

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. 7383

Лосев М.Л.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2018

Постановка задачи.

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

Сведения об используемых функциях и структурах данных.

BYTE_TO_HEX – переводит значение регистра AL в его запись в шестнадцатеричной с/с, помещает ее в AX

WRD_TO_HEX – переводит значение регистра AX в его запись в шестнадцатеричной с/с, помещает ее в память так, что DI указывает на младшую цифру.

BYTE_TO_DEC – переводит значение регистра AL в его запись в десятичной с/с, помещает результат в память так, что SI указывает на младшую цифру.

OUTPUT_PROC – вызывает прерывание DOS вывода строки.

PRINT_UNAVAIL_MEM – получает и выводит адрес первого байта недоступной памяти

PRINT_SEG_ENV – получает и выводит сегментный адрес среды

PRINT_TAIL – получает и выводит хвост командной строки

PRINT_ENV_CONT – получает и выводит содержимое области среды

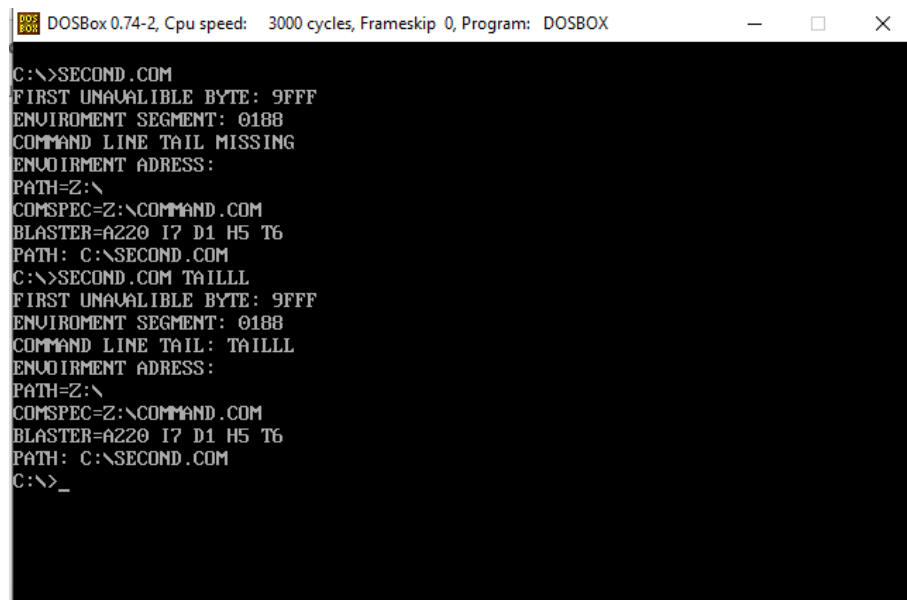
PRINT_PATH – получает и выводит путь загружаемого модуля

Последовательность действий, выполняемых утилитой.

Утилита получает адрес первого байта недоступной памяти по смещению 02Ch, копирует его в регистр, формирует строку, содержащую его шестнадцатеричную запись, и выводит ее на экран. Так же она выводит сегментный адрес среды. Потом выводит хвост командной строки (он начинается с 81h байта, его длинна хранится в байте 80h). После этого по адресу области среды (он хранится в байте 02Ch) выводит содержимое области среды, пока не

встретит два нулевых байта, которыми заканчивается область. За областью среды идет путь исполняемого модуля, который кончается нулем. Его программа тоже выводит.

Скриншот с запуском программы и результатами представлен на рис. 1.



```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
C:\>SECOND.COM
FIRST UNAVAILABLE BYTE: 9FFF
ENVIRONMENT SEGMENT: 018B
COMMAND LINE TAIL MISSING
ENVIRONMENT ADDRESS:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
PATH: C:\SECOND.COM
C:\>SECOND.COM TAILL
FIRST UNAVAILABLE BYTE: 9FFF
ENVIRONMENT SEGMENT: 018B
COMMAND LINE TAIL: TAILL
ENVIRONMENT ADDRESS:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
PATH: C:\SECOND.COM
C:\>_
```

Рисунок 1. Результаты запуска программы

Результаты исследования.

Сегментный адрес недоступной памяти:

1) На какую область памяти указывает адрес недоступной памяти?

На основную область, точнее на ее последний параграф. Следующий за ним параграф уже находится в недоступной памяти.

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

В соответствии со стратегией однопрограммных ОС вся доступная память при загрузке исполняемого модуля выделяется этому модулю. Поэтому этот адрес указывает на последний параграф области памяти, отведенной программе.

3) Можно ли в эту область писать?

Да, можно, потому что в DOS нет защиты памяти.

Среда, передаваемая программе:

1) Что такое среда?

Среда – это набор значений некоторых переменных, определяемых ОС, то есть данные ОС. Область среды содержит строки, описывающие среду.

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

Среда создается при запуске ОС. Область среды формируется перед запуском приложения.

3) Откуда берется информация, записываемая в среду?

Ее формирует файл autoexec.bat.

Заключение.

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 (PSP) и помещает его адрес в сегментный регистр. Был исследован префикс сегмента программы (PSP) и среды, передаваемой программе. Были исследованы области, на которые делится сегмент программы.