

Curso de

# **Sistemas Operacionais**

## **Gerenciamento de Entrada/Saída**

**Prof. Dr. Robson Augusto Siscoutto**  
**e-mail: [robson.siscoutto@unoeste.br](mailto:robson.siscoutto@unoeste.br)**

# **Sistemas Operacionais**

## **Gerenciamento de Entrada/Saída**

---

- **Princípios de Hardware de E/S**
- **Princípios de Software de E/S**
- **Camadas de Software de E/S**
- **Algoritmos de Escalonamento de Braço do Disco**

---

# **Princípios de Hardware de E/S**

---

# Princípios de Hardware de E/S

## Dispositivos de E/S

---

- Os Dispositivos de E/S são divididos em 2 categorias:
  - **(1) Dispositivos de Blocos**
    - Armazena informações em blocos de tamanho fixo;
    - Cada bloco com seu próprio endereço;
    - Os tamanho dos blocos variam de 512 bytes a 32768 bytes;
    - Exemplo: HD
  - **(2) Dispositivos de Caractere**
    - Envia ou recebe um fluxo de caracteres;
    - Sem considerar qualquer estrutura de dados;
    - O fluxo não é endereçável e não dispõe de operações de posicionamento;
    - Exemplo: Impressoras, Interfaces de rede, Mouse.

# Princípios de Hardware de E/S

## Dispositivos de E/S

- Os Dispositivos de E/S
  - Taxas de Dados:

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

# Princípios de Hardware de E/S

## Controladores de Dispositivos

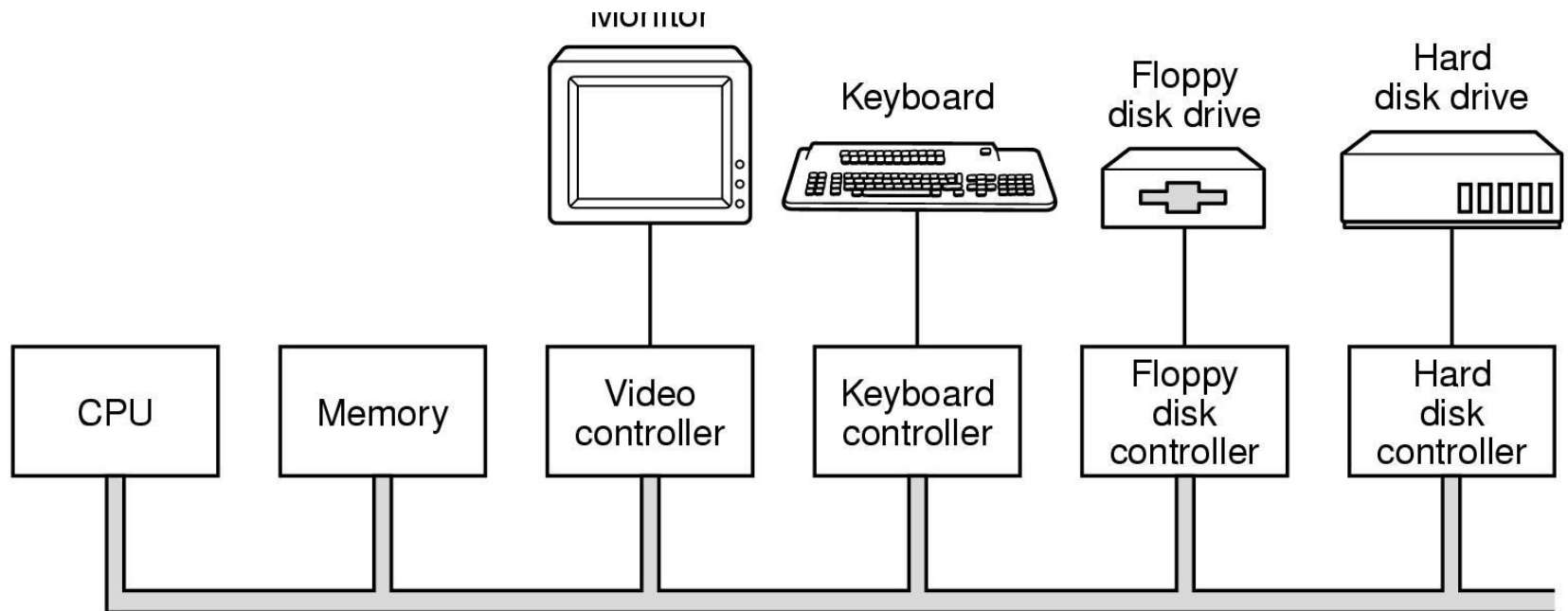
---

- As unidade de E/S consistem de componentes:
  - Mecânico e
  - Eletrônico;
- O componente eletrônico é chamado de:
  - Controlador do Dispositivo ou Adaptador
    - permite manipular múltiplos dispositivos;
    - se apresenta na forma de placa controladora de circuito impresso inserida em um conector de expansão;
- O componente Mecânico é o dispositivo propriamente dito;

# Princípios de Hardware de E/S

## Controladores de Dispositivos

- As unidade de E/S consistem de componentes:
  - Mecânico e
  - Eletrônico;



# Princípios de Hardware de E/S

## Controladores de Dispositivos

---

- As unidade de E/S consistem de componentes:
  - Mecânico e
  - Eletrônico;
- Tarefas do Componente Eletrônico:
  - Converter fluxo serial de bits para bloco de bytes e executar correção de erros (*checksum*)
    - O bloco de bytes é montado, bit a bit, em buffer dentro do controlador;
  - Colocar o bloco na Memória Principal;



# Princípios de Hardware de E/S

## E/S Mapeada na Memória

---

- Controladoras possuem registradores para:
  - Comunicação com a CPU;
  - o S.O. escreve comandos nos registradores para serem executados;
    - Aceitar dados, ligar e desligar, estado do dispositivo, etc;
- **Três Formas de acesso pela CPU aos Registradores :**
  - **1ª Forma:** Cada registrador tem uma porta de E/S (8 ou 16 bits) associada a ele;
    - IN REG-CPU, PORT-CONTROLADOR ( $\text{reg} \leftarrow \text{port}$ )
    - OUT PORT-CONTROLADOR, REG-CPU ( $\text{port} \leftarrow \text{reg}$ )
    - IBM 360

# Princípios de Hardware de E/S

## E/S Mapeada na Memória

---

- **Formas de acesso pela CPU aos Registradores :**
  - continuação do 1ª forma:
    - Espaço de Endereçamento para a memória e E/S são diferentes;
  - **2ª Forma: E/S Mapeada na Memória**
    - Mapear todos os registradores de controle no espaço de endereçamento da memória;
      - Cada registrador é associado a um endereço de memória único;
      - Endereços Associados estão no topo da memória;

# Princípios de Hardware de E/S

## E/S Mapeada na Memória

---

- **Formas de acesso pela CPU aos Registradores**

- Em alguns computadores, os registradores estão em memória

- Exemplos: IBM PC

• controladora	End. E/S	Vetor Interrupção
Disquete	3D0-3F7	14
Teclado	060-063	9
Disco rígido	320-32F	13
Impressora	378-37F	15

# Princípios de Hardware de E/S

## E/S Mapeada na Memória

---

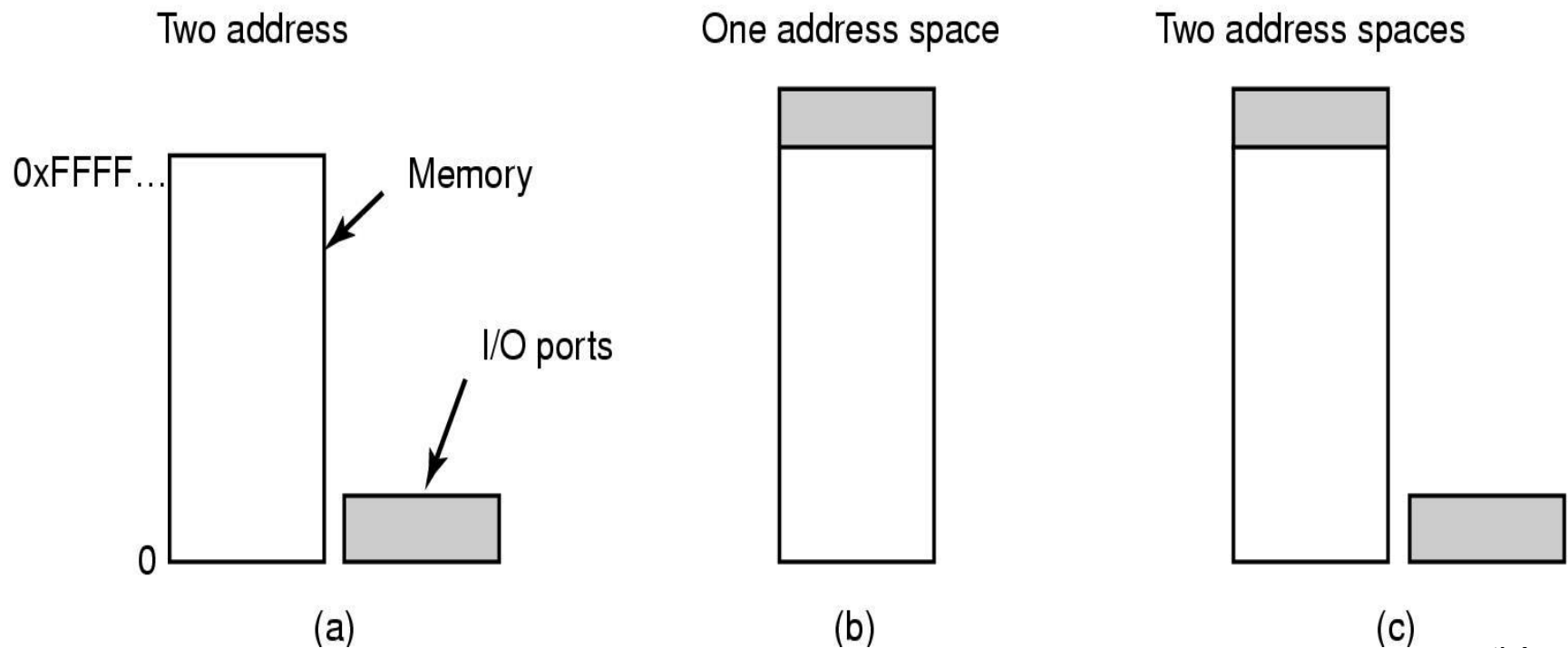
- **Formas de acesso pela CPU aos Registradores :**
  - **3ª Forma: Forma Híbrida**
    - buffers de dados de E/S mapeados em memória e
    - portas de E/S separadas para os registradores de controle
  - Pentium utiliza essa estrutura:
    - endereços de 640 kb a 1 M reservado para os buffers de dados dos dispositivos;
    - portas de E/S de 0 a 64 Kb

# Princípios de Hardware de E/S

## E/S Mapeada na Memória

---

- Formas de acesso pela CPU aos Registradores

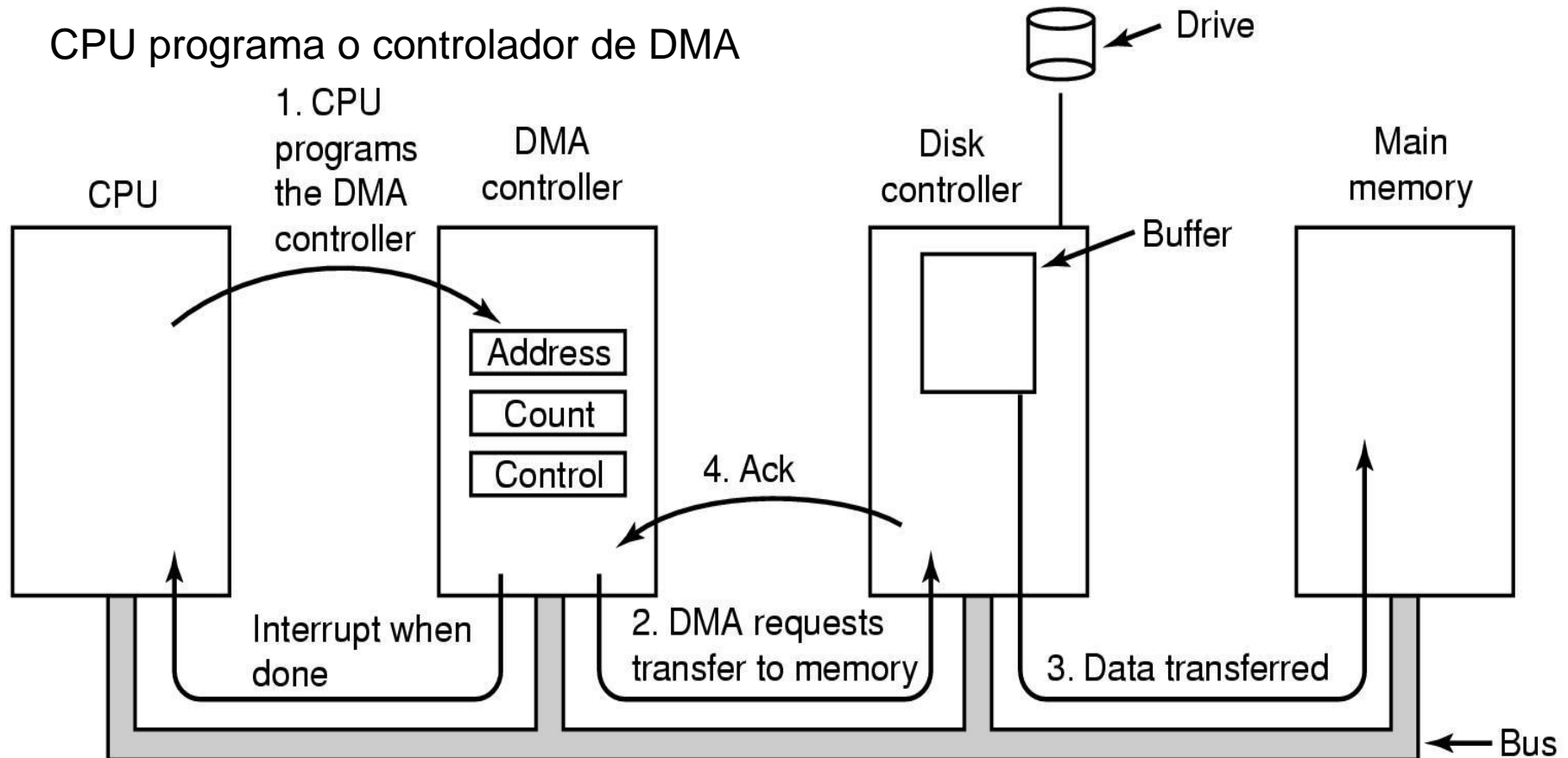


# Princípios de Hardware de E/S

## Acesso Direto à Memória (DMA)

- Operações de uma Transferência DMA

CPU programa o controlador de DMA

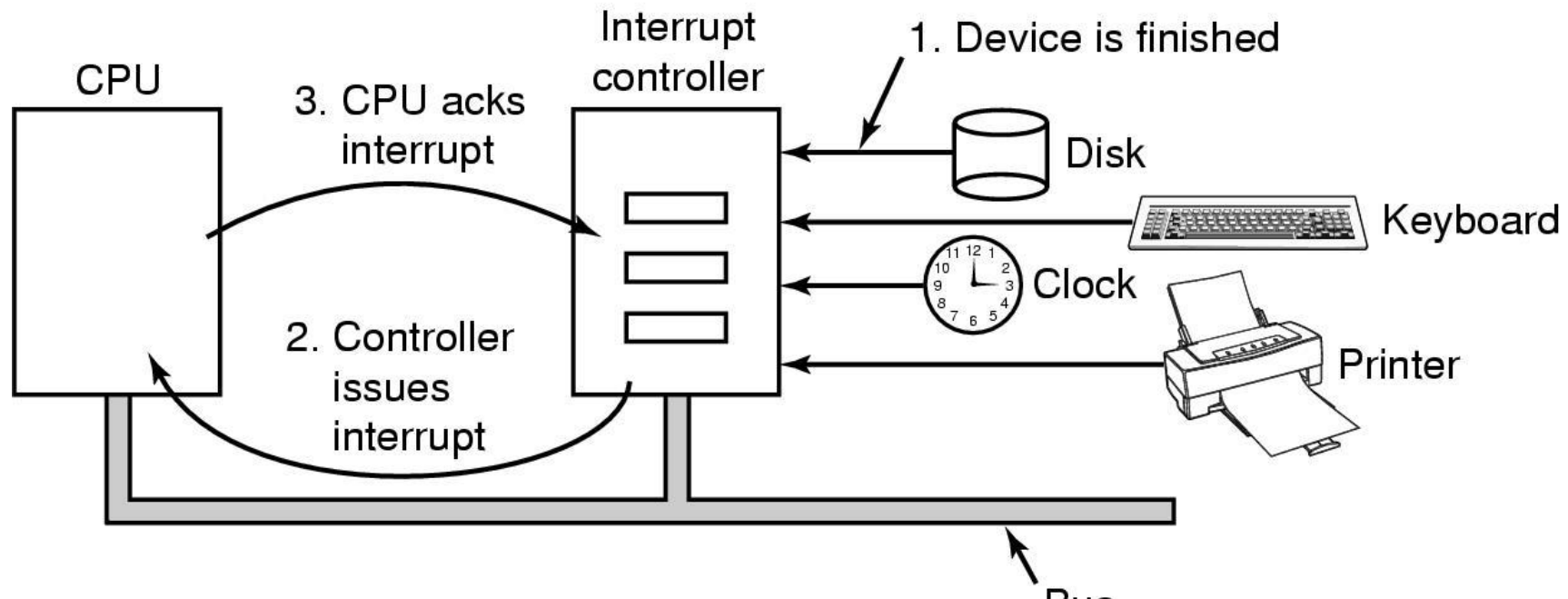


# Princípios de Hardware de E/S

## Interrupções

- Como ocorre uma Interrupção:

- As conexões entre os dispositivos e o controlador de interrupção usam de fato linhas de interrupção no barramento em vez de fios dedicados;



---

# **Princípios de Software de E/S**

---



# Princípios de Software de E/S

## Objetivos do Software de E/S

---

- **Independência do Dispositivo**
  - Programas podem acessar qualquer dispositivos de E/S
  - Sem necessitar alteração no código: floppy, hard drive
- **Uniformidade de identificação**
  - Nome de um arquivo ou dispositivos é uma string de caracteres ou um numero inteiro;
  - Totalmente independente do dispositivo;
- **Tratamento de Erros**
  - Tratados o mais próximo do hardware (p.ex. controladoras);

# Princípios de Software de E/S

## Objetivos do Software de E/S

---

- **Transferência Sincrona vs. Assíncrona**

- Transferência bloqueada (síncrona) vs. dirigida por interrupção (orientada à interrupção);
- Maioria dos dispositivos de E/S são assíncronos;

- **Buferização**

- Dados oriundos de dispositivos não podem ser armazenados no destino final;

- **Dispositivos Compartilhados vs. Dedicados**

- Discos são compartilhados (p.ex. Discos);
- Impressoras não são (p.ex. Fita);

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- Três maneiras diferentes:
  - E/S Programada;
    - CPU faz o trabalho;
  - E/S Orientada à interrupção;
  - E/S que usa DMA;

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Programada;**

- A cadeia de caracteres é montada em um buffer no espaço do usuários;
- O processo do usuário requisita o **dispositivo via chamada** ao sistema para abri-la;
- S.O copia o buffer com a cadeia para um vetor no espaço do núcleo, onde ele é facilmente acessado;
- Uma vez disponível o dispositivo (verificação conhecida como *Espera Ocupada ou Polling*), o S.O copia o primeiro caractere para o registrador de dados do dispositivo e assim sucessivamente
  - Utilizando E/S mapeada em Memória;

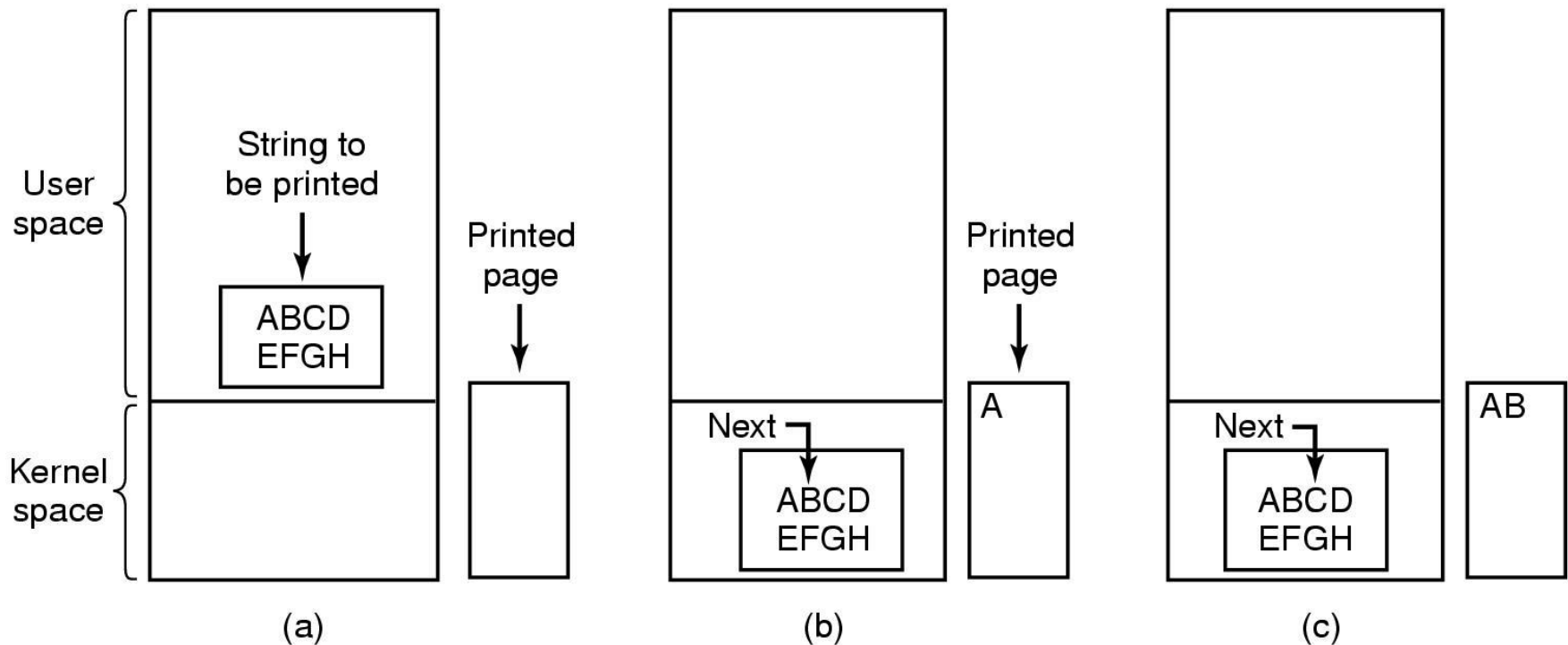
# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Programada**

- Passos na Impressão de uma Cadeia de Caracteres



# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Programada**

- Código seguido pelo S.O na Impressão de uma Cadeia de Caracteres em uma Impressora

```
copy_from_user(buffer, p, count);           /* p is the kernel bufer */
for (i = 0; i < count; i++) {                /* loop on every character */
    while (*printer_status_reg != READY) ;    /* loop until ready */
    *printer_data_register = p[i];            /* output one character */
}
return_to_user();
```

- **Vantagem:** Simplicidade;
- **Desvantagem:** Segura a CPU o tempo todo até que a E/S<sub>22</sub> seja feita;

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Orientada à Interrupção:**

- Utilizando o exemplo da Impressão:
  - Semelhante os passos E/S Programadas;
  - **Diferença:**
    - Uma vez o **caractere copia para a impressora**, a CPU chama o escalonador e outro processo é executado;
    - O **processo** que solicitou a impressão é **bloqueado** até que toda a cadeia seja impressa;
    - Uma vez **impresso um caractere**, para aceitar o próximo é gerado uma **interrupção que bloqueia o processo atual** e salva seu estado;
    - Uma **rotina de tratamento de interrupção** da impressora é executada;

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Orientada à Interrupção:**

```
copy_from_user(buffer, p, count);  
enable_interrupts( );  
while (*printer_status_reg != READY) ;  
*printer_data_register = p[0];  
scheduler();
```

(a)

```
if (count == 0) {  
    unblock_user( );  
} else {  
    *printer_data_register = p[i];  
    count = count - 1;  
    i = i + 1;  
}  
acknowledge_interrupt( );  
return_from_interrupt( );
```

(b)

- **(a)** código executado na Chamada e **(b)** Rotina de Tratamento da Interrupção;



# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Orientada à Interrupção:**

- **Desvantagem:**

- Ocorrência de uma interrupção para cada caractere;
    - Desperdiço de Tempo de CPU;

- **Solução:**

- Utilização de DMA;

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Usando DMA – Acesso Direto a Memória:**
  - Fazer com que o controlador de DMA alimente os caracteres, para a impressora um por vez, sem que a CPU seja perturbada;
  - DMA executada E/S programada:
    - Somente o controlador de DMA faz todo o Trabalho, em vez da CPU principal;
- **Vantagem:**
  - Reduz o numero de interrupções de um por caractere para um por buffer impresso;
- **Desvantagem:**
  - Controladora de DMA mais lenta que a CPU;

# Princípios de Software de E/S

## Maneiras de se Realizar E/S

---

- **E/S Usando DMA – Acesso Direto a Memória:**
  - Impressão de uma cadeia de caractere usando DMA

```
copy_from_user(buffer, p, count);  
set_up_DMA_controller();  
scheduler();
```

(a)

```
acknowledge_interrupt();  
unblock_user();  
return_from_interrupt();
```

(b)

- **(a)** código executado quando a chamada ao sistema print é feita e **(b)** Rotina de Tratamento da Interrupção;

---

# **Camadas do Software de E/S**

---

# Camadas do Software de E/S

---

- Software de E/S é organizado em 4 camadas:
  - Cada camada tem uma função bem definida para executar
  - E uma interface também bem definida para as camadas adjacentes.



# Camadas do Software de E/S

## Tratadores de Interrupção

---

- **Manip. de Interrupção devem ser escondidos:**
  - um processo que iniciou uma operação de E/S deve ficar bloqueado até o termino da operação;
  - ou seja, bloquear o driver que inicio uma operação de E/S até que a E/S se complete e uma interrupção ocorra
- **O procedimento de Interrupção faz esta tarefa:**
  - desbloquear o processo que iniciou a operação;
- Nomarmente, os drivers são estruturados como processo do nucleo do S.O.

# Camadas do Software de E/S

## Drivers dos Dispositivos

---

- Código específico do dispositivos ou conjunto de dispositivos;
- o Driver do dispositivo deve ser parte do núcleo do SO;
- o S.O. classificam os drivers em duas categorias principais:
  - **Dispositivos de Bloco**
    - Contêm vários blocos de dados que podem ser endereçados independentemente;
    - P.ex. Discos
  - **Dispositivos de Caractere**
    - geram ou aceitam uma sequência de caracteres
    - P.ex. teclados e impressoras.

# Camadas do Software de E/S

## Drivers dos Dispositivos

---

- **Funções de Driver:**

- Aceitar e executar requisições como leitura e gravação;
- Iniciar um dispositivo;
- Tratar necessidades de energia e registrar eventos.

- **Estrutura de um Driver:**

- Verifica parâmetros de entrada para validação;
- Verificar o status do dispositivo para ver se a requisição pode ser tratada imediatamente;
- Ligar o dispositivo ou um motor antes de iniciar a transferência;
- Uma vez ligado e pronto, o controle atual pode começar:



# Camadas do Software de E/S

## Drivers dos Dispositivos

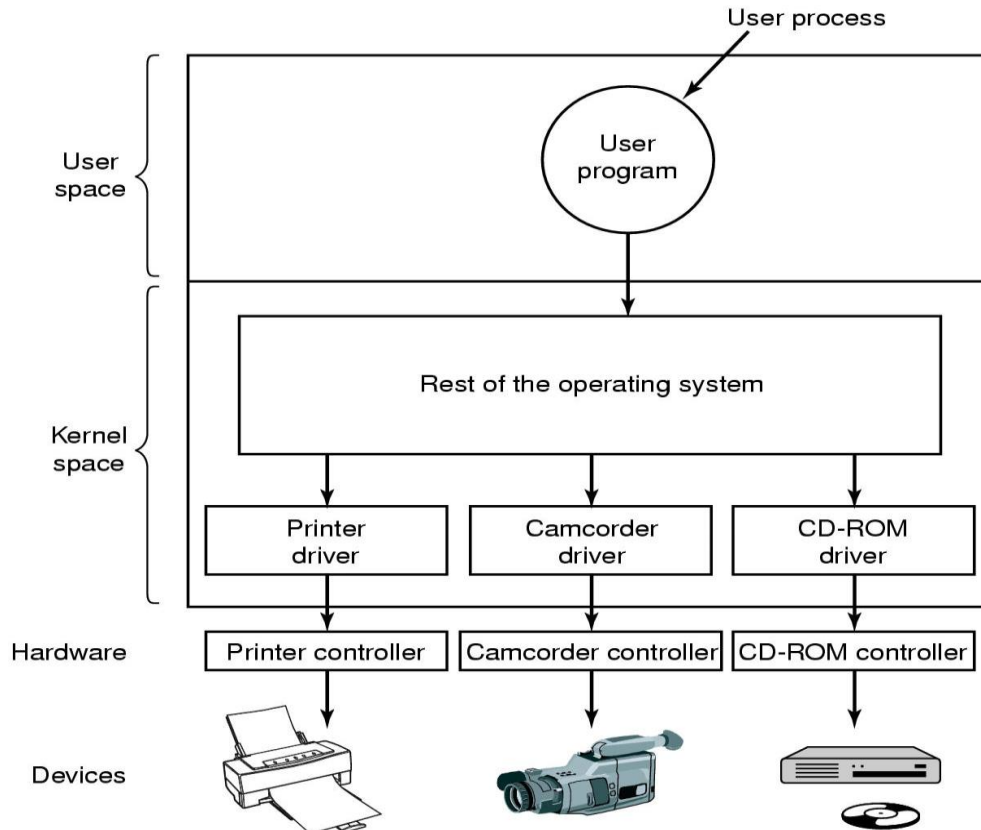
---

- **Controle do Dispositivo:**
  - Significa **emitir uma seqüência de comandos** para ele;
    - O driver é o local onde a seqüência de comandos é determinada;
  - Escreve os **comandos nos registradores** do controlador do dispositivo;
  - **Verifica se o controlador aceitou** o comando e se está preparado para receber o próximo;
  - **Sucessivamente**, até todos os comandos tenham sido emitidos;
  - **Recebido os comandos**, pode ocorrer duas situações:
    - o driver espera o controlador realizar algum trabalho para ele e se autobloqueia, sendo acordado por interrupção;
    - a operação se finaliza sem atraso, de maneira que o driver não precisa se bloquear;

# Camadas do Software de E/S

## Drivers dos Dispositivos

- **Posição lógica dos device drivers:**
  - Comunicação entre drivers e os controladores dos dispositivos é realizada via barramento;



# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

---

- **Objetivos básicos:**

- Realizar a função de E/S que são comuns a todos os dispos.
- Fornecer uma interface uniforme para o nível do SW do usuário;
- Algumas funções do Software de E/S independente do dispositivo:

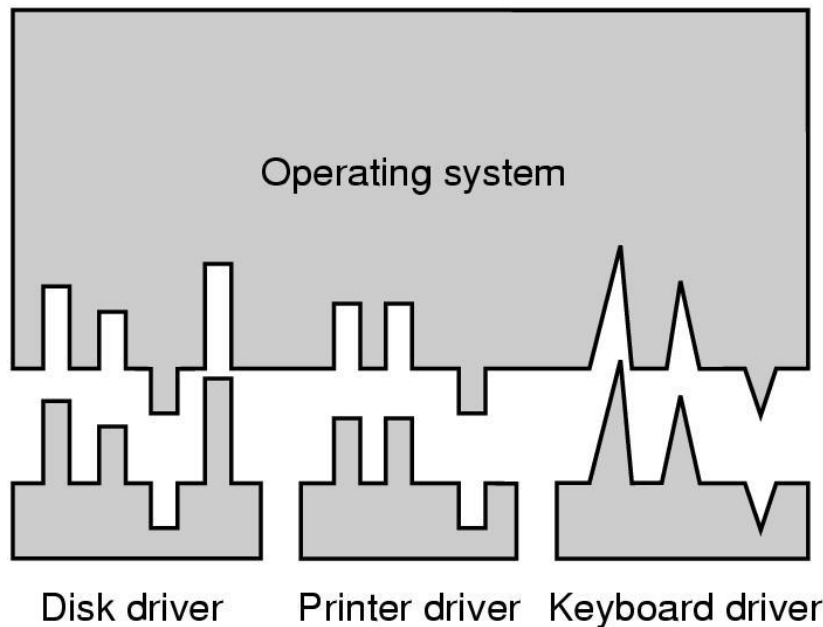
<b>Interface Uniforme para os drivers dos dispositivos (device drivers)</b>
<b>Armazenamento em Buffer</b>
<b>Relatórios de erros</b>
<b>Alocação e Liberação de dispositivos dedicados</b>
<b>Fornecimento de Tamanho de Bloco Independente de Dispositivo</b>

# Camadas do Software de E/S

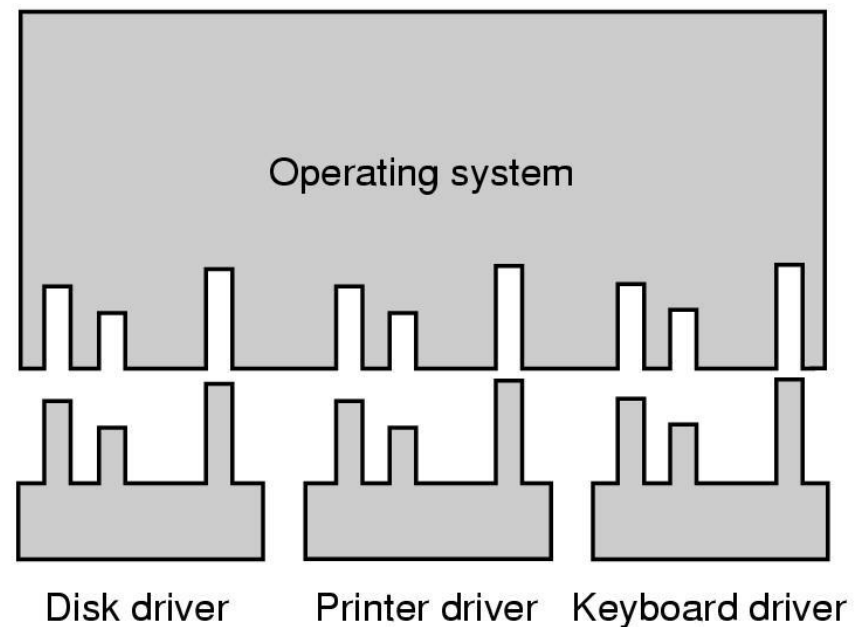
## Software de E/S independente do Dispositivo

---

- **Interface Uniforme para os drivers dos dispositivos (device drivers)**
- (a) Sem uma interface-padrão do driver;
- (b) Com uma interface padrão do dispositivo;



(a)



(b)

# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

---

- **Armazenamento em Buffer:**

- (a) Entrada sem utilização de buffer;
- (b) Utilização do buffer no espaço do usuário;
  - Problema: buffer paginado pode trancar a páginas na memória;
- (c) Utilização de buffer no núcleo seguido da cópia para o espaço do usuário
  - Problema: durante a cópia os caracteres que chegam não têm onde ser colocados;
- (d) Utilização de buffer duplicado no núcleo;
  - Para resolver o problema anterior;
  - Buffers trabalham alternadamente

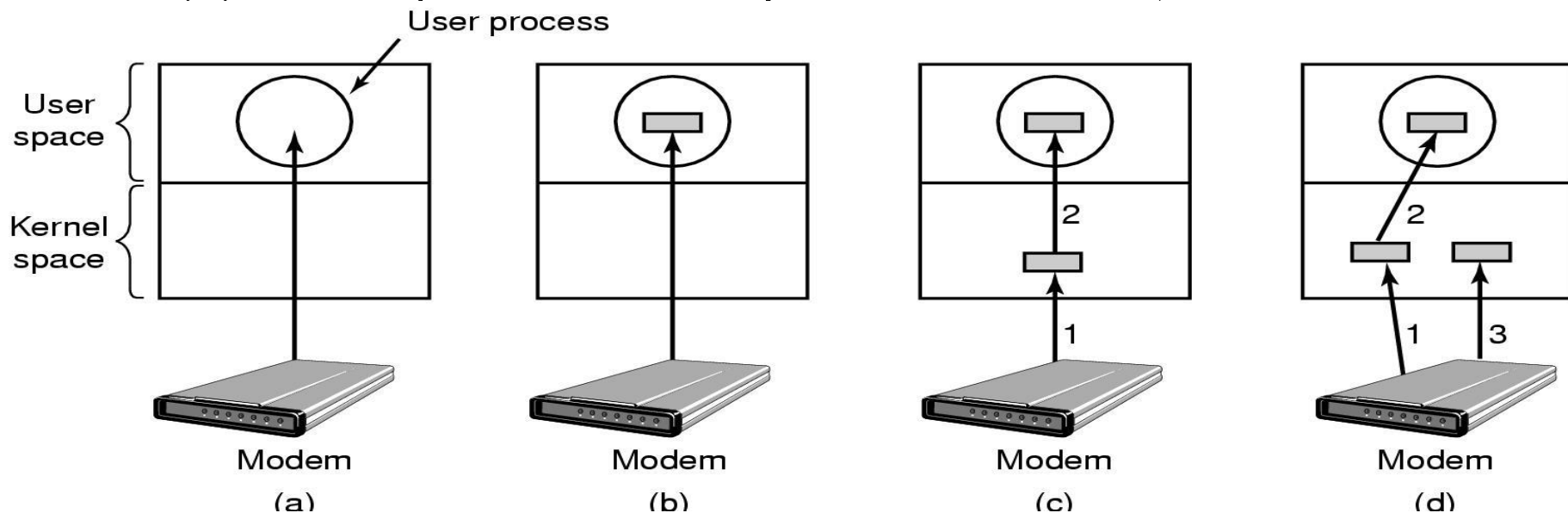
- Esquema: utilização de Buffer Duplicado

# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

- **Armazenamento em Buffer:**

- (a) Entrada sem utilização de buffer;
- (b) Utilização do buffer no espaço do usuário;
- (c) Utilização de buffer no núcleo seguido da cópia para o espaço do usuário
- (d) Utilização de buffer duplicado no núcleo;



# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

---

- **Relatórios de Erros:**

- Classe de erros relacionados aos **erros de programação**:
  - endereçamento errado de buffer ou dispositivo, etc;
- Classe de erros que engloba **erros reais de E/S**
  - Tentativa de escrever em um bloco de disco danificado ou ler de uma câmera de vídeo desligada;
- O que o software faz dependendo do **ambiente e natureza do erro**:
  - Exibir no vídeo uma caixa de dialogo perguntando o que fazer;
  - Ignorar o erro, Matar o processo, Desligar o Sistema;
  - Relatar um código de erro indicando uma falha de sistema;<sup>39</sup>

# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

---

- **Alocação e Liberação de dispositivos dedicados:**
  - Aceitar ou rejeitar acesso a dispositivos dedicados
    - Exemplo: CDROM
- **Maneira Simples:** processos abrir arquivos especiais que estão associados diretamente aos dispositivos;
- **Estratégia Alternativa:** mecanismos especiais para a requisição e liberação de dispositivos;
  - Processos bloqueados são colocados em fila



# Camadas do Software de E/S

## Software de E/S independente do Dispositivo

---

- **Fornecimento de Tamanho de Bloco Independente de Dispositivo:**

- Discos diferentes podem ter tamanhos diferenciados para os setores:
  - SW independente de dispositivo esconde esse detalhe e fornecer um tamanho de bloco uniforme para as camadas superiores;

# Camadas do Software de E/S

## Software de E/S no Espaço do Usuário

---

- **Através de bibliotecas:**

- Ligadas aos programas do usuário - procedimentos;
  - exemplos: write, printf, etc.

- **Sistema de Spooling**

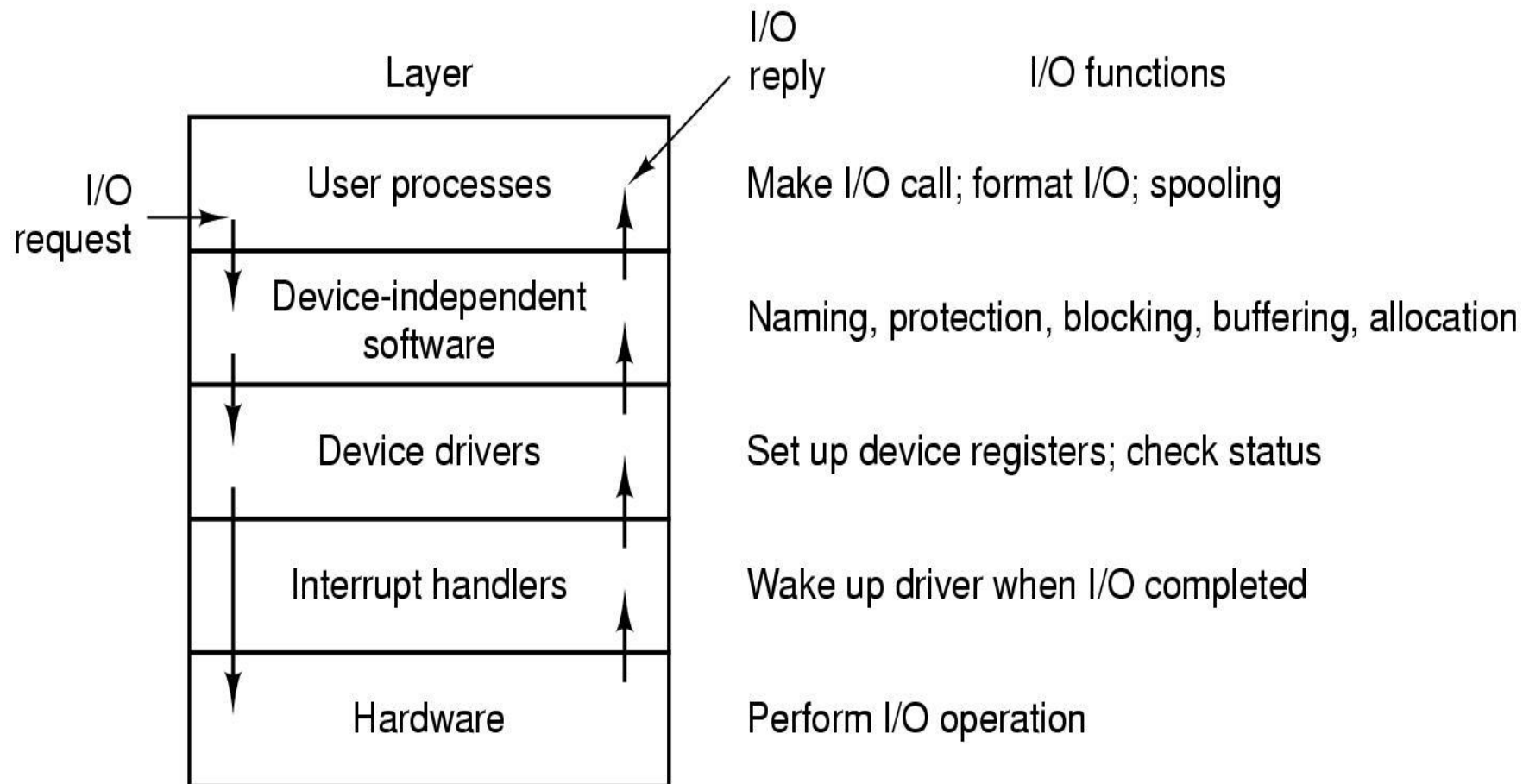
- Maneira de lidar com dispositivos dedicados de E/S em Sistemas de Multiprogramação;
- Utilização de um processo especial chamado de **daemon**, e um diretório especial;
  - arquivo a ser impresso colocado no diretório de spool pelo daemon;
  - Não só para impressão, tb para redes;

# Camadas do Software de E/S

## Software de E/S no Espaço do Usuário

---

- **Resumo das Camadas do Sistema de E/S e as funções principais de cada camada:**



# Discos

## Algoritmos de Escalonamento de Braço do Disco

---

- **Algoritmo de Escalonamento do Braço do Disco FCFS**

- **Primeiro a chegar, primeiro a ser servido**

- Quase nada pode ser feito para otimizar o tempo de posicionamento;

- Estratégia alternativa:

- Utilizar uma tabela onde as requisições pendentes para cada cilindro são encadeados pelo numero do cilindro;

- Exemplo:

- leitura do cilindro 11
- Novas requisições durante o posicionamento de 11
  - 1, 36, 16, 34, 9 e 12
- Uma finalizado 11, será 1, depois o 36, .....

# Discos

## Algoritmos de Escalonamento de Braço do Disco

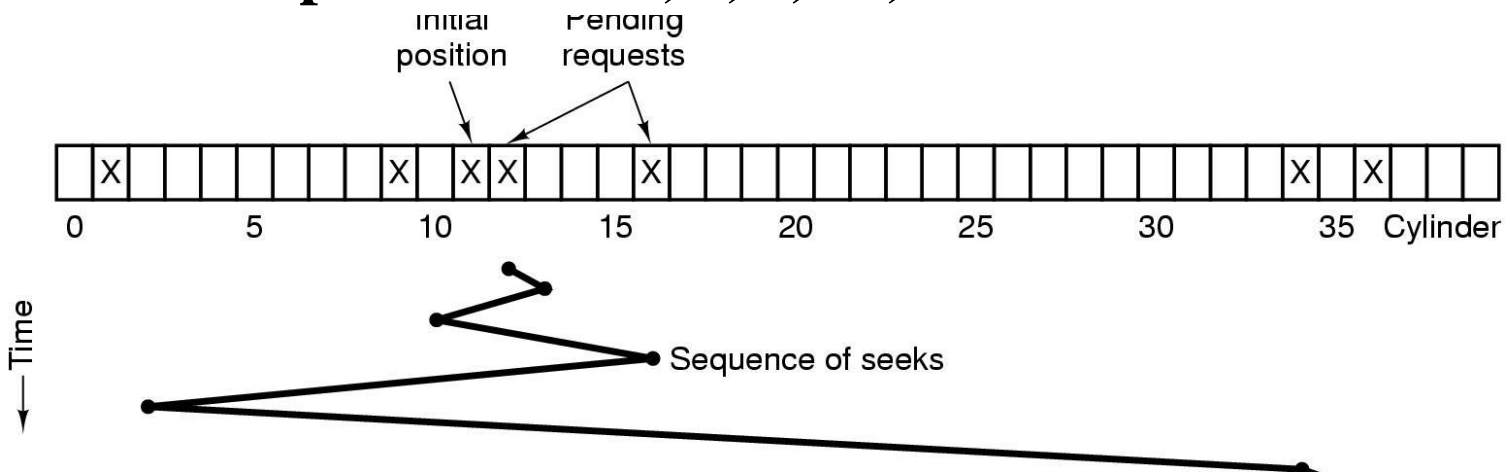
---

- **Algoritmo de Escalonamento do Braço do Disco FCFS**
  - **Primeiro a chegar, primeiro a ser servido**
  - Exemplo: Novas requisições durante o posicionamento de 11
    - 1, 36, 16, 34, 9 e 12
    - Uma finalizado 11, será 1, depois o 36, .....
    - Esse algoritmo posicionará o braço nos cilindros requisitados percorrendo 111 cilindros, ou seja,
      - $11 \rightarrow 1 = 10$
      - $1 \rightarrow 36 = 35$
      - $36 \rightarrow 16 = 20$
      - $16 \rightarrow 34 = 18$
      - $34 \rightarrow 9 = 25$
      - $9 \rightarrow 12 = \underline{3}$
      - $\underline{111}$

# Discos

## Algoritmos de Escalonamento de Braço do Disco

- **Algoritmo de Escalonamento do Braço do Disco SSF**
  - Menor Seek Primeiro – menor deslocamento
    - Coloca as requisições em ordem proximidade:
      - 12, 9, 16, 1, 34 e 36
      - percorrido: 1, 3, 7, 15, 33 e 2 = 67



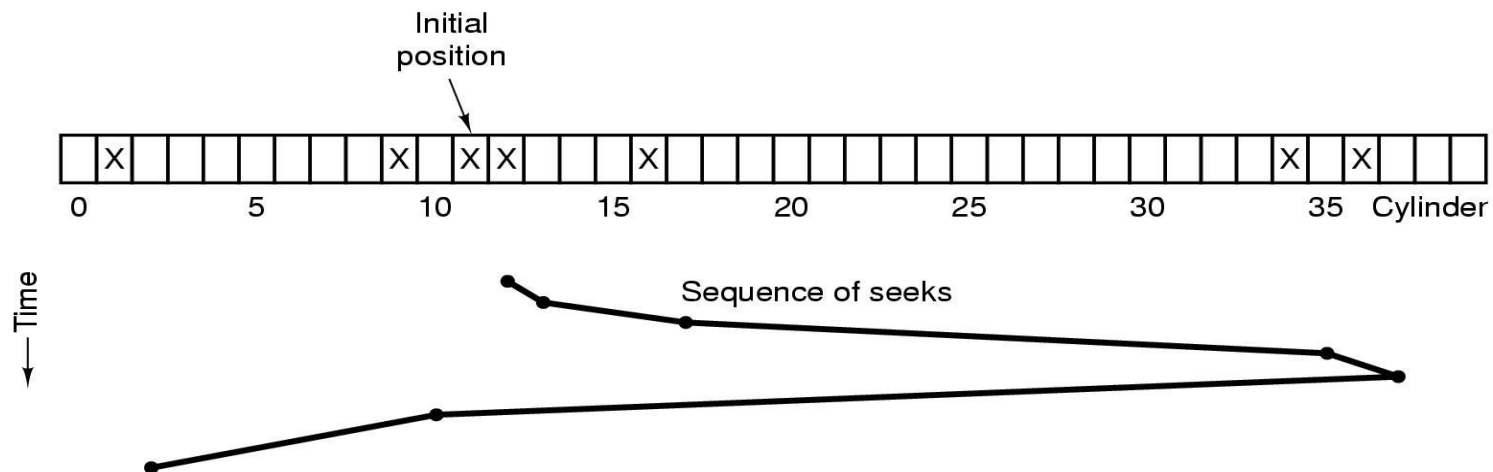
- **Problema:** dependendo da requisição o braço pode sempre ficar no centro do disco;

# Discos

## Algoritmos de Escalonamento de Braço do Disco

- **Algoritmo de Escalonamento do Braço do Disco - Elevador**

- Procura manter em uma direção até não haver mais requisições;
- Requisições: 12, 16, 34, 36, 9 e 1
- Total percorrido = 60 = 1, 4, 18, 2, 27 e 8



- Ligeiramente melhor que o anterior;

# Gerenciamento de Entrada/Saída

---

- **Referências Utilizadas:**

- Livro do Andrew S. Tanenbaum
  - *Sistemas Operacionais Modernos*
    - 2ª Ed Editora Prentice Hall, 2003
    - [www.cs.vu.nl/~ast](http://www.cs.vu.nl/~ast)
- Livro do Abraham Silberschatz et.al.
  - *Operating System Concepts*
    - [www.bell-labs.com/topic/books/aos-book/](http://www.bell-labs.com/topic/books/aos-book/)
  - *Sistemas Operacionais: Conceitos e Aplicações*
    - Editora Campus, 2001