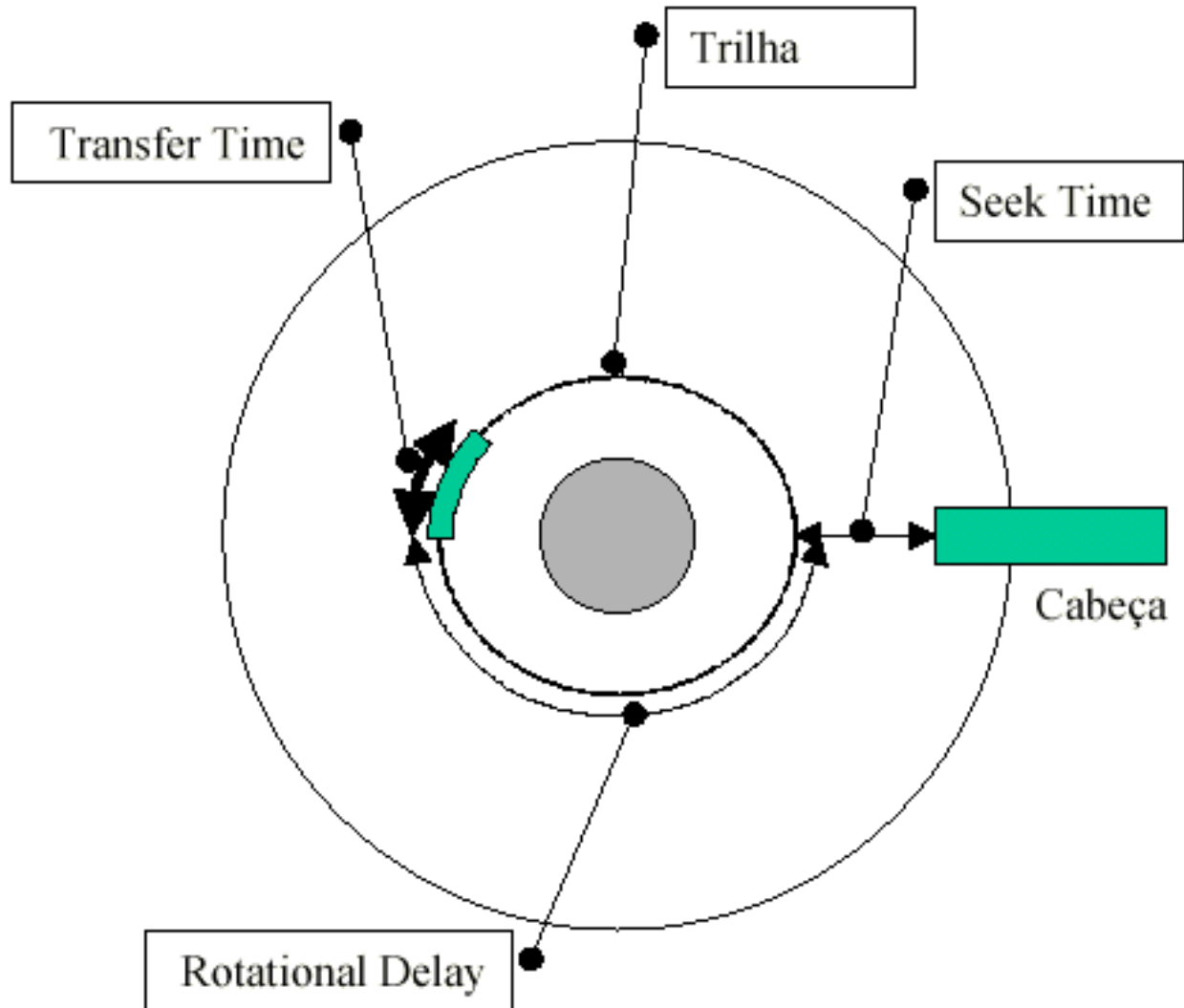
– Tempo de acesso:

$$* T_{seek} + T_{latência} + T_{transferência};$$

* Tempo necessário para o cabeçote se posicionar no setor de escrita/leitura;

Dispositivos de E/S

$$t_{\text{access}} = t_{\text{seek}} + t_{\text{rotationaldelay}} + t_{\text{transfer}}$$



Dispositivos de E/S – Discos

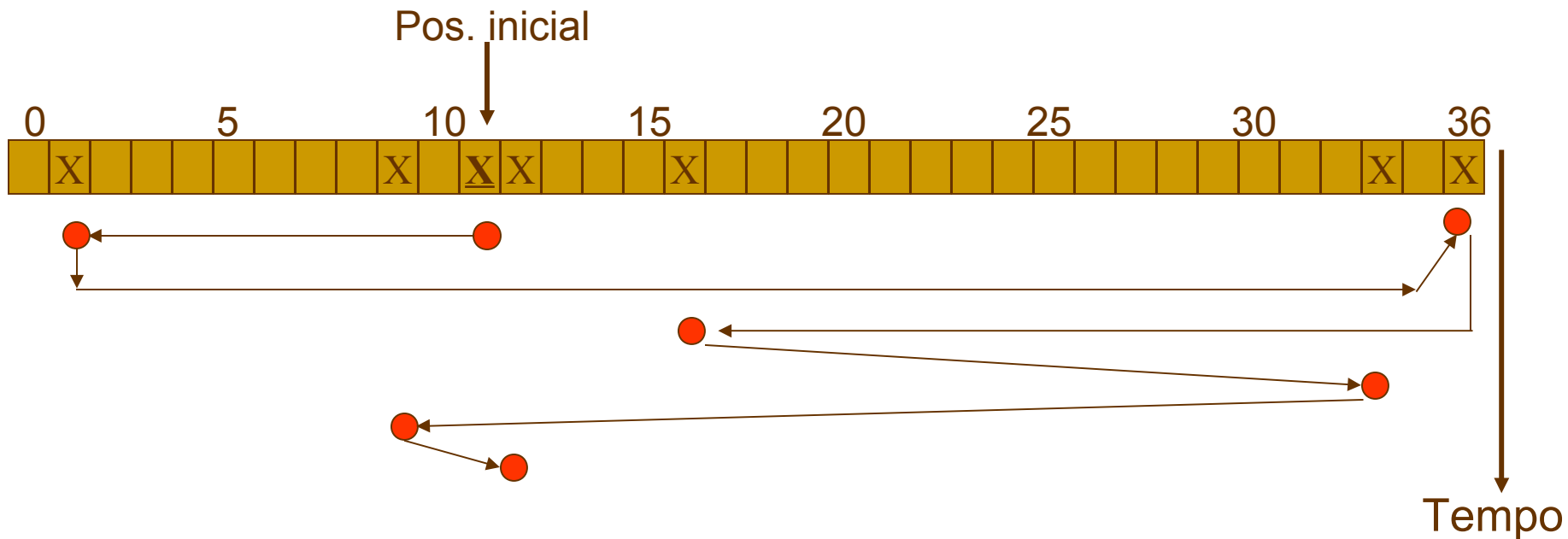
- ★ Algoritmos de escalonamento no disco:
 - FCFS (FIFO) → *First-Come First-Served*;
 - SSF → *Shortest Seek First*;
 - *Elevator* (também conhecido como SCAN);
- ★ Escolha do algoritmo depende do número e do tipo de pedidos;
- ★ *Driver* mantém uma lista encadeada com as requisições para cada cilindro;

Dispositivos de E/S - Discos

Disco com 37 cilindros;

Lendo bloco no cilindro 11;

Requisições: 1,36,16,34,9,12, nesta ordem



FCFS → atendimento: 1,36,16,34,9,12;

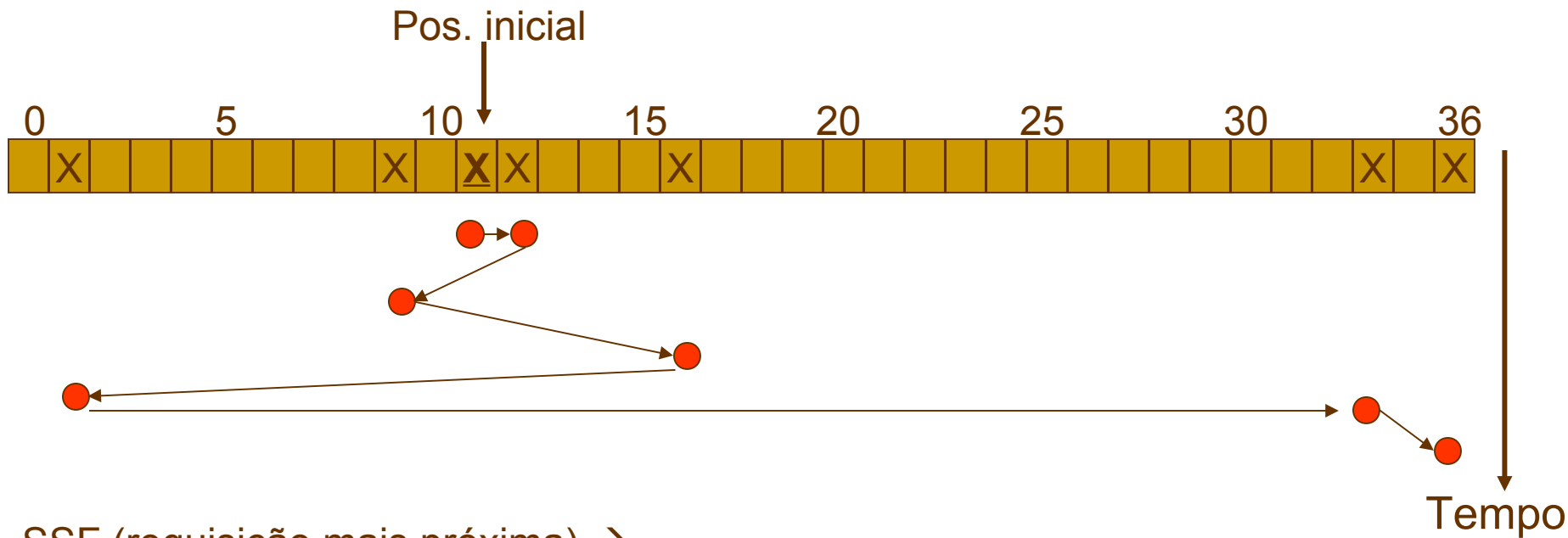
movimentos do braço (número de cilindros): 10,35,20,18,25,3 = 111;

Dispositivos de E/S - Discos

Disco com 37 cilindros;

Lendo bloco no cilindro 11;

Requisições: 1,36,16,34,9,12, nesta ordem



SSF (requisição mais próxima) →

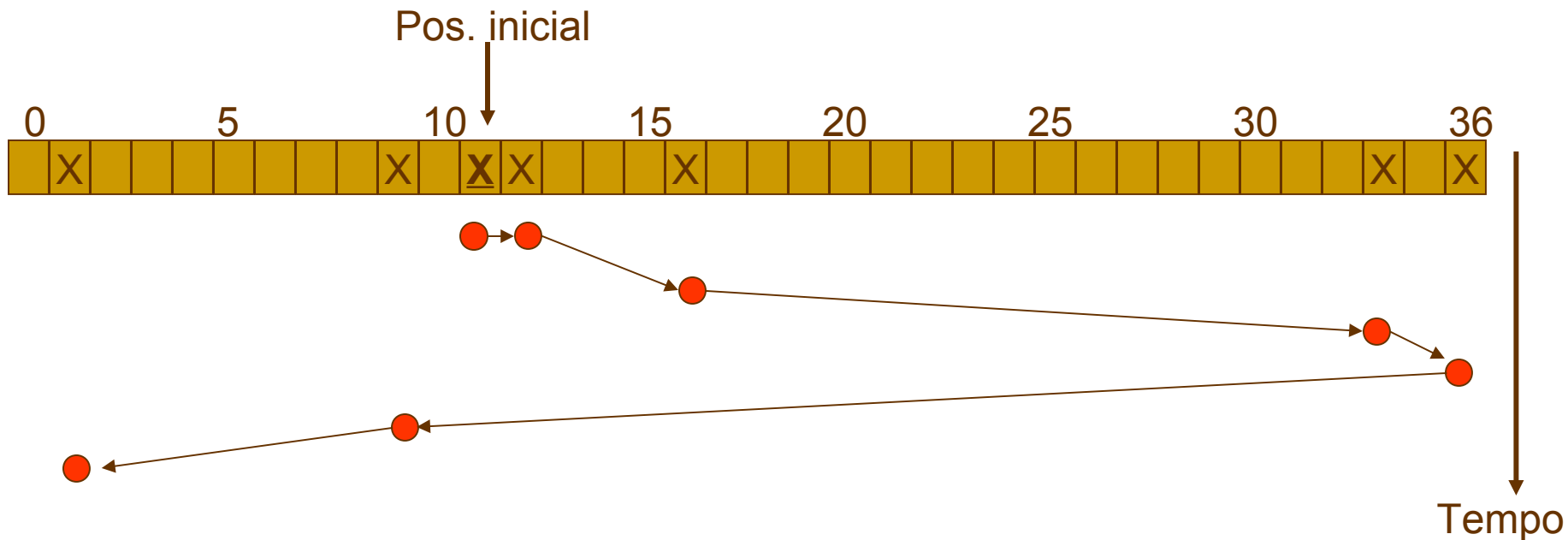
atendimento: 12,9,16,1,34,36;

movimentos do braço (número de cilindros): 1,3,7,15,33,2 = 61;

Dispositivos de E/S - Discos

Disco com 37 cilindros;
Lendo bloco no cilindro 11;
Requisições: 1,36,16,34,9,12, nesta ordem

Bit de direção corrente (driver):
Se *Up* → atende próxima requisição;
senão *Bit = Down*;
muda direção e atende requisição;



Elevator (requisições na mesma direção) →

atendimento: 12,16,34,36,9,1

movimentos do braço (número de cilindros): 1,4,18,2,27,8 = 60;

Dispositivos de E/S – Discos RAID

- ★ RAID (*Redundant Array of Independent Disks*) → armazena grandes quantidades de dados;
- ★ RAID combina diversos discos rígidos em uma estrutura lógica:
 - Aumentar a confiabilidade, capacidade e o desempenho dos discos;
 - Recuperação de dados → redundância dos dados;
 - Armazenamento simultâneo em vários discos permite que os dados fiquem protegidos contra falha (não simultânea) dos discos;
 - Performance de acesso, já que a leitura da informação é simultânea nos vários dispositivos;

Dispositivos de E/S – Discos RAID

- ★ Pode ser implementado por:
 - Hardware (controladora):
 - ★ Instalação de uma placa RAID no servidor, o subsistema RAID é implementado totalmente em hardware;
 - ★ Libera o processador para se dedicar exclusivamente a outras tarefas;
 - ★ A segurança dos dados aumenta no caso de problemas devido à checagem da informação na placa RAID antes da gravação;

Dispositivos de E/S – Discos RAID

- ★ Pode ser implementado por:
 - Software (sistema operacional)
 - ★ Menor desempenho no acesso ao disco;
 - ★ Oferece um menor custo e flexibilidade;
 - ★ Sobrecarrega o processador com leitura/escrita nos discos;
- ★ Para o SO existe um único disco;

Dispositivos de E/S – Discos RAID

- * A forma pela qual os dados são escritos e acessados define os níveis de RAID (até 9 níveis):
 - RAID 0:
 - * Também conhecido como *Stripping*;
 - * Arquivos são espalhados entre os discos em *stripes*;
 - * Melhora desempenho das operações de E/S;
 - * Todo o espaço do disco é utilizado para armazenamento;
 - * Utilizam mesma controladora (controladora RAID);
 - * Funciona bem para grandes solicitações (paralelismo entre os discos);
 - * Não funciona bem quando solicitam dados de um setor de cada vez

Dispositivos de E/S – Discos RAID

- * A forma pela qual os dados são escritos e acessados define os níveis de RAID (até 9 níveis):
 - RAID 0:
 - * Também conhecido como *Stripping*;
 - * Arquivos são espalhados entre os discos em *stripes*;
 - * Melhora desempenho das operações de E/S;
 - * Todo o espaço do disco é utilizado para armazenamento;
 - * Utilizam mesma controladora (controladora RAID);
 - * Funciona bem para grandes solicitações (paralelismo entre os discos);
 - * Não funciona bem quando solicitam dados de um setor de cada vez

Dispositivos de E/S – Discos RAID

– RAID 1:

- * Conhecido como espelhamento (*mirroring*);
- * Operações de escrita no disco primário são replicadas em um disco secundário;
- * Pode ter controladoras diferentes;
- * Desvantagem: espaço físico em dobro (alto custo);
- * Transações *on-line* (tolerância a falhas);

– RAID 10:

- * Combinação dos RAID 1 e RAID 0;

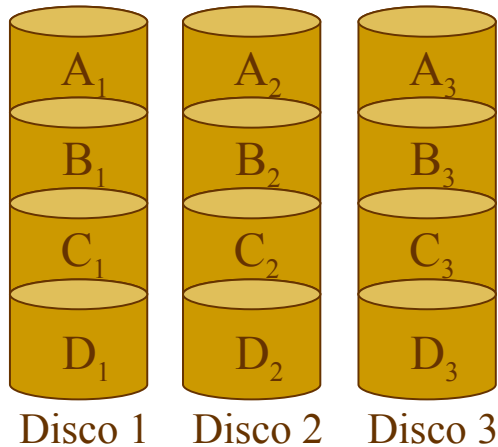
Dispositivos de E/S – Discos RAID

- RAID 2/3/4:
 - * Dados são armazenados em discos diferentes com paridade (permite reconstruir dados perdidos); *Stripes*;
 - * Paridade é mantida em um disco apenas;
 - * Diferença básica: como a paridade é calculada (na transferência):
 - *Raid 2 - Hamming ECC (error-correcting codes)– nível de bit*;
 - *Raid 3 - XOR ECC - nível de byte ou bit*;
 - *Raid 4 – XOR ECC - nível de bloco*;
- RAID 5:
 - * *Stripes*;
 - * Paridade *XOR ECC* distribuída - nível de bloco;
 - * Paridade está distribuída nos discos;
- RAID 6:
 - * *Stripes*;
 - * *Raid 5 com dois discos de paridade*;

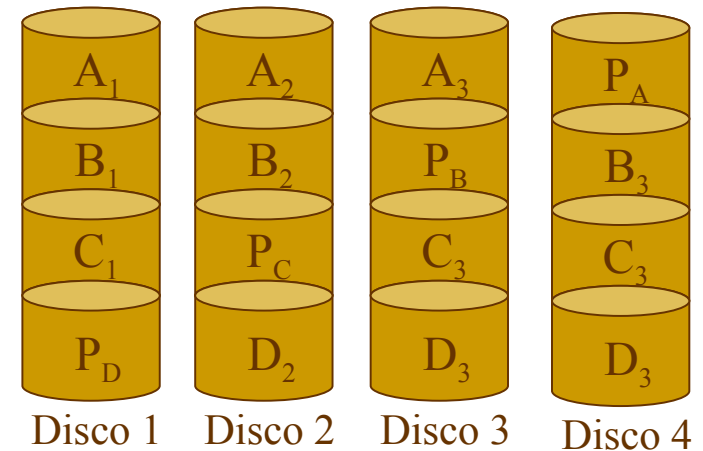
Dispositivos de E/S – Discos

RAID mais comuns

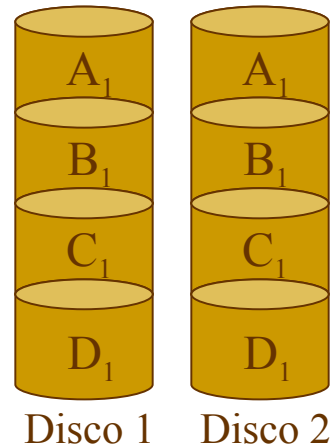
RAID 0

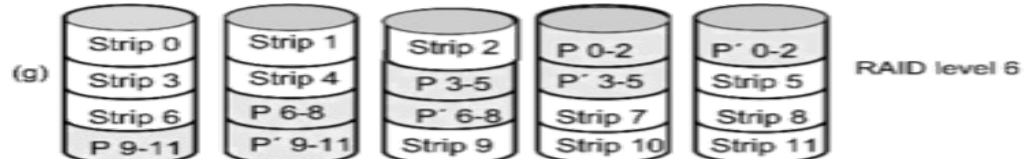
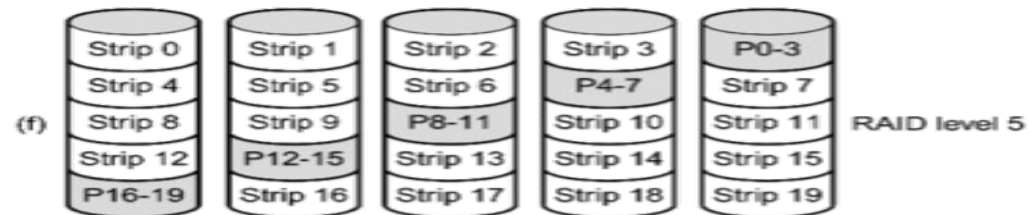
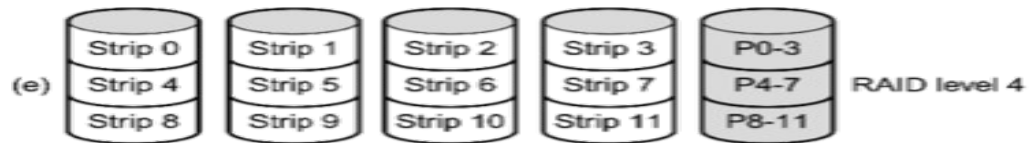
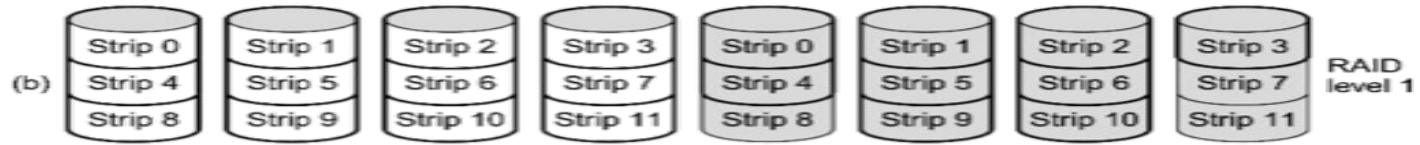
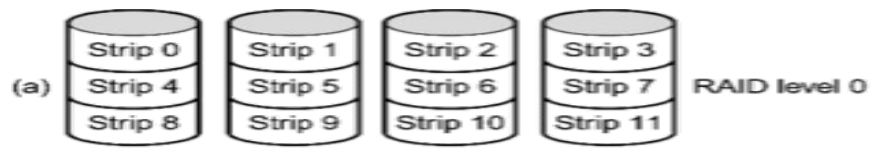


RAID 5



RAID 1





Clocks (*Timers*)

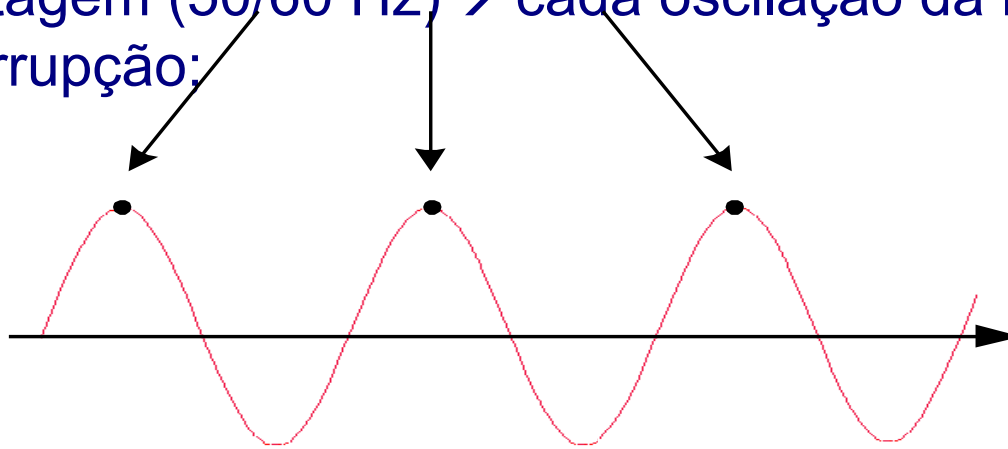
- * Dois tipos de relógios:
 - hardware (*clock hardware*) e software (*clock driver*);
- * *Clock Hardware*:
 - Dispositivo que gera pulsos síncronos;
 - Localizados na CPU ou na placa-mãe;
 - Sinal utilizado para a execução de instruções;
 - Presente em qualquer sistema multiprogramado;
 - Fundamental para ambientes *TimeSharing*;
 - Frequência de *clock*
 - * Número de vezes que o pulso se repete por segundo (Hz);

Clocks – Tipos Hardware

* Hardware

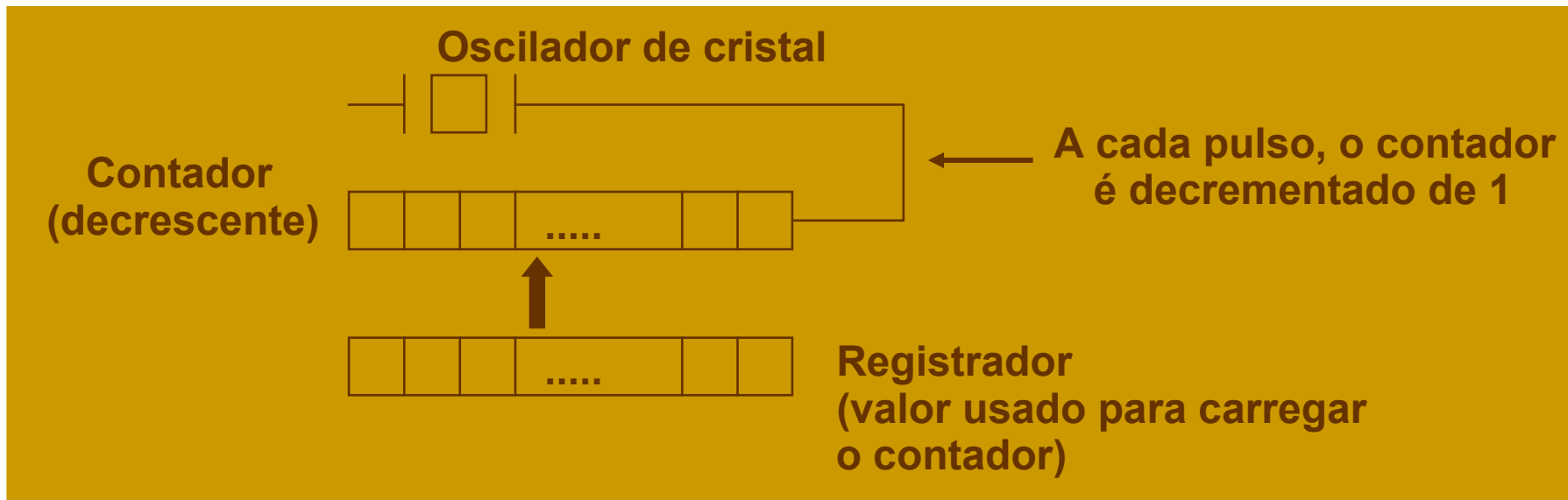
– Dois tipos:

- * Básico: usa o sinal da rede elétrica (110/220 V) para fazer contagem (50/60 Hz) → cada oscilação da rede é uma interrupção:



Clocks – Tipos Hardware

- * Com 03 componentes → oscilador de cristal, contador e registrador;
 - Programável;



Clocks – Tipos Hardware

★ Esquema:

- Contador recebe o valor armazenado no registrador;
- A cada pulso do oscilador, o contador é decrementado de uma unidade;
- Quando o contador zera, é gerada uma interrupção de *clock* (interrupção da CPU);
- Precisão;

Clocks – Tipos Hardware

- ★ Relógios programáveis podem operar de diversos modos:
 - *One-shot mode*
 - ★ Ao ser iniciado, o relógio copia o valor contido no registrador, e decrementa o contador a cada pulso do cristal;
 - ★ Quando o contador chega a zero, uma interrupção ocorre;
 - ★ Recomeça por intervenção de software;

Clocks – Tipos Hardware

- *Square-wave mode*
 - ★ Repete o ciclo automaticamente, sem intervenção de software;
 - ★ Após atingir 0 e causar a interrupção, repete-se todo o processo
- ★ As periódicas interrupções geradas pela CPU são chamadas de clock ticks (pulsos do relógio);

Clocks – Tipos Software

- ★ Hardware → gera interrupções em intervalos conhecidos (*clock ticks*);
- ★ Tudo o mais é feito por Software: *clock driver*;
- ★ Funções do *clock driver*:
 - Manter a hora do dia;
 - Evitar que processos executem por mais tempo que o permitido;
 - Supervisionar o uso da CPU;
 - Cuidar da chamada de sistema `alarm`;
 - Fazer monitoração e estatísticas;
 - Prover temporizadores “guardiões” para os dispositivos de E/S;

Software

Manter a Hora do Dia

- ★ Hora e data correntes:
 - Checa a *CMOS*;
 - ★ Uso de baterias para não perder as informações
 - Pergunta ao usuário;
 - Checa pela rede em algum *host* remoto;
- ★ Número de *clock ticks*:
 - Desde às 12 horas do dia 1º de janeiro de 1970 no UNIX;
 - Desde o dia 1º de janeiro de 1980 no Windows;

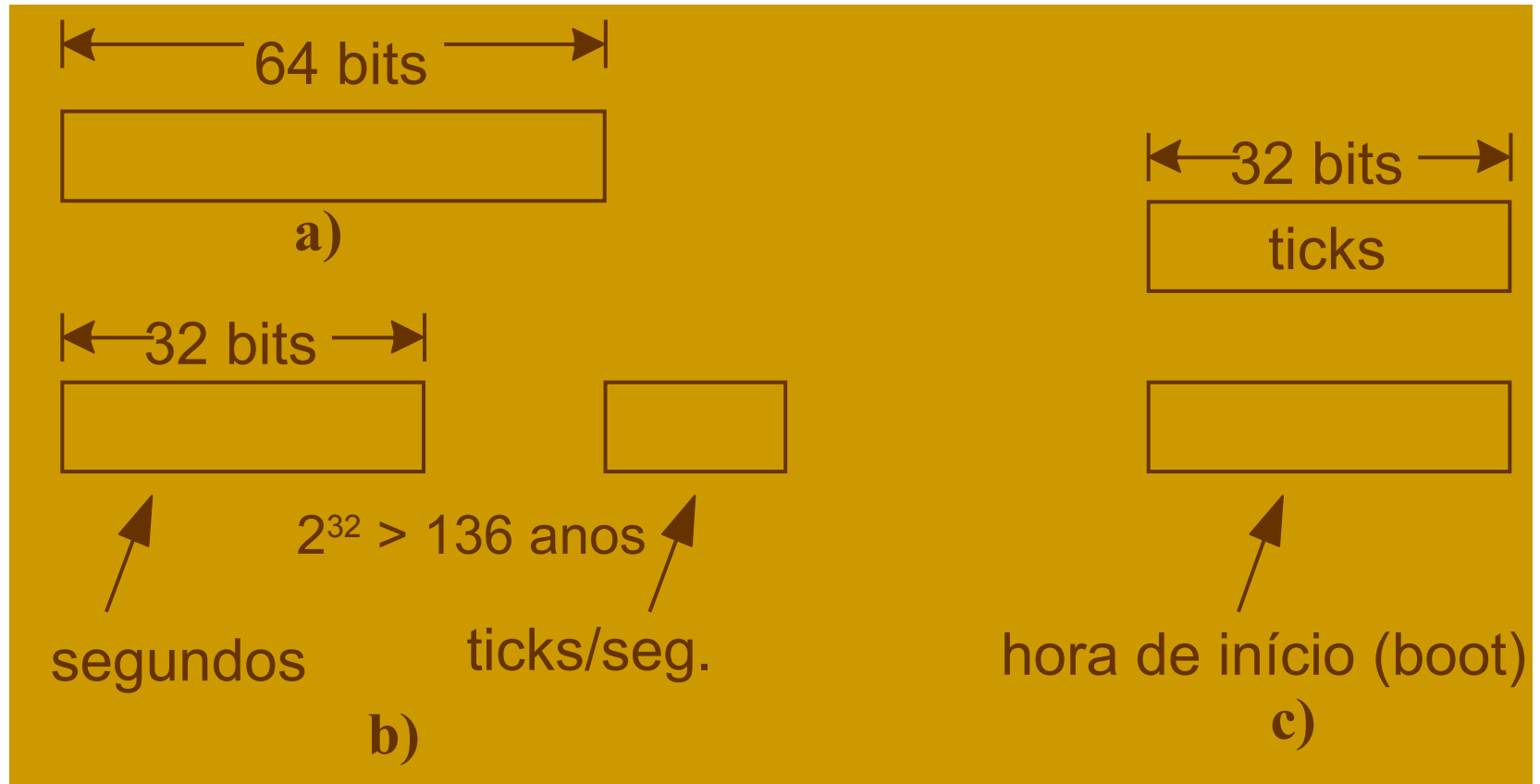
Software

Manter a Hora do Dia

- ★ Incrementar contador a cada *tick*;
- ★ Três abordagens:
 - a) Contador com 64 bits → alto custo;
 - b) Contar em segundos → *ticks/seg*;
 - c) *Ticks* relativos à hora que o sistema foi iniciado (lê o relógio e inicia a contagem);

Software

Manter a Hora do Dia



Software

Controlar duração da Execução dos Processos

- * Execução inicia → escalonador inicia contador → número de *ticks* do *quantum*;
- * Contador é decrementado a cada *tick*;
- * Contador = 0 → hora de acionar escalonador (que pode trocar o processo);

Software

Supervisão do uso da CPU

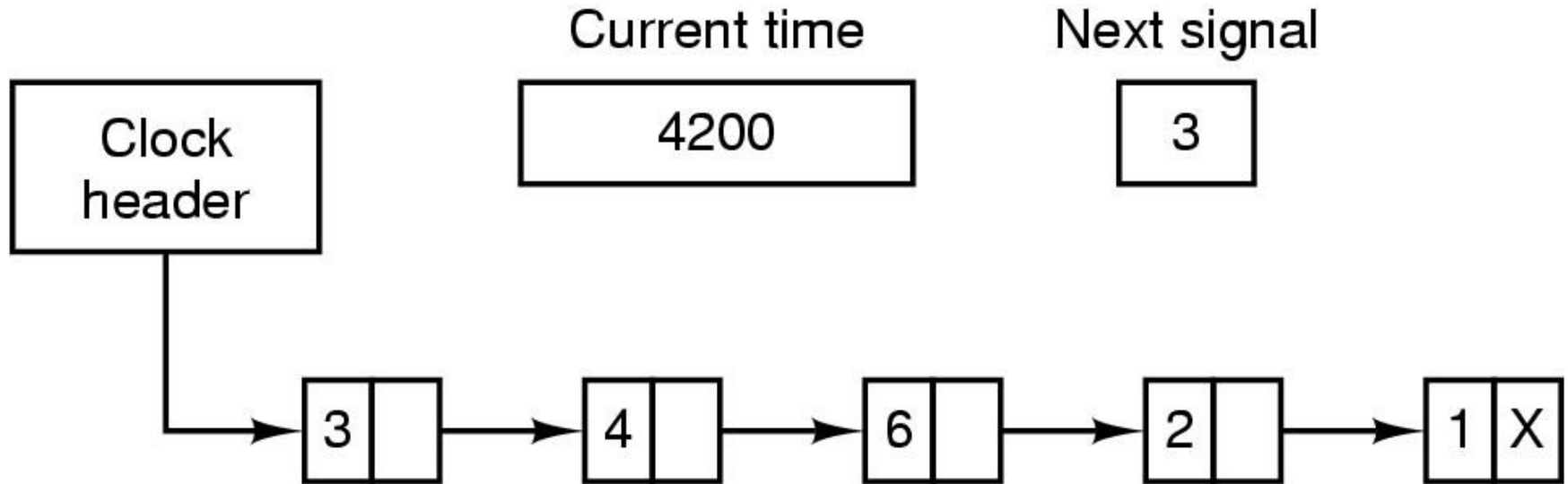
- ★ Quanto tempo o processo já foi executado?
 - Processo inicia → novo *clock* (segundo relógio) é iniciado;
 - Processo é parado → *clock* é lido (ler o tempo atual);
 - Durante interrupções → valor do *clock* é salvo e restaurado depois;
- ★ Possível usar a tabela de processos → variável global armazena o tempo (em *ticks*);

Software

Alarmes (Avisos)

- * Processos podem requerer “avisos” de tempos em tempos;
- * Avisos podem ser: um sinal, uma interrupção ou uma mensagem;
 - Exemplo:
 - * redes de computadores → pacotes não recebidos devem ser retransmitidos;
- * Uma lista encadeada com os tempos dos alarmes pendentes é mantida:
 - Simulação de vários relógios virtuais em um único relógio físico;

Software Alarmes (Avisos)



Simulação de vários relógios virtuais
com um único relógio físico

Tempos dos alarmes: 4203, 4207, 4213, 4215, 4216;

Software

Temporizadores Guardiões (watchdog)

- ★ Esperar por um certo tempo e realizar uma tarefa:
 - ! ! t → registrador (contador);
 - Quando contador zera → procedimento é executado;
 - O temporizador watchdog verifica se o sistema está em funcionamento e reinicia se não estiver
- ★ Onde usar?
 - Exemplo:
 - ★ acionador de disco flexível e/ou CD: somente quando o disco está em rotação na velocidade ideal é que as operações de E/S podem ser iniciadas;

Clocks – Tipos Software

- ★ Tarefas básicas do *driver* de relógio (*clock driver*) durante uma interrupção:
 - Incrementar o tempo real;
 - Decrementar o *quantum* e comparar com 0 (zero);
 - Contabilizar o uso da CPU;
 - Decrementar o contador de alarme;
 - Gerenciar o tempo de acionamento de dispositivos de E/S;

Perguntas?

- ★ Ler o Capítulo 5