

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

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

Студент гр. 8381

\_\_\_\_\_

Киреев К.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### 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.

### Основные теоретические положения.

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. При загрузке модуля типа .EXE сегментные регистры DS и ES указывают на PSP. Именно по этой причине значения этих регистров в модуле .EXE следует переопределять.

Таблица 1 – 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, PROMPT, 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.

### **Выполнение работы.**

Написан текст исходного .COM модуля, который выбирает и распечатывает следующую информацию:

- Сегментный адрес недоступной памяти, взятый из PSP
- Сегментный адрес среды, передаваемой программе
- Хвост командной строки в символьном виде
- Содержимое области среды в символьном виде
- Путь загружаемого модуля

Полученный исходный модуль был отлажен. Результаты выполнения программы представлены на рис. 1.

```
S:\>os.com key
Unavailable memory segment address: 9FFF
Segment address of the environment: 0188
Command line tail: key
Content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path: S:\OS.COM
```

Рисунок 1 – Вывод программы

## Контрольные вопросы

### Сегментный адрес недоступной памяти

- На какую область памяти указывает адрес недоступной памяти?

Область недоступной памяти начинается с 9FFF. Ее адрес указывает на служебную часть памяти, которую DOS не может выделить под программу.

- Где расположен этот адрес по отношению области памяти, отведенной программе?

Адрес недоступной памяти указывает на последний параграф памяти, которую DOS отводит пользовательским программам.

Объем адресного пространства		Физические адреса	Сегментные адреса	
1 Кбайт	Векторы прерываний	00000h	0000h	Обычная память (640 Кбайт)
256 байт	Область данных BIOS	00400h	0040h	
512 байт	Область данных DOS	00500h	0050h	
Более 70 Кбайт	Операционная система MS-DOS	00700h	0070h	
	Загружаемые драйверы DOS			
	COMMAND.COM (резидентная часть)			Верхняя память (384 Кбайт)
	Свободная память для загружаемых прикладных и системных программ (менее 570 Кбайт)			
64 Кбайт	Графическая видеопамять	A0000h	A000h	
32 Кбайт	Свободные адреса	B0000h	B000h	
32 Кбайт	Текстовая видеопамять	B8000h	B800h	
52 Кбайт	ПЗУ расширений BIOS	C0000h	C000h	Расширенная память
12 Кбайт	Свободные адреса	CD000h	CD00h	
128 Кбайт	Свободные адреса	D0000h	D000h	
64 Кбайт	ПЗУ BIOS	F0000h	F000h	
64 Кбайт	НМА	FFFF0h	FFFFh	
До 4 Гбайт (включая первый мегабайт)	XMS	10FFF0h		

Рисунок 2 – Типичное распределение адресного пространства

- **Можно ли в эту область памяти писать?**

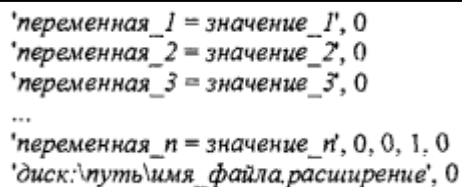
Можно, так как DOS не контролирует обращение программы к памяти.

## **Среда, передаваемая программе**

- **Что такое среда?**

Среда представляет собой область памяти, в которой в виде символьных строк записаны значения переменных, называемых переменными среды. Формат среды приведен на рис. 3.

Имеется ряд переменных среды, имена которых зарезервированы и известны системе, однако пользователь может включать в среду и свои переменные для использования их прикладными программами. Среда служит для передачи программам (как системным, так и прикладным) требуемых параметров. Параметры заносятся в среду с помощью системной команды SET. Системные и прикладные программы могут анализировать текущий состав среды и извлекать из него относящиеся к ним параметры.



```
'переменная_1 = значение_1', 0  
'переменная_2 = значение_2', 0  
'переменная_3 = значение_3', 0  
...  
'переменная_n = значение_n', 0, 0, 1, 0  
'диск:\путь\имя_файла.расширение', 0
```

Рисунок 3 – Формат среды (все числа занимают по одному байту)

- **Когда создается среда? Перед запуском приложения или в другое время?**

В процессе начальной загрузки DOS создает начальную среду, в которой будут работать активизируемые программы, и прежде всего командный процессор COMMAND.COM.

- **Откуда берется информация, записываемая в среду?**

Среда для командного процессора, создаваемая в процессе начальной загрузки, в простейшем случае содержит переменные COMSPEC, PROMPT и PATH, которые заносятся в среду из файла AUTOEXEC.BAT.

**Вывод.**

В результате выполнения данной лабораторной работы был исследован интерфейс управляющей программы и загрузочных модулей. Была написана программа, которая выводит на экран сегментный адрес недоступной памяти, взятый из PSP, сегментный адрес среды, передаваемой программе, хвост командной строки и путь загружаемого модуля.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ. OS.COM

```
lab segment
assume cs:lab, ds:lab, es:nothing, ss:nothing
org 100h
main: jmp processing

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 16 числа в ah
    push cx
    mov  ah, al
    call tetr_to_hex
    xchg al, ah
    mov  cl, 4
    shr  al, cl
    call tetr_to_hex ;в al старшая цифра
    pop  cx          ;в ah младшая цифра
    ret
byte_to_hex    endp

word_to_hex    proc near
    ;перевод в 16 ss 16 разрядного числа
    ;в ah - 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
    xor  ah, ah
    call byte_to_hex
    mov  [di], ah
```

```

        dec    di
        mov    [di], al
        pop    bx
        ret
word_to_hex    endp

```

```

print proc near
    push ax
    mov ah, 09h
    int 21h
    pop ax
    ret
print endp

```

```

processing:
    ;Unavailable memory segment address
    mov ax, ds:[02h]
    mov di, offset memData + 39
    call word_to_hex
    mov dx, offset memData
    call print

    ;Segment address of the environment
    mov di, offset envData + 39
    mov ax, ds:[2ch]
    mov env_seg, ax ;запоминаем сегмент
    call word_to_hex
    mov dx, offset envData
    call print

    ;Command line tail
    mov cl, ds:[80h]
    mov di, offset cmdLine + 18
    test cl, cl
    je noCmdLine
    mov si, 81h
    copy:
        mov al, ds:[si]
        mov [di], al
        inc si
        inc di
        loop copy
    mov dx, offset cmdLine
    call print
    jmp env

```



```

noCmdLine:
    mov dx, offset ZcmdLine
    call print

;Environment
env:
    mov dx, offset content
    call print
    xor di, di
    mov ds, env_seg
    mov cx, 2
reading:
    cmp byte ptr [di], 00h ;конец строки
    je output
    mov dl, [di]
    mov ah, 02h
    int 21h
    jmp env_end
output:
    cmp cx, 0
    jz env_end
    mov dl, 13
    mov ah, 02h
    int 21h
    mov dl, 10
    mov ah, 02h
    int 21h
    dec cx
env_end:
    inc di
    cmp byte ptr [di], 0001h
    je path
    jmp reading

;Path
path:
    add di, 2 ;пропуск байтов 00h, 01h
    mov cx, cs
    mov ds, cx
    lea dx, path_string
    call print
    mov ds, env_seg
reading_path:
    cmp byte ptr [di], 00h
    je end_path

```

```

        mov dl, [di]
        mov ah, 02h
        int 21h
        inc di
        jmp reading_path
end_path:
        mov cx, cs
        mov ds, cx
        lea dx, env_string
        call print
mov ah, 4ch
int 21h
ret

```

```

env_seg dw ?
memData db 'Unavailable memory segment address:      ', 13, 10, '$'
envData db 'Segment address of the environment:      ', 13, 10, '$'
cmdLine db 'Command line tail:                        ', '$'
ZcmdLine db 'No command line tail', '$'
content db 13, 10, 'Content:', 13, 10, '$'
path_string db 13, 10, 'Path: ', '$'
env_string db '$'
lab ends
end main

```