Programming in Assembler

Laboratory manual

Exercise 3

Simple MS-DOS program assembling and debugging

During the Exercise No.3 students are to debug simple programs using the CodeView Debugger. On the next step other programs should be modified to improve their comfort of the usage.
Programs are attached to the documentation in `lab3-1.asm` to `lab3-6.asm` files.

During the laboratory students are to:

1. Create the project to the `lab3-1.asm` file with options for debugging and generating listing file.
2. Create the project to the `lab3-2.asm` file with options for debugging and generating listing file.
3. Assemble first project to the `*.exe` file and run the program.
4. Assemble second project to the `*.com` file and run the program.
5. Using CodeView analyze differences between `*.exe` and `*.com` executables.
6. Create the project to the `lab3-3.asm` file with options for debugging and generating listing file.
7. Assemble project to the `*.exe` file and run the program.
8. Run the CodeView debugger and analyze program execution observing placement (full address) and functions of PSP header (Pogram Segment Prefix).
9. Create the project, assemble and run the program given by the supervisor (`lab3-4.asm`, `lab3-5.asm`, `lab3-6.asm`).
10. Modify the program above to get the file name as a parameter from the command line using information from program `lab3-3.asm`.
11. Debug the modified program with CodeView and run the program.

The report should consist of:

- Title page.
- Explanation of differences between `*.com` and `*.exe` files.
- Description of PSP header fields and placement in the memory.
- Short description of file handling in assembler programs.
- Modified program listing file.
- Conclusions.

Source code (file LAB3-1.ASM):

```
;*************************************************************************
;*                                                                       *
;*               LAB3-1.ASM - Assembler Laboratory ZMiTAC                *
;*                                                                       *
;*                         Sample .EXE program                           *
;*                                                                       *
;*************************************************************************
.model small

.stack 512

.data
sample_text db "Sample EXE program...", 0Dh, 0Ah, '$'

.code
        assume ds:@data
beginning:
        mov ax, @data                ; take address of data segment
        mov ds, ax                   ; set the segment register
        mov ah, 09h                  ; write on the screen
        mov dx, offset sample_text   ; offset of sample_text (segment in
DS)
        int 21h
     mov ax, 4C00h                ; end of the program
     int 21h

end beginning
```

Source code (file LAB3-2.ASM):

```
;*************************************************************************
;*                                                                       *
;*               LAB3-2.ASM - Assembler Laboratory ZMiTAC                *
;*                                                                       *
;*                         Sample .COM program                           *
;*                                                                       *
;*************************************************************************

.model tiny

.code
     org 100h
beginning:
     mov ah, 09h                    ; write on the screen
     mov dx, offset sample_text     ; offset of sample_text (segment in DS)
     int 21h
     mov ax, 4C00h                  ; end of the program
     int 21h

sample_text DB "Sample COM program...", 0Dh, 0Ah, '$'

end beginning
```

Source code (file LAB3-3.ASM):

```
;*************************************************************************
;*                                                                       *
;*               LAB3-3.ASM - Assembler Laboratory ZMiTAC                *
;*                                                                       *
;*                        program that displays command line parameters  *
;*                                                                       *
;*************************************************************************

.model small
.286

.stack 512

.code
        assume ds:nothing
beginning:
        mov bx, 80h           ; address of parameter in PSP block
        mov cl, ds:[bx]       ; parameter length
        xor ch, ch            ; cx - length
        or  cx, cx            ; is the parameter string empty?
        jz  ending
looping:
        inc bx
        mov dl, ds:[bx]       ; load next character
        mov ah, 02h
        int 21h               ; display character
        loop looping
ending:
     mov ax, 4C00h            ; end of the program
     int 21h

end beginning
```

Source code (file LAB3-4.ASM):

```
;*************************************************************************
;*                                                                       *
;*               LAB3-4.ASM - Assembler Laboratory ZMiTAC                *
;*                                                                       *
;*       program that copies characters from keyboard to the file        *
;*                                                                       *
;*************************************************************************

comment %
*****************************************
* copy characters from keyboard to the file *
*****************************************
%
.model small
.stack 512

.data
text_file db "copy.txt", 0
character db ?
handle    dw ?

.code
        assume ds:@data
beginning:
        mov ax, @data           ; take address of data segment
        mov ds, ax              ; set the segment register
        mov ah, 3Dh             ; open file
        mov al, 1
        mov dx, offset text_file
        int 21h
        jnc opened              ; file opened
        mov ah, 3Ch             ; create file
        mov dx, offset text_file
        mov cx, 0               ; ordinary file
        int 21h
        jc  ending              ; jump if error
opened:
        mov handle, ax
        mov ah, 42h             ; go to the end of file
        mov bx, handle
        xor cx, cx              ; zero position
        xor dx, dx
        mov al, 2               ; from the end of the file
        int 21h
looping:
        mov ah, 08h
        int 21h                 ; read the character
        or  al, al              ; is character zero?
        jnz ok                  ; normal character
```

```
        mov ah, 08h             ; read second byte
        int 21h                 ; if special code
        jmp looping
ok:
        cmp al, 1Ah             ; Ctrl-Z - end of the text
        je  closing
        mov character, al
        mov ah, 02h
        mov dl, al
        int 21h                 ; display the character
        mov ah, 40h             ; write it to the file
        mov bx, handle
        mov dx, offset character
        mov cx, 1               ; one character
        int 21h
        jmp looping
closing:
        mov ah, 3Eh
        mov bx, handle
        int 21h
ending:
     mov ax, 4C00h           ; end of the program
     int 21h

end beginning
```

Source code (file LAB3-5.ASM):

```
;*************************************************************************
;*                                                                       *
;*               LAB3-5.ASM - Assembler Laboratory ZMiTAC                *
;*                                                                       *
;*            program that copies characters between files               *
;*                                                                       *
;*************************************************************************
comment %
*****************************************
* copy bytes from one file to other file    *
*****************************************
%

.model small

.stack 512

.data
file1   db "filein.txt", 0
file2   db "fileout.txt", 0

.data?
buffer    db 1024 dup (?)

.code
start:
        mov ax, seg file1
        mov ds, ax
        ; open input file
        mov dx, offset file1    ; DS:DX - ASCIIZ of filename
        mov ah, 3Dh             ; open file
        mov al, 0               ; open for reading
        int 21h                 ; call DOS function
        jc open_error           ; CF set -> error
        mov si, ax              ; file handle in si
        ; create output file
        mov dx, offset file2
        mov cx, 0               ; file attributes: none
        mov ah, 3Ch             ; create file
        int 21h
        jc create_error         ; CF set -> error
        mov di, ax              ; file handle in di
        mov ax, seg buffer
        mov ds, ax

copy_loop:
        mov dx, offset buffer
        ; read block
        mov bx, si              ; source handle
```

```
        mov cx, sizeof buffer
        mov ah, 3Fh             ; read data
        int 21h
        jc loop_error
        cmp ax, 0
        jz loop_error           ; nothing to write
        ; write block
        mov bx, di              ; destination handle
        mov cx, ax
        mov ah, 40h             ; write data
        int 21h
        jz loop_error           ; write error
        ; write '*'
        mov ah, 02h
        mov dl, '*'
        int 21h                 ; write character

        jmp copy_loop

loop_error:
        mov ah, 3Eh             ; close output file
        mov bx, di
        int 21h

create_error:
        mov ah, 3Eh             ; close input file
        mov bx, si
        int 21h
open_error:
        mov ax, 4C00h           ; exit
     int 21h
end start
```

Source code (file LAB3-6.ASM):

```
;*************************************************************************
;*                                                                       *
;*              LAB3-6.ASM - Assembler Laboratory ZMiTAC                 *
;*                                                                       *
;*    program that displays characters from the file on the screen       *
;*                                                                       *
;*************************************************************************
comment %
*****************************************
* display bytes from the text file      *
*****************************************
%

.model small

.stack 512

.data
file      db "disp.txt", 0
character db ?
handle    dw ?

.code
start:
        mov ax, seg file
        mov ds, ax
        ; open input file
        mov dx, offset file      ; DS:DX - ASCIIZ of filename
        mov ah, 3Dh              ; open file
        mov al, 0                ; open for reading
        int 21h                  ; call DOS function
        jc open_error            ; CF set -> error
        mov handle, ax           ; file handle

copy_loop:
        ; read character
        mov dx, offset character ; buffer address
        mov bx, handle           ; file handle
        mov cx, sizeof character ; buffer size
        mov ah, 3Fh              ; read data
        int 21h
        jc close_file            ; jump on read error
        cmp ax, 0
        jz close_file            ; nothing to display
        ; display character
        mov ah, 02h
        mov dl, character
        int 21h                  ; write character
```

```
        jmp copy_loop

close_file:
        mov ah, 3Eh             ; close input file
        mov bx, handle
        int 21h

open_error:
        mov ax, 4C00h           ; exit
     int 21h
end start
```