Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

WINDOW 2.0

Um Sistema Básico de Gerência de Interface

(Versão Turbo C)

por

Roberto da Silva Bigonha
Lucas Martins do Amaral

Relatório Técnico RT 021/91

Caixa Postal, 702
30.161 - Belo Horizonte - MG
Dezembro de 1991

Abstract

WINDOW 2.0 é um gerenciador de interfaces de vídeo em modo caractere. O sistema é constituído por um conjunto de tipos e funções para construção de janelas e menus, implementado em TURBO C versão 2.0 ou posterior. Este gerenciador permite ao usuário incorporar facilmente janelas e menus ao seu programa no sentido de produzir interfaces homem-máquina de boa qualidade. O sistema possibilita ainda a manutenção de várias janelas, superpostas ou não, e fácil seleção de um ou mais itens de menu.

Abstract

A software package for window and menu handling in TURBO C version 2.0 or higher is presented. The functions made available by this package allow easy and quick design and implementation of windows and menus, which can be used to enhance the quality of man-machine interfaces. The package also provides facilities for maintaining simultaneously several windows on the monitor screen, overlapping windows and for exhibiting and selecting one or more itens from menus.

FUNÇÕES

1 Introdução

WINDOW 2.0 é essencialmente um conjunto de funções para construção de janelas e menus, que foi desenvolvido com o objetivo de ser eficiente e fácil de usar, de forma a facilitar a implementação, em TURBO C [1, 2, 3, 4], de interfaces homem-máquina de boa qualidade, dando ao software do usuário um aspecto mais profissional. Esta implementação é totalmente compatível com o sistema WINDOW 2.0 versão TURBO PASCAL [6].

O sistema para gerência de janelas e menus WINDOW 2.0 é formado pelas bibliotecas WINC*i*.LIB, para $i \in \{\text{T,S,M,C,L,H}\}$, sendo cada uma específica para os modelos de memória *tiny, small, medium, compact, large* e *huge*, respectivamente. As funções exportadas pelas bibliotecas WINC*i*.LIB servem para gerenciar a criação e uso de janelas. Em particular, servem para criar janelas a partir das coordenadas de sua diagonal principal, para colocar molduras em janelas ou definir padrões de cores do texto e do fundo de cada janela. Pode-se também empilhar e desempilhar janelas, passar de uma janela a outra e relocá-las na tela do monitor. Há também funções para construir menus, exibi-los em janelas e deles selecionar ou marcar itens.

2 Importação de Identificadores

A importação das constantes, tipos, variáveis e funções da biblioteca do sistema WINDOW 2.0 deve ser feita através da inclusão do arquivo `WINDOWC.H`. Por exemplo, no início do programa deve-se codificar:

```
#include "WINDOWC.H"
```

3 O Monitor de Vídeo

A tela do monitor deve ter 80 colunas por 25 linhas, com a convenção usual de se ter coordenada (1,1) no canto superior esquerdo da tela e (80,25), no canto inferior direito. Nesta convenção uma coordenada `(x,y)` denota a posição de um caractere na coluna `x` e linha `y`. WINDOW 2.0 funciona com os adaptadores de vídeo CGA, EGA, VGA, HERCULES e MONOCHROME. A identificação do tipo de adaptador presente no hardware e as necessárias inicializações são feitas automaticamente pelo sistema, após execução da função `StartUp()` (vide seção 8).

Dark Colors		Light Colors	
0	BLACK	8	DARKGRAY
1	BLUE	9	LIGHTBLUE
2	GREEN	10	LIGHTGREEN
3	CYAN	11	LIGHTCYAN
4	RED	12	LIGHTRED
5	MAGENTA	13	LIGHTMAGENTA
6	BROWN	14	YELLOW
7	LIGHTGRAY	15	WHITE

Table 1: Código de Cores Válidas

4 As Cores

Para cada janela, o usuário deve especificar suas cores. São seis as cores de uma janela: cor do fundo dos caracteres, do texto, das bordas, do título nas bordas, do fundo de um item marcado no menu e dos caracteres do item marcado. Todas estas cores devem ser especificadas, sendo as cores possíveis dadas pelas constantes (do tipo `int`) pré-definidas no TURBO C de acordo com a tabela 1.

Os caracteres do texto podem ter qualquer uma destas 16 cores; o fundo dos caracteres só pode ser uma das 8 *dark colors*. Caracteres podem ser feitos piscantes pela adição da constante `BLINK`, a qual tem o valor inteiro 128, ao valor da cor.

5 Identificadores Pré-Definidos

5.1 Código das Combinações de Teclas

A tabela 2 apresenta os códigos internos atribuídos por funções de WINDOW 2.0 às combinações de teclas que não estão representadas no conjunto ASCII:

O conjunto padrão ASCII acrescido dos códigos definidos para as combinações de teclas da tabela 2 será referido neste texto por **WASCII**.

Código	Significado
130	Shift-Tab
131 a 140	Alt-Q/W/E/R/T/Y/U/I/O/P
145 a 153	Alt-A/S/D/F/G/H/I/J/K/L
159 a 165	Alt-Z/X/C/V/B/N/M
174 a 183	F1 a F10
186	Home
187	Up Arrow
188	PgUp
190	Left Arrow
192	Right Arrow
194	End
195	Down Arrow
196	PgDn
197	Ins
198	Del
199 a 208	Shift-F1 a Shift-F10
209 a 218	Ctrl-F1 a Ctrl-F10
219 a 228	Alt-F1 a Alt-F10
229	Ctrl-PrtSc
230	Ctrl-Left Arrow
231	Ctrl-Right Arrow
232	Ctrl-End
233	Ctrl-PgDn
234	Ctrl-Home
235 a 246	Alt-1/2/3/4/5/6/7/8/9/0/-/=
247	Ctrl-PgUp
248	F11
249	F12
250	Shift-F11
251	Shift-F12
252	Ctrl-F11
253	Ctrl-F12
254	Alt-F11
255	Alt-F12

Table 2: Teclas Especiais

5.2 Constantes

Os seguintes identificadores denotam constantes pré-definidas pelo sistema :

- `VERSION = '2.0'`
  Versão corrente do WINDOW 2.0.
- `SERIALNUMBER` = xxxx
  Número de série da cópia de WINDOW 2.0.
- `MAXWINDOWS = 64`
  Número máximo de janelas que podem ser construídas simultaneamente. Toda janela tem um número de identificação no intervalo [0,`MAXWINDOWS`]. O número 0 é reservado pelo sistema para designar a janela inicial, que corresponde a toda tela do terminal de vídeo do microcomputador. O usuário pode referenciar a janela de número 0; não pode, contudo, construir uma com este número.
- `MAXBLOCKS = 40`
  Número máximo de blocos de um menu. Um bloco de menu é uma lista de itens, contendo um item por linha.
- `MAXLINES = 25`
  Número máximo de linhas por bloco de menu.

5.3 Teclas Especiais

Os seguintes identificadores denotam o código interno atribuído à teclas frequentemente usadas para representar ações especiais: `ESC, PGUP, PGDN, INS, DEL, PLUS, MINUS, UPARROW, DOWNARROW, LEFTARROW, RIGHTARROW, HOME, ENDKEY`

6 Tipos Pré-Definidos

Os seguintes tipos, que estão definidos na interface do sistema WINDOW 2.0, são de interesse do usuário:

- `TMENU`
  Os menus do usuário devem necessariamente ser declarados do tipo `TMENU`. Este tipo descreve as cores do marcador de itens no menu, o número de blocos do menu, as coordenadas de cada bloco dentro da janela e o detalhe de cada item. Estruturas do tipo `TMENU` devem ser devidamente iniciadas pelos funções `DefineMenu, DefineBlock` e `SetItem` (Vide Seção 11.1).

ErrorCond					Explicação do Erro
0000	0000	0000	0000	0	Nenhum erro
0000	0000	0000	0001	1	Overflow da pilha de janelas
0000	0000	0000	0010	2	Overflow da pilha de telas
0000	0000	0000	0100	4	Janela anterior não existe
0000	0000	0000	1000	8	Número da janela inválido
0000	0000	0001	0000	16	Placa de vídeo irreconhecível
0000	0000	0010	0000	32	Linha de menu fora de range
0000	0000	0100	0000	64	Bloco de menu fora de range
0000	0000	1000	0000	128	Coluna de menu fora de range
0000	0001	0000	0000	256	Coordenadas fora de range
0000	0010	0000	0000	512	Menu indefinido
0000	0100	0000	0000	1024	Desempilhou demais
0000	1000	0000	0000	2048	Coordenadas fora de janela
0001	0000	0000	0000	4096	Menu não inicializado
0010	0000	0000	0000	8192	Overflow do heap
0100	0000	0000	0000	16384	Constante de cor inválida
1000	0000	0000	0000	32768	Excesso de janelas (>`MAXWINDOWS`)

Table 3: Condições de Erro

- `TCOLORS`
  Tipo pré-definido com campos para as 6 cores de uma janela, conforme a seguinte definição:

```
typedef struct {
    int windowback, windowtext;
    int markback, marktext;
    int frame, title;
} TCOLORS;
```

7 Condições de Erro

As condições de erro detectadas pelo sistema WINDOW 2.0 são anotadas na variável pré-definida `ErrorCond`, que é do tipo `unsigned int`. Cada um dos 16 bits desta variável é usado para indicar um tipo de erro, sendo, portanto, possível identificar até 16 erros de cada vez, conforme é mostrado na tabela 3.

O usuário pode consultar ou alterar o valor de ErrorCond livremente. Por exemplo, o seguinte trecho de programa ilustra a apuração e listagem de todas as condições de erro detectadas pelo sistema durante uma dada execução:

```
if (ErrorCond != 0) {
    if ((ErrorCond && 1) != 0)
       cputs("Overflow da pilha.\n");
    if ((ErrorCond && 2) != 0)
       cputs("Falta espaco para salva tela.\n");
    if ((ErrorCond && 4) != 0)
       cputs("Tentativa de restaurar tela nao salva.\n");
    if ((ErrorCond && 8) != 0)
       cputs("Numero de identificacao de janela invalido.\n");
    if ((ErrorCond && 16) != 0)
       cputs("Sistema WINDOW 2.0 nao foi inicializado.\n");
    if ((ErrorCond && 32) != 0)
       cputs("Linha do menu fora de range.\n");
    if ((ErrorCond && 64) != 0)
       cputs("Bloco do menu fora de range.\n");
    if ((ErrorCond && 128) != 0)
       cputs("Coluna do menu fora de range.\n");
    if ((ErrorCond && 256) != 0)
       cputs("Coordenadas da janela fora de range.\n");
    if ((ErrorCond && 512) != 0)
       cputs("Tentativa de acesso a menu indefinido.\n");
    if ((ErrorCond && 1024) != 0)
       cputs("Tentativa de deixar janela nao empilhada.\n");
    if ((ErrorCond && 2048) != 0)
       cputs("Coordenadas fora da janela ativa.\n");
    if ((ErrorCond && 4096) != 0)
       cputs("Estrutura de menu nao iniciada.\n");
    if ((ErrorCond && 8192) != 0)
       cputs("Overflow do heap.\n");
    if ((ErrorCond && 16384) != 0)
       cputs("Cor invalida.\n");
    if ((ErrorCond && 32768) != 0)
       cputs("Numero de janelas acima de 64");
     Errorcond = 0;             /* Apaga condicao de erro */
}
else cputs("Tudo se passou na mais perfeita ordem.");
```

8 Inicialização do Sistema

No início de programas que usam funções da biblioteca do WINDOW 2.0, na versão TURBO C, é obrigatória a chamada da função `StartUp()`, para inicializar o sistema de

janelas e menu. A não-ativação desta função caracteriza uma situação de erro, podendo acarretar resultados imprevisíveis.

- `void StartUp(void)`

  Função: Inicializa o sistema de janelas e menus.

9 Entrada de Dados

O teclado do microcomputador da família PC permite a entrada dos caracteres definidos no conjunto WASCII. A função `getch()` do Turbo C devolve para cada tecla pressionada o seu código ASCII e para as combinações de teclas, o código 0 (NULL) seguido de um byte que identifica cada seqüência. Esta função atende satisfatoriamente à maior parte das aplicações; entretanto as funções apresentados a seguir uniformizam e facilitam o tratamento dos caracteres obtidos a partir do teclado.

- `void ReadAnyKey(char *key)`

  Parâmetro :

  - `key` = código WASCII da(s) tecla(s) pressionada(s).

  Função : Lê, sem ecoar,a última tecla ou par de teclas pressionadas e retorna o código WASCII correspondente (vide seção 5.3).

- `void ReadAllowedKey(char *key, char *allowedkeys)`

  Parâmetro :

  - `key` = código WASCII da tecla;
  - `allowedkeys` = conjunto a que `key` deve pertencer.

  Função : Lê, sem ecoar, e retorna, em `key`, o código da última tecla ou seqüência de teclas pressionadas e cujo código WASCII esteja no conjunto dado por `allowedkeys`. Teclas cujos códigos estejam fora do conjunto `allowedkeys` são ignoradas.

10 Janelas

As funções descritas a seguir oferecem facilidades para construir, ativar e relocar janelas na tela do monitor de vídeo. Uma janela é entendida por uma região inteiramente contida na tela.

10.1 Construção de Janelas

Para se construir uma janela (sem moldura) usa-se a função `MakeWindow`. A moldura pode ser desenhada posteriormente via função `DrawFrame`. A moldura não faz parte da área útil da janela, de forma a ficar protegida contra *scrolling* e operações de escrita na tela. O usuário pode, contudo, através de funções especiais, escrever sobre as molduras.

As seguintes funções servem para construir janelas :

- `void NewWindowNumber(int *wnumber)`

  Parâmetro :

  - `wnumber` = número de janela.

  Função : Obtém o número de identificação de uma janela disponível. Retorna 0 se todas as `MAXWINDOWS` janelas do sistema estiverem ocupadas.

- `void FreeWindowNumber(int wnumber)`

  Parâmetro :

  - `wnumber` = número de janela.

  Função : Devolve ao espaço de números de janelas disponíveis o número de identificação de uma janela que não seja mais necessária.

- ```
  void MakeWindow(
            int  wnumber,
            int  x0,
            int  y0,
            int  x1,
            int  y1,
            int  ColorofTheBack,
            int  ColorofText)
  ```

  Parâmetros :

  - `wnumber` = número de identificação da janela.
  - `x0` = coluna esquerda da janela.
  - `y0` = primeira linha da janela.
  - `x1` = coluna direita da janela.
  - `y1` = última linha da janela.
  ```

- `ColorOfTheBack` = código da cor do fundo dos caracteres da janela.
- `ColorOfText` = código da cor do texto da janela.

Função : Constrói uma janela cuja diagonal principal tem as coordenadas de tela `(x0,y0)` e `(x1,y1)`. Note que :

1. Esta função não altera o conteúdo da tela e nem ativa a janela por ela criada, apenas cria uma estrutura de dados com informações necessárias ao uso da janela. Pode-se dizer que `MakeWindow` cria uma janela invisível, que poderá ser ativada, por exemplo, pela função `GoToWindow`.
2. Na primeira vez em que se entrar na janela criada, o cursor ficará na posição (1,1) relativa à janela. Nas demais, o cursor sempre retorna à posição em que estava no momento em que a janela foi abandonada.
3. As coordenadas `(x0,y0)` e `(x1,y1)` são absolutas, referindo-se a toda a tela de 25 linhas por 80 colunas. As únicas funções de WINDOW 2.0 que recebem coordenadas absolutas como parâmetros são `MakeWindow`, `ChangeAttribute` e `RelocateWindow`. Em todos os demais, as coordenadas são relativas à janela que estiver correntemente ativa.
4. A menor janela que se pode construir deve ter pelo menos 1 (uma) linha e 1 (uma) coluna de área útil, excluindo a moldura.

- ```
  void ChangeAttribute(
            int   x,
            int   y,
            char  attribute)
  ```

  Parâmetros :

  - `x` = coluna da tela;
  - `y` = linha da tela;
  - `attribute` = caractere com o atributo a ser armazenado nas coordenadas absolutas `(x,y)` da tela.

  Função : Lê o caractere e seu atributo das coordenadas especificadas da janela ativa e o armazena de volta com o atributo dado.

- `void DrawFrame(int wnumber, int ColorOfFrame)`

  Parâmetros :

  - `wnumber` = número de uma janela construída pela função `MakeWindow`.
  - `ColorOfFrame` = código de cor da moldura da janela `wnumber`.

  Função : Desenha a moldura da janela `wnumber` na cor especificada. Observe que:
  ```

1. A moldura ocupa um total de duas linhas e duas colunas da janela. Estas linhas e colunas não são consideradas partes da janela, de forma a serem preservadas contra *scrolling* da janela.
2. Por default, molduras são desenhadas com caracteres que formam duas linhas paralelas.

- `void ChangeFrameChars(`
  ```
  char  horizontal,
  char  vertical,
  char  upperleft,
  char  lowerleft,
  char  upperright,
  char  lowerright)
  ```

  Parâmetros :

  - `horizontal` = código do caractere usado para fazer as linhas horizontais das bordas de janela.
  - `vertical` = código do caractere usado para fazer as linhas verticais das bordas de janela.
  - `upperleft` = código do caractere usado para fazer o canto superior esquerdo das bordas de janela.
  - `lowerleft` = código do caractere usado para fazer o canto inferior esquerdo das bordas de janela.
  - `upperright` = código do caractere usado para fazer o canto superior direito das bordas de janela.
  - `lowerright` = código do caractere usado para fazer o canto inferior direito.

  Função : Redefine os caracteres usados para desenhar molduras de janelas. Cada redefinição permanece em vigor até nova chamada a `ChangeFrameChars`.

- `void PutTopTitle(`
  ```
  int   wnumber,
  int   column,
  int   TitleColor,
  char  *title)
  ```

  Parâmetros :

  - `wnumber` = número de identificação da janela.
  - `column` = coluna inicial do título relativa à janela.
  - `TitleColor` = código de cor do título.
  - `title` = título a ser escrito na borda superior da janela.

Função : Escreve o título na borda superior da janela a partir da coluna `column`.

- ```
  void PutBottomTitle(
            int   wnumber,
            int   column,
            int   TitleColor,
            char  *title)
  ```

  Parâmetros :

  - `wnumber` = número de identificação da janela.
  - `column` = coluna do título relativa à janela.
  - `TitleColor` = código de cor do título.
  - `title` = título a ser escrito na borda inferior da janela.

  Função : Escreve o título na borda inferior da janela a partir da coluna `column`.

- `void ResetColors(void)`

  Função : Restaura as cores originais de fundo e de texto da janela correntemente ativa, isto é, restabelece as cores definidas na `MakeWindow` da janela.

10.2 Ativação de Janelas

Inicialmente a janela ativa é a de número 0, que corresponde a toda a tela do monitor de vídeo. As janelas do usuário só podem receber números de identificação no intervalo [1,`MAXWINDOW`]. Após criar janelas através da função `MakeWindow`, o usuário pode passar de uma janela a outra pela ativação da função `GoToWindow`.

- `void GoToWindow(int wnumber)`

  Parâmetro :

  - `wnumber` = número de identificação da janela a ser ativada.

  Função : Entra na janela especificada, posicionando o cursor no ponto em que ele estava na última vez que a janela foi visitada. Se for a primeira vez, o cursor vai para a coordenada (1,1) da janela.

- `int ActiveWindow(void)`

  Função : Retorna o número da janela correntemente ativa.
  ```

10.3 Empilhamento e Movimentação de Janelas

Quando se deseja sobrepor janelas, a área da tela que será ocupada pela nova janela deve ser salva de forma a permitir a restauração da tela quando esta janela não for mais necessária. As operações de salvamento e restauração de janelas são efetuadas através das funções `EnterWindow` e `LeaveWindow`, e a movimentação de janelas, via as funções `RelocateWindow` e `DragWindow`.

- `void EnterWindow(int wnumber)`

  Função : Empilha o número da janela correntemente ativa, ativa a janela `wnumber` e salva, numa pilha, a região da tela delimitada por esta janela. Normalmente esta função deve ser chamada antes de se escrever qualquer caractere na janela `wnumber`, de forma a assegurar posterior restauração da região da tela ocupada pela janela.

- `void LeaveWindow(void)`

  Função : Restaura a região que a janela correntemente ativa encobre na tela; a seguir, abandona esta janela, retornando àquela cujo número foi empilhado pela última chamada a `EnterWindow`.

- `void RelocateWindow(int wnumber, int x, int y)`

  Parâmetros :

  - `wnumber` = número da janela que será relocada.
  - `x,y` = novas coordenadas do canto superior esquerdo da janela `wnumber`.

  Função : Reloca a janela `wnumber` para uma nova posição na qual seu canto superior esquerdo passa a ter as coordenadas absolutas `(x,y)`.

- `void DragWindow(int wnumber)`

  Parâmetro :

  - `wnumber` = número de identificação da última janela empilhada.

  Função : Permite que janela `wnumber` seja movimentada na tela atuando-se nas setas do teclado. A função retorna com o acionamento da tecla ESC.

Para que as funções `RelocateWindow` e `DragWindow` produzam os efeitos desejados, a área da tela encoberta pela janela a ser relocada deve necessariamente ter sido empilhada, através da função `EnterWindow`, e a janela não pode estar encoberta, mesmo que

parcialmente, por alguma outra. Tentativas de relocar janelas que não foram empilhadas são ignoradas.

10.4 Outras Operações sobre Janelas

Somente as seguintes funções do TURBO C podem ser usadas livremente pelo usuário de WINDOW 2.0. Estas funções têm efeito apenas na janela que estiver ativa:

`cputs, cgets, cprintf, gotoxy, clrscr, insline, clreol, delline, textcolor, textbackground, wherex e wherey.`

As funções `printf, puts` e `gets` não devem ser usadas porque elas não respeitam os limites de janelas. A função `window` do TURBO C, para criar janelas, também não deve ser utilizada pelo usuário sob o risco de comprometer a robustez do sistema.

11 Menus

Menus são constituídos de blocos, sendo cada bloco formado por linhas contendo as opções ou itens a serem selecionados. Cada janela pode ter um menu associado, e um mesmo menu pode ser associado a várias janelas. Um item ou opção de menu pode possuir uma marca, que pode ser feita trafegar de um item a outro pelo acionamento das setas do teclado ou das teclas Home e End.

11.1 Construção de Menus

Do ponto de vista de WINDOW 2.0, um menu é uma estrutura de dados do tipo `TMENU`, que deve ser declarada e explicitamente inicializada pelo usuário através das funções `DefineMenu, DefineBlock` e `SetItem`, que são definidas a seguir:

- ```
  void DefineMenu(
            TMENU  *menu,
            int    numofblocks,
            int    backofmark,
            int    foreofmark)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  ```

- `numofblocks` = número de blocos do menu. Um bloco é uma coluna de itens ou opções.
- `backofmark` = Código de cor de *background* do item marcado.
- `foreofmark` = Código de cor do texto do item marcado.

Função : Inicializa a estrutura do menu, atribuindo valores a seus campos. Esta função deve, obrigatoriamente, ser chamada uma vez para cada menu declarado pelo usuário.

- ```
  void DefineBlock(
            TMENU  *menu,
            int    block,
            int    x,
            int    y,
            int    numberofrows)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  - `block` = número do bloco considerado do menu. Os blocos são numerados a partir de 1.
  - `x` = coluna inicial do bloco relativa à janela que conterá o menu.
  - `y` = linha inicial do bloco relativa à janela que conterá o menu.
  - `numberofrows` = número máximo de linhas no bloco.

  Função : Define um bloco de menu especificado, alocando as estruturas de dados necessárias para descrever seus itens.

- ```
  void SetItem(
            TMENU  *menu,
            int    block,
            int    line,
            char   itemcode,
            char   *itemtext)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  - `block` = número do bloco do menu que contém o item a ser inicializado.
  - `line` = linha do item do menu a ser inicializado.
  - `itemcode` = código da ação a ser executada quando a linha especificada for selecionada.

– `itemtext` = texto da linha do menu. Deve ser um string de até 80 caracteres.

Função : Define um item de menu, fornecendo o seu texto e o código que será retornado pela `GetOption` quando o item for selecionado.

- `void ReleaseBlocks(TMENU *menu)`

  Função : Reinicializa o menu, devolvendo seus blocos ao *heap* do sistema. Novos blocos poderão ser alocados via função `DefineBlock`.

A descrição de um menu pode ser alterada através das seguintes funções :

- ```
  void ChangeMenuInfo(
            TMENU  *menu,
            int    numofblocks,
            int    backofmark,
            int    foreofmark)
  ```

  Parâmetros :

  – `menu` = descritor do menu.

  – `numofblocks` = número de blocos no menu. Deve ser necessariamente menor ou igual ao `numofblocks` definido pela função `DefineMenu`.

  – `backofmark` = código de cor *background* da marca de item.

  – `foreofmark` = código de cor do texto da marca de item.

  Função : Atualiza os campos do menu cujos valores são passados como parâmetros. Os demais campos não são alterados.

- ```
  void ChangeBlockInfo(
            TMENU  *menu,
            int    block,
            int    x,
            int    y,
            int    numberofrows)
  ```

  Parâmetros :

  – `menu` = descritor do menu.

  – `block` = número do bloco do menu. Os blocos são numerados a partir de 1.

  – `x` = coluna inicial do bloco.

  – `y` = linha inicial do bloco.

- `numberofrows` = número de itens no bloco. Deve ser menor ou igual ao número máximo de itens estabelecidos por `DefineBlock` para o bloco.

Função : Altera as coordenadas e o tamanho de um bloco de menu.

Devido ao uso de apontadores nos descritores de menus, a cópia de um menu não pode ser feita via comando de atribuição. Cópias de menus devem obrigatóriamente ser feitas pela função `Copymenu`, que tem a seguinte definição:

- `void CopyMenu(TMENU sourcemenu, TMENU *destmenu)`

  Parâmetros :

  - `sourcemenu` = descritor do menu.
  - `destmenu` = descritor de menu.

  Função : Copia menu `sourcemenu` para `destmenu`.

11.2 Exibição de Menus e Seleção de Itens

Os menus podem ser exibidos em qualquer janela com o objetivo de selecionar opções. As funções descritas a seguir são encarregadas das funções de exibição de menus e seleção de itens.

- `void DisplayMenu(int wnumber, TMENU *menu)`

  Parâmetros :

  - `wnumber` = número de identificação da janela na qual o menu será exibido.
  - `menu` = descritor do menu.

  Função : Exibe na janela especificada o menu passado como parâmetro e o vincula à esta janela.

- ```
  void GetOption(
           int   wnumber,
           char  *EscapeSet,
           char  *option)
  ```

  Parâmetros :

  - `wnumber` = número de identificação da janela da qual será selecionada uma opção do menu.
  ```

- `EscapeSet` = conjunto de códigos das teclas que, além de **ENTER**, causam o retorno de `GetOption`. Deve ser um subconjunto do conjunto **WASCII**.
- `option` = valor a ser retornado indicando a opção selecionada (Vide função `SetItem`) ou o código da tecla em `EscapeSet` que foi pressionada.

Função : Entra na janela especificada, marca um item do menu e permite ao usuário mover a marca para qualquer outro item do menu, através das setas e das teclas **HOME** e **END** presentes no teclado do PC. A tecla **ENTER** seleciona o item marcado, retornando no parâmetro `option` o código associado à linha selecionada (Vide função `SetItem`). Teclas cujos códigos estiverem em `EscapeSet` causam o retorno com `option` indicando o seu código. Inicialmente o item marcado é o do bloco 1 linha 1. A partir de então, `GetOption` sempre se inicia marcando o último item que foi selecionado na chamada anterior.

- ```
  void GetCursorPosition(
            int  wnumber,
            int  *block,
            int  *line)
  ```

  Parâmetros :

  - `wnumber` = identificação da janela.
  - `block` = número do bloco de menu que contém o item que foi marcado por último.
  - `line` = número da linha de menu que contém o item que foi marcado por último.

  Função : Retorna a posição do item do menu que foi marcado por último na chamada mais recente de `GetOption` ou `MarkOptions`.

- ```
  void SetCursorPosition(
            int  wnumber,
            int  block,
            int  line)
  ```

  Parâmetros :

  - `wnumber` = identificação da janela.
  - `block` = número do bloco de menu que contém o item a ser marcado.
  - `line` = número da linha de menu que contém o item a ser marcado.

  Função : Estabelece a posição do item do menu que deve ser inicialmente marcado pela função `GetOption` ou `MarkOptions`. Por default a posição inicial do item marcado é (bloco 1, linha 1) para a primeira chamada à `GetOption`. Nas demais,

o item marcado é aquele que foi selecionado por último na chamada mais recente. `SetCursorPosition` só precisa ser usado quando um comportamento diferente for desejado.

- `void MarkOptions(`
  ```
  int   wnumber,
  char  *EscapeSet,
  char  *escvalue)
  ```

  Parâmetros :

  - `wnumber` = número de identificação da janela da qual será selecionada uma opção do menu.
  - `EscapeSet` = conjunto de códigos das teclas que causam o retorno da função `MarkOptions`. Deve ser um subconjunto do conjunto **WASCII**.
  - `escvalue` = código da tecla usada para finalizar a operação de `MarkOptions`, devendo ser um do valores em `EscapeSet`.

  Função : Entra na janela especificada e permite ao usuário marcar um ou mais itens do menu. O usuário pode mover o cursor para qualquer item do menu, através das setas e das teclas **HOME** e **END** presentes no teclado do PC. A tecla **ENTER** seleciona ou anula a seleção do item marcado, sem causar retorno da rotina. Os itens selecionados são anotados na estrutura do menu associado à janela dada. O retorno só ocorre quando uma tecla em `EscapeSet` for pressionada, sendo o código desta tecla retornado no parâmetro `escvalue`.

- `void UnMarkMenu(TMENU *menu)`

  Parâmetro :

  - `menu` = descritor do menu.

  Função : Elimina todas as marcas de itens marcados do menu dado.

- `int IsMarkedItem(`
  ```
  TMENU  *menu,
  int    block,
  int    line)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  - `block` = número do bloco de menu a ser considerado.
  - `line` = linha do bloco que contém o item a ser considerado.

  Função : Informa se o item de menu indicado está ou não marcado.

- `void MarkItem(`
  ```
  TMENU  *menu,
  int    block,
  int    line)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  - `block` = número do bloco de menu a ser considerado.
  - `line` = linha do bloco com o item a ser marcado.

  Função : Marca o item de menu indicado.

- `void UnMarkItem(`
  ```
  TMENU  *menu,
  int    block,
  int    line)
  ```

  Parâmetros :

  - `menu` = descritor do menu.
  - `block` = número do bloco de menu a ser considerado.
  - `line` = linha do bloco que contém o item a ser desmarcado.

  Função : Retira a marca do item de menu indicado.

- `void GetFirstOption(TMENU *menu, char *option)`

  Parâmetros :

  - `menu` = descritor do menu.
  - `option` = código associado ao primeiro item de menu que estiver marcado.

  Função : Retorna em `option` o código de ação associado ao primeiro item marcado do menu dado, contando de cima para baixo, da esquerda para a direita. Esta função força a recuperação dos itens marcados a partir do início do menu. Se não houver item marcado, o valor ESC é retornado no parâmetro `option`. Esta função deve ser obrigatoriamente chamada antes das seqüências de chamada a `GetNextOption`.

- `void GetNextOption(TMENU *menu, char *option)`

  Parâmetros :

  - `menu` = descritor do menu.
  - `option` = código associado ao próximo item de menu marcado, que sucede ao último item retornado pela `GetFirstOption` ou `GetNextOption`.

Função : Retorna em `option` o código de ação associado ao primeiro item de menu que estiver marcado na seqüência contada de cima para baixo, da esquerda para a direita. A cada chamada, `GetNextOption` retorna a próxima opção que estiver marcada. Quando não houver mais opções marcadas, o código ESC será retornado. Toda seqüência de chamadas a esta função deve ser previamente iniciada por uma chamada a `GetFirstOption`.

12 Seqüência de Finalização

Ao finalizar o programa, deve-se executar a seqüência de comandos a seguir, de forma a devolver ao MS-DOS o acesso a toda a tela do vídeo :

```
MakeWindow(1,1,1,80,25,BLACK,WHITE);
GoToWindow(1);
clrscr();
```

Note que a janela poderia ser qualquer outra com número menor ou igual ao limite fixado por `MAXWINDOWS`.

13 Exemplos

Nesta seção são apresentados alguns exemplos que ilustram a utilização das funções de WINDOW 2.0.

13.1 Janela sem Moldura

Para construir uma janela com número de identificação 1, sem moldura, com fundo preto e texto branco (verde no caso de vídeo monocromático verde) e coordenadas absolutas — na forma de (coluna,linha) — de sua diagonal principal dadas por (5,10) e (20,30), deve-se programar :

```
MakeWindow(1,5,10,20,30,BLACK,WHITE);
```

13.2 Janela com Moldura

Para construir uma janela com número de identificação 2, com moldura branca, com fundo preto e texto branco (verde no caso de vídeo monocromático verde) e coordenadas absolutas — na forma (coluna,linha) — de sua diagonal principal dadas por (5,10) e (20,30), deve-se programar :

```
MakeWindow(2,5,10,20,30,BLACK,WHITE);
DrawFrame(2,WHITE);
```

13.3 Inscrições em Moldura

Para escrever, por exemplo TITULO DE CIMA, a partir da coluna 4 sobre a borda superior da moldura da janela 2 em letras vermelhas e TITULO DE BAIXO, sobre a borda inferior, na mesma coluna, em letras verdes, deve-se programar :

```
PutTopTitle (2,4,RED, "TITULO DE CIMA");
PutBottomTitle(2,4,GREEN,"TITULO DE BAIXO");
```

13.4 Acesso a uma Janela

Para se ter acesso a uma janela já criada, por exemplo janela 2, para nela escrever um texto qualquer, deve-se executar

```
GoToWindow(2)
```

Após este comando, todas informações enviadas à tela são exibidas na janela 2.

13.5 Pilha de Janelas

Para construir e empilhar uma janela de número 3, com moldura vermelha, fundo branco, texto preto e coordenadas absolutas da diagonal principal (15,10) e (23,25), e posteriormente restaurar a tela, retornando a janela inicial, deve-se programar :

```
MakeWindow(3,15,10,23,25,WHITE,BLACK);
EnterWindow(3);
```

```
DrawFrame(3,RED);
··· outros comandos do usuário
LeaveWindow();
```

13.6 Menu

Para montar um descritor de menu (variável menu) de 1 bloco com 2 linhas, contendo os itens : "item no. 1" com código de ação "A" e "item no. 2", com código de ação "B", que serão exibidos a partir das coordenadas (3,2) de uma janela qualquer, e com a marca de itens em fundo azul e texto amarelo, deve-se programar :

```
DefineMenu(menu,1,BLUE,YELLOW);
DefineBlock(menu,1,3,2,2);
SetItem(menu,1,1,'A',"item no. 1");
SetItem(menu,1,2,'B',"item no. 2");
```

Para exibir este menu na janela número 3 :

```
DisplayMenu(3,menu)
```

Para selecionar um item do menu :

```
GetOption(3,ESC PGDN PGUP,option)
```

Após a execução deste trecho, option terá o valor do código de ESC, PGUP, PGDN, "A" ou "B".

References

[1] Borland International, *Turbo C++ version 1.0*, User's Guide, 1990.

[2] Borland International, *Turbo C++ version 1.0*, Programmer's Guide, 1990.

[3] Borland International, *Turbo C++ version 1.0*, Library Reference, 1990.

[4] Kernighan, B. & Ritchie, D., *The C Programming Language (ANSI C)*, Prentice Hall Software Series, 1988.

[5] Norton, Peter, *Programmer's Guide to the IBM PC*, Microsoft Press, 1985.

[6] Bigonha, R. S., *WINDOW 2.0: Um Sistema Básico de Gerência de Interfaces (versão Turbo Pascal)*, Relatório Técnico 024/90, Departamento de Ciência da Computação, UFMG, 1990.

A APÊNDICE I - Programa `Demo`

O programa `Demo` apresentado a seguir ilustra as principais características do pacote WINDOW 2.0 versão TURBO C para administração de janelas e menus. O uso das funções do pacote é mostrado por meio de um exemplo que constrói várias janelas, e opera em cada uma delas.

Após cada grupo de comandos no programa `Demo`, a execução é temporariamente suspensa para que o usuário possa observar o efeito produzido na tela. Para continuar, basta pressionar uma tecla.

```
#include "windowc.h"

/*================================================================*/
/*                                                                */
/*                          DEMO.C                                */
/*                                                                */
/* Programa de demonstracao para ilustrar a utilizacao das fun-   */
/* coes para construcao de janelas e menus em TURBO C.            */
/*                                                                */
/*================================================================*/

TMENU   menu;                   /* descritor dos blocos do menu        */
int     firstrow,               /* primeira linha de um bloco de menu  */
        firstcolumn,            /* primeira coluna de um bloco de menu */
        numofrows,              /* numero de linhas de um bloco de menu*/
        lastline,
        lastblock,
        x0,y0,                  /* coordenadas absolutas               */
        x1,y1,                  /* coordenadas absolutas               */
        wnumber,                /* numero de identificacao de janelas  */
        block,                  /* bloco inicialmente marcado no menu  */
        line,                   /* linha inicialmente marcada no menu  */
        numofblocks,            /* numero de blocos do menu            */
        i,                      /* indice de comando for               */
        coluna;
char    option,                 /* codigo da opcao selecionada do menu */
        c;                      /* caractere lido para sair de pausa   */


/*================================================================*/
/* Desenha as bordas da janela por cima das janelas presentes no  */
/* video exibe um menu e permite a selecao de um item atraves das */
```

```
/* setas. Apos a selecao de um item, a janela desaparece e a tela */
/* reconstituida.                                                 */
/*================================================================*/

void    LeOpcaoNaJanela10(char *option) {
int     wnumber, coluna;
char    c;
TMENU   menu1;

    wnumber = 7;
    EnterWindow(wnumber);
    DrawFrame(wnumber,WHITE);
    clrscr();
    PutTopTitle(wnumber,28,RED,"JANELA 7");
    cputs("Estamos na janela 7. Janela 8 sera empilhada.");

    c = getch(); /*--- pausa para observacao ---*/

    wnumber = 8;
    EnterWindow(wnumber); /* salva area da janela 8 e nela entra */
    DrawFrame(wnumber,WHITE);
    coluna = 23;
    PutTopTitle(wnumber,coluna,WHITE,"JANELA 8");
    UnMarkMenu(&menu);
    CopyMenu(menu,&menu1);
    MarkItem(&menu,1,3);
    MarkItem(&menu,2,3);
    MarkItem(&menu,3,3);
    DisplayMenu(wnumber,&menu);
    gotoxy(3,12);
    cputs("Use ENTER para marcar/desmarcar itens desejados");
    gotoxy(3,13);
    cputs("Use ESC,Del,Ins, + ou - para finalizar marcacao");
    MarkOptions(wnumber, ESC "+-" INS DEL ,option);
    LeaveWindow(); /* restaura area da janela 8 e volta a 7 */


    wnumber = 10;
    EnterWindow(wnumber); /* salva area da janela 10 e nela entra */
    DrawFrame(wnumber,WHITE);
    coluna = 15;
    PutTopTitle(wnumber,coluna,BLACK,"JANELA 10");
    ChangeMenuInfo(&menu1,2,WHITE,BLACK);
    ChangeBlockInfo(&menu1,1,5,2,4);
    ChangeBlockInfo(&menu1,2,28,2,4);
    DisplayMenu(wnumber,&menu1);
    gotoxy(3,7);
    cputs("Use ENTER para selecionar um item");
```

```
    PutBottomTitle(wnumber,5,YELLOW + BLINK,
                   "Para terminar DEMO pressione F10");
    SetCursorPosition(wnumber,block,line);
    GetOption(wnumber, ESC F10 ,option);
    GetCursorPosition(wnumber,&block,&line);
    LeaveWindow(); /* restaura area da janela 10 e retorna  */


    gotoxy(1,8);
    textcolor(BLACK);
    cputs("Opcoes marcadas na janela 9 : ");
    GetFirstOption(&menu,&c);
    while(c != 0x1B) {
        cprintf("%c ", c);
        GetNextOption(&menu,&c);
    }
    ResetColors();

    if(!strchr(F10, *option)) c = getch(); /*--- pausa para observacao ---*/

    LeaveWindow();    /* restaura area ocupada pela janela 7 */

} /* LeOpcaoNaJanela10 */


/*=================================================================*/
/*               FUNCTION JANELAVARIAVEL                           */
/*=================================================================*/

void    JanelaVariavel(void) {
int     i;
TMENU   menu[3];

    DefineMenu(&menu[0],1,BLACK,WHITE);
    DefineBlock(&menu[0],1,1,1,10);
    SetItem(&menu[0],1,1, 'A'," opcao A ");
    SetItem(&menu[0],1,2, 'B'," opcao B ");
    SetItem(&menu[0],1,3, 'C'," opcao C ");
    SetItem(&menu[0],1,4, 'D'," opcao D ");
    SetItem(&menu[0],1,5, 'E'," opcao E ");
    SetItem(&menu[0],1,6, 'F'," opcao F ");
    SetItem(&menu[0],1,7, 'G'," opcao G ");
    SetItem(&menu[0],1,8, 'H'," opcao H ");
    SetItem(&menu[0],1,9, 'I'," opcao I ");
    SetItem(&menu[0],1,10,'J'," opcao J ");

    DefineMenu(&menu[1],1,BLACK,YELLOW);
    DefineBlock(&menu[1],1,8,1,10);
```

```
SetItem(&menu[1],1,1, 'K'," opcao K ");
SetItem(&menu[1],1,2, 'L'," opcao L ");
SetItem(&menu[1],1,3, 'M'," opcao M ");
SetItem(&menu[1],1,4, 'N'," opcao N ");
SetItem(&menu[1],1,5, 'O'," opcao O ");
SetItem(&menu[1],1,6, 'P'," opcao P ");
SetItem(&menu[1],1,7, 'Q'," opcao Q ");
SetItem(&menu[1],1,8, 'R'," opcao R ");
SetItem(&menu[1],1,9, 'S'," opcao S ");
SetItem(&menu[1],1,10,'T'," opcao T ");

DefineMenu(&menu[2],1,BLACK,YELLOW);
DefineBlock(&menu[2],1,8,1,10);
SetItem(&menu[2],1,1, 'U'," opcao U ");
SetItem(&menu[2],1,2, 'V'," opcao V ");
SetItem(&menu[2],1,3, 'X'," opcao X ");
SetItem(&menu[2],1,4, 'Y'," opcao Y ");
SetItem(&menu[2],1,5, 'Z'," opcao Z ");
SetItem(&menu[2],1,6, '0'," opcao 0 ");
SetItem(&menu[2],1,7, '1'," opcao 1 ");
SetItem(&menu[2],1,8, '2'," opcao 2 ");
SetItem(&menu[2],1,9, '3'," opcao 3 ");
SetItem(&menu[2],1,10,'4'," opcao 4 ");

MakeWindow(9,10,5,21,16,WHITE,BLACK);
EnterWindow(9);

   DrawFrame(9,GREEN);
   DisplayMenu(9,&menu[0]);
   GetOption(9,ESC,&c);

   for(i = 1;i <= 25; i++) RelocateWindow(9,10+i,5);
   GetOption(9, ESC LEFTARROW RIGHTARROW ,&c);

   RelocateWindow(9,10,10);
   GetOption(9, ESC ,&c);

   EnterWindow(6);
      EnterWindow(9);
          EnterWindow(6);
              clrscr();
              cputs("Use Setas para Mover Janela");
              DragWindow(9);
              GetOption(9, ESC , &c);
          LeaveWindow();
          c = getch();
      LeaveWindow();
       c = getch();
```

```
       LeaveWindow();
       c = getch();

    LeaveWindow();
    c = getch();

    for(i = 0;i < 3;i++) ReleaseBlocks(&menu[i]);

}       /* JanelaVariavel */


/*================================================================*/
/*  Inicializa o menu, informando o numero de blocos que o forma  */
/*  e as cores de background e foreground do item marcado.        */
/*================================================================*/
main() {

    StartUp();
    numofblocks  = 3;
    DefineMenu(&menu,numofblocks,RED,GREEN);


/*================================================================*/
/* Monta, atraves de chamadas a funcao Menublock, a descri-       */
/* cao de um menu de 3 blocks e 6 linhas/bloco. As coordenadas dos*/
/* blocos do menu, i.e., firstrow e firstcolumn sao relativas a   */
/* a janela onde o menu sera exibido. O menu so sera exibido apos */
/* ativacao da funcao DISPLAYMENU, que neste exemplo e feito      */
/* na funcao LeOpcaoNaJanela7.                                    */
/* Os itens do bloco sao definidos atraves de chamadas a funcao   */
/* MenuItem.                                                      */
/*================================================================*/

    block       = 1;
    numofrows   = 6;
    firstrow    = 3;
    firstcolumn = 3;
    DefineBlock(&menu,block,firstcolumn,firstrow,numofrows);

    SetItem(&menu,1,1,'A'," opcao A ");
    SetItem(&menu,1,2,'B'," opcao B ");
    SetItem(&menu,1,3,'C'," opcao C ");
    SetItem(&menu,1,4,'D'," opcao D ");
    SetItem(&menu,1,5,'E'," opcao E ");
    SetItem(&menu,1,6,'F'," opcao F ");


    block       = 2;
```

```
    numofrows   = 4;
    firstrow    = 3;
    firstcolumn = 22;
    DefineBlock(&menu,block,firstcolumn,firstrow,numofrows);

    SetItem(&menu,2,1,'G'," opcao G ");
    SetItem(&menu,2,2,'H'," opcao H ");
    SetItem(&menu,2,3,'I'," opcao I ");
    SetItem(&menu,2,4,'J'," opcao J ");



    block       = 3;
    numofrows   = 8;
    firstrow    = 3;
    firstcolumn = 41;
    DefineBlock(&menu,block,firstcolumn,firstrow,numofrows);

    SetItem(&menu,3,1,'K'," opcao K ");
    SetItem(&menu,3,2,'L'," opcao L ");
    SetItem(&menu,3,3,'M'," opcao M ");
    SetItem(&menu,3,4,'N'," opcao N ");
    SetItem(&menu,3,5,'O'," opcao O ");
    SetItem(&menu,3,6,'P'," opcao P ");
    SetItem(&menu,3,7,'Q'," opcao Q ");
    SetItem(&menu,3,8,'R'," opcao R ");


/*================================================================*/
/* Desenha bordas na janela 0. Forca cor de fundo ser GREEN e     */
/* limpa a janela.                                                */
/*================================================================*/

    GoToWindow(0);
    DrawFrame(0,YELLOW);
    textbackground(GREEN);
    clrscr();

/*================================================================*/
/* Salva toda a tela. Prometemos restaura-la no final.            */
/*================================================================*/

   EnterWindow(0);

/*================================================================*/
/* Constroi janela 1 com moldura, fundo BLACK, letra WHITE, sendo  */
/* as coordenadas absolutas da diagonal principal (3,2) e (38,7).  */
/* Identifica a janela, escrevendo sobre a moldura, de forma que   */
```

```
/* que esta identificacao nao vai ser "rolada".                    */
/*=================================================================*/

    wnumber = 1;
    x0      = 3;
    y0      = 2;
    x1      = 38;
    y1      = 7;
    MakeWindow(wnumber,x0,y0,x1,y1,BLACK,WHITE);
    DrawFrame(wnumber,RED);
    coluna = 13;
    PutBottomTitle(wnumber,coluna,GREEN,"JANELA 1");

    c = getch(); /*--- pausa para observacao ---*/

/*=================================================================*/
/* Constroi janela 2 sem moldura, sendo as coordenadas absolutas  */
/* da diagonal principal (3,9) e (38,14).                         */
/*=================================================================*/

    wnumber = 2;
    x0      = 3;
    y0      = 9;
    x1      = 38;
    y1      = 14;
    MakeWindow(wnumber,x0,y0,x1,y1,BLACK,GREEN);
    GoToWindow(wnumber);
    gotoxy(13,4);
    cprintf("Janela %d",wnumber);

    c = getch();  /*--- pausa para observacao ---*/

/*=================================================================*/
/* Constroi janela 3 com moldura, fundo GREEN e letra YELLOW,      */
/* sendo as coordenadas absolutas da diagonal principal (3,16) e   */
/* (38,21).                                                        */
/* Identifica a janela, escrevendo sobre a moldura, de forma que   */
/* que esta identificacao nao vai ser "rolada".                    */
/*=================================================================*/

    wnumber = 3;
    x0      = 3;
    y0      = 16;
    x1      = 38;
    y1      = 21;
    MakeWindow(wnumber,x0,y0,x1,y1,GREEN,YELLOW);
    DrawFrame(wnumber,WHITE);
    coluna = 13;
```

```
    PutTopTitle(wnumber,coluna,WHITE,"JANELA 3");

    c = getch();  /*--- pausa para observacao ---*/

/*=================================================================*/
/* Constroi janela 4 com moldura, fundo RED, letra WHITE, sendo    */
/* as coordenadas absolutas da diagonal principal (41,2) e (76,7). */
/* Identifica a janela, escrevendo sobre a moldura, de forma que   */
/* que esta identificacao nao vai ser "rolada".                    */
/*=================================================================*/

    wnumber = 4;
    x0      = 41;
    y0      = 2;
    x1      = 76;
    y1      = 7;
    MakeWindow(wnumber,x0,y0,x1,y1,RED,WHITE);
    DrawFrame(wnumber,BLUE);
    coluna = 13;
    PutBottomTitle(wnumber,coluna,WHITE,"JANELA 4");

    c = getch();  /*--- pausa para observacao ---*/

/*=================================================================*/
/* Constroi janela 5 sem moldura, sendo as coordenadas absolutas  */
/* da diagonal principal (41,9) e (76,14).                        */
/*=================================================================*/

    wnumber = 5;
    x0      = 41;
    y0      = 9;
    x1      = 76;
    y1      = 14;
    MakeWindow(wnumber,x0,y0,x1,y1,BLACK,RED);
    GoToWindow(wnumber);
    gotoxy(13,4);
    cprintf("Janela %d",wnumber);

    c = getch();  /*--- pausa para observacao ---*/

/*=================================================================*/
/* Constroi janela 6 com moldura, fundo WHITE e letra BLACK,       */
/* sendo as coordenadas absolutas da diagonal principal (41,16) e  */
/* (76,21).                                                        */
/* Identifica a janela, escrevendo sobre a moldura, de forma que   */
/* que esta identificacao nao vai ser "rolada".                    */
/*=================================================================*/
```

```
    wnumber = 6;
    x0      = 41;
    y0      = 16;
    x1      = 76;
    y1      = 21;
    MakeWindow(wnumber,x0,y0,x1,y1,WHITE,BLACK);
    DrawFrame(wnumber,YELLOW);
    coluna = 13;
    PutBottomTitle(wnumber,coluna,RED,"JANELA 6");

    c = getch();  /*--- pausa para observacao ---*/

/*=================================================================*/
/* Limpa todas as janelas criadas.                                 */
/*=================================================================*/

    for(wnumber = 1;wnumber <= 6;wnumber++) {
        GoToWindow(wnumber);
        clrscr(); /* so afeta a janela de numero wnumber */
    }


/*=================================================================*/
/* Constroi janela 7 SEM moldura, fundo RED  e letra WHITE, no     */
/* topo das demais janelas. A diagonal principal da janela tem     */
/* coordenadas absolutas (8,4) e (71,22).                          */
/*=================================================================*/

    x0 = 8;
    y0 = 4;
    x1 = 71;
    y1 = 22;
    MakeWindow(7,x0,y0,x1,y1,YELLOW,BLACK);


/*=================================================================*/
/* Constroi janela 8 sem moldura, fundo BLUE e letra YELLOW, no    */
/* topo das demais janelas. A diagonal principal da janela tem     */
/* coordenadas absolutas (10,5) e (63,17).                         */
/*=================================================================*/

    x0 = 13;
    y0 =  7;
    x1 = 66;
    y1 = 21;
    MakeWindow(8,x0,y0,x1,y1,BLUE,YELLOW);

/*=================================================================*/
```

```
/* Constroi janela 10 sem moldura, fundo BLUE e letra YELLOW, no  */
/* topo das demais janelas. A diagonal principal da janela tem    */
/* coordenadas absolutas (10,5) e (63,17).                        */
/*================================================================*/

    x0 = 20;
    y0 =  7;
    x1 = 60;
    y1 = 16;
    MakeWindow(10,x0,y0,x1,y1,GREEN,YELLOW);



/*================================================================*/
/* Desenha a borda da janela 7 em cima das demais, exibe o        */
/* menu definido acima, permite ao operador selecionar um item    */
/* deste menu  e escreve opcao escolhida nas demais janelas apos  */
/* restaura-las. Estas operacoes sao repetidas ate o operador     */
/* pressionar a tecla PgDn.                                       */
/*================================================================*/


    block   = 1;
    line    = 1;
    wnumber = 6;
    do {

          wnumber = (wnumber % 7) + 1;
          if(wnumber == 7) {

                   /* Exibe janela 7 com menu e le opcao */
                   c = getch();  /*--- pausa para observacao ---*/

                   GetCursorPosition(8,&lastblock,&lastline);
                   if((lastblock == 3) && (lastline == 8)) {
                             block = 1;
                             line  = 1;
                   }

                   JanelaVariavel();

                   LeOpcaoNaJanela10(&option);
          }
          else {

                   /* Escreve opcao lida da 10 nas demais janelas */
                   GoToWindow(wnumber);
                   cprintf("\nOpcao selecionada na janela 10 = %c",option);
```

```
        }
    } while(!strchr(F10, option));


/*================================================================*/
/* Volta a tela inicial, toda cheia de x, conforme prometido.     */
/*================================================================*/

    LeaveWindow();

/*================================================================*/
/* Verifica se houve erro, inspecionando a variavel ErrorCond     */
/*================================================================*/

    if(ErrorCond != 0) {
       if(ErrorCond & 1) cputs("Overflow da pilha de janelas.");
       if(ErrorCond & 2) cputs("Falta espaco para salvar tela.");
       if(ErrorCond & 4) cputs("Tentativa de restaurar tela nao salva.");
       if(ErrorCond & 8) cputs("Numero de identificacao de janela invalido.");
       if(ErrorCond & 16) cputs("Modulo WINDOW nao foi inicializado.");
       if(ErrorCond & 32) cputs("Linha do menu fora de range.");
       if(ErrorCond & 64) cputs("Bloco do menu fora de range.");
       if(ErrorCond & 128) cputs("Coluna do menu fora de range.");
       if(ErrorCond & 256) cputs("Coordenadas de janela fora da tela.");
       if(ErrorCond & 512) cputs("Tentativa de acesso a menu nao exibido.");
       if(ErrorCond & 1024) cputs("Underflow na pilha de janelas.");
       if(ErrorCond & 2048) cputs("Coordenadas da janela invalidas.");
       if(ErrorCond & 4096) cputs(" Menu nao inicializado");
       if(ErrorCond & 8192) cputs("Overflow do heap.");
       if(ErrorCond & 16384) cputs("Codigo de cor invalido.");
       if(ErrorCond & 32768) cputs("Janela demais.");

       ErrorCond = 0; /* Apaga condicao de erro */

    }
    else cputs(" Tudo se passou na mais perfeita ordem.");

    c = getch();  /*--- pausa para observacao ---*/


/*================================================================*/
/* Devolve ao Heap as linhas do menu.                             */
/*================================================================*/

    ReleaseBlocks(&menu);

/*================================================================*/
/* Limpa toda a tela para uso do MS-DOS.                          */
```

```
/*=================================================================*/

    MakeWindow(1,1,1,80,25,BLACK,LIGHTGRAY);
    GoToWindow(1);
    clrscr();

    return 1; /* Successs */
}
```