2.

; multi-segment executable file template.

data segment

; add your data here!

data1 db 00h,01h,02h,03h,04h,05h,06h,07h,08h,09h

data2 db 10dup(0)

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov si,offset data1

mov di,offset data2

mov cx,0ah

l1:

mov al,[si]

mov bl,[di]

mov [di],al

inc si

inc di

loop l1

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

3.

; multi-segment executable file template.

data segment

; add your data here!

data1 db 00h,01h,02h,03h

data2 db 02bh,02ch,02dh,02eh

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov si,offset data1

mov di,offset data2

mov cx,05h

l1:

mov al,[si]

mov bl,[di]

mov [si],bl

mov [di],al

inc si

inc di

loop l1

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

4.

; multi-segment executable file template.

data segment

; add your data here!

a db 09h

b db 02h

c db ?

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov ax,data

mov ds,ax

mov al,a

mov bl,b

add al,bl

mov c,al

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

5.

; multi-segment executable file template.

data segment

; add your data here!

a db 09h

b db 02h

c dw 00h

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov ax,data

mov ds,ax

mov al,a

mov bl,b

mul bl

mov c,ax

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

6.

; multi-segment executable file template.

data segment

; add your data here!

num db 0Fh,08h,14h,09h,05h

res db ?

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov si,offset num

mov bl,[si]

mov res,bl

mov cx,05h

l1:

mov al,[si]

cmp al,bl

jge l2

mov bl,al

mov res,bl

l2:

inc si

loop l1

mov res,bl

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

7.

; multi-segment executable file template.

data segment

; add your data here!

num db 008h,002h

avg db 00h

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov si,offset num

mov cx,02h

mov bl,00h

l1:

mov al,[si]

add al,bl

mov bl,al

inc si

loop l1

mov ax,bx

mov ah,00h

mov bx,02h

div bx

mov avg,al

; wait for any key....

mov ah, 1

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

8.

; multi-segment executable file template.

data segment

array db 0Fh,08h,14h,09h,05h

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov es, ax

mov ch,05h

UP2: mov cl,05h

mov si,offset array

UP1: mov al,[si]

mov bl,[si+1]

cmp al,bl

jc down

mov dl,[si+1]

xchg [si],dl

mov [si+1],dl

DOWN: inc si

dec cl

jnz UP1

dec ch

jnz UP2

int 3

; add your code here

mov ax, 4c00h ; exit to operating system.

int 21h

ends

end start ; set entry point and stop the assembler.

9.

; multi-segment executable file template.

data segment

segdata db 00h,01h,02h,03h,04h,05h,06h,07h,08h,09h

ends

stack segment

dw 128 dup(0)

ends

code segment

; set segment registers:

start:

mov ax,data

mov ds,ax

l3:

mov si,offset segdata

l2:

mov al,[si]

out 0f6h,al

call delay

cmp si,009h

je l3

inc si

jmp l2

proc delay near

push cx

mov cx,43h

l1:nop

nop

loop l1

pop cx

ret

endp

mov ax, 4c00h ; exit to operating system.

int 21h

ends

10.

; multi-segment executable file template.

data segment

segdata db 03fh,06h,05bh,04fh,066h,06dh,07dh,007h,07fh,06fh

ends

stack segment

dw 128 dup(0)

ends

code segment

; set segment registers:

start:

mov ax,data

mov ds,ax

mov si,offset segdata

mov cx,0ah

seg1:

mov al,[si]

out 92h,al

call delay

inc si

loop seg1

loop start

proc delay near

push cx

mov cx,10h

l1:nop

loop l1

pop cx

ret

endp

mov ax, 4c00h ; exit to operating system.

int 21h

ends

11.

; multi-segment executable file template.

data segment

msg1 db 3fh,06h,5bh,4fh,66h,6dh,7dh,07h,7fh,6fh

ends

code segment

start:

mov AX, data

mov ds, AX

mov si, offset msg1

mov al, 80h

out 8eh, al

l1:

mov dl, 00h

jmp l2

x1:inc dl

cmp dl, 60

jb l2

jmp l1

l2: mov cx,500

l4: mov ax,0000h

mov al, dl

mov bl, 10

div bl

mov bh, 00h

mov bl, al

mov al, 01h

out 86h, al

mov al, [si+bx]

out 82h, al

call delay

mov bh, 00h

mov bl, ah

mov al, 02h

out 86h, al

mov al, [si+bx]

out 82h, al

call delay

loop l4

jmp x1

proc delay near

push cx

mov cx, 43

l3:

nop

nop

loop l3

pop cx

ret

endp

ends

end start ; set entry point and stop the assembler.

12.

; multi-segment executable file template.

data segment

; add your data here!

cmd db 01h, 06h, 0Eh, 38h, 80h, 00h

data1 db "Welcome", 00h

cmd1 db 0c0h,00h

data2 db "4IT", 00h

ends

code segment

start:

; set segment registers:

mov ax, data

mov ds, ax

mov al, 80h

out 8eh, al

start1:mov si, offset cmd

l1:

call sendcommand

mov al, [si]

cmp al, 00h

je l2

jmp l1

l2:

mov si, offset data1

l8:

call senddata

mov al, [si]

cmp al, 00h

je l4

jmp l8

l4:

mov si,offset cmd1

l5:

call sendcommand

mov al, [si]

cmp al, 00h

je l6

jmp l5

l6:

mov si, offset data2

l7:

call senddata

mov al,[si]

cmp al, 00h

je start1

jmp l7

proc sendcommand near

mov al, [si]

out 82h, al

mov al, 00000010b

out 86h, al

call delay

mov al, 00000000b

out 86h, al

inc si

ret

endp

proc delay near

mov cx, 43

l3:

nop

nop

loop l3

ret

endp

proc senddata near

mov al, [si]

out 82h, al

mov al, 00000011b

out 86h, al

call delay

mov al, 00000001b

out 86h, al

inc si

ret

endp

ends

end start ; set entry point and stop the assembler.

13.

14.

; multi-segment executable file template.

data segment

msg1 db 0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f

ends

PortA EQU 90H ; Port A's port number in I/O space

PortB EQU 92H ; Port B's port number in I/O space

PortC EQU 94H ; Port C's port number in I/O space

CtrlPT EQU 96H

code segment

start:

; set segment registers:

mov ax,data

mov ds, ax

mov si,offset msg1

;mov si,offset msg1

;mov es, ax

mov bh,00h

mov dl,00h

mov al,82h

out ctrlPT,al

l1: in al,PortB

mov dl,al

call display

jmp l1

proc display near

mov cx,250d

l2:mov al,dl

mov bl,0ah

mov ah,00h

div bl

mov bl,al

mov AL,00000001b

out PortC,al

mov al,[si+bx]

out PortA,al

call delay

mov bl,ah

mov al,00000010b

out PortC,al

mov al,[si+bx]

out PortA,al

call delay

loop l2

ret

endp

proc delay near

push cx

mov cx,86

l4: nop

nop

loop l4

pop cx

ret

endp

; wait for any key....

;mov ah, 1

;int 21h

;mov ax, 4c00h ; exit to operating system.

;int 21h

ends

end start ; set entry point and stop the assembler.