

**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. 8383

\_\_\_\_\_

Ларин 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 их загрузки в основную память.

## Выполнение

Написан код **.COM** модуля, который определяет тип PC и версию системы. В нем читаются байты из ROM BIOS, и интерпретируются в соответствие с таблицей. Затем при помощи системного прерывания запрашивается информация о системе и так же выводится на экран. Данный код был собран в **.COM** модуль, а так же в «плохой» **.EXE** модуль в результате чего было получено предупреждение линковщика: «LINK : warning L4021: no stack segment».

### Результаты исполнения

.COM	«плохой» .EXE	«хороший» .EXE
C:\>COMSAM~1.COM AT 05.00 FF 000004 -	C:\>BADSAMPL.EXE FC 5 0 FF FF FF FF 000004 up PS up PS up PS	C:\>EXESAM~1.EXE AT 05.00 FF 000004 FC

Код представлен в приложении А.

### Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM содержит ровно один сегмент

2. EXE-программа?

EXE содержит произвольное количество сегментов

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

ORG 100h, для размещения PSP. Assume для инициализации регистров

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

Для COM некорректно указание адреса сегмента, т.к. в COM отсутствует таблица настроек.

Шестнадцатеричный xxd вид скомпонованных модулей представлен в приложении Б

### Отличие форматов файлов COM и EXE модулей

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

COM файл содержит единственный сегмент с кодом и данными. В памяти код располагается начиная с 100h

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

Плохой EXE файл содержит в себе только один сегмент с кодом и данными, и тот начинается с адреса 300h. С адреса 0h располагается заголовок и таблица настроек.

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

Плохой EXE содержит только один сегмент. Так же информация начале файла разнится и плохой EXE занимает больше памяти, т. к. по умолчанию объем до 300h зарезервирован под таблицу релокации

При помощи программы TD был загружен COM модуль

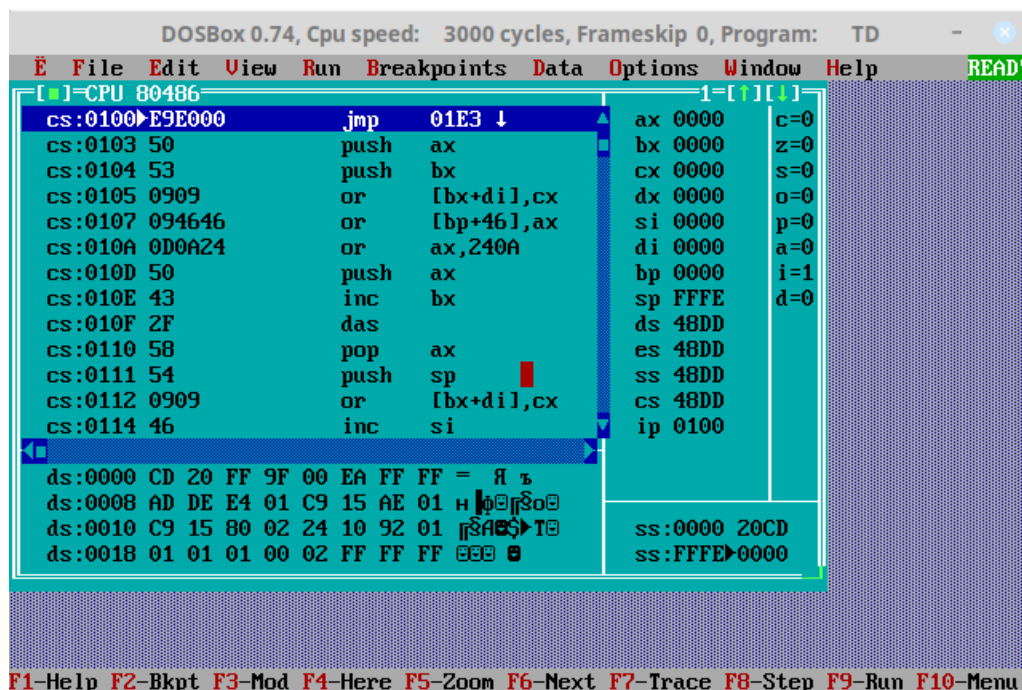


Рисунок — COM открытый в отладчике TD

### Загрузка COM модуля в основную память

1. Какой формат загрузки модуля COM? С какого адреса располагается код?  
Код вместе с данными располагаются с адреса 100h.
2. Что располагается с адреса 0?  
С адреса 0h ОС при загрузке располагает PSP
3. Какие значения имеют сегментные регистры? На какие области памяти они указывают?  
Когда COM-программа начинает работать, все сегментные регистры содержат адрес префикса программного сегмента (PSP)
4. Как определяется стек? Какую область памяти он занимает? Какие адреса?  
Стек занимает всю доступную память вместе с кодом и данными. При загрузке SP устанавливается в FFFE, а BP в 0000.

При помощи программы TD был загружен EXE модуль

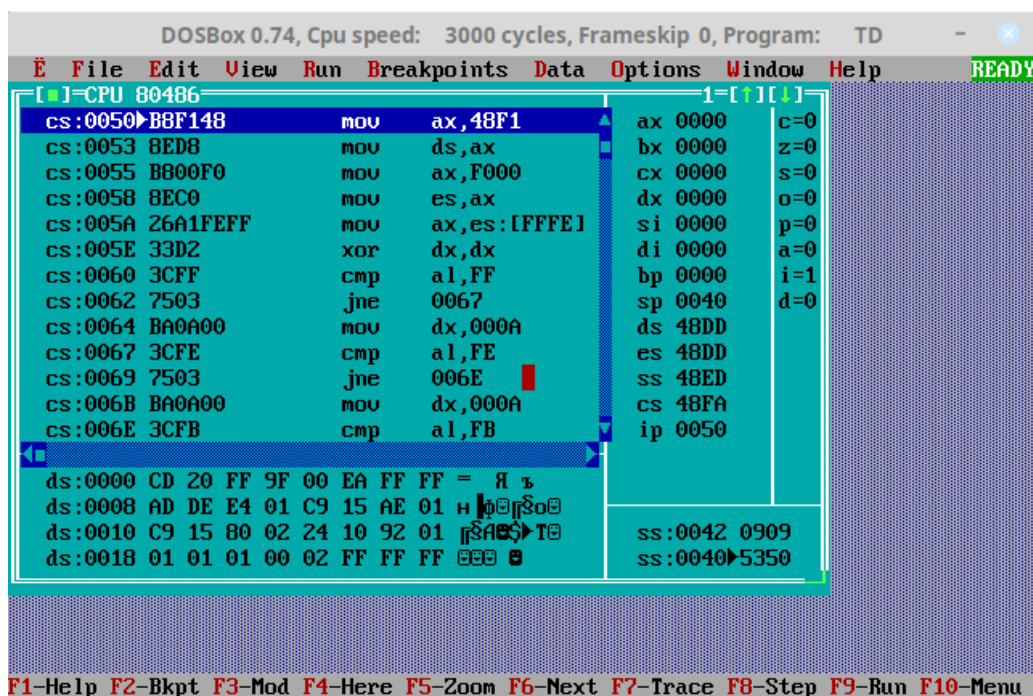


Рисунок — EXE открытый в отладчике TD

### Загрузка «хорошего» EXE модуля в основную память

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

Сегментные регистры CS, SS устанавливаются в начало соответствующего сегмента при загрузке. DS, ES устанавливаются в начало PSP

2. На что указывают регистры DS и ES?

DS, ES устанавливаются в начало PSP

3. Как определяется стек?

Память под стэе выделяется в соответствии с моделью памяти в программе или указаний в программе вручную. При загрузке программы инициализируются регистры BP и SP

4. Как определяется точка входа?

Если точка входа не указана явно, то ей является начало сегмента кода (CS). В программе точка входа указывается в конце директивой END <метка>. Метка и является точкой входа программы.

### **Выводы.**

В результате работы были разобраны некоторые концепции языка ассемблера и работы операционной системы DOS. Были исследованы различия в файлах COM и EXE, из структурой и способом загрузки.

## ПРИЛОЖЕНИЕ А

### COMSAMPLE.ASM

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

```
START:  JMP BEGIN
;data
s1 db 'PS      FF',0DH,0AH,'$'
s2 db 'PC/XT   FE,FB',0DH,0AH,'$'
s3 db 'AT      FC',0DH,0AH,'$'
s4 db 'PS2-30   FA',0DH,0AH,'$'
s5 db 'PS2-50-60 FC',0DH,0AH,'$'
s6 db 'PS2-300  F8',0DH,0AH,'$'
s7 db 'PCjr     FD',0DH,0AH,'$'
s8 db 'PC Convertible F9',0DH,0AH,'$'
s9 db 'XZ',0DH,0AH,'$'
OS_VER db '00.00',0DH,0AH,'$'
NEW_LINE db 0DH,0AH,'$'
```

```
STRING DB 'Some text      ',0DH,0AH,'$'
;procedures
```

```
DIGIT_TO_CHAR PROC near
;AL
```

```
    and al,0Fh
    cmp al,09h
    jle BLW
    add al,'A'
    sub al, 0Ah
    jmp DTC_CONT
```

```
BLW:
    add al,'0'
```

```
DTC_CONT:
    ret
```

```
DIGIT_TO_CHAR ENDP
```

```
;-----
```

```
PRINT_AS_HEX proc near
```

```
;AL - number
;breaks AX,CX,BX
    push dx
    mov bx,dx
```

```

    mov ch,al
    mov cl,4
    shr al,cl
    call DIGIT_TO_CHAR
    mov dl,al
    mov ah,02h
    int 21h
    mov al,ch
    call DIGIT_TO_CHAR
    mov dl,al
    mov ah,02h
    int 21h
    ;mov dx,bx
    pop dx
    ret
PRINT_AS_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_l
    or AL,30h
    mov [SI],AL
end_l:
    pop DX
    pop CX
    ret
BYTE_TO_DEC ENDP
;-----
;-----
BEGIN:
    mov ax,0F000h
    mov es,ax
    mov ax,es:[0FFFEh]

```

```

; mov ax, 2Ah
xor dx, dx
CASE:
    cmp al, 0FFh
    jne LS2
    mov DX, offset s2
; jmp ENDIF
LS2:
    cmp al, 0FFh
    jne SLS2
    mov DX, offset s2
SLS2:
    cmp al, 0FBh
    jne LS3
    mov DX, offset s2
LS3:
    cmp al, 0FCh
    jne LS4
    mov DX, offset s3; s5
LS4:
    cmp al, 0FAh
    jne LS6
    mov DX, offset s4
LS6:
    cmp al, 0F8h
    jne LS7
    mov DX, offset s6
LS7:
    cmp al, 0FDh
    jne LS8
    mov DX, offset s7
LS8:
    cmp al, 0F9h
    jne LS9
    mov DX, offset s8
LS9:
    cmp DX, 0
    jnz END_CASE
    mov DX, offset s9
    mov AH, 09h
    int 21h
    call PRINT_AS_HEX
END_CASE:

    cmp DX, 0
    je VER
    mov AH, 09h
    int 21h

```



VER:

```
mov AH, 30h
int 21h
mov BH,AH
mov si, offset OS_VER
inc si
call BYTE_TO_DEC
```

```
add si,3
xor AH,AH
mov AL,BH
call BYTE_TO_DEC
```

```
mov DX,offset OS_VER
mov AH,09h
int 21h
```

```
mov AH, 30h
int 21h
```

```
mov AL,BH; OEM
call PRINT_AS_HEX
```

```
mov DX,offset NEW_LINE
mov AH,09h
int 21h
```

```
mov AL,BL; Serial
call PRINT_AS_HEX
mov AL,CH
call PRINT_AS_HEX
mov AL,CL
call PRINT_AS_HEX
```

EXIT:

```
xor AL,AL
mov AH,4Ch
int 21h
```

TESTPC ENDS

END START

## **EXESAMPLE.ASM**

```
STACKSG SEGMENT PARA STACK 'Stack'
        DW      32 DUP(?)
STACKSG ENDS
```

```

DATASG SEGMENT PARA 'Data' ;SEG DATA
s1 db 'PS      FF',0DH,0AH,'$'
s2 db 'PC/XT      FE,FB',0DH,0AH,'$'
s3 db 'AT      FC',0DH,0AH,'$'
s4 db 'PS2-30      FA',0DH,0AH,'$'
s5 db 'PS2-50-60 FC',0DH,0AH,'$'
s6 db 'PS2-300      F8',0DH,0AH,'$'
s7 db 'PCjr      FD',0DH,0AH,'$'
s8 db 'PC Convertible F9',0DH,0AH,'$'
s9 db 'XZ',0DH,0AH,'$'
OS_VER db '00.00',0DH,0AH,'$'
NEW_LINE db 0DH,0AH,'$'

STRING DB 'Some text      ',0DH,0AH,'$'
DATASG ENDS ;ENDS DATA

CODE SEGMENT ;SEG CODE
; ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
; ORG 100H
ASSUME DS:DataSG, CS:Code
;procedures
DIGIT_TO_CHAR PROC near
;AL
    and al,0Fh
    cmp al,09h
    jle BLW
    add al,'A'
    sub al, 0Ah
    jmp DTC_CONT
BLW:
    add al,'0'
DTC_CONT:
    ret
DIGIT_TO_CHAR ENDP
;-----
PRINT_AS_HEX proc near
;AL - number
;breaks AX,CX,BX
    push dx
    ;mov bx,dx
    mov ch,al
    mov cl,4
    shr al,cl
    call DIGIT_TO_CHAR
    mov dl,al
    mov ah,02h
    int 21h
    mov al,ch
    call DIGIT_TO_CHAR
    mov dl,al
    mov ah,02h
    int 21h
    ;mov dx,bx
    pop dx
    ret
PRINT_AS_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

;-----

;-----
Main          PROC  FAR

    mov ax, DATASG                ;ds setup
    mov ds, ax

    mov ax,0F000h
    mov es,ax
    mov ax,es:[0FFFEh]
; mov ax,2Ah
    xor dx,dx
CASE:
    cmp al,0FFh
    jne LS2
    mov DX,offset s2
; jmp ENDIF
LS2:
    cmp al,0FEh
    jne SLS2
    mov DX,offset s2
SLS2:
    cmp al,0FBh
    jne LS3
    mov DX,offset s2
LS3:
    cmp al,0FCh
    jne LS4
    mov DX,offset s3;s5
LS4:
    cmp al,0FAh
    jne LS6

```

```

    mov DX,offset s4
LS6:
    cmp al,0F8h
    jne LS7
    mov DX,offset s6
LS7:
    cmp al,0FDh
    jne LS8
    mov DX,offset s7
LS8:
    cmp al,0F9h
    jne LS9
    mov DX,offset s8
LS9:
    cmp DX,0
    jnz END_CASE
    mov DX,offset s9
    mov AH,09h
    int 21h
    call PRINT_AS_HEX
END_CASE:

    cmp DX,0
    je VER
    mov AH,09h
    int 21h

VER:

    mov AH, 30h
    int 21h
    mov BH,AH
    mov si, offset OS_VER
    inc si
    call BYTE_TO_DEC

    add si,3
    xor AH,AH
    mov AL,BH
    call BYTE_TO_DEC

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

    mov AH, 30h
    int 21h

    mov AL,BH; OEM
    call PRINT_AS_HEX

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

    mov AL,BL; Serial
    call PRINT_AS_HEX
    mov AL,CH

```

```
    call PRINT_AS_HEX
    mov AL,CL
    call PRINT_AS_HEX

EXIT:
    xor AL,AL
    mov AH,4Ch
    int 21h
Main      ENDP
CODE      ENDS
END Main
```

## ПРИЛОЖЕНИЕ Б

### ШЕСТНАДЦАТЕРИЧНЫЙ ВИД COM

```
00000000: E9E0 0050 5309 0909 4646 0D0A 2450 432F ...PS...FF...$PC/
00000010: 5854 0909 4645 2C46 420D 0A24 4154 0909 XT..FE,FB...$AT..
00000020: 0946 430D 0A24 5053 322D 3330 0909 4641 .FC...$PS2-30..FA
00000030: 0D0A 2450 5332 2D35 302D 3630 0946 430D ..$PS2-50-60.FC.
00000040: 0A24 5053 322D 3330 3009 0946 380D 0A24 . $PS2-300..F8..$
00000050: 5043 6A72 0909 0946 440D 0A24 5043 2043 PCJR...FD...$PC C
00000060: 6F6E 7665 7274 6962 6C65 2046 390D 0A24 ONVERTIBLE F9...$
00000070: 585A 0D0A 2430 302E 3030 0D0A 240D 0A24 XZ...$00.00...$..$
00000080: 536F 6D65 2074 6578 7420 2020 2020 2020 SOME TEXT
00000090: 0D0A 2424 0F3C 097E 0704 412C 0AEB 0390 ..$.<...A,....
000000A0: 0430 C352 8AE8 B104 D2E8 E8E6 FF8A D0B4 .0.R.....
000000B0: 02CD 218A C5E8 DBFF 8AD0 B402 CD21 5AC3 ...!.....!Z.
000000C0: 5152 32E4 33D2 B90A 00F7 F180 CA30 8814 QR2.3.....0..
000000D0: 4E33 D23D 0A00 73F1 3C00 7404 0C30 8804 N3.=...S.<.T..0..
000000E0: 5A59 C3B8 00F0 8EC0 26A1 FEFF 33D2 3CFF ZY.....&...3.<.
000000F0: 7503 BA0D 013C FF75 03BA 0D01 3CFB 7503 U....<.U....<.U.
00000100: BA0D 013C FC75 03BA 1C01 3CFA 7503 BA26 ...<.U....<.U..&
00000110: 013C F875 03BA 4201 3CFD 7503 BA50 013C .<.U..B.<.U..P.<
00000120: F975 03BA 5C01 83FA 0075 0ABA 7001 B409 .U..\....U..P...
00000130: CD21 E86E FF83 FA00 7404 B409 CD21 B430 .!.N....T....!.0
00000140: CD21 8AFC BE75 0146 E875 FF83 C603 32E4 .!...U.F.U....2.
00000150: 8AC7 E86B FFBA 7501 B409 CD21 B430 CD21 ...K..U....!.0.!
00000160: 8AC7 E83E FFBA 7D01 B409 CD21 8AC3 E832 ...>..}....!...2
00000170: FF8A C5E8 2DFF 8AC1 E828 FF32 C0B4 4CCD ....-....(.2..L.
00000180: 21 !
```

### ШЕСТНАДЦАТЕРИЧНЫЙ ВИД “ПЛОХОГО” EXE

```
00000000: 4D5A 8100 0300 0000 2000 0000 FFFF 0000 MZ.....
00000010: 0000 9169 0001 0000 1E00 0000 0100 0000 ...I.....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000A0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000B0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000C0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000D0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000E0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

000000F0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000100:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000110:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000120:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000130:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000140:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000150:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000160:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000170:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000180:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000190:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001A0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001B0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001C0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001D0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001E0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000001F0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000200:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000210:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000220:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000230:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000240:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000250:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000260:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000270:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000280:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000290:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002A0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002B0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002C0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002D0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002E0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
000002F0:	0000	0000	0000	0000	0000	0000	0000	0000	.....
00000300:	E9E0	0050	5309	0909	4646	0D0A	2450	432F	...PS...FF..\$PC/
00000310:	5854	0909	4645	2C46	420D	0A24	4154	0909	XT..FE,FB..\$AT..
00000320:	0946	430D	0A24	5053	322D	3330	0909	4641	.FC..\$PS2-30..FA
00000330:	0D0A	2450	5332	2D35	302D	3630	0946	430D	..\$PS2-50-60.FC.
00000340:	0A24	5053	322D	3330	3009	0946	380D	0A24	..\$PS2-300..F8..\$
00000350:	5043	6A72	0909	0946	440D	0A24	5043	2043	PCJR...FD..\$PC C
00000360:	6F6E	7665	7274	6962	6C65	2046	390D	0A24	ONVERTIBLE F9..\$
00000370:	585A	0D0A	2430	302E	3030	0D0A	240D	0A24	XZ..\$00.00..\$..\$
00000380:	536F	6D65	2074	6578	7420	2020	2020	2020	SOME TEXT
00000390:	0D0A	2424	0F3C	097E	0704	412C	0AEB	0390	..\$.<...A,....
000003A0:	0430	C352	8AE8	B104	D2E8	E8E6	FF8A	D0B4	.0.R.....

```

000003B0: 02CD 218A C5E8 DBFF 8AD0 B402 CD21 5AC3 ...!.....!Z.
000003C0: 5152 32E4 33D2 B90A 00F7 F180 CA30 8814 QR2.3.....0..
000003D0: 4E33 D23D 0A00 73F1 3C00 7404 0C30 8804 N3.=...S.<.T..0..
000003E0: 5A59 C3B8 00F0 8EC0 26A1 FEFF 33D2 3CFF ZY.....&...3.<.
000003F0: 7503 BA0D 013C FF75 03BA 0D01 3CFB 7503 U....<.U....<.U.
00000400: BA0D 013C FC75 03BA 1C01 3CFA 7503 BA26 ...<.U....<.U..&
00000410: 013C F875 03BA 4201 3CFD 7503 BA50 013C .<.U..B.<.U..P.<
00000420: F975 03BA 5C01 83FA 0075 0ABA 7001 B409 .U..\.U..P...
00000430: CD21 E86E FF83 FA00 7404 B409 CD21 B430 .!.N....T....!.0
00000440: CD21 8AFC BE75 0146 E875 FF83 C603 32E4 .!...U.F.U....2.
00000450: 8AC7 E86B FFBA 7501 B409 CD21 B430 CD21 ...K..U....!.0.!
00000460: 8AC7 E83E FFBA 7D01 B409 CD21 8AC3 E832 ...>...}....!...2
00000470: FF8A C5E8 2DFF 8AC1 E828 FF32 C0B4 4CCD ....-....(.2..L.
00000480: 21 !

```

## ШЕСТНАДЦАТЕРИЧНЫЙ ВИД “ХОРОШЕГО” EXE

```

00000000: 4D5A C301 0200 0100 2000 0000 FFFF 0000 MZ.....
00000010: 4000 5066 5000 0D00 1E00 0000 0100 5100 @.PFP.....Q.
00000020: 0D00 0000 0000 0000 0000 0000 0000 0000 .....
00000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000A0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000B0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000C0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000D0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000E0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000000F0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000110: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000120: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000140: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000150: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000160: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000170: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000180: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000190: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001A0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001B0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001C0: 0000 0000 0000 0000 0000 0000 0000 0000 .....

```



```

000001D0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001E0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000001F0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000200: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000210: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000220: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000230: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000240: 5053 0909 0946 460D 0A24 5043 2F58 5409 PS...FF..$PC/XT.
00000250: 0946 452C 4642 0D0A 2441 5409 0909 4643 .FE,FB..$AT...FC
00000260: 0D0A 2450 5332 2D33 3009 0946 410D 0A24 ..$PS2-30..FA..$
00000270: 5053 322D 3530 2D36 3009 4643 0D0A 2450 PS2-50-60.FC..$P
00000280: 5332 2D33 3030 0909 4638 0D0A 2450 436A S2-300..F8..$PCJ
00000290: 7209 0909 4644 0D0A 2450 4320 436F 6E76 R...FD..$PC CONV
000002A0: 6572 7469 626C 6520 4639 0D0A 2458 5A0D ERTIBLE F9..$XZ.
000002B0: 0A24 3030 2E30 300D 0A24 0D0A 2453 6F6D . $00.00..$..$SOM
000002C0: 6520 7465 7874 2020 2020 2020 200D 0A24 E TEXT ..$
000002D0: 240F 3C09 7E07 0441 2C0A EB03 9004 30C3 $.<..~..A,.....0.
000002E0: 528A E8B1 04D2 E8E8 E6FF 8AD0 B402 CD21 R.....!
000002F0: 8AC5 E8DB FF8A D0B4 02CD 215A C351 5232 .....!Z.QR2
00000300: E433 D2B9 0A00 F7F1 80CA 3088 144E 33D2 .3.....0..N3.
00000310: 3D0A 0073 F13C 0074 040C 3088 045A 59C3 =..S.<.T..0..ZY.
00000320: B804 008E D8B8 00F0 8EC0 26A1 FEF 33D2 .....&...3.
00000330: 3CFF 7503 BA0A 003C FE75 03BA 0A00 3CFB <.U....<.U....<.
00000340: 7503 BA0A 003C FC75 03BA 1900 3CFA 7503 U....<.U....<.U.
00000350: BA23 003C F875 03BA 3F00 3CFD 7503 BA4D .#.<.U..?<.U..M
00000360: 003C F975 03BA 5900 83FA 0075 0ABA 6D00 .<.U..Y....U..M.
00000370: B409 CD21 E869 FF83 FA00 7404 B409 CD21 ...!.I....T....!
00000380: B430 CD21 8AFC BE72 0046 E870 FF83 C603 .0.!...R.F.P....
00000390: 32E4 8AC7 E866 FFBA 7200 B409 CD21 B430 2....F..R....!.0
000003A0: CD21 8AC7 E839 FFBA 7A00 B409 CD21 8AC3 .!...9..Z....!..
000003B0: E82D FF8A C5E8 28FF 8AC1 E823 FF32 C0B4 .-....(....#.2..
000003C0: 4CCD 21 L.!

```