

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Операционные системы»
Тема: ИССЛЕДОВАНИЕ ИНТЕРФЕЙСОВ ПРОГРАММНЫХ МОДУЛЕЙ

Студент гр. 9383

Хотяков Е.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Исследование интерфейса управляющей программы и загрузочных модулей. Исследование префикса сегмента программы (PSP) и среды, передаваемой программе.

Задание.

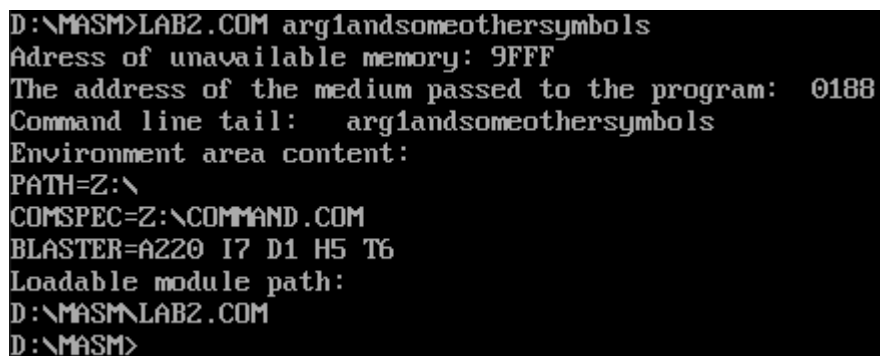
Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа **.COM**, который выбирает и распечатывает следующую информацию:

- 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.

Шаг 2. Оформление отчета в соответствии с требованиями. В отчет включите скриншот с запуском программы и результатами.

Выполнение работы.

Шаг 1: Была написана **.COM** программа на ассемблере, которая выводит требуемую информацию.



```
D:\MASM>LAB2.COM arglandsomeothersymbols
Address of unavailable memory: 9FFF
The address of the medium passed to the program: 0188
Command line tail:  arglandsomeothersymbols
Environment area content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Loadable module path:
D:\MASM\LAB2.COM
D:\MASM>
```

Рисунок 1 - Пример работы программы

Ответы на контрольные вопросы:

Сегментный адрес недоступной памяти

1. На какую область памяти указывает адрес недоступной памяти?

На первые байты сегмента, идущего после программы

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

Недоступная память располагается в стороне бОльших адресов

3. Можно ли в эту область памяти писать?

Да, т.к. в DOS не предусмотрена защита от перезаписи памяти другими программами

Среда, передаваемая программе

1. Что такое среда?

Среда - совокупность значений системных переменных, путей, открытых файловых дескрипторов и других ресурсов операционной системы, передаваемые процессу (программе) при его запуске.

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

Среда создается при запуске операционной системы, но потом копируется в адресное пространство запущенной программе и уже там может быть изменена, если это необходимо программе. В программу, запущенной другой программы копируется среда запустившей(родительской) программы.

3. Откуда берется информация, записываемая в среду?

Она достается из файла Autoexec.BAT, который располагается в корневом каталоге загрузочных устройств.

Вывод:

В результате выполнения работы был исследован интерфейс управляющей программы и загрузочных модулей, а также исследован префикс сегмента программы (PSP) и среды, передаваемой программе.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab2.asm

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

; Данные
UN_MEM db 'Adress of unavailable memory: ',0DH,0AH,$'
ADRESS_MEDIUM db 'The address of the medium passed to the program: ',0DH,0AH,$'
TAIL db 'Command line tail: ', '$'
CONTENT db 'Environment area content: ',0DH,0AH,$'
PATH db 'Loadable module path: ',0DH,0AH,$'
EMPTY db 'Empty',0DH,0AH,$'
NEWLINE 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
; Байт в AL переводится в два символа шестн. числа AX
    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
;-----
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

;-----

WRITE_STRING PROC near; Вывод строки текста

```
    mov AH,09h
    int 21h
    ret
```

WRITE_STRING ENDP

UN_MEM_WRITE PROC near

```
    mov AX, DS:[02h]
    mov DI, offset UN_MEM
    add DI, 33
    call WRD_TO_HEX
    mov DX, offset UN_MEM
    call WRITE_STRING
    ret
```

UN_MEM_WRITE ENDP

ADRESS_MEDIUM_WRITE PROC near

```
    mov AX, DS:[02Ch]
    mov DI, offset ADRESS_MEDIUM
    add DI, 53
    call WRD_TO_HEX
    mov DX, offset ADRESS_MEDIUM
    call WRITE_STRING
    ret
```

ADRESS_MEDIUM_WRITE ENDP

TAIL_WRITE PROC near

```
XOR CX,CX
MOV SI, 0
MOV AH, 02H
MOV DX, offset TAIL
CALL WRITE_STRING
MOV CL, DS:[80h]
cmp CL, 0
jg TAIL_MARK
MOV DX, offset EMPTY
CALL WRITE_STRING
RET
```

TAIL_MARK:

```
MOV DL, DS:[81h+SI]
MOV AH, 02h
INT 21h
INC SI
loop TAIL_MARK
MOV DX, offset NEWLINE
CALL WRITE_STRING
RET
```

TAIL_WRITE ENDP

CONTENT_WRITE PROC near

```
MOV DX, OFFSET CONTENT
CALL WRITE_STRING
MOV AX, DS:[02Ch]
MOV ES, AX
MOV DI, 0
```

ZERO_1:

```
MOV DL, ES:[DI]
CMP DL, 0
JE ZERO_2
```

PRINT_CHAR:

```
MOV AH, 02h
INT 21h
INC DI
JMP ZERO_1
```

```

ZERO_2:
    MOV DX, OFFSET NEWLINE
    CALL WRITE_STRING
    INC DI
    MOV DL, ES:[DI]
    CMP DL, 0
    JNE PRINT_CHAR
    ret
CONTENT_WRITE ENDP

PATH_WRITE PROC near
    MOV DX, OFFSET PATH
    CALL WRITE_STRING
    ADD DI, 3

PATH_MARK:
    MOV DL, ES:[DI]
    CMP DL, 0
    JE END_PATH
    MOV AH, 02h
    INT 21H
    INC DI
    JMP PATH_MARK

END_PATH:
    ret
PATH_WRITE ENDP
;-----
; КОД
BEGIN:
    CALL UN_MEM_WRITE
    CALL ADRESS_MEDIUM_WRITE
    CALL TAIL_WRITE
    CALL CONTENT_WRITE
    CALL PATH_WRITE
;.....
; Выход в DOS
    xor AL,AL
    mov AH,4Ch
    int 21H
TESTPC ENDS

```

END START ; Конец модуля, START - точка входа