

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Операционные системы»
Тема: Исследование структур загрузочных модулей

Студент гр. 0381

Печеркин А. С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Исследование различий в структурах исходных текстов типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Задание.

Напишите текст исходного .COM модуля, который определяет тип PC и версию системы.

Ассемблерная программа должна читать содержимое предпоследнего байта ROM BIOS, по таблице, сравнивая коды, определять тип PC и выводить строку с названием модели. Если код не совпадает ни с одним значением, то двоичный код переводиться в символьную строку, содержащую запись шестнадцатеричного числа и выводиться на экран в виде соответствующего сообщения.

Затем определяется версия системы. Ассемблерная программа должна по значениям регистров AL и AH формировать текстовую строку в формате xx.yy, где xx – номер основной версии, а yy - номер модификации в десятичной системе счисления, формировать строки с серийным номером OEM и серийным номером пользователя. Полученные строки выводятся на экран.

Результатом выполнения этого шага будет «хороший» .COM модуль, а также необходимо построить «плохой» .EXE, полученный из исходного текста для .COM модуля.

Напишите текст исходного .EXE модуля, который выполняет те же функции и постройте его. Таким образом будет получен хороший .EXE модуль.

Сравните исходные тексты для .COM и .EXE модулей. Сравните файлы .COM, «плохого» и «хорошего» .EXE модулей в шестнадцатеричном виде.

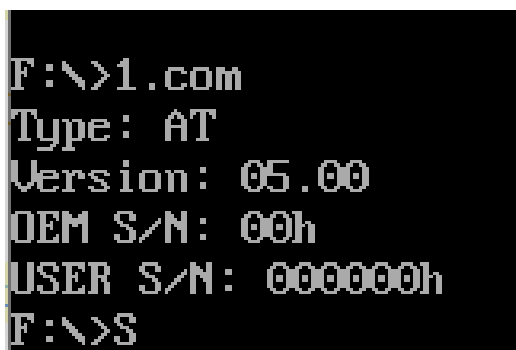
Выполнение работы.

Для написания исходного текста .COM модуля был использован шаблон из методических указаний. Были добавлены строки с названиями моделей для последующего вывода на экран.

При запуске программы выполняется переход на метку BEGIN, где происходит считывание байта, расположенного по адресу F000:FFFEh и содержащего информацию о модели компьютера. Затем этот байт последовательно сравнивается с значениями из таблицы в методических указаниях. Если обнаружено совпадение, выводится строка соответствующая данному коду модели, иначе выводится значение в шестнадцатеричном виде.

Для получения информации о версии DOS используется функция 30h прерывания 21h. Полученные значения переводятся в требуемый формат и выводятся на экран.

Результат работы программы:



```
F:\>1.com
Type: AT
Version: 05.00
OEM S/N: 00h
USER S/N: 000000h
F:\>S
```

Если из этого исходного кода построить .EXE модуль, он будет работать некорректно:

```

F:\>1.exe

      00 Type:

      00 Type:      5 0

      00 Type:

e: 00      5 0      00 Type:
Type:      000000      00
F:\>S

```

Для того, чтобы построить правильный .EXE модуль необходимо разделить программу на сегменты. Для этого в начале исходного текста добавляется описание сегмента стека, а данные и код помещаются в собственные сегменты. Также отсутствует необходимость в директиве ORG 100h, так как код теперь находится в отдельном сегменте. Собранный из этого кода .EXE модуль выводит информацию о системе так же, как и .COM модуль.

```

F:\>11.exe
Type: AT
Version: 05.00
OEM S/N: 00h
USER S/N: 000000h
F:\>S_

```

Ответы на вопросы см. в разделе «Вопросы».

Выводы.

Были исследованы различия в структуре исходных текстов для модулей .COM и .EXE, структура загрузочных файлов этих типов и способы загрузки их в основную память.

ВОПРОСЫ

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать только один сегмент, в котором содержится PSP, данные, код и стек программы.

2. EXE-программа?

Такая программа должна содержать как минимум один сегмент – сегмент кода. Также .EXE модуль может содержать сегменты стека и данных.

3. Какие директивы должны обязательно быть в тексте COM-программы?

ORG 100h – пропускает первые 256 байт сегмента для размещения в них префикса программного сегмента (PSP).

ASSUME – для указания, что этот сегмент будет использоваться в качестве сегмента кода и сегмента данных.

4. Все ли форматы команд можно использовать в COM-программе?

Не поддерживаются команды с указанием сегментов, так как в .COM модуле отсутствует таблица настройки адресов(relocation table). Так как в программе сегментные адреса задаются относительно начала программы, необходимо учитывать смещение начального сегмента программы, для этого используется таблица настройки адресов.

Отличия форматов файлов COM и EXE модулей

1. Какова структура файла COM? С какого адреса располагается код?

COM файл содержит в один сегмент с кодом и данными, размер файла не может превышать 65280 байт. Код располагается с адреса 0, так как в этом файле отсутствует заголовок и таблица настройки адресов.

| Address | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | Dump |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------|
| 00000000 | e9 | ec | 00 | 54 | 79 | 70 | 65 | 3a | 20 | 24 | 50 | 43 | 0d | 0a | 24 | 50 | им.Type: \$PC..\$P |
| 00000010 | 43 | 2f | 58 | 54 | 0d | 0a | 24 | 41 | 54 | 0d | 0a | 24 | 50 | 53 | 32 | 20 | C/XT..\$AT..\$PS2 |
| 00000020 | 6d | 6f | 64 | 65 | 6c | 20 | 33 | 30 | 0d | 0a | 24 | 50 | 53 | 32 | 20 | 6d | model 30..\$PS2 m |
| 00000030 | 6f | 64 | 65 | 6c | 20 | 35 | 30 | 20 | 6f | 72 | 20 | 36 | 30 | 0d | 0a | 24 | odel 50 or 60..\$ |
| 00000040 | 50 | 53 | 32 | 20 | 6d | 6f | 64 | 65 | 6c | 20 | 38 | 30 | 0d | 0a | 24 | 50 | PS2 model 80..\$P |
| 00000050 | 43 | 6a | 72 | 0d | 0a | 24 | 50 | 43 | 20 | 43 | 6f | 6e | 76 | 65 | 72 | 74 | Cjr..\$PC Convert |
| 00000060 | 69 | 62 | 6c | 65 | 0d | 0a | 24 | 68 | 0d | 0a | 24 | 56 | 65 | 72 | 73 | 69 | ible..\$h..\$Versi |
| 00000070 | 6f | 6e | 3a | 20 | 30 | 30 | 2e | 30 | 30 | 0d | 0a | 24 | 4f | 45 | 4d | 20 | on: 00.00..\$OEM |
| 00000080 | 53 | 2f | 4e | 3a | 20 | 24 | 55 | 53 | 45 | 52 | 20 | 53 | 2f | 4e | 3a | 20 | S/N: \$USER S/N: |
| 00000090 | 30 | 30 | 30 | 30 | 30 | 30 | 68 | 24 | 24 | 0f | 3c | 09 | 76 | 02 | 04 | 07 | 000000h\$.<.v... |
| 000000a0 | 04 | 30 | c3 | 51 | 8a | e0 | e8 | ef | ff | 86 | c4 | b1 | 04 | d2 | e8 | e8 | .0ГQБиппл+Д±.Тии |
| 000000b0 | e6 | ff | 59 | c3 | 53 | 8a | fc | e8 | e9 | ff | 88 | 25 | 4f | 88 | 05 | 4f | жяYTSЪийяё%Оё..О |
| 000000c0 | 8a | c7 | e8 | de | ff | 88 | 25 | 4f | 88 | 05 | 5b | c3 | 51 | 52 | 32 | e4 | ЪзиЮяё%Оё. [ГQR2д |
| 000000d0 | 33 | d2 | b9 | 0a | 00 | f7 | f1 | 80 | ca | 30 | 88 | 14 | 4e | 33 | d2 | 3d | ЗТН...чсЪК0ё..NЗТ= |
| 000000e0 | 0a | 00 | 73 | f1 | 3c | 00 | 74 | 04 | 0c | 30 | 88 | 04 | 5a | 59 | c3 | b8 | ..sc<.t...0ё..2YГё |
| 000000f0 | 00 | f0 | 8e | c0 | 26 | 8a | 1e | fe | ff | ba | 03 | 01 | b4 | 09 | cd | 21 | .рЪА&Ъ..юяё...г.Н! |
| 00000100 | 80 | fb | ff | 74 | 2b | 80 | fb | fe | 74 | 2c | 80 | fb | fb | 74 | 27 | 80 | Ъыят+Ъыют,Ъыът'Ъ |
| 00000110 | fb | fc | 74 | 28 | 80 | fb | fa | 74 | 29 | 80 | fb | fc | 74 | 2a | 80 | fb | ыът(Ъыът)Ъыът*Ъы |
| 00000120 | f8 | 74 | 2b | 80 | fb | fd | 74 | 2c | 80 | fb | f9 | 74 | 2d | eb | 38 | 90 | шт+Ъыът,Ъыът-л8. |
| 00000130 | ba | 0a | 01 | eb | 2b | 90 | ba | 0f | 01 | eb | 25 | 90 | ba | 17 | 01 | eb | е...л+.е...л%.е...л |
| 00000140 | 1f | 90 | ba | 1c | 01 | eb | 19 | 90 | ba | 2b | 01 | eb | 13 | 90 | ba | 40 | ..е...л+.е...л%.е8 |
| 00000150 | 01 | eb | 0d | 90 | ba | 4f | 01 | eb | 07 | 90 | ba | 56 | 01 | eb | 01 | 90 | .л..еО.л...еV.л.. |
| 00000160 | b4 | 09 | cd | 21 | eb | 17 | 90 | 8a | c3 | e8 | 37 | ff | 8b | d0 | b4 | 02 | г.Н!л...ЪГи7я<Рг. |
| 00000170 | cd | 21 | 86 | f2 | cd | 21 | ba | 67 | 01 | b4 | 09 | cd | 21 | 8c | d8 | 8e | Н!†Н!ег.г.Н!ЪШП |
| 00000180 | c0 | b4 | 30 | cd | 21 | 8b | d0 | be | 0a | 00 | 81 | c6 | 6b | 01 | e8 | 3b | Аг'ОН!<Рс...Жж.и; |
| 00000190 | ff | 8a | c6 | 83 | c6 | 03 | e8 | 33 | ff | ba | 6b | 01 | b4 | 09 | cd | 21 | яЪЖ&Ж.иЗяек.г.Н! |
| 000001a0 | ba | 7c | 01 | b4 | 09 | cd | 21 | 8a | c7 | e8 | f7 | fe | 8b | d0 | b4 | 02 | е .г.Н!ЪзичюРг. |

2. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?

Такой файл содержит заголовок, таблицу настройки адресов и один сегмент, в котором находятся данные и код. Код располагается с адреса 300h (200h – заголовок и таблица настроек + 100h смещение ORG 100h). С адреса 0 располагается заголовок EXE файла. Заголовок содержит сигнатуру EXE файла, длину образа программы, размер таблицы настройки, сегментный адрес стека, адрес точки входа, а также ряд других параметров, необходимых для загрузки.

| Address | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | Dump |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|
| 00000000 | 4d | 5a | de | 00 | 03 | 00 | 00 | 00 | 20 | 00 | 00 | 00 | ff | ff | 00 | 00 | MZЮ..... ..яя.. |
| 00000010 | 00 | 00 | 4b | 2b | 00 | 01 | 00 | 00 | 1e | 00 | 00 | 00 | 01 | 00 | 00 | 00 | ..K+..... |
| 00000020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |

```

00000270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000300 e9 ec 00 54 79 70 65 3a 20 24 50 43 0d 0a 24 50 йм.Type: $PC..$P
00000310 43 2f 58 54 0d 0a 24 41 54 0d 0a 24 50 53 32 20 C/XT..$AT..$PS2
00000320 6d 6f 64 65 6c 20 33 30 0d 0a 24 50 53 32 20 6d model 30..$PS2 m
00000330 6f 64 65 6c 20 35 30 20 6f 72 20 36 30 0d 0a 24 odel 50 or 60..$
00000340 50 53 32 20 6d 6f 64 65 6c 20 38 30 0d 0a 24 50 PS2 model 80..$P
00000350 43 6a 72 0d 0a 24 50 43 20 43 6f 6e 76 65 72 74 Cjr..$PC Convert
00000360 69 62 6c 65 0d 0a 24 68 0d 0a 24 56 65 72 73 69 ible..$h..$Versi
00000370 6f 6e 3a 20 30 30 2e 30 30 0d 0a 24 4f 45 4d 20 on: 00.00..$OEM
00000380 53 2f 4e 3a 20 24 55 53 45 52 20 53 2f 4e 3a 20 S/N: $USER S/N:
00000390 30 30 30 30 30 30 68 24 24 0f 3c 09 76 02 04 07 000000h$.<.v...
000003a0 04 30 c3 51 8a e0 e8 ef ff 86 c4 b1 04 d2 e8 e8 .0ГQЛайп+Д±.Тии
000003b0 e6 ff 59 c3 53 8a fc e8 e9 ff 88 25 4f 88 05 4f жяYTSЛыйя%О€.О
000003c0 8a c7 e8 de ff 88 25 4f 88 05 5b c3 51 52 32 e4 ЪзиЮя%О€. [ГQR2д
000003d0 33 d2 b9 0a 00 f7 f1 80 ca 30 88 14 4e 33 d2 3d ЗТН..чсЪКО€.NЗТ=
000003e0 0a 00 73 f1 3c 00 74 04 0c 30 88 04 5a 59 c3 b8 ..sc<.t..0€.ZYГё
000003f0 00 f0 8e c0 26 8a 1e fe ff ba 03 01 b4 09 cd 21 .рТА&Ъ.юяе..г.Н!
00000400 80 fb ff 74 2b 80 fb fe 74 2c 80 fb fb 74 27 80 Ыят+Ыют, Ыыт'Ъ
00000410 fb fc 74 28 80 fb fa 74 29 80 fb fc 74 2a 80 fb ыт (Ыгът) Ыыт*Ы

```

3. Какова структура файла «хорошего» EXE? Чем он отличается от файла «плохого» EXE.

«Хороший» EXE также содержит заголовок и таблицу настройки адресов(общая длина 200h). После таблицы идет образ сегмента стека, сегмента данных и сегмента кода, в отличие от всего одного сегмента в «плохом» EXE файле.

| Address | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | Dump |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|
| 00000000 | 4d | 5a | eb | 00 | 03 | 00 | 01 | 00 | 20 | 00 | 00 | 00 | ff | ff | 00 | 00 | MЗл..... ..яя.. |
| 00000010 | 00 | 01 | 68 | 1f | 57 | 00 | 1a | 00 | 1e | 00 | 00 | 00 | 01 | 00 | 58 | 00 | ..h.W...X. |
| 00000020 | 1a | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |

Как видно на рисунке, заголовок хорошего файла отличается, например, изменился сегментный адрес точки входа (выделен на рисунке). В «плохом» файле он был 0000.

Загрузка COM модуля в основную память

1. Какой формат загрузки модуля COM? С какого адреса располагается код?

Выделяется свободный сегмент памяти и его адрес заносится в сегментные регистры. Затем в первые 256 байт этого сегмента записывается PSP. Непосредственно за ним с диска загружается содержимое COM-файла без изменений. Указатель стека устанавливается на конец этого сегмента, и в стек записывается адрес возврата (начало PSP - 0000h). Управление передается по адресу CS:100h.

2. Что располагается с адреса 0?

С адреса 0 располагается префикс программного сегмента (PSP).

3. Какие значения имеют сегментные регистры? На какие области памяти они указывают?

Как видно на скриншоте, все сегментные регистры имеют одно и то же значение (в данном случае 48DD), это и есть сегмент в который загружена программа.

The screenshot shows a debugger window for a CPU 80486. The main window displays assembly code with addresses and instructions. The right-hand pane shows the current values of the 80486 registers. The segment registers (CS, DS, ES, SS) all contain the value 48DD. The instruction pointer (IP) is 0100. The stack pointer (SP) is FFFE. The program counter (PC) is 0100. The status flags (C, Z, S, O, P, A, I, D) are all 0.

| Register | Value |
|----------|-------|
| ax | 0000 |
| bx | 0000 |
| cx | 0000 |
| dx | 0000 |
| si | 0000 |
| di | 0000 |
| bp | 0000 |
| sp | FFFE |
| ds | 48DD |
| es | 48DD |
| ss | 48DD |
| cs | 48DD |
| ip | 0100 |

Assembly code (CS:0100):

```
cs:0100 jmp 01EF ↓
cs:0103 54 push sp
cs:0104 7970 jns 0176
cs:0106 653A20 cmp ah,gs:[bx+si]
cs:0109 2450 and al,50
cs:010B 43 inc bx
cs:010C 0D0A24 or ax,240A
cs:010F 50 push ax
cs:0110 43 inc bx
cs:0111 2F das
cs:0112 58 pop ax
cs:0113 54 push sp
cs:0114 0D0A24 or ax,240A
```


4. Как определяется стек? Какую область памяти он занимает? Какие адреса?

Стек в COM программе определяется автоматически при запуске программы. Он находится в том же сегменте, что и остальная часть программы, указатель стека установлен на конец сегмента (FFFFh). Таким образом под стек отводится оставшаяся часть сегмента после кода и данных.

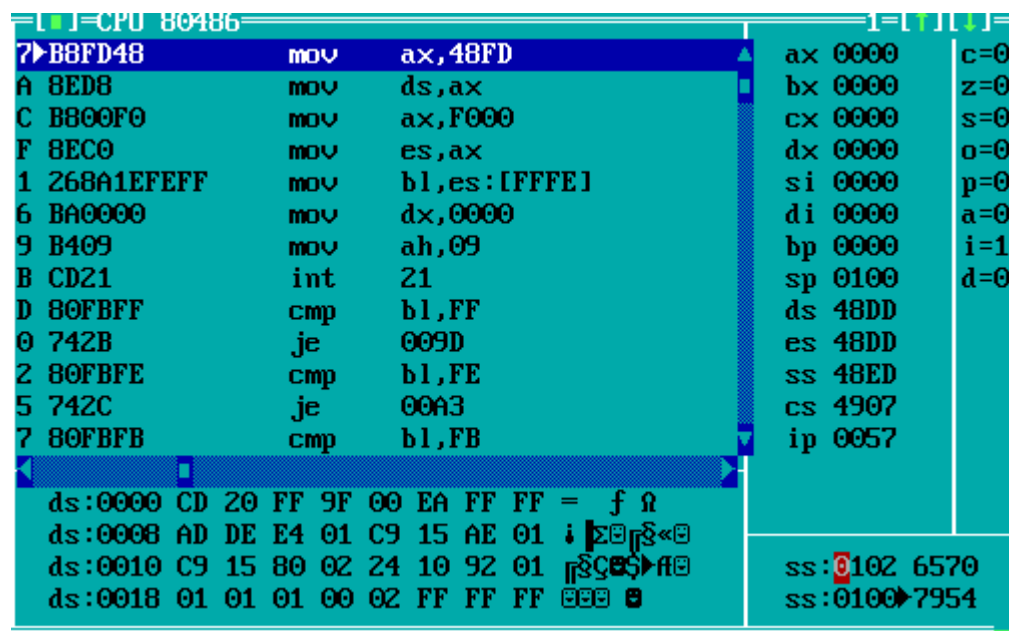
Загрузка «хорошего» EXE модуля в основную память

1. Как загружается «хороший» EXE? Какие значения имеют сегментные регистры?

Определяется сегментный адрес свободного участка памяти, размер которого достаточен для размещения программы. В начале этого участка строится PSP, затем определяется сегментный адрес для загрузки программы(PSP+0010h). Далее обрабатывается таблица настройки адресов: к каждому адресу в таблице добавляется адрес начального сегмента(PSP+0010h).

Таблица содержит в себе смещения на данные или команды, использующие адреса сегментов. Так как сегментные адреса при каждом запуске программы могут быть разными, необходимо корректировать их в программе, для этого и нужна таблица настройки.

Из заголовка файла берется информация о точке входа и адресе стека относительно начального сегмента. К этим адресам также прибавляется адрес начального сегмента. Управление передается по адресу точки входа.



Так как в программе сегмент стека объявлен первым, его смещение относительно начального будет равно 0. DS=ES=48DD указывают на сегмент PSP. Если прибавить к адресу PSP смещение 10h получится адрес начального сегмента - 48ED. Как видно на рисунке именно такой адрес имеет сегмент стека. В CS записывается значение PSP+10h+адрес из заголовка=48DD+10+1A=4907.

2. На что указывают регистры DS и ES?

После загрузки программы регистры DS и ES указывают на сегмент PSP.

3. Как определяется стек?

Стек задается парой регистров SS:SP. При запуске программы в SS помещается смещение сегмента стека относительно начального сегмента программы (содержится в заголовке) + адрес начального сегмента. В SP помещается значение напрямую из заголовка.

4. Как определяется точка входа?

Точка входа задается директивой END и помещается в заголовок файла в виде сегментного адреса сегмента кода относительно начального сегмента и значения IP.