

1.4. Использование ловушек

... И тут Пух все понял. Они с Пятачком попались в Хитрую Ловушку для Пухов, которую в отместку им вырыл Слонопотам!...

—Отрывок из неопубликованной главы повести А.Милн «Винни-Пух и все-все-все»

Вы, конечно, помните, что программы, выполняющие всю «черную» работу, «защиты» в ROM, поэтому изменить их невозможно. Однако, в некоторых важных подпрограммах ROM предусмотрены своеобразные «перехваты» — ловушки («hooks»).

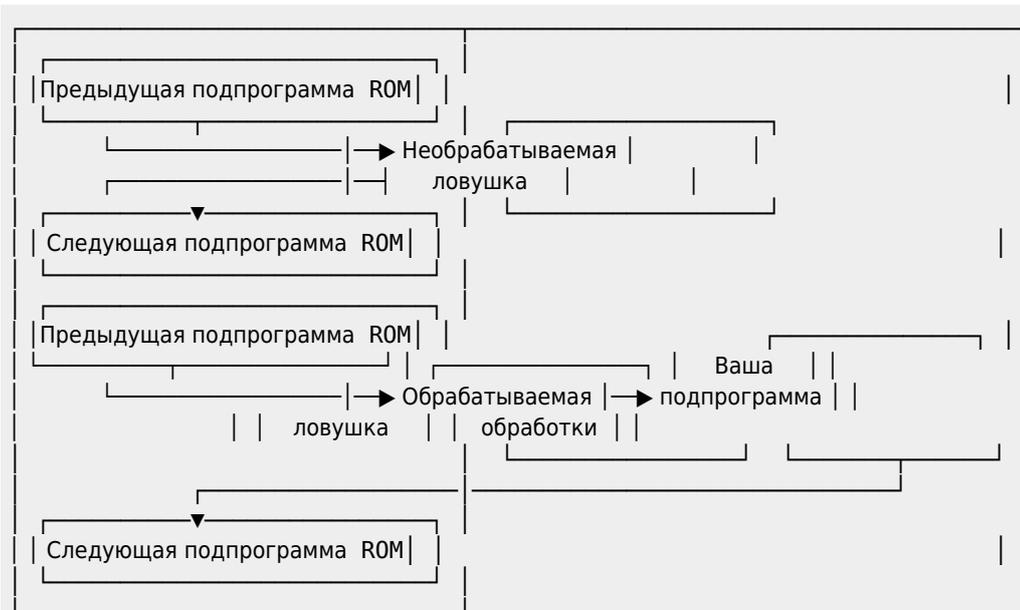
Вызов подпрограммы-ловушки организуется в ROM следующим образом:

```
PUSH HL      ; Сохранение содержимого
PUSH DE      ; регистров микропроцессора
PUSH BC      ; в стеке
PUSH AF      ;
CALL адрес ; Вызов подпрограммы-ловушки по заданному адресу
POP AF       ; Восстановление значения
POP BC       ; регистров микропроцессора
POP DE       ; для дальнейших операций
POP HL       ;
```

При включении компьютера подпрограмма-ловушка содержит только команду возврата (RET). Вы можете заменить эту команду на команду перехода (JP) к подпрограмме, которая написана Вами на машинном языке.

В этом случае о ловушке говорят, что она *обрабатывается*.

Таким образом, Вы можете косвенно изменять ROM и вводить *новые* операторы! Приведенный ниже рисунок показывает, что означает термин «обработать» ловушку.



Пример 1. Отмена действия клавиши `STOP`.

При выполнении программы на языке **MSX BASIC** клавиатура постоянно опрашивается (сканируется). Нажатие клавиши порождает код, который записывается в буфер клавиатуры (начинающийся с адреса &HF55E в рабочей


```

20 ON STOP GOSUB 100:STOP ON
30 PRINT A:A=A+1:GOTO 30
100 RETURN

```

Нажатие клавиши **STOP** теперь не приводит к прекращению программы и выводу на экран курсора. Поскольку клавиша **STOP** превращена в **CTRL+STOP**, то клавиша **STOP** «перестает работать».

Отметим, что нажатие сложной комбинации клавиш **CTRL+SHIFT+GRAPH+PUC** прекращает работу этой программы.

Пример 2. При нажатии функциональной клавиши F4 печатается слово «Миша».

[1040-02.bas](#)

 [1040-02.bas](#)

```

10 CLEAR 200,&HD000
20 POKE &HFDCC,&HC3 ':\
30 POKE &HFDCD,&H0 ': > JP D000
40 POKE &HFDCE,&HD0 ':/
50 AD=&HD000
60 READ A$:IF A$="Z" THEN END
70 POKE AD,VAL("&H"+A$)
80 AD=AD+1:GOTO 60
90 DATA FE,38      :' CP 38h ; A:=38h ?
100 DATA C0       :' RET NZ ; Символ "!=" означает "содержит"
110 DATA 3E,ED    :' LD A,EDh ; A:=EDh → "М"
120 DATA CD,A2,00 :' CALL 00A2 ;
130 DATA 3E,C9    :' LD A,C9h ; A:=C9h → "и"
140 DATA CD,A2,00 :' CALL 00A2 ;
150 DATA 3E,DB    :' LD A,DBh ; A:=DBh → "ш"
160 DATA CD,A2,00 :' CALL 00A2 ;
170 DATA 3E,C1    :' LD A,C1h ; A:=C1h → "а"
180 DATA CD,A2,00 :' CALL 00A2 ;
190 DATA 3E,40    :' LD A,40h ; A:=40h
200 DATA C9       :' RET ;
210 DATA "Z"      :' К о н е ц ;

```

Пример 3. Команда RUN приводит к появлению на экране слова «Миша».

[1040-03.bas](#)

 [1040-03.bas](#)

```

10 CLEAR 200,&HD000
20 POKE &HFECB,&HC3 ':\
30 POKE &HFECB,&H0 ': > JP D000
40 POKE &HFECB,&HD0 ':/
50 AD=&HD000
60 READ A$:IF A$="Z" THEN END
70 POKE AD,VAL("&H"+A$)
80 AD=AD+1:GOTO 60
90 DATA 3E,ED    :' LD A,EDh ; A:=EDh → "М"
100 DATA CD,A2,00 :' CALL 00A2 ;
110 DATA 3E,C9    :' LD A,C9h ; A:=C9h → "и"
120 DATA CD,A2,00 :' CALL 00A2 ;
130 DATA 3E,DB    :' LD A,DBh ; A:=DBh → "ш"
140 DATA CD,A2,00 :' CALL 00A2 ;
150 DATA 3E,C1    :' LD A,C1h ; A:=C1h → "а"
160 DATA CD,A2,00 :' CALL 00A2 ;
170 DATA C9       :' RET ;
180 DATA "Z"      :' К о н е ц ;

```

Пример 4. Программа позволяет на слух оценивать частоту обращений к ловушкам по маскируемым прерываниям. Во время очередного прерывания по таймеру издается звонок-сигнал. Таким образом ловушка «прозванивается».

[1040-04.asm](#)

```

.Z80

LD HL,OBJ          ; Организация псевдо-ROM
LD DE,04000H      ; в 1-й странице слота 3-2
LD BC,80H         ;
LDIR              ;
LD HL,START       ; Создание структуры страниц
LD DE,09000H      ; должно происходить в "верхних"
LD BC,OBJ-START   ; адресах
LDIR              ;
JP 9000H          ;
START: DI         ;
LD A,0FCH         ; 1-я страница - BASIC BIOS
OUT (0A8H),A      ; 2-я страница из слота 3-2
LD A,0A8H         ; 3-я страница из слота 3-2
LD (0FFFFH),A    ; 4-я страница из слота 3-2
EI               ;
LD HL,(4002H)     ;
JP (HL)           ; Переход на начало основной программы
OBJ:              ;
.PHASE 4000H      ;
DB 'AB'           ; Идентификатор ПЗУ
DW NACH           ; Адрес начала программы
DW 0              ; Зарезервированы 12 байтов
DW 0              ;
NACH: XOR A       ;
CALL 00C3H        ; Очистка экрана
CALL 00CCH        ; Выключение функциональных клавиш
DI               ;
IM 1              ; Установка 1-го режима прерываний:
LD A,0C3H         ; - сохраняется программный счетчик в стеке;
LD (0FD9AH),A    ; - переход по адресу 038H, где происходит
LD HL,WORK        ; обращение к адресам 0FD9AH и 0FD9FH
LD (0FD9BH),HL   ; с частотой 60 Гц.Любой из адресов можно
EI               ; использовать для обработки прерываний
LABEL: JR LABEL   ;
WORK:             ;
LD A,7            ; Это и есть подпрограмма, генерирующая
OUT (0A0H),A      ; звонок-сигнал!
LD A,10111110B   ;
OUT (0A1H),A      ;
LD A,8            ;
OUT (0A0H),A      ;
LD A,10H          ;
OUT (0A1H),A      ;
LD A,11           ;
OUT (0A0H),A      ;
LD A,100          ;
OUT (0A1H),A      ;
LD A,12           ;
OUT (0A0H),A      ;
LD A,0            ;
OUT (0A1H),A      ;
LD A,13           ;
OUT (0A0H),A      ;
LD A,1            ;
OUT (0A1H),A      ;
RET              ;

```

```
.DEPHASE      ;
END           ;
```

Пример 5. Перемещение спрайта по прерываниям

[1040-05.asm](#)

 [1040-05.asm](#)

;Некоторые макроопределения:

```
DISK MACRO      ;
  LD      BC,0100H ;
DSK:PUSH  BC      ;
  CALL   0FD9FH   ;
  POP    BC      ;
  LD     A,B      ; Гашение лампочки дисковода
  OR     C        ; для YIS805/128 KUBT 2
  RET    Z        ;
  DEC   BC        ;
  JR     DSK      ;
ENDM         ;
;
GASI MACRO     ;
  BCALL  041H     ; В ы к л ю ч е н и е  э к р а н а
ENDM         ;
;
GORI MACRO     ;
  BCALL  044H     ; В к л ю ч е н и е  э к р а н а
ENDM         ;
;
NETEND MACRO   ;
  RST   030H     ;
  DEFB  08FH     ; Отключение от с е т и (только для MSX-2)
  DEFW  04016H   ;
ENDM         ;
;
BCALL MACRO @X ;
  RST   030H     ;
  DEFB  0        ; Межслотовые вызовы подпрограмм
  DEFW  @X       ;
ENDM         ;
;
CLS MACRO     ;
  XOR   A        ; Очистка экрана
  BCALL 0C3H     ;
ENDM         ;
;
SCREEN2 MACRO  ;
  LD    HL,01800H ;
  LD    (0F3C7H),HL ;
  LD    HL,02000H ;
  LD    (0F3C9H),HL ;
  LD    HL,0      ; Инициализация и установка
  LD    (0F3CBH),HL ; 2-го графического режима
  LD    HL,01B00H ;
  LD    (0F3CDH),HL ;
  LD    HL,03800H ;
  LD    (0F3CFH),HL ;
  BCALL 072H     ;
ENDM         ;
;
COLOR MACRO   ;
  LD    A,01H    ;
  LD    HL,0F3E9H ;
```

```

LD    (HL),A    ;
INC   HL        ; Установка цвета
LD    (HL),A    ;
INC   HL        ;
LD    (HL),A    ;
BCALL 062H      ;
ENDM           ;

INITSPR MACRO  ;
BCALL 069H      ;
LD    BC,06201H ; Инициализация и "разрешение"
BCALL 047H      ; спрайтов размером 16 X 16
ENDM           ;

VIDEO MACRO    ;
SNOW:DI       ;
LD    A,E      ;
OUT   (99H),A  ; Заполнение области видеопамати, начинающейся
LD    A,D      ; с DE длиной в BC из области, начинающейся с HL
OR    40H      ;
OUT   (99H),A  ;
LD    A,(HL)   ;
OUT   (98H),A  ;
LD    A,B      ;
OR    C        ;
EI                      ;
RET   Z        ;
DEC   BC       ;
INC   HL       ;
INC   DE       ;
JR    SNOW     ;
ENDM           ;

START:        ;
CALL  DISKETA  ; Выполнение описанных макроопределений
NETEND        ;
GASI         ;
COLOR        ;
SCREEN2      ;
INITSPR     ;
CLS          ;
LD    HL,SGT  ; Заполнение видеопамати шаблонами спрайта
LD    DE,03800H ;
LD    BC,20H  ;
CALL  VRAM   ;
LD    HL,SAT  ;
LD    DE,01B00H ; Установка атрибутов спрайта
LD    BC,04H  ;
CALL  VRAM   ;
GORI        ;
LD    HL,H00K ;
LD    DE,0FD9FH ; Установка адреса обработки прерываний.
LD    BC,5    ; Обслуживает подпрограмму обработки H00K с
DI                      ; адресом 0FD9Fh
LDIR        ;
EI          ;

SIKL:NOP     ;
LD    HL,SAT  ; Постоянное отображение спрайта на экран, но коор-
LD    DE,01B00H ; динаты его меняются в подпрограмме обработки пре-
LD    BC,4    ; рываний в зависимости от состояния матрицы клави-
CALL  VRAM   ; атуры.
...      ; Здесь может находиться Ваша программа.
...      ; Например, программа движения фона или движения
...      ; других спрайтов.

```

```

... ;
JR CIKL ;

HOOK: RST 30H ;
      DEFB 8BH ; Межслотовый вызов подпрограммы
      DEFW STICK ; обработки прерываний
      RET ;

STICK: LD A,8 ; Подпрограмма обработки прерываний
      DI ;
      CALL BUFF ; После обращения к BUFF
      LD HL,SAT ; в регистре A - состояние ряда матрицы
      PUSH AF ;
      CP 0EFH ;
      JR Z,LEFT ; В зависимости от нажатой клавиши ("стрелки")
      POP AF ; меняется координата
      PUSH AF ;
      CP 0DFH ;
      JR Z,SHIFT ;
      POP AF ;
      PUSH AF ;
      CP 0BFH ;
      JR Z,MAIN ;
      POP AF ;
      PUSH AF ;
      CP 07FH ;
      JR Z,RIGHT ;
END: POP AF ;
     RET ;

SHIFT: DEC (HL) ; Клавиша "Стрелка вверх"
      JR END ;

MAIN: INC (HL) ; Клавиша "Стрелка вниз"
      JR END ;

LEFT: INC HL ; Клавиша "Стрелка влево"
      DEC (HL) ;
      JR END ;

RIGHT: INC HL ; Клавиша "Стрелка вправо"
      INC (HL) ;
      JR END ;

BUFF: LD C,A ; Подпрограмма возвращает состояние "сброшенных"
      DI ; битов для определенного ряда матрицы клавиатуры
      IN A,(0AAH) ;
      AND 0F0H ;
      ADD A,C ;
      OUT (0AAH),A ;
      EI ;
      IN A,(0A9H) ;
      RET ;

DISKETA: ;
      DISK ;

SGT: DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
SAT: DEFB 100,127,0,10

VRAM: VIDEO

```

https://sysadminmosaic.ru/msx/basic_dialogue_programming_language/104

2023-02-19 16:26

