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. 9381 _____ Давыдов Д.С.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

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.

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 РЭР, 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.

Сохраните результаты, полученные программой, и включите их В отчет.

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

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 РЭР. При загрузке модуля типа .EXE сегментные регистры DS и ES указывают на РЭР. 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 COMMANDCOM. Следующие строки содержат информацию, задаваемую командами PATH, PROMPT, 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.

Выполнение работы.

Был написан и отлажен программный модуль типа .COM, который выводит на экран 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

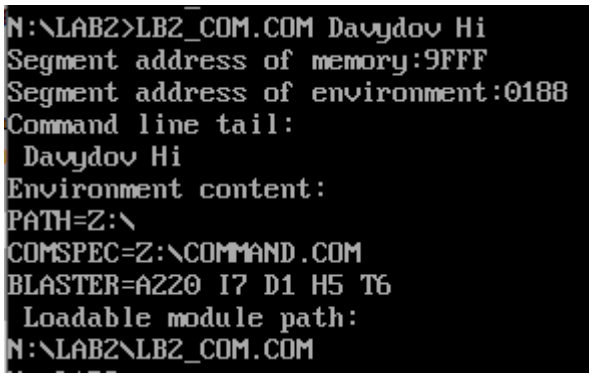
4) Содержимое области среды в символьном виде

5) Путь загружаемого модуля

После запуска COM модуля на экран вывелись данные (рис 1), без введенной пользователем строки (рис. 1) и с ней (рис. 2):

Рисунок 1.

Рисунок 2.



Разработанный программный код смотреть в приложении А.

Ответы на контрольные вопросы.

Сегментный адрес недоступной памяти

1) На какую область памяти указывает адрес недоступной памяти?

Ответ: на сегментный адрес первого байта за памятью, отведенной программе.

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

Ответ: он расположен сразу после области памяти, отведенной программе - с адреса 9FFF.

3) Можно ли в эту область памяти писать?

Ответ: Можно, используя адресацию для сегментного регистра.

Среда передаваемая программе

1) Что такое среда?

Ответ: Среда — это область памяти, в которой в виде символьных строк записаны значения переменных в формате «имя = значение» и байт нулей.

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

Ответ: она создается при загрузке MS DOS. При запуске программы среда только копируется в новую область памяти.

3) Откуда берется информация, записываемая в среду?

Ответ: копируется из системного файла autoexec.bat, расположенного в корневом каталоге загрузочного устройства.

Выводы.

Был исследован интерфейс управляющей программы и загрузочных модулей, префикс сегмента программы и среда, которая передается программам.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.asm

```
TESTPC SEGMENT

ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
       ORG 100H

START: JMP MAIN

;ДАННЫЕ
segAddrUnvailbleMemory     db    'Segment      address      of      memory:
',0DH,0AH,'$'
segAddressEnvironment db    'Segment      address      of      environment:
',0DH,0AH,'$'
tailCommandString          db    'Command line tail: ',0DH,0AH,'$'
noTail          db    'Tail is empty',0DH,0AH,'$'
tailInfo   db    '  $'
environmentContent    db    'Environment content:   ',0DH,0AH,'$'
newLine         db    0DH,0AH,'$'
path       db    'Loadable module path:',0DH,0AH,'$'

;Представление 4 бита регистра al в виде цифры 16ой с.с. и представление
её в символьном виде.
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

;Представление al как два числа в 16-ой с.с. и перемещение их в ax
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

; перевод в 16 с.с 16-ти разрядного числа
; в AX - число, а в DI - адрес последнего символа
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

; перевод в 10 с/с. SI - адрес поля младшей цифры
BYTE_TO_DEC PROC near
      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
      mov  ah, 09h
      int 21h
      ret
PRINT      endp

MAIN:
;Сегментный адрес недоступной памяти
      mov  ax, ds:[02h]
      mov  di, offset segAddrUnvailbleMemory
      add  di, 29
      call wrd_to_hex
      mov dx, offset segAddrUnvailbleMemory
      call print

;Сегментный адрес среды
      mov  ax, ds:[2ch]
      mov  di, offset segAddressEnvironment
      add  di, 34
      call wrd_to_hex
      mov  dx, offset segAddressEnvironment
      call  print

;Хвост командной строки
      mov  dx, offset tailCommandString
```


8

```
      call print
      mov  cl, ds:[80h]
      cmp  cl, 0
      je   empty
      xor  si, si

tailLoop:
      mov  dl, ds:[81h+si]
      mov  ah, 02h
      int  21h
      inc  si
      dec  cl
      cmp  cl, 0
      jne tailloop
      mov  dx, offset newLine
      call print
      jmp envcon

empty:
      mov  dx, offset noTail
      call  print

envcon:
;Содержимое области среды
      mov  dx, offset environmentContent
      call print
      mov  es, ds:[2ch]
      xor  si, si
printStr:
      mov  al, es:[si]
      cmp  al, 0
      jne  printSymbol
      inc si
      mov  al, es:[si]
      mov  dx, offset newLine
      call print
printSymbol:
      mov  dl, al
      mov  ah, 02h
      int  21h
```


9

```
      inc   si
      mov   ax, es:[si]
      cmp   ax, 0001
      jne   printStr


;Путь загружаемого модуля
      mov   dx, offset path
      call print
      add   si, 2
printSymb:
      mov al, es:[si]
      cmp al, 0
      je    exit
      mov   dl, al
      mov ah, 02h
      int   21h
      inc   si
      jmp   printSymb


exit: ; Выход в DOS
      xor AL,AL
      mov AH,4Ch
      int 21H
TESTPC ENDS
      END START
```


10