CS10003:
Programming & Data Structures

Dept. of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Autumn 2020

File Handling

Storage seen so far

- All variables stored in memory
- Problem: the contents of memory are wiped out when the computer is powered off
- Example: Consider keeping students' records
  - 100 students records are added in array of structures
  - Machine is then powered off after sometime
  - When the machine is powered on, the 100 records entered earlier are all gone!
  - Have to enter again if they are needed

Solution: Files

- A named collection of data, stored in secondary storage like disk, CD-ROM, USB drives etc.
- Persistent storage, not lost when machine is powered off
- Save data in memory to files if needed (file write)
- Read data from file later whenever needed (file read)

Organization of a file

- Stored as sequence of bytes, logically contiguous
  - May not be physically contiguous on disk, but you do not need to worry about that
- The last byte of a file contains the end-of-file character (**EOF**), with ASCII code 1A (hex).
  - While reading a text file, the EOF character can be checked to know the end
- Two kinds of files:
  - Text : contains ASCII codes only
  - Binary : can contain non-ASCII characters
    - Example: Image, audio, video, executable, etc.
    - EOF cannot be used to check end of file

Basic operations on a file

- Open
- Read
- Write
- Close
- Mainly we want to do read or write, but a file has to be opened before read/write, and should be closed after all read/write is over

Opening a File: fopen()

- FILE * is a datatype used to represent a pointer to a file
- fopen takes two parameters, the name of the file to open and the mode in which it is to be opened
- It returns the pointer to the file if the file is opened successfully, or NULL to indicate that it is unable to open the file

Example: opening file.dat for write

```
FILE *fptr;
char filename[ ]= "file2.dat";
fptr = fopen (filename,"w");
if (fptr == NULL) {
    printf (“ERROR IN FILE CREATION”);
    /* DO SOMETHING */
}
```

Modes for opening files

- The second argument of fopen is the mode in which we open the file.

Modes for opening files

- The second argument of fopen is the mode in which we open the file.
  - "r" : opens a file for reading (can only read)
    - Error if the file does not already exists
    - "r+" : allows write also

Modes for opening files

- The second argument of fopen is the mode in which we open the file.
  - "r" : opens a file for reading (can only read)
    - Error if the file does not already exists
    - "r+" : allows write also
  - "w" : creates a file for writing (can only write)
    - Will create the file if it does not exist
    - Caution: writes over all previous contents if the flle already exists
    - "w+" : allows read also

Modes for opening files

- The second argument of fopen is the mode in which we open the file.
  - "r" : opens a file for reading (can only read)
    - Error if the file does not already exists
    - "r+" : allows write also
  - "w" : creates a file for writing (can only write)
    - Will create the file if it does not exist
    - Caution: writes over all previous contents if the flle already exists
    - "w+" : allows read also
  - "a" : opens a file for appending (write at the end of the file)
    - "a+" : allows read also

The exit() function

- Sometimes error checking means we want an emergency exit from a program
- Can be done by the exit() function
- The exit() function, called from anywhere in your C program, will terminate the program at once

Usage of exit( )

```
FILE *fptr;
char filename[]= "file2.dat";
fptr = fopen (filename,"w");
if (fptr == NULL) {
    printf (“ERROR IN FILE CREATION”);
   /* Do something */
   exit(-1);
}
………rest of the program………
```

Thank You!