

Gerenciamento de dispositivos de entrada e saída

Prof. Marcos Ribeiro Quinet de Andrade
Instituto de Ciência e Tecnologia - ICT
Universidade Federal Fluminense - UFF



Dispositivos de Entrada e Saída

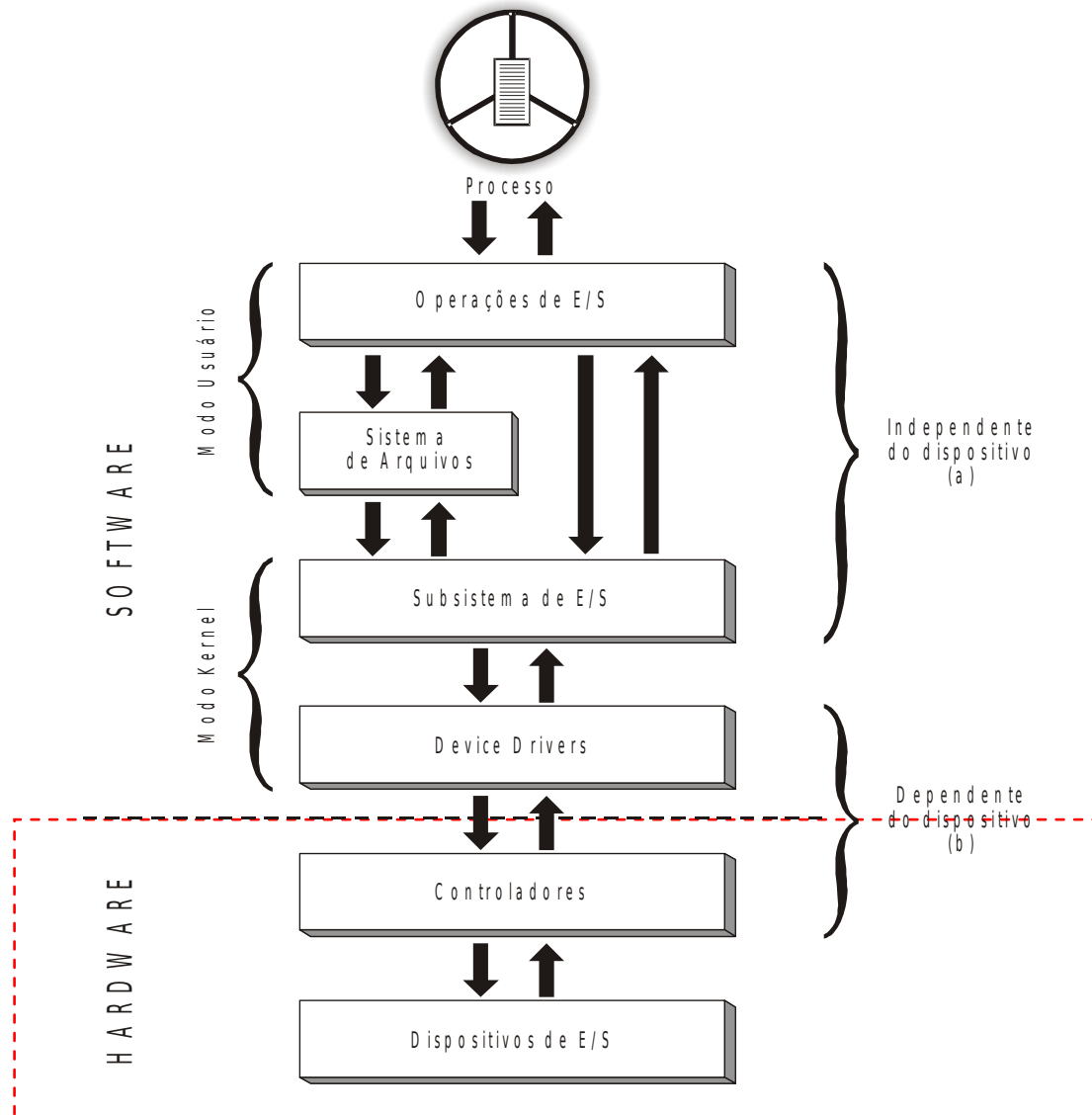
- O S.O. pode atuar de duas maneiras distintas (já abordadas no capítulo 1):
 - Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário;
 - Como gerenciador de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador;



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 (deve ser a mesma para todos os dispositivos, **o que nem sempre é possível!**);
- Princípios divididos em camadas:
 - *Hardware*;
 - *Software*;

Dispositivos de Entrada e Saída





Dispositivos de E/S

Princípios de Hardware

- Podem ser divididos em duas categorias:
 - Dispositivos baseados em bloco: a informação é armazenada em blocos de tamanho fixo, cada um com um endereço próprio; todas as unidades de transferência são em unidades de um ou mais blocos inteiros (consecutivos)
 - Tamanho varia entre 512 bytes e 65.536 bytes;
 - Permitem leitura e escrita independentemente de outros dispositivos;
 - Permitem operações de posicionamento do meio de armazenamento de dados;
 - Ex.: discos rígidos, memórias *flash*, mídias ópticas;



Dispositivos de E/S

Princípios de Hardware

- Dispositivos baseados em caracter: envia ou aceita uma sequência de caracteres, sem se importar com a estrutura de blocos; informação não é endereçável e não possuem operações de posicionamento;
- Ex.: impressoras, interfaces de rede (placas de rede), mouses, placas de som;



Dispositivos de E/S

Princípios de Hardware

- A classificação não é perfeita, pois alguns dispositivos não se encaixam em nenhuma das duas categorias:
 - *Clocks*: provocam interrupções em intervalos definidos; não são endereçáveis por blocos nem enviam ou recebem fluxos de caracteres;
 - Telas de toque.
- A classificação auxilia na obtenção de independência ao dispositivo;
 - Parte dependente está a cargo dos *drivers*, que são os *softwares* que controlam o acionamento dos dispositivos;

Dispositivos de E/S

Princípios de Hardware

- Os dispositivos de E/S podem apresentar uma grande variedade de velocidade; fica a cargo do software operar corretamente com taxas de transferências de dados de diferentes ordens de magnitude

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Scanner at 300 dpi	1 MB/sec
Digital camcorder	3.5 MB/sec
4x Blu-ray disc	18 MB/sec
802.11n Wireless	37.5 MB/sec
USB 2.0	60 MB/sec
FireWire 800	100 MB/sec
Gigabit Ethernet	125 MB/sec
SATA 3 disk drive	600 MB/sec
USB 3.0	625 MB/sec
SCSI Ultra 5 bus	640 MB/sec
Single-lane PCIe 3.0 bus	985 MB/sec
Thunderbolt 2 bus	2.5 GB/sec
SONET OC-768 network	5 GB/sec



Dispositivos de E/S

Controladores de dispositivos

- Dispositivos de E/S possuem basicamente dois componentes:
 - Mecânico → o dispositivo propriamente dito;
 - Eletrônico → controladores ou adaptadores (placas);
- Um controlador podem lidar com múltiplos dispositivos idênticos (por exemplo, múltiplas portas USB)
- O dispositivo (periférico) e a controladora se comunicam por meio de uma **interface**:
 - Serial ou paralela;
 - Barramentos: IDE, ISA, SCSI, AGP, USB, PCI, etc.



Dispositivos de E/S

Controladores de dispositivos

- O controlador e o dispositivo hardware operam em um nível muito baixo, enquanto o S.O. trabalha com dados em um nível mais alto
 - Por exemplo, o controlador converte um fluxo serial de bits de um disco em blocos de bytes, montados em um *buffer* e com a checagem de integridade de dados verificada;
 - Só então o bloco é liberado para ser copiado para a memória principal.

Mapeamento de E/S na Memória

- Cada controladora possui um conjunto de registradores de controle, que são utilizados na comunicação com a UCP
- Através destes registradores, o S.O. pode ordenar ao dispositivo realizar certas tarefas, como:
 - enviar ou receber dados;
 - checar o estado do dispositivo;
 - ligar e desligar;
 - etc.
- Além dos registradores, alguns dispositivos possuem um *buffer* de dados:
 - Ex.: placa de vídeo (buffer RAM); algumas impressoras;



Mapeamento de E/S na Memória

- O S.O. gerencia, utilizando os drivers, os dispositivos de E/S, através de operações de leitura e/ou escrita nos buffers e registradores
 - Comunicação em baixo nível - instruções em Assembler;
 - Enviar comandos para os dispositivos;
 - Saber o estado dos dispositivos;
- Duas abordagens podem ser empregadas para que esta comunicação seja feita:

Mapeamento de E/S na Memória

- **Porta**: cada registrador de controle possui um número de **porta de E/S** (geralmente, de 8 ou 16 bits);
 - Instrução em *Assembler* para acessar os registradores;
 - Espaço de endereçamento diferente para a memória e para os dispositivos de E/S;
 - Este **primeiro método** foi introduzido pelos *Mainframes* IBM;
 - O conjunto de portas de E/S forma o **espaço de endereçamento de portas de E/S**; apenas o SO pode usá-las, ficando protegidas contra acessos indevidos dos programas de usuários

Mapeamento de E/S na Memória

- Exemplos:
 - IN REG, PORT → Permite a UCP ler no registrador de controle PORT e armazenar o resultado no registrador REG da UCP;
 - OUT REG, PORT → Permite a UCP escrever o conteúdo de REG para o registrador de controle PORT.

Por exemplo:

IN, R0, 4 # lê o conteúdo da porta de E/S 4 e
 # armazena em R0

MOV R0, 4 # lê o conteúdo da palavra de memória
 # 4 e armazena em R0

- Observe que os valores '4' referem-se a espaços de endereçamento diferentes e não relacionados

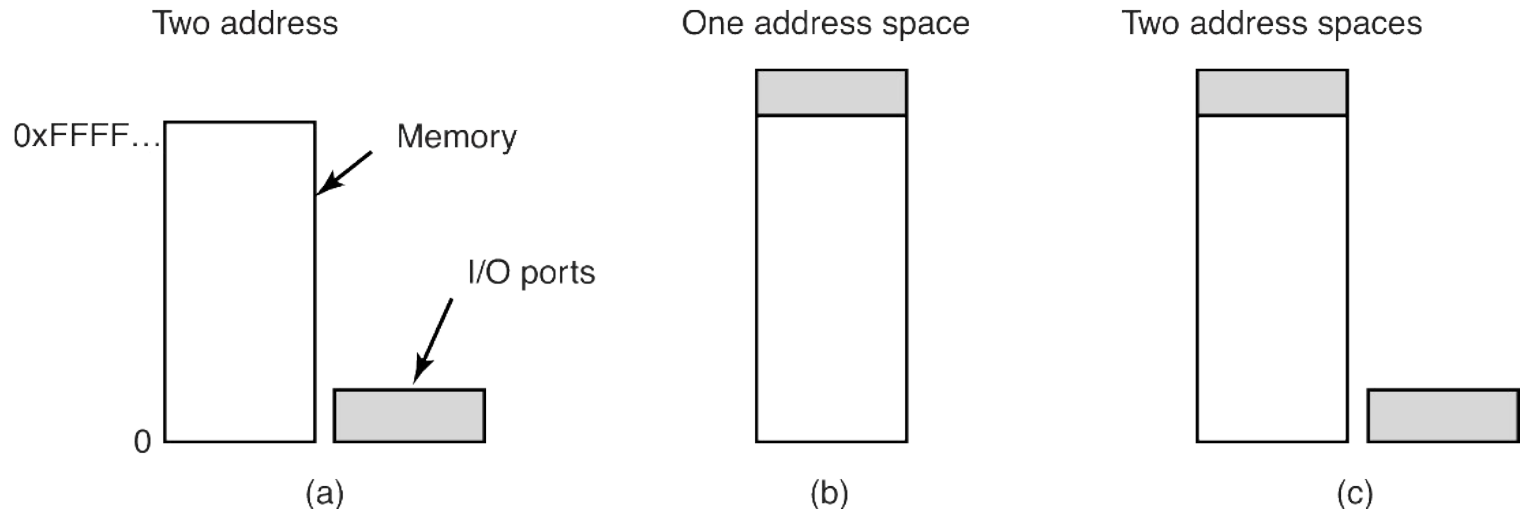
Mapeamento de E/S na Memória

- O **segundo método** de comunicação com os registradores de controle foi apresentado com o surgimento do PDP-11:
- **Memory-mapped (mapeada na memória)**: mapear os registradores de controle em espaços de memória;
 - Cada registrador é mapeado em um endereço da memória;
 - Em geral, os endereços estão no topo da memória protegidos em endereços não utilizados por processos;
 - Uso de linguagem de alto nível, já que registradores são apenas variáveis na memória;
 - SOs utilizam essa estratégia para os dispositivos de vídeo.

Mapeamento de E/S na Memória

- O **terceiro método** é implementado através de uma estratégia híbrida, que emprega dois espaços de endereçamento:
 - Registradores de controle são mapeados para portas de E/S;
 - *Buffers* de dados de E/S são mapeados para a memória;
 - Exemplo: Pentium - endereços de 640k a 1M para os *buffers* de dados e as portas de E/S de 0 a 64k - 1;

Mapeamento de E/S na Memória



- (a) Espaços de memória e E/S separados (uso de portas);
- (b) E/S mapeada na memória;
- (c) Modelo híbrido (dois espaços de endereçamento).



Mapeamento de E/S na Memória

- Como funciona a comunicação da UCP com os dispositivos nestes métodos?
- Quando a UCP deseja ler uma palavra (da memória ou de uma porta de E/S), ela coloca o endereço que ela está desejando no barramento de endereço e manda um comando **READ** no barramento de controle;
- Uma segunda linha de sinal é usada para informar se o espaço requisitado é de E/S ou de memória;
- No caso (a), espaços de memória e E/S separados, apenas um dos espaços de memória, o do dispositivo correspondente, responderá a requisição;



Mapeamento de E/S na Memória

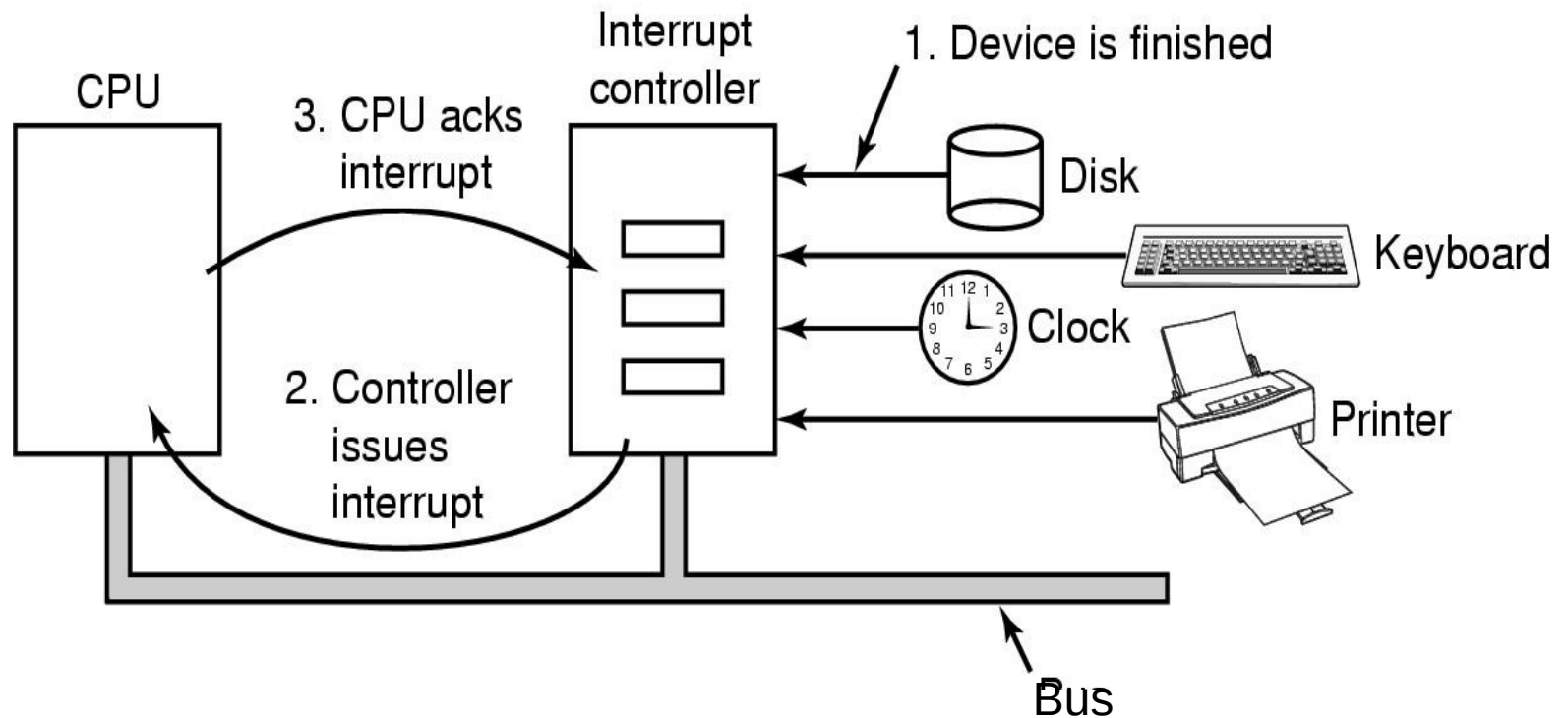
- No modelo (b), E/S mapeada na memória, como só há um espaço de memória, cada módulo de memória e cada dispositivo de E/S comparam as linhas de endereço com a faixa de endereços a eles designadas; responde aquele ao qual o endereço esteja contido na faixa;
- Uma grande vantagem deste modelo é, caso o programador precise fazer uso de instruções de E/S especiais, os registradores de controle do dispositivo são mapeados como variáveis na memória, e por isso podem ser manipulados diretamente por uma linguagem de alto nível, como C (no modelo anterior, o uso de uma linguagem de montagem seria obrigatório)

Uso de interrupções

- Interrupções de E/S (*interrupt-driven I/O*):
 - Sinais de interrupção são enviados (através dos barramentos) pelos dispositivos ao processador;
 - Após receber uma interrupção, o **controlador de interrupções** decide o que fazer;
 - Envia para UCP;
 - Ignora no momento, pois outra interrupção está em andamento ou recebeu simultaneamente uma de maior prioridade; neste caso, os dispositivos geram novos sinais de interrupção até serem atendidos;
 - O controlador envia para a UCP através do barramento de endereço um número que serve para identificar qual dispositivo de E/S precisa ser atendido.

Uso de interrupções

- Exemplo de como uma interrupção ocorre:





Uso de interrupções

- Controlador de Interrupções:
 - Está presente na placa-mãe;
 - Possui vários manipuladores de interrupção;
 - Diferentes tipos de interrupções: IRQs (*Interrupt ReQuest*);
- Manipuladores de interrupção:
 - Gerenciam interrupções realizadas pelos dispositivos de E/S;
 - Bloqueiam *driver* até dispositivo terminar a tarefa;
- O Sinal (linha) de interrupção é amostrado dentro de cada ciclo de instrução do processador;
 - Se houver um sinal ativo, a UCP salva o contexto atual e atende a interrupção;

Uso de interrupções

Exemplo de tabela de interrupções dos dispositivos:

IRQ	Uso padrão	Outras utilizações
00	Timer do sistema	Nenhum
01	Teclado	Nenhum
02	IRQs 8 a 15	Modem, placa de vídeo, porta serial (3, 4), IRQ 9
03	Porta serial 2	Modem, placa de som, placa de rede
04	Porta serial 1	Modem, placa de som, placa de rede
05	Placa de som (codec)	LPT2, COM 3 e 4, Modem, placa de rede, HDC
06	FDC	Placa aceleradora de fita
07	Porta paralela 1	COM 3 e 4, Modem, placa de som, placa de rede
08	Relógio de tempo real	Nenhum
09	Placa de som (midi)	Placa de rede, SCSI, PCI
10	Nenhum	Placa de rede, placa de som, SCSI, PCI, IDE 2
11	Placa de vídeo VGA	Placa de rede, placa de som, SCSI, PCI, IDE 3
12	Mouse P/S2	Placa de rede, placa de som, SCSI, PCI, IDE 3, VGA
13	FPU (<i>Float Point Unit</i>)	Nenhum
14	IDE primária	Adaptador SCSI
15	IDE secundária	Placa de Rede, adaptador SCSI



Maneiras de realizar E/S

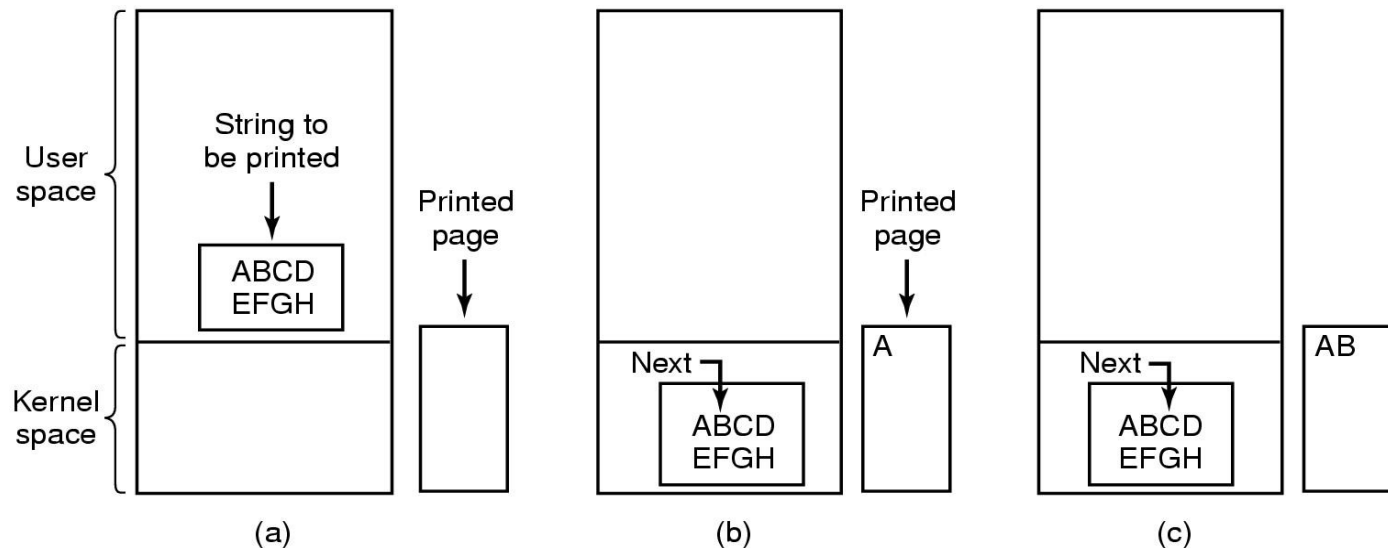
- Essencialmente, há três maneiras de implementar um método de controle de E/S:
 - E/S programada;
 - Mais usada atualmente em sistemas embarcados/embutidos;
 - E/S orientada à interrupção;
 - E/S com uso de DMA;

Maneiras de realizar E/S

- E/S programada:
 - É a forma mais simples, todo o trabalho é realizado pela UCP;
 - O sistema operacional verifica o estado do dispositivo, e caso esteja disponível, envia um caracter por vez para o registrador de dados do dispositivo de E/S;
 - Após copiado o primeiro caracter, o S.O. verifica se o dispositivo está próximo para aceitar o próximo caracter, geralmente verificando o valor em um registrador de estado;
 - Caso o dispositivo de E/S ainda esteja processando o caracter anterior, seu estado é 'indisponível', o que faz com que o S.O. permaneça verificando o valor continuamente, no que é chamado de espera ocupada ou *pooling*;

Maneiras de realizar E/S

- E/S programada: passos para impressão de uma cadeia de caracteres (laço até que toda a cadeia tenha sido impressa);



- Desvantagem:
 - UCP é ocupada o tempo todo até que a E/S seja completada;

Maneiras de realizar E/S

- E/S orientada à interrupção:
 - A UCP comanda a um dispositivo de E/S que realize uma operação, e enquanto ela é realizada, ela é escalonada para executar outro processo, até que receba uma interrupção do dispositivo de E/S, avisando que terminou a operação;
 - No exemplo da impressão, a impressora não armazena os caracteres, imprime a medida que chegam; o tempo que leva para realizar a operação de impressão, a UCP faz outra tarefa pendente
 - Quando a impressora termina a impressão e está pronta para receber outros caracteres, gera uma interrupção;

Maneiras de realizar E/S

- E/S orientada à interrupção:
 - A UCP comanda a um dispositivo de E/S que realize uma operação, e enquanto ela é realizada, ela é escalonada para executar outro processo, até que receba uma interrupção do dispositivo de E/S, avisando que terminou a operação;
 - No exemplo da impressão, a impressora não armazena os caracteres, imprime a medida que chegam; o tempo que leva para realizar a operação de impressão, a UCP faz outra tarefa pendente
 - Quando a impressora termina a impressão e está pronta para receber outros caracteres, gera uma interrupção;

Maneiras de realizar E/S

- E/S com uso da DMA:
 - Usa um controlador de interrupções via hardware, projetado para esta tarefa em particular;
 - O DMA executa o mesmo método da E/S programada, fazendo todo o trabalho que seria realizado pela UCP;
 - Redução do número de interrupções;
 - Presente principalmente em dispositivos baseados em bloco, como discos rígidos e outros dispositivos de armazenamento;
 - Pode estar na placa-mãe e servir vários dispositivos como controladora de DMA independente do dispositivo
 - O controlador de DMA tem acesso ao barramento do sistema, independente da UCP;
 - A única desvantagem é que o controlador de DMA é mais lento que a UCP, porém seu uso só não é vantajoso quando a UCP não tem outras tarefas para fazer;



Maneiras de realizar E/S

- O controlador de DMA contém vários registradores que podem ser lidos e escritos pela UCP (basicamente, podendo ser 'programado'):
 - Registrador de endereço de memória;
 - Registrador contador de bytes;
 - Registrador(es) de controle;
 - Porta de E/S em uso;
 - Tipo da transferência (leitura ou escrita);
 - Unidade de transferência (byte ou palavra);
 - Número de bytes a ser transferido;

Maneiras de realizar E/S

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

Maneiras de realizar E/S

- **Com DMA:** Leitura de um bloco de dados em um disco:
 1. Além do endereço a ser lido, a UCP 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;
 3. Controladora do dispositivo lê o bloco de dados e o armazena em seu *buffer*, verificando consistência;
 4. Controladora do dispositivo copia os dados para RAM no endereço especificado na DMA (modo direto);

(continua)

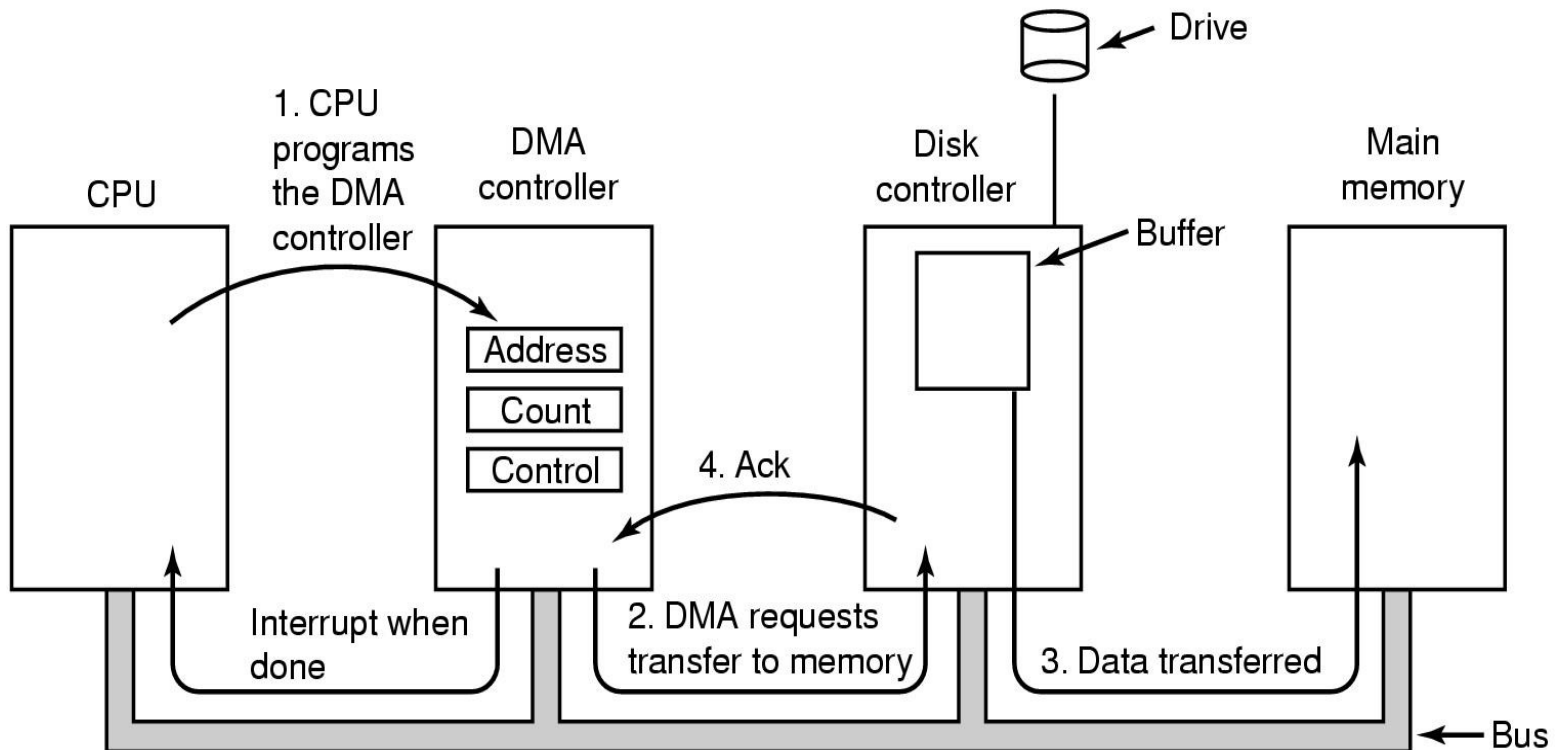


Maneiras de realizar E/S

5. 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;
6. 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 UCP;
7. Quando o SO inicia o atendimento à interrupção, o bloco de dados já está na RAM.

Maneiras de realizar E/S

Etapas de uma operação de E/S com DMA:





Maneiras de realizar E/S

- A DMA pode tratar múltiplas transferências simultaneamente, para isso, é necessário:
 - Possuir vários conjuntos de registradores (um para cada operação);
 - Decidir quais requisições devem ser atendidas a cada momento: implementação de uma política de escalonamento (*Round-Robin* ou prioridades, por exemplo);



Maneiras de realizar E/S

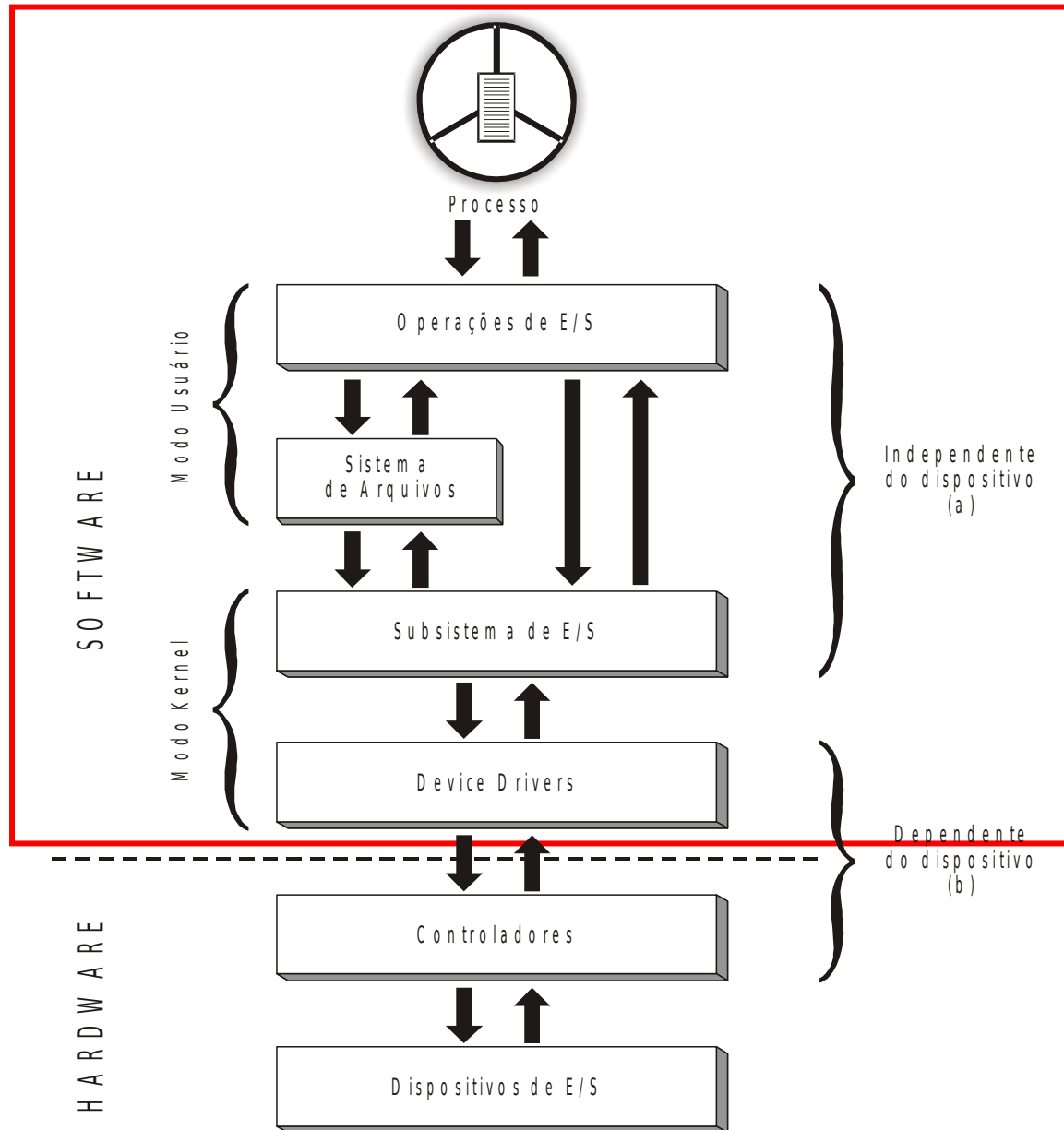
- Por que a DMA não escreve diretamente na RAM?
 - Permite realizar consistência dos dados antes de iniciar alguma transferência;
 - Dados (bits) são transferidos do disco a uma taxa constante, independentemente da controladora estar pronta ou não;
 - Acesso à memória depende de acesso ao barramento, que pode estar ocupado com outra tarefa;
 - Com o *buffer*, o barramento é usado apenas quando a DMA opera;



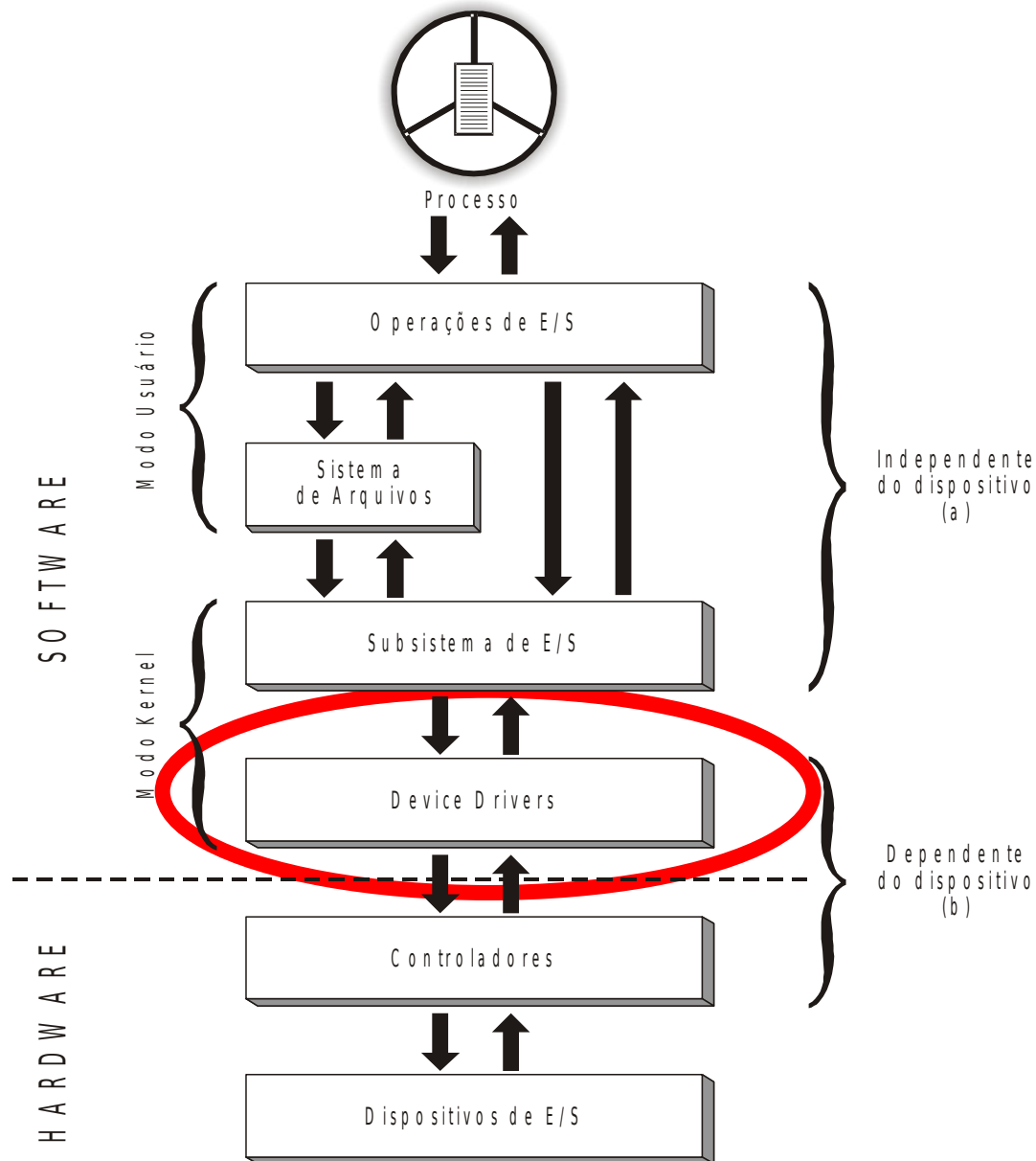
Princípios de Software de E/S

- A organização do *software* como uma série de camadas facilita a **independência dos dispositivos**, que possibilita aos programadores escrever programas capazes de acessar quaisquer dispositivos de E/S sem ter que especificá-los antecipadamente:
 - 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;

Organização do sistema de E/S



Organização do sistema de E/S



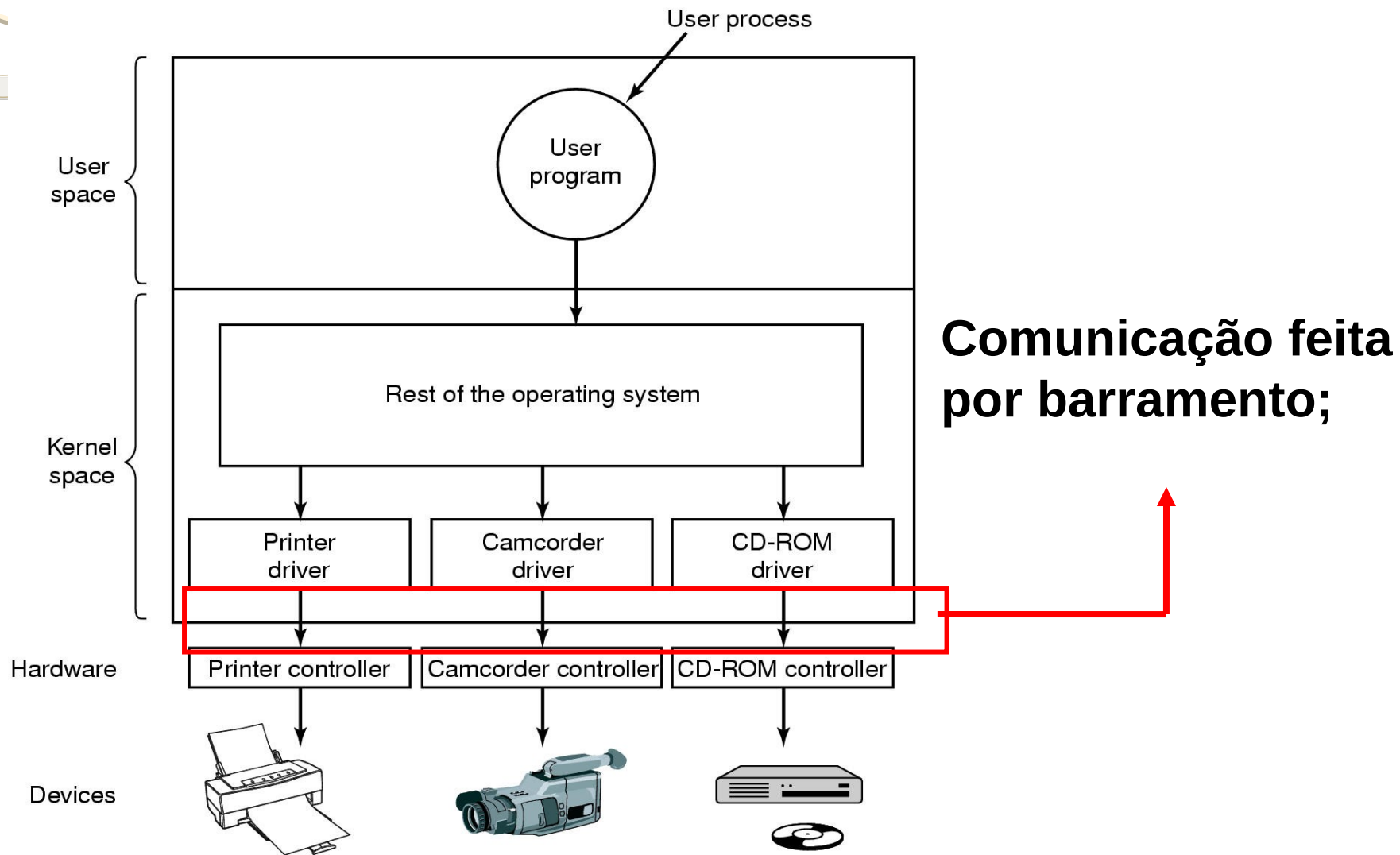
Princípios de Software de E/S

- ***Drivers:***

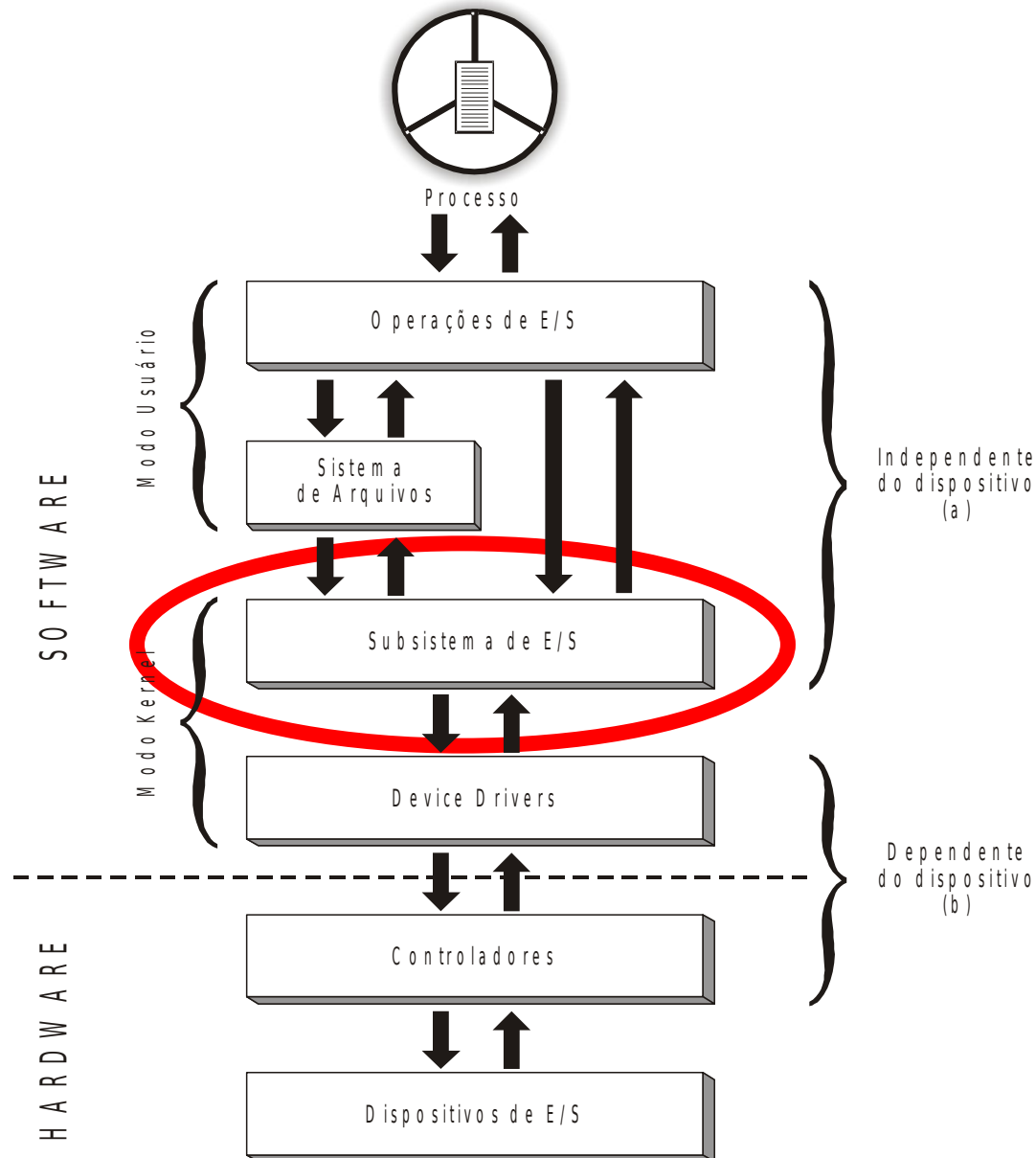
- São programas que fazem a comunicação entre o S.O. (que os gerencia pelo *kernel*) e o *hardware*, contendo todo o código dependente do dispositivo;
- Sua função é a de controlar funcionamento dos dispositivos por meio de sequência de comandos escritos/lidos nos/dos registradores da controladora;
- Dispositivos diferentes possuem *drivers* diferentes;
 - Classes de dispositivos podem ter o mesmo *driver*;
- São dinamicamente carregados;

Dispositivos de E/S

Princípios de Software - Camadas



Dispositivos de Entrada e Saída





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*);



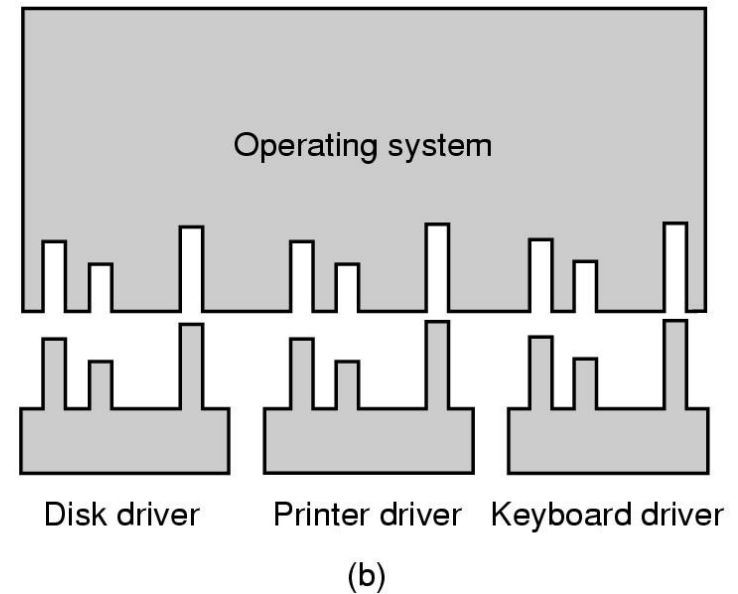
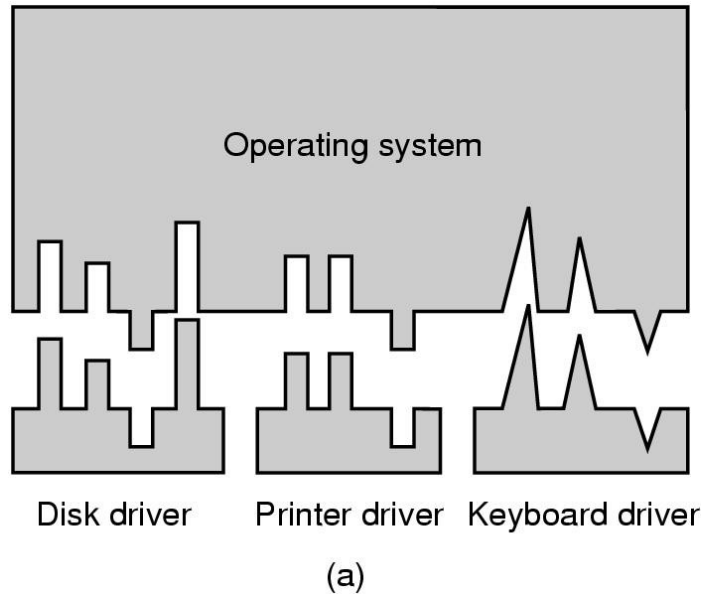
Dispositivos de E/S

Princípios de Software - Camadas

- **Software Independente de E/S:**
 - Realizar as funções comuns a qualquer dispositivo;
 - Prover uma interface uniforme para os drivers dos dispositivos
 - Número de procedimentos que o restante do SO pode utilizar para fazer o driver trabalhar para ele;
 - 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 frequentemente acessados;

Dispositivos de E/S

Princípios de Software - Camadas



(a) Sem padrão de interface

(b) Com padrão de interface (uniforme)



Dispositivos de E/S

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 (necessário para permitir 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;



Exemplo de operação de E/S (ilustrando as operações em diferentes níveis)

1. Um processo emite uma chamada de sistema bloqueante (por exemplo: `read`) para um arquivo que já esteve aberto (`open`);
2. 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;
3. Se os parâmetros estiverem corretos, mas o arquivo não estiver no *buffer*, a E/S precisa ser realizada;
 - 3.1. E/S é escalonada;
 - 3.2. Subsistema envia pedido para o *driver*;



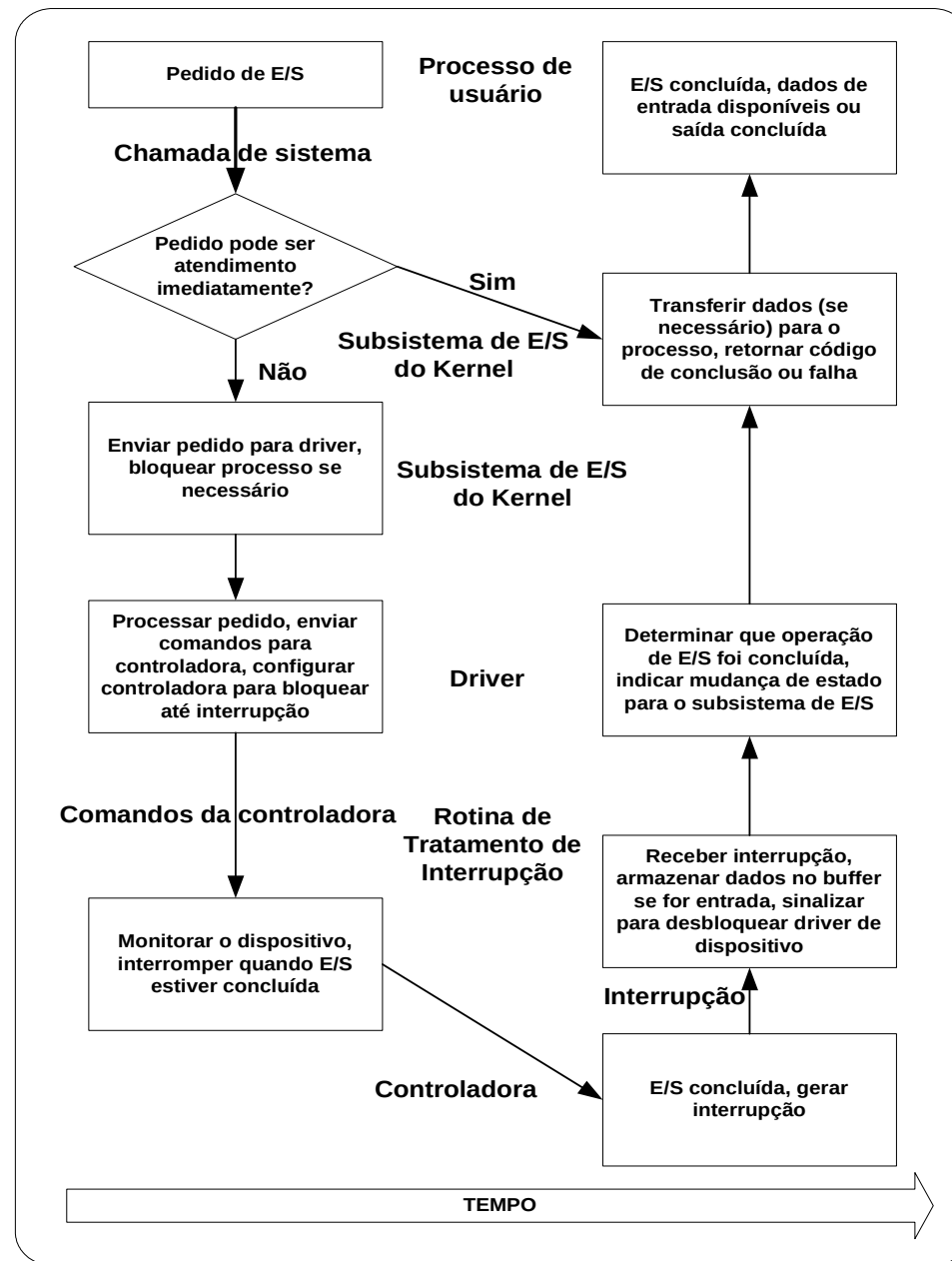
Exemplo de operação de E/S (ilustrando as operações em diferentes níveis)

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

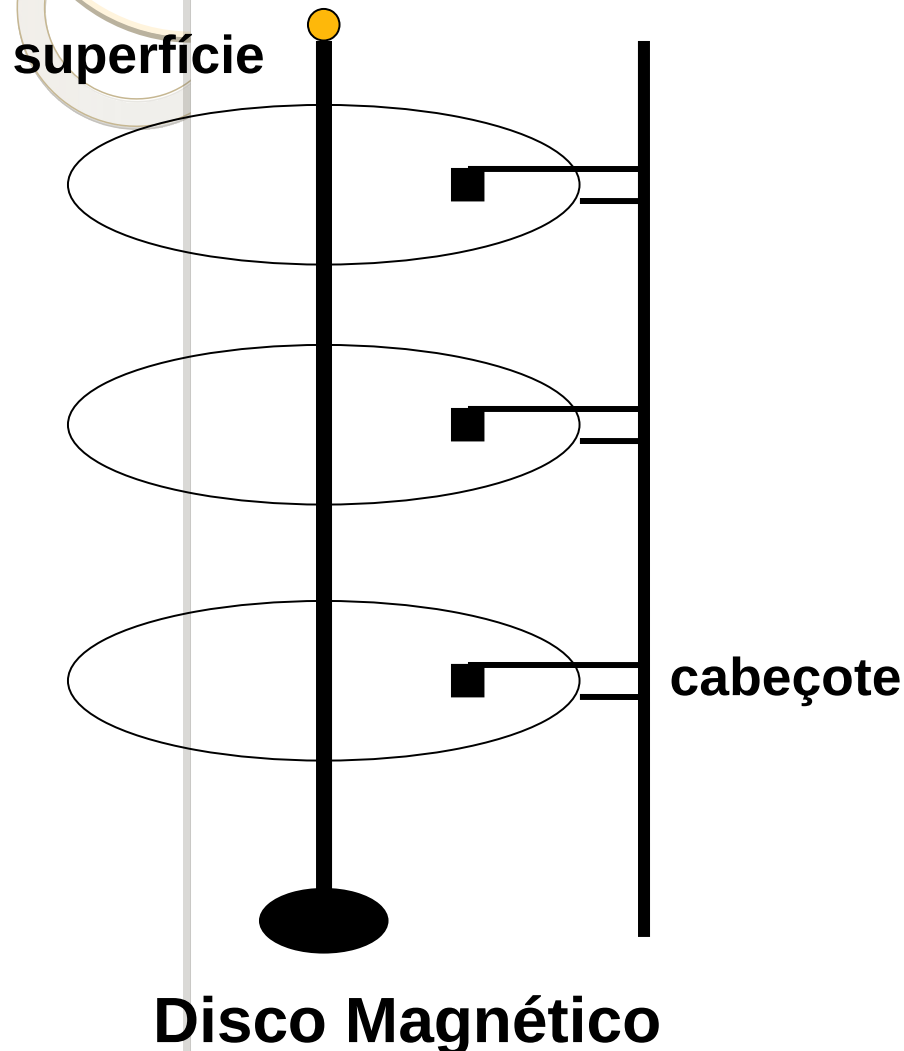


Exemplo de operação de E/S (ilustrando as operações em diferentes níveis)

8. *O driver* recebe o sinal, determina qual pedido de E/S foi concluído, determina o status e sinaliza que o pedido está concluído;
9. O 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;
10. Quando o escalonador escalona o processo para a UCP, ele retoma a execução na conclusão da chamada ao sistema.



Dispositivos de E/S - Discos



- 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;



Dispositivos de E/S - Discos

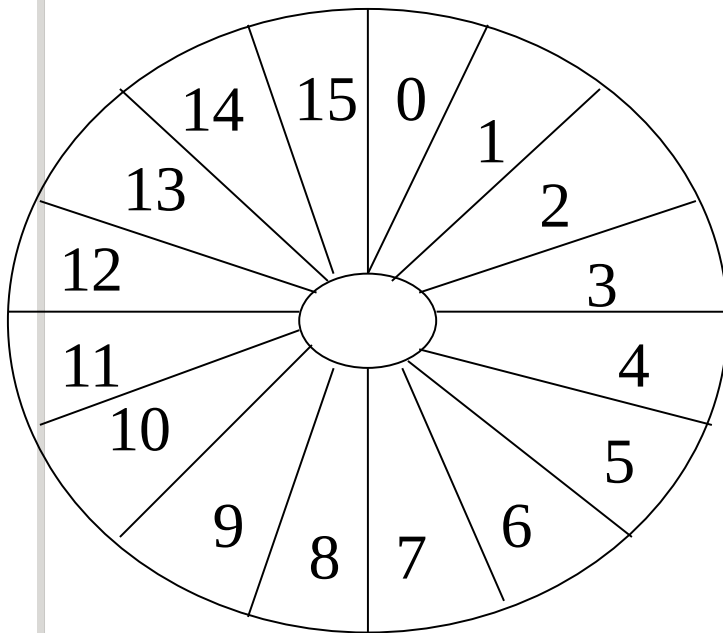
- Discos Magnéticos:
 - Grande evolução em relação a:
 - Velocidade de acesso (*seek*): tempo de deslocamento do cabeçote até o cilindro correspondente à trilha a ser acessada;
 - Transferências: tempo para transferência (leitura/escrita) dos dados;
 - Capacidade;
 - Preço;

Dispositivos de E/S - Discos

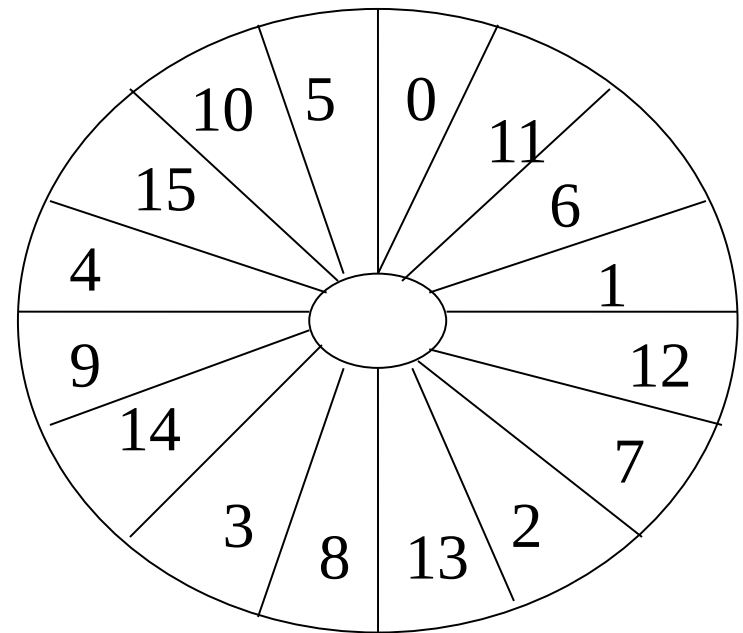
- Técnica para reduzir o tempo de acesso: entrelaçamento (*interleaving*):
 - Setores são numerados com um espaço entre eles;
 - Entre o **setor K** e o **setor K+1** existem **n** (fator de entrelaçamento) setores;
 - Número **n** depende da velocidade do processador, do barramento, da controladora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

Dispositivos de E/S - Discos

Exemplo: 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 – Discos

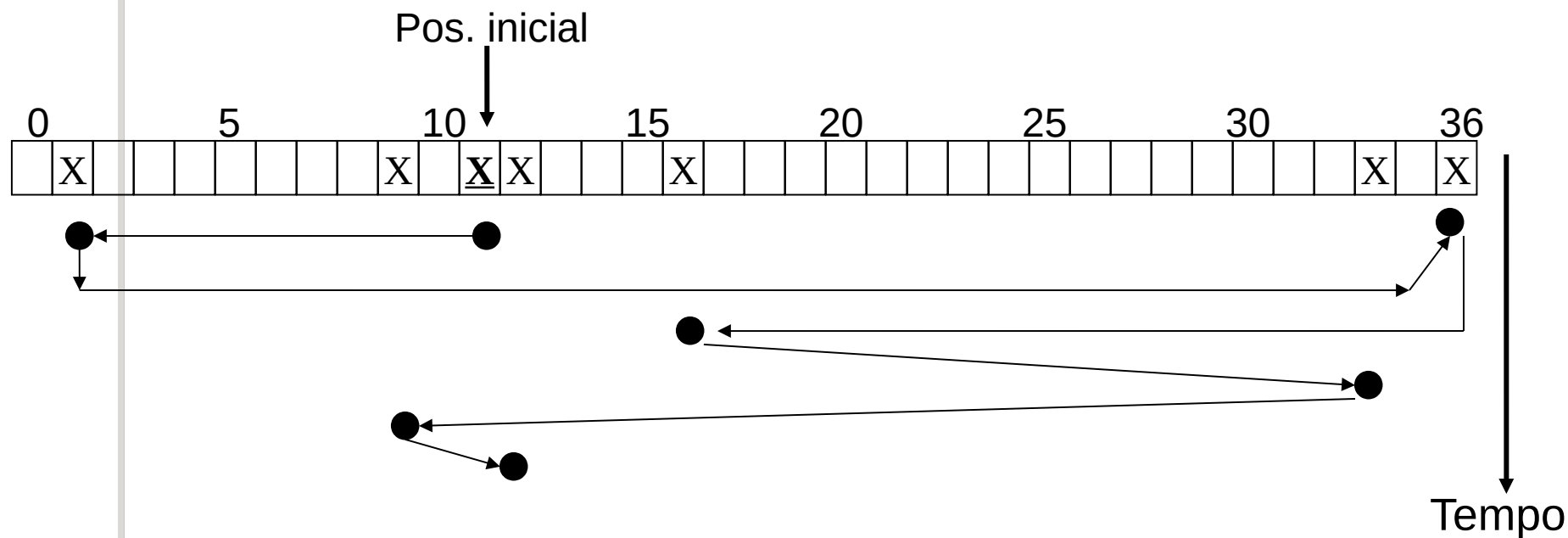
- Algoritmos de escalonamento no disco:
 - FCFS (FIFO): *First-Come First-Served*;
 - SSF: *Shortest Seek First*;
 - *Elevator* (também conhecido como SCAN);
 - Variação: Circular Scan (*C-SCAN*)
- Escolha do algoritmo depende do número e do tipo de pedidos;
- *Driver* mantém uma lista encadeada com 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;

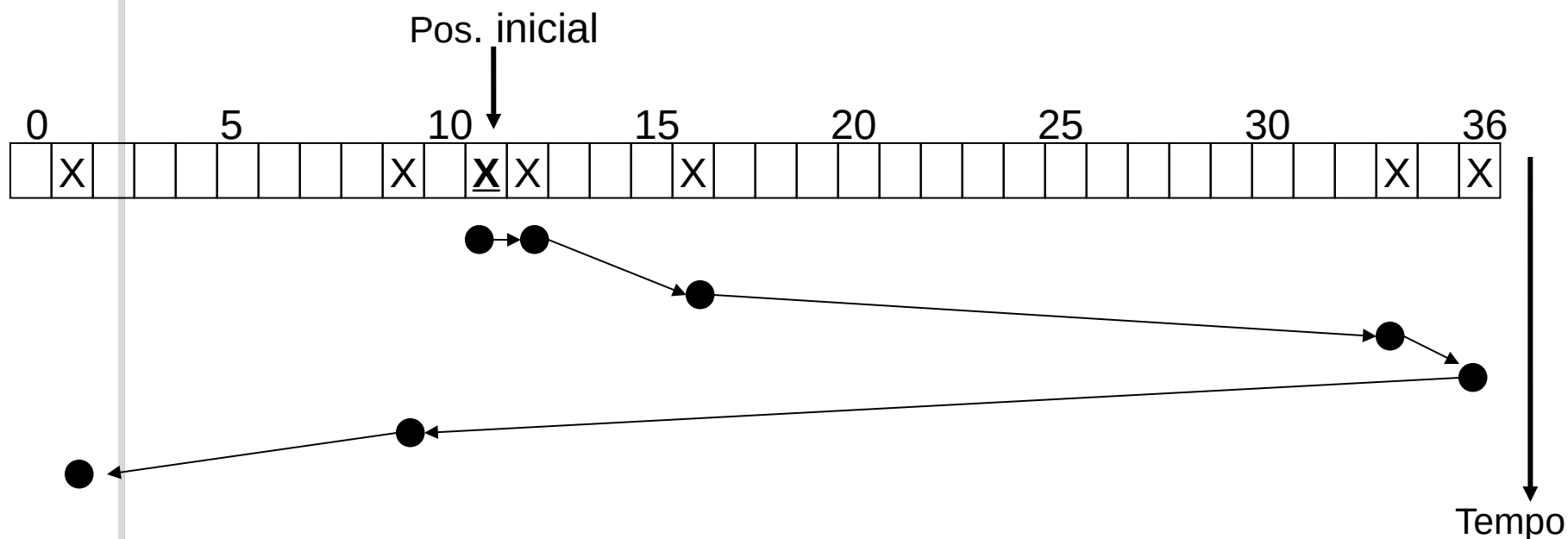
movimentos do braço (número de cilindros): 10,35,20,18,25,3 = 111;

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*) → organização de duas ou mais unidades de memória, para obter maior segurança e/ou desempenho;
- 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 S.O., só existe um único disco (visão lógica);

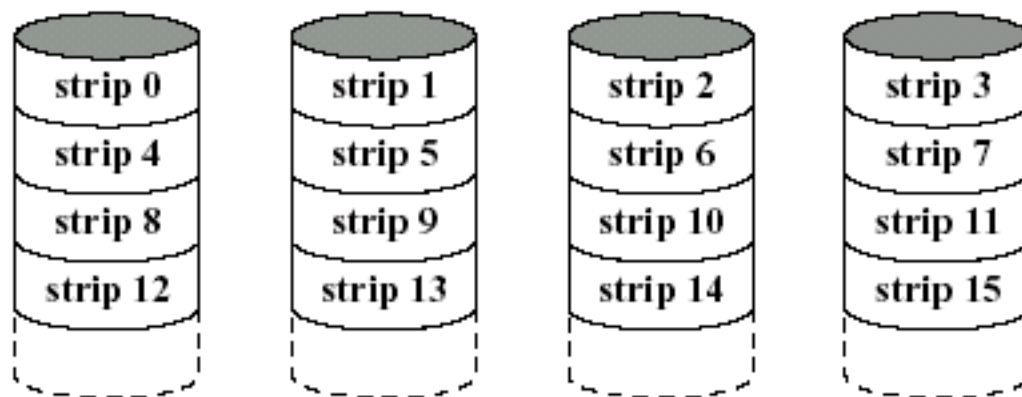


Dispositivos de E/S – Discos RAID

- 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;
 - Sem controle ou correção de erros;
 - Todo o espaço do disco é utilizado para armazenamento;
 - Utilizam mesma controladora (controladora RAID);
 - Aplicações multimídia (alta taxa de transferência);

Dispositivos de E/S – Discos RAID

- RAID 0:



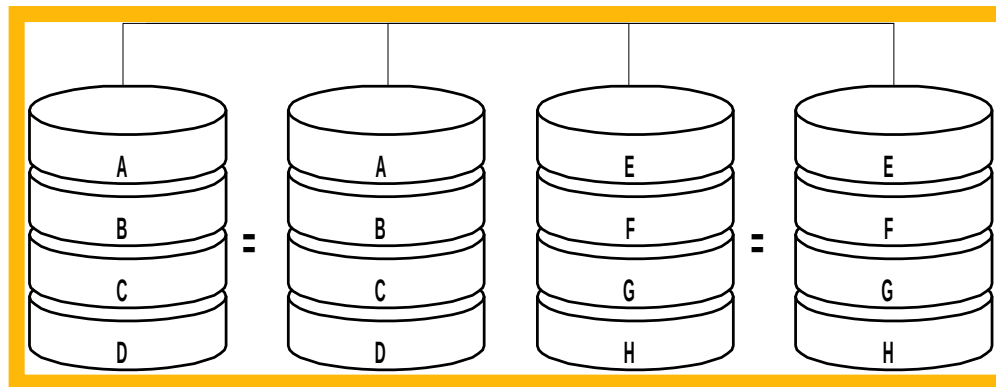


Dispositivos de E/S – Discos RAID

- 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 (ou 0+1):
 - Combinação dos RAID 1 e RAID 0;

Dispositivos de E/S – Discos RAID

- Nível 10 (ou 0+1):

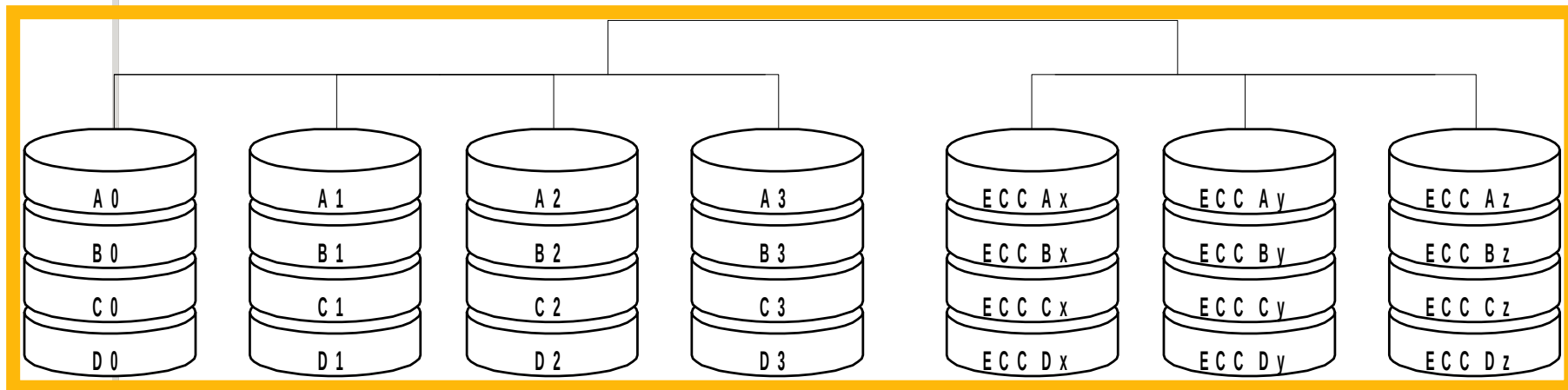


Dispositivos de E/S – Discos RAID

- 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;

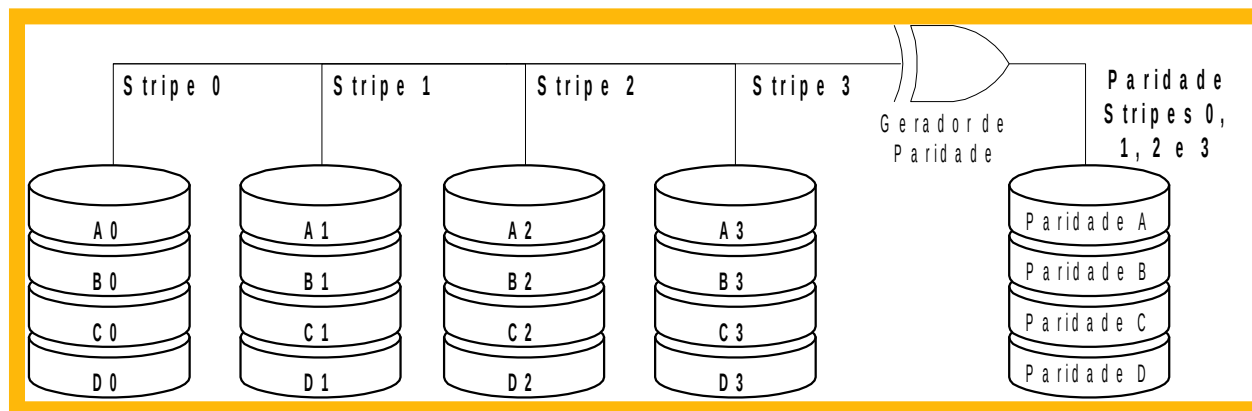
Dispositivos de E/S – Discos RAID

- Nível 2



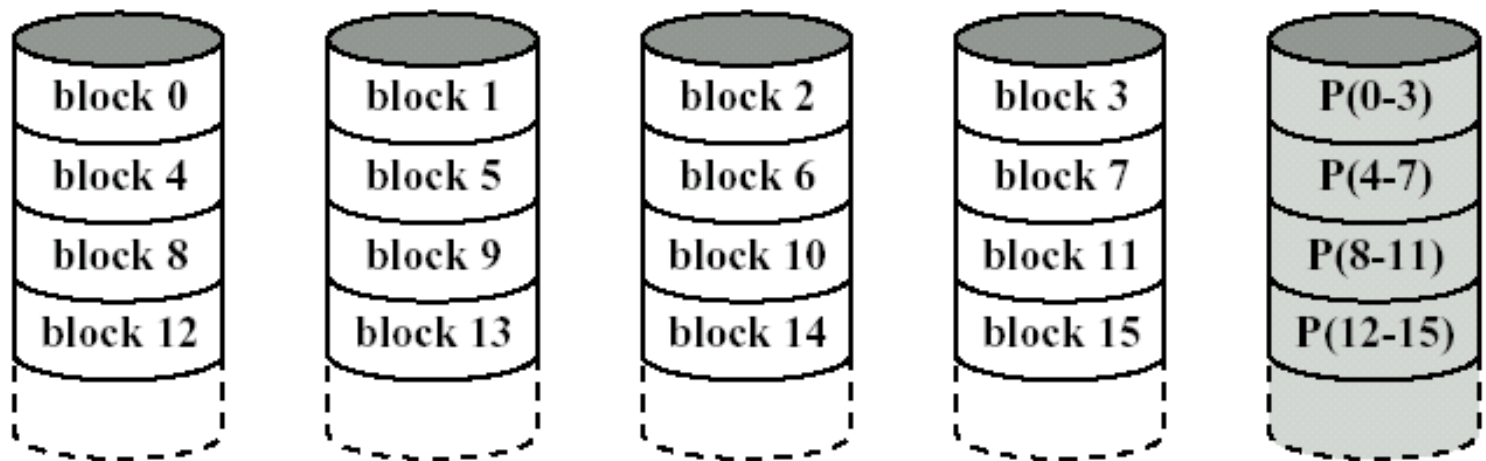
Dispositivos de E/S – Discos RAID

- Nível 3



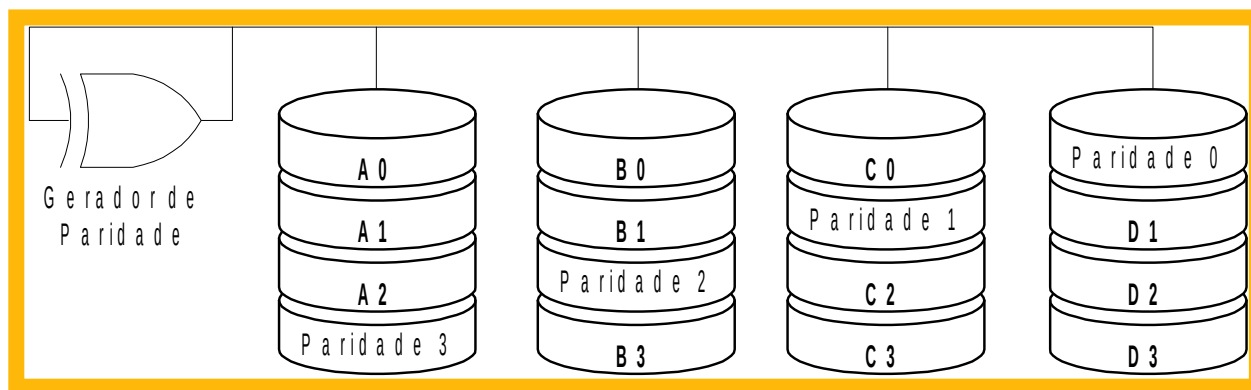
Dispositivos de E/S – Discos RAID

- Nível 4



Dispositivos de E/S – Discos RAID

- Nível 5



Dispositivos de E/S – Discos RAID

- Nível 6
 - 2 cálculos diferentes de paridade

