2.Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)

a) **Open and write**

```
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdio.h>
void main()
{
  int fd;
  fd=creat("n1",0666);
  if(fd!=-1)
  {
        fd=open("n1",O_WRONLY);
        if(fd==-1)
        {
              printf("Cannot open file");
        }
        write(fd,"This is OS Program",20);
        close(fd);
  }
}
```

OUTPUT:

This is OS Program

b) Open and read

```
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdio.h>
void main()
{
  int fd,n=1;
  char buf;
  fd=open("opening.c",O_RDONLY);
  if(fd==-1)
  {
        printf("File doesnot exist");
  }
  while(n>0)
  {
        n=read(fd,&buf,1);
        printf("%c",buf);
  }
}
```

OUTPUT:

c) Opendir and readdir

```
#include<stdio.h>
#include<dirent.h>
void main(int argc,char **argv)
{
  DIR *dp;
  struct dirent *link;
  dp=opendir(argv[1]);
  printf("Contents of the directory %s are \n",argv[1]);
  while((link=readdir(dp))!=0)
  {
            printf("%s\n",link->d_name);
  }
  closedir(dp);
}
```

Output:

OUTPUT:

d) Stat

```
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdio.h>
void main()
{
    struct stat fs;
    if(stat("opening.c",&fs)==-1)
    {
            printf("Error in stat system call");
    }
    printf("No of links: %ld\n",fs.st_nlink);
    printf("Filesize: %ld bytes\n",fs.st_size);
    printf("File in ode: %ld\n",fs.st_ino);
}
```

Output:

e) seek_set

```
#include<fcntl.h>
#include<unistd.h>
#include<sys/stat.h>
#include<stdio.h>
void main()
{
        char buf;
        int fd=open("opening.c",O_RDONLY);
        lseek(fd,5,SEEK_SET);
        read(fd,&buf,1);
        printf("The char at offset 5=%c\n",buf);
        close(fd);
}
```

Output:

f) seek_cur

```
#include<sys/types.h>
#include<fcntl.h>
#include<unistd.h>
#include<sys/stat.h>
#include<stdio.h>
void main()
{
    char buf;
    int fd=open("opening.c",O_RDONLY);
    lseek(fd, -5,SEEK_CUR);
    read(fd,&buf,1);
    printf("The char at offset 5=%c\n",buf);
```

```
    close(fd);
}
```
Output: