

RELATÓRIO TÉCNICO DO DEPARTAMENTO
DE CIÊNCIA DA COMPUTAÇÃO
NÚMERO 788

WINDOW 3.0

Janelas e Menus em PASCAL TURBO

por

Roberto da Silva Bigonha

Belo Horizonte
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
Caixa Postal 702
1988

SINOPSE

Um conjunto de procedimentos para construção de janelas e menus implementado em PASCAL TURBO é apresentado. Estes procedimentos permitem ao usuário incorporar facilmente janelas e menus ao seu programa no sentido de produzir interfaces homem-máquina de boa qualidade. Os procedimentos do conjunto possibilitam a manutenção de várias janelas, superpostas ou não, e fácil seleção de itens de menu.

ABSTRACT

A software package for window and menu handling in TURBO PASCAL is presented. The procedures 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 items from menus.

SUMÁRIO

1. Introdução	1
2. Importação de Procedimentos de WINDOW.PAS	2
3. Identificadores Reservados	2
4. A Tela do Monitor	3
5. As Cores	3
6. Erros de Processamento	4
7. Tipo TMENU	5
8. Janelas	6
8.1 Sequência de Inicialização	6
8.2 Construção de Janelas	7
8.3 Ativação de Janelas	10
8.4 Empilhamento de Janelas	10
8.5 Outras Operações sobre Janelas	11
9. Menus	11
9.1 Construção de Menus	11
9.2 Exibição de Menus e Seleção de Itens	13
10. Sequência de Finalização	14
11. Exemplos	15
11.1 Janela sem Moldura	15
11.2 Janela com Moldura	15
11.3 Inscrições em Moldura	16
11.4 Acesso a uma Janela	16
11.5 Pilha de Janelas	16
11.6 Menu	17
12. Agradecimentos	17
13. Referências Bibliográficas	17
APÊNDICE A	
A.1 Programa de Demonstração Demo3.....	18
A.2 Fonte do arquivo WINDOW3.PAS	27

1. Introdução

Um conjunto de procedimentos para construção de janelas e menus foi desenvolvido com o objetivo de facilitar a implementação de interfaces homem-máquina mais agradáveis de se usar, dando ao software do usuário um aspecto mais profissional.

Os procedimentos implementados podem ser facilmente incorporados a qualquer programa escrito em PASCAL TURBO [1] desde que o usuário respeite algumas convenções em relação a nomes de seus identificadores definidos no mesmo escopo.

WINDOW 3.0 é uma nova versão do sistema WINDOW descrito em [1]. Estes dois sistemas possuem diferenças significativas na forma de ativação e uso de seus procedimentos. Contudo, a conversão de WINDOW em WINDOW 3.0 é relativamente simples.

O arquivo ou módulo contendo o sistema tem nome WINDOW3.PAS. Neste arquivo existem procedimentos para criar janelas a partir das coordenadas de suas diagonais principais, para colocar molduras em janelas ou redefinir padrões de cores do texto e do fundo de cada janela. Pode-se empilhar e desempilhar janelas, bem como passar de uma janela a outra. Há também procedimentos para construir menus, exibí-los em janelas e deles selecionar itens.

O conjunto de procedimentos e as estruturas de dados de WINDOW3.PAS ocupam um pouco menos de 5 Kbytes em área de código e cerca de 8 Kbytes em área de dados. Através do mecanismo de OVERLAY de procedimentos existente no TURBO PASCAL é possível reduzir a área de código para cerca de 2 Kbytes. Os grupos de OVERLAY recomendados estão indicados em comentários no cabeçalho dos procedimentos do arquivo WINDOW3.PAS. Em relação à área de dados, 8.000 bytes são reservados para salvar até duas telas completas de 25X80 caracteres e seus atributos de cor. Esta área pode, dependendo da aplicação ser substancialmente reduzida, alterando-se o valor da constante WMAXofSaveArea, que define o seu tamanho máximo. As demais estruturas de dados de WINDOW3.PAS ocupam pouco espaço.

2. Importação dos Procedimentos de WINDOW3.PAS

Para tornar os procedimentos de WINDOW3.PAS disponíveis em seu programa, o usuário deve incluir a seguinte diretiva do PASCAL TURBO imediatamente antes da palavra chave VAR, que inicia seções de declarações de variáveis:

```
(*$I WINDOW3.PAS *)
```

A partir deste ponto, os procedimentos e os tipos TMENU e TLINE definidos em WINDOW3.PAS ficam visíveis no escopo do bloco do usuário que contém o INCLUDE acima.

3. Identificadores Reservados

Os identificadores apresentados a seguir estão declarados no arquivo WINDOW3.PAS, e pela regra de escopo do PASCAL TURBO são visíveis no programa do usuário no escopo. Recomenda-se ao usuário não utilizar estes identificadores diretamente sob pena de comprometer a robustez do conjunto. Nada impede, contudo, o livre acesso a estes identificadores, inclusive para alterar seus valores, no caso de variáveis.

Para minimizar a possibilidade de conflitos de identificadores do usuário com os declarados em WINDOW3.PAS, estes têm seus nomes sempre iniciados pela letra W.

a) Constantes

Os seguintes identificadores denotam constantes que definem os tamanhos máximos de estruturas de dados usadas pelo sistema. O usuário poderá ajustar estes valores diretamente no arquivo WINDOW3.PAS de acordo com suas conveniências.

■ WMAXNUM = 8

Número máximo de janelas que podem ser construídas simultaneamente. Toda janela tem um número de identificação no intervalo [0, WMAXNUM]. O número 0 é reservado pelo sistema para designar a janela inicial correspondendo 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.

■ WMAXSTACK = 10

Tamanho máximo da pilha de descritores de áreas de vídeo salvas devido a empilhamento de janelas.

■ WMAXLINE = 25

Tamanho máximo de uma linha do vídeo.

■ WMAXBLOCK = 4

Número máximo de blocos de um menu. Um bloco de menu é uma lista de itens, contendo um item por linha.

■ WMAXofSaveArea = 4000

Número máximo de caracteres e respectivos atributos de cor que podem ser salvos, em função de empilhamento de janelas. Cada caractere salvo necessita de um byte para seu código ASCII e um byte para o seu atributo de cor.

b) Tipos, Variáveis e Procedimentos

Os seguintes identificadores somente são de interesse local ao módulo WINDOW3 :

WcurBackColor, WcurTextColor, WActiveWindow, WWindowInfo, WStackTop, WWindowStack, WAreaPtr, WScreenArea, WInitVideo, WcheckInitialization, WcheckWindowNumber, WcheckIndices, WLeaveWindow, WInitMenu, WcleanMenu e WPutTitle.

4. A Tela do Monitor

A tela do monitor de ter 80 colunas X 25 linhas, com a convenção usual de se ter (coluna 1, linha 1) no canto superior esquerdo da tela e (coluna 80, linha 25), no canto inferior direito. Nesta convenção uma coordenada (x,y) denota a posição de um caractere na coluna x e linha y.

5. As Cores

Ao construir janelas, o usuário deve especificar as cores do fundo e do texto desejadas. As cores possíveis podem ser especificadas pelas seguintes constantes (do tipo INTEGER) disponíveis

no PASCAL TURBO:

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

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

6. Erros de Processamento

As condições de erro detectadas pelo sistema WINDOW são anotadas na variável global ErrorCond, que é do tipo INTEGER. Cada um dos bits desta variável é usado para indicar um tipo de erro, conforme é mostrado pelo seguinte quadro :

ErrorCond	Explicação do Erro
0000 0000 0000 0000 0000 (0)	= nenhum erro
0000 0000 0000 0000 0001 (1)	= Overflow da pilha de janelas
0000 0000 0000 0000 0010 (2)	= Overflow da pilha de telas
0000 0000 0000 0000 0100 (4)	= Janela anterior não existe
0000 0000 0000 0000 1000 (8)	= Número da janela invalido
0000 0000 0000 0001 0000 (16)	= WINDOW3 nao foi inicializado
0000 0000 0000 0010 0000 (32)	= Linha de menu fora de range
0000 0000 0000 0100 0000 (64)	= Bloco de menu fora de range
0000 0000 0000 1000 0000 (128)	= Coluna de menu fora de range
0000 0000 0001 0000 0000 (256)	= Coordenadas fora de range
0000 0000 0010 0000 0000 (512)	= Menu indefinido

O usuário pode consultar ou alterar o valor de ErrorCond livremente. Por exemplo, o seguinte trecho de programa ilustra a

apuração e listagem das condições de erro detectadas pelo sistema durante uma dada execução:

```
if ErrorCond <> 0
then begin
  if ErrorCond and 1 <> 0
  then writeln('Overflow da pilha. ');
  if ErrorCond and 2 <> 0
  then writeln('Falta espaço para salva tela. ');
  if ErrorCond and 4 <> 0
  then writeln('Tentativa de restaurar tela nao salva. ');
  if ErrorCond and 8 <> 0
  then writeln('Numero de identificacao de janela invalido. ');
  if ErrorCond and 16 <> 0
  then writeln('Modulo WINDOW nao foi inicializado. ');
  if ErrorCond and 32 <> 0
  then writeln('Linha do menu fora de range. ');
  if ErrorCond and 64 <> 0
  then writeln('Bloco do menu fora de range. ');
  if ErrorCond and 128 <> 0
  then writeln('Coluna do menu fora de range. ');
  if ErrorCond and 256 <> 0
  then writeln('Coordenadas da janela fora de range. ');
  if ErrorCond and 512 <> 0
  then writeln('Tentativa de acesso a menu indefinido. ');
  Errorcond := 0; (* Apaga condicao de erro *)
end
else write('Tudo se passou na mais perfeita ordem.');
```

7. Tipo TMENU

Os menus do usuário devem necessariamente ser declarados do tipo TMENU. Este tipo, que é importado de WINDOW.PAS, descreve para cada bloco de menu os seguintes parâmetros :

■ firstrow

Número da primeira linha do bloco, relativo a janela associada.

■ firstcolumn

Número da coluna do bloco, relativo à janela associada.

■ numofrows

Número de linhas no bloco do menu.

Para cada linha do bloco, TMENU descreve :

■ text

Texto da linha do bloco, que é exibido na janela. Este campo tem tipo TLINE, que é compatível com STRING[80].

■ code

Código (do tipo CHAR) que deve ser retornado se esta linha do menu for selecionada (Vide procedimento GetOption).

Não é necessário se ter acesso diretamente a estes componentes do tipo TMENU. Basta que o usuário declare suas variáveis de menu com este tipo e use os procedimentos InitMenu, MenuBlock e MenuItem para montar os descritores de menu desejados.

8. Janelas

Os procedimentos necessários à construção de janelas e de menus são descritos a seguir.

8.1 Sequência de Inicialização

O conjunto de procedimentos de WINDOW3.PAS utiliza internamente estruturas de dados que devem ser devidamente inicializadas para garantir o bom funcionamento do sistema. É também necessário que se informe o tipo de adaptador de vídeo em uso no microcomputador, para permitir o correto acesso à área de memória do vídeo [2]. Esta inicialização é executada por um dos procedimentos abaixo, que deve ser necessariamente ativado antes de qualquer outro de WINDOW3.PAS que execute ações relacionadas com a tela.

■ **procedure** InitMonoVideo

Função :

Inicializa o sistema WINDOW3 em equipamento que tem o

MONOCHROME ADAPTER.

■ **procedure** InitHerculesVideo

Função #

Inicializa sistema WINDOW3 em equipamento com placa HERCULES DISPLAY ADAPTER.

■ **procedure** InitCGAVideo

Função #

Inicializa o sistema WINDOW3 em equipamento com o Color/Graphics Adapter. Este é o tipo de adaptador utilizado pela maioria dos microcomputadores nacionais.

Se você desconhecer o tipo de adaptador de seu vídeo, experimente CGA em primeiro lugar. O sintoma característico de uma inicialização errada é restauração incorreta de tela após desempilhamento de janelas. Uma outra alternativa é usar a função HerculesPresent de WINDOW3 para determinar se o adaptador presente é HERCULES. No caso de não ser, é quase certo que, no Brasil, seja CGA. Esta função tem a seguinte interface #

■ **function** HerculesPresent : boolean;

Função #

Retorna TRUE se o adaptador de vídeo do microcomputador é HERCULES; do contrário retorna FALSE.

O usuário também deve tomar o cuidado de compilar o arquivo WINDOW3.PAS com um PASCAL TURBO corretamente instalado para operar com o adaptador de vídeo de seu computador. Arquivos .COM gerados pelo PASCAL TURBO podem não ser transportáveis entre máquinas com diferentes adaptadores de vídeo.

8.2 Construção de Janelas

Uma janela pode, opcionalmente, ter uma moldura. Para se construir uma janela sem moldura, usa-se o procedimento MakeWindow. A moldura pode ser colocada posteriormente via procedimento DrawFrame. A moldura não faz parte da área útil da janela, de forma a ficar protegida contra "scrolling" e escrita

na tela. O usuário pode contudo, através de procedimentos especiais, escrever sobre as molduras.

Cada janela tem seu próprio padrão de cor, que é definido no momento de sua criação, quando deve-se especificar a cor de fundo e a do texto da janela.

```
■ procedure MakeWindow(wnumber      : integer;
                       x0,y0,x1,y1   : integer;
                       ColorOfTheBack : integer;
                       ColorOfText   : integer)
```

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 de fundo da janela.
ColorOfText = código da cor do texto de janela.

Função :

Constrói uma janela cuja diagonal principal tem as coordenadas da tela (x0,y0) e (x1,y1).

Observações :

- a) Este procedimento não altera o conteúdo da tela e nem ativa a janela 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 pelo procedimento GotoWindow;
- b) Após o retorno deste procedimento, volta-se à janela em que se estava imediatamente antes de sua chamada.
- c) Na primeira vez em que se entrar na janela criada, o cursor ficará na posição (1,1). Nas demais, o cursor retorna à posição em que estava no momento em que a janela foi abandonada.
- d) As coordenadas (x0,y0) e (x1,y1) são absolutas referindo-se

a toda a tela de 25 linhas por 80 colunas. Este é o único procedimento em que as coordenadas passadas como parâmetros são absolutas. Em todos os outros as coordenadas são relativas à janela que estiver ativa.

■ **procedure** DrawFrame(wnumber : integer)

Parâmetro :

wnumber = número de uma janela construída pelo procedimento MakeWindow.

Função :

Desenha uma moldura formada por duas linhas paralelas, envolvendo a janela wnumber.

Observações :

- a) A moldura ocupa duas linhas e duas colunas da janela. Estas linhas e colunas não são consideradas partes da janela, de forma que são preservadas contra "scrolling" da janela.
- b) Após o retorno deste procedimento, volta-se à janela em que se estava imediatamente antes de sua chamada.

■ **procedure** PutTopTitle(wnumber : integer;
 column : integer;
 txt : TLINE);

Parâmetro :

wnumber = número de identificação da janela.
column = coluna dentro da janela.
txt = texto a ser escrito na borda superior da janela.

Função :

Escrever o texto txt na borda superior da janela, a partir da coluna column.

■ **procedure** PutBottomTitle(wnumber : integer;
 column : integer;
 txt : TLINE);

Parâmetro :

wnumber = número de identificação da janela.
column = coluna dentro da janela.
txt = texto a ser escrito na borda inferior da janela.

Função #

Escrever o texto txt na borda inferior da janela, a partir da coluna column.

8.3 Ativação de Janelas

Após a inicialização do sistema via um dos procedimentos descritos na Seção 8.1, a janela ativa é a de número 0. As janelas do usuário só podem receber números de identificação no intervalo [1,WMAXNUM]. Após criar janelas através do procedimento MakeWindow, o usuário pode passar de uma janela a outra pela ativação do procedimento GotoWindow.

■ **procedure** GotoWindow(wnumber # integer)

Parâmetro #

wnumber = número de identificação da janela a ser ativada.

Função #

Entrar 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 (1,1).

Observações #

- a) GotoWindow é o único meio de se passar de uma janela a outra.
- b) Somente uma janela pode estar ativa de cada vez.

8.4 Empilhamento de Janelas

Quando se deseja sobrepor janelas, a área da tela que será ocupada por cada nova janela deverá ser salva de forma a permitir a restauração da tela quando a nova janela não for mais necessária. As operações de salvamento e restauração de janela são efetuadas através dos procedimentos PushWindow e PopWindow.

■ **procedure** PushWindow;

Função #

Salva em uma pilha os caracteres (e seus respectivos atributos de cor) que estão contidos no retângulo definido pela

janela correntemente ativa, incluindo a moldura, se houver.

■ procedure PopWindow;

Função :

Restaura a janela que foi salva mais recentemente.

Observação :

- a) PopWindow não causa mudança de janela corrente. Se a janela que estava ativa imediatamente antes da chamada a este procedimento desaparecer com sua execução, ainda poder-se-á ter acesso a ela, porém, as informações apagadas da tela, como por exemplo, molduras, não são recuperadas.

8.5 Outras Operações sobre Janelas

Os seguintes procedimentos do PASCAL TURBO, que podem ser usados livremente pelo usuário de WINDOW3, têm efeito apenas na janela que estiver ativa:

write, writeln, GoToXY, ClrScr, ClrEol, DelLine, TextColor, TextBackground, WhereX e WhereY.

O procedimento Window do PASCAL TURBO, para criar janelas, não deve ser utilizado pelo usuário sob o risco de comprometer a robustez de todo o sistema.

9. Menus

Menus são constituídos de blocos, sendo cada bloco formado por linhas contendo as opções a serem selecionadas. Cada janela pode ter um menu associado, e um mesmo menu pode ser associado a várias janelas.

9.1 Construção de Menus

Do ponto de vista de WINDOW3, um menu é uma estrutura de dados do tipo TMENU, que deve ser declarada e explicitamente inicializada pelo usuário através dos procedimentos InitMenu, MenuBlock e MenuLine.

```
■ procedure InitMenu(var menu      # TMENU);  
                    numofblocks # integer);
```

Parâmetros #

menu = descritor do menu.
numofblocks = número de blocos que formam o menu. Um bloco é uma coluna de itens ou opções.

Função #

Inicializa toda a estrutura do menu, atribuindo valores default a todos os seus campos.

Observação #

a) Este procedimento deve, obrigatoriamente, ser chamado uma vez para cada menu declarado pelo usuário.

```
■ procedure InitBlock(var menu      # TMENU;  
                     block        # integer;  
                     blockfirstrow # integer;  
                     blockfirstcolumn # integer;  
                     blocknumofrows # integer)
```

Parâmetros #

menu = descritor do menu.
block = número do bloco do menu. Os blocos são numerados a partir de 1.
blockfirstrow = primeira linha do bloco, relativa a janela.
blockfirstcolumn = primeira coluna do bloco, relativa a janela.
blocknumofrows = número de linhas no bloco.

Função #

Define os parâmetros de um bloco de um menu. Este procedimento deve ser chamado para cada um dos blocos de um menu.

```
■ procedure InitItem(var menu # TMENU;  
                    block # integer;  
                    line # integer;  
                    code # char;  
                    text # WTLIN)
```

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.
code = código da ação a ser executada quando a linha especificada for selecionada.
text = texto da linha do menu. Deve ser um string de até 80 caracteres ou variável do tipo TLINE;

Função :

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

9.2 Exibição de Menus e Seleção de Itens

Os menus definidos através dos procedimentos InitMenu, InitBlock e InitItem podem ser exibidos em qualquer janela com o objetivo de selecionar opções. Os procedimentos descritos a seguir são encarregados das funções de exibição de menu e seleção de itens.

■ procedure DisplayMenu(wnumber : integer;
 menu : TMENU);

Parâmetros :

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

Função :

Exibir na janela especificada o menu passado como parâmetro ao mesmo tempo em que vincula o menu à janela wnumber.

■ procedure GetOption(wnumber : integer;
 VAR option : char);

Parâmetros :

wnumber = número de identificação da janela da qual será selecionada uma opção do menu.
option = código da opção selecionada (Vide procedimento InitItem).

Função :

Entrar na janela especificada, marcar um item do menu e permitir ao usuário passar a marca para qualquer outro item do menu, através das setas e teclas HOME e END presentes no teclado dos PCs. A tecla RETURN seleciona o item marcado, retornando no parâmetro option o código associado à linha selecionada (Vide procedimento MenuItem). A tecla ESC força um retorno com option = ESC.

Observação :

a) Inicialmente o item marcado é o do bloco 1 linha 1. A partir de então, GetOption sempre inicia-se marcando o último item que foi marcado na chamada anterior.

```
■ procedure GetMarkPosition( wnumber : integer;  
                             var block : integer;  
                             var line : integer);
```

Parâmetros :

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

Função :

Retornar a posição do item do menu que foi marcado por último na chamada mais recente da GetOption.

```
■ procedure SetMarkPosition( wnumber : integer;  
                             block : integer;  
                             line : integer);
```

Parâmetros :

wnumber = identificação da janela.
block = número do bloco que contém o item a ser marcado.
line = número da linha que contém o item a ser marcado por último.

Função :

Estabelece a posição do item do menu que deve ser inicial-

mente marcado pela `GetOption`.

Observação :

- a) Por default a posição inicial do item marcado é bloco 1, linha 1 para a primeira chamada à `GetOption`. Nas demais chamadas, o item marcado é aquele que foi marcado por último na chamada mais recente. `SetMarkPosition` só precisa ser usado quando um comportamento diferente for desejado.

10. Sequência de Finalização

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

```
GoToWindow(0);  
ClrScr;
```

11. Exemplos

Nesta seção são apresentados alguns exemplos para ilustrar a utilização dos procedimentos de `WINDOW3`. Suponha que `menu` seja uma variável declarada com tipo `TMENU`.

11.1 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);
```

11.2 Janela com Moldura

Para construir uma janela com número de identificação 2, com moldura, 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);
```

11.3 Inscrições em Moldura

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

```
PutTopTitle (2,4,'TITULO DE CIMA');  
PutBottomTitle(2,4,'TITULO DE BAIXO');
```

11.4 Acesso a uma Janela

Para se ter acesso a uma janela já criada, por exemplo à 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.

11.5 Pilha de Janelas

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

```
MakeWindow(3,5,20,30,WHITE,BLACK);  
DrawFrame(3);  
GoToWindow(3);  
PushWindow;  
..... outros comandos do usuário  
PopWindow;
```

11.6 Menu

Para montar um descritor de menu de 1 bloco de 2 linhas, com os itens # "item no. 1" com código de ação "A" e "item no. 2" com código de ação "B" e que será exibido a partir das coordenadas (3,3) de uma janela qualquer, deve-se programar :

```
InitMenu(menu,1);  
InitBlock(menu,1,3,3,2);  
InitItem(menu,1,1,'A','item no. 1');  
InitItem(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,option)
```

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

12. Agradecimentos

Agradeço à Mariza Andrade da Silva Bigonha, pela co-autoria de alguns dos procedimentos utilizados pelo WINDOW3.PAS, ao Wagner Machado Nunan Zola pela implementação da função HerculesPresent e aos colegas João Eduardo de Rezende Dantas, Ivan Moura Campos e Osvaldo Sérgio Farhat de Carvalho pelas preciosas sugestões.

13. Referências Bibliográficas

1. Borland International, Turbo Pascal version 3.0 Reference Manual, 1985.
2. Norton, Peter, Programmer's Guide to the IBM PC, Microsoft Press, 1985.
3. Bigonha, R. S., "Conjunto de Procedimentos em PASCAL TURBO para construção de Janelas e Menus", Relatório Técnico No. 18/87, Departamento de Ciência da Computação, UFMG, 1987.

APÊNDICE A

A.1 Programa de Demonstração Demo3

O programa Demo3 apresentado a seguir ilustra as principais características do pacote WINDOW3 para administração de janelas e menus. O uso dos procedimentos do pacote são mostrados por meio de um exemplo que constrói oito (8) janelas, numeradas de 0 a 7, e opera em cada uma delas .

A janela inicial, que automaticamente recebe o número de identificação 0, é totalmente preenchida com a letra 'O', para melhor demonstrar o efeito causado pelos procedimentos ativados.

A janela de número 7, que é temporariamente construída sobre as demais, permite selecionar uma opção de um menu. O programa, a título de ilustração, sempre escreve a opção selecionada em todas as outras janelas.

Após cada grupo de comandos no programa DEMO3, a execução é temporariamente suspensa para que o usuário possa observar o efeito produzido na tela. Para continuar basta pressionar uma tecla qualquer.

A execução do DEMO3 entra em fase de encerramento quando a tecla ESC for pressionada estando o cursor dentro da janela 7.