

ЛАБОРАТОРНАЯ РАБОТА №9

ПЛАН

По дисциплине: Программирование встроенных систем

Тема занятия: **Путь запуска программы**

Цель занятия: усвоение понятий сегментного адреса и адреса среды окружения программы

Количество часов: 2

Содержание работы:

Адрес среды окружения находится в PSP со смещением 2Ch. Сегментный адрес PSP находится в регистрах DS и ES при старте программы. Необходимо получить из PSP адрес среды окружения, в среде окружения найти строку, которая описывает путь запуска программы, и выдать ее на экран.

Перед путем запуска программы выдать сообщение «Это путь запуска программы:» с помощью функции ah=09h прерывания int 21h.

Вывод пути запуска программы осуществить с помощью функцию ah=02h прерывания int 21h.

Структура среды окружения:

```
db      строка_1,0
db      строка_2,0
...
db      строка_n,0,0
db      x,x
db      строка_запуска,0
```

Дополнительная информация

Функция DOS 09h. Записать строку в STDOUT (стандартное устройство вывода, экран) с проверкой на Ctrl-Break.

Вход: AH=09h.

DS:DX = адрес строки, заканчивающейся символом \$ (24h).

Выход: Нет, согласно документации, но на самом деле: AL = 24h (код последнего символа).

Функция DOS 02h: Записать символ в STDOUT с проверкой на Ctrl-Break.

Вход: AH = 02h DL = ASCII-код символа

Выход: Никакого, согласно документации, но на самом деле: AL = код последнего записанного символа (равен DL, кроме случая, когда DL = 09h (табуляция), тогда в AL возвращается 20h).

Данная функция обрабатывает некоторые управляющие символы: при выводе символа BEL (07h) появляется звуковой сигнал, посредством BS (08h) курсор перемещается влево на одну позицию, символ HT (09h) используется для замены на несколько пробелов, символ LF (0Ah) - для перевода курсора на одну позицию вниз, а CR (0Dh) - для перехода на начало текущей строки.

Если в ходе работы этой функции была нажата комбинация клавиш Ctrl-Break, вызывается прерывание 23h, которое по умолчанию осуществляет выход из программы.

```
.model small
.286
.data
.code
start:
    mov ax,[ds:2ch]
    mov ds,ax
    mov si,0
k1:mov dl,[ds:si]
    inc si
    cmp dl,0
    jne k1
    mov dl,[ds:si]
    inc si
    cmp dl,0
    jne k1
    add si,2
k3:mov dl,[ds:si]
    cmp dl,0
    je k2
    mov ah,2h
    int 21h
    inc si
    jmp k3
k2:mov ah,4ch
    int 21h
end start
```