

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

Студент гр. 8382

Чирков С.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

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

Выполнение работы.

Для определения типа PC и версии системы были написаны тексты .COM и .EXE модулей (см. Приложение А и В). Тип IBM PC хранится в байте по адресу 0F000:0FFFEh (предпоследний байт ROM BIOS).

Для определения версии MS DOS была использована функция 30H прерывания 21H. После её вызова версия системы определяется значением регистров AL, AH. В регистре BH – серийный номер OEM, в BL:CH – 24-битовый серийный номер пользователя.

Результат выполнения .COM модуля представлен на рис. 1. Результат выполнения «плохого» .EXE модуля, полученного из исходного текста для .COM модуля, представлен на рис. 2. Результат выполнения «хорошего» .EXE модуля представлен на рис. 3.

```
C:\>lr1com.com
Your IBM PC type is PS2 ver. 50/60 or AT
Your MSDOS type is 05.00
Your serial OEM number is 255
Your serial user number is 000000
```

Рисунок 1. Результат работы LR1COM.COM

```
C:\>lr1com.exe

  Your IBM PC type is

  Your IBM PC type is

  Your IBM PC type is 5 0

  Your IBM PC type is 255

  Your IBM PC type is 000000
```

Рисунок 2. Результат работы LR1COM.EXE

```

C:\>lr1exe.exe
Your IBM PC type is PS2 ver. 50/60 or AT
Your MSDOS type is 05.00
Your serial OEM number is 255
Your serial user number is 000000

```

Рисунок 3. Результат работы LR1EXE.EXE

Вид исходных файлов в шестнадцатеричном виде представлен на рисунках 4-8.

0000000000:	E9 1E 01 59 6F 75 72 20	49 42 4D 20 50 43 20 74	é▲Your IBM PC t
0000000010:	79 70 65 20 69 73 20 24	50 43 0A 0D 24 50 43 2F	ype is \$PC\$PC/
0000000020:	58 54 0A 0D 24 50 53 32	20 76 65 72 2E 20 33 30	XT\$PS2 ver. 30
0000000030:	0A 0D 24 50 53 32 20 76	65 72 2E 20 35 30 2F 36	\$PS2 ver. 50/6
0000000040:	30 20 6F 72 20 41 54 0A	0D 24 50 53 32 20 76 65	0 or AT\$PS2 ve
0000000050:	72 2E 20 38 30 0A 0D 24	50 43 6A 72 0A 0D 24 50	r. 80\$PCjr\$P
0000000060:	43 20 43 6F 6E 76 65 72	74 69 62 6C 65 0A 0D 24	C Convertible\$
0000000070:	59 6F 75 72 20 4D 53 44	4F 53 20 74 79 70 65 20	Your MSDOS type
0000000080:	69 73 20 24 3C 32 2E 30	0A 0D 24 30 78 2E 30 79	is \$<2.0\$0x.0y
0000000090:	0A 0D 24 59 6F 75 72 20	73 65 72 69 61 6C 20 4F	\$Your serial 0
00000000A0:	45 4D 20 6E 75 6D 62 65	72 20 69 73 20 24 0A 0D	EM number is \$
00000000B0:	59 6F 75 72 20 73 65 72	69 61 6C 20 75 73 65 72	Your serial user
00000000C0:	20 6E 75 6D 62 65 72 20	69 73 20 24 51 52 E8 1C	number is \$QRèL
00000000D0:	00 8A EC 8A D0 B4 02 CD	21 8A D5 B4 02 CD 21 5A	ŠiŠD`oÍ!ŠD`oÍ!Z
00000000E0:	59 C3 24 0F 3C 09 76 02	04 07 04 30 C3 51 8A E0	YĂ\$<ov0♦♦0ĂQŠà
00000000F0:	E8 EF FF 86 C4 B1 04 D2	E8 E8 E6 FF 59 C3 51 52	èiÿ+Ă±♦0èèæÿYĂQR
0000000100:	32 E4 33 D2 B9 0A 00 F7	F1 80 CA 30 88 14 4E 33	2ă30¹ ÷ñ€Ê0^ĴN3
0000000110:	D2 3D 0A 00 73 F1 3C 00	74 04 0C 30 88 04 5A 59	Ô= sñ< t♦00^♦ZY
0000000120:	C3 B8 00 F0 8E C0 26 A0	FE FF BA 03 01 B4 09 CD	Ă. ôŽĂ& pÿ♥o`oÍ
0000000130:	21 BA 18 01 3C FF 74 31	BA 1D 01 3C FE 74 2A 3C	!e†0<ÿt1e+0<pt*<
0000000140:	FB 74 26 BA 33 01 3C FC	74 1F BA 25 01 3C FA 74	ût&e30<ût▼e%0<ût
0000000150:	18 BA 4A 01 3C F8 74 11	BA 58 01 3C FD 74 0A BA	†eJ0<øt◀eX0<ÿt
0000000160:	5F 01 3C F9 75 00 E8 63	FF B4 09 CD 21 BA 70 01	_0<ûu ècÿ`oÍ!epe
0000000170:	B4 09 CD 21 B4 30 CD 21	3C 00 75 07 BA 84 01 B4	`oÍ!`oÍ!< u•e„o`
0000000180:	09 CD 21 BE 8B 01 83 C6	01 E8 72 FF 83 C6 04 8A	oÍ!%<0fæ0èÿfæ♦Š
0000000190:	C4 E8 6A FF BA 8B 01 B4	09 CD 21 B4 30 CD 21 BA	Ăèjÿe<0`oÍ!`oÍ!e
00000001A0:	93 01 B4 09 CD 21 8A C7	E8 53 FF 8A 14 B4 02 CD	“0`oÍ!ŠçèsÿŠĴ`oÍ
00000001B0:	21 8A 54 01 CD 21 8A 54	02 CD 21 B4 30 CD 21 BA	!ŠT0Í!ŠT0Í!`oÍ!e
00000001C0:	AE 01 B4 09 CD 21 8A C3	E8 01 FF 8A C5 E8 FC FE	00`oÍ!ŠĂè0ÿŠĂèüþ
00000001D0:	8A C1 E8 F7 FE 32 C0 B4	4C CD 21	ŠĂè÷þ2Ă`LÍ!

Рисунок 4. LR1COM.COM в шестнадцатеричном виде

```

00000000: 4D 5A DB 00 03 00 00 00 20 00 00 00 FF FF 00 00 MZU  y
00000001: 00 00 00 00 00 01 00 00 3E 00 00 00 01 00 FB 30  >  u
00000002: 6A 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 jr
00000003: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000004: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000005: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000006: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000007: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000008: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000009: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000011: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000012: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000013: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000014: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000015: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000017: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000018: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000019: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000021: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000022: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000023: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000024: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000025: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Рисунок 5. LR1COM.EXE в шестнадцатеричном виде (часть 1)

```

00000026: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000027: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000028: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000029: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030: E9 1E 01 59 6F 75 72 20 49 42 4D 20 50 43 20 74  é@Your IBM PC t
00000031: 79 70 65 20 69 73 20 24 50 43 0A 0D 24 50 43 2F  ype is $PC$PC/
00000032: 58 54 0A 0D 24 50 53 32 20 76 65 72 2E 20 33 30  XT$PS2 ver. 30
00000033: 0A 0D 24 50 53 32 20 76 65 72 2E 20 35 30 2F 36  $PS2 ver. 50/6
00000034: 30 20 6F 72 20 41 54 0A 0D 24 50 53 32 20 76 65  0 or AT$PS2 ve
00000035: 72 2E 20 38 30 0A 0D 24 50 43 6A 72 0A 0D 24 50  r. 80$PCjr$P
00000036: 43 20 43 6F 6E 76 65 72 74 69 62 6C 65 0A 0D 24  C Convertible$
00000037: 59 6F 75 72 20 40 53 44 4F 53 20 74 79 70 65 20  Your MSDOS type
00000038: 69 73 20 24 3C 32 2E 30 0A 0D 24 30 78 2E 30 79  is $<2.0$X.0y
00000039: 0A 0D 24 59 6F 75 72 20 73 65 72 73 65 72 69 61 6C 20 4F  $Your serial 0
0000003A: 45 4D 20 6E 75 6D 62 65 72 20 69 73 20 24 0A 0D  EM number is $
0000003B: 59 6F 75 72 20 73 65 72 69 61 6C 20 75 73 65 72  Your serial user
0000003C: 20 6E 75 6D 62 65 72 20 69 73 20 24 51 52 E8 1C  number is $QRèL
0000003D: 00 8A EC 8A D0 B4 02 CD 21 8A D5 B4 02 CD 21 5A  $iS0'0i!S0'0i!Z
0000003E: 59 C3 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0  YÂ$ov0+0AQ5â
0000003F: E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 51 52  èiytA+0èèyYÂQR
00000040: 32 E4 33 D2 B9 0A 00 F7 F1 80 CA 30 88 14 4E 33  2â30*â+ñ€E0*ñN3
00000041: D2 3D 0A 00 73 F1 3C 00 74 04 0C 30 88 04 5A 59  0= sñ< t*00*ZY
00000042: C3 B8 00 F0 8E C0 26 A0 FE FF BA 03 01 B4 09 CD  A. ôZÂ& by€'oi
00000043: 21 BA 18 01 3C FF 74 31 BA 1D 01 3C FE 74 2A 3C  !€t<yt1€+€<pt*<
00000044: FB 74 26 BA 33 01 3C FC 74 1F BA 25 01 3C FA 74  út&€30<út€%€<út
00000045: 18 BA 4A 01 3C F8 74 11 BA 58 01 3C FD 74 0A BA  †€J0<€t*€X0<yt€€
00000046: 5F 01 3C F9 75 00 E8 63 FF B4 09 CD 21 BA 70 01  _€<uu €cy'oi!€p€
00000047: B4 09 CD 21 B4 30 CD 21 3C 00 75 07 BA 84 01 B4  'oi!'0i!< u*€,0'
00000048: 09 CD 21 BE 8B 01 83 C6 01 E8 72 FF 83 C6 04 8A  oi!X<€f€0èryfA*Š
00000049: C4 E8 6A FF BA 8B 01 B4 09 CD 21 B4 30 CD 21 BA  Aèjy€<€'oi! '0i!€
0000004A: 93 01 B4 09 CD 21 8A C7 E8 53 FF 8A 14 B4 02 CD  "€'oi!ŠcâsyŠŋ'0i
0000004B: 21 8A 54 01 CD 21 8A 54 02 CD 21 B4 30 CD 21 BA  !Št0i!Št0i! '0i!€
0000004C: AE 01 B4 09 CD 21 8A C3 E8 01 FF 8A C5 E8 FC FE  €'oi!ŠAèyŠAèyb
0000004D: 8A C1 E8 F7 FE 32 C0 B4 4C CD 21 ŠAè+â2A'Lí!

```

Рисунок 6. LR1COM.EXE в шестнадцатеричном виде (часть 2)

0000000000:	4D 5A E4 01 03 00 01 00	20 00 00 00 FF FF 00 00	MZä@♥ @	ÿÿ
0000000010:	00 02 00 00 55 00 2D 00	3E 00 00 00 01 00 FB 30	Ⓢ U - >	Ⓢ ůⓈ
0000000020:	6A 72 00 00 00 00 00 00	00 00 00 00 00 00 00 00	jṛ	
0000000030:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 56 00		V
0000000040:	2D 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	-	
0000000050:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000060:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000070:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000080:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000090:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000100:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000110:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000120:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000130:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000140:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000150:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000160:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000170:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000180:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000190:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000200:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000210:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000220:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000230:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000240:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000250:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000260:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000270:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000280:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000290:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000300:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		

Рисунок 7. LR1EXE.EXE в шестнадцатеричном виде (часть 1)

0000000310:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000320:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000330:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000340:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000350:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000360:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000370:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000380:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000390:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000003F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000400:	59 6F 75 72 20 49 42 4D	20 50 43 20 74 79 70 65	Your IBM PC type
0000000410:	20 69 73 20 24 50 43 0A	0D 24 50 43 2F 58 54 0A	is \$PC\$PC/XT
0000000420:	0D 24 50 53 32 20 76 65	72 2E 20 33 30 0A 0D 24	\$PS2 ver. 30\$
0000000430:	50 53 32 20 76 65 72 2E	20 35 30 2F 36 30 20 6F	PS2 ver. 50/60 o
0000000440:	72 20 41 54 0A 0D 24 50	53 32 20 76 65 72 2E 20	r AT\$PS2 ver.
0000000450:	38 30 0A 0D 24 50 43 6A	72 0A 0D 24 50 43 20 43	80\$PCjr\$PC C
0000000460:	6F 6E 76 65 72 74 69 62	6C 65 0A 0D 24 59 6F 75	onvertible\$You
0000000470:	72 20 4D 53 44 4F 53 20	74 79 70 65 20 69 73 20	r MSDOS type is
0000000480:	24 3C 32 2E 30 0A 0D 24	30 78 2E 30 79 0A 0D 24	\$<2.0\$0x.0y\$
0000000490:	59 6F 75 72 20 73 65 72	69 61 6C 20 4F 45 4D 20	Your serial OEM
00000004A0:	6E 75 6D 62 65 72 20 69	73 20 24 0A 0D 59 6F 75	number is \$You
00000004B0:	72 20 73 65 72 69 61 6C	20 75 73 65 72 20 6E 75	r serial user nu
00000004C0:	6D 62 65 72 20 69 73 20	24 00 00 00 00 00 00 00	mber is \$
00000004D0:	51 52 E8 1C 00 8A EC 8A	D0 B4 02 CD 21 8A D5 B4	QRèL ŠiŠD'ŌÍ!ŠD'
00000004E0:	02 CD 21 5A 59 C3 24 0F	3C 09 76 02 04 07 04 30	ŌÍ!ZYĂ\$◊◊ov◊◊
00000004F0:	C3 51 8A E0 E8 EF FF 86	C4 B1 04 D2 E8 E8 E6 FF	ĂQŠàèiŷ†Ă±◊Đèèæŷ
0000000500:	59 C3 51 52 32 E4 33 D2	B9 0A 00 F7 F1 80 CA 30	YĂQR2ă3Đ¹÷ñ€Ê0
0000000510:	88 14 4E 33 D2 3D 0A 00	73 F1 3C 00 74 04 0C 30	^ŷN3Đ= sñ< t◊90
0000000520:	88 04 5A 59 C3 B8 20 00	8E D8 B8 00 F0 8E C0 26	^◊ZYĂ. Ž0. đŽĂ&
0000000530:	A0 FE FF BA 00 00 B4 09	CD 21 BA 15 00 3C FF 74	pŷ° 'oÍ!°š <ŷt
0000000540:	31 BA 1A 00 3C FE 74 2A	3C FB 74 26 BA 30 00 3C	1°→ <pt*<ût&°0 <
0000000550:	FC 74 1F BA 22 00 3C FA	74 18 BA 47 00 3C F8 74	ût▼°" <út†°G <øt
0000000560:	11 BA 55 00 3C FD 74 0A	BA 5C 00 3C F9 75 00 E8	◄°U <ŷt°\ <ùu è
0000000570:	5E FF B4 09 CD 21 BA 6D	00 B4 09 CD 21 B4 30 CD	^ŷ'oÍ!°m 'oÍ!'ŌÍ
0000000580:	21 3C 00 75 07 BA 81 00	B4 09 CD 21 BE 88 00 83	!< u•° 'oÍ!½^ f
0000000590:	C6 01 E8 6D FF 83 C6 04	8A C4 E8 65 FF BA 88 00	Æ0èmŷfÆ◊ŠĂèèŷ°^
00000005A0:	B4 09 CD 21 B4 30 CD 21	BA 90 00 B4 09 CD 21 8A	'oÍ!'ŌÍ!° 'oÍ!Š
00000005B0:	C7 E8 4E FF 8A 14 B4 02	CD 21 8A 54 01 CD 21 8A	ÇèNŷŠŷ'ŌÍ!ŠTŌÍ!Š
00000005C0:	54 02 CD 21 B4 30 CD 21	BA AB 00 B4 09 CD 21 8A	TŌÍ!'ŌÍ!°« 'oÍ!Š
00000005D0:	C3 E8 FC FE 8A C5 E8 F7	FE 8A C1 E8 F2 FE 32 C0	ĂèüþŠĂè÷þŠĂèðþ2Ă
00000005E0:	B4 4C CD 21		'LÍ!

Рисунок 8. LR1EXE.EXE в шестнадцатеричном виде (часть 2)

Вид файлов модулей .COM и .EXE в отладчике представлен на рисунках 9 и 10 соответственно.

CPU 80486			
cs:0100	E91E01	jmp	0221 ↓
cs:0103	59	pop	cx
cs:0104	6F	outsw	
cs:0105	7572	jne	0179
cs:0107	204942	and	[bx+di+42],cl
cs:010A	4D	dec	bp
cs:010B	205043	and	[bx+si+43],dl
cs:010E	207479	and	[si+79],dh
cs:0111	7065	jo	0178
cs:0113	206973	and	[bx+di+73],ch
cs:0116	2024	and	[si],ah
cs:0118	50	push	ax
cs:0119	43	inc	bx

ax	0000	c=0
bx	0000	z=0
cx	0000	s=0
dx	0000	o=0
si	0000	p=0
di	0000	a=0
bp	0000	i=1
sp	FFFE	d=0
ds	449D	
es	449D	
ss	449D	
cs	449D	
ip	0100	

ds:0000	CD 20 FF 9F 00 EA FF FF	= f Ω
ds:0008	AD DE E5 01 00 15 AF 01	↓ ⌂ ⌂ ⌂
ds:0010	00 15 7D 02 1C 0F 92 01	⌂ ⌂ ⌂ ⌂ ⌂
ds:0018	01 01 01 00 02 FF FF FF	⌂ ⌂ ⌂ ⌂

ss:0000	20CD
ss:FFFE	0000

Рисунок 9. LR1COM.COM в отладчике TD.EXE

CPU 80486			
cs:0055	B8CD44	mov	ax,44CD
cs:0058	8ED8	mov	ds,ax
cs:005A	B800F0	mov	ax,F000
cs:005D	8EC0	mov	es,ax
cs:005F	26A0FEFF	mov	al,es:[FFFE]
cs:0063	BA0000	mov	dx,0000
cs:0066	B409	mov	ah,09
cs:0068	CD21	int	21
cs:006A	BA1500	mov	dx,0015
cs:006D	3CFF	cmp	al,FF
cs:006F	7431	je	00A2
cs:0071	BA1A00	mov	dx,001A
cs:0074	3CFE	cmp	al,FE

ax	0000	c=0
bx	0000	z=0
cx	0000	s=0
dx	0000	o=0
si	0000	p=0
di	0000	a=0
bp	0000	i=1
sp	0200	d=0
ds	449D	
es	449D	
ss	44AD	
cs	44DA	
ip	0055	

ds:0000	CD 20 FF 9F 00 EA FF FF	= f Ω
ds:0008	AD DE E5 01 00 15 AF 01	↓ ⌂ ⌂ ⌂
ds:0010	00 15 7D 02 1C 0F 92 01	⌂ ⌂ ⌂ ⌂ ⌂
ds:0018	01 01 01 00 02 FF FF FF	⌂ ⌂ ⌂ ⌂

ss:0202	7275
ss:0200	6F59

Рисунок 10. LR1EXE.EXE в отладчике TD.EXE

Контрольные вопросы.

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM-программа содержит один сегмент.

2. EXE-программа?

EXE-программа может содержать различное количество сегментов в зависимости от модели памяти (например compact – один сегмент кода, несколько сегментов данных, сегмент стека)

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

ASSUME(устанавливает соответствие сегментных регистров и сегмента) и ORG 100h(устанавливает смещение в 100h для PSP в начале программы)

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

Нельзя использовать команды, использующие адрес сегмента, так как он определяется после запуска программы. Также нельзя создать больше одного сегмента.

Отличия форматов файлов COM и EXE модулей

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

COM содержит один сегмент, а код располагается с адреса 0h.

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

«Плохой» EXE также содержит один сегмент, код располагается с адреса 300h, с адреса 0 располагается заголовок(сигнатура файла, число элементов и адрес таблицы настройки адресов, длина заголовка, объём памяти, требующийся для выделения и прочие данные) и таблица настройки адресов(состоит из длинных указателей вида смещение:сегмент на слова в загрузочном модуле, которые содержат настраиваемые сегментные адреса).

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

Структура файла аналогична, за исключением наличия трёх сегментов вместо одного. Теперь сегмент кода расположен по адресу 400h, так как перед ним появился сегмент стека размером 100h.

Загрузка COM модуля в основную память

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

Код располагается с адреса 100h, перед ним – PSP.

2. Что располагается с адреса 0?

PSP — структура данных, которая используется в операционных системах семейства DOS и CP/M для сохранения состояния компьютерных программ.

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

Сегментные регистры имеют значение 449D, указывающее на начало PSP.

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

Указатель стека (регистр SP) устанавливается на конец основного сегмента. В стек записывается 0000h (адрес возврата для команды ret).

Загрузка «хорошего» EXE модуля в основную память

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

DS, ES (449D) – начало PSP;

SS (44AD) – начало сегмента стека;

CS (44DA) – начало сегмента кода.

2. На что указывают регистры DS и ES?

Эти регистры указывают на начало PSP.

3. Как определяется стек?

Директивой `.STACK` или с помощью `ASSUME SS:<ссылка на сегмент, который будет определен как сегмент стека>`

4. Как определяется точка входа?

Директивой `END <ссылка на точку входа>`.

Выводы.

В ходе работы было исследовано различие в структурах исходных текстов модулей типов `.COM` и `.EXE`, структур файлов загрузочных модулей и способов из загрузки в основную память с помощью двух вариантов программы, которая определяет тип и версию системы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ LR1COM.ASM

```
TESTPC SEGMENT
    ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
    ORG 100H
START: JMP BEGIN

TypePc db 'Your IBM PC type is ','$'
FF db 'PC',10,13,'$'
FE_FB db 'PC/XT',10,13,'$'
FA db 'PS2 ver. 30',10,13,'$'
FC db 'PS2 ver. 50/60 or AT',10,13,'$'
F8 db 'PS2 ver. 80',10,13,'$'
FD db 'PCjr',10,13,'$'
F9 db 'PC Convertible',10,13,'$'
TypeMSDOS db 'Your MSDOS type is ','$'
old db '<2.0',10,13,'$'
new db '0x.0y',10,13,'$'
serial db 'Your serial OEM number is ','$'
serial2 db 10,13,'Your serial user number is ','$'

PRINT PROC near
    push CX
    push DX

    call BYTE_TO_HEX
    mov CH, AH

    mov DL, AL
    mov AH, 02h
    int 21h

    mov DL, CH
    mov AH, 02h
    int 21h

    pop DX
    pop CX
    ret
PRINT ENDP

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
        pop CX
        ret
BYTE_TO_HEX ENDP

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_1
        or AL,30h
        mov [SI],AL
end_1: pop DX
        pop CX
        ret
BYTE_TO_DEC ENDP

BEGIN:
        mov ax,0F000h
        mov es,ax
        mov al,es:[0FFFEh]

        mov DX,offset TypePc
        mov AH,09h
        int 21h

        mov DX,offset FF
        cmp al, 0FFh
        je rdy
        mov DX,offset FE_FB
        cmp al, 0FEh
        je rdy
        cmp al, 0FBh
        je rdy
        mov DX,offset FC
        cmp al, 0FCh
        je rdy
        mov DX,offset FA
        cmp al, 0FAh
        je rdy
        mov DX,offset F8
        cmp al, 0F8h
        je rdy

```

```

        mov DX,offset FD
        cmp al, 0FDh
        je rdy
        mov DX,offset F9
        cmp al, 0F9h
        jne exception
exception:
        call print
rdy:
        mov AH,09h
        int 21h

        mov DX,offset TypeMSDOS
        mov AH,09h
        int 21h

        mov AH,30h
        int 21h

        cmp AL,0
        jne modern
        mov DX,offset old
        mov AH,09h
        int 21h
modern:
        mov SI,offset new
        add SI,1
        call BYTE_TO_DEC
        add SI,4
        mov AL,AH
        call BYTE_TO_DEC
        mov DX,offset new
        mov AH,09h
        int 21h

        mov AH,30h
        int 21h

        mov DX,offset serial
        mov AH,09h
        int 21h
        mov AL,BH
        call BYTE_TO_DEC
        mov DL, [SI]
        mov AH, 02h
        int 21h
        mov DL, [SI+1]
        int 21h
        mov DL, [SI+2]
        int 21h

        mov AH,30h
        int 21h

        mov DX,offset serial2
        mov AH,09h

```

```
    int 21h
    mov AL, BL
    call PRINT
    mov AL, CH
    call PRINT
    mov AL, CL
    call PRINT

    xor AL,AL
    mov AH,4Ch
    int 21H
TESTPC ENDS
    END START;
```


ПРИЛОЖЕНИЕ Б

ИСХОДНЫЙ КОД ПРОГРАММЫ LR1EXE.ASM

```
AStack SEGMENT STACK 'STACK'
    DW 100h DUP(?)
AStack ENDS

DATA SEGMENT
    TypePc db 'Your IBM PC type is ','$'
    FF db 'PC',10,13,'$'
    FE_FB db 'PC/XT',10,13,'$'
    FA db 'PS2 ver. 30',10,13,'$'
    FC db 'PS2 ver. 50/60 or AT',10,13,'$'
    F8 db 'PS2 ver. 80',10,13,'$'
    FD db 'PCjr',10,13,'$'
    F9 db 'PC Convertible',10,13,'$'
    TypeMSDOS db 'Your MSDOS type is ','$'
    old db '<2.0',10,13,'$'
    new db '0x.0y',10,13,'$'
    serial db 'Your serial OEM number is ','$'
    serial2 db 10,13,'Your serial user number is ','$'
DATA ENDS

CODE SEGMENT
    ASSUME SS:AStack,DS:DATA,CS:CODE

PRINT PROC near
    push CX
    push DX

    call BYTE_TO_HEX
    mov CH, AH

    mov DL, AL
    mov AH, 02h
    int 21h

    mov DL, CH
    mov AH, 02h
    int 21h

    pop DX
    pop CX
    ret
PRINT ENDP

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
    pop CX
    ret
BYTE_TO_HEX ENDP

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_1
    or AL,30h
    mov [SI],AL
end_1: pop DX
    pop CX
    ret
BYTE_TO_DEC ENDP

BEGIN PROC far
    mov AX, DATA
    mov DS, AX

    mov ax,0F000h
    mov es,ax
    mov al,es:[0FFFEh]

    mov DX,offset TypePc
    mov AH,09h
    int 21h

    mov DX,offset FF
    cmp al, 0FFh
    je rdy
    mov DX,offset FE_FB
    cmp al, 0FEh
    je rdy
    cmp al, 0FBh
    je rdy
    mov DX,offset FC
    cmp al, 0FCh

```

```

        je rdy
        mov DX,offset FA
        cmp al, 0FAh
        je rdy
        mov DX,offset F8
        cmp al, 0F8h
        je rdy
        mov DX,offset FD
        cmp al, 0FDh
        je rdy
        mov DX,offset F9
        cmp al, 0F9h
        jne exception
exception:
        call print
rdy:
        mov AH,09h
        int 21h

        mov DX,offset TypeMSDOS
        mov AH,09h
        int 21h

        mov AH,30h
        int 21h

        cmp AL,0
        jne modern
        mov DX,offset old
        mov AH,09h
        int 21h
modern:
        mov SI,offset new
        add SI,1
        call BYTE_TO_DEC
        add SI,4
        mov AL,AH
        call BYTE_TO_DEC
        mov DX,offset new
        mov AH,09h
        int 21h

        mov AH,30h
        int 21h

        mov DX,offset serial
        mov AH,09h
        int 21h
        mov AL,BH
        call BYTE_TO_DEC
        mov DL, [SI]
        mov AH, 02h
        int 21h
        mov DL, [SI+1]
        int 21h
        mov DL, [SI+2]

```

```

    int 21h

    mov AH,30h
    int 21h

    mov DX,offset serial2
    mov AH,09h
    int 21h
    mov AL, BL
    call PRINT
    mov AL, CH
    call PRINT
    mov AL, CL
    call PRINT

    xor AL,AL
    mov AH,4Ch
    int 21h
BEGIN      ENDP
CODE      ENDS
END      BEGIN

```