**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 |
|  |  |  |

Код представлен в приложении А.

**Отличия исходных текстов COM и EXE программ**

1. COM содержит ровно один сегмент
2. EXE содержит произвольное количество сегментов
3. ORG 100h, для размещения PSP
4. Для COM некорректно указание адреса сегмента

Шестнадцатеричный xxd вид скомпонованных модулей представлен в приложении Б

**Отличие форматов файлов COM и EXE модулей**

1. COM файл содержит единственный сегмент с кодом и данными. В памяти код располагается начиная с 100h
2. Плохой EXE файл содержит в себе только один сегмент с кодом и данными, и тот начинается с адреса 300h. С адреса 0h распологается метаинформация и множество пустых ячеек памяти
3. Плохой EXE содержит только один сегмент. Так же информация начале файла разнится и плохой EXE занимает больше памяти

При помощи программы TD был загружен COM модуль

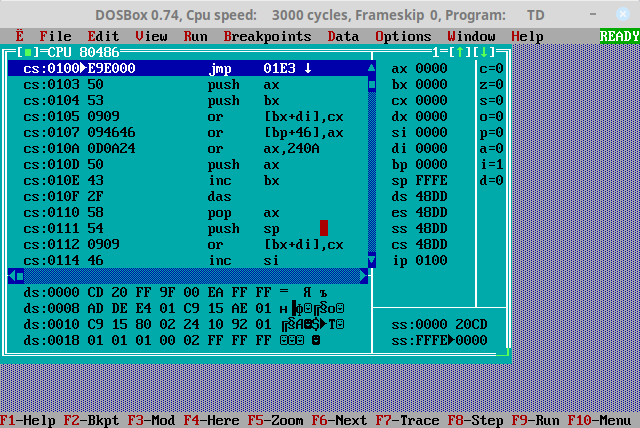


Рисунок — COM открытый в отладчике TD

**Загрузка COM модуля в основную память**

1. Код вместе с данными располагаются с адреса 100h.
2. С адреса 0h ОС при азгрузке располагает PSP
3. Когда COM-программа начинает работать, все сегментные регистры содержат адрес префикса программного сегмента (PSP)
4. Стэк занимает всю доступную память вместе с кодом и данными. При загрузке SP устанавливается в FFFE, а BP в 0000.

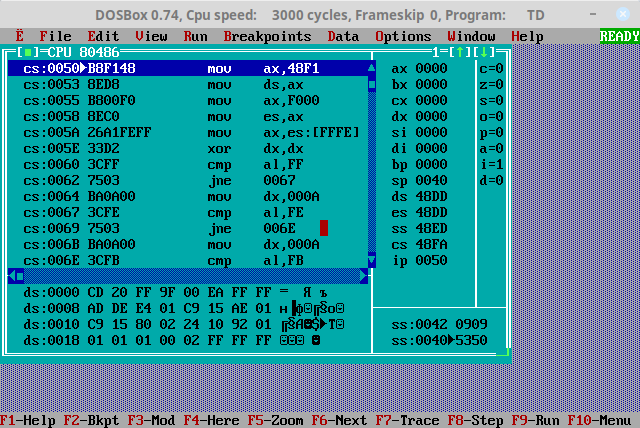
При помощи программы TD был загружен EXE модуль

Рисунок — EXE открытый в отладчике TD

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

1. Сегментные регистры CS, SS устанавливаются в начало соответствующего сегмента при загрузке. DS, ES устанавливаются в начало PSP
2. DS, ES устанавливаются в начало PSP
3. 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. 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\_l

or AL,30h

mov [SI],AL

end\_l:

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 feff 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.!