

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

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

Студентка гр. 9382

Пя С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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

Сведения о функциях и структурах.

TETR_TO_HEX: процедура перевода из 10-ой сс в символы

BYTE_TO_HEX: процедура перевода байта из 16-ой сс в символы

WRD_TO_HEX: перевод слова из 16-ой сс в символы

BYTE_TO_DEC: перевод байта из 16-ой сс в 10-ую и символы

print_string: процедура вывода строки на экран

print_type_of_PC: процедура вывода типа IBM PC на экран

print_version_of_PC: процедура вывода версии MS DOS на экран

Последовательность действий, выполняемых утилитой.

1. Определяется тип PC путем считывания содержимого предпоследнего байта ROM BIOS с представленной таблицей в методических указаниях, и выводится строка, содержащая название соответствующего типа.
2. Определяется версия PC. Выводится строка в определенном формате, содержащая номер основной версии и номер модификации в десятичной системе счисления, затем выводится серийный номер OEM и серийный номер пользователя. Путь их определения написан в методических указаниях. Вывод «правильного» файла EXE:

```
C:\>LB1_EXE.EXE
Type of PC: AT
Version of MS DOS: 5.0
Serial number of OEM: 0
Serial number of user: 000000
```

Вывод «плохого» EXE файла:

```
C:\>LB1_COM.EXE

Type of PC:

Type of PC: 5 0

Type of PC: 0

Type of PC:

Type of PC: 000000

Type of PC:

Type of PC:
```

Вывод COM файла:

```
C:\>LB1_COM.COM
Type of PC: AT
Version of MS DOS: 5.0
Serial number of OEM: 0
Serial number of user: 000000
C:\>_
```

Выводы.

В ходе выполнения лабораторной работы была написана программа для определения типа и версии PC, изучены различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Ответы на контрольные вопросы по лабораторной работе №1.

Отличия исходных текстов COM и EXE программ

1) Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать только один сегмент, стек генерируется автоматически, а привычные сегменты кода и данных содержатся в одном сегменте.

2) EXE-программа?

EXE-программа должна содержать более одного сегмента. Код, данные и стек будут храниться в отдельных сегментах.

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

В тексте COM-программы обязательно должны быть директивы ORG 100h и ASSUME, так как при загрузке модуля в начале определяется 100h префикс программного сегмента, так что адресация должна быть смещена на столько же и вторая для того, чтобы указать один сегмент как сегмент данных и кода.

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

Не все форматы команд можно использовать в COM-программе, а именно команды типа mov (регистр), seg (наименование сегмента). COM-программы имеют один сегмент, поэтому сегментные регистры имеют одни и те же значения. А EXE-программы могут быть больше 64КБ, поэтому загрузчик ОС распределяет содержимое по нескольким сегментам.

Отличия форматов файлов COM и EXE модулей

- 1) Какова структура файла COM? С какого адреса располагается код?

Файл COM состоит из одного сегмента и может быть размером не больше 64КБ. Код располагается с 0h, но при загрузке модуля смещается на 100h.

```
E:\OC 2021\LB1_COM.COM
00000000: E9 B7 01 54 79 70 65 20 6F 66 20 50 43 3A 20 24 é·Type of PC: $
00000001: 50 43 0D 0A 24 50 43 2F 58 54 0D 0A 24 41 54 0D PC $PC/XT $AT
00000002: 0A 24 50 53 32 20 6D 6F 64 65 6C 20 33 30 0D 0A $PS2 model 30 $
00000003: 24 50 53 32 20 6D 6F 64 65 6C 20 38 30 0D 0A 24 $PS2 model 80 $
00000004: 50 43 6A 72 0D 0A 24 50 43 20 43 6F 6E 76 65 72 PCjr $PC Conver
00000005: 74 69 62 6C 65 0D 0A 24 56 65 72 73 69 6F 6E 20 tible $Version
00000006: 6F 66 20 4D 53 20 44 4F 53 3A 20 20 2E 20 20 0D of MS DOS: .
00000007: 0A 24 53 65 72 69 61 6C 20 6E 75 6D 62 65 72 20 $Serial number
00000008: 6F 66 20 4F 45 4D 3A 20 20 20 24 53 65 72 69 61 of OEM: $Seria
00000009: 6C 20 6E 75 6D 62 65 72 20 6F 66 20 75 73 65 72 l number of user
0000000A: 3A 20 20 20 20 20 20 20 20 20 24 0D 0A 24 24 0F : $ $$$
0000000B: 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF FF 86 <v$Q$àëiÿ+
0000000C: C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8 E9 FF A±0ëæÿÿÄ$Süëÿ
0000000D: 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F 88 05 "%0"0$Çapÿ"%0"
0000000E: 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80 CA 30 [ÄQR2a30" ÷ñë0
0000000F: 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04 0C 30 "N30= sñ< t00
00000010: 88 04 5A 59 C3 B4 09 CD 21 C3 BA 03 01 E8 F5 FF "ZYÄ"ñíÄ"0ë0ÿ
00000011: B8 00 F0 8E C0 26 A0 FE FF 3C FF 74 1C 3C FE 74 , ðŽÀ& þÿ<ÿt0<pt
00000012: 1E 3C FB 74 1A 3C FC 74 1C 3C FA 74 1E 3C F8 74 0<ÿt0<ÿt0<ÿt0<0t
00000013: 20 3C FD 74 22 3C F9 74 24 BA 10 01 EB 22 90 BA <ÿt"<ÿt$000ë"00
00000014: 15 01 EB 1C 90 BA 1D 01 EB 16 90 BA 22 01 EB 10 000000000000"0ë0
00000015: 90 BA 31 01 EB 0A 90 BA 40 01 EB 04 90 BA 47 01 0010ë00000000000G0
00000016: E8 A2 FF C3 B4 30 CD 21 50 BE 58 01 83 C6 13 E8 èçÿÄ"0í!PXX0fæ0è
00000017: 70 FF 58 8A C4 83 C6 03 E8 67 FF BA 58 01 E8 84 pÿX$Äfæ0ègÿ0X0è,,
00000018: FF BE 72 01 83 C6 16 8A C7 E8 56 FF BA 72 01 E8 ÿXr0fæ0$Çèÿÿ0r0è
00000019: 73 FF BA AB 01 E8 6D FF BF 8B 01 83 C7 1C 8B C1 sÿ0«0ëÿÿ: <0fÇ0«Ä
0000001A: E8 27 FF 8A C3 E8 11 FF 83 EF 02 89 05 BA 8B 01 è'ÿ$Äèÿÿfi0000«0
0000001B: E8 52 FF BA AB 01 E8 4C FF C3 E8 4D FF E8 A4 FF èÿÿ0«0èÿÿÄèÿÿèÿÿ
0000001C: 32 C0 B4 4C CD 21 2Ä'Lí!
```

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

У файла «плохого» EXE код и данные будут располагаться в одном сегменте, код будет располагаться с адреса 300h, с адреса 0h будет таблица настроек.

```

00000002F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000300: E9 B7 01 54 79 70 65 20 6F 66 20 50 43 3A 20 24 6F 66 20 50 43 3A 20 24 6F 66 20 50 43 3A 20 24 6F 66 20 50 43 3A 20 24
0000000310: 50 43 0D 0A 24 50 43 2F 64 65 6C 20 33 30 0D 0A 64 65 6C 20 33 30 0D 0A 64 65 6C 20 33 30 0D 0A 64 65 6C 20 33 30 0D 0A
0000000320: 0A 24 50 53 32 20 6D 6F 65 6C 20 38 30 0D 0A 64 65 6C 20 38 30 0D 0A 64 65 6C 20 38 30 0D 0A 64 65 6C 20 38 30 0D 0A
0000000330: 24 50 53 32 20 6D 6F 64 43 20 43 6F 6E 76 65 72 43 20 43 6F 6E 76 65 72 43 20 43 6F 6E 76 65 72 43 20 43 6F 6E 76 65 72
0000000340: 50 43 6A 72 0D 0A 24 50 56 65 72 73 69 6F 6E 20 53 3A 20 20 2E 20 20 0D 20 6E 75 6D 62 65 72 20 20 24 53 65 72 69 61
0000000350: 74 69 62 6C 65 0D 0A 24 53 3A 20 20 2E 20 20 0D 20 6E 75 6D 62 65 72 20 20 24 53 65 72 69 61
0000000360: 6F 66 20 4D 53 20 44 4F 20 6E 75 6D 62 65 72 20 20 24 53 65 72 69 61
0000000370: 0A 24 53 65 72 69 61 6C 20 6F 66 20 4F 45 4D 3A 20 20 6F 66 20 75 73 65 72 20 20 24 0D 0A 24 24 0F
0000000380: 6F 66 20 4F 45 4D 3A 20 20 6F 66 20 75 73 65 72 20 20 24 0D 0A 24 24 0F
0000000390: 6C 20 6E 75 6D 62 65 72 20 20 24 0D 0A 24 24 0F
00000003A0: 3A 20 20 20 20 20 20 20 20 20 24 0D 0A 24 24 0F
00000003B0: 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF FF 86
00000003C0: C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8 E9 FF
00000003D0: 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F 88 05
00000003E0: 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80 CA 30
00000003F0: 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04 0C 30
0000000400: 88 04 5A 59 C3 B4 09 CD 21 C3 BA 03 01 E8 F5 FF
0000000410: B8 00 F0 8E C0 26 A0 FE FF 3C FF 74 1C 3C FE 74
0000000420: 1E 3C FB 74 1A 3C FC 74 1C 3C FA 74 1E 3C F8 74
0000000430: 20 3C FD 74 22 3C F9 74 24 BA 10 01 EB 22 90 BA
0000000440: 15 01 EB 1C 90 BA 1D 01 EB 16 90 BA 22 01 EB 10
0000000450: 90 BA 31 01 EB 0A 90 BA 40 01 EB 04 90 BA 47 01
0000000460: E8 A2 FF C3 B4 30 CD 21 50 BE 58 01 83 C6 13 E8
0000000470: 70 FF 58 8A C4 83 C6 03 E8 67 FF BA 58 01 E8 84
0000000480: FF BE 72 01 83 C6 16 8A C7 E8 56 FF BA 72 01 E8
0000000490: 73 FF BA AB 01 E8 6D FF BF 8B 01 83 C7 1C 8B C1
00000004A0: E8 27 FF 8A C3 E8 11 FF 83 EF 02 89 05 BA 8B 01
00000004B0: E8 52 FF BA AB 01 E8 4C FF C3 E8 4D FF E8 A4 FF
00000004C0: 32 C0 B4 4C CD 21

```

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

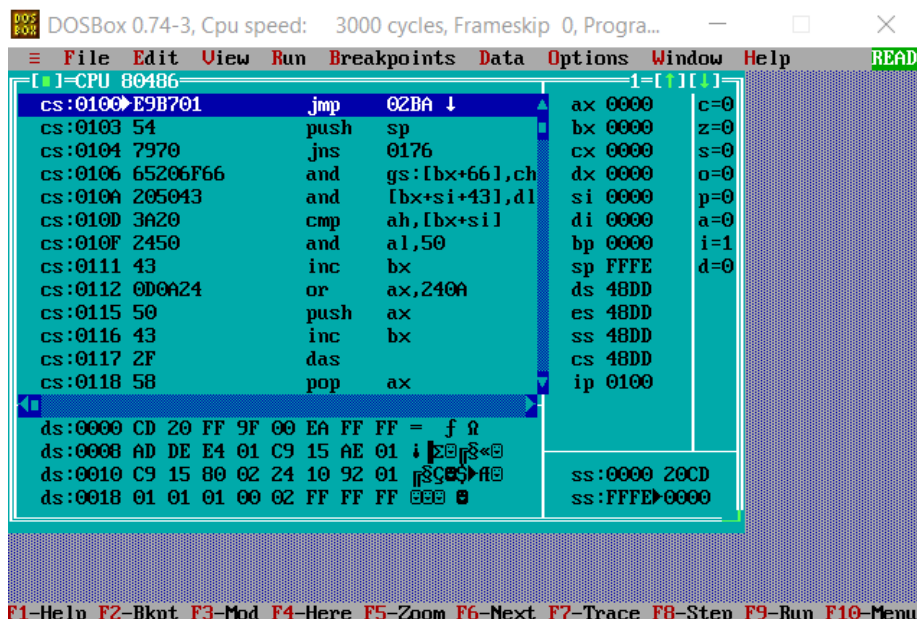
У файла «хорошего» EXE данные, код и стек размещены в разных сегментах, и файл может быть большего размера, чем 64КБ. У «плохого» файла EXE код начинается с 300h, так как он получается из файла COM, в котором код смещен на 100h, а 200h – это размер PSP модуля, а у «хорошего» EXE выделяется память под стек между PSP и кодом.

0000000300:	54 79 70 65 20 6F 66 20	50 43 3A 20 24 50 43 0D	Type of PC: \$PC
0000000310:	0A 24 50 43 2F 58 54 0D	0A 24 41 54 0D 0A 24 50	\$PC/XT \$AT \$P
0000000320:	53 32 20 6D 6F 64 65 6C	20 33 30 0D 0A 24 50 53	S2 model 30 \$PS
0000000330:	32 20 6D 6F 64 65 6C 20	38 30 0D 0A 24 50 43 6A	2 model 80 \$PCj
0000000340:	72 0D 0A 24 50 43 20 43	6F 6E 76 65 72 74 69 62	r \$PC Convertib
0000000350:	6C 65 0D 0A 24 56 65 72	73 69 6F 6E 20 6F 66 20	le \$Version of
0000000360:	4D 53 20 44 4F 53 3A 20	20 2E 20 20 0D 0A 24 53	MS DOS: . \$S
0000000370:	65 72 69 61 6C 20 6E 75	6D 62 65 72 20 6F 66 20	erial number of
0000000380:	4F 45 4D 3A 20 20 20 24	53 65 72 69 61 6C 20 6E	OEM: \$Serial n
0000000390:	75 6D 62 65 72 20 6F 66	20 75 73 65 72 3A 20 20	umber of user:
00000003A0:	20 20 20 20 20 20 20 24	0D 0A 24 00 00 00 00 00	\$ \$
00000003B0:	24 0F 3C 09 76 02 04 07	04 30 C3 51 8A E0 E8 EF	\$<v\$QŠàèi
00000003C0:	FF 86 C4 B1 04 D2 E8 E8	E6 FF 59 C3 53 8A FC E8	ÿ+Ä±DèèæÿYÄSŠuè
00000003D0:	E9 FF 88 25 4F 88 05 4F	8A C7 E8 DE FF 88 25 4F	éÿ^%O^OSÇèpÿ^%O
00000003E0:	88 05 5B C3 51 52 32 E4	33 D2 B9 0A 00 F7 F1 80	^[ÄQR2ä30¹ ÷ñ€
00000003F0:	CA 30 88 14 4E 33 D2 3D	0A 00 73 F1 3C 00 74 04	Ê0^N3D= sñ< t
0000000400:	0C 30 88 04 5A 59 C3 B4	09 CD 21 C3 BA 00 00 E8	00ZYÄ'Í!Ä° è
0000000410:	F5 FF B8 00 F0 8E C0 26	A0 FE FF 3C FF 74 1C 3C	öÿ, ðŽÄ& pÿ<ÿt<
0000000420:	FE 74 1E 3C FB 74 1A 3C	FC 74 1C 3C FA 74 1E 3C	pt<û t<û t<û t<
0000000430:	F8 74 20 3C FD 74 22 3C	F9 74 24 BA 0D 00 EB 22	øt <ÿt"<û t\$° ë"
0000000440:	90 BA 12 00 EB 1C 90 BA	1A 00 EB 16 90 BA 1F 00	°° ë°°°°° ë°°°°°
0000000450:	EB 10 90 BA 2E 00 EB 0A	90 BA 3D 00 EB 04 90 BA	ë°°°. ë°°°°= ë°°°°
0000000460:	44 00 E8 A2 FF C3 B4 30	CD 21 50 BE 55 00 83 C6	D èøÿÄ'0Í!P%U fÆ
0000000470:	13 E8 70 FF 58 8A C4 83	C6 03 E8 67 FF BA 55 00	èèpÿXŠÄfÆèègÿ°U
0000000480:	E8 84 FF BE 6F 00 83 C6	16 8A C7 E8 56 FF BA 6F	è„ÿ%o fÆŠÇèVÿ°o
0000000490:	00 E8 73 FF BA A8 00 E8	6D FF BF 88 00 83 C7 1C	èsÿ°" èmÿ¿^ fÇ
00000004A0:	8B C1 E8 27 FF 8A C3 E8	11 FF 83 EF 02 89 05 BA	<Äè'ÿŠÄèÿfi%°°
00000004B0:	88 00 E8 52 FF BA A8 00	E8 4C FF C3 2B C0 50 B8	^ èRÿ°" èLÿÄ+ÄP,
00000004C0:	10 00 8E D8 E8 45 FF E8	9C FF 32 C0 B4 4C CD 21	ŽøèEÿèæÿ2Ä'Í!
00000004D0:	CB	E	

Загрузка СОМ модуля в основную память

- 1) Какой формат загрузки модуля СОМ? С какого адреса располагается код?

Сначала определяется сегментный адрес участка ОП, способного вместить загрузку программы, затем создается блок памяти для PSP и программы, СОМ-файл считывается помещается в память с 100h. После сегментные регистры устанавливаются на начало PSP. SP устанавливается на конец PSP, 0000h помещается в стек, в IP записывается 100h. Код располагается с адреса 100h.



2) Что располагается с адреса 0?

PSP сегмент располагается с адреса 0.

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

Сегментные регистры имеют значения 48DD и указывают на PSP.

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

Стек генерируется автоматически. SP указывает на конец стека, а SS – на начало. Адреса расположены в диапазоне 0h–FFFEh (потому что это последний адрес, который кратен двум).

Загрузка «хорошего» EXE модуля в основную память

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

Загружается «хороший» EXE со считыванием информации заголовка EXE, выполняется перемещение адресов сегментов, ES и DS устанавливаются на начало PSP, SS – на начало сегмента стека, а CS – на начало сегмента команд.

The screenshot shows the DOSBox 0.74-3 interface. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The menu bar includes File, Edit, View, Run, Breakpoints, Data, Options, Window, and Help. The main window is divided into several sections:

- Assembly List:** A list of instructions with their addresses and hex values. The current instruction is at address 010C: 2BC0, which is a "sub ax,ax" instruction.
- Registers:** A table showing the current values of 80486 registers. For example, ax is 0000, bx is 0000, cx is 0000, dx is 0000, si is 0000, di is 0000, bp is 0000, sp is 0100, ds is 48DD, es is 48DD, ss is 48ED, cs is 4908, and ip is 010C.
- Memory Dump:** A section showing memory contents at various addresses, such as ds:0000, ds:0008, ds:0010, and ds:0018.
- Stack:** A section showing the stack pointer (sp) and its current value, which is 0100.

At the bottom of the window, there is a status bar with function key shortcuts: F1-Help, F2-Bkpt, F3-Mod, F4-Here, F5-Zoom, F6-Next, F7-Trace, F8-SteP, F9-Run, and F10-Menu.

2) На что указывают регистры DS и ES?

ES и DS указывают на начало PSP.

3) Как определяется стек?

Стек определяется с помощью директивы .stack с указанием размера стека. SS указывает на начало сегмента стека, а SP – на конец.

4) Как определяется точка входа?

Точка входа определяется директивой END.