

Practical 6

Directory handling- II

Program 1

//This program demonstrate the working of lstat() function

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

```
#include<unistd.h>
```

```
#include<sys/stat.h>
```

```
#include<sys/types.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int n;
```

```
    struct stat buf;
```

```
    if ((n=lstat(argv[1],&buf)) == -1)
```

```
    {
```

```
        exit(1);
```

```
    }
```

```
    printf("File = %s\n",argv[1]);
```

```
    printf("i-node number = %d\n",buf.st_ino);
```

```
    printf("uid number = %d\n",buf.st_uid);
```

```
    printf("gid number = %d\n",buf.st_gid);
```

```
    printf("type and permission = %o\n",buf.st_mode);
```

```
    printf("number of links = %d\n",buf.st_nlink);
```

```
    printf("block size = %d\n",buf.st_blksize);
```

```
    printf("number of blocks = %d\n",buf.st_blocks);
```

```
    printf("file size = %d\n",buf.st_size);
```

```
    if (S_ISREG(buf.st_mode))
```

```
        printf("regular file \n");
```

```
    if (S_ISDIR(buf.st_mode))
```

```
        printf("directory file \n");
```

```
    if (S_ISCHR(buf.st_mode))
```

```
        printf("character file \n");
```

```

    if (S_ISBLK(buf.st_mode))
        printf("block file \n");
    if (S_ISLNK(buf.st_mode))
        printf("symbolic file \n");
    if (S_ISFIFO(buf.st_mode))
        printf("FIFO file \n");
    if (S_ISSOCK(buf.st_mode))
        printf("socket file \n");
    exit(0);
}

```

Program 2

//This program takes directory name as argument and list all subdirectories

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

```
#include<dirent.h>
```

```
#include<sys/stat.h>
```

```
#include<sys/types.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    DIR *dir;
```

```
    struct stat statbuf;
```

```
    struct dirent *dentry;
```

```
    mode_t f_type, f_perm;
```

```
    dir=opendir(argv[1]);
```

```
    if(dir==NULL)
```

```
    {
```

```
        printf("Cannot open directory...\n");
```

```
        exit(1);
```

```
    }
```

```
    chdir(argv[1]);
```

// chdir call is essential because the d_name member of the dirent

//structure evaluates to a filename without slashes. Without chdir lstat would

```

//look for the //file in the current directory
while((dentry=readdir(dir))!=NULL)
{
    lstat(dentry->d_name,&statbuf);
    if(S_ISDIR(statbuf.st_mode))
    {
        f_type=statbuf.st_mode & S_IFMT;
        f_perm=statbuf.st_mode & ~S_IFMT;
        printf("%o %o %s\n",f_type,f_perm,dentry->d_name);
    }
}
closedir(dir);
}

```

Program 3

// This program accepts path and filename as arguments and check availability of file.If exists as file then //delete and create directory with same name. If directory then display appropriate message

```

#include<stdio.h>
#include<dirent.h>
#include<sys/stat.h>

int main(int argc, char *argv[])
{
    struct dirent *dentry;
    struct stat statbuf;
    DIR *dir;
    int flag=0;
    if((dir=opendir(argv[1]))==NULL)
    {
        printf("Cannot open directory...\n");
        exit(1);
    }
    chdir(argv[1]);
}

```

```

while((dentry=readdir(dir))!=NULL)
{
    if((strcmp(dentry->d_name,argv[2])==0))
    {
        flag=1; //File found

        lstat(dentry->d_name,&statbuf);
        if(S_ISREG(statbuf.st_mode))
        {
            unlink(argv[2]);
            mkdir(argv[2],0777);
            printf("Directory created...\n");
        }
        else
        if(S_ISDIR(statbuf.st_mode))
            printf("%s is directory\n",dentry->d_name);
    }

}

if(flag==0)
    printf("%s file not found\n",argv[2]);
closedir(dir);
}

```