

1.3. Работа с подпрограммами BDOS

Кроме подпрограмм BIOS в версии MSX Disk BASIC, а также в операционной системе MSX-DOS поддерживаются подпрограммы системы BDOS.

Подпрограммы BDOS вызываются следующим образом:

- α) из операционной системы MSX-DOS:

```
LD    C,номер подпрограммы
CALL  0005h
```


- β) из системы MSX Disk BASIC:

```
LD    C,номер подпрограммы
CALL  F37Dh
```

Приведем несколько примеров вызова подпрограмм BDOS из системы MSX Disk BASIC.

Пример 1. Вывод символа на экран.

[1030-01.bas](#)

 [1030-01.bas](#)

```
10 CLEAR 200,&HC000:DEFUSR=&HC000
20 DATA &H1E,64                : 'В регистре E - код символа
30 DATA &H0E,&H02                : 'LD C,02h          ;Номер подпрограммы
40 DATA &HCD,&H7D,&HF3            : '                ;Обращение к BDOS
50 DATA &HC9                    : 'RET
60 FOR T=0 TO 7:READ J:POKE &HC000+T,J:NEXT
70 A=USR(A)                      : ' На экране появляется символ!
```

Пример 2. Попробуем уничтожить имя файла с диска.

[1030-02.bas](#)

 [1030-02.bas](#)

```
10 CLEAR 200,&HC000:DEFUSR=&HC000
20 PRINT"Имя файла состоит из 8 символов, точки и еще 3 символов!"
30 INPUT"Какое имя уничтожить: ";F$
70 'Запись имени файла в FCB
80 POKE &HE000,1                  'Дисковод A:
90 FOR T=1 TO 8:POKE &HE000+T,ASC(MID$(F$,T,1)):NEXT 'Имя файла
100 FOR T=9 TO 11:POKE &HE000+T,ASC(MID$(F$,T+1,1)):NEXT 'Расширение
110 'Подпрограмма DELETE FILE
120 DATA &H11,&H00,&hE0          : 'LD DE,E000h;адрес FCB
130 DATA &H0E,&H13                : 'LD C,13h      ;Номер подпрограммы
140 DATA &HCD,&H7D,&HF3            : 'CALL F37Dh
150 DATA &HC9                    : 'RET
160 FOR T=0 TO 8:READ J:POKE &HC000+T,J:NEXT
170 A=USR(A): PRINT"Operation complete"
```

Пример 3. Вывод четырехбайтного числа, содержащегося в регистрах HL и DE на экран дисплея (авторы алгоритма: J.Suzuki и T.Patterson). Программа «вырезана» А.Никитиным из файла COMMAND.COM, поэтому, к сожалению, она написана на Макроассемблере.

[1030-03.asm](#)

 [1030-03.asm](#)

```
BDOS    EQU    #0005              ; Адрес, нужный для доступа к BDOS
;
LD      HL,12                    ; В регистре HL содержатся младшие два байта вы-
LD      DE,0                     ; водимого числа, а в регистре DE - старшие два
; байта
LD      BC,0                    ; Очистка ячеек памяти, нужных для перевода чис-
```

```

LD (METKA),BC ; ла из шестнадцатеричного вида в десятичный
LD (METKA+2),BC ;
LD C,#20 ; В регистре C - число 20h (п о ч е м у!?)
LOOP1: ADD HL,HL ; Сложение и сдвиги необходимы для установки
RL E ; флага C регистра флагов, а следовательно, и
RL D ; для увеличения содержимого регистра E, а затем
PUSH HL ; и регистра D .
LD HL,METKA ; Считывание предыдущих результатов вычислений
LD B,#04 ; и (при выполнении условия переполнения регист-
ZAPOL: LD A,(HL) ; ра D) увеличение их содержимого
ADC A,A ; Обратите внимание на эту команду: она увеличи-
DAA ; вает содержимое регистра A при установленном
LD (HL),A ; флаге C
INC HL ;
DJNZ ZAPOL ;
POP HL ;
DEC C ; Уменьшение значения цикла.
JR NZ,LOOP1 ; На выходе из этого цикла в ячейках METKA ÷
; METKA+3 будет лежать Ваше число в десятичном
; виде
LD BC,#0410 ; Первый байт - данные для работы цикла, второй
; байт нужен для коррекции кода числа
LD HL,METKA+3 ; Число выводится со старшего байта
PRINT: LD A,(HL) ; Блок печати числа
RRA ;
RRA ;
RRA ;
RRA ; "Переворачивание" числа...
CALL POLBYTE ; . . . и переход на печать младшего полубайта
LD A,#01 ; Проверка, последняя ли это цифра
SUB B ;
JR NZ,STEK ; Если выводимая цифра является последней и она-
LD C,A ; - ноль, то вывод символа "ноль"
STEK: LD A,(HL) ;
DEC HL ;
CALL POLBYTE ; Переход на печать старшего полубайта
DJNZ PRINT ;
RET ; К о н е ц программы
POLBYTE: AND #0F ; "Отрезание" младшего полубайта
JR Z,SUMMA ; Если это число - ноль, то - печать символа
; "пробел"
LD C,#00 ; Иначе - печать самого числа
SUMMA: OR #30 ; Получение кода данного числа
SUB C ;
WYWOD: PUSH AF ; Сохранение содержимого всех регистров
PUSH BC ;
PUSH DE ;
PUSH HL ;
LD E,A ; Печать числа по коду (из регистра E)
LD C,#02 ;
CALL BDOS ;
POP HL ;
POP DE ;
POP BC ;
POP AF ;
RET ;
METKA: DEFS 4 ; Область памяти,необходимая для арифметических
; расчетов

```

https://sysadminmosaic.ru/msx/basic_dialogue_programming_language/103?rev=1675591385

2023-02-05 13:03

