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 загрузочных модулей

Студент гр. 9381 _____ Прибылов Н.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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

Ход работы.

1. Был написан и отлажен .COM модуль, который выбирает и распечатывает следующую информацию:

а) Сегментный адрес недоступной памяти, взятый из PSP, в шестнадцатеричном виде.

б) Сегментный адрес среды, передаваеой программе, в шестандцатеричном виде.

в) Хвост командной строки в символьном виде.

г) Содержимое области среды в символьном виде.

д) Путь загружаемого модуля.

Результат работы программы:

Функции.

Названия	Описание
TETR_TO_HEX	Перевод тетрады (4 младших бита AL) в 16-ю систему. Результат в AL.
BYTE_TO_HEX	Перевод байта AL в 16-ю систему. Результат: старшая цифра в AL, младшая в AX.
WRD_TO_HEX	Перевод слова AX в 16-ю систему. Адрес последнего символа результата в DI.
BYTE_TO_DEC	Перевод байта AL в 10-ю систему. Адрес младшей цифры результата в SI.
PRINT	Вывод строки из DX на экран.

Контрольные вопросы.

Сегментный адрес недоступной памяти

1) На какую область памяти указывает адрес недоступной памяти?

На ту, которая следует сразу после области памяти, отведённой программе.

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

Сразу после памяти, выделенной программе. Расположен в сторону увеличения адресов.

3) Можно ли в эту область памяти писать?

Да, так как в DOS нет защиты памяти.

Среда, передаваемая программе

1) Что такое среда?

Среда представляет собой последовательность нуль-терминированных строк вида *параметр=значение*, представляющие собой системные переменные, которые могут понадобиться программе для её работы.

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

Среда создаётся при запуске ОС. При запуске программы ей передаётся копия среды ОС. Если загруженная программа запускает дочерний процесс, она передаёт этому процессу свою копию среды.

3) Откуда берётся информация, записываемая в среду?

Из системного файла AUTOEXEC.BAT.

Выводы.

Были исследованы интерфейсы управляющей программы и загрузочных модулей, а так же префикса 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
endline                                 db  0dh,0ah,'$'
inaccessible_memory_address_string    db    '>Inaccessible  memory
address: $'
inaccessible_memory_address         db  '    ',0dh,0ah,'$'
environment_address_string          db  '>Environment address: $'
environment_address                 db  '    ',0dh,0ah,'$'
command_line_tail_string                      db    '>Command  line
tail:',0dh,0ah,'$'
environment_content_string              db   '>Environment content:
',0dh,0ah,'$'
module_path_string                            db    '>Module  path:
',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
;--------------------------------------------------
BYTE_TO_DEC PROC near
; Перевод AL 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_l
        or AL,30h
        mov [SI],AL
end_l:
        pop DX
        pop CX
        ret
BYTE_TO_DEC ENDP
;------------------------------
PRINT   PROC    near
        push    ax
        mov     ah, 09h
        int     21h
        pop     ax
        ret
PRINT   ENDP
;------------------------------

; КОД
BEGIN:
print_inaccessible_memory_address:
        lea     dx, inaccessible_memory_address_string
        call    PRINT
        mov     ax, es:[2h]
        lea     di, inaccessible_memory_address + 3
        call    WRD_TO_HEX
        lea     dx, inaccessible_memory_address
        call    PRINT
```

```
print_environment_address:
        lea     dx, environment_address_string
        call    PRINT
        mov     ax, es:[2ch]
        lea     di, environment_address + 3
        call    WRD_TO_HEX
        lea     dx, environment_address
        call    PRINT

print_command_line_tail:
        lea     dx, command_line_tail_string
        call    PRINT
        mov     bx, 81h
        xor     ch, ch
        mov     cl, es:[80h]
        cmp     cl, 0
        je      print_environment_content
tail_loop:
        mov     dl, es:[bx]
        mov     ah, 02h
        int     21h
        inc     bx
        loopnz  tail_loop
        lea     dx, endline
        call    PRINT

print_environment_content:
        push    es
        lea     dx, environment_content_string
        call    PRINT
        mov     es, es:[2ch]
        mov     cx, 0
        mov     bx, 0
        cmp     es:[bx], cx
        je      print_module_path
environment_loop:
        mov     dl, es:[bx]
        mov     ah, 02h
        int     21h
        inc     bx
        cmp     es:[bx], cl
        jne     environment_loop
        lea     dx, endline
        call    PRINT
        inc     bx
        cmp     es:[bx], cl
        jne     environment_loop

print_module_path:
        lea     dx, module_path_string
        call    PRINT
        add     bx, 3
module_path_loop:
        mov     dl, es:[bx]
        mov     ah, 02h
        int     21h
        inc     bx
```

```
        cmp     es:[bx], cl
        jne     module_path_loop
        pop     es

exit:
        mov     ah, 4ch
        int     21H
TESTPC  ENDS
        END START ; Конец модуля, START - точка входа
```