

Chapter 14

FILE HANDLING IN C

Q1. What is a stream? Also discuss its different types.

Ans.

File:

- The data is stored in the form of files.
- A file is a set of related records.

Stream:

- A stream is a logical interface to a file.
- A stream is associated to a file using an open operation.
- A stream is not associated to a file using a close operation.
- Stream refers to flow of data from source to destination.
- The process of entering data from the source is known as reading, fetching, getting or extracting the data.
- The process of producing output data to the destination is known as writing, storing, inserting or putting the data.
- There are two types of streams.
 - o Text Stream
 - o Binary Stream

Text Stream:

- A text stream is a sequence of characters.
- Character translations may occur in a text stream e.g. new line is represented as carriage return.
- There may not be one to one relation between the characters written and those on the external device.

Binary Stream:

- It is a sequence of bytes.
- The number of bytes written or read is always the same as those on the external device. It means there is one to one relation between the bytes written or read and those on the external device.
- No character translations occur in binary stream.
- Some additional bytes such as the file sector on the disk are added to the binary stream.

Q2. What are file markers? Explain the use of BOF, EOF and New Line.

Ans.

BOF, EOF, New line:

- A file contains a number of characters or bytes and each file stored on a disk has a start and an end.
- The **start** of first byte or character is called Beginning of file (**BOF**).
- The **end** of last byte or character is called End of file (**EOF**).
- The position where data read or data write operations are performed is called current position.
- EOF characters are placed after the last character in C.
- **Enter key** is used in text editor for **new line** but in C new line character is `\n` to be placed at the end of each line.

Q3. How to open a file using c language? Also discuss its different file opening modes.

Ans.

File Opening:

- Before read or write operations, a file must be opened.
- All standard file handling functions of C are declared in `stdio.h`.
- When a file is opened, a portion of memory is reserved for file to read and write this memory is called memory buffer.
- When a file is read, it is first stored in the memory buffer and then program reads the file from the memory buffer.
- When a file is written, then it is written in buffer, then the contents are transferred on the disk.
- The transfer of data from buffer to the disk is called flushing of buffer.
- The writing and reading through file through disk is time consuming. Therefore time of reading and writing data to a file is reduced through buffer.

The File Pointer:

- A pointer is like a variable whose content is the address of another memory cell (variable).
- `*` represents a pointer to the variable with which it is used.
- A file is not directly accessible in C.
- Files are accessed through I/O buffers.

- A file pointer is a variable of FILE type defined in stdio.h
- FILE * represents a pointer to a variable of FILE type
- The fopen() function is used to open a file.
- The function prototype is:

```
FILE *fp;    // it specifies the file pointer.  
fp=fopen(file,mode);
```
- fopen function takes two parameters.
Name of a file: if the file is not in the current directory then its absolute path is given and use escapes \ and \\ in the absolute path.
Opening mode: It is a string and enclosed in double quotes. Mode is a type of operation that is performed on the file

Example:

```
FILE *fp  
fp=fopen("XYZ.DAT","r");
```

Example:

```
FILE *fpr  
fpr=fopen("c:\\tc\\XYZ.DAT","r");
```

File opening Modes:

- "r" open a text file for reading, each file must exists before reading .
- "w" open a text file for writing, if the file already exists then it contents are overwritten, if it does not exists, it will be created.
- "a" open a text file for append. Data is appended at the end of existing file. If the file does not exist, it will be created.
- "r+" opens a file for both reading and writing. The file must already exists
- "w+" opens a file for reading and writing, its contents are overwritten. If the file does not exist, it is created.
- "a+" opens a text file for both reading and appending. If the file does not exist, it is created for both reading and writing.
- The fopen function returns the null pointer, if it fails to open the file (most common reason is file does not exist).

Q4. How to close a file in c language?

Ans.

File Closing:

- An open file must be closed after completing the tasks.
- When a file is closed then the buffer is flushed (data is transferred on the disk).
- If the file is not closed then the data may be lost.

- The `fclose()` function is used to close a file.
- Syntax:
- `fclose(fp);`
- Where `fp` is a file pointer associated to a file which is to be closed.
- `fclose()` function returns 0 if the close operation is successfully completed.
- `fclose()` function returns EOF if any error occurs.

Q5. How to read from and write character to a file using C language?

Ans.

READING AND WRITING CHARACTERS TO A FILE:

Once a file has been opened depending on its opening mode then either a character can be read from a file or a character can be written to a file using the following functions

putc():

- The `putc()` function writes a single character to the file.
- The `putc()` function returns the character, if it successfully writes it.
- The `putc()` function returns the EOF if an error occurs.
- Syntax:
- `putc(character ch , file_pointer);`
- Character `ch` can be character, int, constant or a variable.
- File pointer specifies variable of FILE type associated with the file in which character is to be written.

Example: Design a program to write characters into a text file.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    FILE *fp;
```



```

    char ch;
    clrscr();
    fp = fopen("c:\\test.txt","w");
    printf("Press ESC key to finish : \n");
    while(ch!=27)
    {
        ch=getche();
        putc(ch,fp);
        if(ch=='\r')printf("\n");
    }
    fclose(fp);
}

```

Example: Design a program to append data into an existing file writing character by characters into a text file.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fp;
    char ch, filename[30];
    clrscr();
    printf("\n Enter Name of file to append data ➡ ");
    scanf("%s", &filename);
    fp=fopen(filename,"a");
    if(fp==NULL)printf("\n File Does not exist");
    else printf("\n Type Data press ESC to finish >>>>>\n");
    while(ch!=27)
    {
        ch=getche();
        putc(ch, fp);
        if(ch == '\r')printf("\n");
    }
    fclose(fp);
}

```

getc():

- The getc() function reads the next character from the file.
- If an error occurs it returns EOF.
- The getc() function also returns EOF when the EOF is encountered.
- The file must be in open mode to use this function.
- It automatically moves the next character when the function is again used.

Syntax:

Variable = getc(file_pointer);

Variable is a character variable to which the character read from the file.

Example: Write a program to copy data from one text file into another text file (reading and writing character by character)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *input;
    FILE *output;
    char ch;
    clrscr();
    input = fopen("c:\\test1.txt","r");
    if(input == NULL)    printf("\n can not Open test.txt file\n");
    else
    {
        output = fopen("c:\\test2.txt","w");
        if(output == NULL)
            printf("\n can not Open test2.txt file\n");
        else
            while((ch==getc(input))!=EOF)
                fputc(ch, output);
    }
    printf("\n After Execution data is copied from test1.txt to test2.txt\n ");
    fclose(input);
    fclose(output);
}
```

Q6. How strings are stored and retrieved in c language.

Ans.

String Handling:

- String is a combination of characters and it is enclosed in double quotes.
- Character array is used to hold strings.
- `char name_of_string [string_length];`
- `char` is data type and **string length** is the number of characters in a string.
- A string variable can also be declared without specifying the number of characters.
- The variable that is used to store a string is called **string variable**.
- The last character of a string must be a **Null character \0 i.e. Null character \0** is added to the end of the string.
- Null character \0 is also included in the string length.
- The string can be initialized:
 - `char name[20] = "Lahore";`
 - `char newstr[] = "I Love Pakistan"`
- C uses `string.h` header file to handle the strings.
- The most common string functions are:
 - o `fputs()`
 - o `fgets()`
 - o `strcpy()`

Q7. What is `fputs()` function? Explain with the help of an example.

Ans.

`fputs()` :

- It is used to write a string of characters in to a file.
- This function writes the string until the null character \0 is reached.
- It does not add the null character to the file.
- The file must be opened or in append mode to write this function.

Syntax:

`fputs(string, file_pointer)`

`string` are the characters that are to be written in the file.

```
FILE *fp;
```

```
fp = fopen("test1.txt", "w");
```

```
fputs("The String of your choice :", fp);
```

The above string is written in test1.txt file.

Example: Develop a program that writes three strings lengths of 50 by writing one string at a time in a new text file.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    FILE *fp;
```

```
    char str[50];
```

```
    int r=1;
```

```
    clrscr();
```

```
    fp = fopen("c:\\test1.txt","w");
```

```
    while(r<=3)
```

```
    {
```

```
        gets(str);
```

```
        fputs(str, fp);
```

```
        fputs("\n",fp);
```

```
        r++;
```

```
    }
```

```
    fclose(fp);
```

```
    getch();
```

```
}
```

Example: Develop a program that appends three strings lengths of 50 by writing one string at a time in an existing text file.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main( )
```

```
{
```

```
    FILE *fp;
```

```

char str[50],fname[30];
int r=1;
clrscr();
printf("\n Enter Name of file to append Data >>");
scanf("%s",&fname);
fp = fopen(fname, "a");
if(fp == NULL) printf("\n File Does Not exist ----?");
while(r<=3)
{
    gets(str);
    fputs(str, fp);
    fputs("\n",fp);
    r++;
}
fclose(fp);
getch();
}

```

Q8. What is fgets() function? Explain with the help of an example.

Ans.

fgets() :

- It is used to read a string of characters of a certain length from a file.
- The file must be opened or in append mode to write this function.

Syntax:

fgets(string or string variable, n, file_pointer)

n is the number of characters read from the file.

Example: Program to display the contents of a file. Also compute and display the number of lines of the file.

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main( )
```

```
{
```

```
    FILE *sr;
```

```
    char str[80],fn[12];
```

```
    int c=0;
```

```

clrscr();
printf("\n Enter Name of file to READ DATA >>");
scanf("%s", &fn);
sr = fopen(fn, "r");
if(sr == NULL) printf("\n File Does Not exist ----?");
while(fgets(str,80,sr)!=NULL)
{
    puts(str);
    c++;
}
printf("\n Total Number of Lines =%d", c);
fclose(sr);
getch();
}

```

EXAMPLE: Write a program that name and telephone numbers of your friends and write them in a file.

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
    FILE *fp;
    char name[30],tel[12];
    fp=fopen("c:\\contact.txt","w");
    if(fp == NULL) printf("\n can't open file for writing !\n");
    else
    {
        do
        {
            printf("\n Enter the Name or press Enter to quit ➡ ");
            gets(name);
            if(strlen(name)>0)
            {
                printf("Enter Telephone Number of maximum 10 characters : ");
            }
        }
    }
}

```

```

        gets(tel);
        fputs(name, fp);
        fputs("!", fp);
        fputs(tel, fp);
        fputs("\n", fp);
    }
    while ( strlen (name) > 0);
    fclose(fp);
}
}

```

EXAMPLE: Write a program that will read your contact.txt file and display it on the screen

```

#include<stdio.h>
#include<conio.h>
void main( )
{
    FILE *fp;
    char ch;
    int line =3;
    clrscr( );
    fp=fopen("c:\\contact.txt","r");
    if(fp == NULL) printf("\n can't open file for writing !\n");
    else
    {
        printf("Name");
        gotoxy(35,1);
        printf("Phone #\n");
        printf("-----\n");
        while((ch==getc(fp))!=EOF)
        {
            if(ch=='!')
                gotoxy(35,line);
            else if(ch=='\n')
                gotoxy(1,++line);
            else
                printf("%c ", ch);
        }
    }
}

```

```
        fclose(fp);
        getch( );
    }
}
```

EXAMPLE: Write a program that will append records in your contact.txt file.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
    FILE *fp;
    char name[30], tel[12];
    fp=fopen("c:\\contact.txt","a");
    if(fp == NULL) printf("\n can't open file for writing !\n");
    else
    {
        do
        {
            printf("\n Enter the Name or press Enter to quit ➡ ");
            gets(name);
            if(strlen(name)>0)
            {
                printf("Enter Telephone Number of maximum 10 characters : ");
                gets(tel);
                fputs(name, fp);
                fputs(" !", fp);
                fputs(tel, fp);
                fputs("\n", fp);
            }
        }
        while ( strlen (name) > 0);
    }
}
```



```
fclose(fp);
```

```
}
```

```
}
```

Q9. How to read from and write to formatted data in a file using c language?

Ans.

Formatted I/O:

- The formatted file input and output functions are used for reading and writing formatted data.
- In formatted I/O data is read or data is written in a specific manner in the file.
- The fprintf() function is used for formatted output in a file.
- Its Syntax is:
- fprintf(file_pointer, control_string, arguments);
- The arguments are variables and are optional.
- The fscanf() function is used to read formatted data from a file.
- Its syntax is:
- fscanf(file_pointer, control_string, arguments);

EXAMPLE: Write a program that name and telephone numbers of your friends and write them in a file using formatted I/O.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main( )
{
FILE *fp;
char name[30], tel[12];
fp=fopen("c:\\contact.txt","w");
if(fp == NULL) printf("\n can't open file for writing !\n");
else
{
do
{
printf("\n Enter the Name or press Enter to quit ➡ ");
gets(name);
if(strlen(name)>0)
{
printf("Enter Telephone Number of maximum 10 characters : ");
gets(tel);
fprintf(fp, "%s!%s\n", name, tel);
}
}
}
```

```
while ( strlen (name) > 0);  
fclose(fp);  
}  
}
```