



**3-я Международная конференция
"Крым 96"**



**Third International Conference
"Crimea 96"**

Конференция проводится в рамках
мероприятий ИФЛА 1996 г.

The conferences is held within the framework
of IFLA 1996 activities

**Библиотеки и ассоциации
в меняющемся мире:
НОВЫЕ ТЕХНОЛОГИИ
и новые формы сотрудничества**

***Libraries and Associations
in the Transient World:
New Technologies
and New Forms of Cooperation***

**Материалы конференции
*Conference Proceedings***

**Том 2
Volume 2**

**Форос, Ялта, Автономная Республика Крым, Украина
1-9 июня 1996**

**Foros, Yalta, Autonomous Republic of Crimea, Ukraine
1-9 June 1996**

Разработка и использование драйвера манипулятора "мышь" для программ, написанных на ISIS-Pascal

Development and use of a driver of a manipulator a mouse for the programs, written on ISIS-Pascal

*Баженов И.С., Баженов С.Р., Мазов Н.А.
Государственная публичная научно-техническая библиотека
Сибирского отделения Российской Академии наук, Новосибирск, Россия*

*Bazhenov I.S., Bazhenov S.R., Mazov N.A.
Russian Academy of Sciences, Siberian Branch, State Public Scientific-Technological Library, Novosibirsk, Russia*

A driver of a mouse developed for use in programs, written on ISIS-Pascal is described. Programming of all functions of a mouse is provided with the help of it. Procedures of using the driver on ISIS-Pascal are also indicated.

В последнее время манипулятор "мышь" занял среди устройств ввода едва ли не такое же место, как клавиатура. Практически все современное программное обеспечение позволяет использовать при работе манипулятор "мышь". "Мышь" существенно облегчает работу в тех системах, где присутствуют многоуровневые меню, окна, а также места на экране, при нажатии мышью на которые происходит какое-либо событие, т.е. кнопки, пиктограммы и т.д.

Поддержка "мыши" реализована и во многих СУБД, где, кроме использования "мыши" при работе с базами данных необходимо дать возможность программисту использовать мышь в своих программах (если СУБД имеет свой язык программирования). К сожалению, в CDS/ISIS поддержки "мыши" нет как при работе с базами данных, так и в ISIS-Pascal, поэтому для программиста не представляется возможность программировать в своей программе "мышь".

Для решения этой задачи нами был разработан драйвер, который в совокупности с основным драйвером "мыши" позволяет программисту довольно простыми способами реализовать в своей программе ее поддержку.

Драйвер представляет из себя резидентную программу (TSR), которая дублирует некоторые функции основного драйвера "мыши", но вызов этих функций и получение возвращаемых значений происходит не через прерывания и регистры процессора, к которым в ISIS-Pascal не доступиться, а через процедуры ISIS-Pascal: Write и KdbKey.

Драйвер называется "isismous.com" (далее драйвер) и вызывается перед самым запуском программы ISIS, после чего остается в памяти и находится там до конца работы ISIS. После окончания выполнения программы необходимо выгрузить драйвер из памяти командой "isismous.com /U", чтобы драйвер не занимал память и чтобы не было конфликтов с другими программами.

Если до вызова драйвера не был загружен основной драйвер "мыши", то драйвер не загрузится и выдаст соответствующее сообщение.

Ниже приводятся основные функции драйвера:

Драйвер позволяет отследить следующие события "мыши": 1. Нажатие левой кнопки 2. Отпускание левой кнопки 3. Двойной клик левой кнопкой 4. Перемещение мыши при нажатой левой кнопке 5. Нажатие правой кнопки 6. Отпускание правой кнопки 7. Двойной клик правой кнопкой 8. Перемещение мыши при нажатой правой кнопки.

Также драйвер может принимать некоторые команды по управлению: 1. Определить нахождение драйвера в памяти. 2. Отключить драйвер. 3. Включить драйвер. 4. Сделать курсор мыши невидимым. 5. Сделать курсор мыши видимым. 6. Отключить отслеживание передвижений мыши при нажатой кнопке. 7. Включить отслеживание передвижений мыши при нажатой кнопке.

Обращение к драйверу, т.е. передача определенной команды реализуется посредством следующей команды ISIS-Pascal: Write(Chr(255);'x'), где первый символ — символ с десятичным кодом 255, а второй — код команды (будет описан ниже).

Получение информации от драйвера реализуется посредством функции ISIS-Pascal KdbKey: ScanCode:=KdbKey(SymbolCode). Информация передается после каждого события "мыши", описанного выше.

Для удобства использования программирования драйвера написан ряд основных процедур для работы с манипулятором "мышь".

Рассмотрим работу с драйвером (передачу и получение информации) на примере учебной программы, написанной на ISIS-Pascal. Для удобства программа разделена на отдельные блоки (процедуры), каждый из которых отдельно описан. Все комментарии к блокам написаны с учетом синтаксиса комментария ISIS-Pascal.

Блок № 1. Описание необходимых переменных.

```
Program TestMouse; { * Название программы *}
var MouseInst: real; { * Флаг установки драйвера: *} { * 0 — драйвер не установлен *} { * 1 — драйвер установлен *}
MouseFunc: real; { * Код функции, полученный от драйвера *} MouseX: real; { * Текущая координата X курсора мыши *} MouseY: real; { * Текущая координата Y курсора мыши *} KeybC: real; { * Числовой код клавиши *} KeybS: string; { * Символьный код клавиши *}
```

Блок № 2. Процедура, определяющая наличие драйвера в памяти.

Комментарий: Процедура необходима для проверки нахождения в памяти драйвера, т.к., если его в памяти нет, то в связи с тем, что обращение к драйверу происходит с помощью процедуры Write, на экран будут написаны коды обращения к драйверу.

После передачи кода операции должен вернуться десятичный код 41, читаемый функцией KbdKey. Если этот код получен, то драйвер находится в памяти.

```
procedure AskMouse; var KeybC: real;
KeybS: string; begin
while KeyPress = 1 do { * -Т— Очистить клавиатуру *} KeybC:=KbdKey(KeybS); { * — *}
MouseInst:=0; { * Очистить флаг присутствия мыши *} Cursor(26,1); { * Написать за экран *}
Write(Chr(255)|Chr(255)); { * ... код операции *} if KeyPress = 1 then { * Если в буфере появился код, то *}
begin KeybC:=KbdKey(KeybS); { * ... получим его *} if KeybC = 41 then { * Если код = 41, то *}
MouseInst:=1; { * ... драйвер установлен *} end;
end;
```

Блок № 3. Процедура, отключающая драйвер.

Комментарий: Отключать драйвер (т.е., чтобы он не передавал информацию о событиях мыши) необходимо на время исполнения длинной операции, во время которой не происходит проверка на нажатие клавиши (т.е. не вызывается функция KbdKey). Если драйвер не отключить, то при работе в это время с мышью может переполниться клавиатурный буфер, что отрицательно повлияет на работу драйвера и программы.

После отключения драйвера курсор мыши остается таким же, каким он был до отключения.

После загрузки драйвера в память он отключен.

```
procedure MouseOff; begin
if MouseInst = 1 then { * Если драйвер есть в памяти, то *} Write(Chr(255))"4"); { * ... отключить его *}
end;
```

Блок № 4. Процедура, включающая драйвер.

Комментарий: После включения драйвера курсор мыши остается таким же, каким он был до включения.

```
procedure MouseOn; begin
if MouseInst = 1 then { * Если драйвер есть в памяти, то *} Write(Chr(255))"3"); { * ... включить его *}
end;
```

Блок № 5. Процедура, делающая курсор невидимым.

Комментарий: Делать курсор невидимым необходимо на время работы с экраном через видеопамять (через видеопамять работают следующие процедуры ISIS-Pascal: SaveScr, Page, ChAttr, Attr). Если курсор не сделать невидимым, то на экране будут образовываться некрасивые дырки.

После загрузки драйвера в память курсор невидим.

```

procedure HideMouse; begin
if MouseInst = 1 then { * Если драйвер есть в памяти, то * } Write(Chr(255)|"0"); { * ... сделать курсор невидимым * }
end;

```

Блок № 6. Процедура, делающая курсор видимым.

```

procedure ShowMouse; begin
if MouseInst = 1 then { * Если драйвер есть в памяти, то * } Write(Chr(255)|"1"); { * ... сделать курсор видимым * }
end;

```

Блок № 7. Процедура, отключающая реакцию драйвера на передвижение мыши при нажатой кнопке

Комментарий: При передвижении "мыши" при нажатой кнопке в буфер клавиатуры засылаются коды, которые должны считываться функцией KbdKey. Если в момент передвижения мыши при нажатой кнопке не происходит вызов функции KbdKey, то может переполниться клавиатурный буфер, что отрицательно повлияет на работу драйвера и программы.

Чтобы этого избежать необходимо использовать нижеследующую функцию.

```

procedure MoveOff; begin
if MouseInst = 1 then Write(Chr(255)|"5"); end;

```

Блок № 8. Процедура, включающая реакцию драйвера на передвижение мыши при нажатой кнопке

```

procedure MoveOn; begin
if MouseInst = 1 then Write(Chr(255)|"6"); end;

```

Блок № 9. Процедура, включающая реакцию драйвера на передвижение мыши при нажатой кнопке

Комментарий: Процедура предназначена для получения от драйвера следующих данных:

- код операции мыши;
- текущая координата X;
- текущая координата Y.

Все данные передаются в символьном виде. В процедуре они преобразуются в числовое значение, указанное ниже.

Коды событий мыши (числовое значение): 10 — Нажата левая кнопка; 11 — Отпущена левая кнопка; 12 — Двойной клик левой кнопкой; 13 — Перемещение мыши при нажатой левой кнопке; 20 — Нажата правая кнопка; 21 — Отпущена правая кнопка; 22 — Двойной клик правой кнопкой; 23 — Перемещение мыши при нажатой правой кнопке.

```

procedure GetMouse; var KeybC: real;
KeybS: string; begin
MouseFunc:=Ord(KeybS); { * Получить код операции мыши * } KeybC:=KbdKey(KeybS); { * -> * }
MouseX:=Ord(KeybS)-32; { * -+ Получить координату X * } KeybC:=KbdKey(KeybS); { * -> * }
MouseY:=Ord(KeybS)-32; { * -+ Получить координату Y * }
end;

```

Блок № 10. Программа, показывающая пример использования процедур.

```

begin
AskMouse: { * Узнать, установлен ли драйвер * } ShowMouse; { * Сделать курсор мыши видимым * }

{ ** Какие-нибудь действия ***** } { ..... }

{ ***** } MouseOn; { * Включить драйвер ** }
MoveOn: { * Включить отслеживание передвижения мыши * }
{ * при нажатой кнопке * } repeat

```

```

KeybC:=KbdKey(KeybS); { * Считать код из клавиатурного буфера *} if KeybC = 41 then { * Если код = 41
то *}
begin { * Обработка событий мыши *} GetMouse; { * Получить код операции и *}
{ * координаты мыши *} { .....} { * Действия с полученными кодами *}
end; until KeybC = 1; { ESC } MoveOff; { * Выключить отслеживание передвижения мыши *}
{ * при нажатой кнопке *} MouseOff; { * Выключить драйвер *}

{*****} HideMouse; { * Сделать
курсор мыши невидимым *}
end. Как видно из данного примера, использование драйвера isismouse.com
в программах на ISIS-Pascal достаточно просто и эффективно.
*****

```