



Department of Computer Engineering

Academic Year : 2020-21

SUBJECT: DATA STRUCTURES AND ALGORITHM

CLASS: SE COMPuTer – 1

SEMESTER: 4

ASSIGNMENT NO. : DSA OCW

DATE OF SUBMISSION: 23/06/2021

NAME OF STUDENT: Kaustubh S Kabra

ERP NO: 34

TEAMS SR NO: 20

TOPIC: FILE INPUT/OUTPUT

WEBSITE URL REFERRED: shorturl.at/etZ15

Summary/Abstract/Review:

A file is used to store programs, executables, pictures, music, etc. A stream is a flow of input or output data like characters, numbers or bytes. A series of input coming from an input device to a program is called as input stream. Similarly, a series of output, from a program to an output device is known as output stream.

INPUT/OUTPUT IN C

C has no built in statements for input and output. A library of functions is supplied to perform these operations. The I/O library functions are listed in the “header” file <stdio.h>. There are 3 standard streams in C. They are:

1. The standard input stream, also called the stdin. It is normally connected to the keyboard.
2. Standard output stream, also called stdout. It is normally connected to the display screen.
3. Standard error stream, also called stderr. It is normally connected to the screen.

The input functions like scanf() and getchar() normally read from stdin, while the output functions like printf() and putchar() normally write to stdout.

FILE ACCESS

Files from different location need to be connected to the program.

Reading from or writing to a file in C requires 3 basic steps:

1. Open the file
2. Do all the reading or writing.
3. Close the file.

Internally, a file is referred to using a file pointer which points to a structure that contains info about the file.

OPENING A FILE

Declare a pointer and open a file using the function fopen().

Syntax-

FILE *fp;

fp=fopen(name, mode);

Example:

```
fp=fopen("data.txt","r");
```

MODES OF FILE OPENING

1. "r" – Opening an existing file for reading only.
2. "w" – Open the file for writing only.
3. "a" – Open the file for append access, i.e. writing at the end of file only.

END OF FILE

The end of file indicator informs the program when there is no more data to be processed. There are a number of ways to test for the end-of-file condition. One is to use the feof function which returns a true or false condition. Another way is to use the value returned by the fscanf function.

CLOSING FILES

The statements:

```
fclose (fptr1);
```

```
fclose(fptr2);
```

will close the files and release the file descriptor space and I/O buffer memory.

Closing a file is very important, especially with output files because the output is often buffered. If the file is not closed, then this buffer may be flushed to the monitor or the hard disk.

COMMAND LINE ARGUMENTS

Parameters can be passed to C programs by giving arguments when the program is executed.

Example:

```
>echo hello, world
```

Prints the output

```
Hello, world
```

The main program in C is called with two implicit arguments argc and argv. argc is an integer value for the number of arguments in the command line. If no arguments are there, then argc=1 and the program name is the only argument. The *argv[] is a pointer to a pointer. It is an array of strings.

1. argv[0] is the name of the program.
2. argv[1] is the first argument.
3. argv[2] is the next.....and so on....
4. argv[argc] is a null pointer.

Conclusion:

Thus we have studied about files and different streams present in C. We have studied the various modes of opening a file. We have understood the end of file. We have observed how to close a file and different reasons why is it important to close a file. We also have studied command line arguments.

Name & Sign of Subject In-charge:

Dr. K. S. Wagh

Marks: