

Sistemas Operacionais

Gerência de entrada e saída

Edson Moreno

edson.moreno@pucrs.br

<http://www.inf.pucrs.br/~emoreno>

Introdução

- Uma das funções do SO é:
 - Controlar os dispositivos de E/S
- Operações comumente realizadas
 - Tratamento de interrupções
 - Tratamento erros
 - “Interfaceamento” entre os dispositivos e o resto do sistema
 - Envio de comandos para os dispositivos/periféricos
 - Leitura / escrita dos entre dispositivo e sistema

Princípio de hardware de E/S

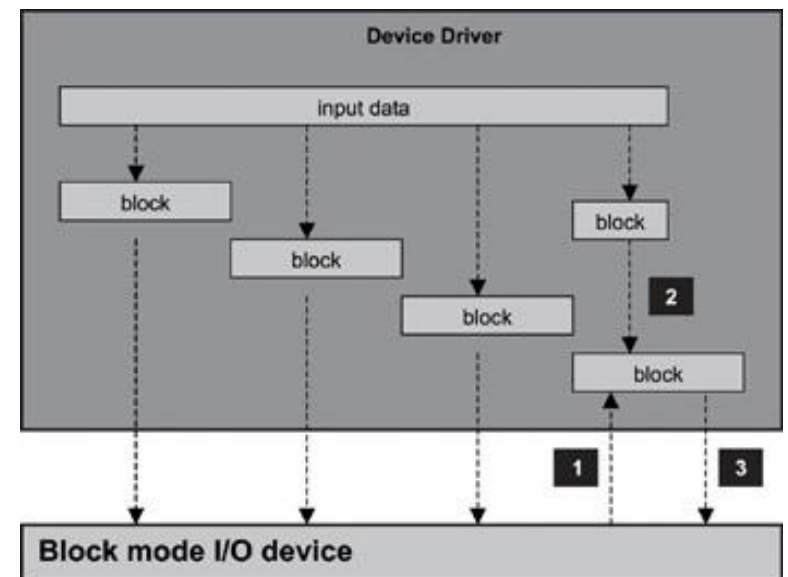
- Visões diferentes do HW de E/S:
 - Engenheiros elétricos:
 - Componentes de HW
 - chips, fios, fontes de alimentação, motores, etc
 - Programadores:
 - Interface apresentada para o SW
 - Comandos que o HW aceita
 - Funções que realiza
 - Erros que podem ser repassados ao software

Princípio de hardware de E/S

- Dispositivos de E/S
 - Genericamente dividido em duas categorias
 - Dispositivo de bloco
 - Dispositivo de caracter
 - Há dispositivos que não se encaixam em nenhuma das categorias
 - Relógio
 - Causam interrupções em intervalos definidos

Princípio de hardware de E/S

- Dispositivo de bloco
 - Armazena informações em blocos de tamanho fixo
 - Cada bloco com seu próprio endereço
 - Tamanho típico de bloco: 512 a 32768 Bytes
 - Manipulação de blocos ocorre de forma independente
 - Transferência
 - Ocorrem em unidades de blocos
 - Um ou mais
- Exemplo:
 - Disco, Floppy, Pendrive, CD-ROM



Princípio de hardware de E/S

- Dispositivos de caracter
 - Dispositivo envia e recebe uma sequência de caracteres
 - Não considera qualquer estrutura de blocos.
 - Não é endereçável
 - Não explora operações de posicionamento
 - Exemplos:
 - Impressora, interface de rede, teclado, mouse

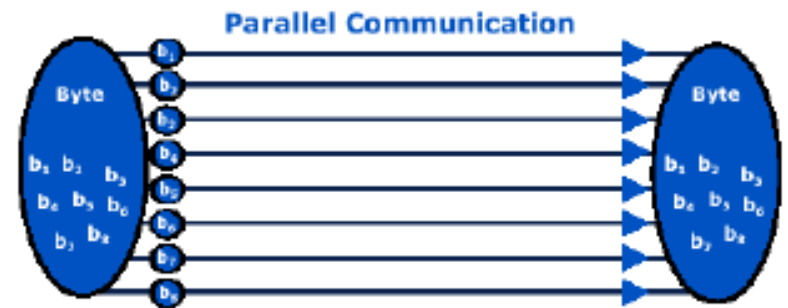
Princípio de hardware de E/S

- Velocidade dos periféricos é variável
 - Software deve garantir bom funcionamento com diferentes taxas
 - Dispositivos apresentam melhora de desempenho com passar do tempo
 - Dispositivo não pode ser o gargalo do sistema computacional

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

Princípio de hardware de E/S

- Comunicação com dispositivos
 - Interface:
 - Utilizado para conectar um dispositivo/periféricos a um computador
 - Interconectadas aos barramentos internos do computador
 - Tipos de conexão
 - Serial: 1 linha para transmissão de dados
 - Paralelo: várias linhas para transmissão de dados

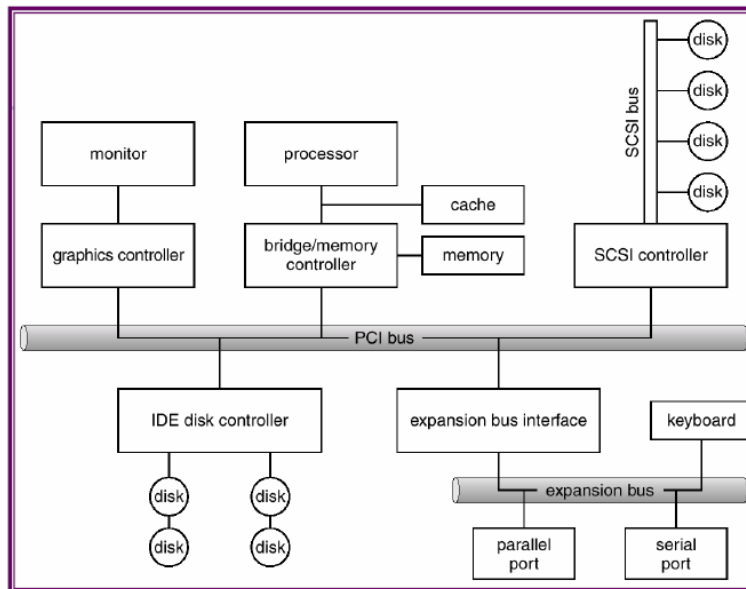


Princípio de hardware de E/S

- Unidade de E/S formado por
 - Componente mecânico
 - Componente eletrônico
- Componente eletrônico
 - Controladores de Dispositivos ou adaptador
- Tarefas da controladora
 - Converte fluxo serial de bits em bloco de bytes
 - Traduz operações genéricas (leitura / escrita)
 - Gerar seqüência de acionamentos para manipulação do dispositivo alvo
 - Realizar correções de erro se necessário

Princípio de hardware de E/S

- Controladora
 - Normalmente possui interface padrão
 - Exemplo de órgãos normatizadores: IEEE, ANSI, ISSO
 - Exemplos de interfaces padronizadas: IDE, SCSI, USB, Firewire
 - Alguns são capazes de tratar mais de um dispositivo idêntico
 - Distintos padrões podem conviver em um mesmo sistema computacional

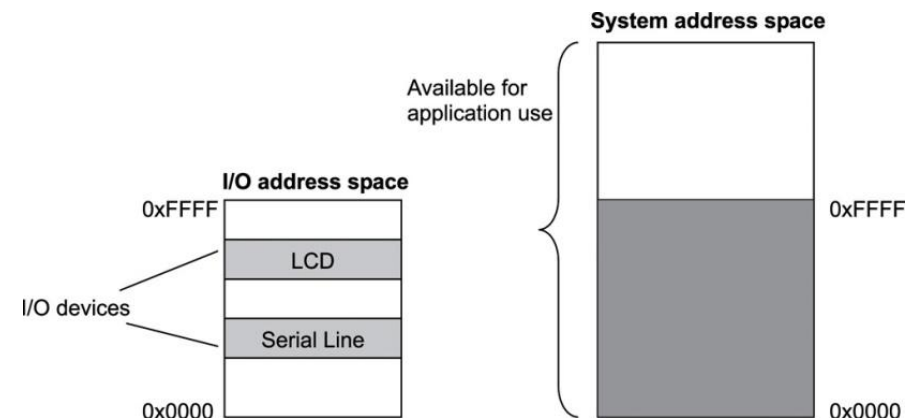
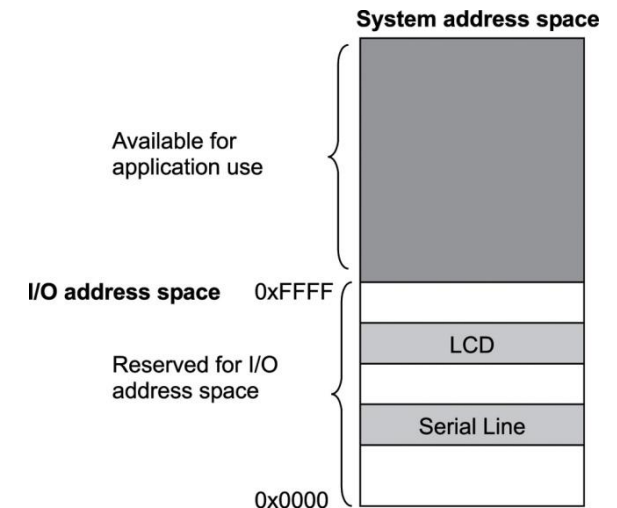


Princípio de hardware de E/S

- Formas de acesso as controladoras
 - Comunicação com a CPU feita a partir de
 - **Registadores da controladora**
 - Escrita nestes registadores permite o envio de comandos
 - Entrega / aceitação de dados,
 - Ligar / desligar dispositivo
 - Executar operação específica
 - Leitura destes registadores permite capturar estado
 - Ligado / desligado
 - Ocupado / disponível
 - **Buffers de dados da controladora**
 - Meio acessível por programas e pelo SO

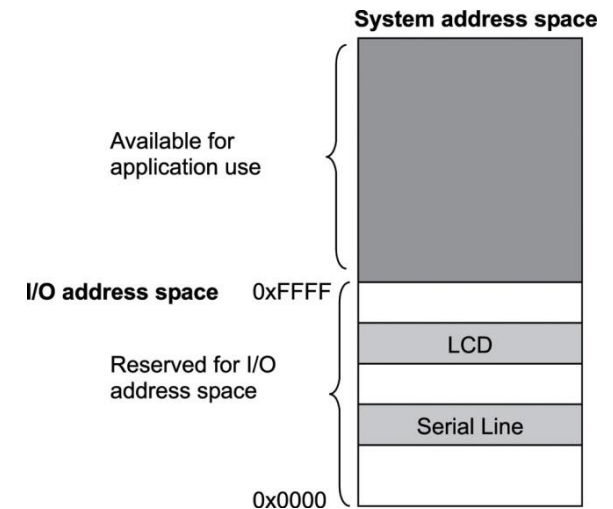
Acesso aos dispositivos

- Formas de acesso
 - Mapeamento em espaço de memória
 - Definição de uma região da memória utilizada pelos dispositivos de E/S
 - Mapeamento em espaço de E/S
 - Definição de um espaço de memória separado utilizado pelos dispositivos de E/S



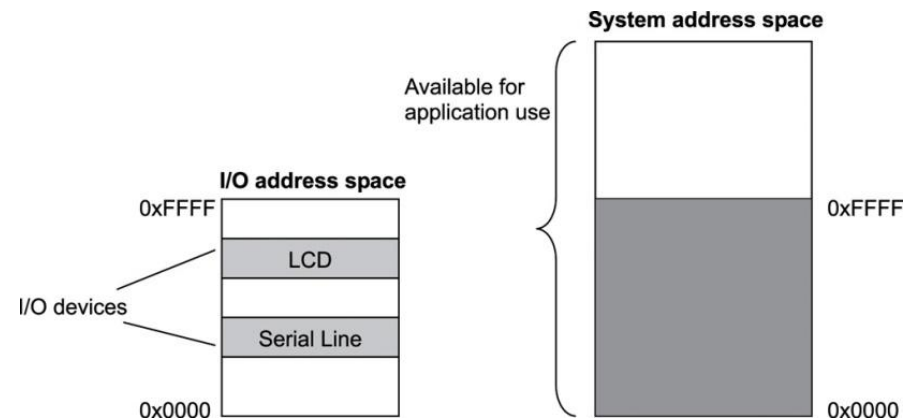
Acesso aos dispositivos

- Formas de acesso às controladoras
 - Mapeamento em espaço de memória
 - Mapeia os registradores em espaço de endereçamento de memória
 - Cada registrador da controladora é associado com um endereço de memória único onde não há memória associada.
 - Geralmente associados no topo da memória
 - Acesso utilizando instruções especiais para manipulação de E/S



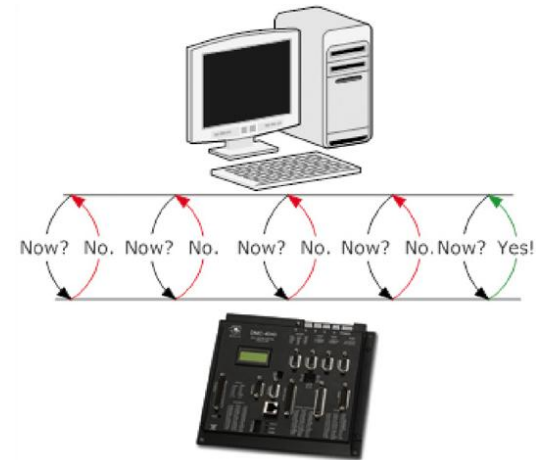
Acesso aos dispositivos

- Formas de acesso as controladoras
 - Mapeamento em espaço de E/S
 - Cada registrador de controle é associado a um número de porta de E/S
 - Conjunto de portas forma o espaço de portas de E/S
 - Somente o SO pode acessá-lo
 - Normalmente protegido de programas do usuário
 - Acesso utilizando instruções especiais para manipulação de E/S
 - IN REG, PORT
 - OUT PORT, REG



Acesso aos dispositivos

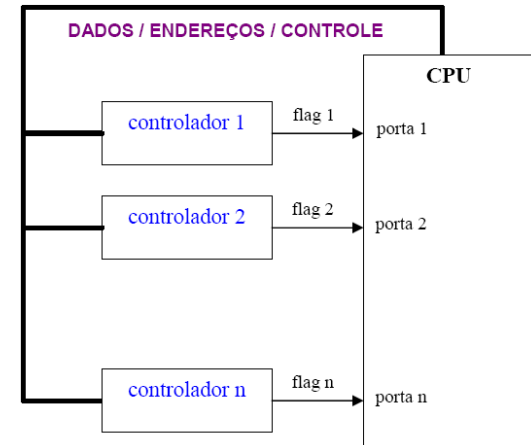
- Técnicas para acesso aos dispositivos de E/S
 - Interação processador-dispositivo
 - E/S programada
 - Polling, Interjeição
 - Interrupções:
 - Interrompe o processador e chama tratador de interrupção
 - Acesso direto à memória (DMA):
 - Transferência de dados da interface para memória
 - Controladora de DMA



E/S Programada

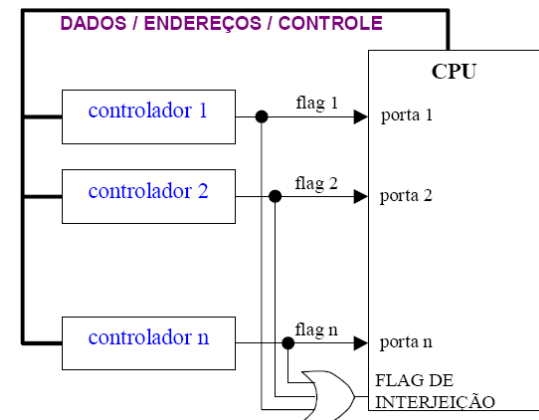
- Polling

- CPU possui controle total
 - Instantes de tempo em que ocorrerão definidos pelo programa/CPU
 - Dispositivos são testados periodicamente para leitura de estado



- Interjeição

- Uma verificação geral é realizada antes de observar cada pedido
 - Reduz o tempo de teste se necessário

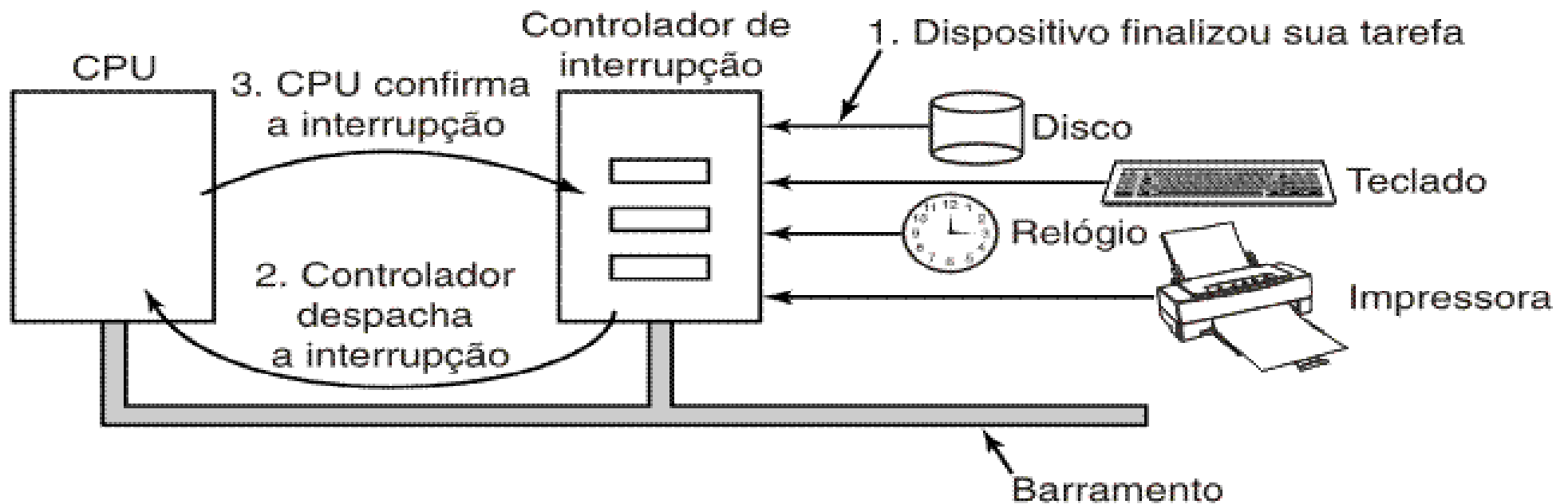


Interrupção

- Funcionamento da interrupção
 - Finalização de uma tarefa por um dispositivo gera interrupção
 - Sinal representado por um sinal dedicado
 - Sinal de interrupção concentrado em um controlador de interrupção
 - Dispositivos enviam sinal via barramento
 - Cada dispositivo possui uma identificação de interrupção (IRQ)
 - Controlador de interrupção
 - Processa a interrupção conforme disponibilidade/prioridade
 - Controlador repassa sinal para interromper a CPU
 - Coloca um número nas linhas de endereço do barramento
 - CPU
 - Número das linhas de endereço definem índice do vetor de interrupções
 - Vetor de interrupções: Contem o endereço base da rotina de tratamento

Acesso aos dispositivos

- Interrupções



Exemplo

Interrupção: Controladoras PC/MS-DOS

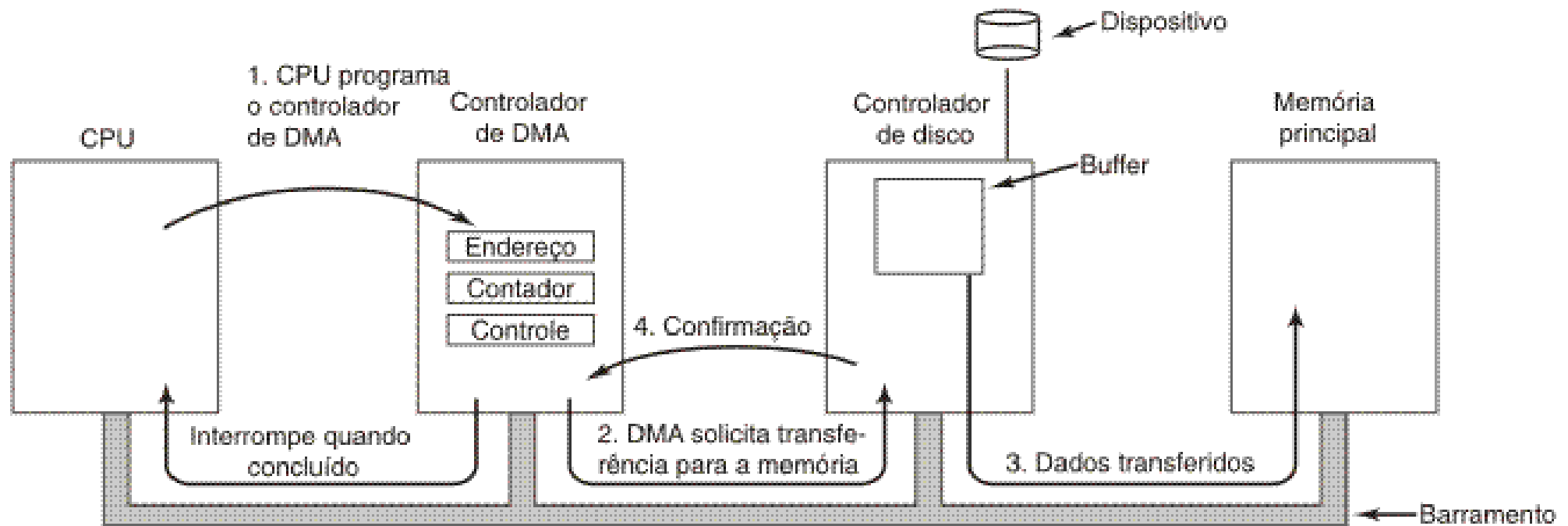
<i>Controladora</i>	<i>Endereço</i>	<i>IRQ</i>	<i>Vetor de interrupção</i>
Relógio	040 - 043	0	8
Teclado	060 - 063	1	9
Disco Rígido	1F0 - 1F7	14	118
RS232 secundário	2F8 - 2FF	3	11
Impressora	378 - 37F	7	15
Disquete	3F0 - 3F7	6	14
RS232 primário	3F8 - 3FF	4	12

Acesso aos dispositivos

- DMA
 - Dispositivo tem acesso direto à memória
 - Bypass do processador
 - Exemplo: leitura de arquivo do disco
 - sem DMA
 - processador recebe interrupção indicando que dados estão no buffer da controladora
 - rotina realiza cópia do bloco byte a byte a partir do registrador da controladora para memória
 - CPU executa processo de cópia
 - com DMA
 - processador configura controladora: endereço de memória para enviar o bloco e quantidade de bytes para transferir
 - quando acabar transmissão, é gerada uma interrupção

Acesso aos dispositivos

- DMA
 - Fluxo de operação de leitura



Gerenciamento de Entrada e Saída

- Software de manipulação de E/S
 - Maneira pela qual o SW programado irá lidar com o HW.
 - Diferentes formas de tratar as diversas situações apresentadas pelas implementações de HW.

Gerenciamento de Entrada e Saída

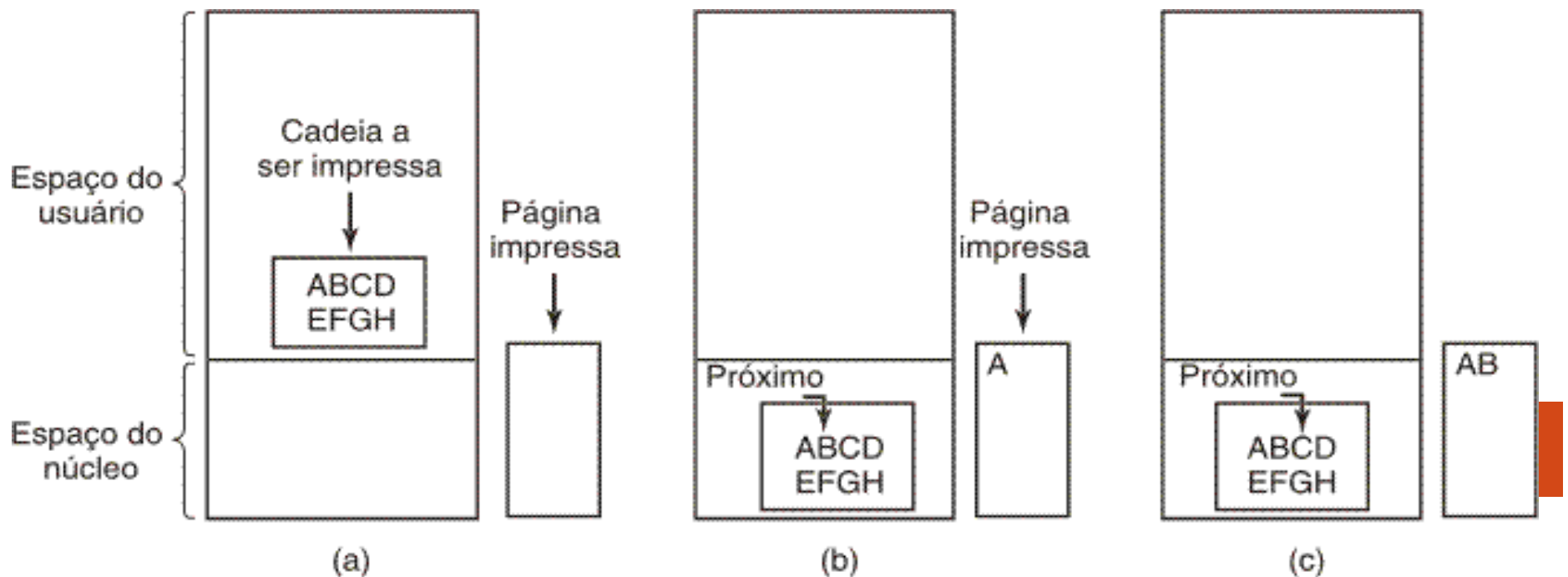
- Objetivos do software de E/S:
 - *Independência de dispositivo*
 - Programas podem acessar qualquer dispositivo de E/S sem especificar previamente qual (ex: disco / pendrive / etc)
 - *Nomeação uniforme*
 - Nome/caminho de arquivo ou dispositivo deveria ser uma cadeia de caracteres ou um número inteiro independente do dispositivo
 - *Tratamento de erros*
 - Erros deveriam ser tratados em nível hardware / driver
 - *Transferências síncronas vs. assíncronas*
 - Transferências bloqueantes vs. orientadas à interrupção.
 - *Utilização de buffer para armazenamento temporário*
 - Recepção de pacotes da rede a serem tratados tardiamente
 - *Gerenciamento de dispositivos compartilhados vs. Dedicados*

Gerenciamento de Entrada e Saída

- Formas de realizar E/S:
 - E/S Programada
 - Processo fica esperando (busy-waiting) pelo término da operação
 - E/S dirigida por interrupção
 - Comando de E/S é emitido
 - Processador continua execução
 - Módulo de E/S envia interrupção quando pronto
 - E/S usando DMA
 - Driver de E/S programa controlador DMA
 - Processador não participa da transferência

Gerenciamento de Entrada e Saída

- E/S Programada:
 - Espera ociosa ou *polling*.
 - Simples.
 - Segura a CPU o tempo todo até que a E/S seja executada.



Gerenciamento de Entrada e Saída

- E/S Programada:
 - CPU faz todo o trabalho
 - Dados são copiados para o núcleo
 - Envio de caractere feito a cada iteração do laço
 - CPU avalia a condição de envio de cada caractere até o final da cadeia

```
copy_from_user(buffer, p, count);  
for (i = 0; i < count; i++) {  
    while (*printer_status_reg != READY) ;  
    *printer_data_register = p[i];  
}  
return_to_user();
```

/* p is the kernel bufer */
/* loop on every character */
/* loop until ready */
/* output one character */

Gerenciamento de Entrada e Saída

- E/S Orientada à Interrupção:
 - Buffer é copiado para o espaço do núcleo do sistema operacional.
 - Primeiro caractere é copiado para a impressora tão logo ela concorde em aceitá-lo.
 - CPU chama o escalonador e outro processo é executado.
 - Processo que solicitou a impressão é bloqueado até que toda a cadeia seja impressa.

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

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

Gerenciamento de Entrada e Saída

- E/S Orientada à Interrupção:
 - Quando impressora estiver pronta para receber o próximo caractere para impressão, ela gera uma interrupção.
 - A interrupção para o processo atual e salva seu estado.
 - A rotina de tratamento de interrupção da impressora é executada.

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

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

Gerenciamento de Entrada e Saída

- E/S Orientada à Interrupção:
 - Se não existem mais caracteres para imprimir
 - O usuário solicitante é desbloqueado
 - Se existem
 - O tratador de interrupção
 - Envia o próximo caractere
 - Confirma a interrupção
 - Retorna para o processo que estava executando antes da interrupção

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

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

Gerenciamento de Entrada e Saída

- E/S usando DMA:
 - Controlador de DMA alimenta os caracteres para a impressora um por vez, sem que a CPU seja perturbada.
 - DMA executa E/S programada, com o controlador de DMA fazendo todo o trabalho.
 - Reduz número de interrupções de uma por caractere para uma por buffer.

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

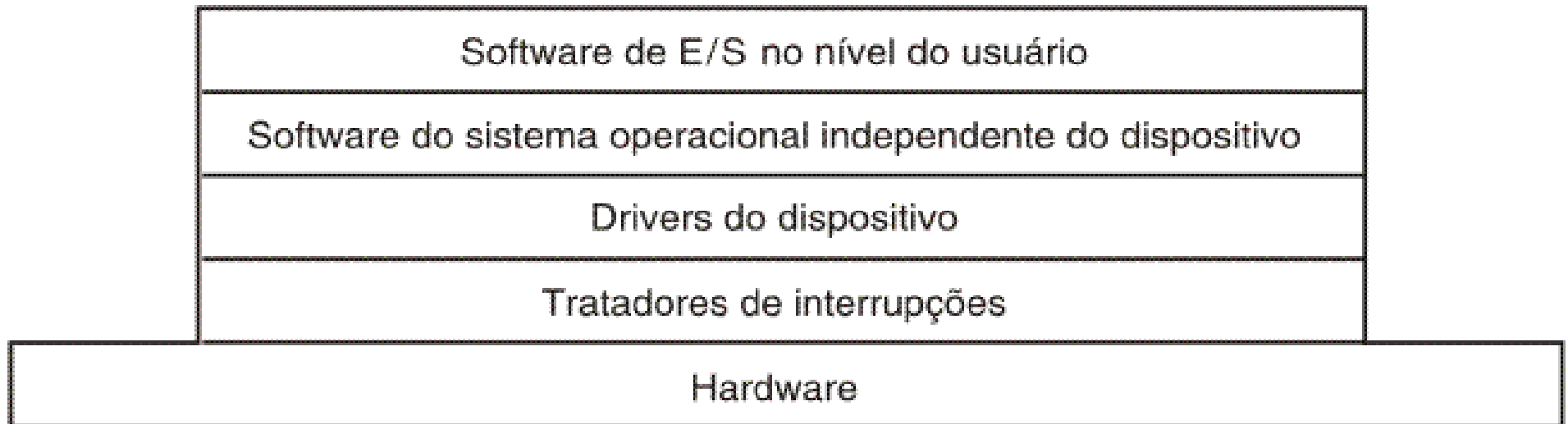
(a)

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

(b)

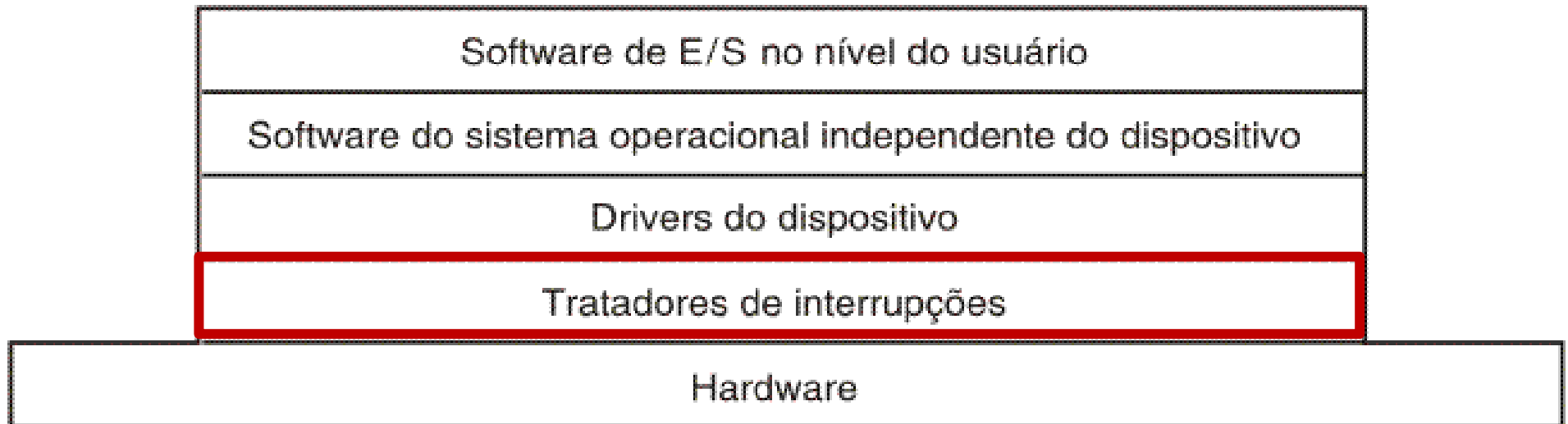
Gerenciamento de Entrada e Saída

- Camadas do SW de E/S:
 - Organização convencional dos softwares de entrada e saída
 - Cada camada possui função específica
 - Funcionalidades e interface difere de SO para SO



Gerenciamento de Entrada e Saída

- Camadas do SW de E/S:
 - Tratadores de interrupção



Gerenciamento de Entrada e Saída

- Tratadores de Interrupção
 - Rotina responsável por
 - Acomodação de processo em operação no processador
 - Remove do processador para tratamento da interrupção
 - Desvio para rotina de tratamento de interrupção
 - Normalmente posicionada em posição fixa da memória
 - Determinar a causa da interrupção
 - Em linhas gerais
 - CPU
 - Captura uma sinalização de interrupção
 - Despacha para o tratador de interrupção
 - Tratador de interrupção
 - Limpa a interrupção, atendendo ao dispositivo

Gerenciamento de Entrada e Saída

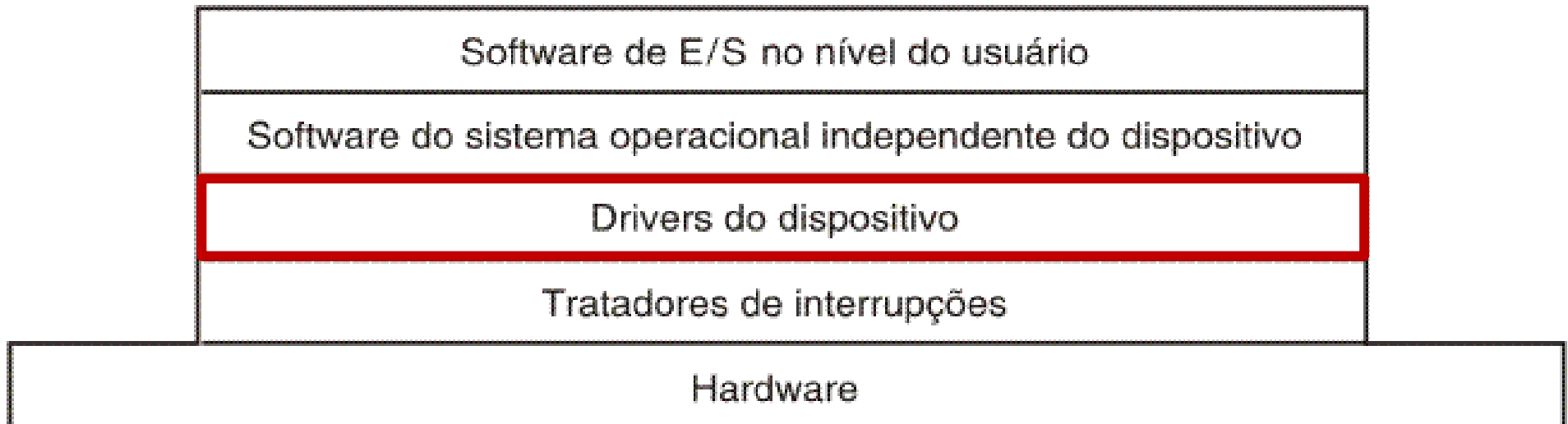
- Tratadores de Interrupção
 - Bloqueiam *driver* que iniciou operação de E/S até que a E/S se complete e uma interrupção ocorra
 - Quando ocorre interrupção, uma rotina específica a trata e em seguida desbloqueia o driver que a chamou
 - Salva registradores ainda não salvos pelo hardware de interrupção
 - Estabelece um contexto para a rotina de tratamento da interrupção
 - Estabelece uma pilha para a rotina de tratamento da interrupção
 - Sinaliza o controlador de interrupção

Gerenciamento de Entrada e Saída

- Tratadores de Interrupção:
 - Copia os registradores de onde eles foram salvos para a tabela de processos
 - Executa a rotina de tratamento de interrupção.
 - Escolhe o próximo processo a executar.
 - Estabelece o contexto da MMU para o próximo processo a executar.
 - Carrega os registradores do novo processo.
 - Inicia a execução do novo processo.

Gerenciamento de Entrada e Saída

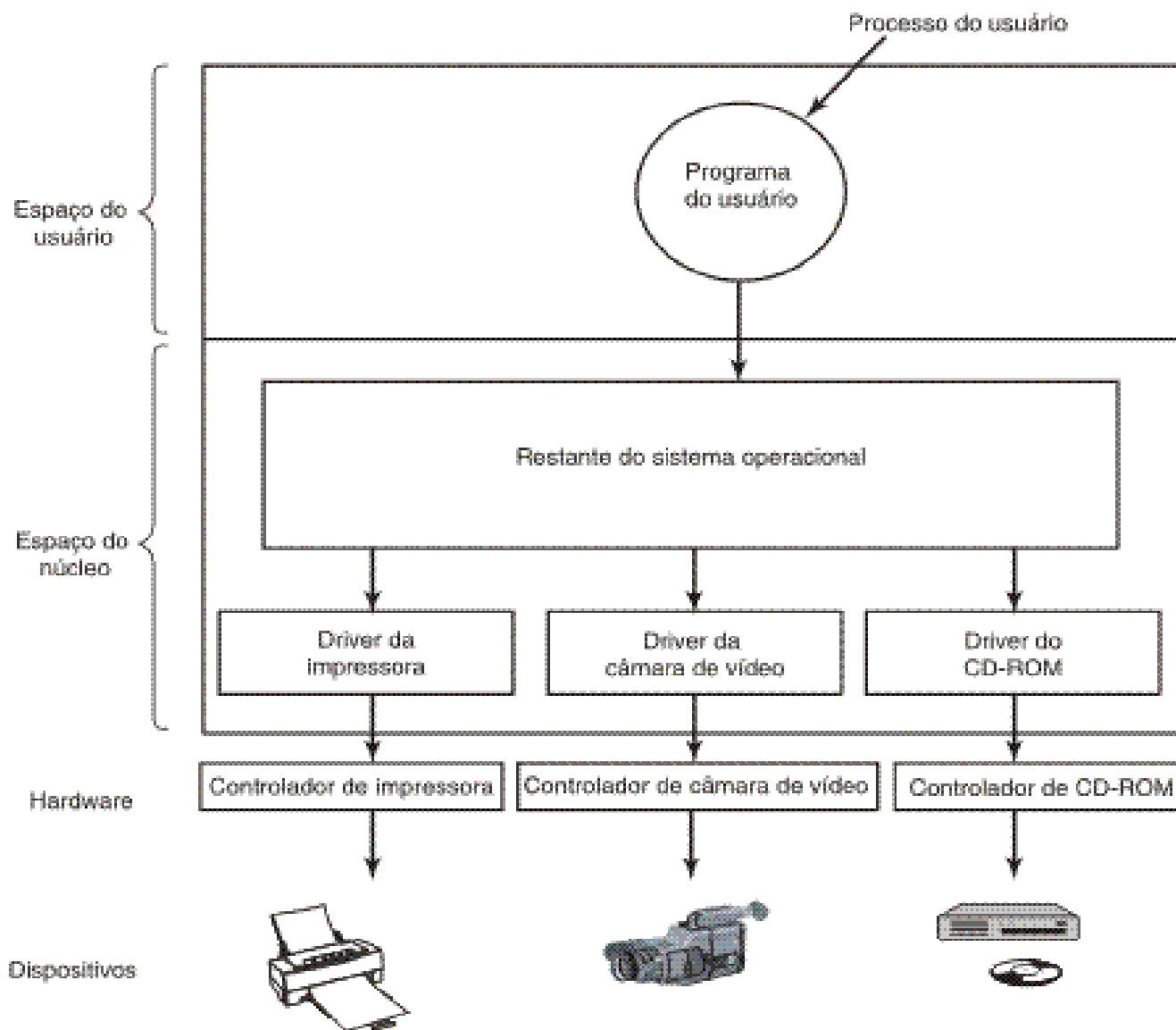
- Camadas do SW de E/S:
 - Drivers do dispositivo



Gerenciamento de Entrada e Saída

- *Drivers* dos Dispositivos:
 - Código específico do dispositivo para controlá-lo.
 - Diversidade de dispositivos leva a tal especificidade
 - Escrito pelo fabricante do dispositivo e fornecido juntamente com o dispositivo.
 - Para acessar os registradores do controlador, driver do dispositivo normalmente deve fazer parte do núcleo do sistema operacional
 - Dispositivos de bloco
 - Contêm vários blocos de dados que podem ser endereçados independentemente.
 - Dispositivos de caractere
 - Geram ou aceitam uma sequência de caracteres.

Gerenciamento de Entrada e Saída



Gerenciamento de Entrada e Saída

- *Drivers* dos Dispositivos:
 - Interfaces-padrão para drivers de blocos e de caracteres
 - Procedimentos que o restante do sistema operacional utiliza para fazer o driver trabalhar
- Funções:
 - Aceita e executa requisições abstratas de leitura ou gravação de um SW independente de dispositivo localizado na camada superior.
 - Inicia o dispositivo
 - Trata necessidades de energia
 - Registra eventos

Gerenciamento de Entrada e Saída

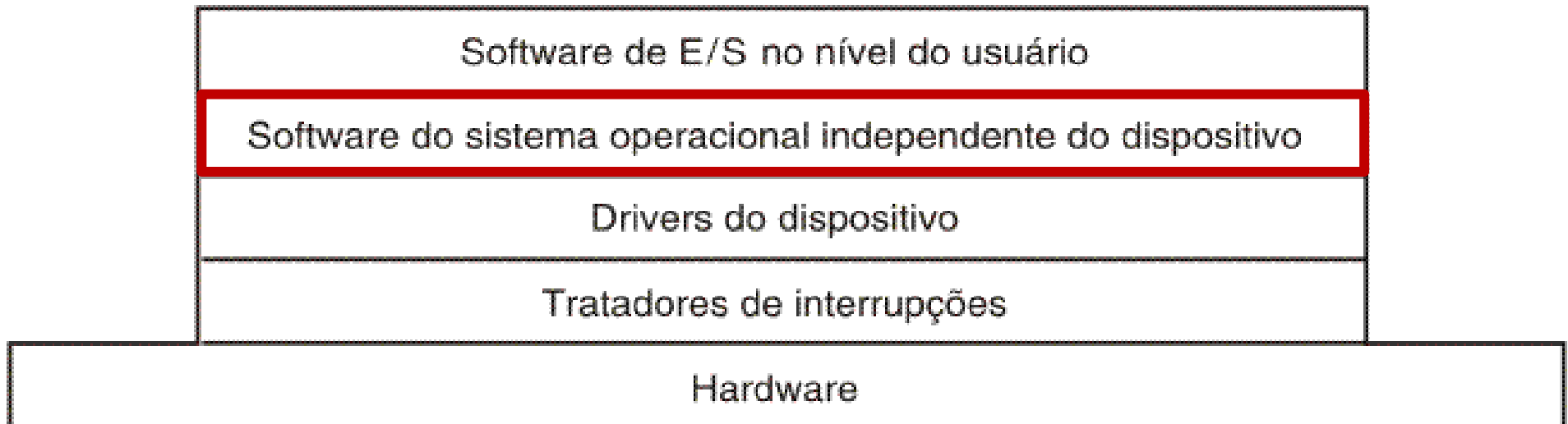
- *Drivers* dos Dispositivos:
 - *Driver* inicia verificando parâmetros de entrada para ver se são válidos
 - Se não, um erro é retornado.
 - Caso sim, uma tradução é feita.
 - *Driver* verifica se dispositivo está em uso
 - Caso sim, requisição é enfileirada.
 - Caso não, controle pode começar.

Gerenciamento de Entrada e Saída

- *Drivers* dos Dispositivos:
 - Controlar o dispositivo consiste em emitir uma sequência de comandos para ele.
 - Driver é o local onde a sequência de comandos é determinada.
 - Após saber quais comandos emitir, eles são escritos nos registradores do controlador do dispositivo.
 - Após escrever cada comando para o controlador, verifica se controlador aceitou o comando e está preparado para receber o próximo.
 - Sequência continua até todos os comandos serem emitidos.
 - Deve ser reentrante
 - Antes de finalizar um uma primeira chamada, uma segunda pode ser iniciada.

Gerenciamento de Entrada e Saída

- Camadas do SW de E/S:
 - Software do SO independente do dispositivo



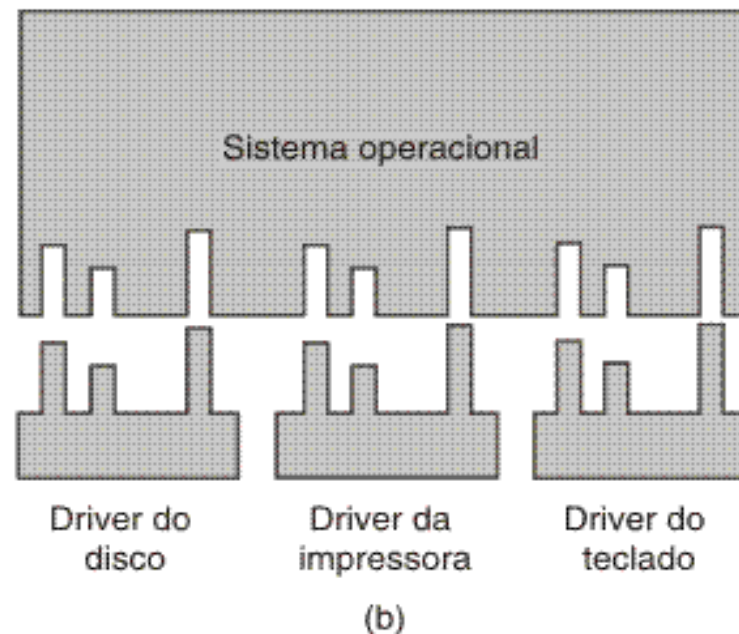
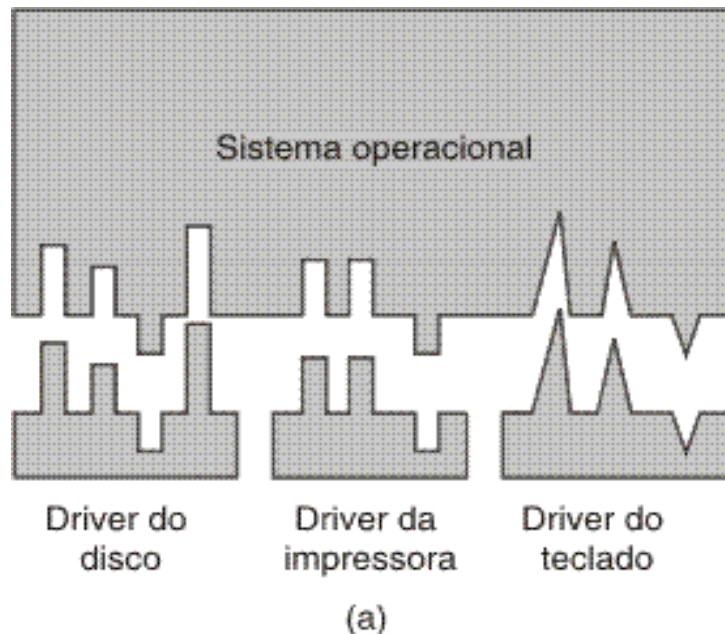
Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo (funções):

Interface uniforme para os <i>drivers</i> dos dispositivos
Armazenamento em <i>buffer</i>
Relatório dos erros
Alocação e liberação de dispositivos dedicados
Tamanho de bloco independente de dispositivo

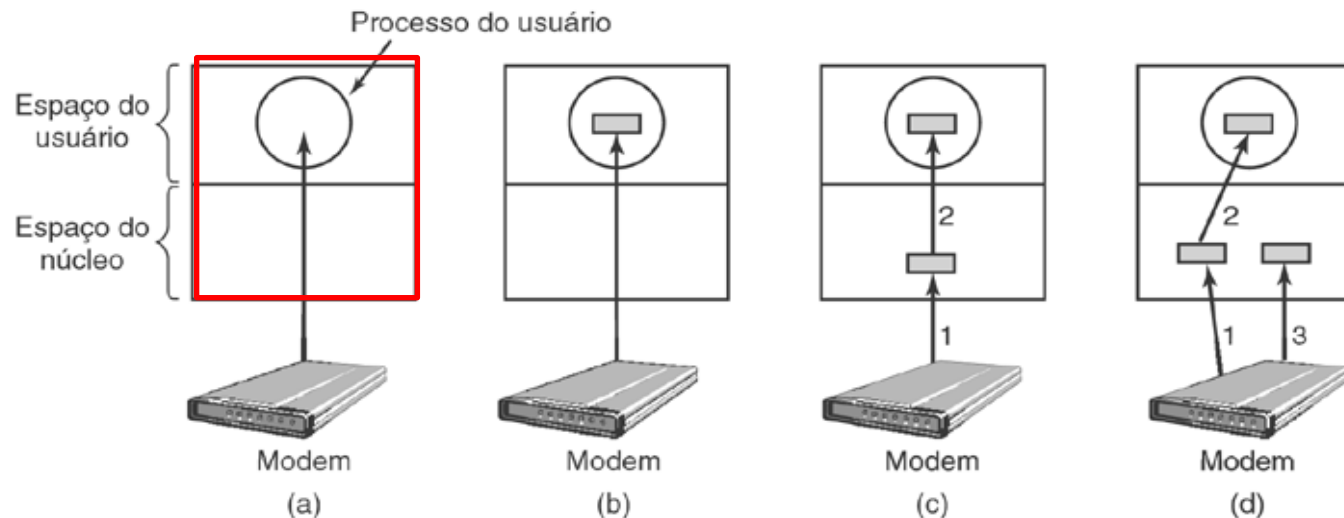
Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Interface uniforme para os drivers dos dispositivos
 - Objetivo é evitar retrabalho para cada novo dispositivo
 - Funcionalidades são agrupadas por classes de dispositivo



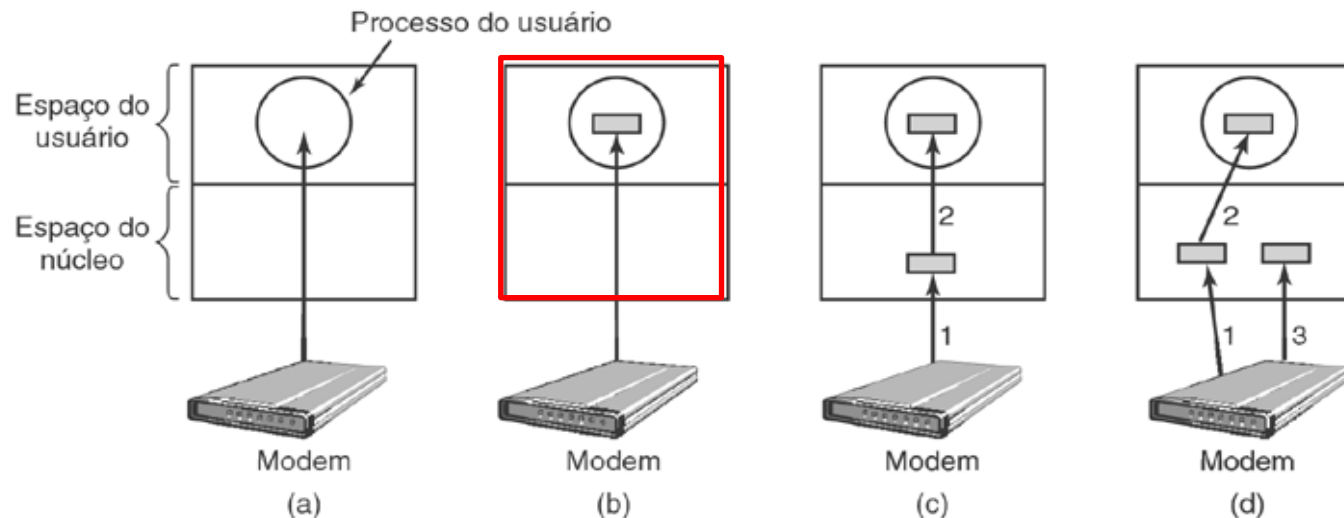
Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Utilização de buffer: entrada de dados.
 - Leitura de dado de um modem (A)
 - Usuário faz uma chamada de leitura e bloqueia até receber
 - À chegada de cada caractere é gerada uma interrupção de tratamento
 - O tratamento da interrupção passa o caractere ao usuário e desbloqueia



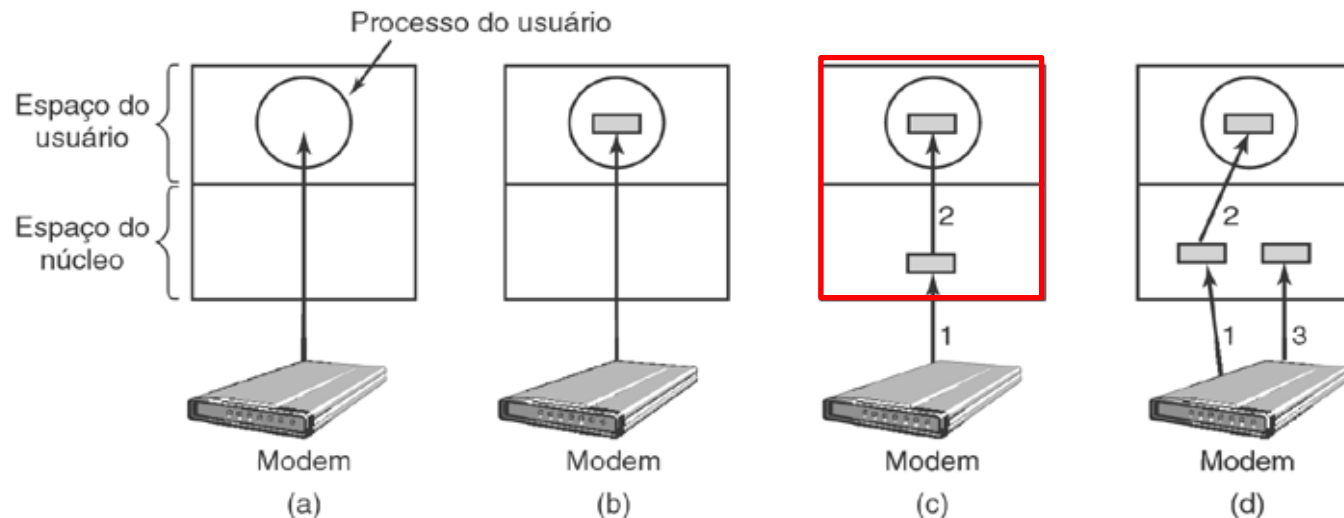
Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Utilização de buffer: entrada de dados.
 - Leitura de dado de um modem (B)
 - A partir da disponibilização de uma área no buffer em nível de usuário
 - Do preenchimento do buffer, o processo é desbloqueado
 - Pode causar perda de desempenho se buffer for mapeado em disco



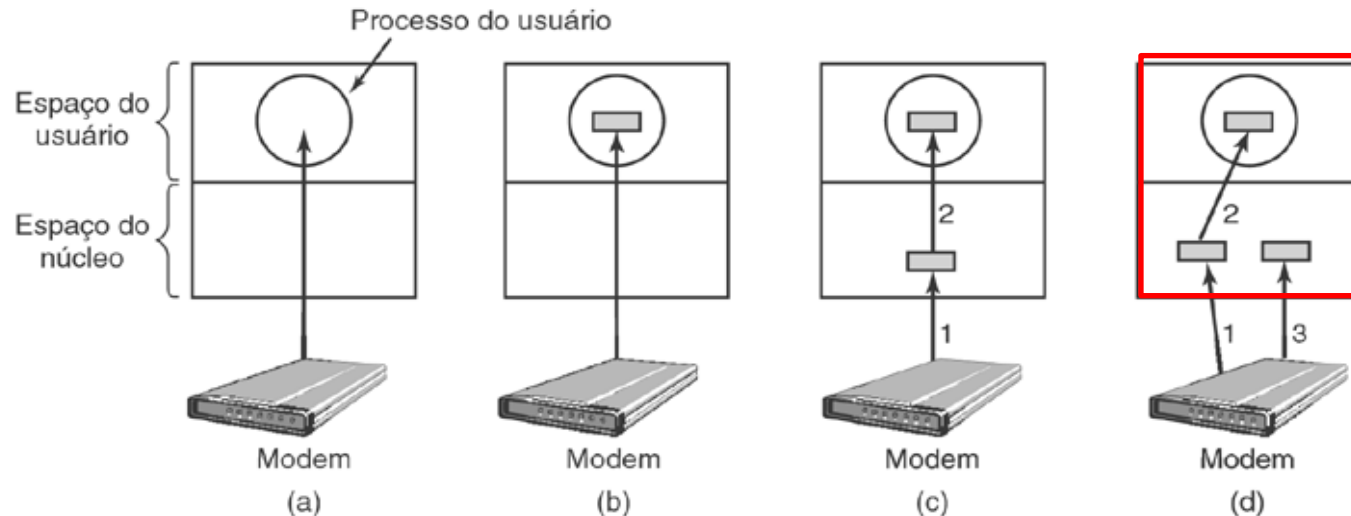
Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Utilização de buffer: entrada de dados.
 - Leitura de dado de um modem (C)
 - Núcleo cria área intermediária
 - Bloco é transferido em um a única operação
 - Mas o que ocorre quando o buffer está sendo transferido e novos caracteres são recebidos?



Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Utilização de buffer: entrada de dados.
 - Leitura de dado de um modem (D)
 - Buffer duplo permite a transferência e recepção em um mesmo instante de tempo

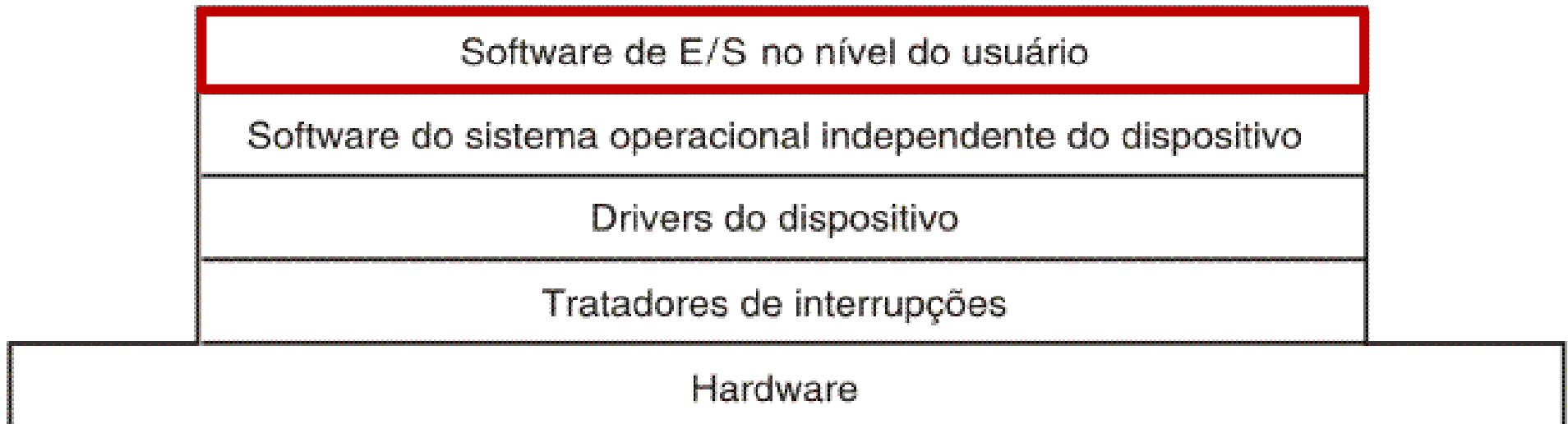


Gerenciamento de Entrada e Saída

- SW de E/S Independente de Dispositivo:
 - Relatório de erros:
 - Erros de programação – sistema operacional relata para o processo chamador um código de erro.
 - Erros reais de E/S – driver determina o que fazer ou repassa o problema para o software independente de dispositivo.
 - Alocação e liberação de dispositivos dedicados:
 - Tentativa de utilização de dispositivo indisponível causa uma falha na chamada ao sistema.
 - Tentativa de utilização de dispositivo indisponível bloqueia o processo chamador.
 - Tamanho de bloco independente de dispositivo:
 - Software independente de dispositivo deve esconder detalhes de discos com tamanhos diferenciados para os setores, fornecendo um tamanho de bloco uniforme.

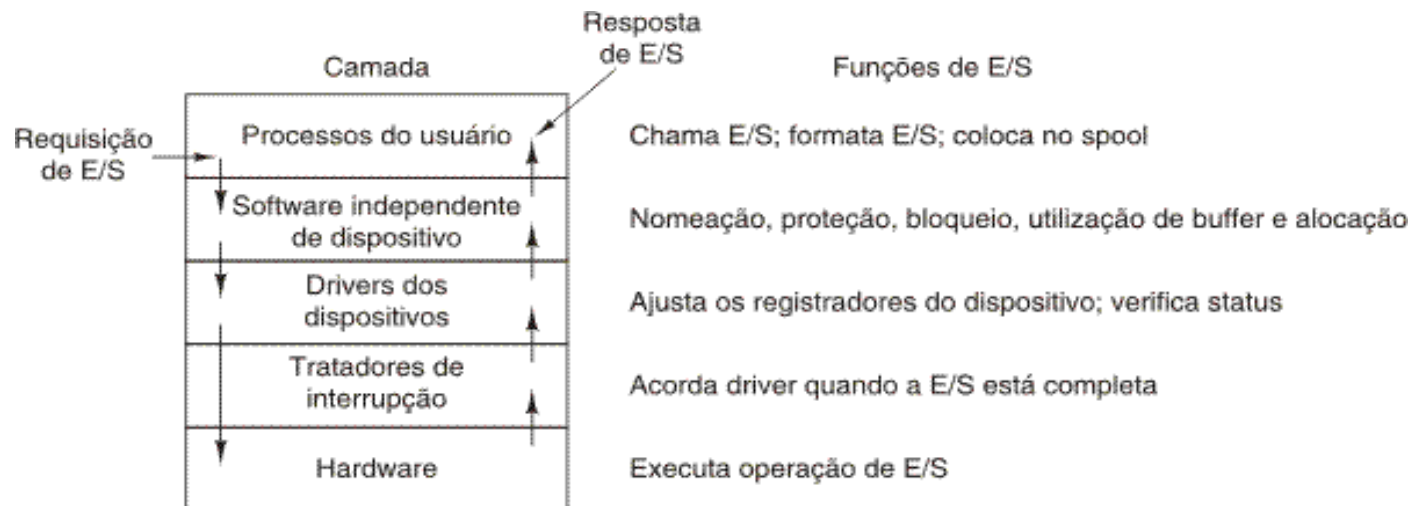
Gerenciamento de Entrada e Saída

- Camadas do SW de E/S:
 - Software de E/S no nível do usuário



Gerenciamento de Entrada e Saída

- Princípios do SW de E/S:
 - Camadas do SW de E/S:
 - SW de E/S do Espaço do Usuário:
 - Procedimentos de biblioteca
 - e.g. `printf("hello world\n");`
 - Sistema de spooling.
 - e.g. Envio de arquivos para impressora (proc. daemon é o responsável)



Dispositivos de E/S típicos

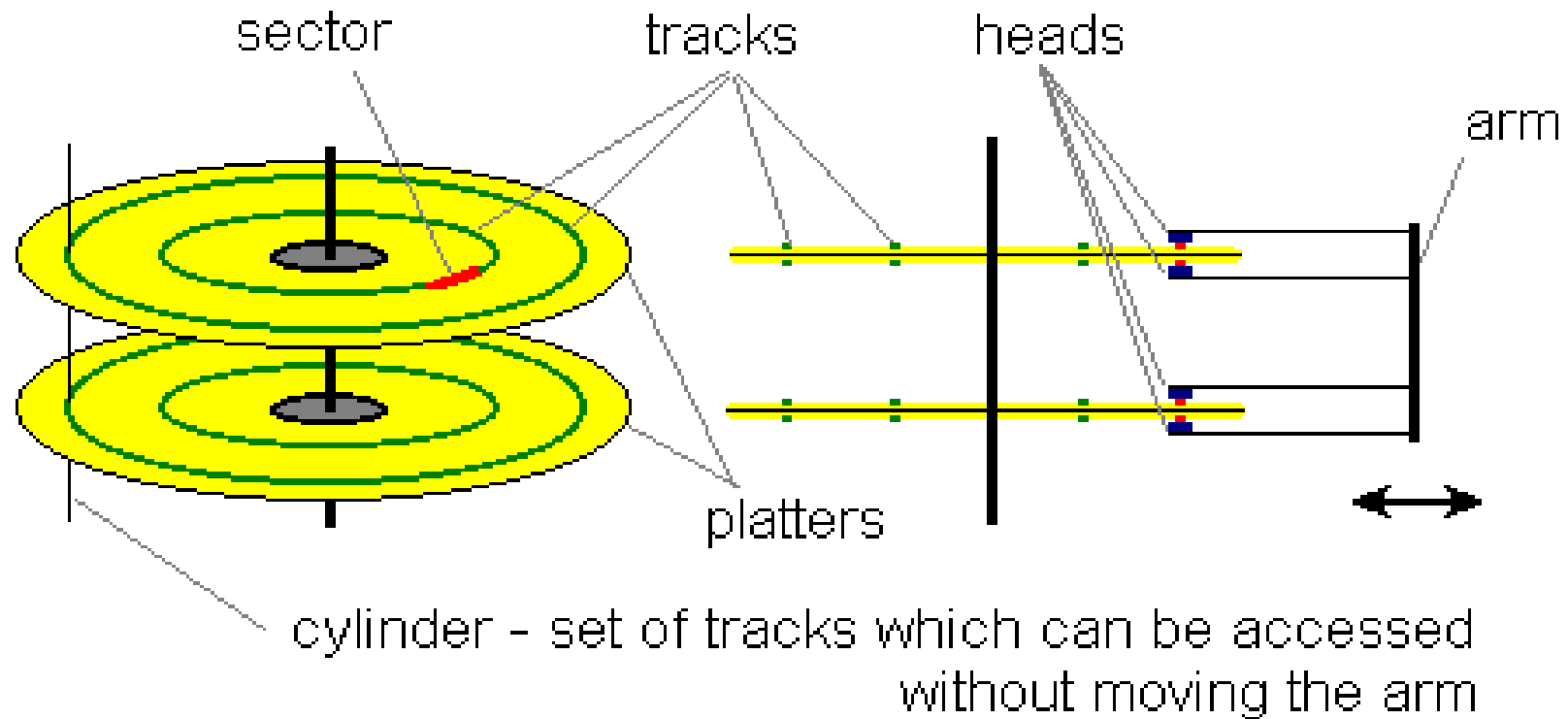
- Dispositivo de armazenamento
 - Disco rígido
- Dispositivo de saída
 - Vídeo
- Dispositivo de entrada
 - Teclado
- Dispositivo de temporização
 - Relógio

Dispositivo de armazenamento

- Disco rígido
 - Conjunto de **discos metálicos** empilhados, cada um com dois **cabeçotes** (duas faces do disco)
 - Disco dividido em **trilhas** (circunferências),
 - Trilhas divididas em **setores** (radial) com tamanho entre 512 e 4096 bytes
 - Tamanho definido pela unidade mínima de leitura e gravação de um disco
 - **Cilindro**: conjunto de trilhas de todas as superfícies do disco que ficam exatamente à mesma distância do eixo central

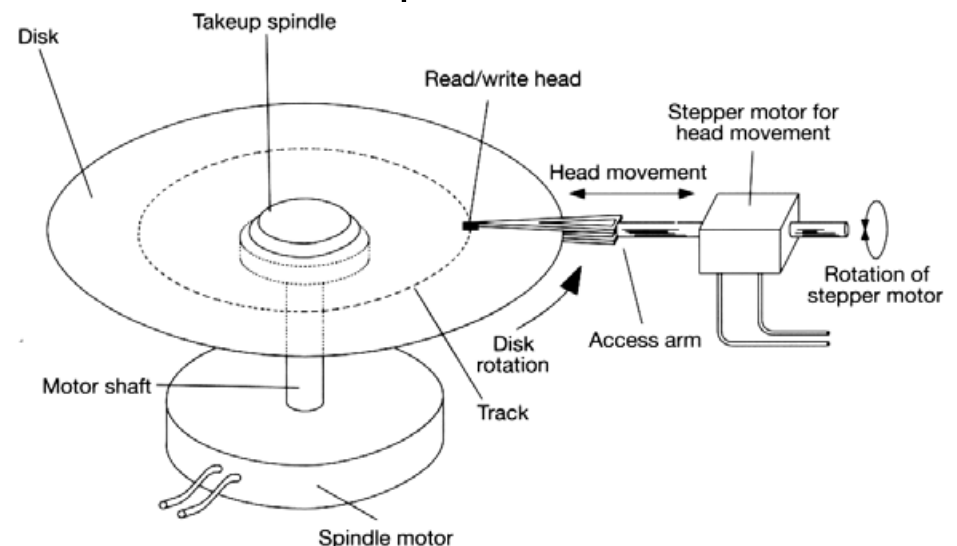
Dispositivo de armazenemento

- Disco rígido



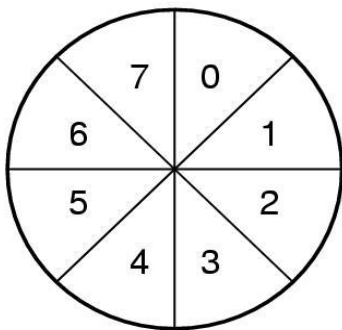
Dispositivo de armazenemento

- Disco rígido
 - Acesso aos dados do disco: cilindro, superfície, setor
- Tempo de acesso = seek time + latency time + transfer time
 - **Seek time**: deslocamento do cabeçote/braço até o cilindro/trilha
 - **Latency time**: deslocamento do cabeçote até o setor / por rotação do disco
 - **Transfer time**: tempo para leitura/escrita do dado para a/da memória da controladora

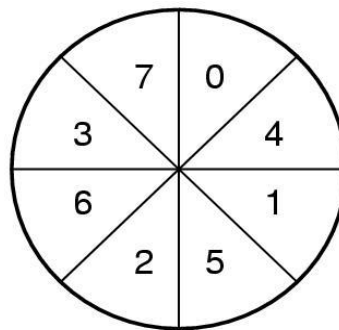


Dispositivo de armazenamento

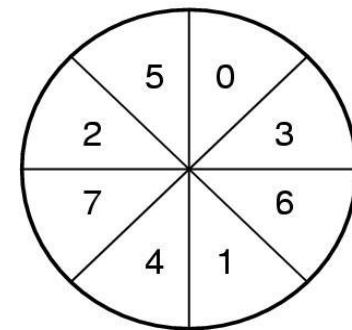
- Disco rígido
 - Entrelaçamento:
 - Técnica empregada para otimizar o tempo de acesso a setores contíguos
 - Reduz latency time, reduz volta completa
 - Após setor ser lido, precisa ser bufferizado, mas rotação continua
 - Para ler próximo setor é necessário uma volta completa
 - Enumera os setores com espaços entre eles
 - Espaçamento entre setores contínuos depende do hardware
 - Velocidade do processador, barramento, controladora e rotação do disco



(a)



(b)



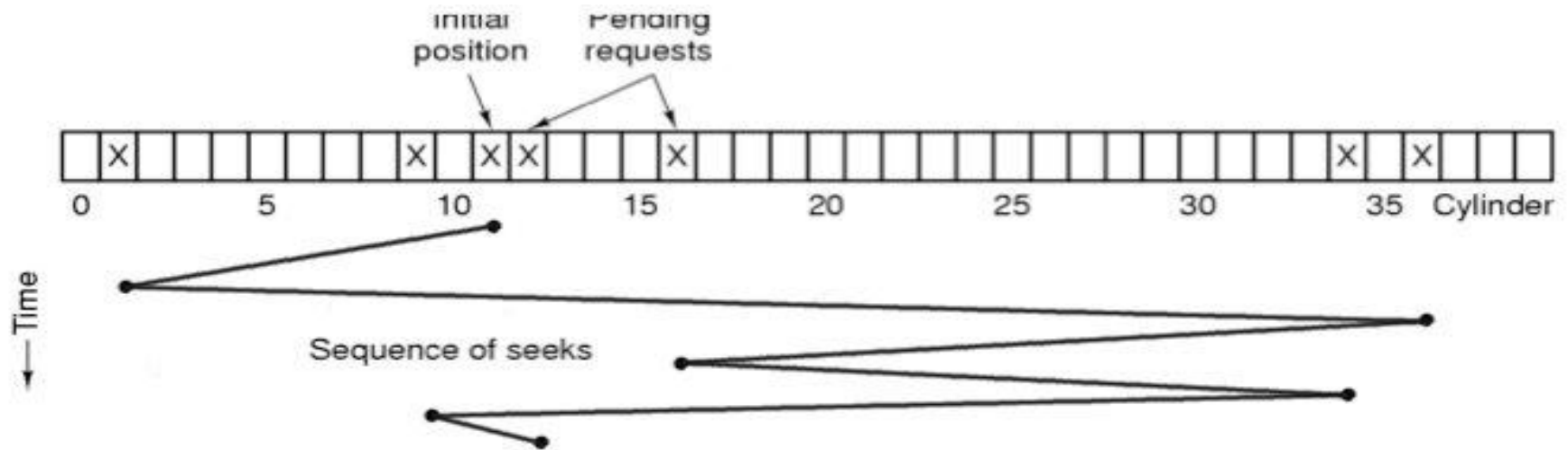
(c)

Dispositivo de armazenamento

- Escalonamento do disco rígido
 - Objetivo
 - Maximizar o atendimento as operações de leitura e escrita
 - Minimizar o tempo em que processos ficam bloqueados
 - Diminuir seek time
 - Variável crítica no tempo de acesso ao disco
 - Ordenar as requisições de operação

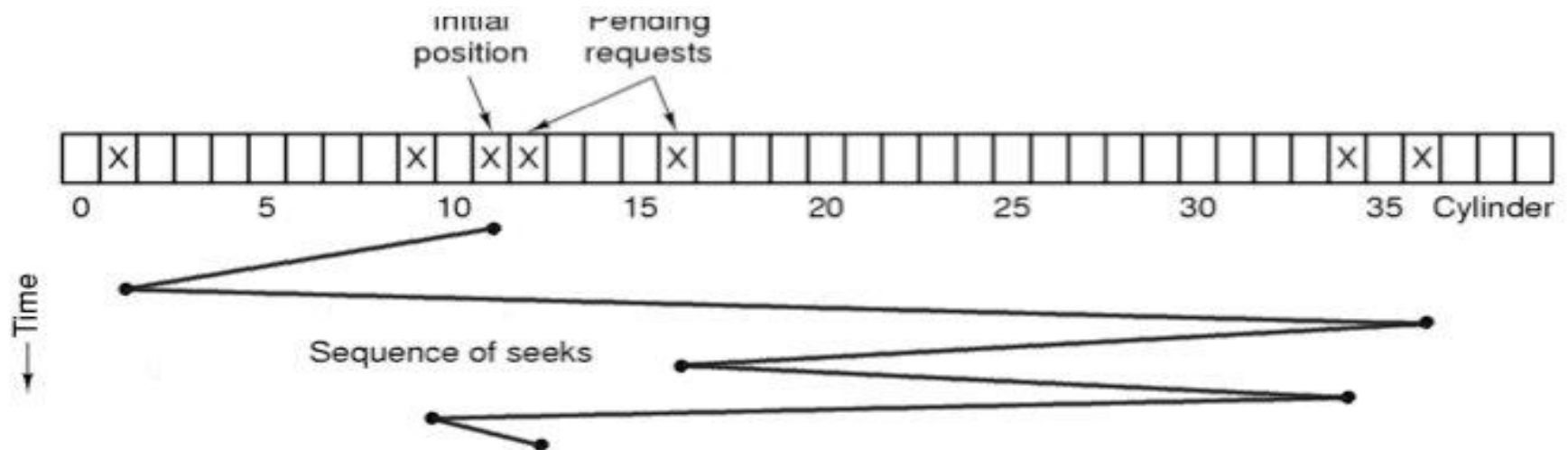
Dispositivo de armazenemento

- Escalonamento do disco rígido
 - FCFS (First Come First Served)
 - Requisições são atendidas na ordem em que chegada



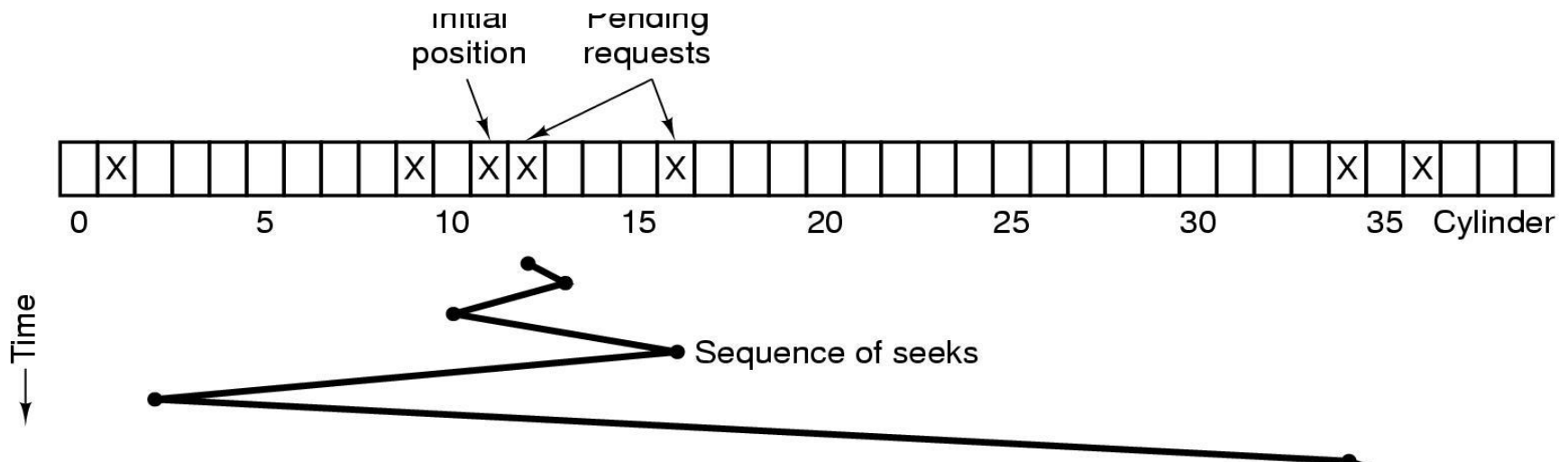
Dispositivo de armazenemento

- Escalonamento do disco rígido
 - FCFS (First Come First Served)
 - Solução trivial
 - Requisições aos cilindros (posição inicial: 11)
 - Ordem das requisições: 1, 36, 16, 34, 9 e 12
 - Movimentação total do braço: $10 + 35 + 20 + 18 + 25 + 3 = 111$ cilindros



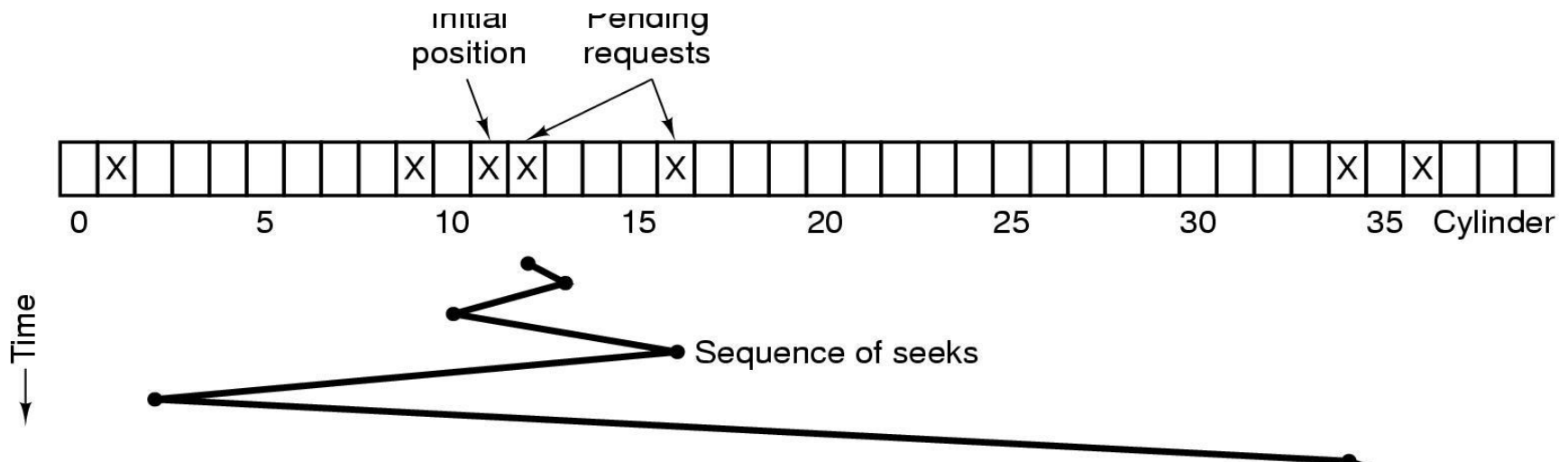
Dispositivo de armazenemento

- Escalonamento de disco rígido
 - SSTF (Shortest Seek Time First):
 - Foco total no problema
 - Ordena a fila para minimizar o movimento do cabeçote entre cilindros
 - Problema:
 - Possível postergação indefinida ou demora de atendimento



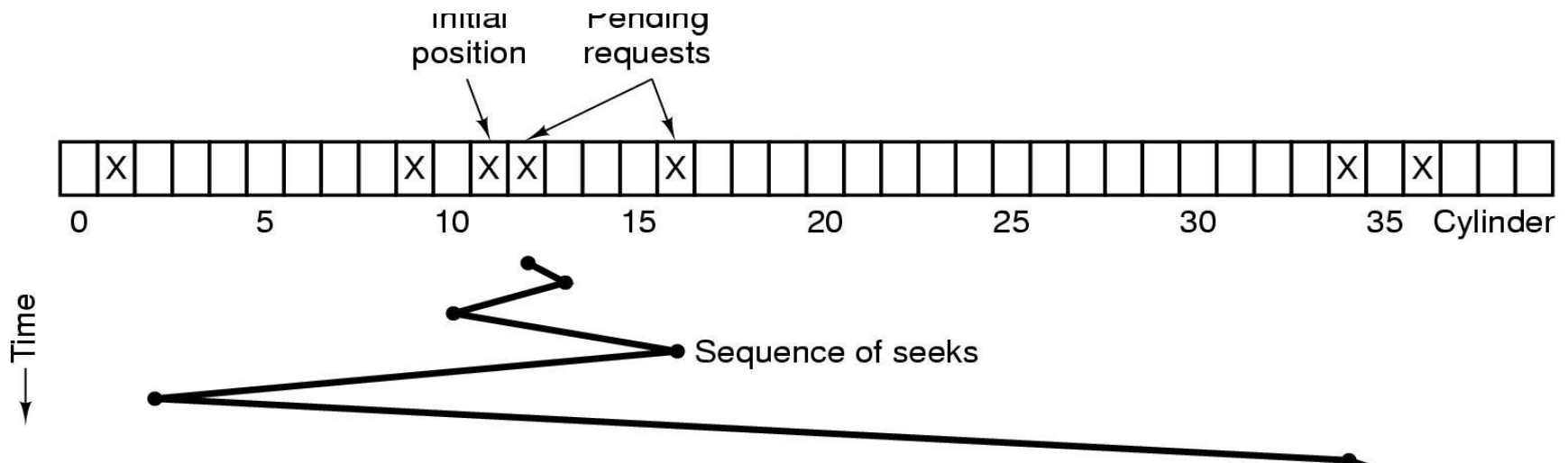
Dispositivo de armazenemento

- Escalonamento de disco rígido
 - SSTF (Shortest Seek Time First):
 - Requisições aos cilindros (posição inicial: 11)
 - Ordem das requisições: 1, 36, 16, 34, 9 e 12
 - Reordenando: 12, 9, 16, 1, 34 e 36
 - Movimentação total do braço: $1+3+5+7+15+33+2 = 61$ cilindros



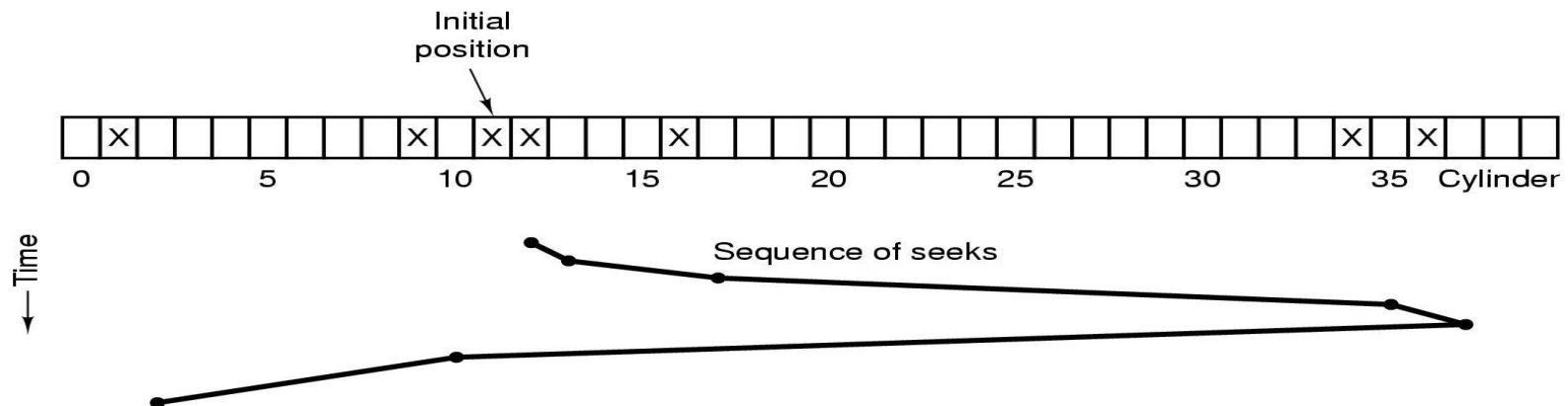
Dispositivo de armazenemento

- Escalonamento de disco rígido
 - SSTF (Shortest Seek Time First):
 - Problema:
 - Possível postergação indefinida ou demora de atendimento



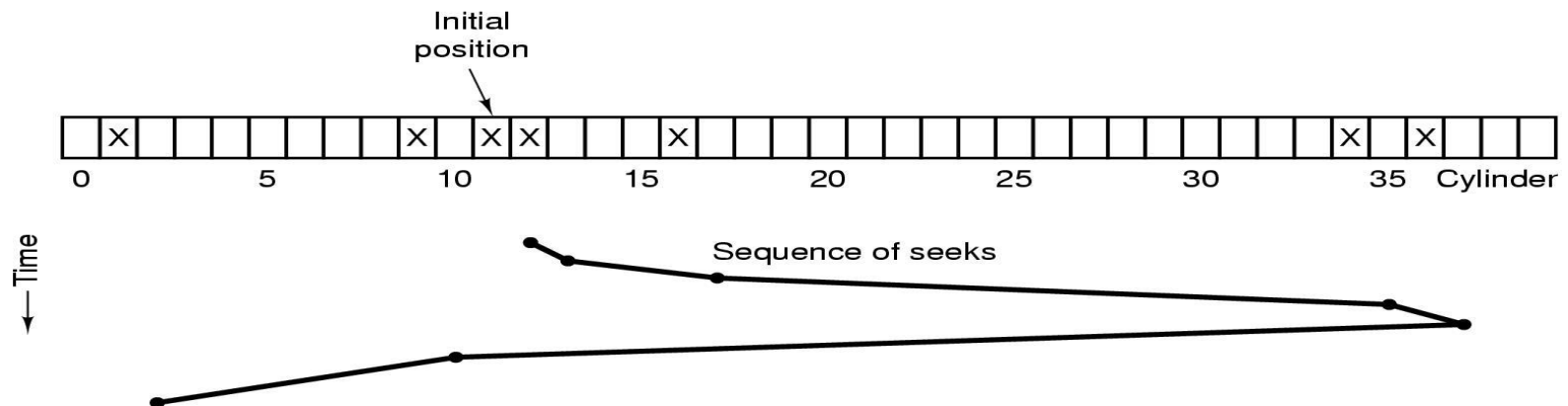
Dispositivo de armazenemento

- Escalonamento de disco rígido
 - SCAN:
 - Variação do SSTF
 - Minimiza movimento do cabeçote, mas somente em uma direção,
 - Quando chega no limite muda direção (algoritmo do elevador)
 - Problema:
 - Privilegia cilindros das extremidades



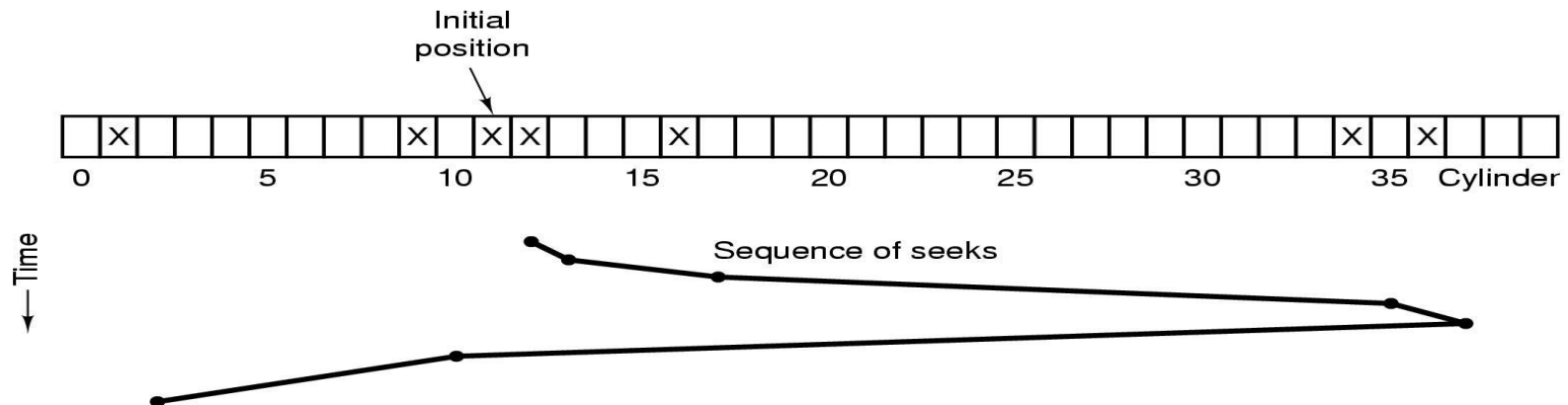
Dispositivo de armazenemento

- Escalonamento de disco rígido
 - SCAN:
 - Requisições aos cilindros (posição inicial: 11)
 - Ordem das requisições: 1, 36, 16, 34, 9 e 12
 - Reordenando: 12, 16, 34, 36, 9, 1
 - Movimentação total do braço: $1+4+18+2+27+8 = 60$ cilindros



Dispositivo de armazenamento

- Escalonamento de disco rígido
 - SCAN:
 - Problema:
 - Requisições das extremidades são privilegiadas
 - Ao alcançar uma extremidade
 - o As requisições próximas já foram atendidas
 - o As requisições da outra extremidade tem tempo de atendimento não preciso



Dispositivos de armazenamento

- Escalonamento de disco rígido
 - C-SCAN (Circular SCAN)
 - Assume somente uma direção para operação
 - Quando chega no limite retorna para outro extremo
 - LOOK
 - Similar ao SCAN, porém
 - Para movimento do cabeçote quando não existem mais requisições na direção atual
 - C-LOOK (Circular LOOK)
 - Combinação do C-SCAN com LOOK
 - Serve até a última requisição em uma única direção, depois retorna para o outro extremo

Dispositivos de armazenamento

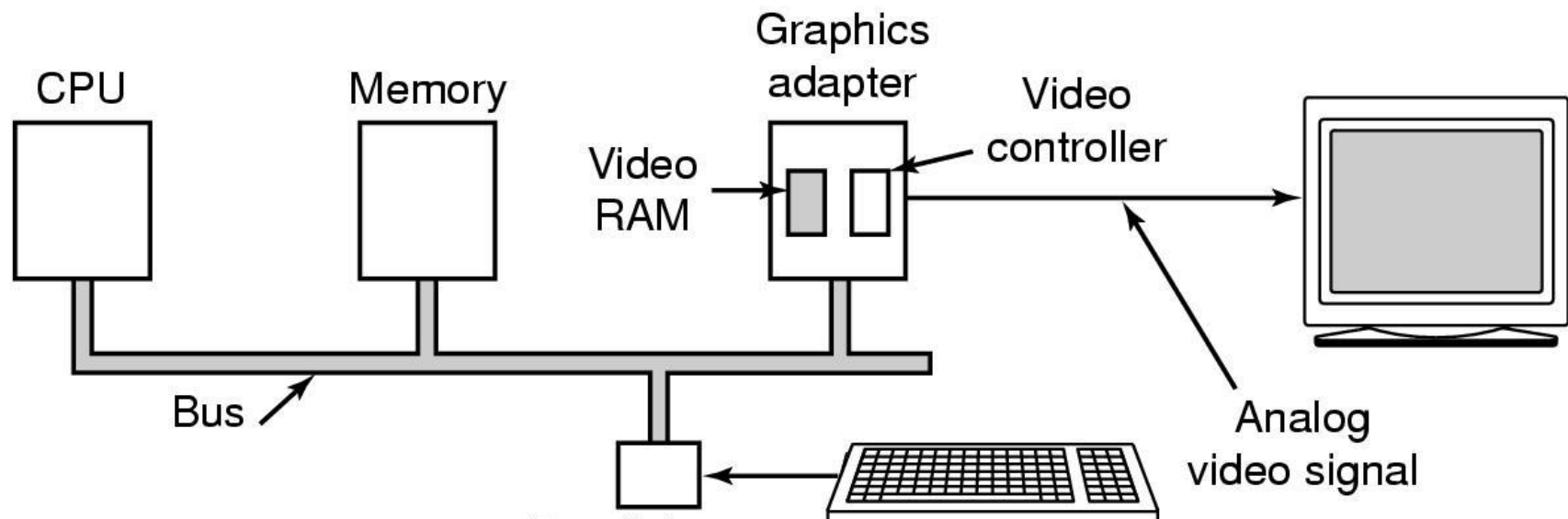
- Escalonamento de disco rígido
 - Qual o melhor ?
 - Custo de ordenação vs desempenho!
- Tecnologias de disco simples – floppy
 - Utilizam FCFS!

Dispositivo de saída

- Vídeo
 - Definido a partir de uma matriz de linhas e colunas
- Modos:
 - Texto (matriz de caracteres)
 - Gráfico (matriz de pixels)
- Memória de vídeo determina
 - Resolução (linhas x colunas)
 - Número de cores
 - Determina a controladora (EGA, VGA, SVGA, etc)
- *Driver* de vídeo
 - Determina memória de vídeo disponível
 - Modo de operação (texto ou gráfico)
 - Frequências para varreduras horizontal e vertical
 - Geração de sinais RGB

Dispositivo de saída

- Vídeo
 - Mapeado em memória
 - *Driver* escreve diretamente na memória RAM de vídeo

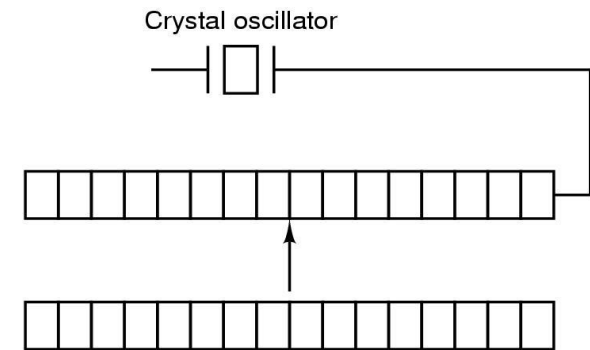


Dispositivo de Entrada

- Teclado
 - Tecla acionada gera scancode (código de varredura)
 - Dois códigos – pressionamento e liberação
 - Ao pressionar uma tecla é gerada uma interrupção
 - Mapas de códigos
 - *Driver* responsável pela conversão do *scancode* em caractere (e.g. ASCII)
 - Permite converter caracteres do buffer do teclado de forma apropriada
 - Exemplo: `+a=à

Dispositivo de temporização

- Relógio (Solução em *hardware*)
 - Oscilador de cristal
 - Tensão é aplicada sobre cristal
 - Contador
 - Cada oscilação do cristal decrementa contador
 - Quando contador chega em zero
 - Gera interrupção (clock tick)
 - Registrador de apoio
 - Contém valor base a ser carregado no contador quando de um clock tique ocorre ()

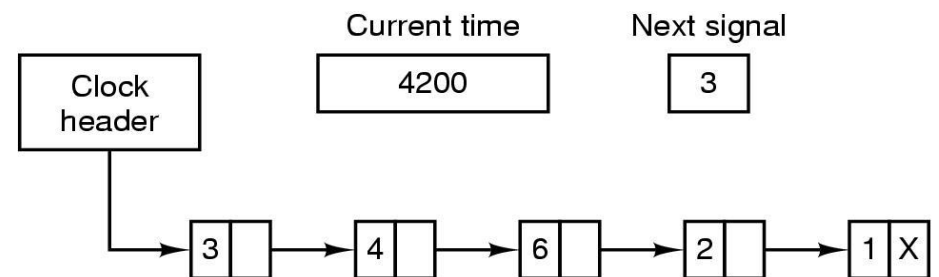


Dispositivo de temporização

- Relógio
 - Funções do *driver*
 - Manter a hora do dia
 - Incrementar registradores de data/hora
 - Impedir que processos executem infinitamente
 - Garantir preempção da CPU
 - o Aplicando escalonamento Round Robin
 - Contabilizar o uso de CPU
 - Referente aos processos em execução
 - Tratar timers disparados pelo usuário (ALARM)
 - Traçar perfis
 - Profiling de processos
 - Análise do uso de CPU pelas rotinas de um processo

Dispositivo de temporização

- Relógio
 - Alarmes em nível de usuário
 - Implementação de diversos timers usando somente um relógio
 - A cada clock tick, timer apontado é decrementado
 - o Quando chega à 0, dispara o sinal desejado e remove da lista
 - Exemplo: *timers* – 4203, 4207, 4213, 4215, 4216
 - Aplicação: timeout de aplicações (envio de pacote / resposta de usuário)
 - Alarmes em nível de sistema
 - Watch dog
 - Define intervalos aceitáveis de tratamento de interrupção
 - Permite estabelecer intervalos para acionamento de um dispositivo



Bibliografia

- Tanenbaum, A. “*Sistemas Operacionais: projeto e implementação*”. Capítulo 3
- Silberschatz, G. “*Operating System Concepts*”. Capítulos 12 e 13