

ЛАБОРАТОРНАЯ РАБОТА №1

Программирование в формате .COM, передача параметров из командной строки, команды ввода-вывода, программирование портов

Цель работы.

Целью лабораторной работы является изучение особенностей структуры программы типа .COM, возможностей передачи строки параметров из командной строки MS-DOS в программу, методики работы с портами, а также получение навыков и умений по применению изученных возможностей для программирования.

Основные теоретические сведения

1. Структура программы типа .COM

Программа типа .COM отличается от программы типа .EXE тем, что содержит лишь один сегмент, включающий все компоненты программы: префикс программного сегмента (PSP), программный код (т.е. оттранслированные в машинные коды программные строки), данные и стек.

Структура типичной программы типа .COM на языке ассемблера выглядит следующим образом:

```
; Программа типа .COM
text segment 'code'
    assume CS:text,SS:text,DS:text,ES:text
    org 100h
main proc
```

; текст программы

```
    mov AX,4C00h ; завершение
    INT 21h      ; программы
main endp
; определения данных
```

```
text ends
```

```
end main
```

Программа содержит единственный сегмент text, которому присвоен класс 'CODE'. В операторе ASSUME указано, что все четыре сегментных регистра будут указывать на этот единственный сегмент. Как и в программе типа .EXE, определения ES:text и SS:text не обязательны. Оператор ORG 100h резервирует 256 байт для PSP. Заполнять PSP будет по-прежнему система, но место под него в начале сегмента должен отвести программист. В программе нет необходимости инициализировать сегментный регистр DS, поскольку его, как и остальные сегментные регистры, инициализирует система. Данные можно разместить после программной процедуры или перед ней. Следует иметь в виду, что при загрузке программы типа .COM регистр IP всегда инициализируется числом 100h, поэтому сразу вслед за оператором ORG 100h должна стоять первая выполняемая строка программы. Если данные желательно расположить в начале программы, перед ними следует поместить оператор JMP.

После загрузки программы все сегментные регистры указывают на начало единственного сегмента, т.е. фактически на начало PSP. Указатель стека автоматически инициализируется числом FFFEh. Таким образом, независимо от фактического размера программы ей выделяется 64K адресного пространства, всю нижнюю часть которого занимает стек. Верхняя граница стека не определена и зависит от интенсивности и способа использования стека.

Процесс отладки программы типа .COM включает следующие этапы:

- создание или редактирование текста программы с помощью текстового редактора;
- трансляция программы с целью получения объектного файла (>TASM prog.asm);
- компоновка объектного модуля (>TLINK prog /t).

2. Передача параметров в программу из командной строки

В некоторых случаях в программу бывает необходимо передать параметры вызова, чаще всего имена файлов, а также ключей, определяющих

режим работы программы. Одним из способов является передача параметров из командной строки, которые записываются в хвост команды. Например:

- 1.) >DIR prog.asm - командная строка, где
DIR - команда,
prog.asm - хвост команды (параметр);
- 2.) >MYPROG abcd - командная строка, где
MYPROG - команда,
abcd - хвост команды (параметр).

Следует отметить, что пробел(ы), отделяющие команду от параметров, является частью хвоста.

В процессе загрузки программы в память DOS пересылает в префикс программного сегмента в байт по адресу 80h длину хвоста, а начиная с адреса 81h содержимое хвоста команды.

Например :

- 1.) >MYPROG abcd

↓ адрес от начала PSP (Hex)

80	81	82	83	84	85	86
05	20	61	62	63	64	0D

↖ содержимое PSP (Hex)

5		a	b	c	D
---	--	---	---	---	---

↑ ↑
Длина Пробел

↑
Код завершения хвоста (не считается)

- 2.) >MYPROG

↓ адрес от начала PSP (Hex)

80	81
00	0D

↖ содержимое PSP (Hex)

0	
---	--

↑ ↑
Длина Код завершения хвоста (не считается)

Пример извлечения хвоста из PSP в строку STR1 :

; фрагмент программы для формата .COM

```
XOR CH,CH      ; CH = 0
MOV CL,DS:[80h] ; CX = длина хвоста
JCXZ VIXOD      ; переход на метку, если хвоста нет
DEC CX          ; скорректируем длину на пробел
MOV SI,82h      ; SI = адрес начала хвоста без первого пробела
LEA DI,STR1     ; DI = адрес приемника
CLD             ; DF = 0
REP MOVSB       ; пересылка хвоста в строку STR1
VIXOD:          ; параметры отсутствуют
; определения данных
STR1 DB 256 DUP(' ')
```

3. Команды ввода-вывода. Программирование портов

Управление разнообразной аппаратурой компьютера - контроллером клавиатуры, видеоадаптером, принтером и др., осуществляется через управляющие регистры этой аппаратуры. Каждый из регистров имеет закрепленный за ним номер в диапазоне от 0000h до FFFFh. Этот номер называется портом, и программирование аппаратуры путем непосредственного обращения к ее регистрам носит название программирование через порты. Обращение к портам осуществляется с помощью двух команд - IN (input, ввод) для ввода и OUT (output, вывод) для вывода. Никакие другие команды для работы с портами не годятся, поэтому непосредственно в регистрах клавиатуры нельзя выполнять арифметические или логические операции. При необходимости изменить содержимое аппаратного регистра, надо сначала с помощью команды IN прочитать содержимое этого регистра в регистр процессора или ячейку памяти, там требуемым образом изменить его и затем с помощью команды OUT вывести новое данные назад в регистр аппаратуры.

Команда IN вводит в регистр AL или AX соответственно байт или слово из порта, указываемого вторым операндом непосредственным значением, если адрес не превышает FFh, иначе адрес предварительно необходимо поместить в регистр DX.

Команда OUT выполняет противоположное действие. Формат команды -
OUT порт, AX или AL.

Рассмотрим методику программирования ввода-вывода через порты на примере управления динамиком компьютера.

Программирование звука в простейшем случае осуществляется путем периодического включения и выключения тока, протекающего через динамик, по командам программы. Если частота переключения тока лежит в пределах приблизительно от 16Гц до 16КГц, мы слышим звуковой тон соответствующей частоты. Для управления током динамика служит бит 1 порта с номером 61h. Установка этого бита в 1 включает ток в динамике, установка в 0 - выключает. Таким образом, для получения звукового тона программа должна непрерывно переключать этот бит, что исключает выполнение программой какой-либо работы одновременно с выводом звука. Для того чтобы организовать звуковое сопровождение, т.е. генерацию звука одновременно с работой программы, следует воспользоваться более сложным методом.

Пример. Фрагмент программы генерации звука.

```
cli                ;Запрет аппаратных прерываний
in  AL,61h         ;Введем содержимое порта 61h
mov CX,2000        ;Установим длит. звук. сигнала
begin: push CX      ;Сохраним счетчик цикла
      or  AL,00000010b ;Установим бит 1
      out 61h,AL      ;Выведем в порт, вкл. динамик
      mov CX,1000     ;Организуем паузу, в течение
cikl: loop cikl      ;которой через динамик течет ток
      and AL,11111101b ;Сбросим бит 1
      out 61h,AL      ;Выведем в порт, выключив динамик
      mov CX,1000     ;Организуем паузу, в течение кото-
cikl1: loop cikl1    ;рой через динамик не течет ток
      pop CX          ;Восстановим счетчик цикла
```

```
loop begin        ;Повторять CX раз
sti              ;Разрешение аппаратных прерываний
```

Описанная методика генерации звукового тона может быть использована в программах для подачи предупреждающих сигналов, однако их частота, а вместе с ней и длительность, будут зависеть от скорости работы процессора, что чрезвычайно неудобно.

Задание для практической работы

Написать и отладить программу типа .COM, которая анализирует параметр, передаваемый через командную строку. В случае его отсутствия - генерирует звуковой сигнал.

Варианты:

- 1.) Удалить пробелы слева и вывести параметр на печать;
- 2.) Проверить - является ли параметр числом?
- 3.) Вывести на экран число единичных бит первого отличного от пробела символа параметра;
- 4.) Вывести на экран число нулевых бит первого символа параметра;
- 5.) Вывести на экран шестнадцатеричный ASCII код первого символа параметра;
- 6.) Вывести на экран десятичный ASCII код первого символа параметра.
- 7.) Удалить пробелы справа и вывести параметр на печать.

ЛАБОРАТОРНАЯ РАБОТА №2

Средства ввода данных с клавиатуры в MS DOS

Цель работы

Целью лабораторной работы является изучение системной процедуры обработки прерываний от клавиатуры и средств, предоставляемых операционной системой для ввода данных с клавиатуры, а также получение практических навыков программирования операций ввода с клавиатуры.