JECRC Foundation

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Year & Semester - B.Tech I year (I Semester)

Subject - Programming for Problem Solving

Presented by - Ms. Abhilasha /Ms. Yogita Punjabi/

Mr. Gajendra Sharma

Designation - Asst. Professor

Department - Computer Science (First Year)

VISSION OF INSTITUTE

To become a renowned centre of outcome based learning, and work towards academic, professional, cultural and social enrichment of the lives of individuals and communities.

MISSION OF INSTITUTE

❖**Focus on evaluation of learning outcomes and motivate students to inculcate research aptitude by project based learning.**

❖**Identify, based on informed perception of Indian, regional and global needs, the areas of focus and provide platform to gain knowledge and solutions.**

❖**Offer opportunities for interaction between academia and industry.**

❖**Develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders may emerge.**

Programming for Problem Solving : Course Outcomes

Students will be able to:
CO1: Understand concept of low-level and high-level languages, primary and secondary memory. Represent algorithm through flowchart and pseudo code for problem solving.
CO2: Represent and convert numbers & alphabets in various notations.
CO3: Analyze and implement decision making statements and looping.
CO4: Apply pointers, memory allocation and data handling through files in ‘C’ Programming Language.

Dynamic memory allocation in C

As you know, an array is a collection of a fixed number of values. Once the size of an array is declared, you cannot change it.

Sometimes the size of the array you declared may be insufficient. To solve this issue, you can allocate memory manually during run-time. This is known as dynamic memory allocation in C programming.

To allocate memory dynamically, library functions are

- malloc()
- calloc()
- realloc()
- free()

These functions are defined in the <stdlib.h> header file.

Syntax Of Function

function	Syntax
malloc()	malloc (number *sizeof(int));
calloc()	calloc (number, sizeof(int));
realloc()	realloc (pointer_name, number * sizeof(int));
free()	free (pointer_name);

C malloc()

- The name "malloc" stands for memory allocation.
- The malloc() function reserves a block of memory of the specified number of bytes. And, it returns a pointer of void which can be casted into pointers of any form.
- malloc function does not initialize the memory allocated during execution. It carries garbage value.
- malloc function returns null pointer if it couldn't able to allocate requested amount of memory.

Example

- ptr = (float*) malloc(100 * sizeof(float));
- The above statement allocates 400 bytes of memory. It's because the size of float is 4 bytes. And, the pointer ptr holds the address of the first byte in the allocated memory.

C calloc()

- The name "calloc" stands for contiguous allocation.
- The malloc() function allocates memory and leaves the memory uninitialized. Whereas, the calloc() function allocates memory and initializes all bits to zero.
-

  Example:
- ptr = (float*) calloc(25, sizeof(float));
- The above statement allocates contiguous space in memory for 25 elements of type float.

Cont...

C free()

Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on their own. You must explicitly use free() to release the space.

C realloc()

If the dynamically allocated memory is insufficient or more than required, you can change the size of previously allocated memory using the realloc() function.

Example of malloc,realloc,free

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
 int *ptr, i , n1, n2;
printf("Enter size: ");
scanf("%d", &n1);
ptr = (int*)
 malloc(n1 * sizeof(int));
 printf("Addresses of previously allocated memory: ");
 for(i = 0; i < n1; ++i)
printf("%u\n",ptr + i);
 printf("\nEnter the new size: ");
 scanf("%d", &n2);
ptr = realloc(ptr, n2 * sizeof(int));
printf("Addresses of newly allocated memory: ");
for(i = 0; i < n2; ++i)
printf("%u\n", ptr + i);
free(ptr);
return 0;
 }
```

Cont...

OUTPUT:

```
Enter size: 2
Addresses of previously allocated memory:26855472 26855476
Enter the new size: 4
Addresses of newly allocated memory:
26855472 26855476 26855480 26855484
```

File Handling in C

C files I/O functions handle data on a secondary storage.

Why files are needed?

- When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.
- If you have to enter a large number of data, it will take a lot of time to enter them all.
  However, if you have a file containing all the data, you can easily access the contents of the file using a few commands in C.
- You can easily move your data from one computer to another without any changes.

Types of Files

When dealing with files, there are two types of files you should know about:

Text files

Binary files

1. Text files

Text files are the normal **.txt** files. You can easily create text files using any simple text editors such as Notepad.

2. Binary files

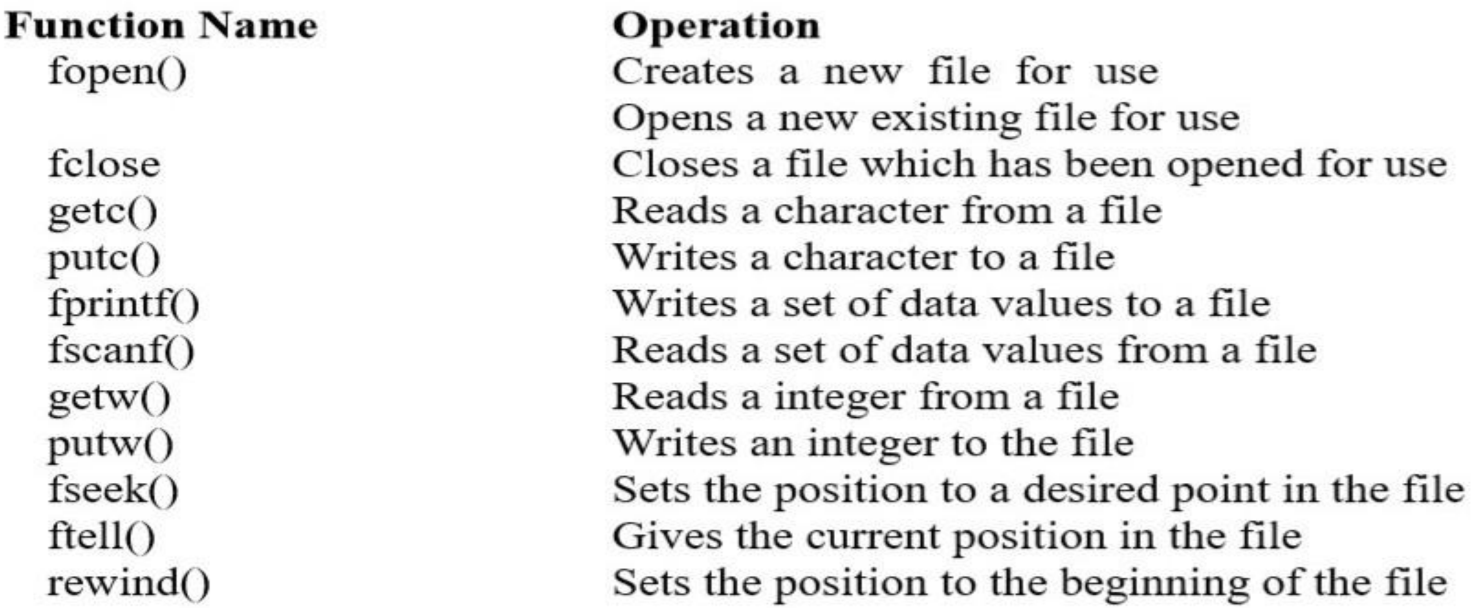
Binary files are mostly the **.bin** files in your computer.

Instead of storing data in plain text, they store it in the binary form (0's and 1's).

File Operations

In C, you can perform four major operations on files:

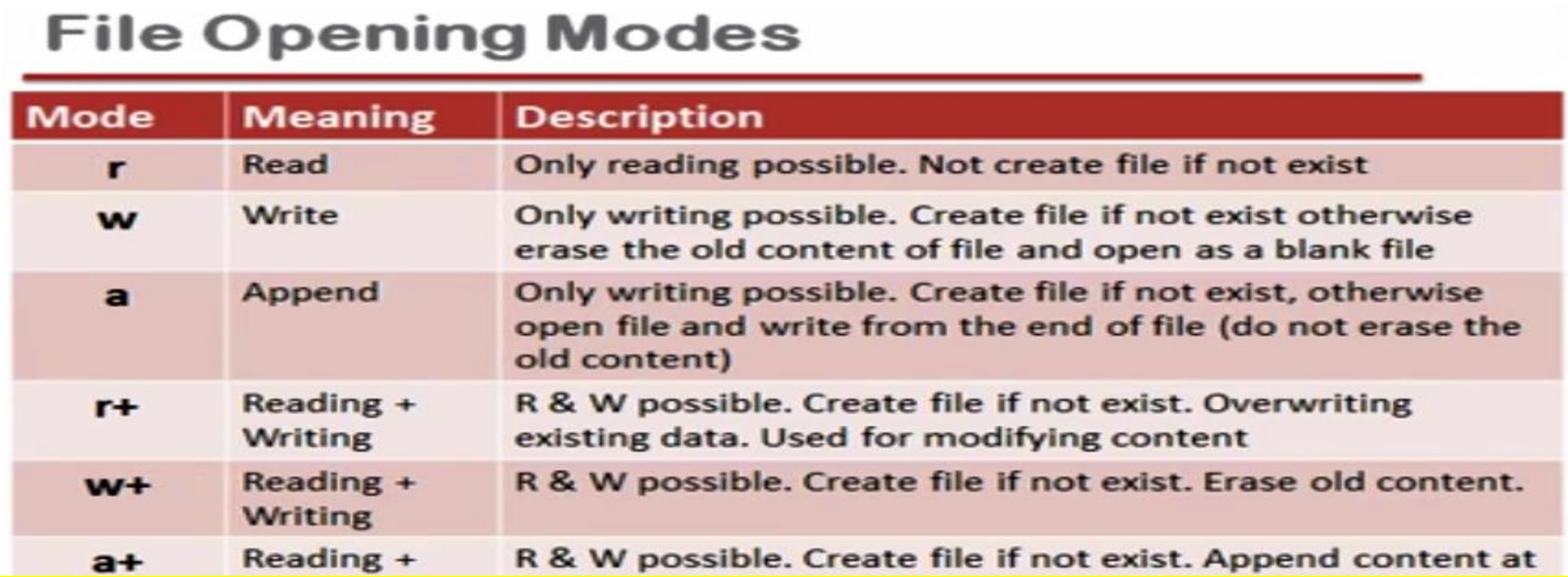
1. Creating a new file
2. Opening an existing file
3. Closing a file
4. Reading from and writing information to a file



Function Name	Operation
fopen()	Creates a new file for use
	Opens a new existing file for use
fclose	Closes a file which has been opened for use
getc()	Reads a character from a file
putc()	Writes a character to a file
fprintf()	Writes a set of data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads a integer from a file
putw()	Writes an integer to the file
fseek()	Sets the position to a desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

Defining and opening a file

- If we want to store data in a file into the secondary memory, we must specify certain things about the file to the operating system. They include the filename, data structure, purpose.
  The general format of the function used for opening a file is
  FILE *fp;
  fp=fopen(“filename”,”**mode**”);
- **filename** is the name of the file to be opened and **mode** specifies the purpose of opening the file. Mode can be of following types,



File Opening Modes

Mode	Meaning	Description
r	Read	Only reading possible. Not create file if not exist
w	Write	Only writing possible. Create file if not exist otherwise erase the old content of file and open as a blank file
a	Append	Only writing possible. Create file if not exist, otherwise open file and write from the end of file (do not erase the old content)
r+	Reading + Writing	R & W possible. Create file if not exist. Overwriting existing data. Used for modifying content
w+	Reading + Writing	R & W possible. Create file if not exist. Erase old content.
a+	Reading +	R & W possible. Create file if not exist. Append content at

Syntax of opens a file in write mode.

Opening a file - for creation and edit

Opening a file is performed using the fopen() function defined in the stdio.h header file.

The syntax for opening a file in standard I/O is:

```
ptr = fopen("fileopen","mode");
```

For example,

```
fopen("E:\\cprogram\\newprogram.txt","w");
fopen("E:\\cprogram\\oldprogram.bin","rb");
```

Example which opens a file in write mode.

```
#include<stdio.h>
void main( )
{
FILE *fp ;
char ch ;
fp = fopen("file_handle.c","r") ;
while ( 1 )
{
ch = fgetc ( fp ) ;
if ( ch == EOF )
break ;
printf("%c",ch) ;
}
fclose (fp ) ;
}
```

Output

The content of the file will be printed.

Bibliography

Programming in ANSI C by E.Balagurusamy, McGrawHill

Let us C Yashavant Kanetkar, BPB Publication

https://www.programiz.com/c-programming/c-dynamic-memory-allocation

https://www.javatpoint.com/dynamic-memory-allocation-in-c

https://studyfreevrdotcom.wordpress.com/2018/10/25/file-opening-modes-in-c-language/

JECRC Foundation

JAIPUR ENGINEERING COLLEGE
AND RESEARCH CENTRE

Thank you!