

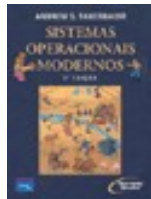
Capítulo 5

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**

Discos

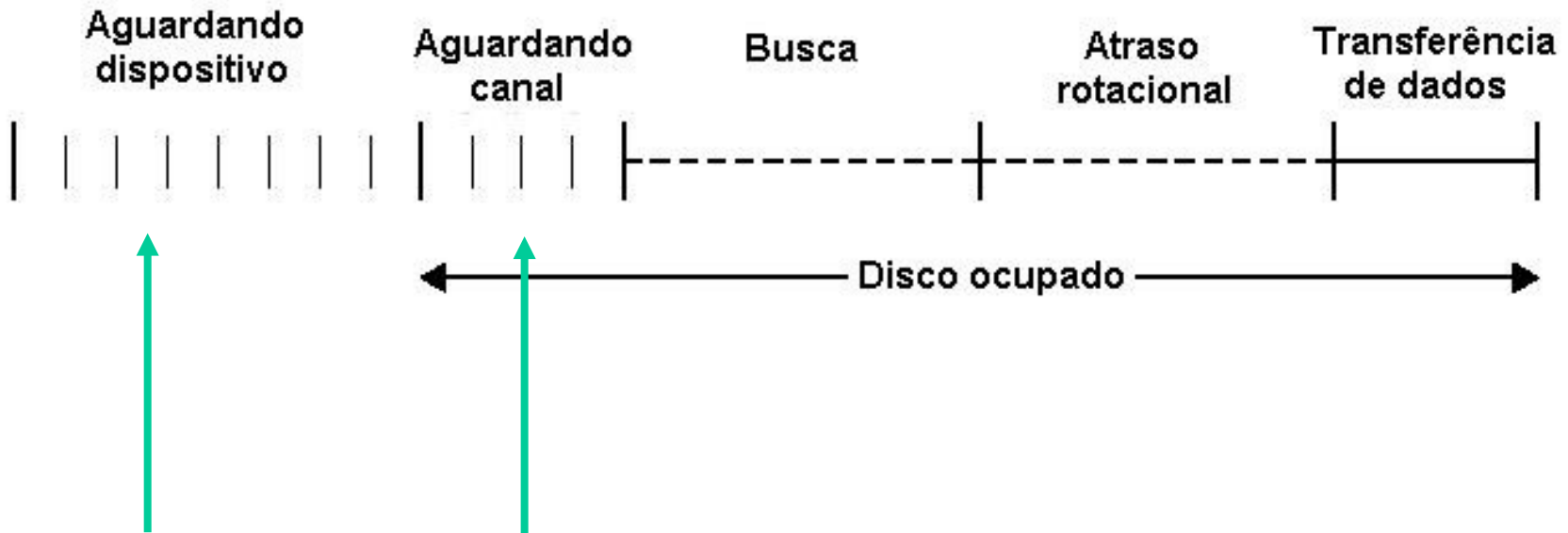
Hardware do Disco (1)



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

Velocidade



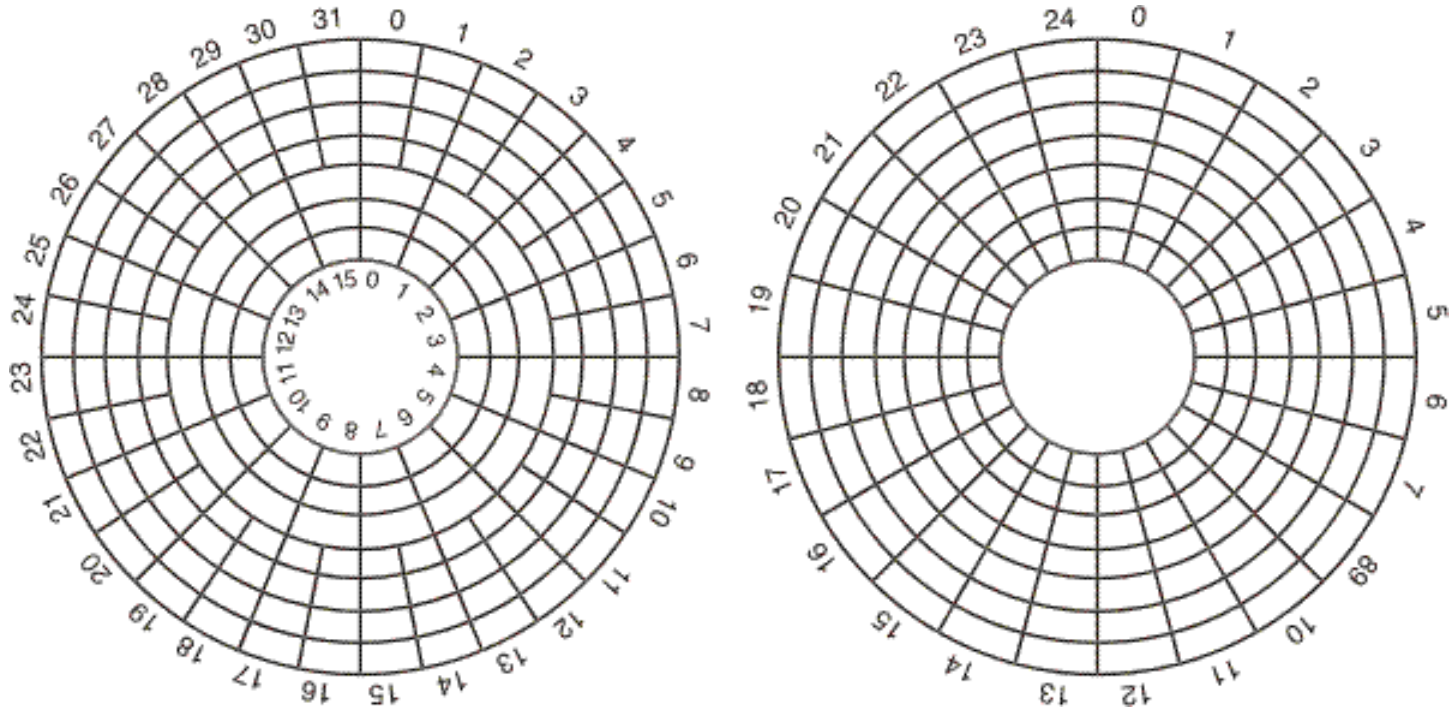
Não depende do dispositivo de disco magnético.
Somente do sistema operacional e das condições de uso.

Velocidade (2)



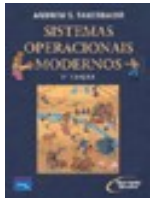
- Tempo de Busca
 - Posicionar a cabeça de leitura na trilha correta
- Atraso Rotacional
 - Esperando dado “rodar” para baixo da cabeçote
- Tempo de Acesso = Tempo de Busca + Atraso
- Taxa de Transferência

Hardware do Disco (2)



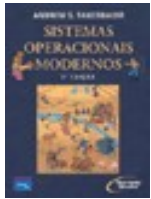
- Geometria física de um disco com duas zonas
- Uma possível geometria virtual para esse disco

RAID



- Redundant Array of Independent Disks
 - Conjunto Redundante de Discos Independentes
- 6 níveis mais comuns
- Não é uma hierarquia
 - O nível 3 não é necessariamente melhor que o 2
- Conjunto de discos físicos vistos como um só pelo sistema operacional

RAID (2)



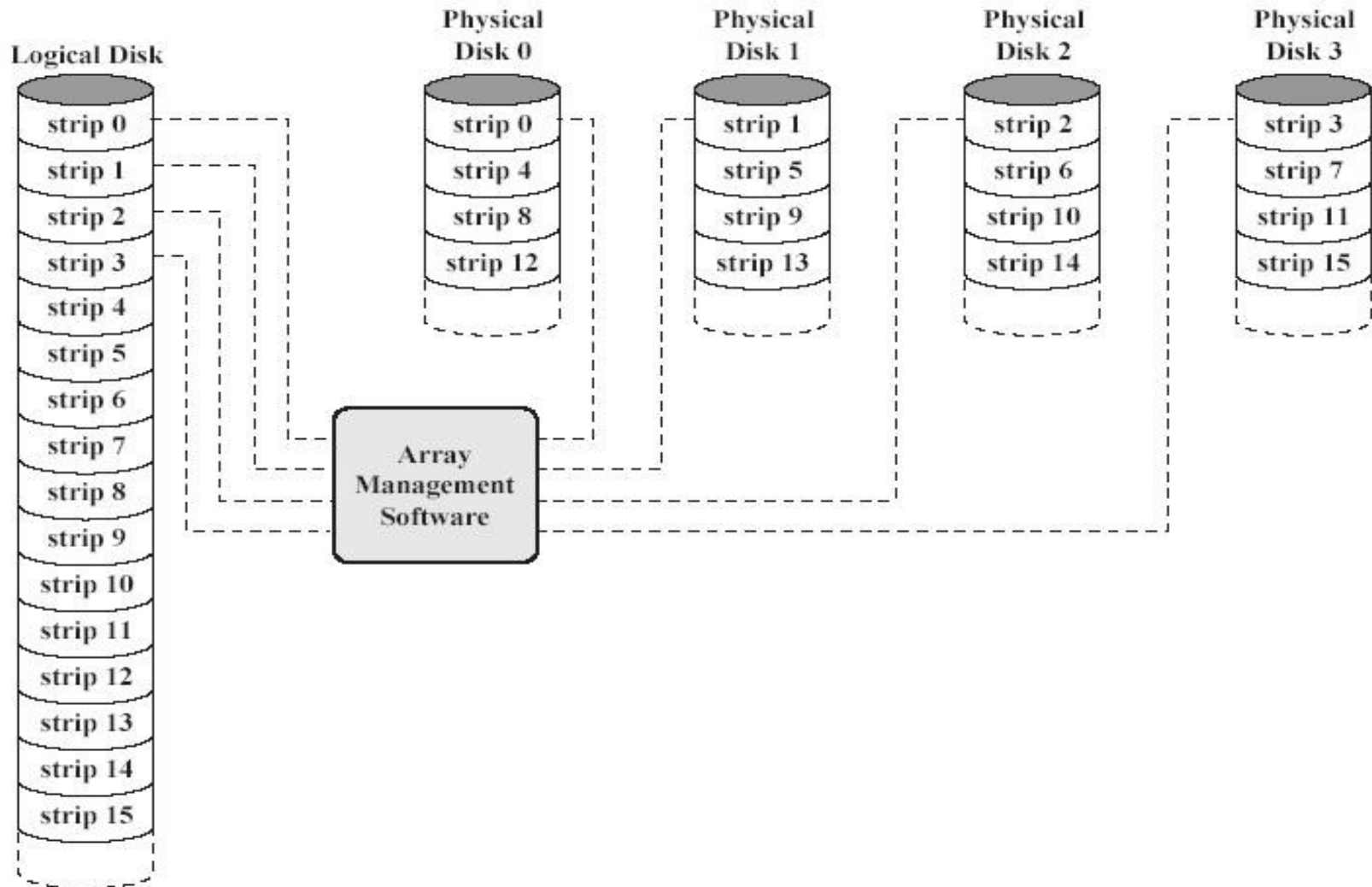
- Dados distribuídos pelos discos físicos
- Pode-se armazenar dados de paridade
- Acesso em paralelo aos discos
- Substituir discos de grande capacidade (caros) por vários de menor capacidade (mais baratos)

RAID 0



- Não há redundância nem replicação
- Dados divididos nos vários discos
- Uma tira em cada disco
 - Cada tira podem ser de 32, 64 ou 128Kb
- Mais rápido
 - Múltiplas requisições de dados provavelmente serão em discos diferentes
 - Discos funcionando em paralelo
 - Um conjunto de dados estará dividido em vários discos

RAID 0 (2)



RAID 1

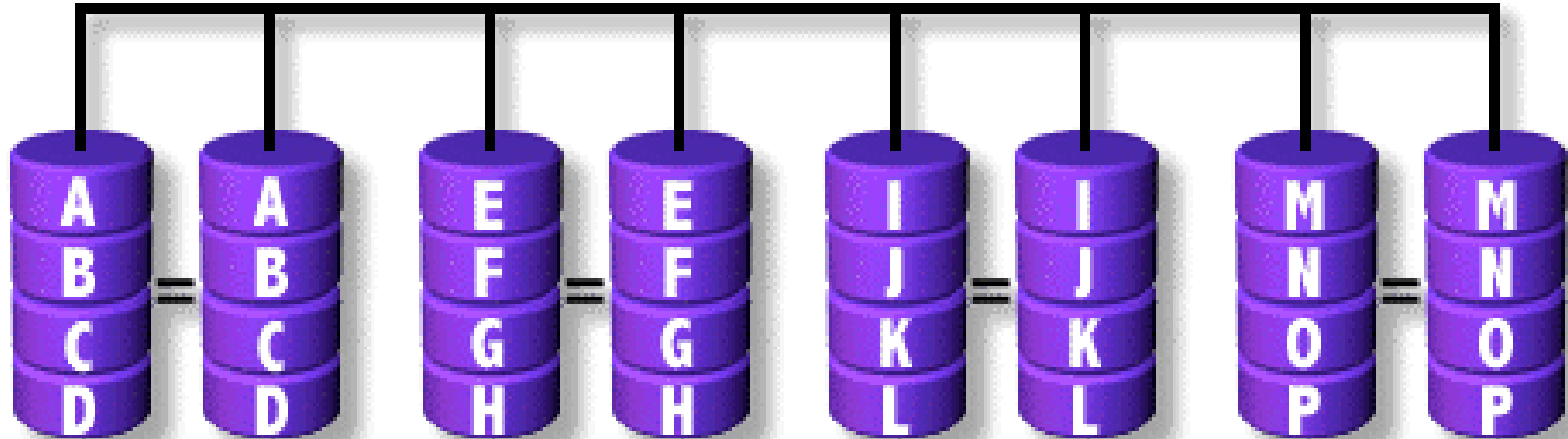


- Discos espelhados
- Dados divididos em discos
- 2 copias de cada tira em discos separados
- Le de qualquer um
- Escreve nos dois
- Recuperação simples
 - Inverte disco com problema e refaz espelho
 - Sem down time
- Caro

RAID 1



RAID 1



COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

RAID 2

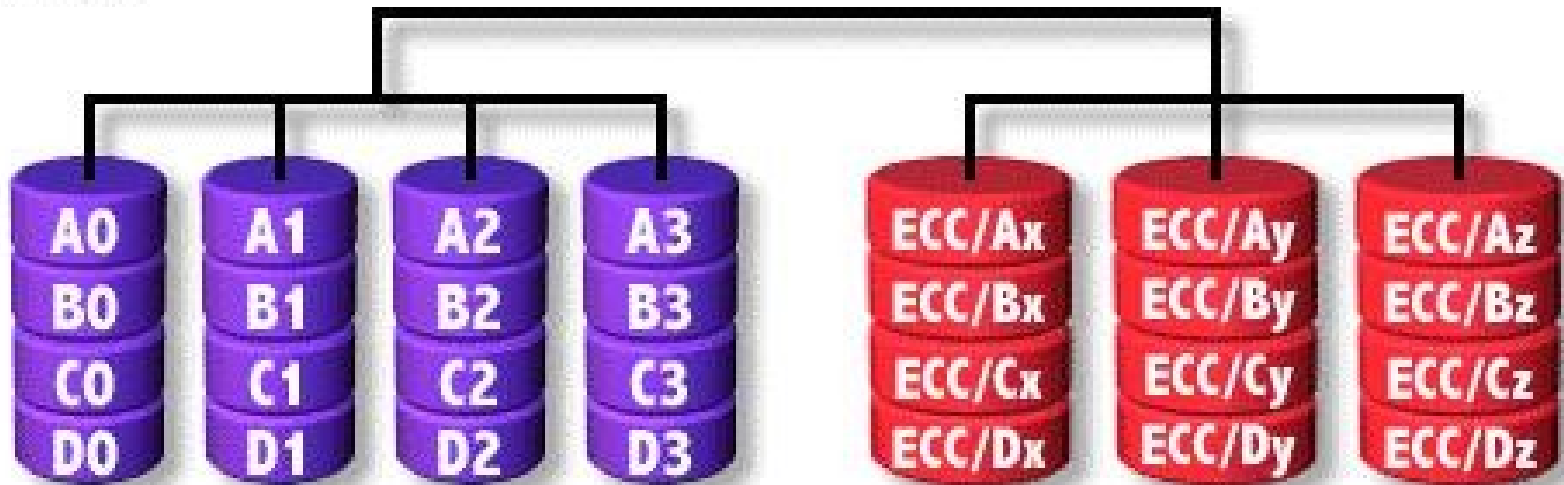


- Discos são sincronizados
- Tiras muito pequenas
 - Um byte ou palavra
- Correção de erros é calculada de acordo com os bits correspondentes nos discos
- Múltiplos discos de paridade armazenam códigos de Hamming para correção de erros
- Muita redundância
 - Caro
 - Não usado

RAID 2 (2)



RAID 2

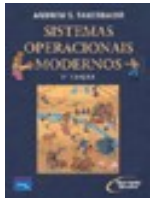


A0 to A3=Word A; B0 to B3 = Word B;
C0 to C3=Word C; D0 to D3 = Word D

ECC/Ax to Az=Word A ECC; ECC/Bx to Bz = Word B ECC;
ECC/Cx to Cz=Word C ECC; ECC/Dx to Dz = Word D ECC

COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

RAID 3

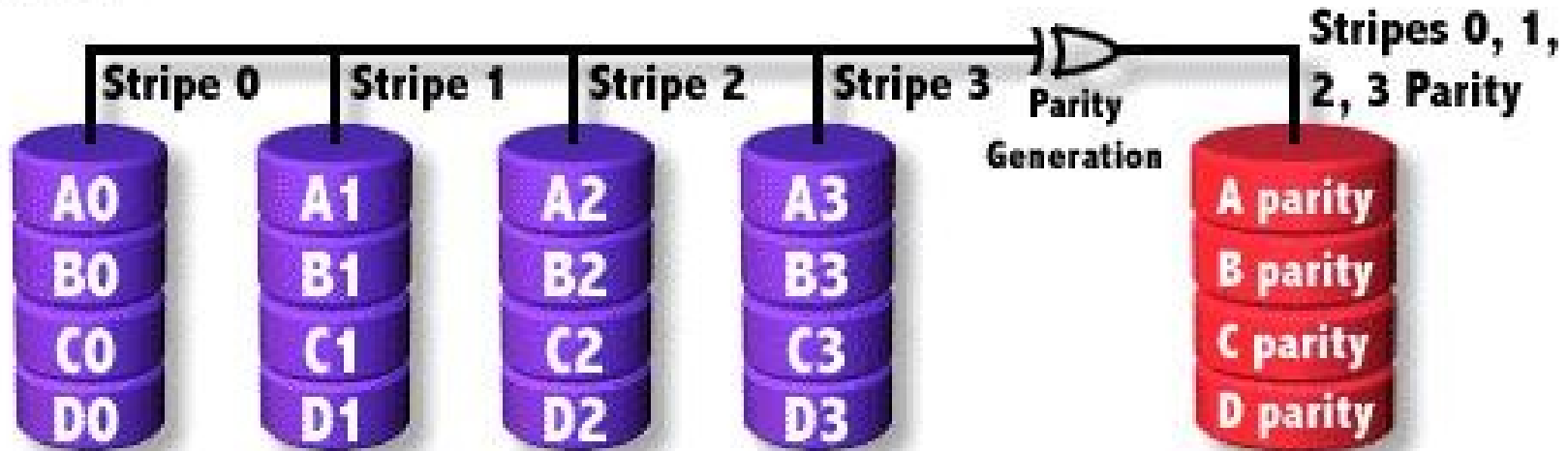


- Similar ao RAID 2
- Apenas um disco redundante, não importa tamanho do conjunto
- Bit simples de paridade para cada conjunto correspondente de bits
- Se um disco falhar basta substituí-lo e ele pode ser reconstruído a partir dos outros e da paridade
- Elevadas taxas de transferência de dados

RAID 3

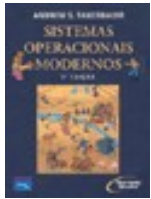


RAID 3



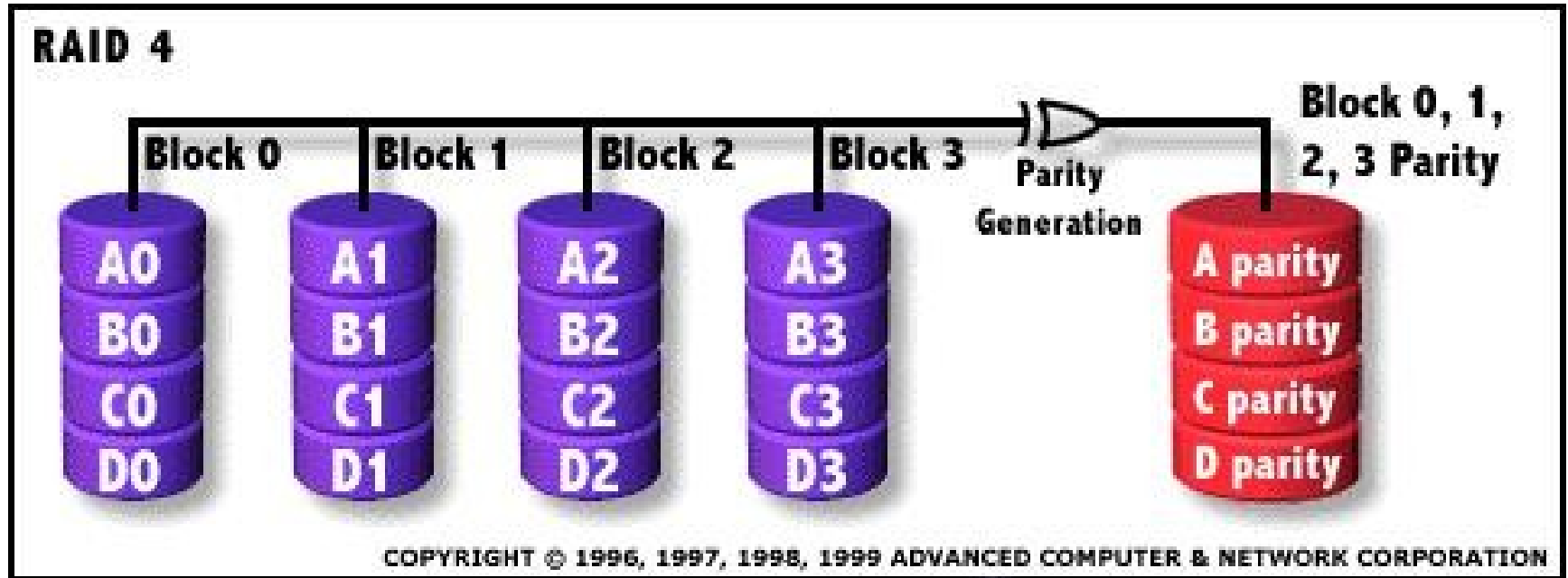
COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

RAID 4

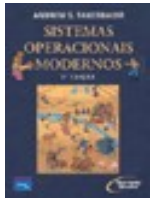


- Cada disco opera independentemente
- Bom para altas taxas de transferência
- Tiras grandes
- Paridade calculada bit-a-bit pelas tiras de cada disco
- Paridade armazenada no disco de paridade
- Bom para altas taxas de requisições de E/S (não transferência)

RAID 4



RAID 5

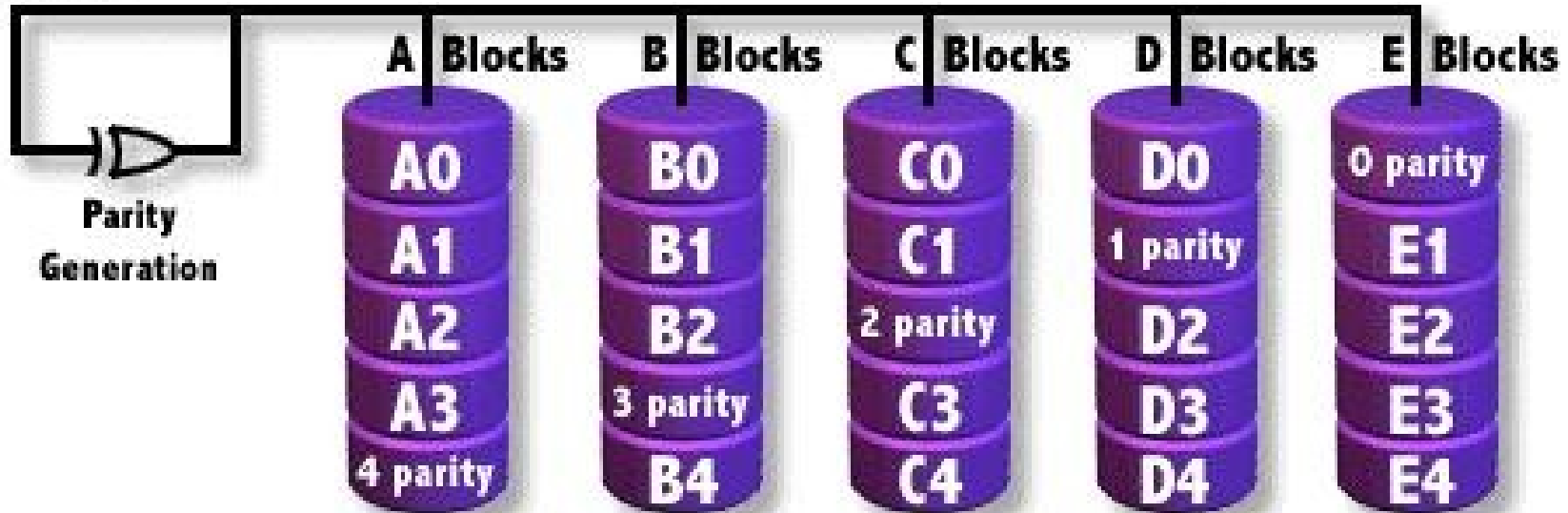


- Igual RAID 4
- Paridades são divididas em todos os discos
- Evita gargalos no disco de paridade
- Geralmente usado em servidores de rede

RAID 5



RAID 5



COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

RAID 6

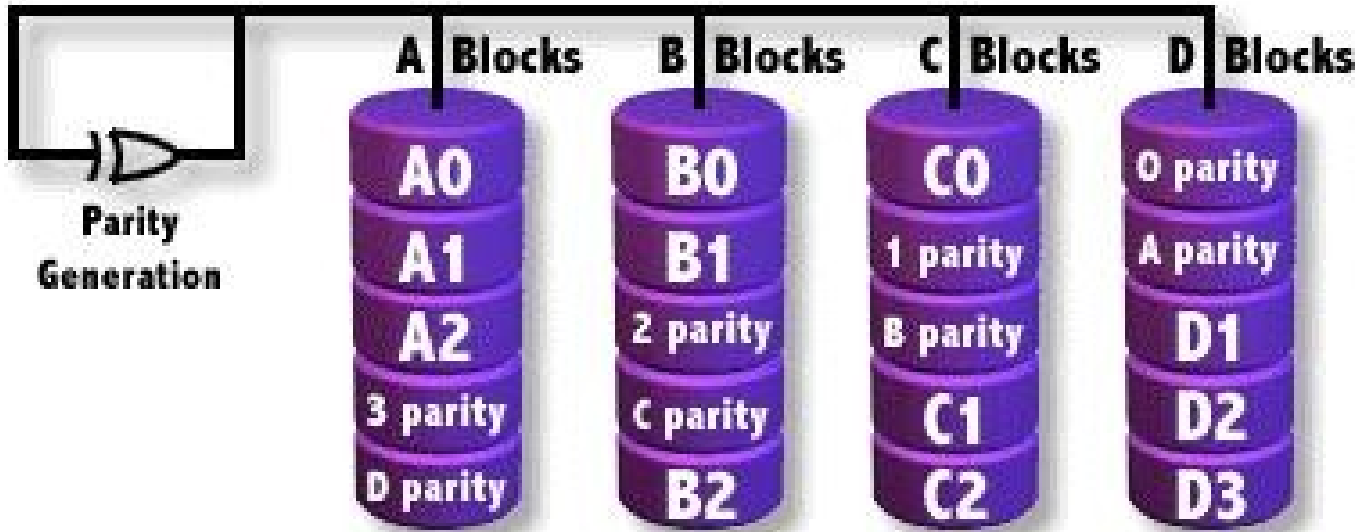


- Quase igual RAID 5
- Usa 2 algoritmos de cálculo de paridade
- Pouco usado

RAID 6

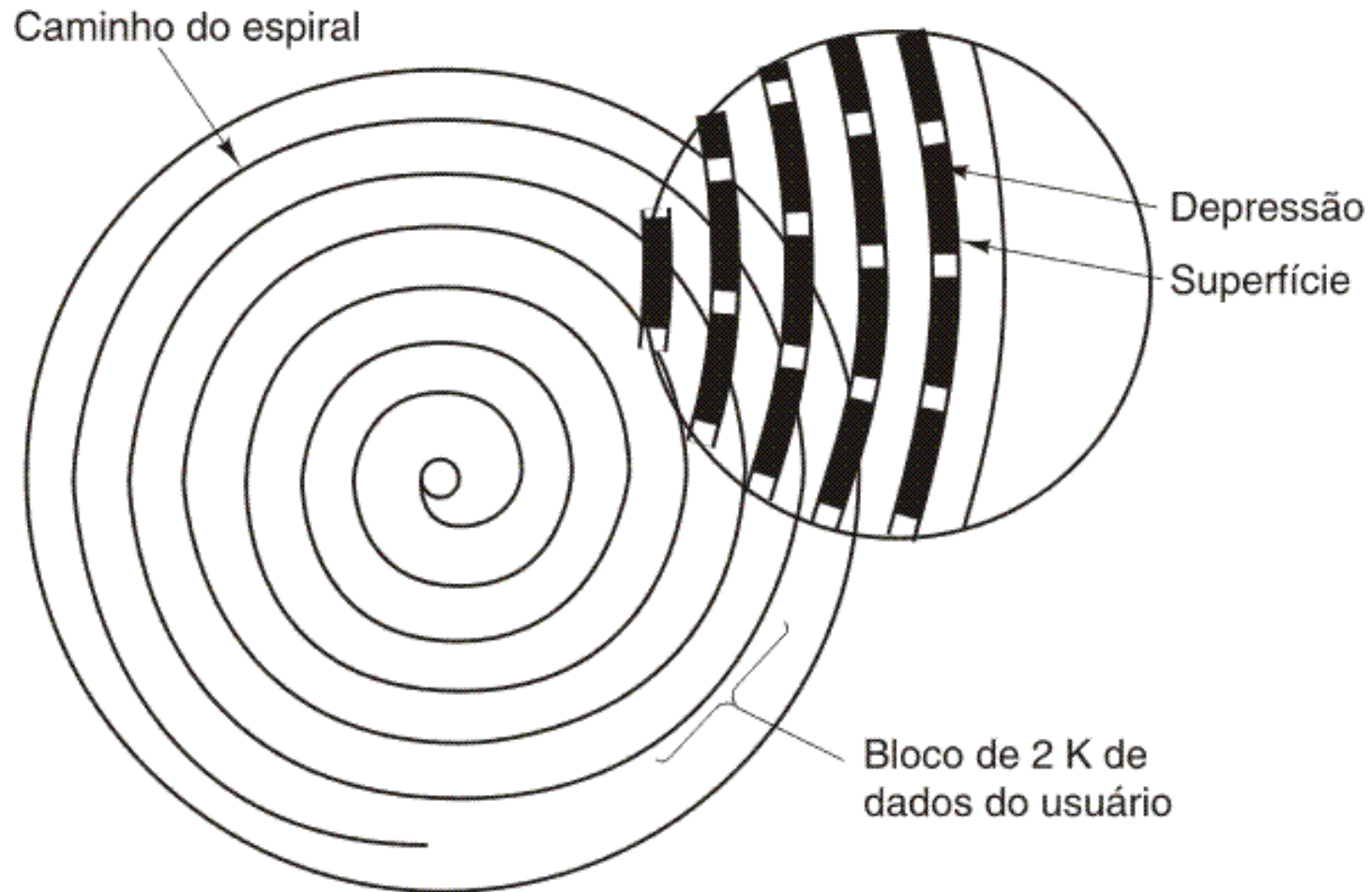


RAID 6



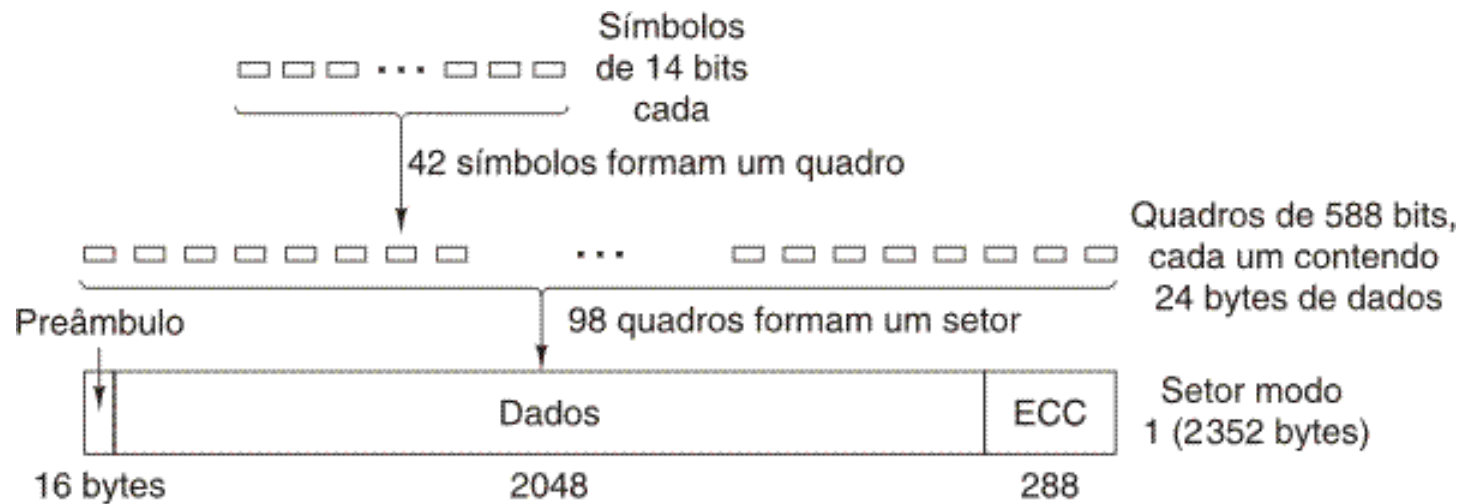
COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

Hardware do Disco (3)



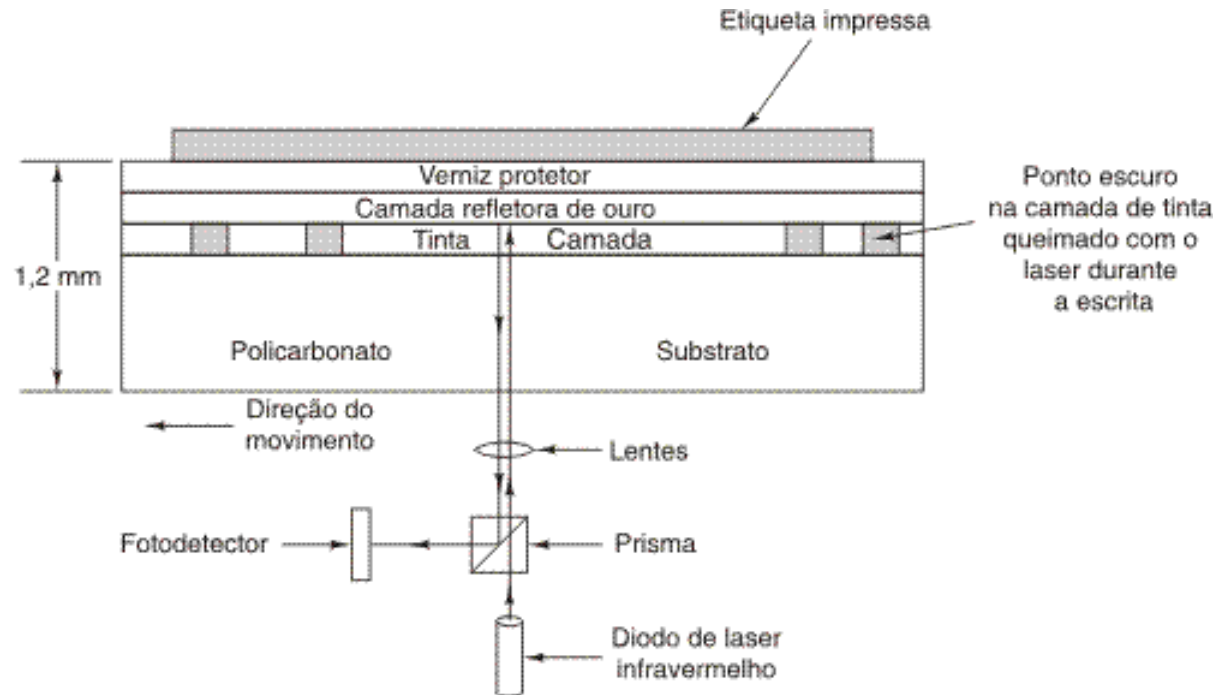
Estrutura de gravação de um CD ou CD-ROM

Hardware do Disco (4)



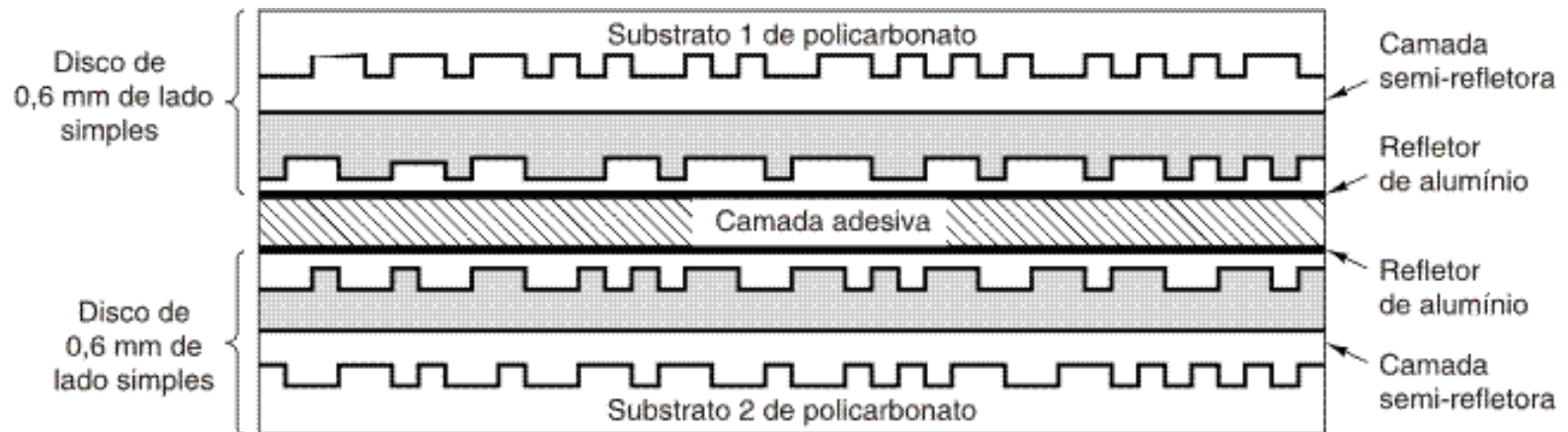
Esquema lógico dos dados em um CD-ROM

Hardware do Disco (5)



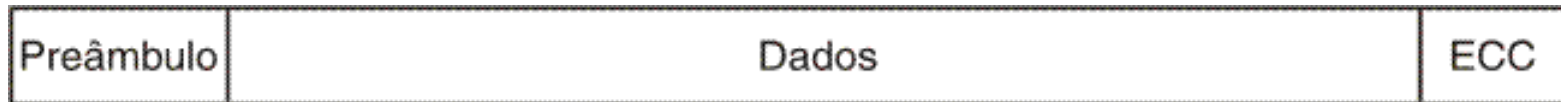
- Secção transversal de um disco CD-R e laser
 - sem escala
- CD-ROM prateado tem estrutura similar
 - sem camada de tinta
 - com camada de alumínio em vez de ouro

Hardware do Disco (6)



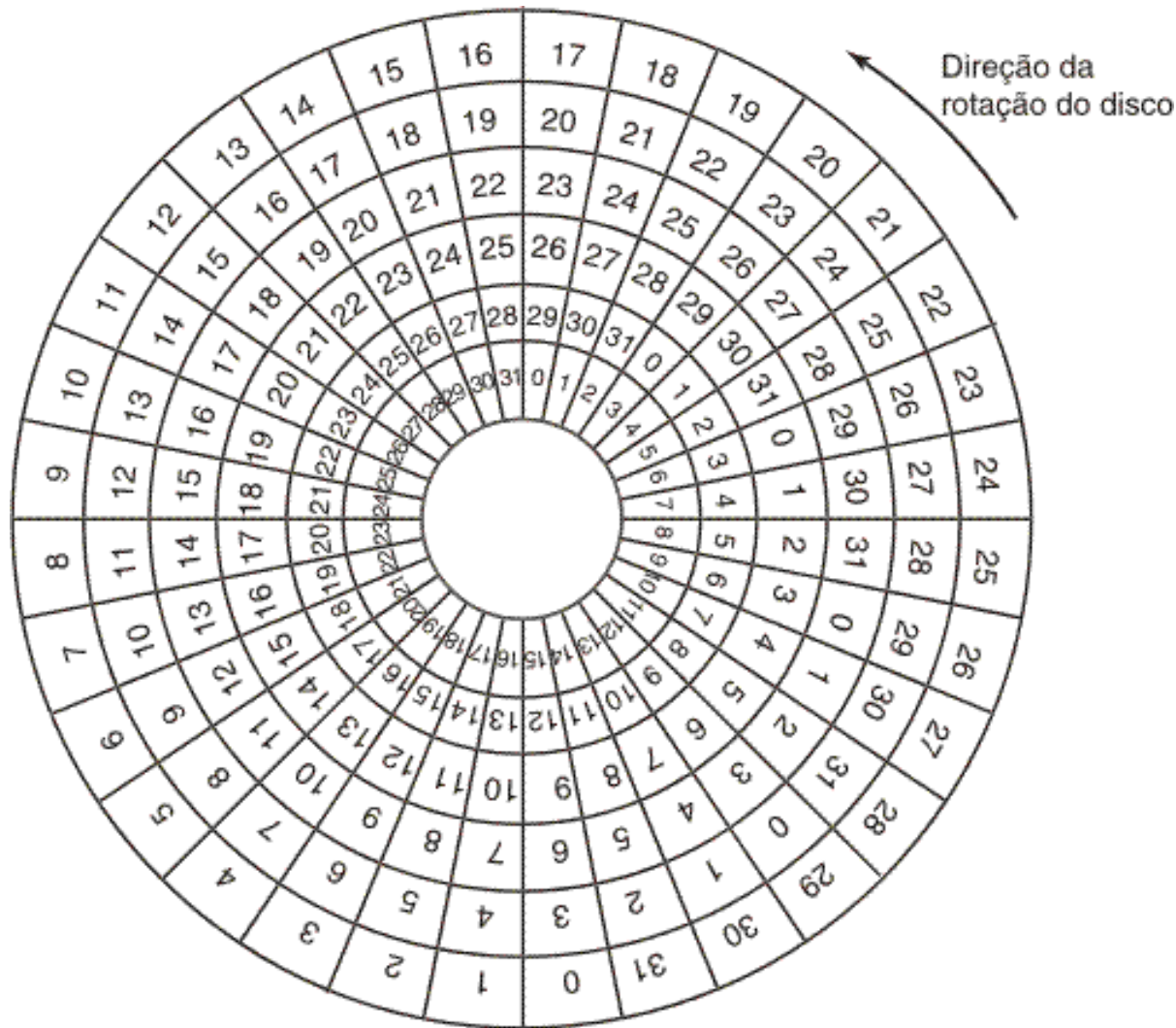
Disco DVD com lado duplo e camada dupla

Formatação de Disco (1)



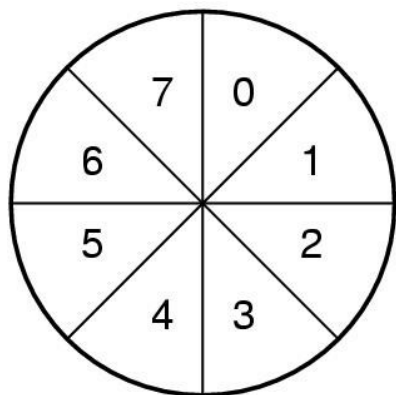
Um setor do disco

Formatação de Disco (2)

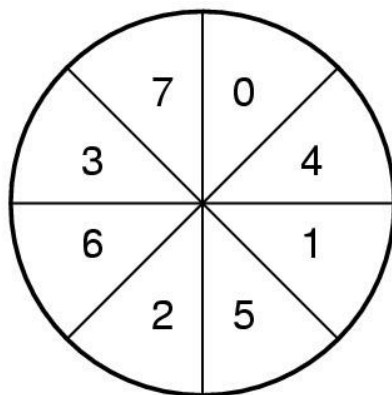


Uma ilustração da torção cilíndrica

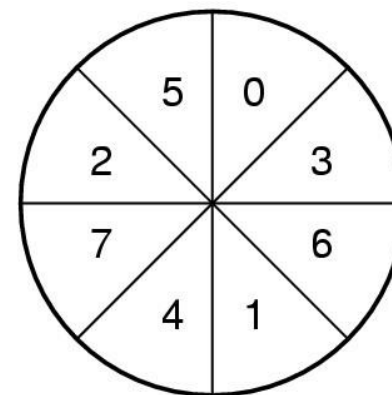
Formatação de Disco (3)



(a)



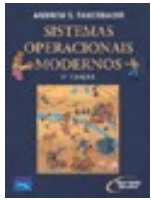
(b)



(c)

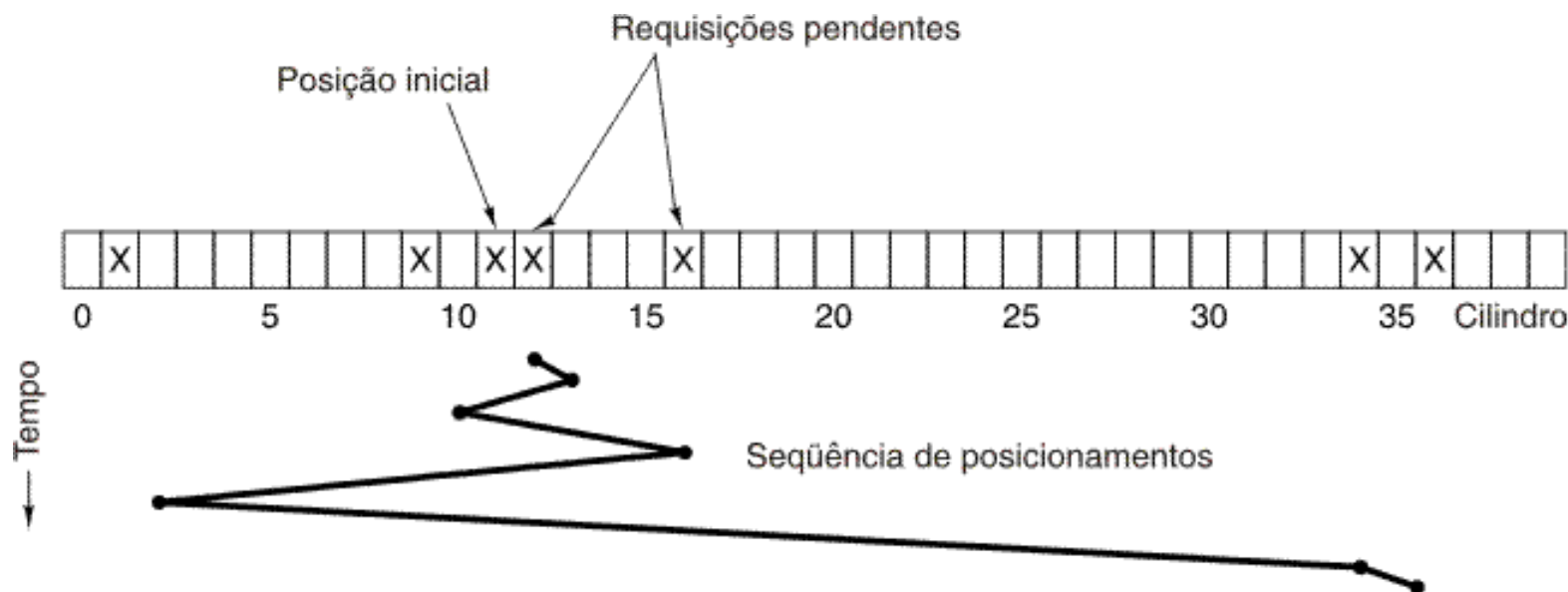
- a) Sem entrelaçamento
- b) Entrelaçamento simples
- c) Entrelaçamento duplo

Algoritmos de Escalonamento de Braço de Disco (1)



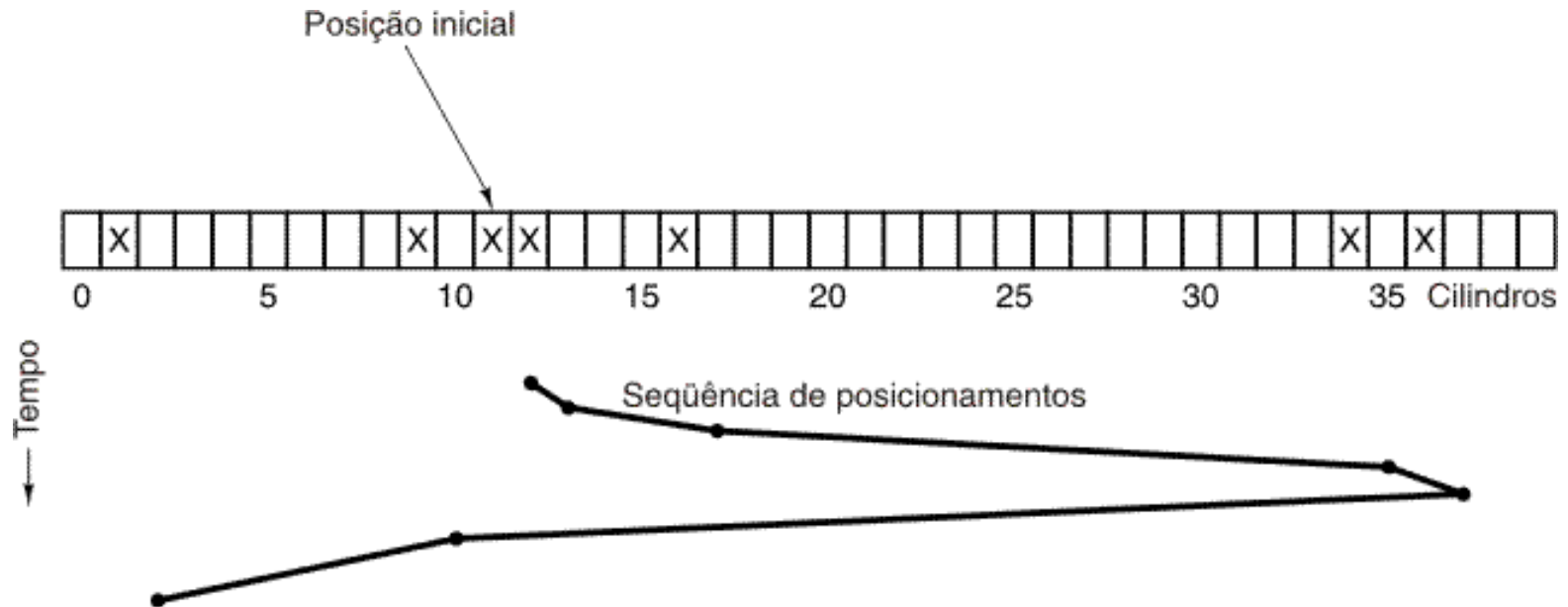
- Tempo necessário para ler ou escrever um bloco de disco é determinado por 3 fatores
 1. tempo de posicionamento
 2. atraso de rotação
 3. tempo de transferência do dado
- Tempo de posicionamento domina
- Checagem de erro é feita por controladores

Algoritmos de Escalonamento de Braço de Disco (2)



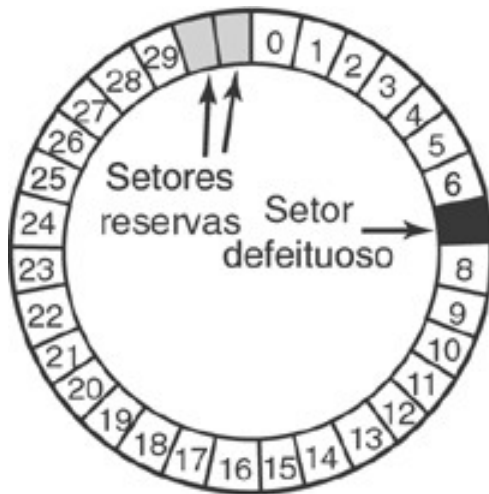
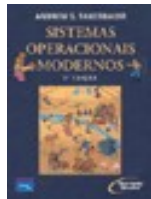
Algoritmo de escalonamento de disco *Posicionamento Mais Curto Primeiro (SSF)*

Algoritmos de Escalonamento de Braço de Disco (3)

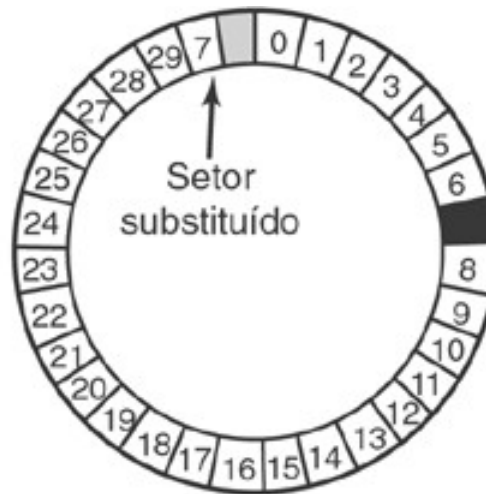


O algoritmo do elevador para o escalonamento das requisições do disco

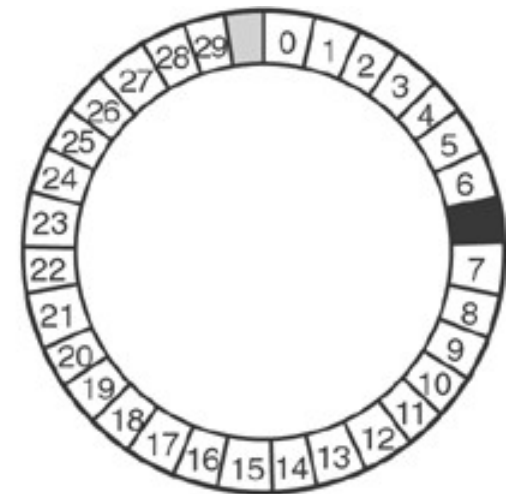
Tratamento de Erro



(a)



(b)

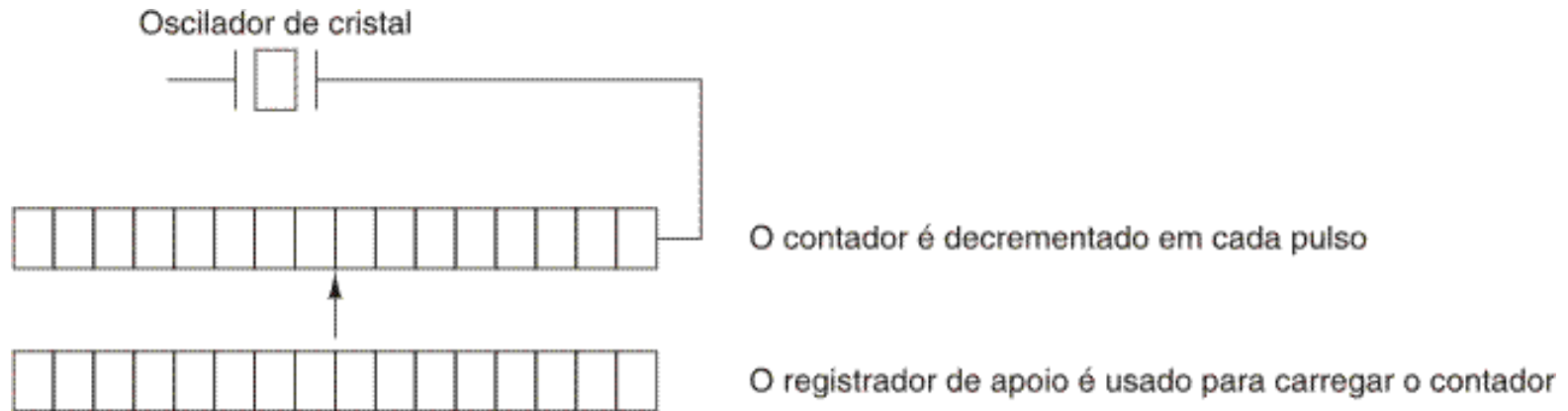


(c)

- a) Uma trilha de disco com um setor defeituoso
- b) Substituindo um setor reserva por um setor defeituoso
- c) Deslocando todos os setores para pular o setor defeituoso

Relógios

Hardware do Relógio



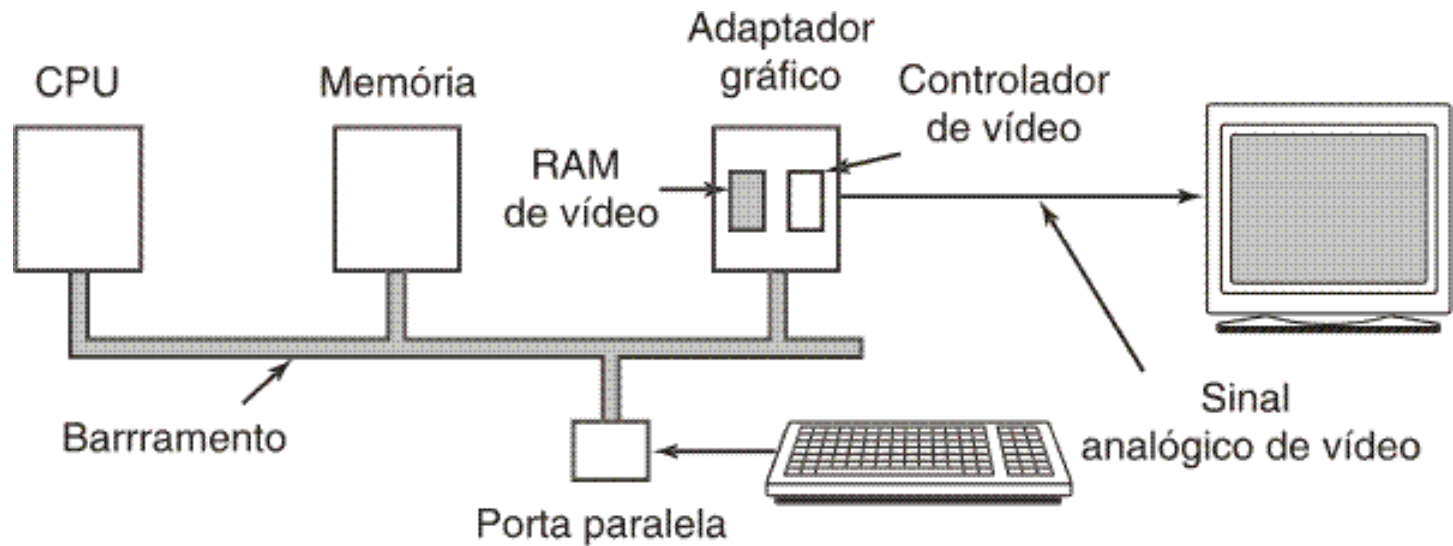
Um relógio programável

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

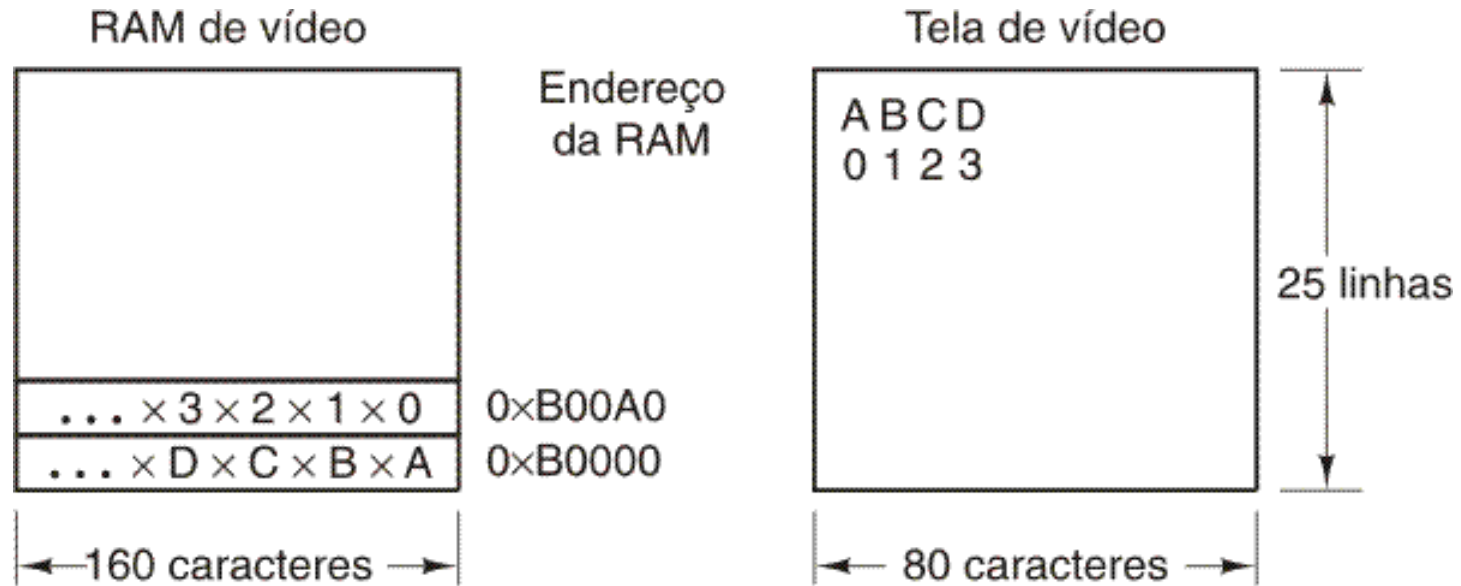
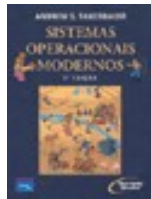
Hardware de Vídeo (1)



Vídeos mapeados na memória

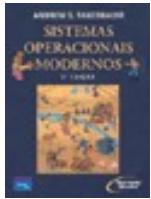
- driver escreve diretamente na RAM de vídeo do monitor

Hardware de Vídeo (2)



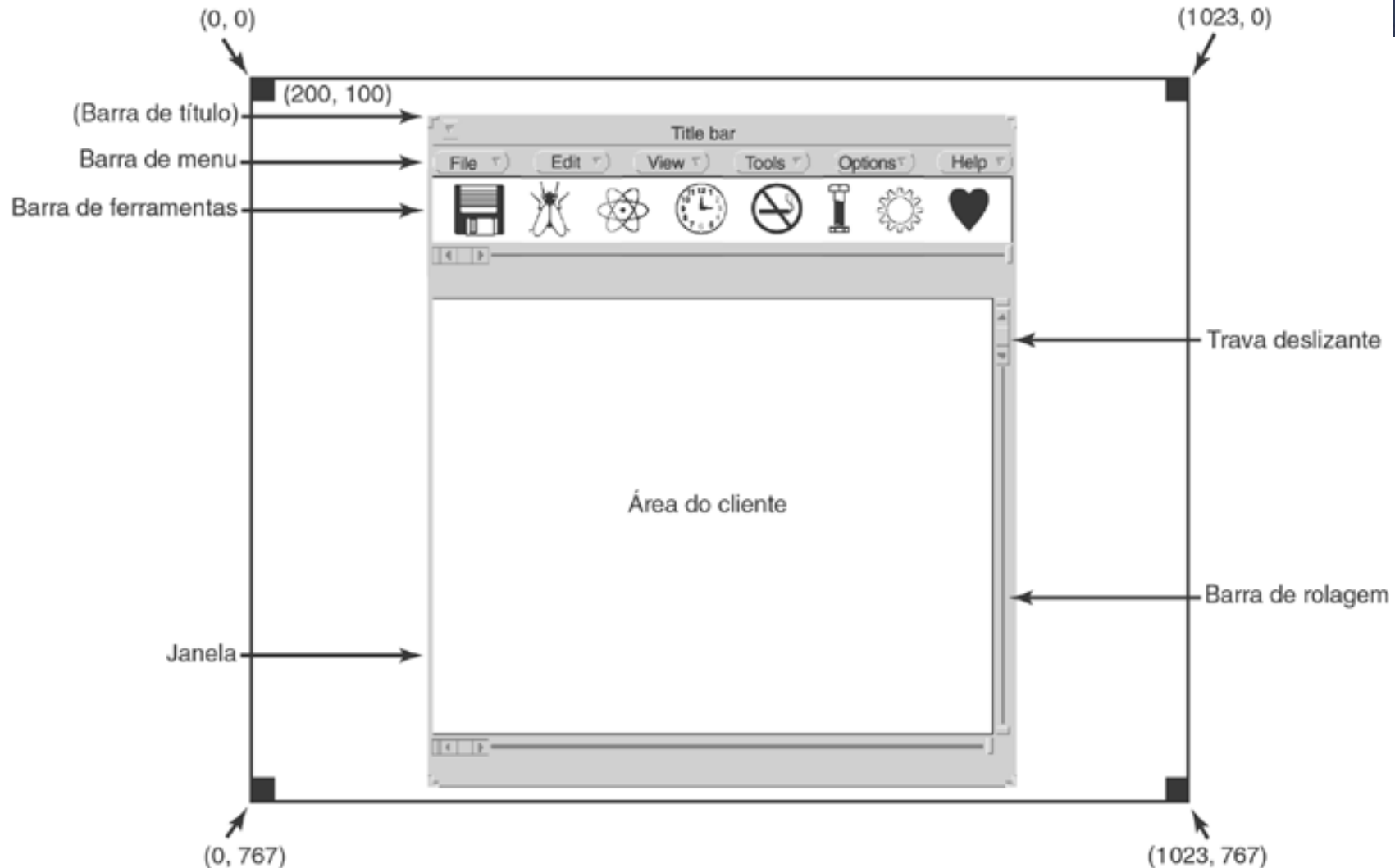
- (a)
- Uma imagem da RAM de vídeo
 - _ tela monocromática simples
 - _ modo caractere
- Tela correspondente
 - _ os x's são bytes de atributos
- (b)

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)

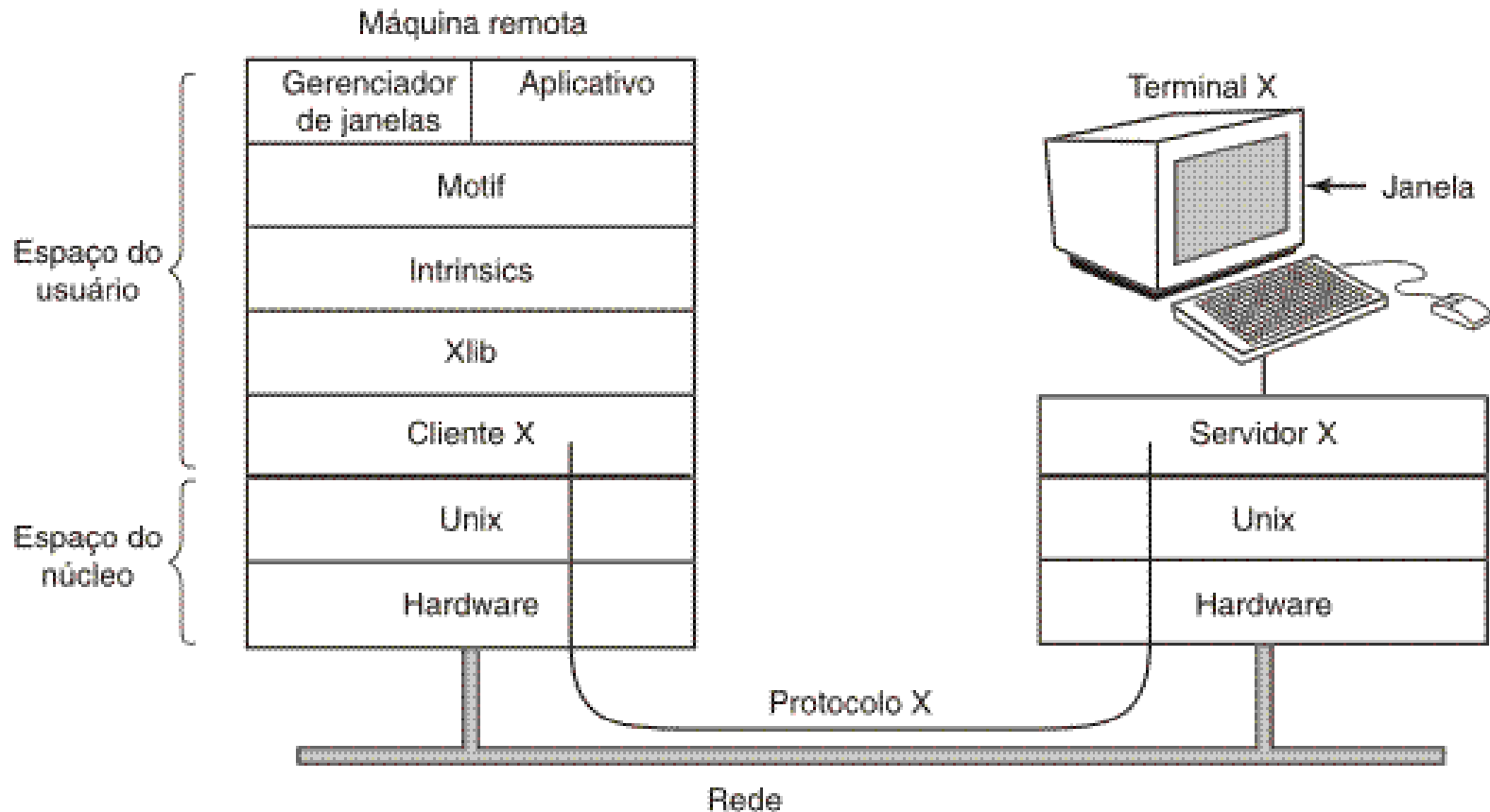


Figura 5-33. Cliente e servidor no Sistema X Window do MIT

Software de Saída para Janelas (3)



Tipos de mensagens entre cliente e servidor:

1. Comandos de desenho do programa para o SO.
2. Respostas do SO para o programa.
3. Eventos do teclado, mouse e outros
4. Mensagens de erro.

Software de Saída para Janelas (4)



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

main(int argc, char *argv[])
{
    Display disp;                /* server identifier */
    Window win;                  /* window identifier */
    GC gc;                       /* graphic context identifier */
    XEvent event;                /* storage for one event */
    int running = 1;

    disp = XOpenDisplay("display_name"); /* connect to the X server */
    win = XCreateSimpleWindow(disp, ... ); /* allocate memory for new window */
    XSetStandardProperties(disp, ...);    /* announces window to window mgr */
    gc = XCreateGC(disp, win, 0, 0);      /* create graphic context */
    XSelectInput(disp, win, ButtonPressMask | KeyPressMask | ExposureMask);
    XMapRaised(disp, win);               /* display window; send Expose event */

    while (running) {
        XNextEvent(disp, &event);        /* get next event */
        switch (event.type) {
            case Expose: break; /* repaint window */
        }
    }
}
```

Figura 5-34. Esqueleto de programa X Window.

Software de Saída para Janelas (5)



```
#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 (6)



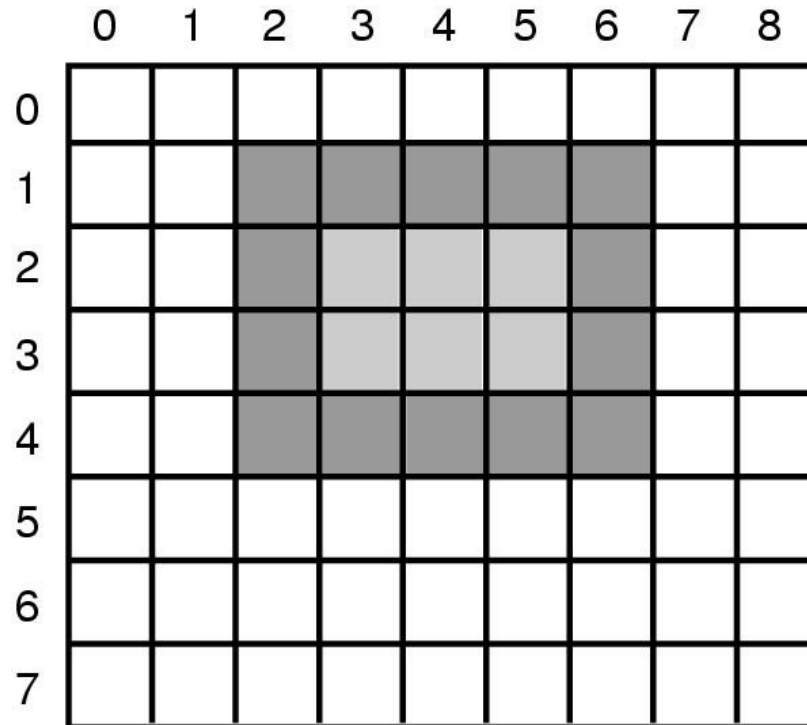
```
while (GetMessage(&msg, NULL, 0, 0)) { /* obtém mensagem da fila */
    TranslateMessage(&msg); /* traduz a mensagem */
    DispatchMessage(&msg); /* envia msg para o procedimento apropriado */
}
return(msg.wParam);
}

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 */
}
```

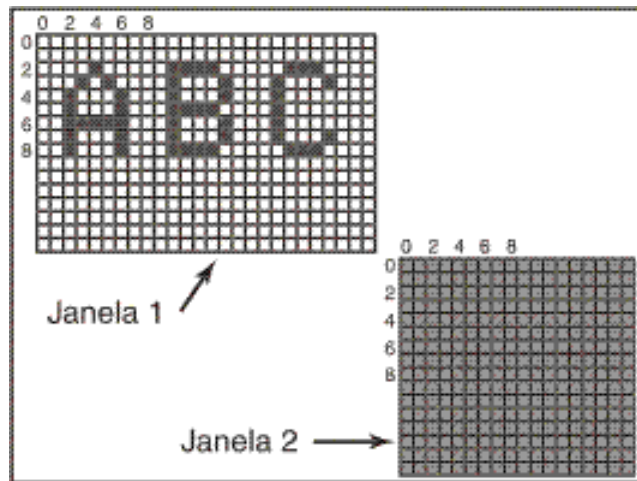
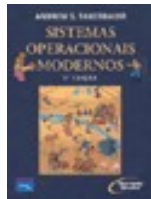
Esqueleto de um programa principal para janelas (parte 2)

Software de Saída para Janelas (7)

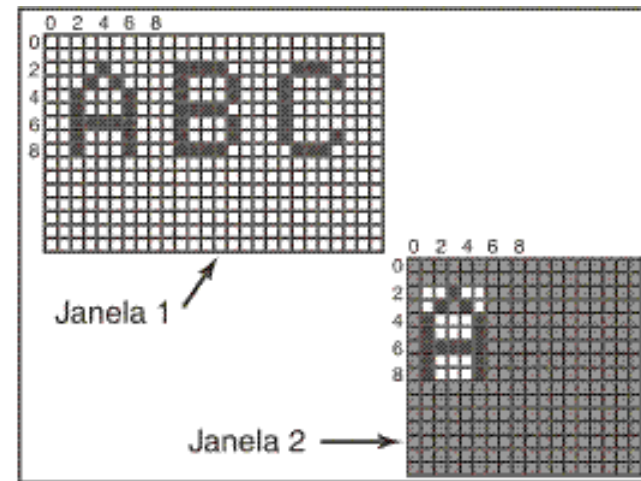


Exemplo de um retângulo desenhado usando
Rectangle

Software de Saída para Janelas (8)



(a)



(b)

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



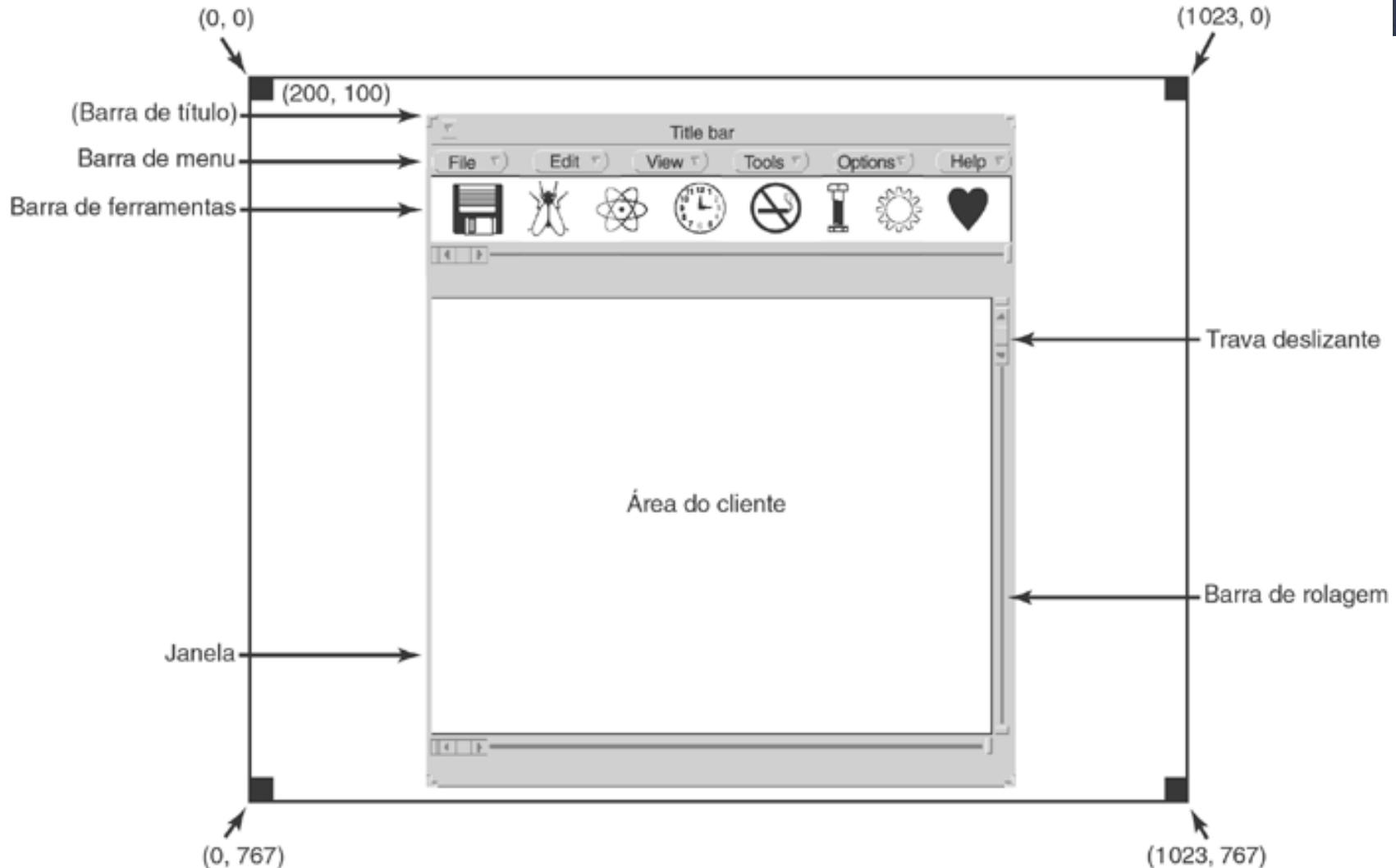
20 pt: abcdefgh

53 pt: abcdefgh

81 pt: abcdefgh

Exemplos de contornos de caracteres de diferentes tamanhos em pontos

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)



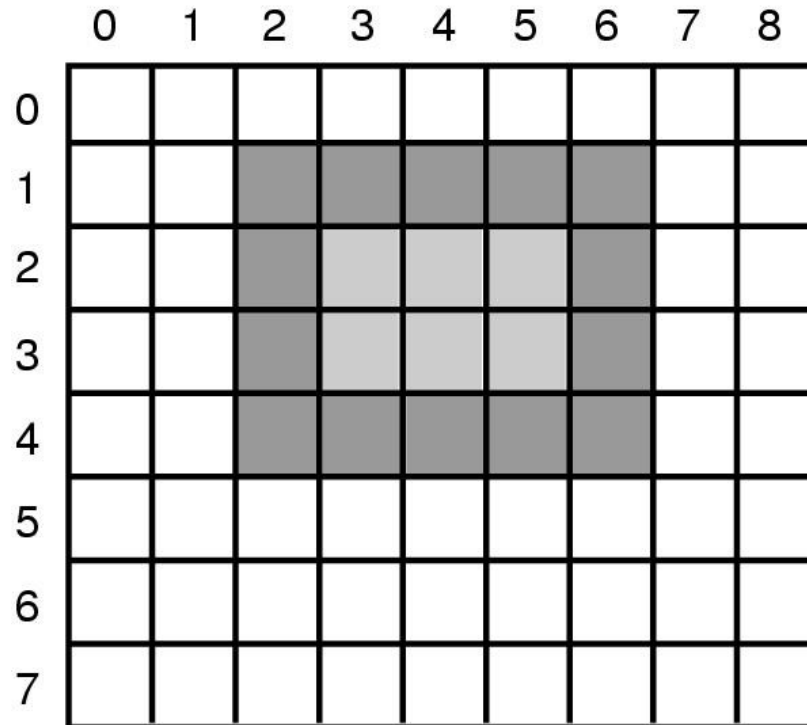
```
while (GetMessage(&msg, NULL, 0, 0)) { /* obtém mensagem da fila */
    TranslateMessage(&msg); /* traduz a mensagem */
    DispatchMessage(&msg); /* envia msg para o procedimento apropriado */
}
return(msg.wParam);
}

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 */
}
```

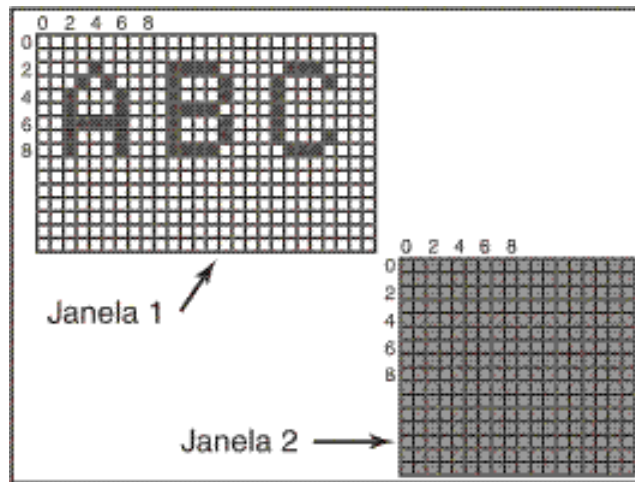
Esqueleto de um programa principal para janelas (parte 2)

Software de Saída para Janelas (4)

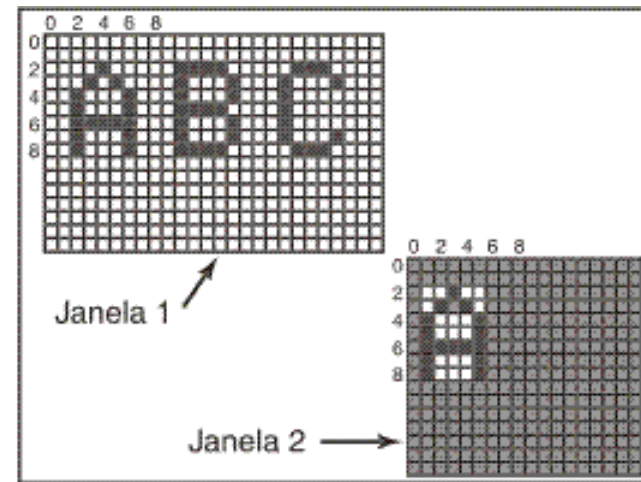


Exemplo de um retângulo desenhado usando
Rectangle

Software de Saída para Janelas (5)



(a)



(b)

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

Software de Saída para Janelas (6)



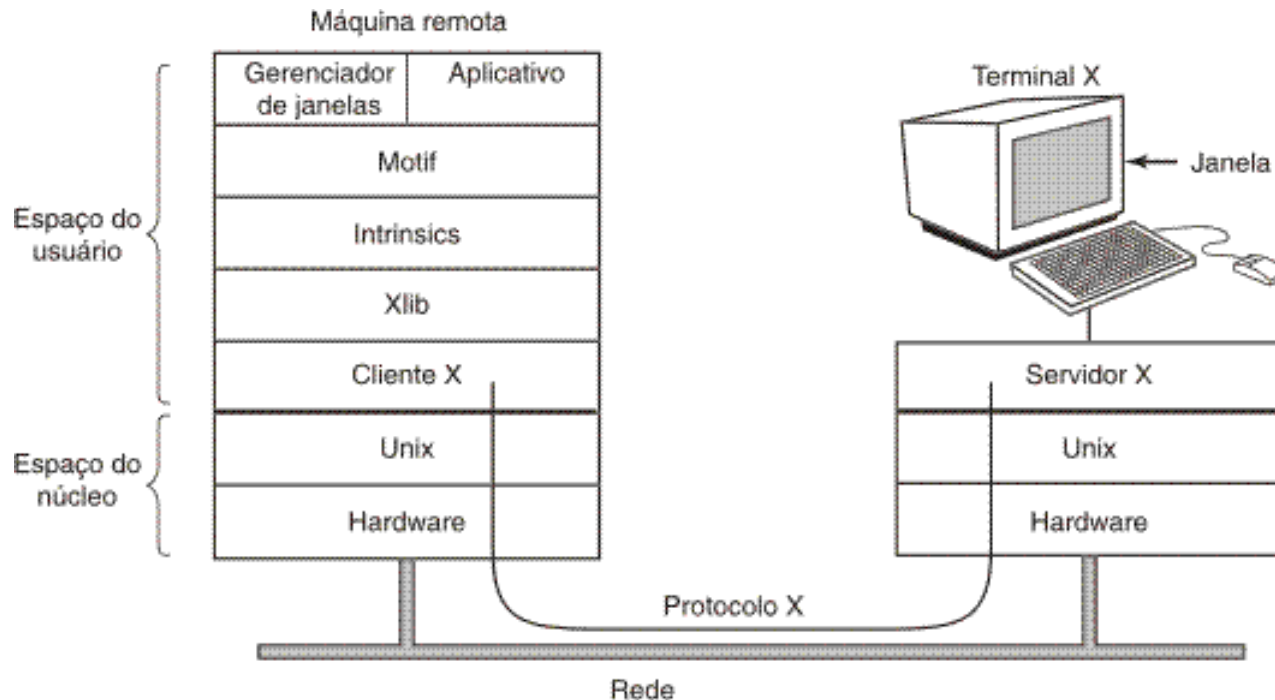
20 pt: abcdefgh

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