**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6381 |  | Сергухин В.Ю. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Постановка задачи**

**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.

**Необходимые сведения для составления программы.**

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. Именно по этой причине значения этих регистров в модуле .EXE следует переопределять.

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, PROMT, 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.

**Сведения о функциях и структурах данных управляющей программы.**

Сведения о функциях:

|  |  |
| --- | --- |
| Название | Назначение |
| TETR\_TO\_HEX | перевод полвины байта из 2-ой СС в 16-ую СС |
| BYTE\_TO\_HEX | перевод байта, расположенного в AL, в 2 символа 16-ой числа в AX |
| WRD\_TO\_HEX | перевод в 16-ую СС 16-ти разрядного числа |
| BYTE\_TO\_DEC | перевод в 10-ую СС |
| print\_addr\_env | Определение и печать начала сегментного адреса среды |
| print\_tail\_of\_cmd\_str | Определение и печать хвоста командной строки |
| print\_path\_to\_prog | определение и печать содержимого области среды и пути загружаемого модуля |

Сведения о структурах данных:

|  |  |  |
| --- | --- | --- |
| Название | Тип | Назначение |
| ADDR\_MEM | db | Адрес недоступной памяти |
| ADDR\_ENV | db | Адрес среды |
| TAIL\_OF\_CMD | db | Хвост командной строки |
| CONTENT | db | Содержимое области среды |
| ENV | db | Путь загружаемого модуля |
| NEW\_LINE | db | Переход на новую строку |

**Последовательность действий, выполняемых программой.**

Модуль .СОМ выбирает и выводит на экран:

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.

**Результат работы программы.**

****

рис.1 Результат работы программы с пустым хвостом CMD

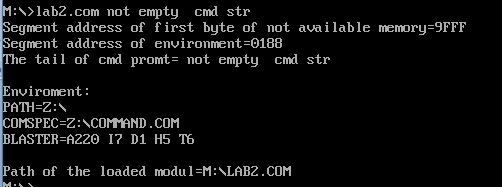
****

рис.1 Результат работы программы с непустым хвостом CMD

**Описание результатов исследования проблем, поставленных в лабораторной работе**

**Сегментный адрес недоступной памяти.**

1. На какую область памяти указывает адрес недоступной памяти?

Ответ: Адрес недоступной памяти указывает на конец оперативной памяти и на границу области, доступной для загрузки программ, за ним располагается ROM BIOS;

1. Где расположен этот адрес по отношению области памяти, отведённой программе?

Ответ: Адрес расположен сразу за областью памяти, отведенной программе (начиная с адреса 9FFF);

1. Можно ли в эту область памяти писать?

Ответ: При условии отсутствия в управляющей программе операционной системы механизма защиты памяти (как в DOS), запись в эту область возможна.

**Среда, передаваемая программе.**

1. Что такое среда?

Ответ: среда – это область памяти, содержащая переменные, в которых хранятся данные о некоторых директориях операционной системы и конфигурации компьютера, записанные в виде строк формата *имя=параметр*;

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

Ответ: среда создается при загрузке DOS, перед запуском приложения эта среда только копируется в новую область памяти;

1. Откуда берется информация, записываемая в среду?

Ответ: информация, записываемая в среду, берется из системного файла AUTOEXEC.BAT.

**Заключение**

В ходе работы было проведено 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 (РSР) и помещает его адрес в сегментный регистр.

Приложения

**Код программы** **lab2**.**asm**

CODE SEGMENT

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

ORG 100H

START: JMP BEGIN

; данные

ADDR\_MEM DB 'Segment address of first byte of not available memory= ', 0AH, 0DH,'$'

ADDR\_ENV DB 'Segment address of environment= ', 0AH, 0DH, '$'

TAIL\_OF\_CMD DB 'The tail of cmd promt=' , '$'

NEW\_LINE DB 0AH, 0DH, '$'

ENV DB 0AH, 0DH,'Enviroment:', 0AH, 0DH, '$'

PATH\_TO\_PROG DB 'Path of the loaded modul=', '$'

; 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 ;в AL старшая цифра

pop CX ;в AH младшая

ret

BYTE\_TO\_HEX ENDP

WRD\_TO\_HEX PROC near ;перевод в 16 с/с 16-ти разрядного числа

push BX ; в AX - число, DI - адрес последнего символа

mov BH,AH ; now it aclually converts byte to string, last sybmol adress is di

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 байта в 10с/с, SI - адрес поля младшей цифры

push AX ; AL содержит исходный байт

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

pop AX

ret

BYTE\_TO\_DEC ENDP

print\_addr\_env proc near

mov dx, offset ENV

mov ah, 09h

int 21h

mov bx, 2Ch

mov ax, [bx]

mov ds, ax

mov di,0

mov dx, di

cicl:

cmp byte ptr [di], 0

je exit

cf:

inc di

cmp byte ptr [di], 0

jne cf

mov byte ptr [di], '$'

mov ah, 09h

int 21h

push dx

push ds

push es

mov dx, offset NEW\_LINE

pop ds

int 21h

pop ds

pop dx

mov byte ptr [di], 0h ; восстанвливаем 0

inc di

mov dx, di

jmp cicl

exit:

push es

pop ds

mov dx, offset NEW\_LINE

mov ah, 09h

int 21h

ret

print\_addr\_env endp

print\_tail\_of\_cmd\_str proc near

mov dx, offset TAIL\_OF\_CMD

mov ah, 09h

int 21h

mov bx, 80h ;число символов в строке

mov al, [bx]

cmp al, 0

je empty

mov ah, 0

mov di, ax

mov al, [di+81h]

push ax

mov byte ptr [di+81h], '$'

mov dx, 81h

mov ah, 09h

int 21h

pop ax

mov [di+81h], al

empty:

mov dx, offset NEW\_LINE

mov ah, 09h

int 21h

ret

print\_tail\_of\_cmd\_str endp

print\_path\_to\_prog proc near ; В DI хранится адрес конца среды

mov dx, offset PATH\_TO\_PROG

mov ah, 09h

int 21h

mov bx, 2Ch

mov ax, [bx]

mov ds, ax

add di, 3

mov dx, di

find:

inc di

cmp byte ptr [di], 0

jne find

mov byte ptr [di], '$'

mov ah, 09h

int 21h

mov byte ptr [di], 0h ; восстанвливаем 0

push es

pop ds

ret

print\_path\_to\_prog endp

BEGIN:

push DS

sub AX,AX

push AX

; 1)Сегметный адрес недоступной памяти

mov bx, 2h

mov ax, [bx]

mov DI, OFFSET (ADDR\_MEM+57)

call WRD\_TO\_HEX

mov dx, offset ADDR\_MEM

mov ah, 09h

int 21h

; 2)Сегметный адрес среды

mov bx, 2ch

mov ax, [bx]

mov DI, OFFSET (ADDR\_ENV+34)

call WRD\_TO\_HEX

mov dx, offset ADDR\_ENV

mov ah, 09h

int 21h

; 3)Хвост командной строки

call print\_tail\_of\_cmd\_str

; 4)Содержимое области среды

call print\_addr\_env

; 5)Путь загружаемого модуля

call print\_path\_to\_prog

; Выход в DOS

xor AL,AL

mov AH,4Ch

int 21H

CODE ENDS

END START