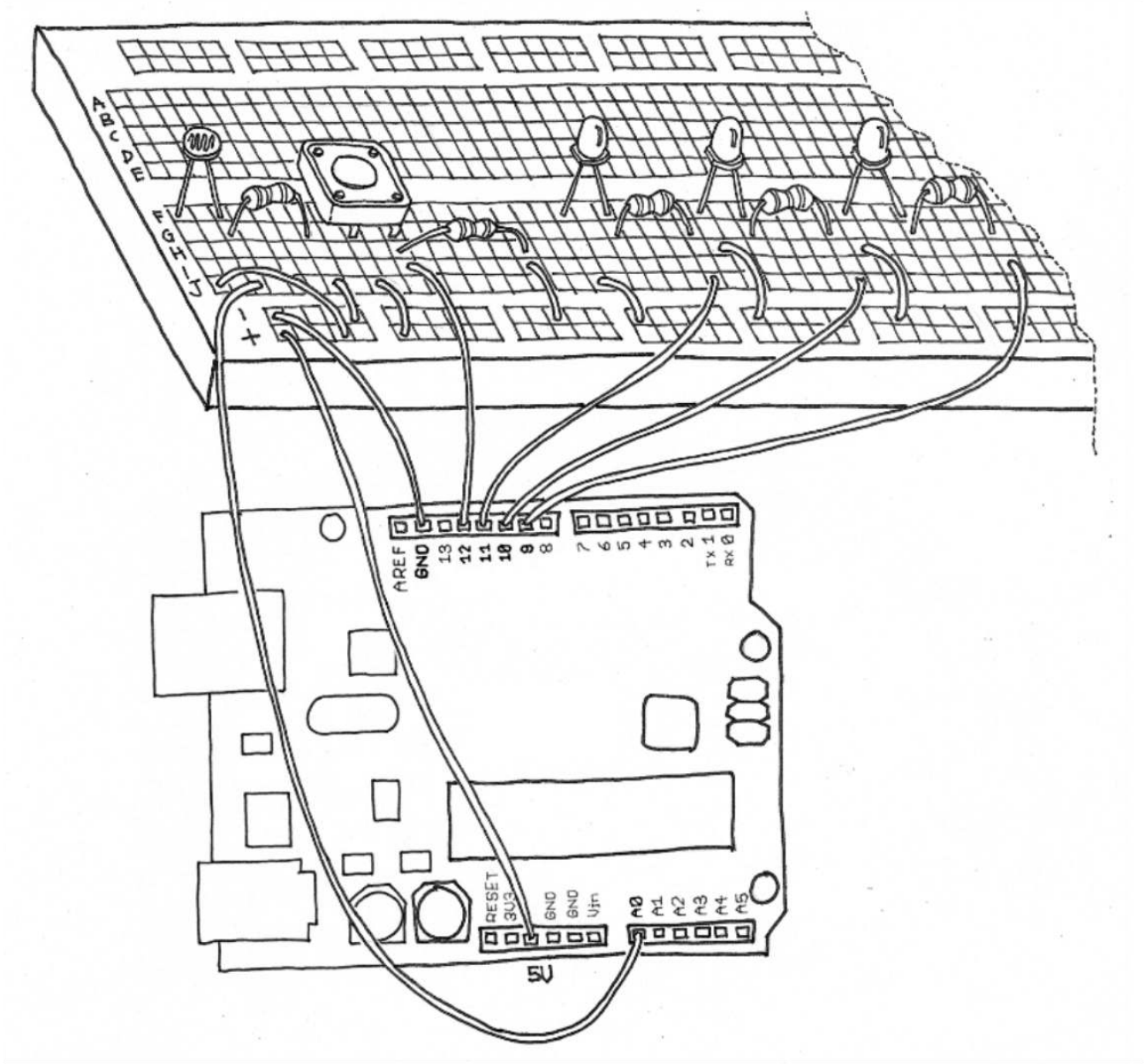
CS/EEE/INSTR F241
Microprocessor Programming and Interfacing

Lab 6 - File Operations in DOS



Dr. Vinay Chamola and Anubhav Elhence

Introduction to DOS Interrupts for files

All DOS files are sequential files. All sequential files are stored and accessed from the beginning of the file towards the end. File is usually accessed through DOS INT 21H function calls. In this session we will practice, how to create, read and write a file. There are two ways of handling a file. First via the file control block and second via the file handle. We will use file handle, as this is the most common and easier of the two methods.

File handle

DOS File Handle Functions - a group of INT 21h functions that allows DOS to track open file data in its own internal tables. File Handle Functions also permit users to specify file path names. For the purposes of the following discussion, reading means copying all or part of an existing file into memory; writing a file means copying data from memory to a file; rewriting a file means replacing a file's content with other data.

Additional information about the File Handle, File Pointer, and the Error Codes is available at the bottom.

File Interrupts

Function 3Ch: Create a File

Input:

AH = 3Ch

DS:DX = address of filename

(an ASCIIZ string ending with a zero byte)

e.g of ASCIIZ – 'C:\MASM611\BIN\abc.txt',0

CL = attribute

Attribute – Bit map

BITS	7	6	5	4	3	2	1	0
Description	Shareable	-	Archive	Directory	Vol. Label	System	Hidden	Read-Only

Output:

If successful CF =0 , AX = file handle

Error: if CF = 1, AX = error code (3, 4, or 5)

Function 3Dh: Open an existing file

Input:

AH = 3DH

AL = access and sharing modes

0 = open for reading

1 = open for writing

2 = open for read/write

DS:DX = ASCIZ filename

Output:

CF clear if successful, AX = file handle

CF set on error AX = error code (01h,02h,03h,04h,05h,0Ch,56h)

Function 40h: Write to a file

Input:

AH = 40h

BX = file handle

CX = number of bytes to write

DS:DX = data address

Output:

AX = count of bytes written.

If AX < CX, error (disk full).

If CF = 1, AX = error code (5, 6)

Function 3Eh: Close a file

Input:

AH = 3Eh

BX = file handle

Output:

Error if CF = 1, AX = error code (6)

Function 3Fh: Read an existing file

Input:

AH = 3Fh

BX = file handle

CX = number of bytes to read

DS:DX = memory buffer address

Output:

AX = count of bytes actually read.

If AX = 0 or AX < CX, EOF

If CF = 1, AX = error code (5, 6)

Function 41h: Delete a file

Input:

AH = 41H

DS: DX = address of the ASCII-Z string file name

Output:

AX = error code if carry is set

Function 56h: Rename a file

Input:

AH = 56H

DS:DX = ASCIZ filename of existing file

ES:DI = ASCIZ new filename

CL = attribute mask

Output:

CF clear if successful

CF set on error, AX= error code (02h,03h,05h,11h)

Example to Create a file

```
1   .model tiny
2   .data
    3 references
3   fname   db 'test.txt',0
    5 references
4   handle  dw ?
5   .code
6   .startup
7       mov ah, 3ch
8       lea dx, fname
9       mov cl, 1h
10      int 21h
11      mov handle, ax
12  .exit
    2 references
13  end
```

Example to Open, Write and Close a File

```
.model tiny
.data
4 references
fname   db 'sec.txt',0
6 references
handle  dw ?
3 references
msg     db 'MuP docks!'
.code
.startup

    ; Create a file if it
    ; is not existing
    mov ah, 3ch
    lea dx, fname
    mov cl, 1h
    int 21h
    mov handle, ax

    ; open file
    mov ah, 3dh
    mov al, 1h
    lea dx, fname
    int 21h
    mov handle, ax

    ; write msg to file
    mov ah, 40h
    mov bx, handle
    mov cx, 10
    lea dx, msg
    int 21h

    ; close file
    mov ah, 3eh
    int 21h
.exit
```

Example to Open, Write and Close a File

```
.model tiny
.data
4 references
fname   db 'USER.txt', 0
6 references
handle  dw ?
3 references
msg db 20 dup('$')
.code
.startup
    ; open file
    mov ah, 3dh
    mov al, 0h
    lea dx, fname
    int 21h
    mov handle, ax

    ; read content into msg
    mov ah, 3fh
    mov bx, handle
    mov cx, 10
    lea dx, msg
    int 21h

    ; print msg
    lea dx, msg
    mov ah, 09h
    int 21h

    ; close file
    mov ah, 3eh
    int 21h
.exit
2 references
end
```

Lab Task 1:

Read data from console and write to file

1. Print line on screen asking “Enter your name:”
2. Give your name as input, and save it in a local variable
3. Create a new text file
4. Write your name to this file

week6_q1.asm > ...

```
1    .model tiny
2    .data
     0 references
3    str1     db   'Enter your name: $'
4
     0 references
5    max1     db   32
     0 references
6    act1     db   ?
     0 references
7    inp1     db   32 dup('$')
8
     7 references
9    fname    db   'testing.txt',0
     9 references
10   handle   dw   ?
11   .code
12   .startup
13
14       ; WRITE
15       ; YOUR
16       ; CODE
17       ; HERE
18
19   .exit
     3 references
20   end
21
```

Lab task 2

Take substring from 2 files, and write to a third file

1. Part A

1. Create two files, “name.txt” and “id.txt”
2. Write your first name to “name.txt”, and your ID to “id.txt” by taking inputs from the terminal prompt

```
.model tiny
.data
0 references
fname1  db      'name.txt',0
1 reference
fname2  db      'id.txt',0
1 reference
handle1 dw      ?
0 references
handle2 dw ?

0 references
msg1            db 'Anubhav'
0 references
len1            db 06h
0 references
msg2            db '2021PHXP0426P'
0 references
len2            db 0dh
.code
.startup

        ; WRITE
        ; YOUR
        ; CODE
        ; HERE


.exit
4 references
end
```

2. Part B

1. Create 3rd file “splice.txt”
2. Write a string in a file by concatenating the two strings in the following form :
3. Write the new string in the file called

Example: If name = "Anubhav" & id = "2021PHXP0426P"

then string = "2021PHXP0426PAnubhav"

week6_q2_partb.asm > end

```
1    .model tiny
2    .data
     0 references
3    fname1  db 'name.txt',0
     1 reference
4    handle1 dw ?
     1 reference
5    fname2  db  'id.txt',0
     0 references
6    handle2 dw ?
     0 references
7    fname3  db  'splice.txt',0
     0 references
8    handle3 dw  ?
9
     0 references
10   part1       db  8 dup('$')
     0 references
11   part2       db 6 dup('$')
12
13   .code
14   .startup
15
16       ; WRITE
17       ; YOUR
18       ; CODE
19       ; HERE
20
21   .exit
     4 references
22   end
```


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्