File

```
main( )
  {
     FILE  *fp ;
     fp = fopen ( "PR1.C", "r" ) ;
     .....
  }
```

- **fopen( ) performs three important tasks when you open the file in “r” mode:**

1. Firstly it searches on the disk the file to be opened.
2. Then it loads the file from the disk into a place in memory called buffer.
3. It sets up a character pointer that points to the first character of the buffer.

- **Different modes:**

"r" -- Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the first character in it. If the file cannot be opened fopen( ) returns NULL. Operations possible – reading from the file.

"w“-- Searches file. If the file exists, its contents are overwritten. If the file doesn’t exist, a new file is created. Returns NULL, if unable to open file.

Operations possible – writing to the file.

"a" Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.
Operations possible - adding new contents at the end of file.

- Here FILE is a structure. fopen( ) returns the address of this structure, which we have collected in the structure pointer called fp.

FILE READ(Display the contents of a file)

```
void main()
{
 FILE *fp1;
 char ch;
 clrscr();
fp1=fopen(“msit.txt","r");
 if(fp1==NULL)
   {
        printf("\nUnable to
        open file ");
        exit(0);
   }
ch=fgetc(fp1);
 while(ch!=EOF)
   {
        printf("%c",ch);
        ch=fgetc(fp1);
   }
 fclose(fp1);
 getch();
 }
```

FILE COPY

```
void main()
{
 FILE *fp1,*fp2;
 char ch,file1[30],file2[30];
 printf("\nEnter the sourse file name: ");
 gets(file1);
 printf("\nEnter the destination file name: ");
 gets(file2);
 fp1=fopen(file1,"r");
 if(fp1==NULL)
 {
   printf("\nUnable to open file %s",file1);
   exit(0);
 }
 fp2=fopen(file2,"w");
  if(fp2==NULL)
  {
    printf("\nUnable to open file %s",file2);
    exit(0);
  }
  ch=fgetc(fp1);
  while(ch!=EOF)
  {
       fputc(ch,fp2);
       ch=fgetc(fp1);
  }
  fclose(fp1);
  fclose(fp2);
  getch();
 }
```

FILE COPY USING COMMAND LINE ARGUMENTS

```
void main(int argc, char *argv[])
{
FILE *fp1,*fp2;
 char ch;
 if(argc < 3)
    {
      printf(“enter source and destination
       file name properly”);
      exit(0);
     }
 fp1=fopen(argv[1],"r");
 if(fp1==NULL)
 {
   printf("\nUnable to open file %s",argv[1]);
   exit(0);
 }
```

```
 fp2=fopen(argv[2],"w");
  if(fp2==NULL)
  {
    printf("Unable to open file %s",argv[2]);
    exit(0);
  }
  ch=fgetc(fp1);
  while(ch!=EOF)
  {
        fputc(ch,fp2);
        ch=fgetc(fp1);
  }
  fclose(fp1);
  fclose(fp2);
  getch();
 }
```

FILE REVERSE

```
ch=fgetc(fp1);
while(ch!=EOF)
   {
            if(ch!='\n')
           {
             str[i++]=ch;
            }
           else
            {
           for(j=i-1;j>=0;j--)
                        fputc(str[j],fp2);
           fputc('\n',fp2);
           i=0;
            }
   ch=fgetc(fp1);
}
           for(j=i-1;j>=0;j--)
                    fputc(str[j],fp2);
           fclose(fp1);
           fclose(fp2);
           getch();
}
```