

**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**

Студентка гр. 8382

\_\_\_\_\_

Кузина А.М.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## Цель работы.

Исследование интерфейса управляющей программы и загрузочных модулей.  
Исследование префикс сегмента программы и среды, передаваемой программе.

## Ход выполнения работы.

Была написана программа, исходный код которой приведен в приложении А, распечатывающая на экран следующую информацию:

- Сегментный адрес недоступной памяти, взятый из PSP.
- Сегментный адрес среды, передаваемой программе.
- Последовательность символов после имени вызываемого модуля — хвост командной строки.
- Содержимое области среды.
- Путь загружаемого модуля.

Результат работы программы представлен на рисунке 1.

Рисунок 1 — результат работы программы.



```
C:\>I2.com test line :)
Unavailable memory address: 9FFF
Environment address: 0188
Command line tail: test line :)
Environment contents:
  PATH=Z:\
  COMSPEC=Z:\COMMAND.COM
  BLASTER=A220 I7 D1 H5 T6
Load module path: C:\I2.COM
```

Необходимые для выполнения работы данные представлены в таблице 1.

Таблица 1 - необходимые данные из PSP.

Смещение в PSP	Содержимое поля
02h	Сегментный адрес первого байта недоступной памяти
2Ch	Сегментный адрес среды, передаваемой программе
80h	Число символов в конце командной строки
81h	Хвост командной строки

Содержимое области среды представляет из себя последовательность строк, начиная с адреса es:[2Ch]. Каждая из строк заканчивается байтом нулей, также как и среда. Путь загружаемого модуля следует за средой, через два байта.

## **Контрольные вопросы**

Сегментный адрес недоступной памяти.

1) На какую область памяти указывает адрес недоступной памяти?

Адрес недоступной памяти указывает на адрес, начиная с которого идет память, недоступная для загрузки в нее пользовательских программ.

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

Он располагается за выделенной под программу памятью.

3) Можно ли писать в эту область памяти?

Можно, т. к. DOS не контролирует, в какую область памяти обращается программа

Среда, передаваемая программе.

1) Что такое среда?

Среда — это последовательность символьных строк, вида: имя параметра = значение параметра. Эти строки содержат различную нужную информацию, например о настройках системы и т. д.

2) - 3) Когда создается среда? Перед запуском приложения или в другое время? Откуда берется информация, записываемая в среду?

В процессе загрузки DOS создается окружение — корневая среда. При запуске какой-либо программы выделяется память для среды, которая по умолчанию будет являться копией окружения, процесса-родителя. В процессе выполнения программы, ее среда может быть изменена.

## **Выводы**

В ходе выполнения лабораторной работы была написана программа, выводящая на экран различную информацию из PSP. Также были исследованы интерфейсы управляющей программы и загрузочных модулей, префикс сегмент программы и среда, передаваемая программе.

## ПРИЛОЖЕНИЕ А

### Исходный код программы l2.asm

```
STT    SEGMENT
        ASSUME CS:STT, DS:STT, ES:NOTHING, SS:NOTHING
        ORG 100H
START: JMP BEGIN

MemAdd db 'Unavailable memory address:      ',13,10,'$'
EnvAdd db 'Environment address:      ', 13, 10, '$'
LineTail db 'Command line tail: ', '$'
EndLine db ' ', 13, 10, '$'
Tab db ' ', '$'
EnvCon db 'Enviroment contents: ', '$'
Path db 'Load module path: ', '$'

BEGIN:

;Memory adress
    mov ax, ss:[02h]
    mov di, offset MemAdd
    add di, 31
    call WrdToHex
    mov [di], ax
    mov dx, offset MemAdd
    mov ah, 09h
    int 21h

;environment address
    mov ax, ss:[2Ch]
    mov di, offset EnvAdd
    add di, 24
    call WrdToHex
    mov [di], ax
    mov dx, offset EnvAdd
    mov ah, 09h
    int 21h

;command line tail
    mov bx, 0
    mov cl, ss:[80h]
    mov ax, ss:[81h]
    mov dx, offset LineTail
    mov ah, 09h
    int 21h
    cmp cl, 0
    je writel
    mov ah, 02h

line:
    mov dl, ss:[81h + bx]
    int 21h
    inc bx
    loop line

writel:
    mov dx, offset EndLine
    mov ah, 09h
    int 21h

;enviroment contents
    mov dx, offset EnvCon
    mov ah, 09h
    int 21h
    mov bx, 0
    mov es, ss:[2Ch]
```

[illegible]

```
WrdToHex PROC near
    push bx
    mov bh,ah
    call ByteToHex
    mov [di],ah
    dec di
    mov [di],al
    dec di
    mov al,bh
    call ByteToHex
    mov [di],ah
    dec di
    mov [di],al
    pop bx
    ret
WrdToHex ENDP

STT     ENDS
        END START
```