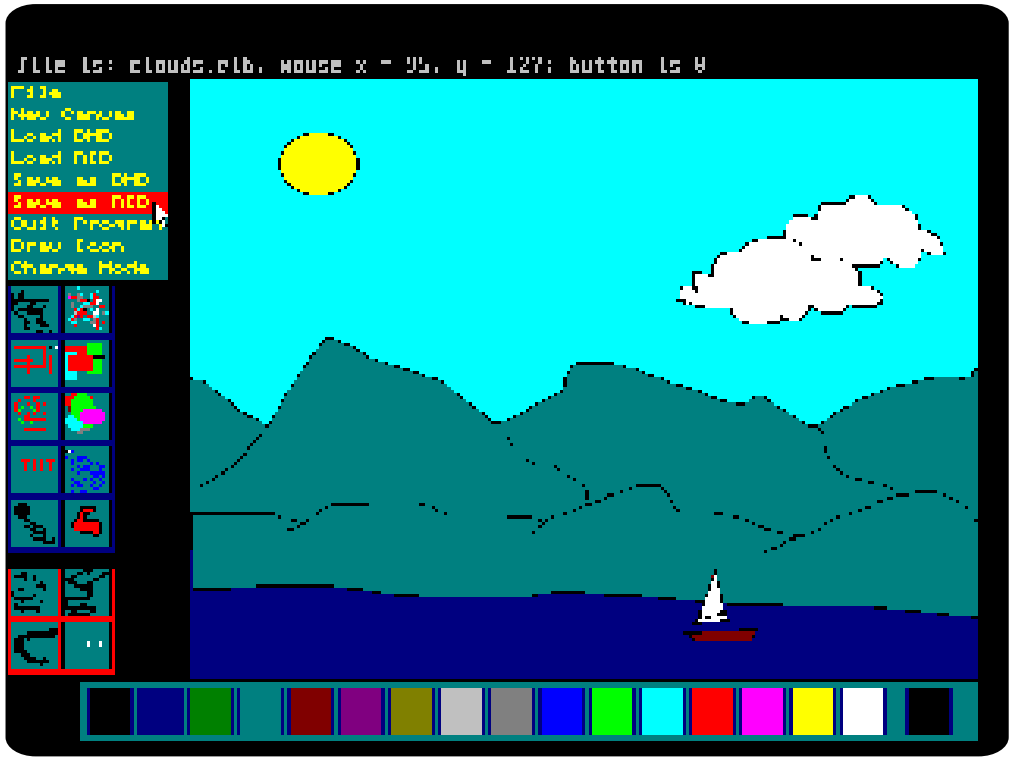
CS/EEE/INSTR F241
Lab 6 – File Operations in DOS

Anubhav Elhence

DOS interrupts for file operations

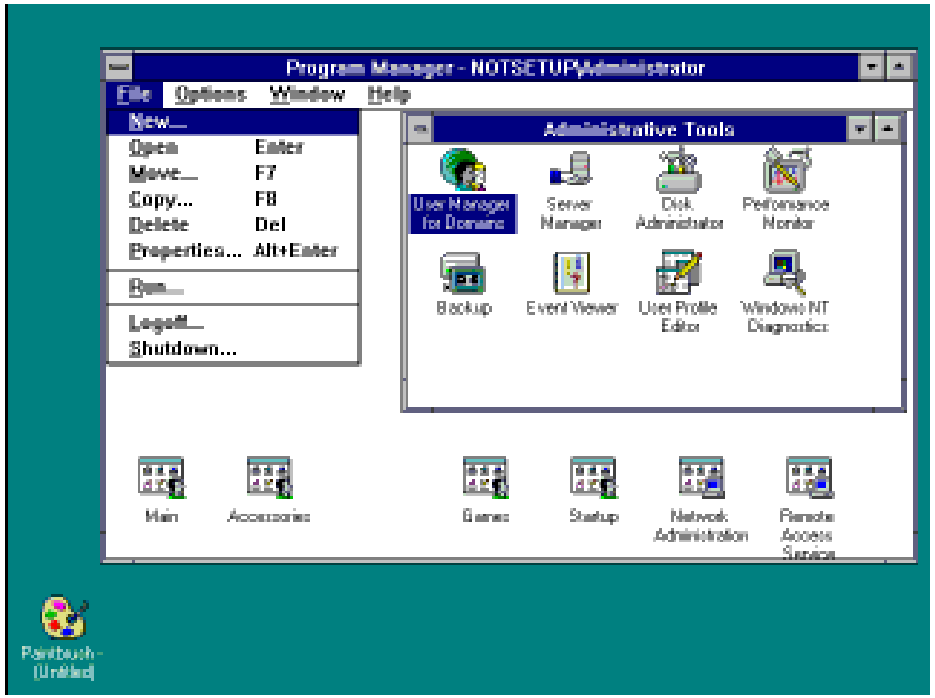


- DOS (Disk Operating System) is an operating system that was widely used in personal computers in the 1980s and early 1990s. One of the key features of DOS was its ability to access files on disk drives. DOS provides a set of interrupts, or software interrupts, which allow programs to interact with the operating system to access and manipulate files. These interrupts provide a simple and standardized way for programs to perform common file operations, such as creating, opening, reading, writing, and closing files.

DOS interrupts for file operations

The DOS interrupts for files are still relevant today, as many programming languages and tools still use DOS-style file I/O. While modern operating systems provide more sophisticated file APIs, understanding how DOS interrupts work can be useful for low-level file I/O operations on older systems or embedded devices.

The DOS interrupts for files are accessed through the INT 21h interrupt, which is a software interrupt that allows programs to call various DOS services, including file I/O. Each file-related operation is performed by calling a specific function code, or "sub-function," using the AH register. The other registers are used to pass input parameters to the interrupt and receive output values.



DOS File handle Functions

- In DOS, a file handle is a unique identifier used to access an open file. The DOS file handle functions are a set of interrupts that allow programs to create, open, read from, write to, and close files using file handles. Here are some of the most commonly used DOS file handle functions:

- INT 21h, AH=3Ch: Create File

  This interrupt is used to create a new file and returns a file handle that can be used to access the file.

- INT 21h, AH=3Dh: Open File

  This interrupt is used to open an existing file and returns a file handle that can be used to access the file.

- INT 21h, AH=3Fh: Read From File

  This interrupt is used to read data from an open file using a specified file handle.

- INT 21h, AH=40h: Write To File

  This interrupt is used to write data to an open file using a specified file handle.

- INT 21h, AH=3Eh: Close File

  This interrupt is used to close an open file using a specified file handle.

- INT 21h, AH=44h: Get File Information

  This interrupt is used to retrieve information about a file using a specified file handle.

- INT 21h, AH=4Eh: Find First File

  This interrupt is used to search for the first file that matches a specified file pattern and returns a file handle that can be used to access the file.

INT 21h, AH=3Ch: Create File

- INT 21h, AH=3Ch is a DOS interrupt used to create a new file. The function code for creating a new file is passed in the AH register, while other input parameters are passed in other registers as follows:
- CX: Specifies the file attributes. The most commonly used attributes are:
  - 00h: Normal file
  - 01h: Read-only file
  - 02h: Hidden file
  - 04h: System file
- DX: Specifies the address of a null-terminated string that contains the filename.
- DS: Specifies the segment address of the filename string.
- AL: Specifies the action to take if the file already exists. The most commonly used actions are:
  - 00h: Create a new file or truncate an existing file.
  - 01h: Create a new file or fail if the file already exists.
  - 02h: Open an existing file or fail if the file does not exist.
- When the interrupt is executed, DOS checks if the specified file exists. If the file exists, the action specified in AL determines what happens next.
- If the file does not exist, a new file is created and a file handle is returned in the AX register. The file handle is a unique identifier that is used in subsequent file access operations. If the interrupt fails, the CF flag is set and the AX register contains an error code.

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 3Eh: Close a file

- Input:
  - AH = 3Eh
  - BX = file handle
- Output:
  - Error if CF = 1, AX = error code (6)

Follow Along example:

- Creating a random file

week6_c1.asm > ...

```
1    .model tiny
2    .data
     4 references
3    fname   db 'testing',0
     6 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
```


Anubhav Elhence and Dr. Vinay Chamola

Follow Along example 2:

- Writing into a file

week6_c2.asm > ...

```
1    .model tiny
2    .data
     4 references
3    fname   db 'second.txt',0
     6 references
4    handle  dw ?
     3 references
5    msg     db 'MuP rocks!'
6    .code
7    .startup
8
9        ; Create a file if it
10       ; is not existing
11       mov ah, 3ch
12       lea dx, fname
13       mov cl, 1h
14       int 21h
15       mov handle, ax
16
17       ; open file
18       mov ah, 3dh
19       mov al, 1h
20       lea dx, fname
21       int 21h
22       mov handle, ax
23
24       ; write msg to file
25       mov ah, 40h
26       mov bx, handle
27       mov cx, 10
28       lea dx, msg
29       int 21h
30
31       ; close file
32       mov ah, 3eh
33       int 21h
34   .exit
     2 references
35   end
```


Anubhav Elhence and Dr. Vinay Chamola

Follow Along example 3:

- Reading from a file

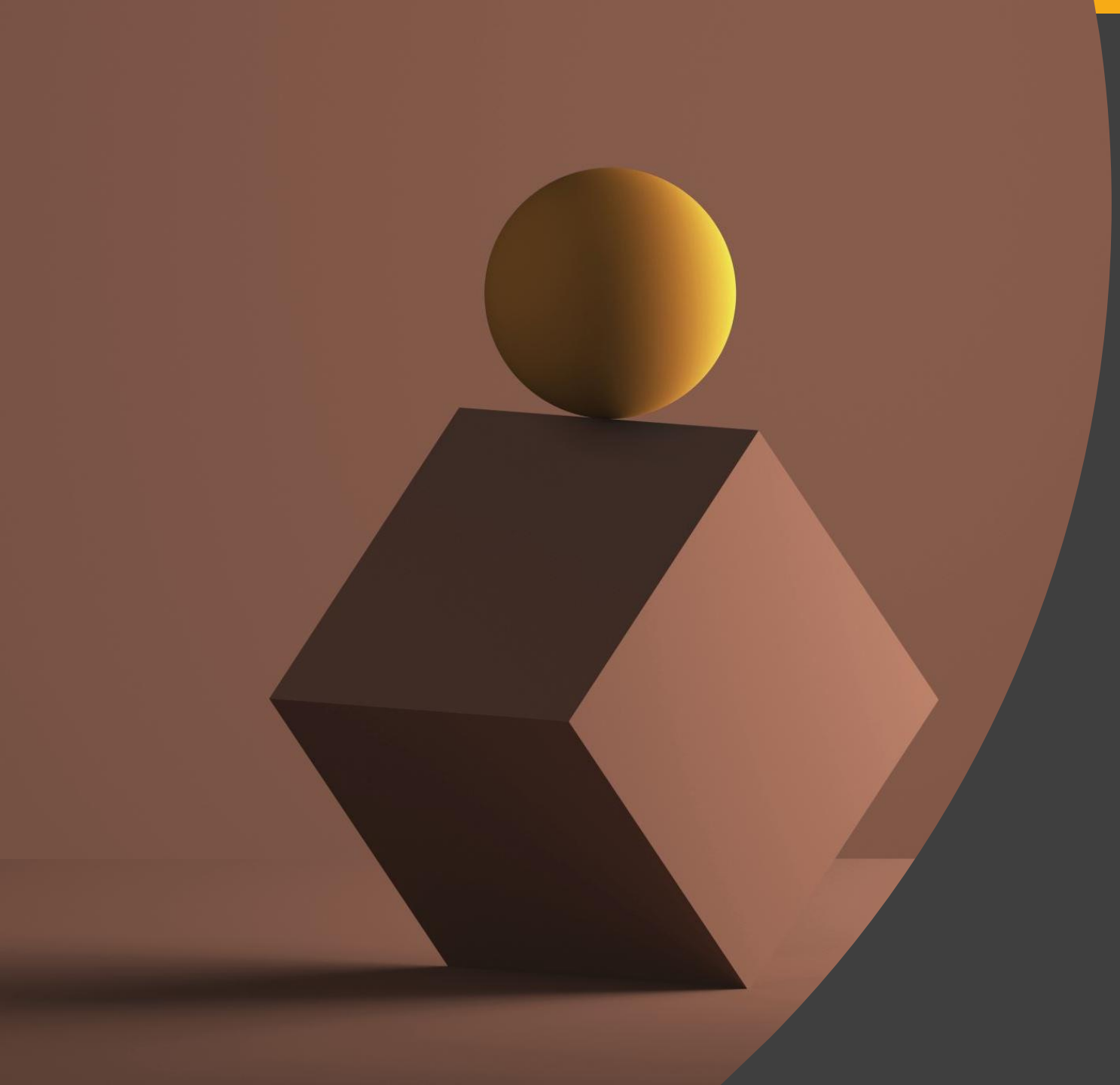
week6_c3.asm > ...

```
1   .model tiny
2   .data
    4 references
3   fname   db 'USER.txt', 0
    6 references
4   handle  dw ?
    3 references
5   msg db 20 dup('$')
6   .code
7   .startup
8       ; open file
9       mov ah, 3dh
10      mov al, 0h
11      lea dx, fname
12      int 21h
13      mov handle, ax
14
15      ; read content into msg
16      mov ah, 3fh
17      mov bx, handle
18      mov cx, 10
19      lea dx, msg
20      int 21h
21
22      ; print msg
23      lea dx, msg
24      mov ah, 09h
25      int 21h
26
27      ; close file
28      mov ah, 3eh
29      int 21h
30  .exit
    2 references
31  end
```


Anubhav Elhence and Dr. Vinay Chamola

Time for Lab Tasks:

Please check the description of this video.



Thankyou

Anubhav Elhence and Dr. Vinay Chamola