

**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

\_\_\_\_\_

Адамов Я.В.

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.

### **Описание функций.**

Название функции	Описание
TETR_TO_HEX	Перевод 4-битного числа в код символа в 16 с/с. Вспомогательная функция для BYTE_TO_HEX.
BYTE_TO_HEX	Байт в AL переводится в два символа шестн. числа в AX.
WRD_TO_HEX	Перевод в 16 с/с 16-ти разрядного числа, в AX – число, DI – адрес последнего символа.
PrintMsg	Печать строки, адрес которой помещен в DX.
PrintSegmentAddressOfUnavailableMemory	Печать сегментного адреса недоступной памяти, взятого из PSP, в шестнадцатеричном виде.
PrintSegmentAddressOfEnvironment	Печать сегментного адреса среды, передаваемой программе, в шестнадцатеричном виде.
PrintCommandLineTail	Печать хвоста командной строки в символьном виде.
PrintEnvironmentContentAndModulePath	Печать содержимого области среды и пути загружаемого модуля в символьном виде.

### Описание структур данных.

Название	Тип	Описание
SegmentAddressOfUnavailableMemory	db	Строка: «Segment address of unavailable memory: h».
SegmentAddressOfEnvironment	db	Строка: « Segment address of environment: h».
Tail	db	Строка: « Tail:».
ContentOfEnvironment	db	Строка: « Content of environment:».
PathOfModule	db	Строка: « Path of module:».
EndOfString	db	Строка, содержащая символы для перевода строки.
EndOfStringWithTab	db	Перевод строки с дальнейшей табуляцией.

### Описание работы утилиты.

Программа выводит на экран сегментный адрес первого байта недоступной памяти, адрес среды, передаваемой программе, хвост командной строки, содержимое области среды и путь загружаемого модуля. Результат работы программы представлен на рис. 1.

```
Segment address of unavailable memory: 9FFFh
Segment address of environment: 0188h
Tail:
Content of environment:
  PATH=Z:\
  COMSPEC=Z:\COMMAND.COM
  BLASTER=A220 I7 D1 H5 T6
Path of module:
  C:\CODE.COM
```

Рисунок 1 – результат работы программы.

## **Вывод.**

В ходе выполнения лабораторной работы были исследованы интерфейс управляющей программы и загрузочных модулей.

## **Ответы на контрольные вопросы.**

Сегментный адрес недоступной памяти:

*На какую область памяти указывает адрес недоступной памяти?*

На границу между доступной оперативной памятью и не доступной, которая необходимая операционной системе.

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

Сразу за областью памяти, отведенной программе.

*Можно ли в эту область памяти писать?*

Да, но это может привести к непредсказуемым последствиям.

Среда передаваемая программе:

*Что такое среда?*

Среда – область памяти, содержащая набор параметров системы.

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

Во время загрузки DOS.

*Откуда берется информация, записываемая в среду?*

Из системного файла autoexec.bat.

## Приложение А. Код программы.

TESTPC SEGMENT

ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING

ORG 100H

START: JMP BEGIN

; \_\_\_\_\_

; Данные

SegmentAddressOfUnavailableMemory db 0dh,0ah,'Segment address of  
unavailable memory: h',0dh,0ah,'\$'

SegmentAddressOfEnvironment db 'Segment address of environment:  
h',0dh,0ah,'\$'

Tail db 'Tail:', '\$'

ContentOfEnvironment db 'Content of environment:\$'

PathOfModule db 0dh,'Path of module:\$'

EndOfString db 0dh,0ah,'\$'

EndOfStringWithTab db 0dh,0ah,' \$'

; \_\_\_\_\_

; Процедуры

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
    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
    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
```

```
PrintMsg PROC near
    push ax
    mov ah,09h
    int 21h
    pop ax
    ret
PrintMsg ENDP
```

```
PrintSegmentAddressOfUnavailableMemory PROC near
    push ax
    push di
    push dx

    mov ax,es:[2]
    lea di,SegmentAddressOfUnavailableMemory
    mov dx,di
    add di,44
    call WRD_TO_HEX
    call PrintMsg
```

```
pop dx
pop di
pop ax
ret
```

PrintSegmentAddressOfUnavailableMemory ENDP

PrintSegmentAddressOfEnvironment PROC near

```
push ax
push di
push dx
```

```
mov ax,es:[2Ch]
lea di,SegmentAddressOfEnvironment
mov dx,di
add di,35
call WRD_TO_HEX
call PrintMsg
```

```
pop dx
pop di
pop ax
ret
```

PrintSegmentAddressOfEnvironment ENDP

PrintCommandLineTail PROC near



```
push ax
push cx
push dx
push si

lea dx,Tail
call PrintMsg
xor ax,ax
mov al,es:[80h]
add al,81h
mov si,ax
push es:[si]
mov byte ptr es:[si+1],'$'
push ds
mov cx,es
mov ds,cx
mov dx,81h
call PrintMsg
lea dx,EndOfString
call PrintMsg

pop ds
pop es:[si]
pop si
pop dx
pop cx
pop ax
ret
```

```
PrintCommandLineTail ENDP
```

```
PrintEnvironmentContentAndModulePath PROC near
```

```
    push si
```

```
    push es
```

```
    push ax
```

```
    push bx
```

```
    push cx
```

```
    push dx
```

```
    lea dx,ContentOfEnvironment
```

```
    call PrintMsg
```

```
    mov bx,1
```

```
    mov es,es:[2ch]
```

```
    mov si,0
```

```
p_1:
```

```
    lea dx,EndOfStringWithTab
```

```
    call PrintMsg
```

```
    mov ax,si
```

```
p_2:
```

```
    cmp byte ptr es:[si],0
```

```
    je p_3
```

```
    inc si
```

```
    jmp p_2
```

```
p_3:
```

```
    push es:[si]
```

```
    mov byte ptr es:[si], '$'
```

```

    push ds
    mov cx,es
    mov ds,cx
    mov dx,ax
    call PrintMsg
    pop ds
    pop es:[si]
    cmp bx,0
    jz p_4
    inc si
    cmp byte ptr es:[si],01h
    jne p_1
    lea dx,PathOfModule
    call PrintMsg
    mov bx,0
    add si,2
    jmp p_1
p_4:
    pop dx
    pop cx
    pop bx
    pop ax
    pop es
    pop si
    ret
PrintEnvironmentContentAndModulePath ENDP

```

```
; _____  
; Код
```

BEGIN:

```
    call PrintSegmentAddressOfUnavailableMemory  
    call PrintSegmentAddressOfEnvironment  
    call PrintCommandLineTail  
    call PrintEnvironmentContentAndModulePath  
    lea dx,EndOfString  
    call PrintMsg
```

```
    xor al,al  
    mov ah,04Ch  
    int 21h  
    ret
```

TESTPC ENDS

END START