

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Операционные системы»
Тема: «Исследование интерфейсов программных модулей»

Студент гр. 7381

Тарасенко Е.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

Цель работы.

Исследование интерфейса управляющей программы и загрузочных модулей. Этот интерфейс состоит в передаче запускаемой программе управляющего блока, содержащего адреса и системные данные. Так загрузчик строит префикс сегмента программы (PSP) и помещает его адрес в сегментный регистр. Исследование префикса сегмента программы (PSP) и среды, передаваемой программе.

Необходимые сведения для составления программы.

При начальной загрузке программы формируется PSP, который размещается в начале первого сегмента программы. PSP занимает 256 байт и располагается с адреса, кратного границе сегмента. При загрузке модулей типа .COM все сегментные регистры указывают на адрес PSP. Именно по этой причине значения этих регистров в модуле .EXE следует переопределять.

Формат PSP:

Смещение	Длина поля (байт)	Содержимое поля
0	2	int 20h
2	2	Сегментный адрес первого байта недоступной памяти. Программа не должна модифицировать содержимое памяти за этим адресом.
4	6	Зарезервировано
0Ah(10)	4	Вектор прерывания 22h (IP, CS)
0Eh(14)	4	Вектор прерывания 23h (IP, CS)
12h(18)	4	Вектор прерывания 24h (IP, CS)
2Ch(44)	2	Сегментный адрес среды, передаваемой программе.
5Ch		Область форматируется как стандартный неоткрытый блок управления файлом (FCB)

6Ch		Область форматируется как стандартный неоткрытый блок управления файлом (FCB). Перекрывается, если FCB с адреса 5Ch открыт.
80h	1	Число символов в хвосте командной строки.
81h		Хвост командной строки – последовательность символов после имени вызываемого модуля.

Область среды содержит последовательность символьных строк вида:

имя=параметр

Каждая строка завершается байтом нулей.

В первой строке указывается имя COMSPEC, которая определяет используемый командный процессор и путь к COMMAND.COM. Следующие строки содержат информацию, задаваемую командами PATH, PROMT, SET.

Среда заканчивается также байтом нулей. Таким образом, два нулевых байта являются признаком конца переменных среды. Затем идут два байта, содержащих 00h, 01h, после которых располагается маршрут загруженной программы. Маршрут также заканчивается байтом 00h.

Постановка задачи.

Необходимо написать и отладить программный модуль типа .COM, выбирает и распечатывает следующую информацию.

1. Сегментный адрес недоступной памяти, взятый из PSP, в шестнадцатеричном виде.
2. Сегментный адрес среды, передаваемой программе, в шестнадцатеричном виде.
3. Хвост командной строки в символьном виде.
4. Содержимое области среды в символьном виде.

5. Путь загружаемого модуля.

Ход работы.

В ходе данной лабораторной работы была реализована программа, выводящая в консоль сегментный адрес недоступной памяти и сегментный адрес среды в шестнадцатеричном виде; хвост командной строки и содержимое области среды в символьном виде. Результат работы программы см. на рис. 1.



```
C:\>code.com

Locked memory address: 9FFFh
Environment address: 0188h
There are no sybmols
Content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6

Path:
C:\CODE.COM_
```

Рис. 1 – Результат работы программы

Сегментный адрес недоступной памяти.

1) На какую область памяти указывает адрес недоступной памяти?

На область памяти ROM BIOS.

2) Где расположен этот адрес по отношению к области памяти, отведенной программе?

Сразу после области памяти, отведенной программе.

3) Можно ли в эту область памяти писать?

Да, можно (в MS-DOS нет защиты памяти).

Среда, передаваемая программе.

1) Что такое среда?

Среда – область памяти, в которой в виде символьных строк (имя = параметр) записаны значения переменных, называемых переменными средами

и содержащих данные о некоторых директориях операционной системы и конфигурации компьютера.

2) Когда создается среда? Перед запуском приложения или в другое время?

При запуске MS-DOS.

3) Откуда берется информация, записываемая в среду?

Из системного файла AUTOEXEC.BAT (для данной ОС).

Вывод.

В процессе выполнения данной работы было произведено ознакомление с интерфейсом загрузочной программы и загрузочных модулей. Получены необходимые знания о среде, передаваемой программе, и недопустимых сегментах памяти.

ПРИЛОЖЕНИЕ А.

Исходный текст программы

```
TESTPC      SEGMENT
              ASSUME  CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
              org 100H
START:       JMP      BEGIN

; DATA
MEM db 13, 10, "Locked memory address:      h$"
ENV db 13, 10, "Environment address:        h$"
TAIL db 13, 10, "Command line tail:         $"
EMP db 13, 10, "There are no sybmols$"
CONT db 13, 10, "Content:", 13, 10, "$"
ENT db 13, 10, "$"
PATH db 13, 10, "Path:", 13, 10, "$"

; PROCS
WRITE PROC near
    push ax
    mov ah, 09h
    int 21h
    pop ax
    ret
WRITE ENDP

INFO PROC near
    ; Memory
    mov ax, ds:[02h]
    mov di, offset MEM
    add di, 28
    call WRD_TO_HEX
    mov dx, offset MEM
    call WRITE

    ; Environment
    mov ax, ds:[2Ch]
    mov di, offset ENV
    add di, 26
    call WRD_TO_HEX
    mov dx, offset ENV
```

```

call WRITE

; Tail
xor cx, cx
mov cl, ds:[80h]
mov si, offset TAIL
add si, 20
test cl, cl
jz empty
xor di, di
xor ax, ax
readtail:
    mov al, ds:[81h + di]
    mov [si], al
    inc di
    inc si
    loop readtail
    mov dx, offset TAIL
    call WRITE
    jmp nextaction
empty:
    mov dx, offset EMP
    call WRITE
nextaction: nop

; Environment content
mov dx, offset CONT
call WRITE
xor di, di
mov bx, 2Ch
mov ds, [bx]
readstring:
    cmp byte ptr [di], 00h
    jz pressenter
    mov dl, [di]
    mov ah, 02h
    int 21h
    jmp findend
pressenter:
    push ds
    mov cx, cs

```

```

        mov ds, cx
        mov dx, offset ENT
        call WRITE
        pop ds
findend:
        inc di
        cmp word ptr [di], 0001h
        jz readpath
        jmp readstring
readpath:
        push ds
        mov ax, cs
        mov ds, ax
        mov dx, offset PATH
        call WRITE
        pop ds
        add di, 2
pathloop:
        cmp byte ptr [di], 00h
        jz final
        mov dl, [di]
        mov ah, 02h
        int 21h
        inc di
        jmp pathloop
final:
        ret
INFO ENDP

TETR_TO_HEX    PROC    near
                and     AL,0Fh
                cmp     AL,09
                jbe     NEXT
                add     AL,07
NEXT:          add     AL,30h
                ret
TETR_TO_HEX    ENDP

BYTE_TO_HEX    PROC    near
; байт в AL переводится в два символа шестн. числа в AX
                push    CX

```



```

        mov     AH,AL
        call    TETR_TO_HEX
        xchg    AL,AH
        mov     CL,4
        shr     AL,CL
        call    TETR_TO_HEX
        pop     CX
        ret
BYTE_TO_HEX ENDP

```

```

WRD_TO_HEX  PROC  near
; перевод в 16 с/с 16-ти разрядного числа
; в AX - число, DI - адрес последнего символа

```

```

        push    BX
        mov     BH,AH
        call    BYTE_TO_HEX
        mov     [DI],AH
        dec     DI
        mov     [DI],AL
        dec     DI
        mov     AL,BH
        call    BYTE_TO_HEX
        mov     [DI],AH
        dec     DI
        mov     [DI],AL
        pop     BX
        ret
WRD_TO_HEX ENDP

```

```

BYTE_TO_DEC  PROC  near
; перевод в 10с/с, SI - адрес поля младшей цифры

```

```

        push    CX
        push    DX
        xor     AH,AH
        xor     DX,DX
        mov     CX,10
loop_bd:  div     CX
        or      DL,30h
        mov     [SI],DL
        dec     si
        xor     DX,DX

```

```

        cmp     AX,10
        jae     loop_bd
        cmp     AL,00h
        je      end_1
        or      AL,30h
        mov     [SI],AL

end_1:   pop     DX
        pop     CX
        ret

BYTE_TO_DEC   ENDP

```

```

; CODE

```

```

BEGIN:

        call INFO
        mov ah, 10h
        int 16h

```

```

; Выход в DOS

```

```

        xor     al, al
        mov     ah,4Ch
        int     21H

```

```

TESTPC   ENDS
END       START

```