**OPERATING SYSTEM LAB 5**

**Q1.**

PROGRAM:

#include <sys/types.h>

#include <sys/stat.h>

#include <stdio.h>

#include <unistd.h>

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

{

struct stat sb;

int ret;

if(argc == 2)

{

ret = stat (argv[1], &sb);

if (ret)

{

perror("stat");

return 1;

}

printf("Inode number: %ld\n",sb.st\_ino);

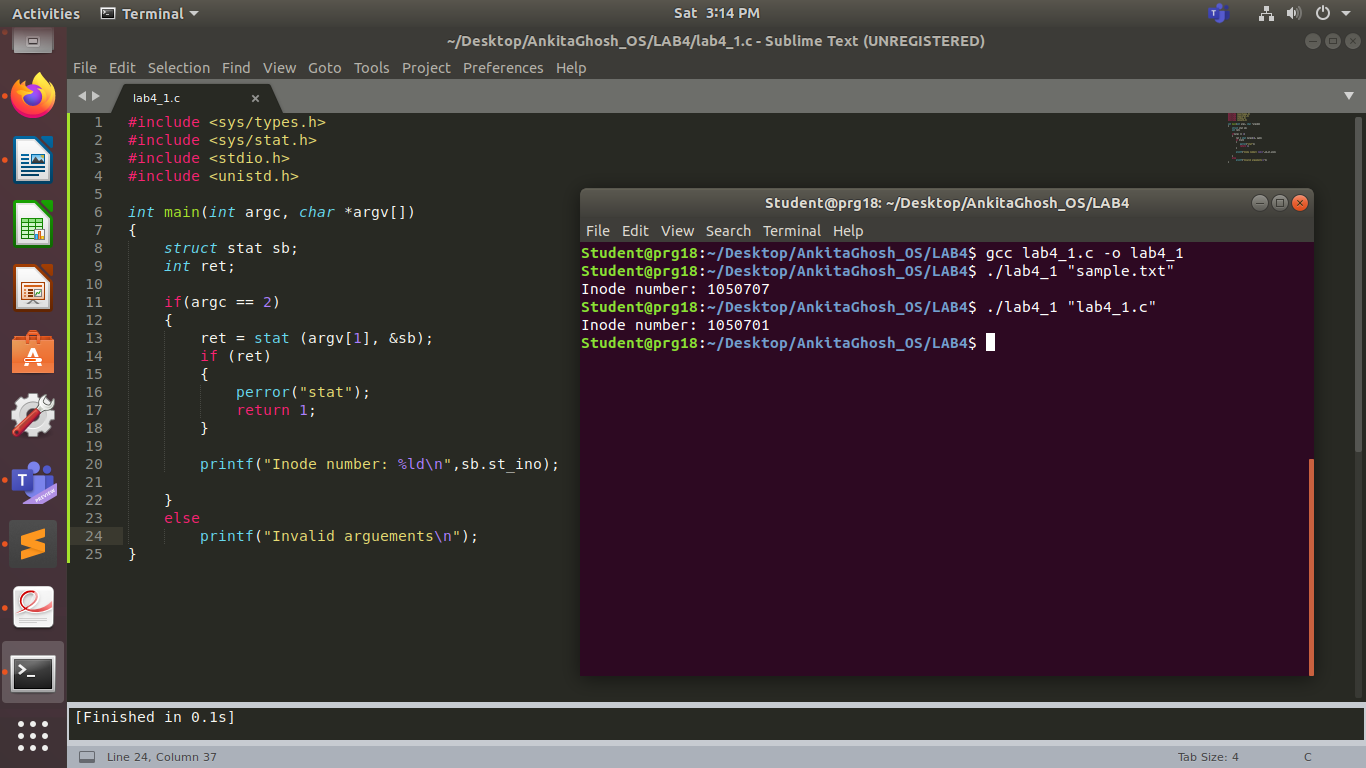
}

else

printf("Invalid arguements\n");

}

OUTPUT:



**Q2.**

PROGRAM:

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/types.h>

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

{

struct stat sb;

int ret;

ret = stat(argv[1], &sb);

if(argc==2)

{

if(ret)

{

perror("stat");

return 1;

}

else

{

printf("ID of device: %ld\n",sb.st\_dev);

printf("Inode number: %ld\n",sb.st\_ino);

printf("Permissions: %d\n",sb.st\_mode);

printf("No. of hard links: %ld\n",sb.st\_nlink);

printf("User ID: %d\n",sb.st\_uid);

printf("Group ID: %d\n",sb.st\_gid);

printf("Device ID (special file): %ld\n",sb.st\_rdev);

printf("Total size: %ld bytes\n",sb.st\_size);

printf("Block size: %ld\n",sb.st\_blksize);

printf("No. of blocks: %ld\n",sb.st\_blocks);

printf("Last access time: %ld\n",sb.st\_atime);

printf("Last modification time: %ld\n",sb.st\_mtime);

printf("Last status change time: %ld\n",sb.st\_ctime);

}

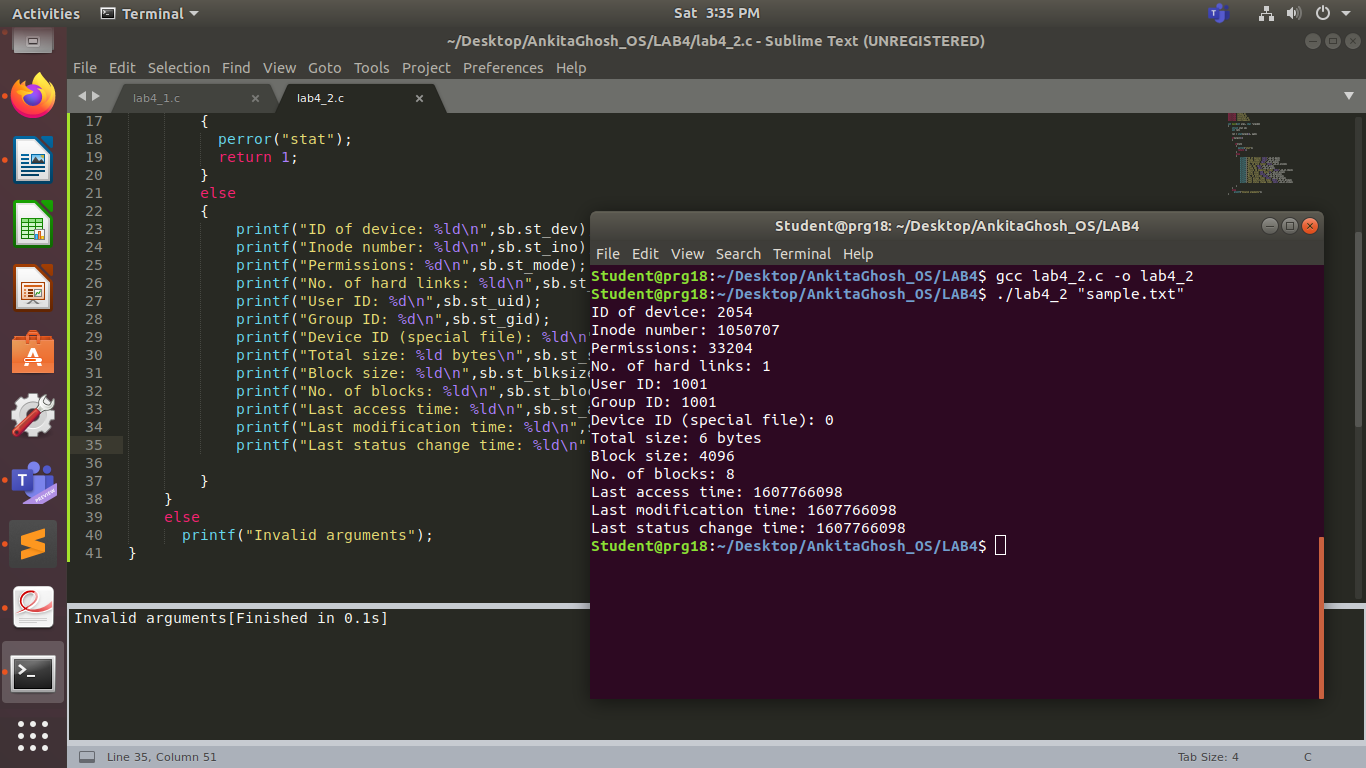
}

else

printf("Invalid arguments");

}

OUTPUT:



**Q3.**

PROGRAM:

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/types.h>

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

{

struct stat sb; int ln; int uln;

int ret;

ret = stat(argv[1], &sb);

if(argc==3)

{

if(ret)

{

perror("stat");

return 1;

}

else

{

printf("Creating new hard link\n");

ln=link(argv[1],argv[2]);

if(ln)

{

perror("link");

return 1;

}

printf("Unlinking\n");

uln=unlink(argv[2]);

if(uln)

{

perror("link");

return 1;

}

}

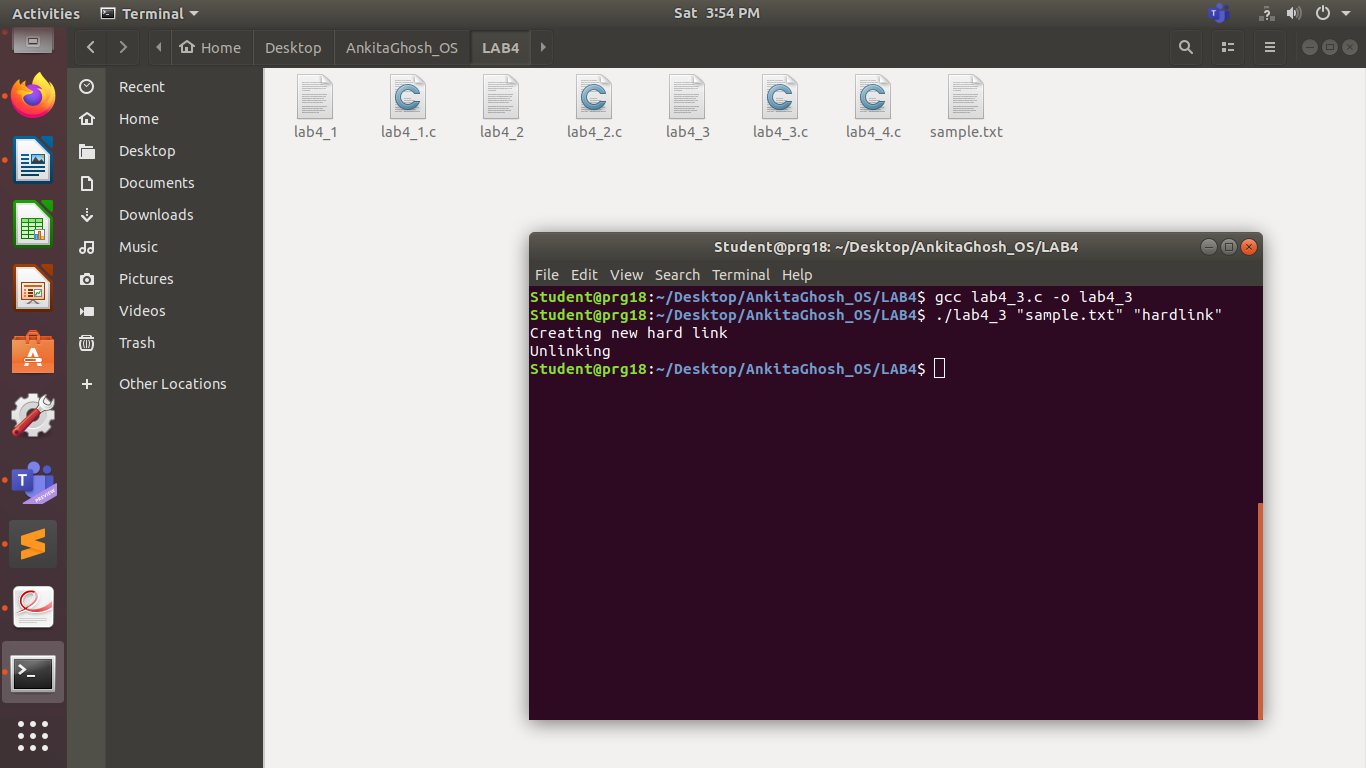
}

else

printf("Invalid arguments");

}

OUTPUT:



**Q4.**

PROGRAM:

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/types.h>

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

{

struct stat sb;

int ret; int ln; int uln;

ret = stat(argv[1], &sb);

if(argc==3)

{

if(ret)

{

perror("stat");

return 1;

}

else

{

printf("Creating new soft link\n");

ln=symlink(argv[1],argv[2]);

if(ln)

{

perror("link");

return 1;

}

printf("Unlinking\n");

uln=unlink(argv[1]);

if(uln)

{

perror("link");

return 1;

}

}

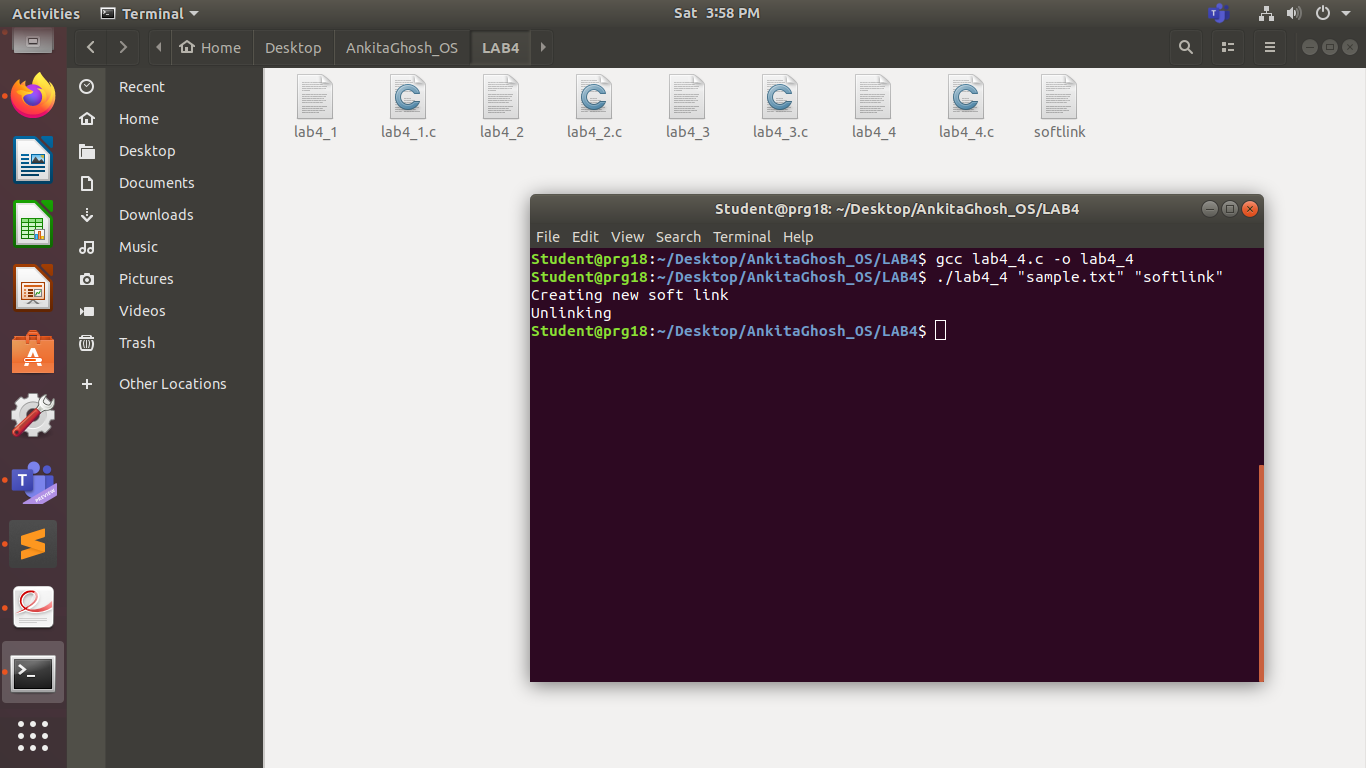
}

else

printf("Invalid arguments");

}

OUTPUT:



**Q5. (Additional Q1)**

PROGRAM:

#include <unistd.h>

#include <stdio.h>

#include <dirent.h>

#include <string.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <sys/types.h>

void printdir(char \*dir, int depth) {

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

if((dp = opendir(dir)) == NULL) {

fprintf(stderr, "cannot open dir: %s\n", dir);

return;

}

chdir(dir);

while((entry = readdir(dp)) != NULL) {

lstat(entry->d\_name, &statbuf);

if(!S\_ISDIR(statbuf.st\_mode))

{

printf("%s\t%ld\n",entry->d\_name,statbuf.st\_ino);

}

}

chdir("..");

closedir(dp);

}

int main(int argc, char\* argv[]){

if(argc != 2) {

printf("Invalid args. Only enter required directory.");

}

printdir(argv[1], 0);

return 0;

}

OUTPUT:

