

Capítulo 5

Entrada/Saída

- 5.1 Princípios do hardware de E/S
- 5.2 Princípios do software de E/S
- 5.3 Camadas do software de E/S
- 5.4 Discos
- 5.5 Relógios
- 5.6 Terminais com base em caracteres
- 5.7 Interfaces gráficas do usuário
- 5.8 Terminais de rede
- 5.9 Gerenciamento de energia

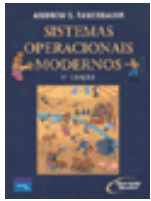
Princípios do Hardware de E/S



Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Canal telefônico	8 KB/s
Linhas ISDN dual	16 KB/s
Impressora a laser	100 KB/s
Scanner	400 KB/s
Ethernet clássica	1,25 MB/s
USB (<i>universal serial bus</i> — barramento serial universal)	1,5 MB/s
Câmara de vídeo digital	4 MB/s
Disco IDE	5 MB/s
CD-ROM 40x	6 MB/s
Ethernet rápida	12,5 MB/s
Barramento ISA	16,7 MB/s
Disco EIDE (ATA-2)	16,7 MB/s
FireWire (IEEE 1394)	50 MB/s
Monitor XGA	60 MB/s
Rede SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Ethernet Gigabit	125 MB/s
Dispositivo de Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s
Barramento da Sun Gigaplane XB	20 GB/s

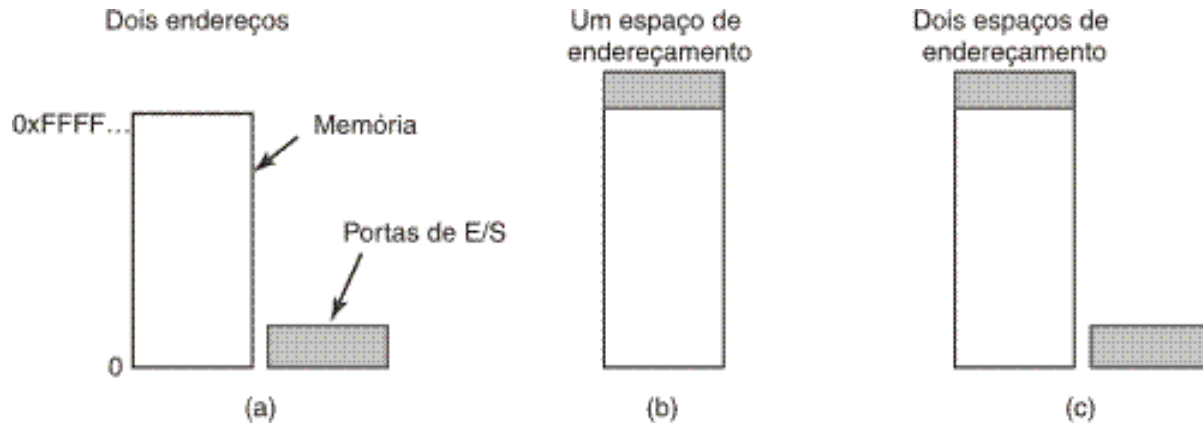
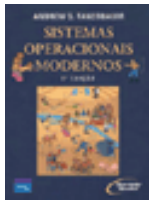
Taxas de dados típicas de dispositivos, redes e barramentos

Controladores de Dispositivos



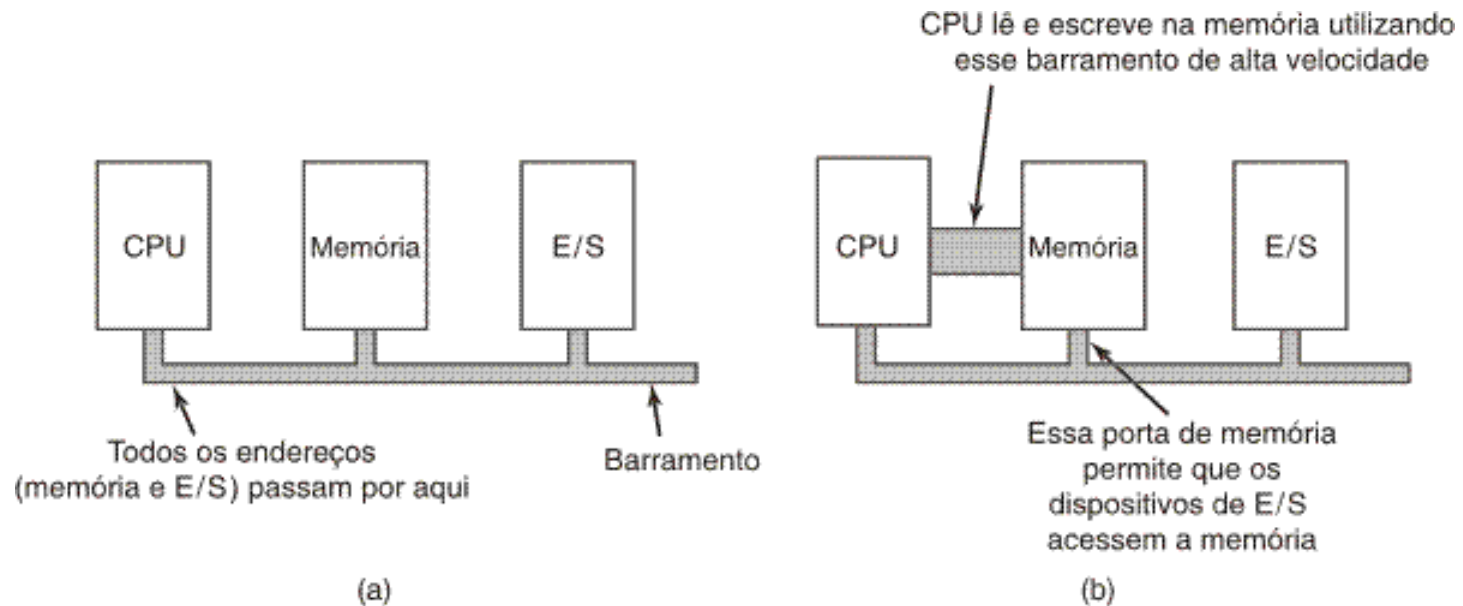
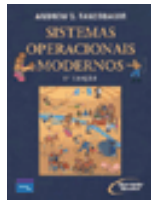
- Componentes de dispositivos de E/S
 - mecânico
 - eletrônico
- O componente eletrônico é o controlador do dispositivo
 - pode ser capaz de tratar múltiplos dispositivos
- Tarefas do controlador
 - converter fluxo serial de bits em bloco de bytes
 - executar toda correção de erro necessária
 - tornar o bloco disponível para ser copiado para a memória principal

E/S mapeada na memória (1)



- a) Espaços de memória e E/S separados
- b) E/S mapeada na memória
- c) Híbrido

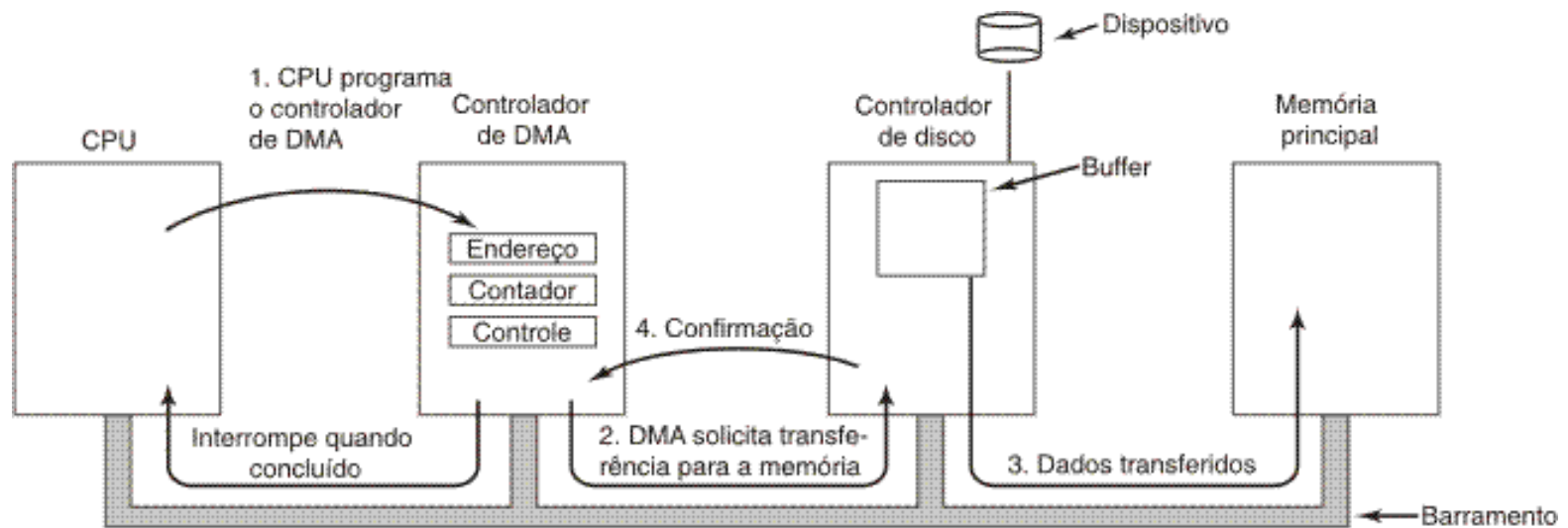
E/S mapeada na memória (2)



(a) Arquitetura com barramento único

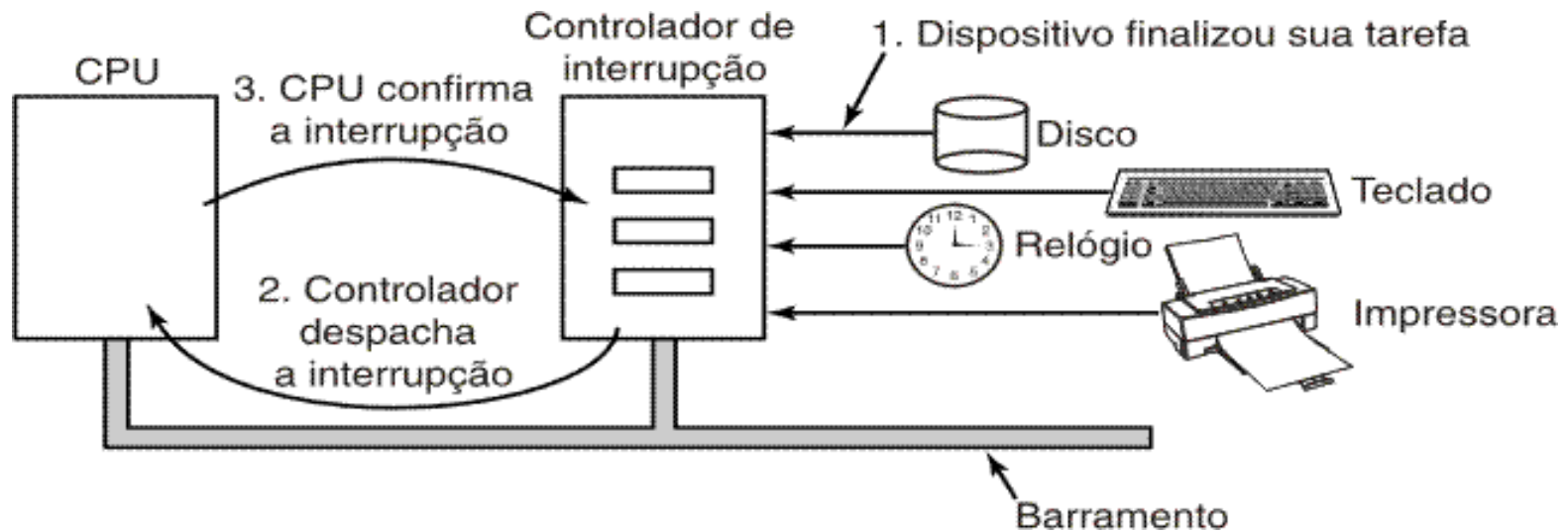
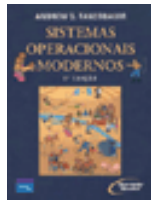
(b) Arquitetura com barramento dual

Acesso Direto à Memória (DMA)



Operação de uma transferência com DMA

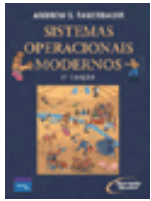
Interrupções Revisitadas



Como ocorre uma interrupção. Conexões entre dispositivos e controlador de interrupção usam linhas de interrupção no barramento em vez de fios dedicados

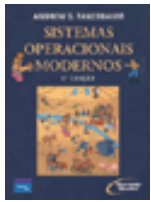
Princípios do Software de E/S

Objetivos do Software de E/S (1)



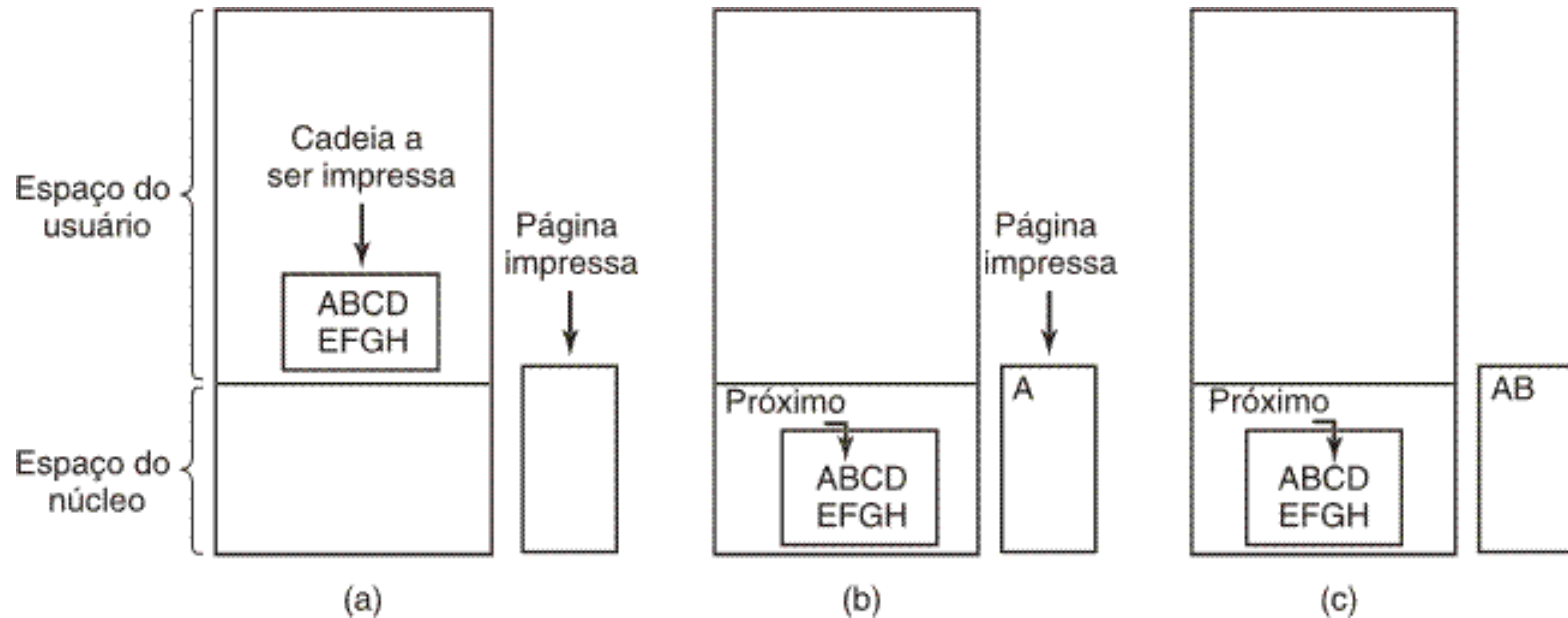
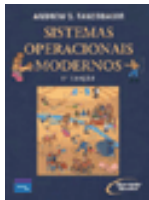
- Independência de dispositivo
 - Programas podem acessar qualquer dispositivo de E/S sem especificar previamente qual (disquete, disco rígido ou CD-ROM)
- Nomeação uniforme
 - Nome de um arquivo ou dispositivo pode ser uma cadeia de caracteres ou um número inteiro que é independente do dispositivo
- Tratamento de erro
 - Trata o mais próximo possível do hardware

Objetivos do Software de E/S (2)



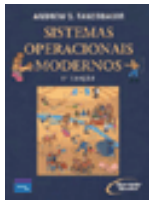
- **Transferências Síncronas vs. Assíncronas**
 - transferências bloqueantes vs. orientadas a interrupção
 - utilização de buffer para armazenamento temporário
 - dados provenientes de um dispositivo muitas vezes não podem ser armazenados diretamente em seu destino final
- **Dispositivos Compartilháveis vs. Dedicados**
 - discos são compartilháveis
 - unidades de fita não são

E/S Programada (1)



Passos da impressão de uma cadeia de caracteres

E/S Programada (2)



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

```
/* p é o buffer do núcleo */
/* executa o laço para cada caractere */
/* executa o laço até PRONTO */
/* envia um caractere para a saída */
```

Escrita de uma cadeia de caracteres para a impressora usando E/S programada

E/S Orientada à Interrupção



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

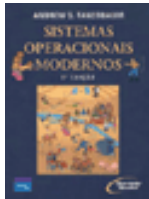
(a)

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

(b)

- Escrita de uma cadeia de caracteres para a impressora usando E/S orientada à interrupção
 - a) Código executado quando quando é feita a chamada ao sistema para impressão
 - b) Rotina de tratamento de interrupção

E/S Usando DMA



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

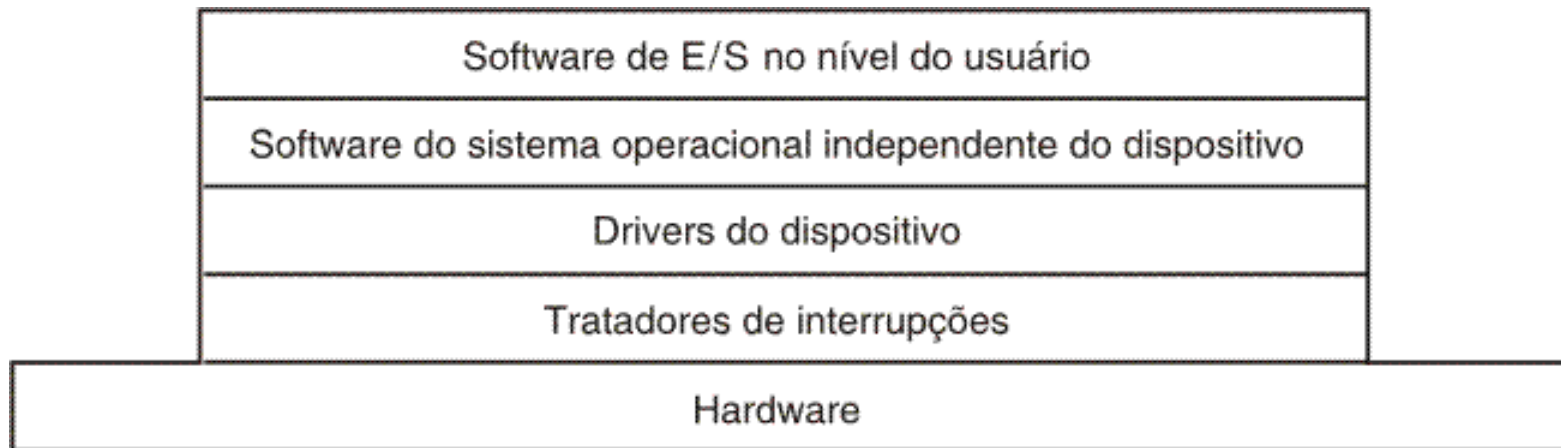
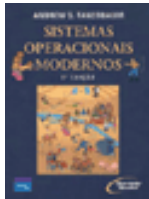
(a)

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

(b)

- Impressão de uma cadeia de caracteres usando DMA
 - a) Código executado quando quando é feita a chamada ao sistema para impressão
 - b) Rotina de tratamento de interrupção

Camadas do Software de E/S



Camadas do sistema de software de E/S

Tratadores de Interrupção (1)



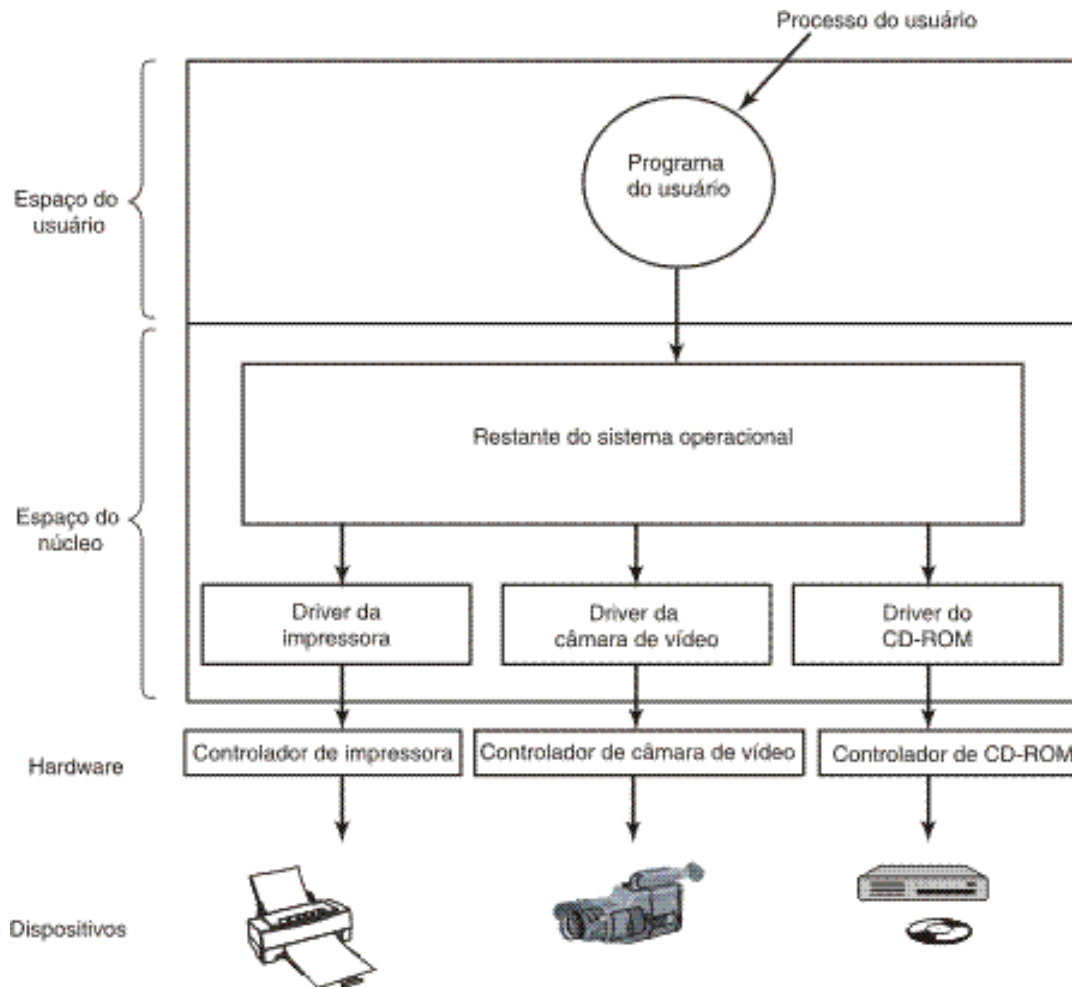
- As interrupções devem ser escondidas o máximo possível
 - uma forma de fazer isso é bloqueando o driver que iniciou uma operação de E/S até que uma interrupção notifique que a E/S foi completada
- Rotina de tratamento de interrupção cumpre sua tarefa
 - e então desbloqueia o driver que a chamou

Tratadores de Interrupção (2)



- **Passos que devem ser executados em software depois da interrupção ter sido concluída**
- 2. salva registradores que ainda não foram salvos pelo hardware de interrupção
- 3. estabelece contexto para rotina de tratamento de interrupção
- 4. estabelece uma pilha para a rotina de tratamento de interrupção
- 5. sinaliza o controlador de interrupção, reabilita as interrupções
- 6. copia os registradores de onde eles foram salvos
- 7. executa rotina de tratamento de interrupção
- 8. escolhe o próximo processo a executar
- 9. estabelece o contexto da MMU para o próximo processo a executar
- 10. carrega os registradores do novo processo
- 11. começa a executar o novo processo

Drivers dos Dispositivos



- Posição lógica dos drivers dos dispositivos
- A comunicação entre os drivers e os controladores de dispositivos é feita por meio do barramento

Software de E/S Independente de Dispositivo (1)



Interface uniforme para os drivers dos dispositivos

Armazenamento em buffer

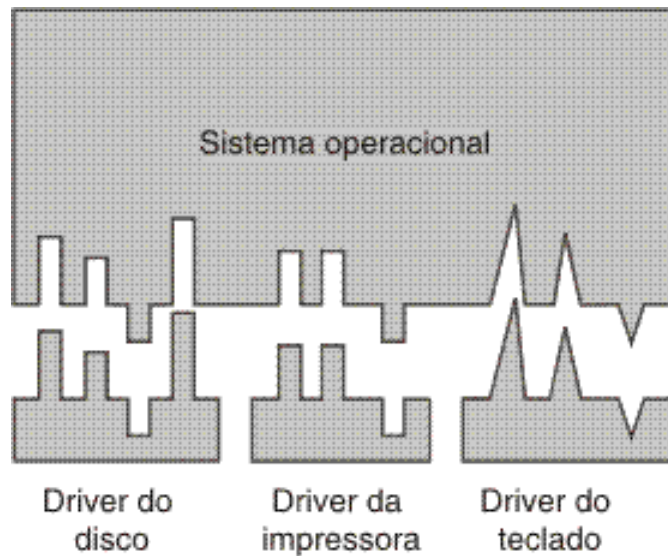
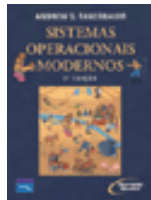
Relatório dos erros

Alocação e liberação de dispositivos dedicados

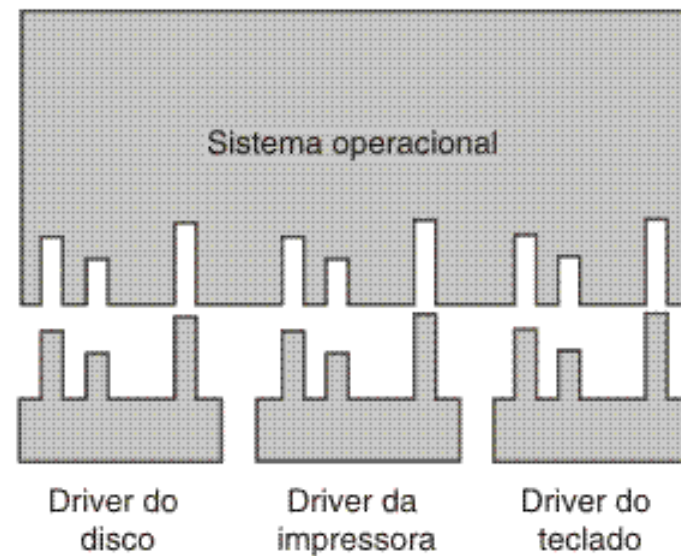
Fornecimento de tamanho de bloco independente
de dispositivo

Funções do software de E/S independente de dispositivo

Software de E/S Independente de Dispositivo (2)



(a)

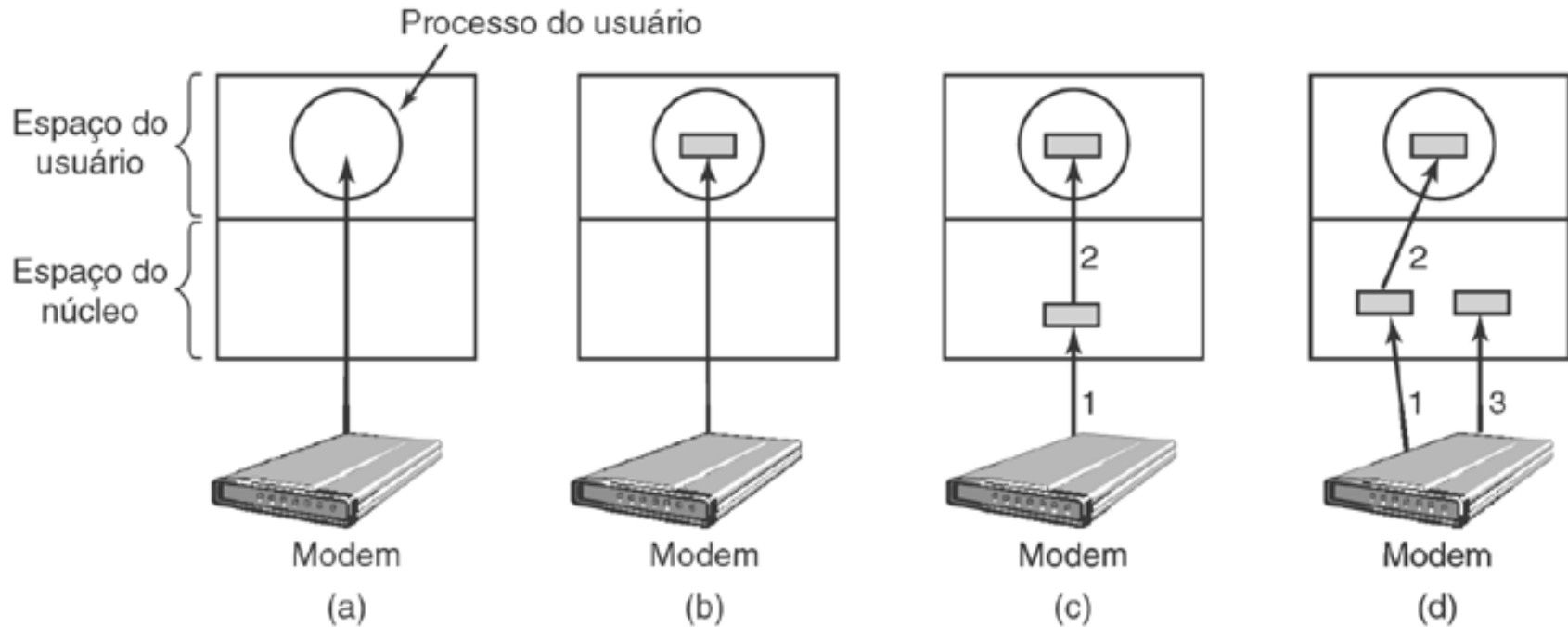
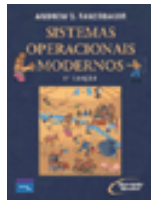


(b)

(a) Sem uma interface-padrão do driver

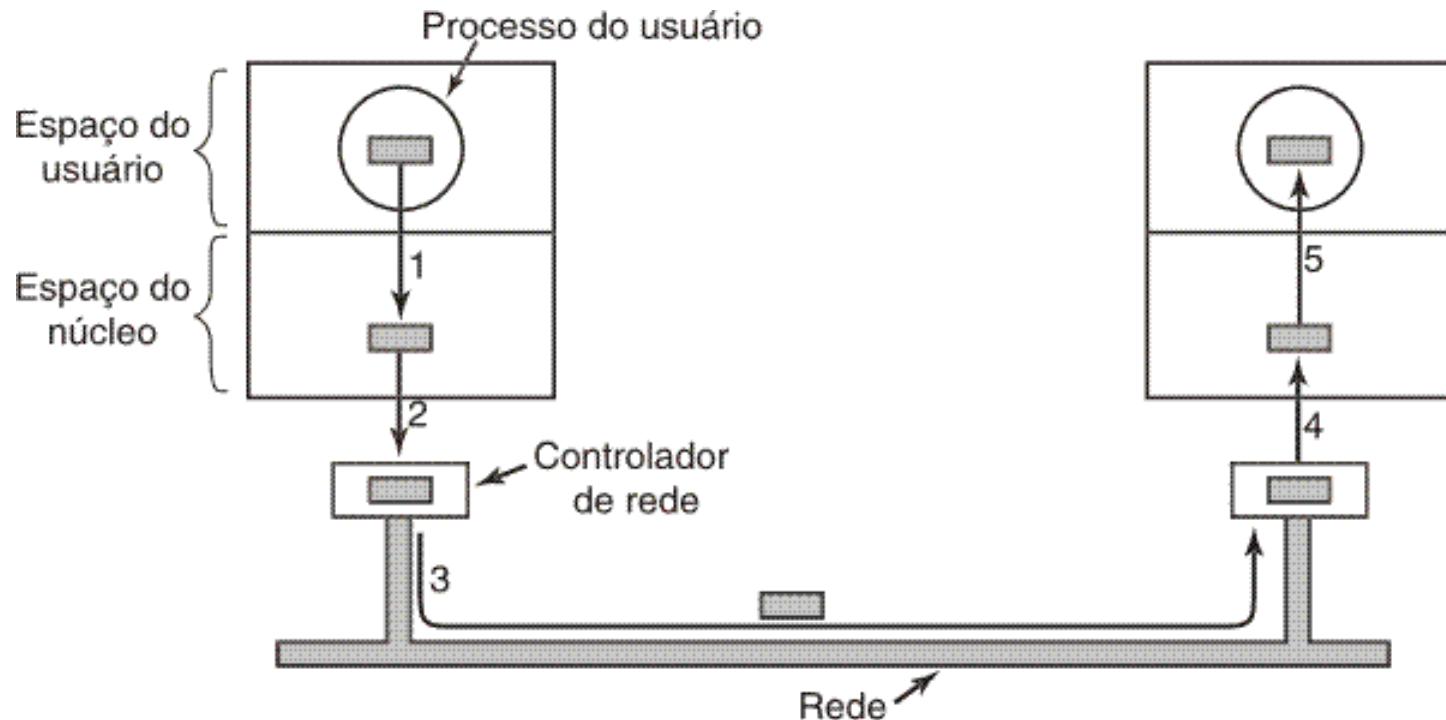
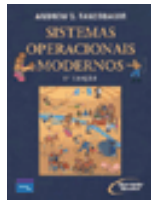
(b) Com uma interface-padrão do driver

Software de E/S Independente de Dispositivo (3)



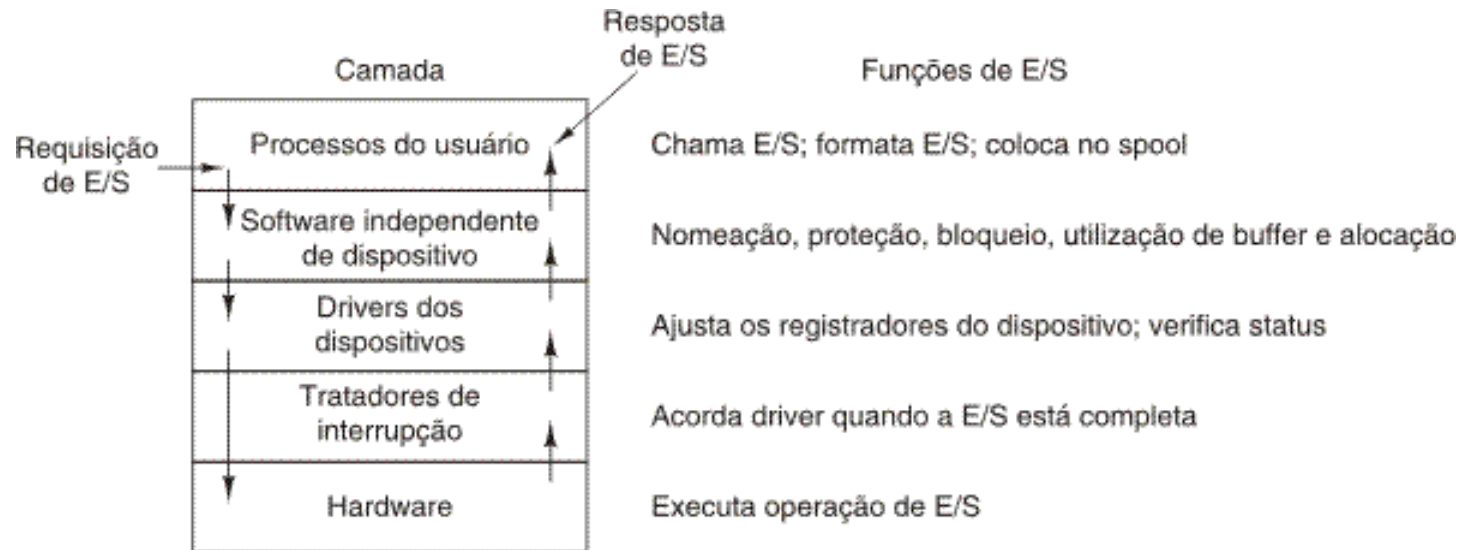
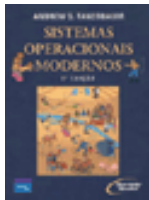
- a) Entrada sem utilização de buffer
- b) Utilização de buffer no espaço do usuário
- c) Utilização de buffer no núcleo seguido de cópia para o espaço do usuário
- d) Utilização de buffer duplo no núcleo

Software de E/S Independente de Dispositivo (4)



A operação em rede pode envolver muitas cópias de um pacote

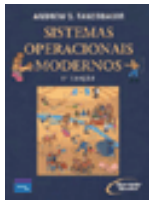
Software de E/S no Espaço do Usuário



Camadas do sistema de E/S e as principais funções de cada camada

Discos

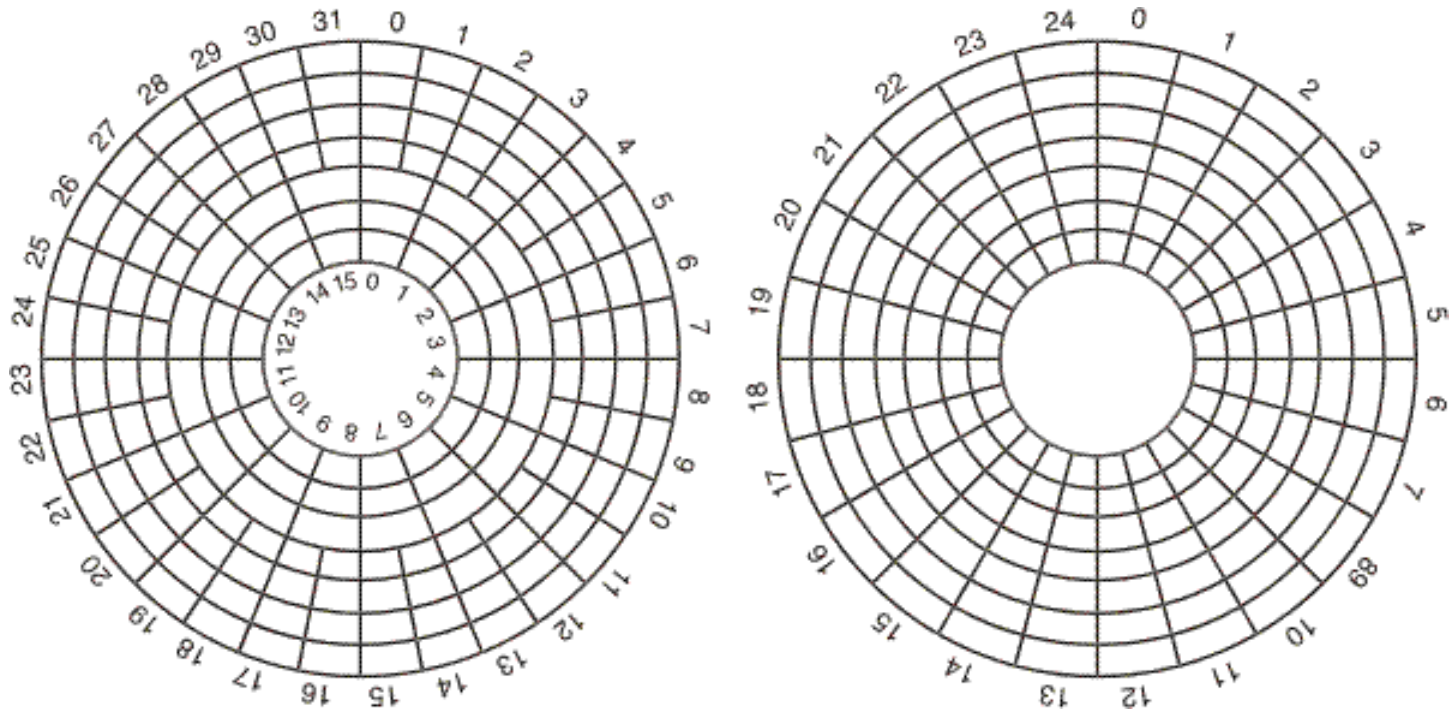
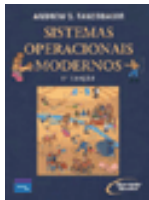
Hardware do Disco (1)



Parâmetro	Disco flexível IBM 360 KB	Disco rígido WD 18300
Número de cilindros	40	10 601
Trilhas por cilindro	2	12
Setores por trilha	9	281 (avg)
Setores por disco	720	35 742 000
Bytes por setor	512	512
Capacidade do disco	360 KB	18,3 GB
Tempo de posicionamento (cilindros adjacentes)	6 ms	0,8 ms
Tempo de posicionamento (caso médio)	77 ms	6,9 ms
Tempo de rotação	200 ms	8,33 ms
Tempo de pára/inicia do motor	250 ms	20 s
Tempo de transferência para um setor	22 ms	17 μ s

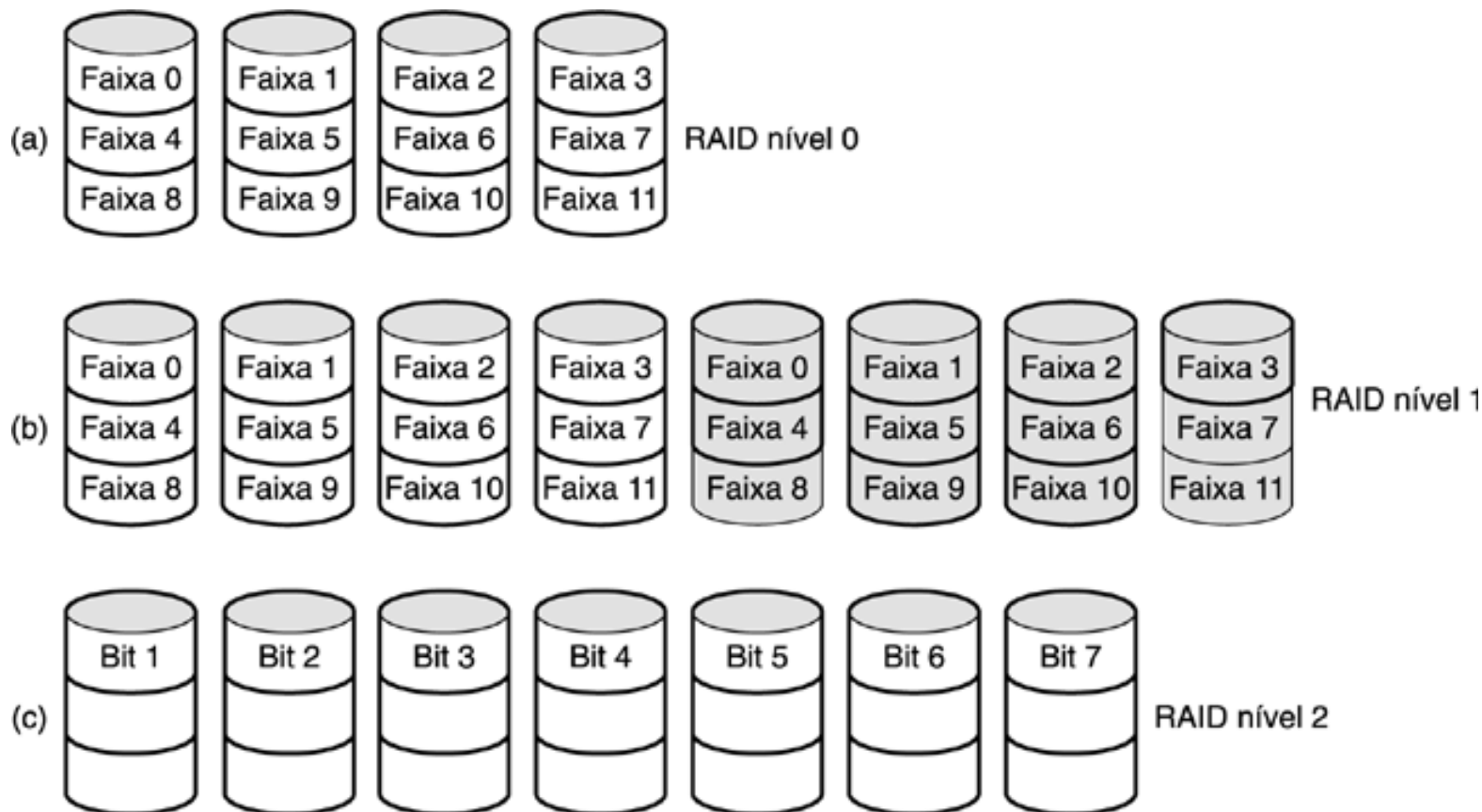
Parâmetros de disco para o disco flexível original do IBM PC e o disco rígido da Western Digital WD 18300

Hardware do Disco (2)



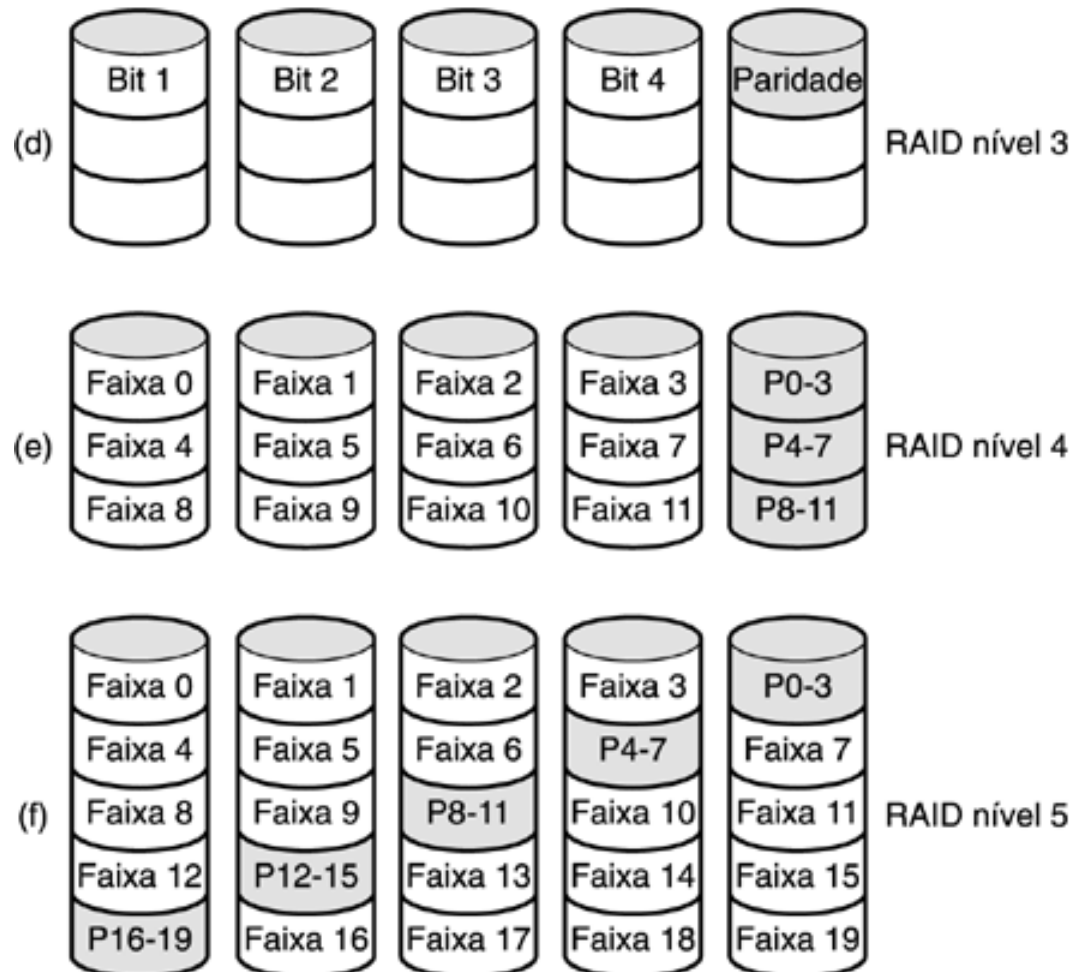
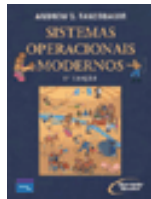
- Geometria física de um disco com duas zonas
- Uma possível geometria virtual para esse disco

Hardware do Disco (3)



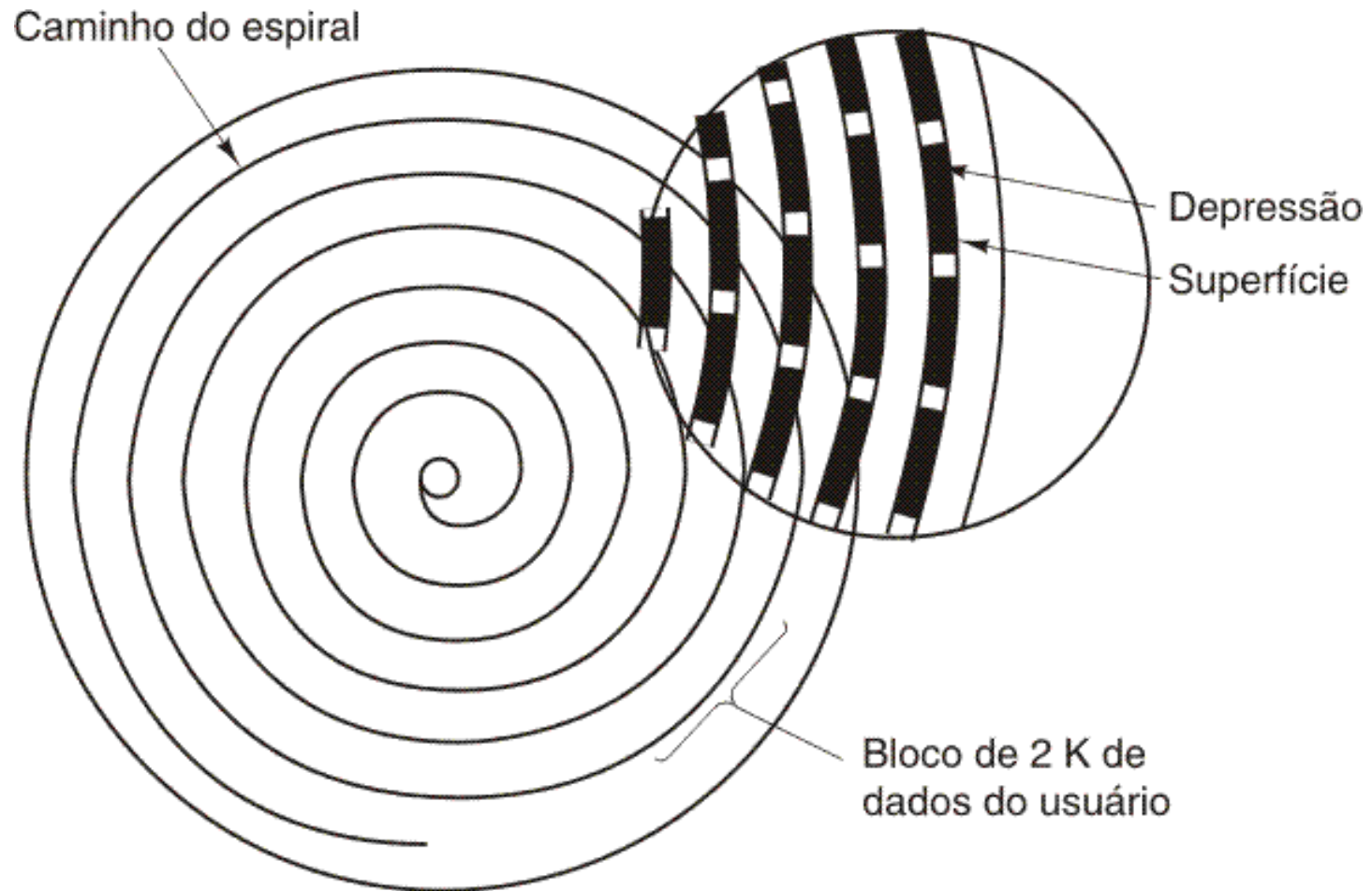
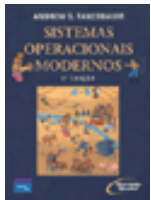
- RAIDs níveis 0 a 2
- Discos de segurança e de paridade são os sombreados

Hardware do Disco (4)



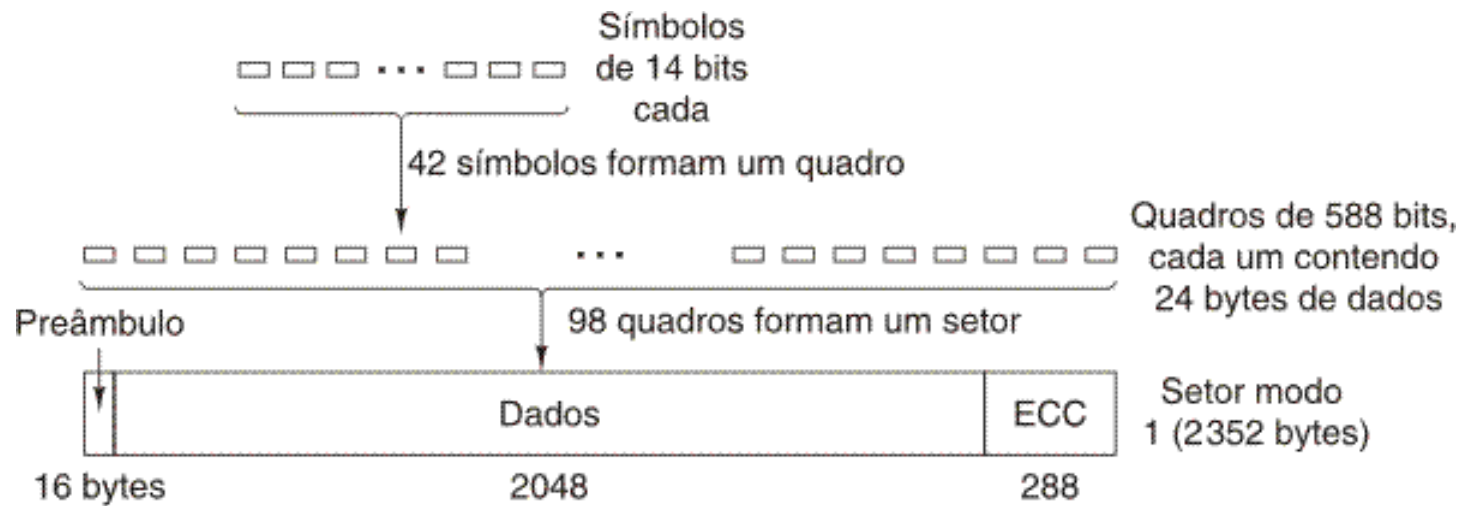
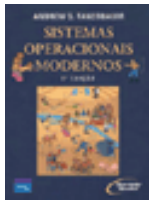
- RAIDs níveis 3 a 5
- Discos de segurança e de paridade são os sombreados

Hardware do Disco (5)



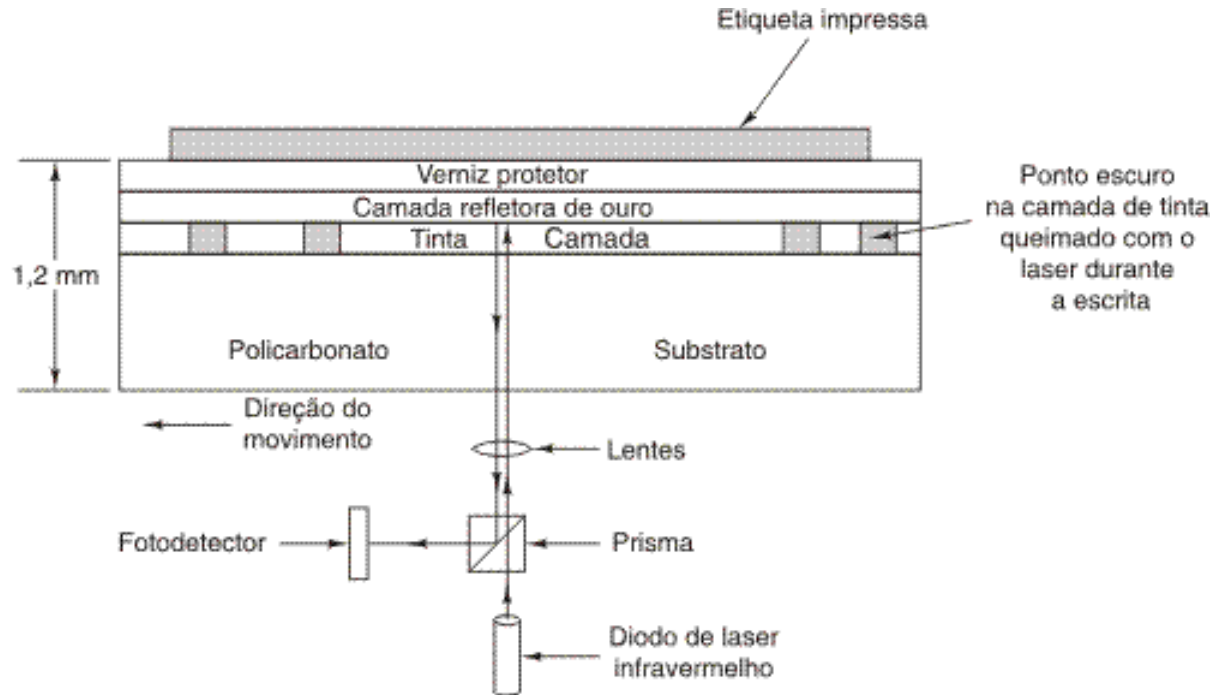
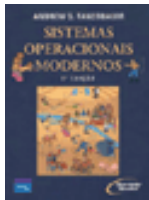
Estrutura de gravação de um CD ou CD-ROM

Hardware do Disco (6)



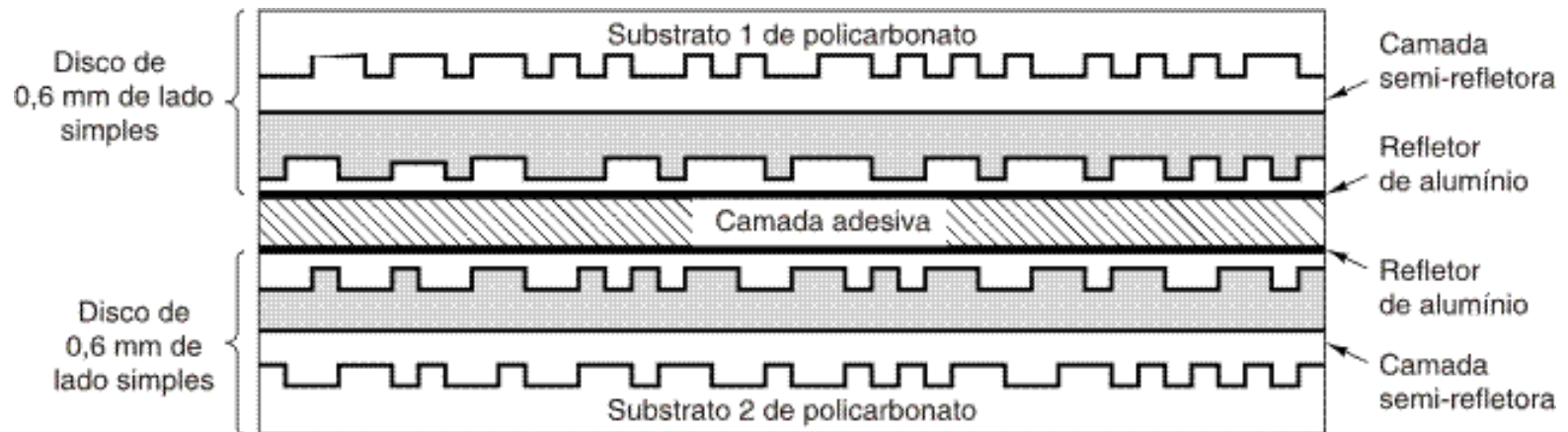
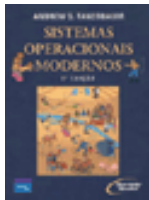
Esquema lógico dos dados em um CD-ROM

Hardware do Disco (7)



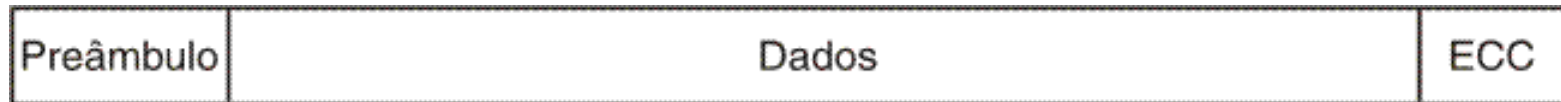
- Secção transversal de um disco CD-R e laser
 - sem escala
- CD-ROM prateado tem estrutura similar
 - sem camada de tinta
 - com camada de alumínio em vez de ouro

Hardware do Disco (8)



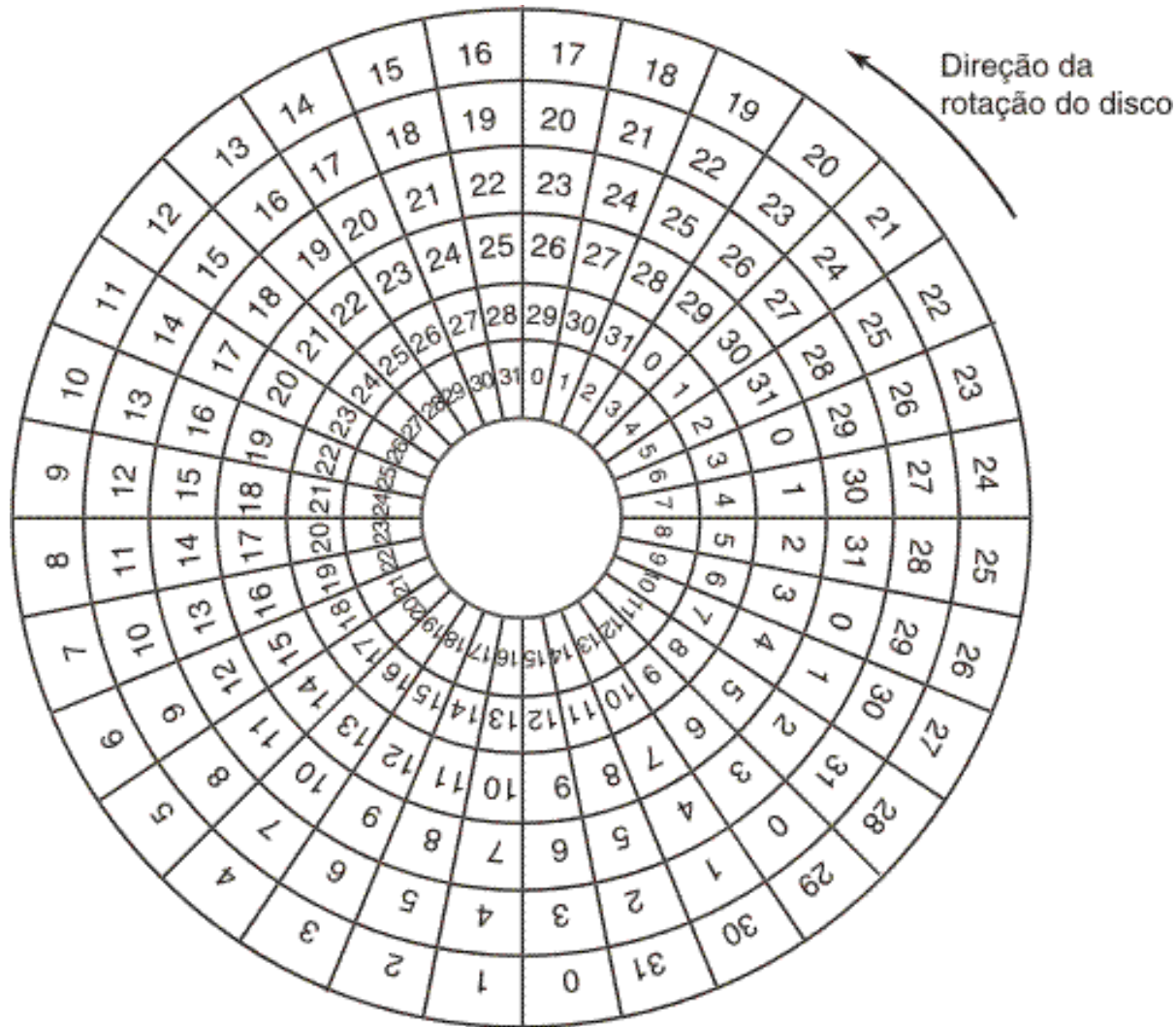
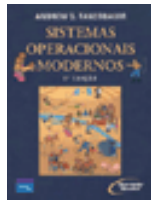
Disco DVD com lado duplo e camada dupla

Formatação de Disco (1)



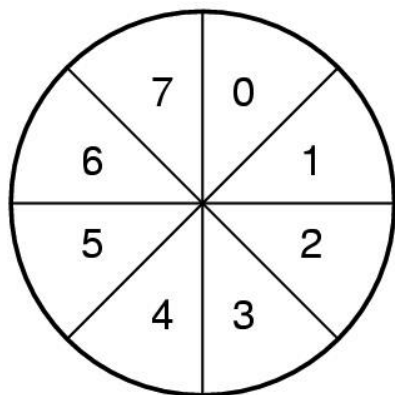
Um setor do disco

Formatação de Disco (2)

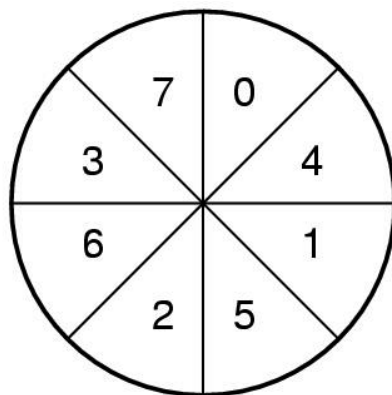


Uma ilustração da torção cilíndrica

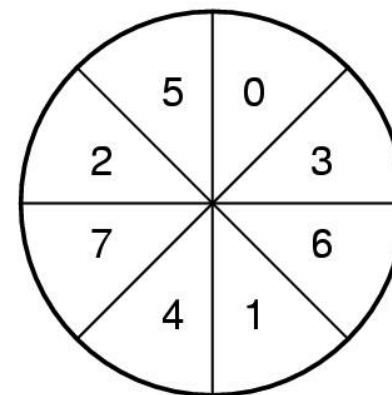
Formatação de Disco (3)



(a)



(b)



(c)

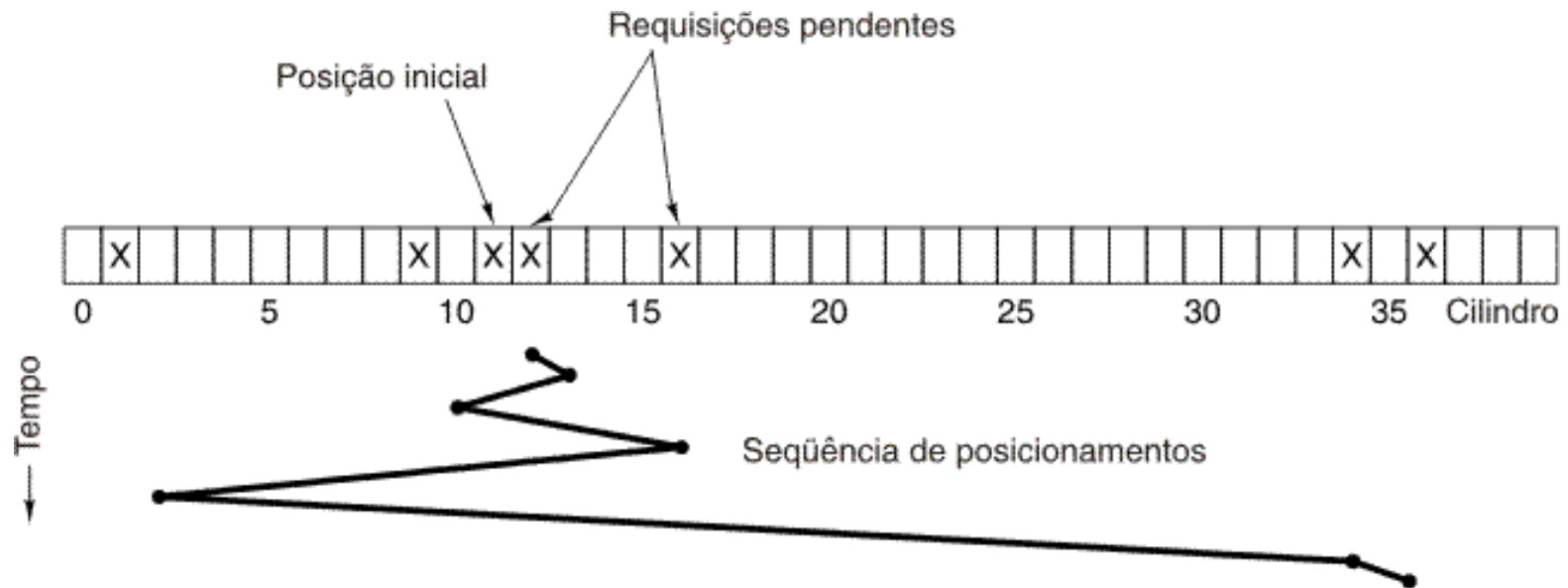
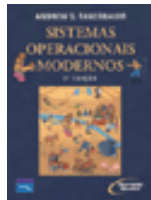
- a) Sem entrelaçamento
- b) Entrelaçamento simples
- c) Entrelaçamento duplo

Algoritmos de Escalonamento de Braço de Disco (1)



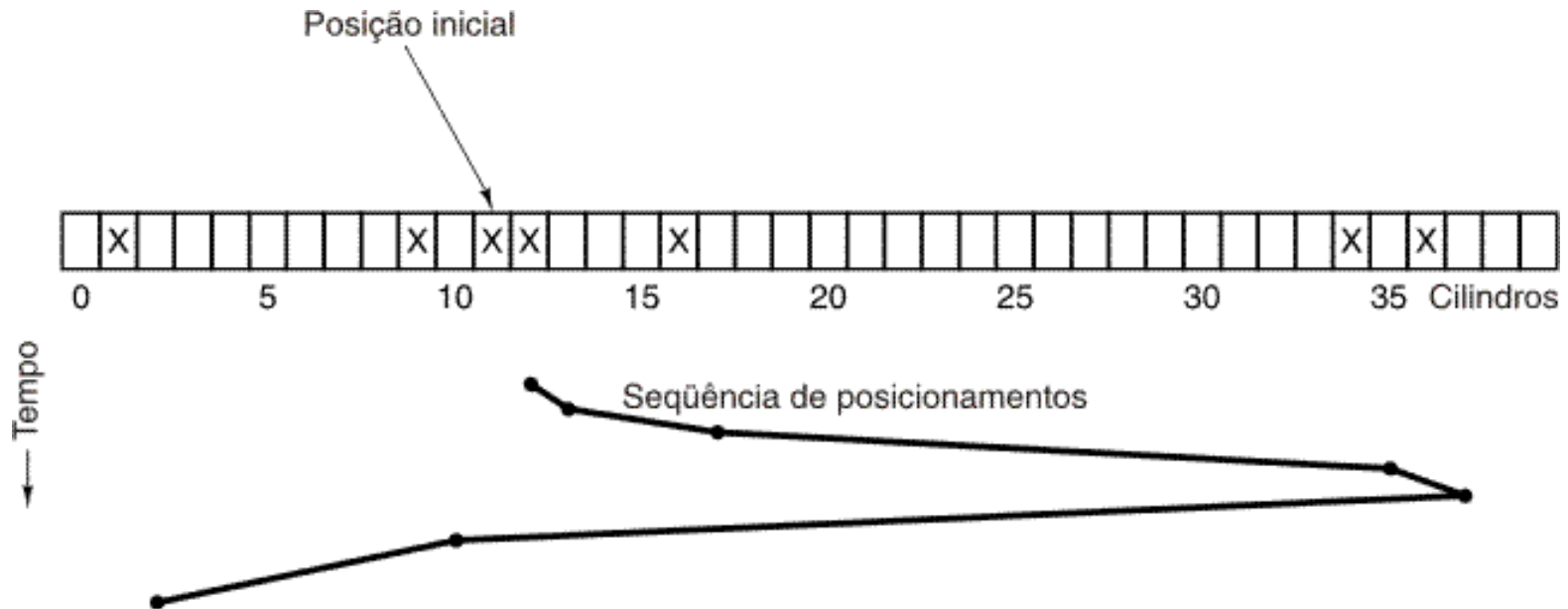
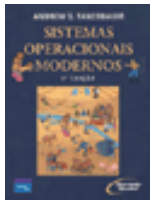
- Tempo necessário para ler ou escrever um bloco de disco é determinado por 3 fatores
 1. tempo de posicionamento
 2. atraso de rotação
 3. tempo de transferência do dado
- Tempo de posicionamento domina
- Checagem de erro é feita por controladores

Algoritmos de Escalonamento de Braço de Disco (2)



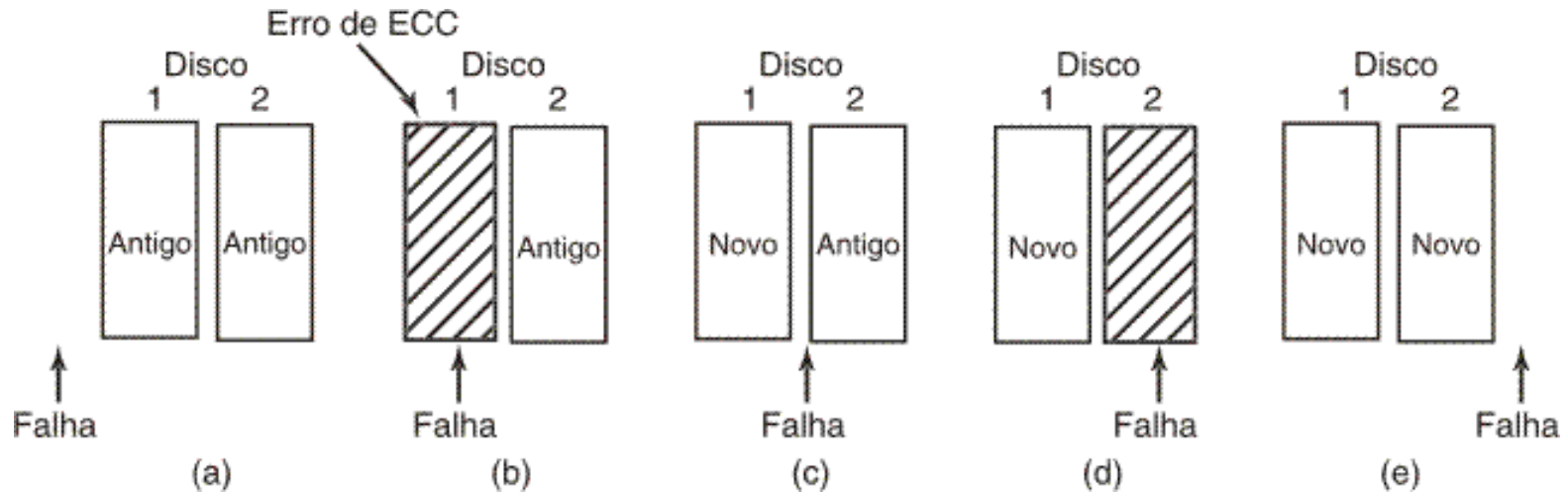
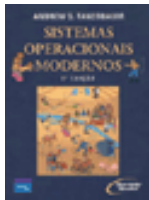
Algoritmo de escalonamento de disco *Posicionamento Mais Curto Primeiro (SSF)*

Algoritmos de Escalonamento de Braço de Disco (3)



O algoritmo do elevador para o escalonamento das requisições do disco

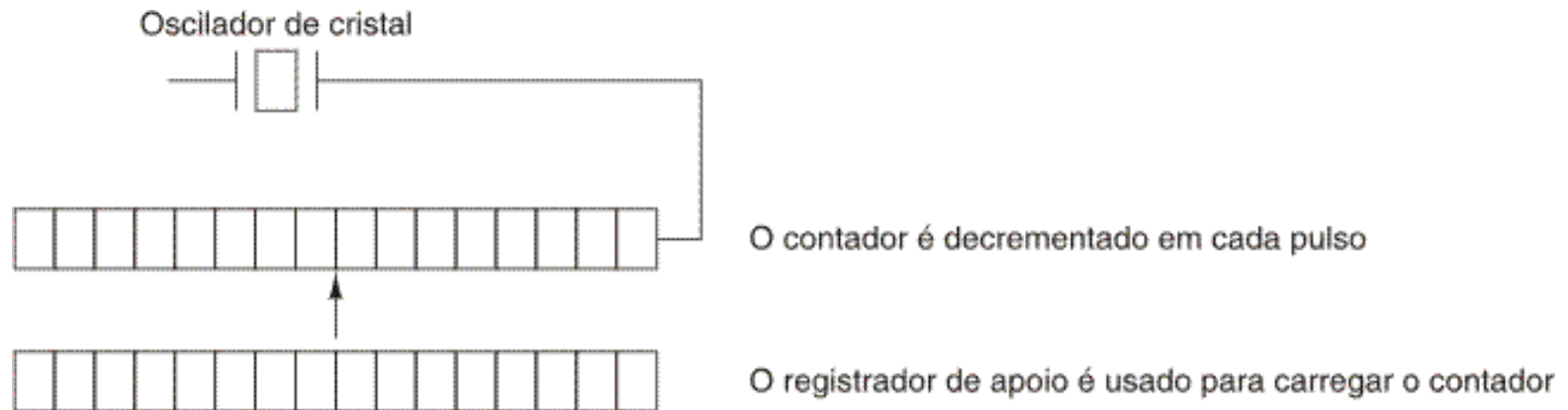
Armazenamento estável



Análise da influência das falhas nas escritas estáveis

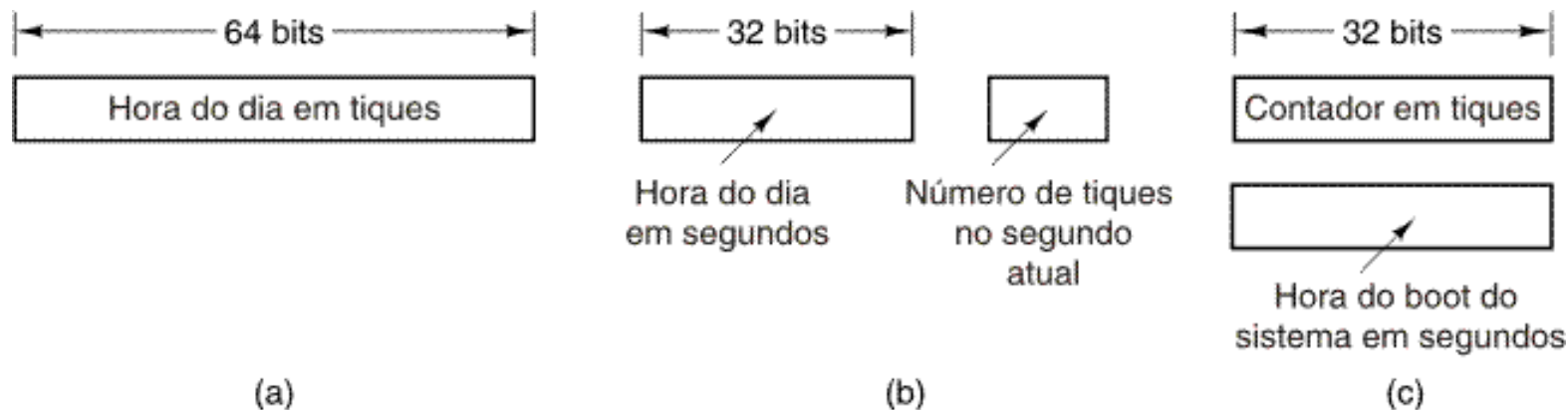
Relógios

Hardware do Relógio



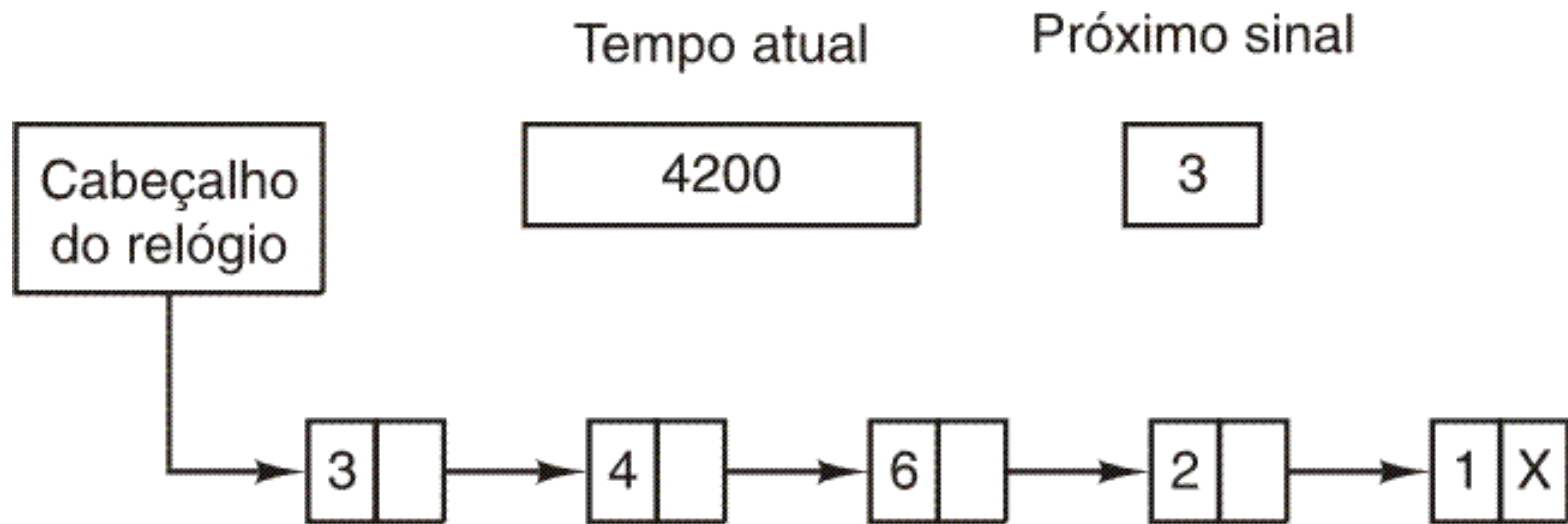
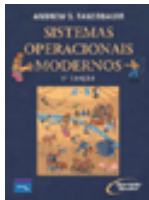
Um relógio programável

Software do Relógio (1)



Três maneiras para manter a hora do dia

Software do Relógio (2)



Simulação de vários temporizadores com um único relógio

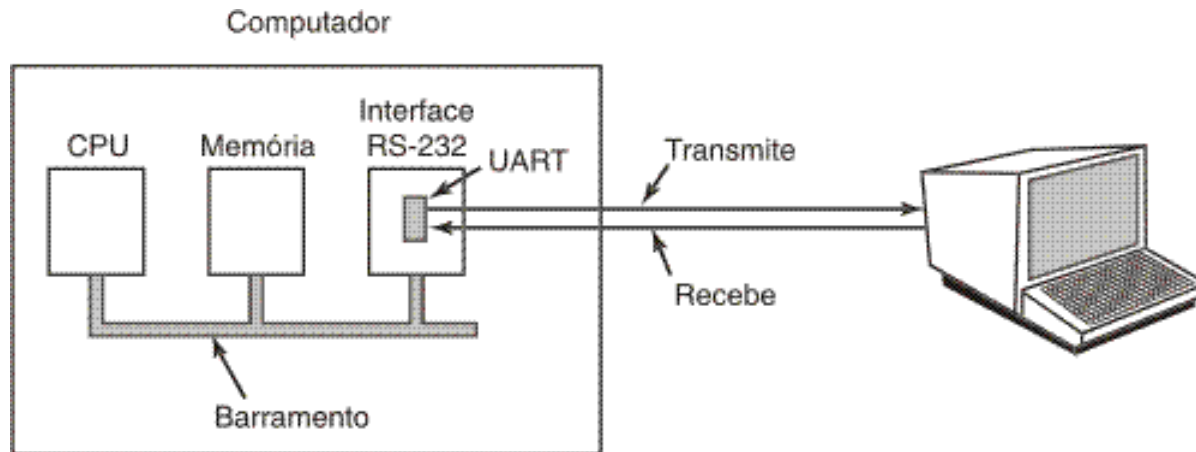
Temporizadores de Software



- Um segundo relógio programável para interrupções de temporização
 - ajustado para causar interrupções em qualquer taxa que um programa precisar
 - sem problemas se a frequência de interrupção é baixa
- Temporizadores de software evitam interrupções
 - núcleo checa se o temporizador de software expirou antes de retornar para o modo usuário
 - quão bem isso funciona depende da taxa de entradas no núcleo

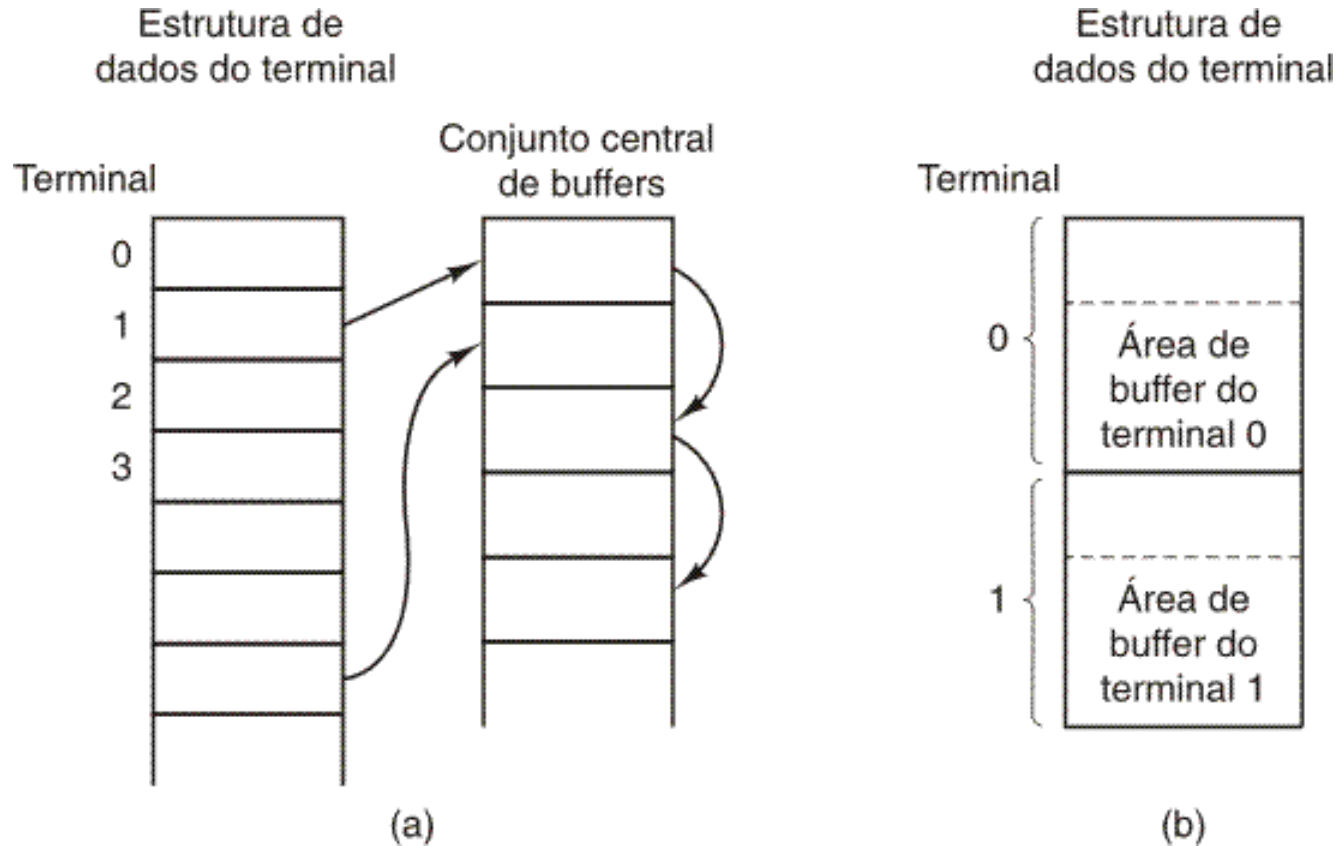
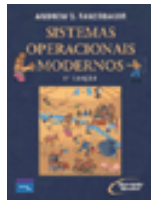
Terminais com Base em Caracteres

Hardware do Terminal RS-232



- Um terminal RS-232 se comunica com um computador bit a bit
- Chamado de linha serial – os bits saem em série, 1 bit por vez
- Windows usa portas COM1 e COM2, primeiro para linhas seriais
- Computador e terminal são completamente independentes

Software de Entrada (1)



a) Conjunto central de buffers

b) Buffer dedicado para cada terminal

Software de Entrada (2)



Caractere	Nome POSIX	Comentário
CTRL-H	ERASE	Retrocede um caractere
CTRL-U	KILL	Apaga a linha toda que está sendo digitada
CTRL-V	LNEXT	Interpreta literalmente o próximo caractere
CTRL-S	STOP	Pára a saída
CTRL-Q	START	Inicia a saída
DEL	INTR	Interrompe o processo (SIGINT)
CTRL-\	QUIT	Força a gravação da imagem da memória (SIGQUIT)
CTRL-D	EOF	Finaliza o arquivo
CTRL-M	CR	Retorno do carro (inalterável)
CTRL-J	LF	Próxima linha (inalterável)

Caracteres tratados de forma especial no modo canônico

Software de Saída

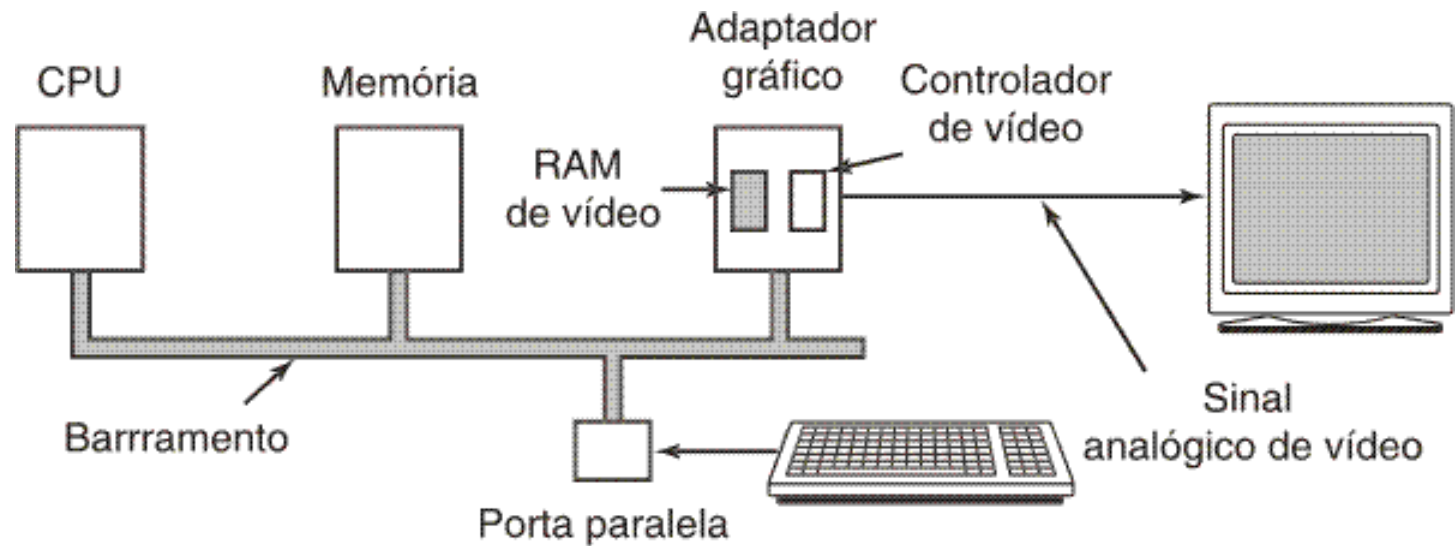


Seqüência de escapes	Significado
ESC [<i>n</i> A	Move <i>n</i> linhas acima
ESC [<i>n</i> B	Move <i>n</i> linhas abaixo
ESC [<i>n</i> C	Move <i>n</i> espaços à direita
ESC [<i>n</i> D	Move <i>n</i> espaços à esquerda
ESC [<i>m</i> ; <i>n</i> H	Move o cursor para (<i>m</i> , <i>n</i>)
ESC [<i>s</i> J	Limpa a tela a partir do cursor (0 até o final, 1 desde o início, 2 tudo)
ESC [<i>s</i> K	Limpa a linha a partir do cursor (0 até o final, 1 desde o início, 2 tudo)
ESC [<i>n</i> L	Insere <i>n</i> linhas a partir da posição do cursor
ESC [<i>n</i> M	Remove <i>n</i> linhas a partir da posição do cursor
ESC [<i>n</i> P	Remove <i>n</i> caracteres a partir da posição
ESC [<i>n</i> @	Insere <i>n</i> caracteres a partir da posição do cursor
ESC [<i>n</i> m	Habilita substituição do tipo <i>n</i> (0=normal, 4=negrito, 5=piscante, 7=invertido)
ESC M	Rola a tela para trás se o cursor está na linha do topo

As seqüências de escapes ANSI

- aceitas pelo driver do terminal na saída
- ESC é o caractere de escape ASCII (0x1B)
- *n*, *m*, e *s* são parâmetros numéricos opcionais

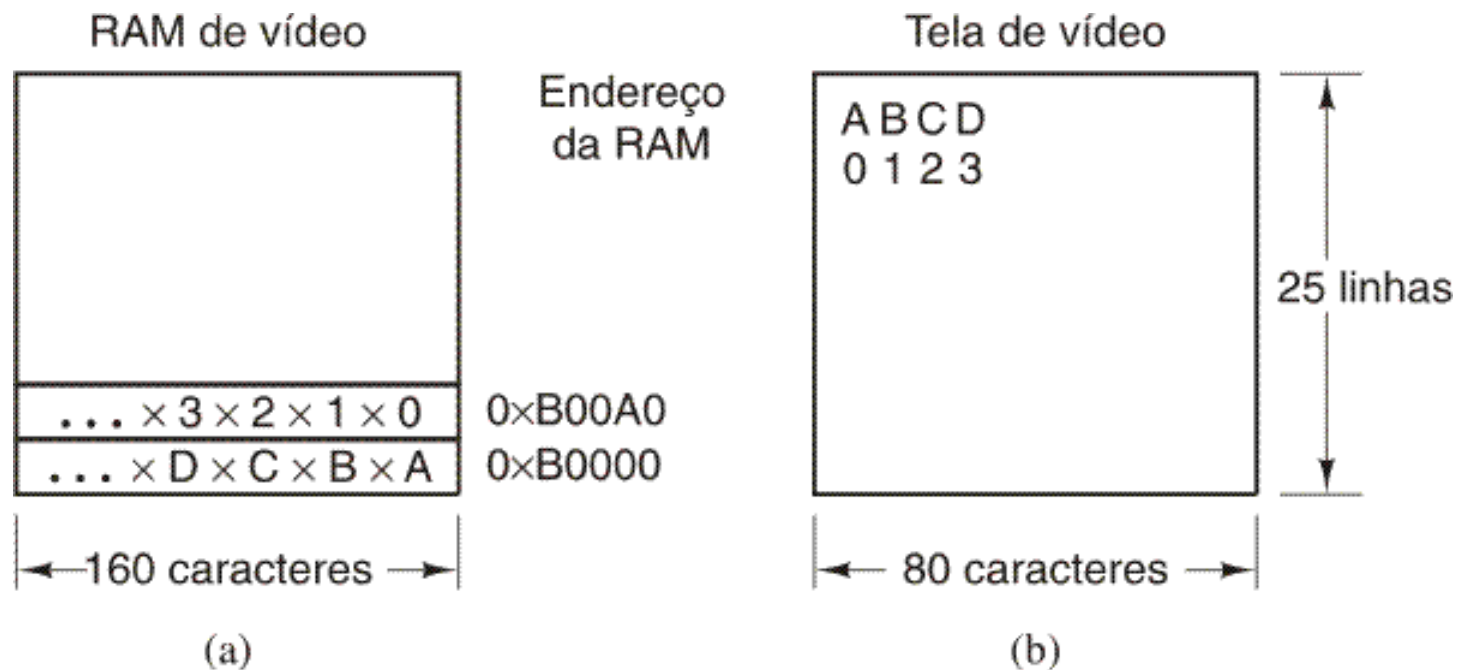
Hardware de Vídeo (1)



Vídeos mapeados na memória

- driver escreve diretamente na RAM de vídeo do monitor

Hardware de Vídeo (2)



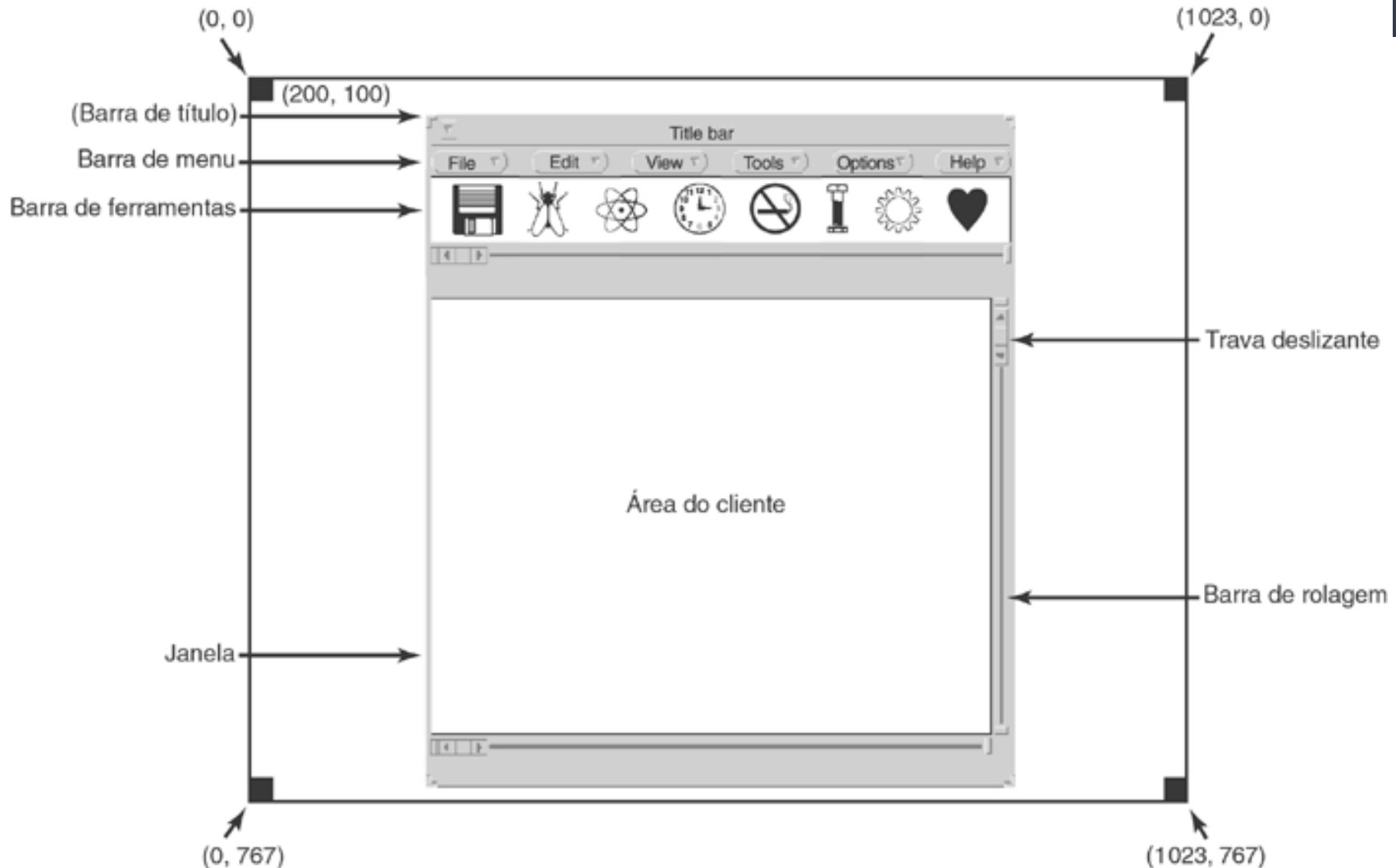
- Uma imagem da RAM de vídeo
 - tela monocromática simples
 - modo caractere
- Tela correspondente
 - os x's são bytes de atributos

Software de Entrada



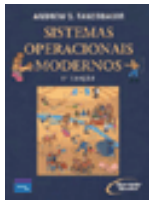
- Driver de teclado entrega um número
 - driver converte para caracteres
 - usa uma tabela ASCII
- Exceções, adaptações necessárias para outras linguagens
 - muitos SOs fornecem mapas de teclas ou páginas de códigos carregáveis

Software de Saída para Janelas (1)



Um exemplo de janela em $(200, 100)$ em um vídeo XGA

Software de Saída para Janelas (2)



```
#include <windows.h>
```

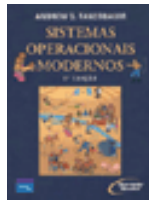
```
int WINAPI WinMain(HINSTANCE h, HINSTANCE, hprev, char *szCmd, int iCmdShow)
{
    WNDCLASS wndclass;           /* objeto-classe para esta janela */
    MSG msg;                     /* mensagens que chegam são aqui armazenadas */
    HWND hwnd;                   /* ponteiro para o objeto janela */

    /* Inicializa wndclass */
    wndclass.lpfnWndProc = WndProc; /* indica qual procedimento chamar */
    wndclass.lpszClassName = "Program name"; /* Texto para a Barra de Título */
    wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION); /* carrega ícone do programa */
    wndclass.hCursor = LoadCursor(NULL, IDC_ARROW); /* carrega cursor do mouse */

    RegisterClass(&wndclass); /* avisa o Windows sobre wndclass */
    hwnd = CreateWindow ( ... ) /* aloca espaço para a janela */
    ShowWindow(hwnd, iCmdShow); /* mostra a janela na tela */
    UpdateWindow(hwnd); /* avisa a janela para pintar-se */
}
```

Esqueleto de um programa principal para janelas (parte 1)

Software de Saída para Janelas (3)



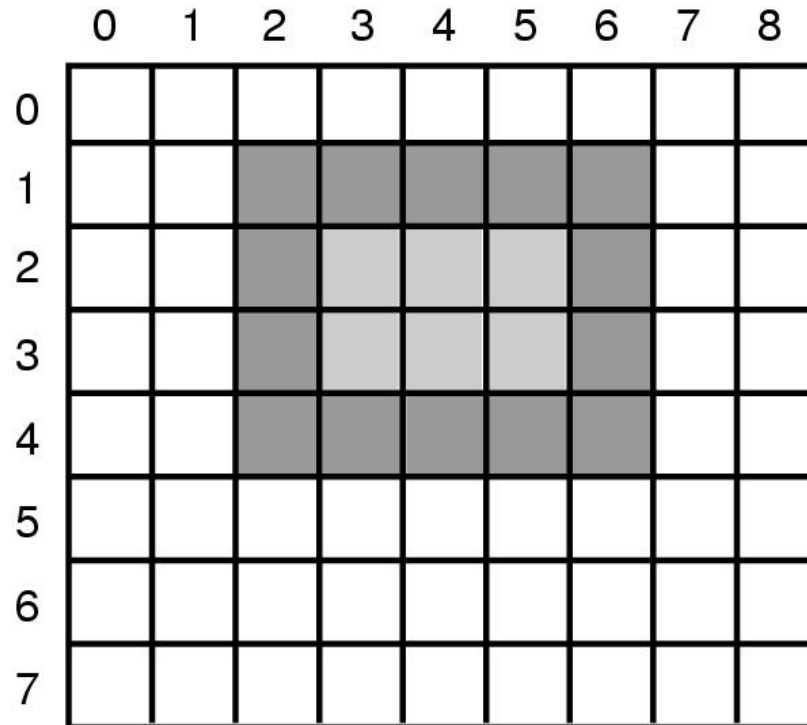
```
while (GetMessage(&msg, NULL, 0, 0)) { /* obtém mensagem da fila */
    TranslateMessage(&msg); /* traduz a mensagem */
    DispatchMessage(&msg); /* envia msg para o procedimento apropriado */
}
return(msg.wParam);
}

long CALLBACK WndProc(HWND hwnd, UINT message, UINT wParam, long lParam)
{
    /* Declarações são colocadas aqui */

    switch (message) {
        case WM_CREATE:    ... ; return ... ; /* cria janela */
        case WM_PAINT:     ... ; return ... ; /* repinta conteúdo da janela */
        case WM_DESTROY:  ... ; return ... ; /* destrói janela */
    }
    return(DefWindowProc(hwnd, message, wParam, lParam)); /* default */
}
```

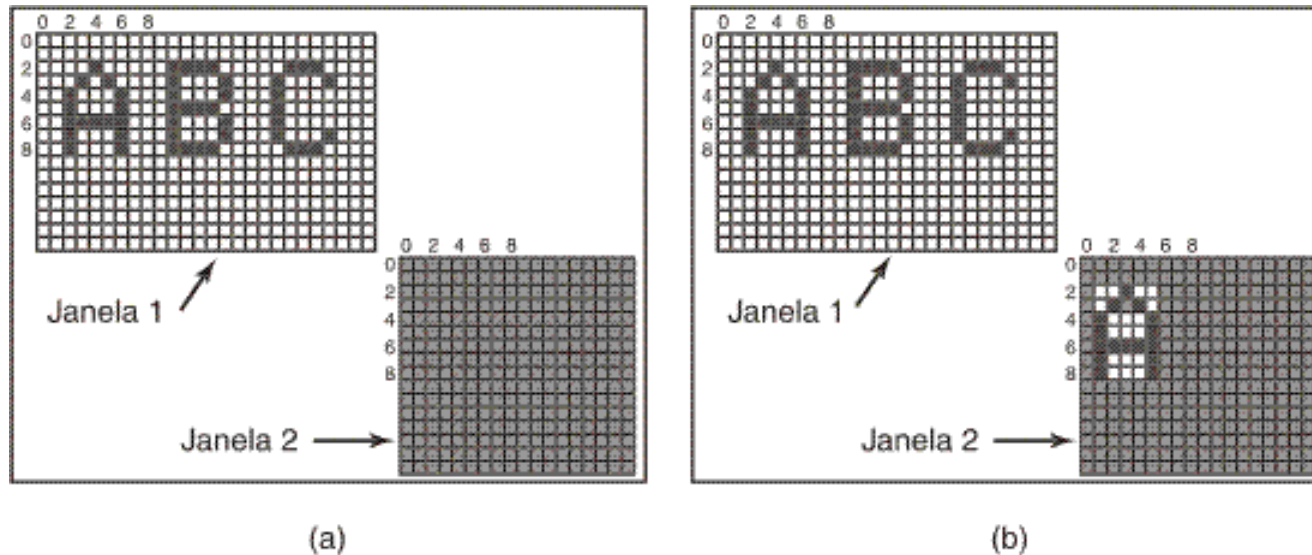
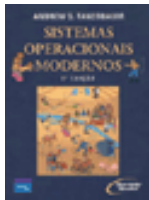
Esqueleto de um programa principal para janelas (parte 2)

Software de Saída para Janelas (4)



Exemplo de um retângulo desenhado usando
Rectangle

Software de Saída para Janelas (5)



- Copiando mapas de bits usando *BitBlt*
 - a) antes
 - b) depois

Software de Saída para Janelas (6)



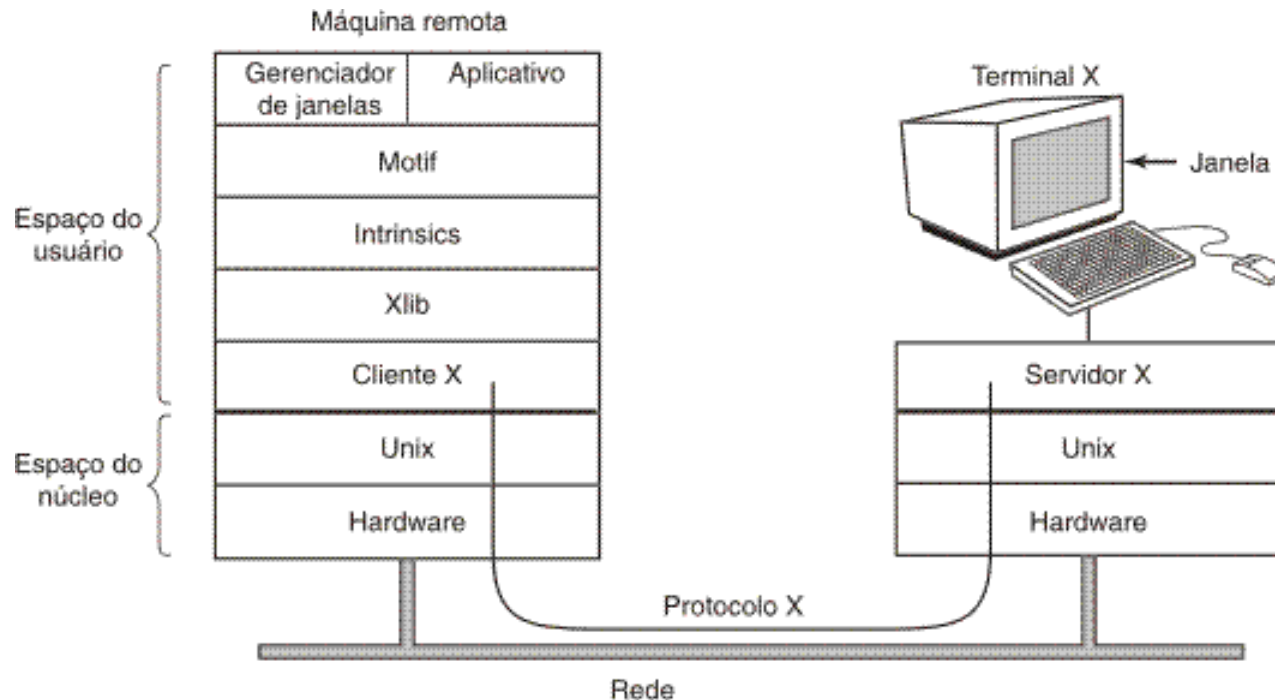
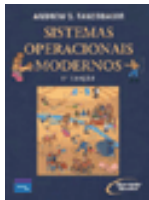
20 pt: abcdefgh

53 pt: abcdefgh

81 pt: abcdefgh

Exemplos de contornos de caracteres de diferentes
tamanhos em pontos

Terminais de rede X Windows (1)



Clientes e servidores no sistema X Windows do MIT

X Windows (2)



```
#include <X11/Xlib.h>
#include <X11/Xutil.h>

main(int argc, char *argv[])
{
    Display disp;                /* identificador do servidor */
    Window win;                  /* identificador da janela */
    GC gc;                       /* identificador do contexto gráfico */
    XEvent event;                /* armazenamento para um evento */
    int running = 1;

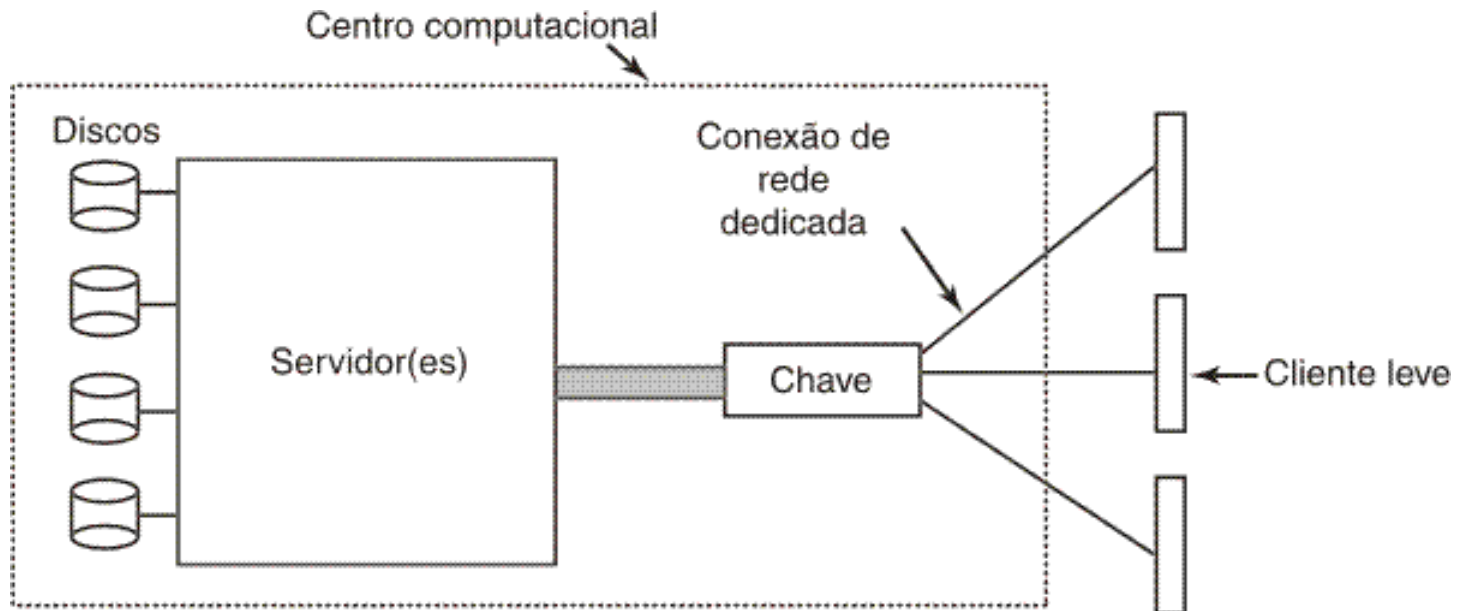
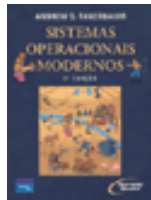
    disp = XOpenDisplay("display_name"); /* conecta ao servidor X */
    win = XCreateSimpleWindows(disp, ...); /* aloca memória para a nova janela */
    XSetStandardProperties(disp, ...);    /* anuncia a nova janela para o gerenciador de janelas */
    gc = XCreateGC(disp, win, 0, 0);      /* cria contexto gráfico */
    XSelectInput(disp, win, ButtonPressMask | KeyPressMask | ExposureMask);
    XMapRaised(disp, win);              /* mostra a janela; envia evento de exposição de janela */

    While (running) {
        XNextEvent(disp, &event);        /* obtém próximo evento */
        switch (event.type) {
            case Expose: ...; break;
            case ButtonPress; ...; break; /* processa clique do mouse */
            case Keypress; ...; break;   /* processa entrada do teclado */
        }
    }

    XFreeGC(disp, gc);                /* libera contexto gráfico */
    XDestroyWindow(disp, win);         /* desaloca espaço de memória da janela */
    XCloseDisplay(disp);               /* termina a conexão de rede */
}
```

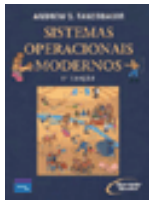
Esqueleto de um programa de aplicação
usando o sistema X Windows

O Terminal de Rede SLIM (1)



A arquitetura do sistema de terminal SLIM

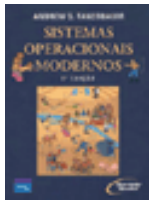
O Terminal de Rede SLIM (2)



Mensagem	Significado
SET	Atualiza um retângulo com novos pixels
FILL	Preenche um retângulo com valor de um pixel
BITMAP	Expande um mapa de bits para preencher um retângulo
COPY	Copia um retângulo de uma parte do buffer de quadros para outra
CSCS	Converte um retângulo de cor de televisão (YUV) para RGB

Mensagens usadas no protocolo SLIM do servidor para os terminais

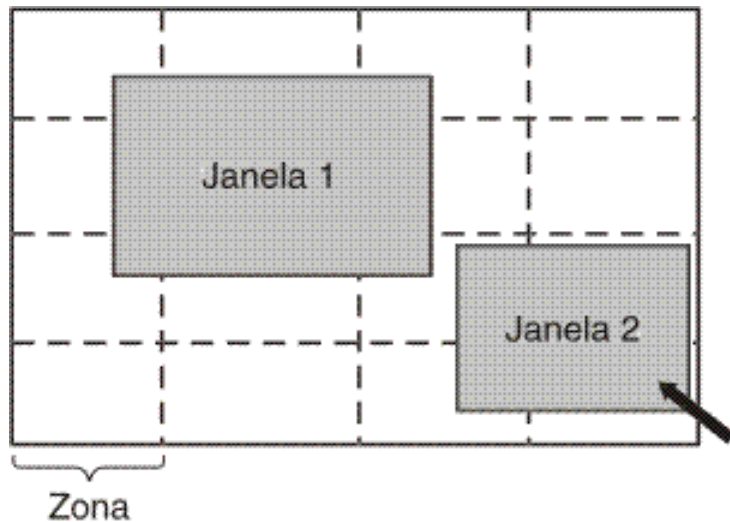
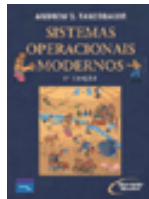
Gerenciamento de Energia (1)



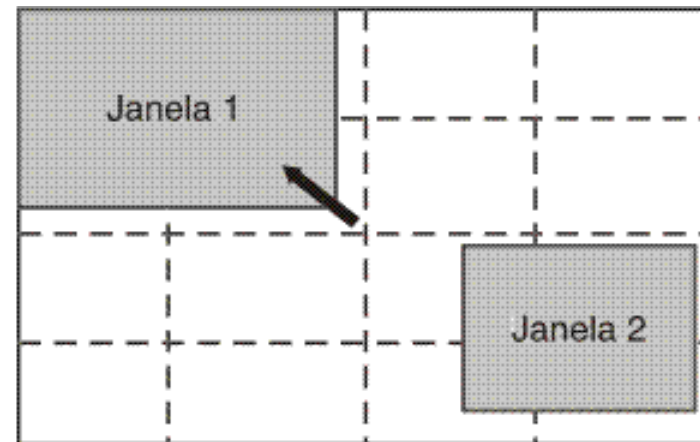
Dispositivo	Li <i>et al.</i> (1994)	Lorch e Smith (1998)
Monitor de vídeo	68%	39%
CPU	12%	18%
Disco rígido	20%	12%
Modem		6%
Som		2%
Memória	0,5%	1%
Outros		22%

Consumo de energia de várias partes de um computador laptop

Gerenciamento de Energia (2)



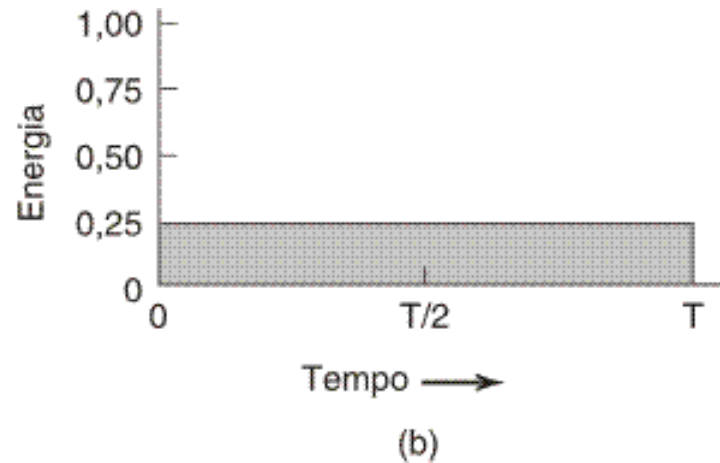
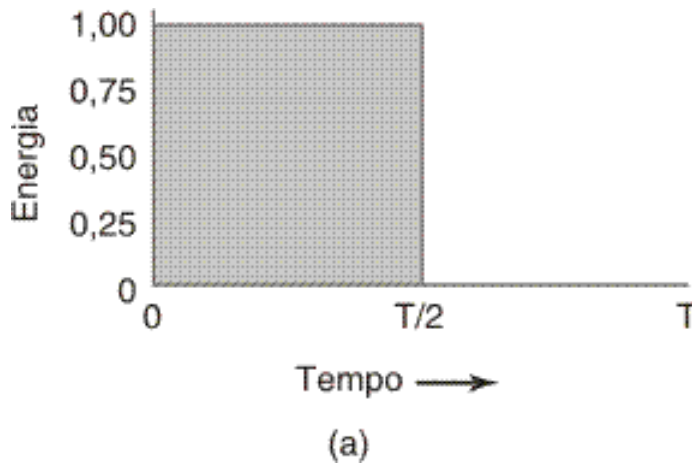
(a)



(b)

O uso de zonas para reanimar a iluminação do monitor de vídeo

Gerenciamento de Energia (3)



- Execução em velocidade máxima do relógio
- Cortando a voltagem pela metade
 - corta a velocidade do relógio também pela metade,
 - consumo de energia cai para 4 vezes menos

Gerenciamento de Energia (4)



- Dizer aos programas para usar menos energia
 - pode significar experiências mais pobres para o usuário
- Exemplos
 - muda de saída colorida para preto e branco
 - reconhecimento de fala reduz vocabulário
 - menos resolução ou detalhe em uma imagem