

## File Handling in C

\* A file is used to store large volume of data permanently.

\* A file is of two types:-

- (a) Binary file → All the data is present in the form of 0 and 1
- (b) Text file → data is present in the form of alphabets, digits, special symbols.

### Operations performed on a file :-

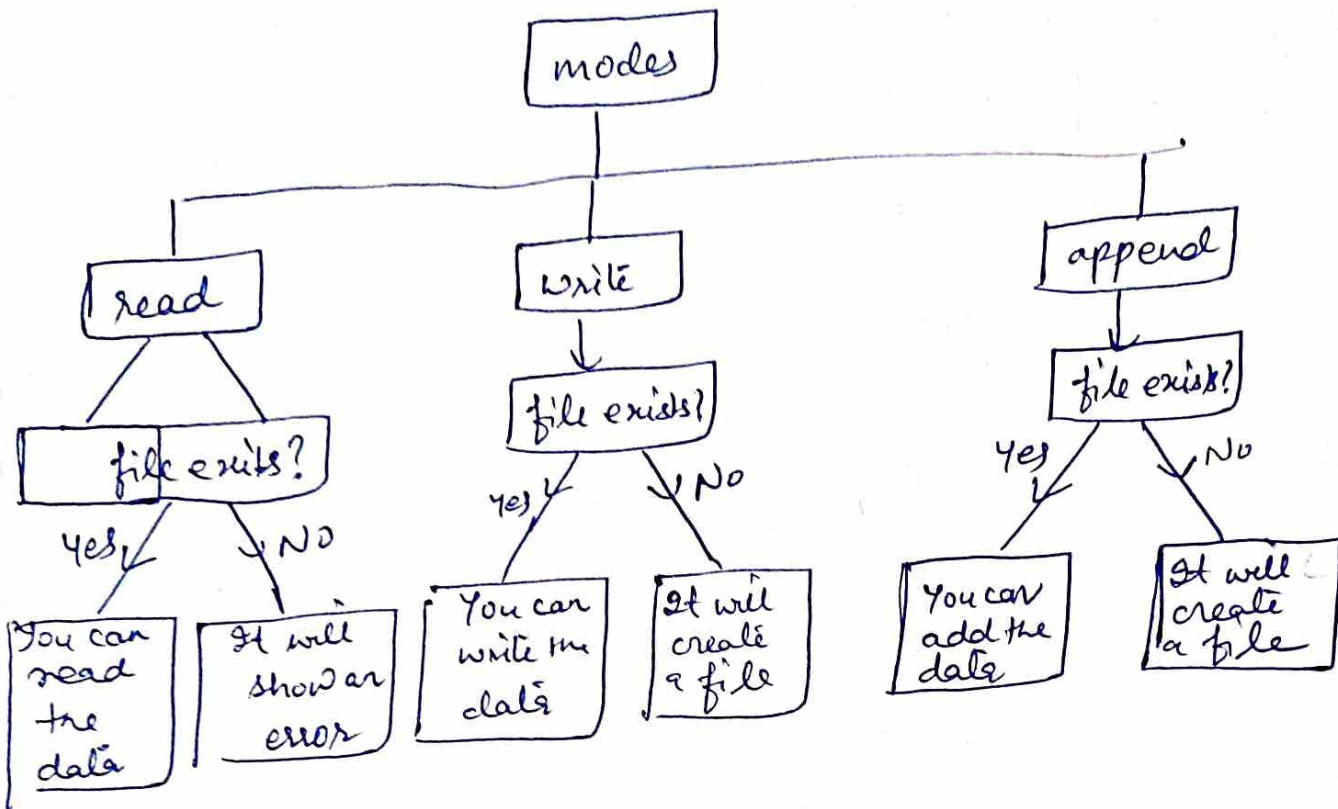
- (i) naming a file
- (ii) opening a file
- (iii) reading from a file
- (iv) writing on a file
- (v) closing a file

### File Handling functions in C:-

- (i) `fopen()` → to open or create a file
- (ii) `fclose()` → to close a file
- (iii) `getc()` → to read a character from a file
- (iv) `putc()` → to write a " in a file
- (v) `getw()` → to read an integer from a file
- (vi) `putw()` → to write an " in a file
- (vii) `fscanf()` → to read set of values from a file
- (viii) `fprintf()` → to write " " " in a file
- (ix) `ferror()` → to check for errors in a file
- (x) `ftell()` → to tell the current position in a file
- (xi) `fseek()` → to set the pointer to desired location in a file

## modes of file:-

- (i) read mode (r)
- (ii) write mode (w)
- (iii) append mode (a)



fopen() → syntax :- `fopen("file-name", "mode");`

(eg) `fopen("file1.c", "r");`

fclose() → syntax :- `fclose(fopen(file-pointer));`

(eg) `fclose(fp);`

How to create/open a file :-

Syntax :-

```
FILE *fp;
fp = fopen("file-name", "mode");
```

(eg)

```
FILE *fp;
fp = fopen("file1.c", "w");
```



## How to read & write a character in a file

putc() → to write a single character.

Syntax:-

putc(ch, file-pointer);

ch → character variable

(eg)

putc(ch, fp);

getc() → to read a single character from a file

syntax:-

ch = getc(file-pointer);

(eg)

ch = getc(fp);

## How to read & write an integer in a file

putw() → to write an integer

syntax →

putw(n, file-pointer);

n → integer variable

(eg)

putw(n, fp);

getw() → to read an integer

syntax -

n = getw(file-pointer);

(eg)

n = getw(fp);

## How to read & write set of values

→ fprintf() → to write a set of values

syntax:-

fprintf(file-pointer, "format specifier", address of variable);

(eg)

fprintf(fp, "%f", &a);

fprintf(fp, "%d.%c.%f", &a, &b, &c);

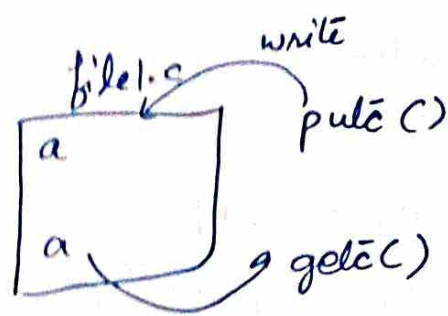
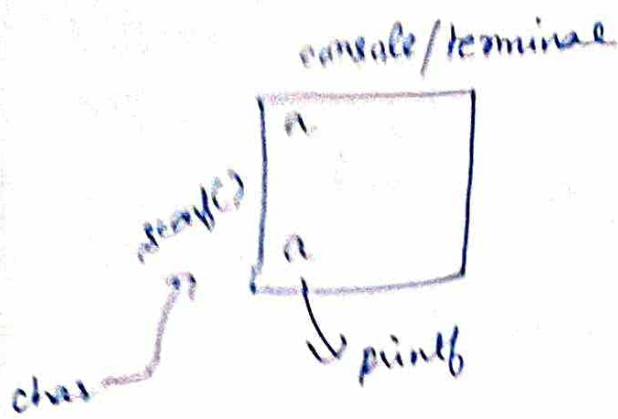
→ fscanf() → to read a set of values

syntax →

fscanf(file-pointer, "format-specifier", name of variable);

(eg)

fscanf(fp, "%f", a);



Q Write a program to write a character on a file and also display this character.

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

```
main()
```

```
{
```

```
FILE *fp;
```

```
fp = fopen("file1.c", "w");
```

```
char ch;
```

```
printf("enter the character");
```

```
scanf("%c", &ch); // input on console
```

```
putc(ch, fp); // write on file
```

```
fclose(fp);
```

```
fp = fopen("file1.c", "r");
```

```
ch = getc(fp); // read from file
```

```
printf("%c", ch); // print on console
```

```
fclose(fp);
```

```
}
```

Q Write a program to write multiple characters on a file & also display them.

Soln

```
#include <stdio.h>
main()
{
    FILE *fp;
    fp = fopen("file1.c", "w");
    char ch;
    printf("enter the character");
    scanf("%c", &ch);
    while (ch != EOF)
    {
        putc(ch, fp);
        scanf("%c", &ch);
    }
    fclose(fp);
    fp = fopen("file1.c", "r");
    ch = getc(fp);
    while (ch != EOF)
    {
        printf("%c", ch);
        ch = getc(fp);
    }
    fp = fopen
    fclose(fp);
}
```

EOF → End of file  
It indicates the end of file.

⇒ press ctrl + Z

console  
apple

file1  
apple

Q ~~\*\*\*~~ 2

Write a C program to copy the contents of a file into another file.

read — file1.  
Hello

file2 — write



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

```
main()
```

```
{
```

```
FILE *fp1, *fp2;
```

```
fp1 = fopen("file1.c", "r");
```

```
fp2 = fopen("file2.c", "w");
```

```
char ch;
```

```
ch = getc(fp1); // read from file 1
```

```
while (ch != EOF)
```

```
{
```

```
putc(ch, fp2); // write in for file 2
```

```
ch = getc(fp1);
```

```
}
```

```
fclose(fp1);
```

```
fclose(fp2);
```

```
}
```

Q WAP to count no. of characters in a file.

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

```
main()
```

```
{
```

```
FILE *fp;
```

```
fp = fopen("file1.c", "r");
```

```
char ch;
```

```
int c = 0;
```

```
ch = getc(fp);
```

```
while (ch != EOF)
```

```
{
```

```
c = c + 1;
```

```
ch = getc(fp);
```

```
}
```

```
printf("no. of characters = %d", c);
```

```
fclose(fp);
```

```
}
```

Q  
2  
★★

WAP to check whether two files are identical or not.

OR

WAP to check whether contents of two files are equal or not.

Soln<sup>n</sup>

```
#include <stdio.h>
main()
```

```
{
```

```
FILE *fp1, *fp2;
```

```
fp1 = fopen("file1.c", "r");
```

```
fp2 = fopen("file2.c", "r");
```

```
char ch1, ch2;
```

```
int c = 0;
```

```
ch1 = getc(fp1);
```

```
ch2 = getc(fp2);
```

```
while (ch1 != EOF & ch2 != EOF)
```

```
{
```

```
if (ch1 != ch2)
```

```
{ c = 1;
```

```
break;
```

```
}
```

```
ch1 = getc(fp1);
```

```
ch2 = getc(fp2);
```

```
}
```

```
if (c == 0)
```

```
{ printf("Identical");
```

```
}
```

```
else
```

```
{ printf("not");
```

```
}
```

```
fclose(fp1);
```

```
fclose(fp2);
```

```
}
```

Q \*\*\* WAP to write 10 integers in a file named as number.c. Out of these 10 integers, write all the even numbers in a file named EVEN.c and all the odd numbers in a file named ODD.c

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

```
main()
```

```
{ FILE *fp1, *fp2, *fp3;
  fp1 = fopen("number.c", "w");
```

```
  int n;
```

```
  printf("enter the integer");
```

```
  scanf("%d", &n);
```

```
  while (n != EOF)
```

```
  {
```

```
    putw(n, fp1);
```

```
    scanf("%d", &n);
```

```
  }
```

```
  fclose(fp1);
```

```
  fp1 = fopen("number.c", "r");
```

```
  fp2 = fopen("even.c", "w");
```

```
  fp3 = fopen("odd.c", "w");
```

```
  n = getw(fp1);
```

```
  while (n != EOF)
```

```
  { if (n % 2 == 0)
```

```
    {
```

```
      putw(n, fp2);
```

```
    }
```

```
  else
```

```
    { putw(n, fp3);
```

```
    }
```

```
  } n = getw(fp1);
```

```
  fclose(fp1);
```

```
  fclose(fp2);
```

```
  fclose(fp3);
```

console, Number.c

123
456
78910

even.c
246
810

odd.c
1357
9