

104884-0



INTERFACE COM DISPOSITIVOS  
DE ENTRADA GRÁFICA

por

Silvia Delgado Olabbarriaga \*

RP no. 83

DEZEMBRO 1987

Atualização:

DEZEMBRO 1988

-----  
| Nota técnica do projeto "Banco de Dados é |  
Ferramentas para CAD de Sistemas Digitais"

Orientação : Profa. Carla M. Dal Sasso-Freitas \*\*

\* trabalho realizado com apoio da FINEP

\*\* trabalho realizado com apoio do CNPQ

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
Av. Osvaldo Aranha, 99  
90001 - Porto Alegre - RS - BRASIL  
Telefone : (0512) 21-8499  
Telex : (051) 2680 - CCUF BR

Correspondência : UFRGS - CPGCC  
Caixa Postal 1501  
90001 - Porto Alegre - RS - Brasil

UFRGS  
INSTITUTO DE INFORMÁTICA  
BIBLIOTECA

Comissão Editorial: Taisy Silva Weber  
Carla Maria Dal Sasso Freitas

**UFRGS**

Reitor: Prof. GERHARD JACOB

Pró-reitor de Pesquisa e Pós-Graduação: Prof. ABILIO A. BAETA NEVES

Coordenador do CPGCC: Profa. Ingrid J. Pôrto

Comissão Coordenadora do CPGCC: Prof. Carlos A. Heuser  
Prof. Dalcídio M. Claudio  
Prof. Flavio Wagner  
Profa. Ingrid J. Pôrto  
Prof. Roberto T. Price  
Prof. Ricardo Reis

Bibliotecária CPGCC/CPD: Margarida Buchmann

**UFRGS**  
**INSTITUTO DE INFORMÁTICA**  
**BIBLIOTECA**

## RESUMO

O relatório apresenta uma interface padrão para acesso a dispositivos de entrada gráfica constituída de um grupo de rotinas dependentes de dispositivo. O objetivo principal deste conjunto de rotinas é o de servir de base para o pacote gráfico (independente de dispositivo) do AMPLO. A interface implementa funções de inicialização, obtenção das coordenadas do cursor, leitura do estado do botão associado ao cursor e de interrogação de características dependentes dos dispositivos.

PALAVRAS-CHAVE : computação gráfica, interface de entrada, funções gráficas.

## ABSTRACT

This report presents a device-dependent interface constituted by a set of graphical input functions for the acquisition of graphical data. The main goal of these functions is to implement the lower layer of the graphical software used in AMPLO. On the top of this layer there is a graphical, device-independent package. The functions allow initialization, reading the cursor position and button status as well as interrogation of devices features.

KEYWORDS : computer graphics, input interface, graphical functions.

## SUMÁRIO

LISTA DE FIGURAS .....	6
1. INTRODUÇÃO .....	7
2. O QUE É "IE" .....	8
3. DISPOSITIVO PADRÃO DE ENTRADA .....	9
4. FUNÇÕES DA IE .....	10
4.1 Controle .....	10
4.2 Entrada .....	10
4.3 Interrogação .....	10
5. COMO USAR A IE EM UM PROGRAMA APLICATIVO .....	11
5.1 Uso de mouse através da IE .....	11
5.2 Uso de mesa digitalizadora através da IE .....	12
5.3 Exemplo .....	13
6. MANUAL DE REFERÊNCIA .....	15
7. IMPLEMENTAÇÃO DA IE .....	17
7.1 IEMOUSE .....	17
7.2 IEMESA .....	18
REFERÊNCIAS BIBLIOGRÁFICAS .....	20
ANEXO: IMPLEMENTAÇÃO DA IE PARA MOUSE E TECLADO .....	21

## LISTA DE FIGURAS

Figura 3.1 - Dispositivo padrão de entrada gráfica .....	9
Figura 6.1 - Limites de coordenadas .....	16

## 1. INTRODUÇÃO

Um dos aspectos principais do AMPLO (Ambiente para Projeto Lógico de Sistemas Digitais) [WAG 86, WAG 87b] é a preocupação com a interface homem-máquina. Todas as ferramentas do ambiente serão ativadas a partir de uma interface de alto-nível, denominada LAGO (Linguagem de Acesso Global ao AMPLO), onde recursos gráficos deverão ser utilizados para se atingir maior eficiência. A partir desta interface poderão ser ativados os compiladores e editores (gráficos) das linguagens de descrição de hardware [WAG 87a], os simuladores associados a cada nível de descrição e realizadas consultas à base de dados. Cada uma destas ferramentas interage com o projetista segundo uma linguagem de comandos. No caso dos compiladores e das consultas à base de dados, esta linguagem é textual, orientada por cardápios. No caso dos simuladores, está prevista a utilização de representação gráfica de parâmetros de entrada e saída. No caso dos editores gráficos, obviamente é intensa a utilização de comunicação através de símbolos gráficos.

A necessidade de recursos gráficos por parte de diversas ferramentas no ambiente, bem como o fato da linguagem de programação escolhida (C) não contar com biblioteca de sub-rotinas gráficas, originaram a definição de um pacote gráfico, independente de dispositivo, contendo muitas das funções que deveriam ser implementadas dentro destas ferramentas. Para adaptar este pacote ao hardware existente, foram projetadas interfaces padronizadas para acesso a dispositivos de saída gráfica [OLA 87] e de entrada de dados gráficos (objeto deste relatório). Estas interfaces podem ser também usadas para a construção de aplicativos que não precisem ser independentes de dispositivo.

As finalidades da interface de entrada são múltiplas :

a) fornecer um conjunto de funções básicas para acesso a dispositivos gráficos de entrada;

b) fornecer um ambiente padronizado para realização de entrada a partir de dispositivos gráficos, sejam eles mesa digitalizadora ou mouse;

c) permitir que um programa utilize diversos dispositivos gráficos, sem que isto signifique alterações em seu código-fonte;

d) facilitar o transporte de software entre instalações.

## 2. O QUE É "IE"

"IE" é um conjunto de funções que possibilitam ao pacote gráfico (ou aplicativo) o acesso aos dados fornecidos por dispositivos de entrada gráfica (mouse e mesa digitalizadora). Tais dados podem ser coordenadas de um ponto indicado pelo usuário através de um "cursor" ou o estado de um botão localizado sobre este cursor. O pacote trata o dispositivo de entrada de forma padronizada, conforme descrito no capítulo 3.

Situações de erro podem ser detectadas pelo programa-usuário da IE através de um código retornado pelas funções.

As funções da IE são classificadas em

- a) controle : permitem a inicialização do dispositivo;
- b) entrada : permitem a obtenção de dados fornecidos pelo usuário através do dispositivo;
- c) interrogação : permitem ao pacote a obtenção de dados dependentes de dispositivo.



### 3. DISPOSITIVO PADRÃO DE ENTRADA

As funções da IE emulam o acesso a um dispositivo virtual para entrada de dados gráficos, definido a fim de permitir que mesa digitalizadora e mouse sejam manipulados da mesma forma.

Este dispositivo caracteriza-se pelos seguintes aspectos :

a) fornece coordenadas positivas de pontos localizados em seu sistema de referência. A origem e limites deste sistema podem ser conhecidos pelo pacote através das funções de interrogação. A origem do sistema de coordenadas é indicada pelo usuário, sob comando do pacote gráfico. A posição desta sobre a área de movimentação do cursor é determinada pelo usuário, que pode escolhê-la conforme sua conveniência;

b) o usuário pode apontar determinado ponto através da movimentação de um cursor ligado ao dispositivo de entrada. O reflexo de tal movimento sobre a tela gráfica do sistema é responsabilidade do pacote gráfico; à IE cabe apenas a tarefa de fornecer a ele a posição corrente do cursor na área de trabalho (ver figura 3.1). As coordenadas nos eixos x e y sempre são positivas. Se o usuário levar o cursor para fora da área de trabalho, o dispositivo padrão considera as coordenadas limites (por exemplo, coordenadas negativas são transformadas em zero);

c) o usuário pode ainda pressionar (ou não) um botão existente sobre o cursor. O estado deste pode ser obtido pelo pacote gráfico, que usará este dado da forma mais conveniente.

A IE fornece um conjunto de funções para cada dispositivo gráfico. A interface e funções são idênticas em todas as versões. Em tempo de ligação, o programador deve escolher um conjunto ou outro de acordo com o dispositivo que será utilizado para operação do sistema.



Figura 3.1 - Dispositivo padrão de entrada gráfica.

## 4. FUNÇÕES DA IE

O nome das funções inicia por "ie" e é seguido de um string que indica o procedimento por elas executado. Podem receber parâmetros e retornar valores, conforme apresentado no capítulo 6.

### 4.1 Controle

`ie01_inic` - inicializa o hardware do dispositivo de entrada gráfica.

### 4.2 Entrada

`ie10_lexybotao` - obtém as coordenadas atuais do cursor e estado do botão.

### 4.3 Interrogação

`ie20_coordisp` - obtém limites de valores para coordenadas do sistema de referência do dispositivo.

## 5. COMO USAR A IE EM UM PROGRAMA APLICATIVO

O programa deve ser codificado na linguagem C (Microsoft versão 4.0 ou TURBO C). Fica a cargo do programador a implantação de procedimentos especiais para compatibilização das funções com outras linguagens.

Os procedimentos do aplicativo (e operador) podem variar conforme o dispositivo de entrada gráfica utilizado, devido a características intrínsecas aos mesmos. De maneira geral, entretanto, a forma de tratamento de ambos os dispositivos pelo programa aplicativo é a mesma, conforme ilustrado pelo exemplo da seção 5.3.

Bibliotecas disponíveis :

IEMOUSMS.LIB : funções para mouse, Microsoft.  
IEMOUSTC.LIB : funções para mouse, Turbo C.  
IEMESAMS.LIB : funções para mesa digitalizadora,  
Microsoft.  
IEMESATC.LIB : funções para mesa digitalizadora,  
Turbo C.

Apenas uma destas bibliotecas deve ser ligada ao aplicativo.

EX : LINK EXEMPLO, EXEMPLO,,IEMOUSEMS ....

### 5.1 Uso de mouse através da IE

Para utilizar o mouse como dispositivo de entrada, é necessário determinar a origem de seu sistema de coordenadas. Para tanto, o aplicativo deve solicitar ao operador o posicionamento do cursor no canto inferior esquerdo da área que ele pretende usar para movimento. Esta posição será o referencial para determinação das coordenadas do cursor.

A seguir, o aplicativo deve executar a função "ie01\_inic", fazendo com que este referencial seja armazenado pelo controlador do mouse. Para que esta seja executada corretamente, o arquivo "INMOUSE.COM" deve estar presente no diretório corrente do sistema.

A partir daí, o estado do cursor (posição + botão) pode ser obtido através da função "ie10\_lexybotao". Estando qualquer um dos botões do mouse pressionado, o botão do dispositivo de

entrada é considerado "pressionado".

Se o aplicativo necessita trabalhar com outros sistemas de coordenadas além daquele do mouse, pode obter os limites deste através da função "ie20\_coordisp" e calcular o fator de escala correspondente.

## 5.2 Uso de mesa digitalizadora através da IE

Inicialmente, a origem do sistema de referência da mesa digitalizadora deve ser determinada. O aplicativo deve pedir ao usuário que posicione o cursor no canto inferior esquerdo da área a ser usada e pressione o botão de "reset". Esta posição será o referencial (0,0) para a determinação das coordenadas do cursor.

A seguir, o usuário deve colocar a mesa em modo de operação "contínuo", para que esta constantemente envie ao hospedeiro as coordenadas do cursor.

O aplicativo deve, então, executar a função "ie01\_inic", que programa a porta de comunicações para ligação adequada à mesa. Para que esta função seja executada corretamente, o arquivo "MODE.EXE" deve estar presente no diretório corrente do sistema.

A partir deste instante, a mesa pode ser usada de forma idêntica ao mouse.

### 5.3 Exemplo

```
/*-----*/
/* EXEMPLO DE USO DA IE COM DISPOSITIVOS GRAFICOS */
/*-----*/

#include "istipos.h"

static int maix, maiy;

main()
(
  int x, y, botao;
  int xt, yt;
  int mxmin, mymin, mxmax, mymax;
  int txmin, tymin, txmax, tymax;
  float ex, ey;                                     /* fator de escala */

                                           /* inicializa origem do dispositivo */

  printf("\n\n\n");
  printf("Posicione cursor no canto inferior esquerdo da area");
  printf(" de trabalho\nRESET no dispositivo");
  printf("\nPressione <return>\n");
  getch();

  if ( ie01_inic() )
  (
    printf("\nDispositivo nao esta´ instalado\n");
    return;
  )

  /* inicializa disp de saida e calcula fator de escala */

  is20_inic();
  is30_coordisp( &txmin, &txmax, &tymax, &tymin);
  is09_posic(txmin,tymin);
  is24_retang(txmax,tymax);

  maix = ( txmin > txmax ) ? txmin : txmax;
  maiy = ( tymin > tymax ) ? tymin : tymax;
  ie20_coordisp( &mxmin, &mxmax, &mymin, &mymax );
  ex = (float) (txmax - txmin) / (mxmax - mxmin);
  ey = (float) (tymax - tymin) / (mymax - mymin);
```

```
/* movimenta uma cruz sobre a tela ate que o usuario pressione o
   botao do dispositivo de entrada */
```

```
is01_setcorlin(INVERSO);
xt = txmin;
yt = tymin;
cursor(xt,yt);
do {
    ie10_lexybotao( &x, &y, &botao );
    x = (x - mxmin) * ex + txmin;
    y = (y - mymin) * ey + tymin;
    cursor(xt,yt);
    cursor(x,y);
    xt = x;
    yt = y;
}
while ( !botao );

is28_fim();
}
```

```
/* desenha cursor */
```

```
cursor ( x, y )
int x,y;
{
    if ( x-3 >= 0 && x+3 <= maix && y-3 >= 0 && y+3 <= maiy )
        {
            is09_posic(x-3, y);
            is22_linhah(x+3);
            is09_posic(x, y-3);
            is23_linhav(y+3);
        }
}
```

## 6. MANUAL DE REFERENCIA

A seguir serão apresentadas informações relevantes para utilização das rotinas da IE, organizadas sob os seguintes tópicos:

a) protótipo da função, indicando tipo do valor retornado, nome e tipos dos parâmetros recebidos;

b) breve descrição do funcionamento e valores retornados.

### ie01\_inic

#### RESUMO

```
int ie01_inic ()
```

#### DESCRIÇÃO

Inicializa o hardware do dispositivo de entrada gráfica.

Pode retornar:

0 : inicialização OK.

1 : erro na inicialização (não instalado)

Dependendo do tipo de dispositivo efetivamente conectado, esta rotina pode exigir a presença de um arquivo especial no diretório corrente do sistema (ver capítulo 5).

## ie10\_lexybotao

### RESUMO

```
ie10_lexybotao ( x, y, botao )  
int *x, *y;  
int *botao;
```

### DESCRIÇÃO

Retorna nas variáveis recebidas como parâmetro a posição atual do cursor e o estado do botão (ver capítulo 3) :

x : posição no eixo x  
y : posição no eixo y  
botao : estado do botão  
0 : não pressionado  
1 : pressionado

## ie20\_coordisp

### RESUMO

```
ie20_coordisp ( xmin, xmax, ymin, ymax )  
int *xmin, *xmax,  
    *ymin, *ymax;
```

### DESCRIÇÃO

Retorna nas variáveis recebidas como parâmetro os valores máximos e mínimos do sistema de coordenadas do dispositivo. (xmin,ymin) corresponde ao canto inferior esquerdo da área usada para movimentação do cursor.

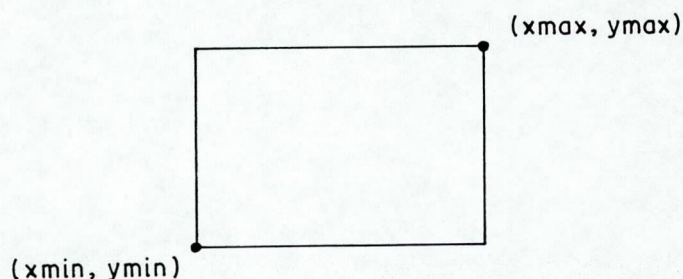


Figura 6.1 - Limites de coordenadas.



## 7. IMPLEMENTAÇÃO DA IE

As funções da IE são escritas na linguagem C e são organizadas em dois arquivos :

IEMOUSE.C - contém funções para mouse.

IEMESA.C - contém funções para mesa digitalizadora.

### 7.1 IEMOUSE

A obtenção de dados do mouse é feita através do BIOS, interrupção número 51, utilizando a função "int86" da biblioteca do compilador C. Através desta interrupção é possível executar funções que acessam o hardware do mouse. Estas são fornecidas pelo fabricante do equipamento e devem ser carregadas na memória através de procedimentos descritos na seção 5.1. Recebem parâmetros e retornam valores através de registradores, carregados conforme o conteúdo da estrutura de dados recebida pela função "int86" [MIC 85].

O código da função a ser executada é indicado no registrador AX e pode ser um dos seguintes valores [FIG 85]:

- 0 : inicializa hardware do mouse.  
Retorna em AX o estado :
  - 0 : mouse não está instalado.
  - 1 : inicialização OK.
- 1 : habilita exibição automática do cursor (não usado pela IE).
- 2 : inibir exibição do cursor.
- 3 : ler posição e estado do botão do mouse.  
Retorna :
  - BX : botão que está pressionado
    - 0 : nenhum
    - 1 : esquerdo
    - 2 : direito
    - 3 : ambos
  - CX : posição em x
  - DX : posição em y
- 4 : inicializar sistema de referência do mouse, indicando coordenadas da posição atual do cursor. Parâmetros :
  - CX : posição em x
  - DX : posição em y

- 7 : determinar limites de coordenadas na horizontal (eixo x). Parâmetros :  
CX : valor mínimo  
DX : valor máximo
- 8 : determinar limites de coordenadas na vertical (eixo y). Parâmetros :  
CX : valor mínimo  
DX : valor máximo
- 9 : determinar a forma do cursor automático (não usado pela IE).

As funções que manipulam este dispositivo são :

a) ie01\_inic : carrega rotinas do BIOS na memória (INMOUSE), inicializa hardware do mouse, determina limites do sistema de referência do mouse e indica que a origem é a posição atual do cursor;

b) ie10\_lexybotao : lê coordenadas da posição atual e estado do cursor através do BIOS e atribui os valores às variáveis recebidas como parâmetro;

c) ie20\_coordisp : indica que o canto inferior esquerdo da área de movimentação do cursor tem coordenadas (0,400) (xmin,ymin). A origem do sistema de referência do mouse é, na verdade, o canto superior esquerdo da referida área.

## 7.2 IEMESA

A obtenção dos dados fornecidos pela mesa digitalizadora PD 1060 [STI 85] é feita através de acesso direto aos registradores da porta de comunicação serial e assíncrona (COM1). Tais registradores são acessados através das funções "inp" e "outp" da biblioteca do compilador [MIC 85].

A inicialização da porta de comunicações é feita pelo comando "MODE" do sistema operacional, com os seguintes parâmetros :

```
MODE COM1:4800,n,8,2
```

```
velocidade = 4800 bps  
8 bits de dados  
2 stop bits  
sem paridade
```

Para obtenção dos caracteres recebidos pela porta serial são acessados dois registradores, mapeados nos seguintes endereços de entrada/saída :

- 0x3FD (registrador de estado) : deve ser lido para verificar se a porta recebeu algum caractere. O bit 0 contém :

0 : nada foi recebido

1 : recebeu um byte

- 0x3FB (registrador de dados) : deve ser lido para obtenção do caractere recebido pela porta.

Os caracteres lidos da interface serial são interpretados conforme o formato descrito em [STI 85].

As funções que manipulam este dispositivo são :

a) ie01\_inic : apenas executa o comando "MODE";

b) ie10\_lexybotao : sincroniza a leitura (espera LF), lê um string, cujo formato é descrito em [STI 85] (código do botão + coordenadas em x e y) e converte dados para binário;

c) ie20\_coordisp : indica que o canto inferior esquerdo da área de movimentação do cursor tem coordenadas (0,0) (xmin, ymin).

## REFERENCIAS BIBLIOGRAFICAS

- [FIG 85] FIGUEIRAS, J. Turbo PASCAL drives the mouse. BYTE, Peterborough, 10(9):161-8, Sept. 1985.
- [MIC 85] MICROSOFT. C Compiler: Run-Time Library Reference. 1985.
- [OLA 87] OLABARRIAGA S., PINHO M. & COMBA J. Interface de saída com dispositivos gráficos. Porto Alegre, CPGCC da UFRGS, setembro 1987. (RP no. 079)
- [STI 85] STI. PD 1060 - Mesa digitalizadora - manual de instalação e operação. Porto Alegre, 1985.
- [WAG 86] WAGNER, F.R. et al. Ambiente integrado para Projeto de Sistemas Digitais Auxiliado por Computador. In: Congresso Nacional de Informática, XIX. Rio de Janeiro, 18-25 Agosto 1986. Anais. SUCESU, Rio de Janeiro, 1986. pp 111-116, vol.2.
- [WAG 87a] WAGNER, F.R., C.M.D.S.-FREITAS e L.G.GOLENDZINER. Linguagens de Descrição de Hardware para Suporte à Integração do Processo de Projeto em AMPLO. Porto Alegre, PGCC da UFRGS, março 1987. (RP no. 065)
- [WAG 87b] WAGNER, F.R., C.M.D.S.-FREITAS e L.G.GOLENDZINER. A Digital Systems Design Methodology based on Nets of Agencies. In: BARBACCI, M. e C.J.KOOMEN (eds.) CHDL'87. Amsterdam, North-Holland, 1987. pp. 213-224.

## ANEXO: IMPLEMENTAÇÃO DA IE PARA MOUSE E TECLADO

Autores: Bernardo Copstein e Silvia D. Olabarriaga

### 1. Introdução

Dentre as finalidades da IE original destacava-se o seguinte: "Fornecer um ambiente padronizado para realização de entrada a partir de dispositivos gráficos, sejam eles mesa digitalizadora ou mouse".

A realidade, porém, mostrou que seria interessante se pudessemos utilizar o teclado como dispositivo de entrada gráfica. Este uso se justificaria de duas formas:

a) A grande maioria dos equipamentos não dispõe de mouse ou mesa digitalizadora ( devido ao alto custo desses equipamentos ) mas todos dispõe de teclado.

b) O teclado é um dispositivo que pode ser mais preciso no caso de termos de fazer ajustes finos, uma vez que o deslocamento do cursor a cada vez que pressionamos uma tecla é perfeitamente conhecido e controlável.

Seria ideal, então, que a IE nos fornecesse um ambiente padronizado para realização de entrada a partir de dispositivos gráficos, fossem eles mesa digitalizadora, mouse ou teclado.

Esbarramos, porém, em um problema de ordem técnica. A mesa digitalizadora é um dispositivo que nos devolve coordenadas absolutas, e o teclado nos devolve deslocamentos, o que inviabiliza sua utilização conjunta. O mouse, por outro lado, é um dispositivo mais flexível e pode ser adaptado às duas filosofias.

Em face do apresentado, desenvolveu-se uma nova versão da IE, que nos oferece um ambiente padronizado para trabalharmos com dispositivos de entrada, sejam eles mouse ou teclado ( ou ainda os dois simultaneamente ). Uma vez que as duas versões da IE estão disponíveis, o usuário deverá optar por aquela que melhor se adapte as suas necessidades.

### 2. Funcionamento do teclado em conjunto com o mouse

A compatibilização entre o teclado e o mouse no que diz respeito ao significado das teclas, foi definida da seguinte forma:

a) As teclas do teclado numérico reduzido ( que geralmente possuem setas de direção desenhadas sobre si ) correspondem ao ato de deslocarmos o mouse. Cada vez que pressionamos uma destas teclas as coordenadas do cursor

serão incrementadas ou decrementadas da mesma forma que o seriam se o mouse tivesse sido deslocado na direção indicada na seta.

b) As teclas <+> e <-> não possuem correspondência física com o mouse. Elas servem para aumentar (<+>) ou diminuir (<->) o deslocamento que será aplicado as coordenadas do cursor cada vez que uma tecla de direção tiver sido pressionada. Na verdade essas duas teclas servem para simular a velocidade com que deslocamos o mouse sobre a mesa. Ao teclarmos <+> o valor do incremento será adicionado de cinco unidades. Ao teclarmos <-> o valor do incremento será diminuído de cinco unidades. O incremento tem um valor mínimo inicial de um.

c) A tecla <barra de espaço> simula o botão do mouse. Ao pressionarmos a <barra de espaço> a rotina interpretará exatamente da mesma forma que faria se tivesse sido pressionado um dos botões do mouse.

### 3. Funções

A seguir temos a relação das rotinas oferecidas pela nova versão da IE com suas respectivas alterações quando existirem.

ie01\_inic

RESUMO

```
#include "iedefs.h"
```

```
int ie01_inic( mouse )  
int mouse;
```

DESCRIÇÃO

Inicializa o hardware dos dispositivos de entrada gráfica. O parâmetro "mouse" deverá indicar se existe ( mouse=1 ) ou não ( mouse = 0 ) mouse conectado ao sistema.

VALORES RETORNADOS

ZERO - mouse e teclado instalados corretamente.  
UM - apenas o teclado instalado corretamente.

ie02\_redfm

#### RESUMO

```
ie02_redfm( x,y )
int x,y;
```

#### DESCRIÇÃO

Indica para o BIOS que a posição corrente do mouse é (x,y). Esta rotina é usada para compatibilizar as coordenadas indicadas via teclado com o mouse.

ie10\_lexybotao

#### RESUMO

```
#include "iedefs.h"

ie10_lexybotao( x,y,botao )
int *x, *y;
int *botao;
```

#### DESCRIÇÃO

Esta rotina permite a obtenção de coordenadas x,y e estado de botão do dispositivo de entrada gráfico (mouse ou teclado). A prioridade de amostragem é dada ao teclado, sendo o mouse lido apenas quando não estiver sendo pressionada tecla alguma. Se o usuário pressiona uma tecla diferente daquelas que permitem o movimento do cursor (setas, <+>, <-> e <barra de espaço>), esta será devolvida ao programa chamador sem interpretação. A seguir, o algoritmo detalhado da rotina é apresentado:

```
se o mouse está ativo
então olha o teclado
    se tem tecla
```

```
    <tecla de seta> - altera posição atual do
                    cursor ( ver cap. 2 item
                    a ).
```

```
    < + > ou < - > - altera o incremento usado
                    para alterar a posição do
                    cursor no caso de <tecla
                    de seta> ( ver cap. 2
                    item b ).
```

```
    <return>      - simula pressionamento do
                    botão do mouse ( ver cap. 2
                    item c ).
```

```

        <outra tecla> - retorna o código da tecla
                      ( SCAN ou ASCII ) no para-
                      metro "botão".
senão
    lê o mouse e atualiza "x", "y" e "botão".
senão olha o teclado
    se tem tecla
    então
        testa as teclas da mesma forma que quando o
        mouse está ativo
senão
    retorna "nenhuma tecla"
Fim

```

#### VALORES RETORNADOS

A rotina `ie10_lexybotao` retorna através de sua chamada um valor que permite identificar a informação disponível no parametro "botão". Para melhor clareza foram definidas constantes para cada um destes valores. Estas constantes são declaradas no arquivo "IEDEFS.h".

Obs: em todas as situações, os parâmetros "x" e "y" possuem a coordenada atualizada do cursor.

**KMOUSE** - este valor é retornado no caso de ter sido pressionada uma tecla de seta, as teclas <+> e <->, <barra de espaço> ou ter sido utilizado o mouse.

**COMUM** - este valor é retornado se foi pressionada uma tecla qualquer e indica que o parametro botão possui o código ASCII da tecla pressionada.

**ESPECIAL** - este valor é retornado se foi pressionada uma tecla especial ( F1 a F10, INS, DEL, etc ) e indica que o parametro botão possui o SCAN-CODE da tecla.

**NENHUMA\_TECLA** - este valor é retornado quando o mouse não estiver ativo e não for pressionada nenhuma tecla.

`is20_coordisp`

#### RESUMO

```

is20_coordisp(· x,y )
int *x, *y;

```

#### DESCRIÇÃO

Não foram feitas alterações nesta rotina.



#### 4. Arquivo

As rotinas estão armazenadas no arquivo "IEKMLGTC.LIB" que deve ser ligado ao aplicativo conforme descrito em documentação anterior.