Name : Rajvardhan Reddy
Reg No : 180905093
Sec : B
Roll No : 19

OS LAB – 4 : FILE SYSTEMS

Lab Exercises :

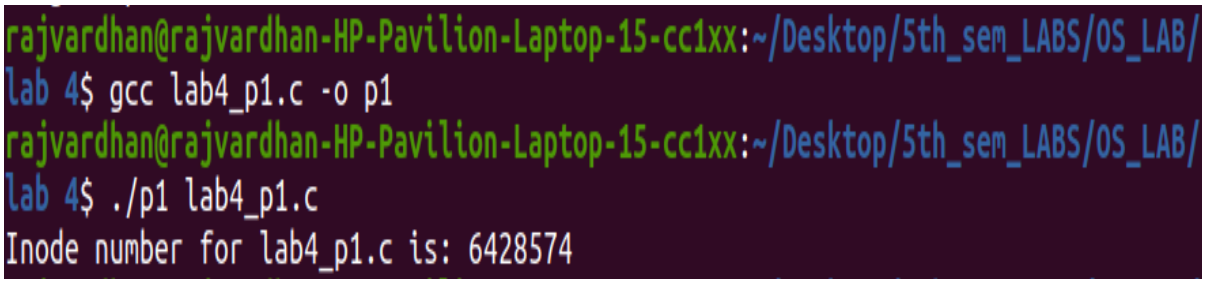
P1) Write a program to find the inode number of an existing file in a directory. Take the input as a filename and print the inode number of the file.

Program :

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
struct stat sb;
int ret;
if (argc < 2)
{
fprintf(stderr, "usage: %s <file>\n", argv[0]);
return 1;
}
ret = stat(argv[1], &sb);
if (ret)
{
perror("stat");
return 1;
}
printf("Inode number for %s is: %ld \n", argv[1],sb.st_ino);
return 0;
}
```

Output :



P2) Write a program to print out the complete stat structure of a file.

Program :

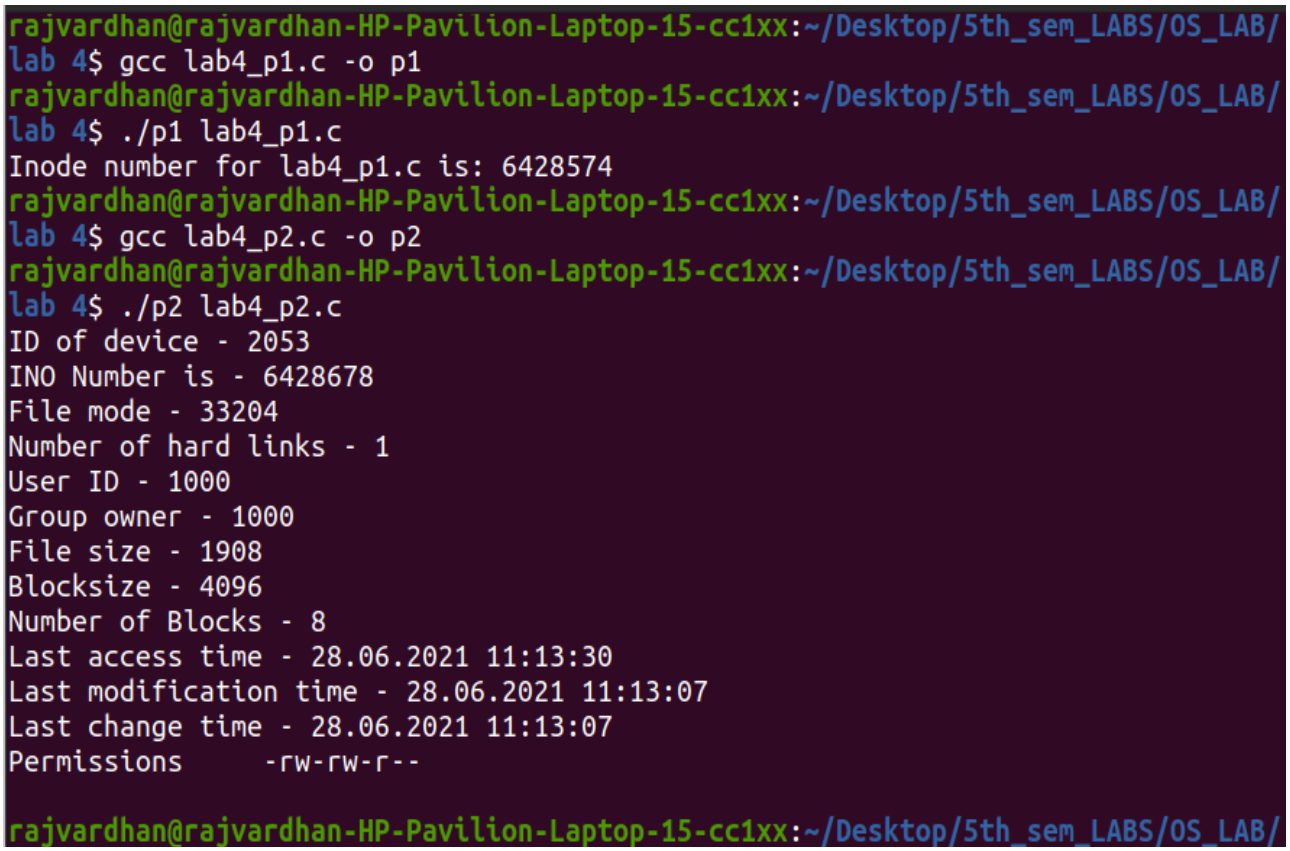
```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>

char *formatdate(char *str, time_t val)
{
strftime(str, 36, "%d.%m.%Y %H:%M:%S", localtime(&val));
return str;
}
int main(int argc, char *argv[])
{
struct stat sb;
if (argc < 2)
{
printf("Insufficient arguments!\n");
return 1;
}
int ret;
char date[36];
ret = stat(argv[1], &sb);
if (ret)
{
perror("stat");
```

```
return 1;
}
printf("ID of device - %lu\n", sb.st_dev);
printf("INO Number is - %lu\n", sb.st_ino);
printf("File mode - %d\n", sb.st_mode);
printf("Number of hard links - %ld\n", sb.st_nlink);
printf("User ID - %d\n", sb.st_uid);
printf("Group owner - %d\n", sb.st_gid);
printf("File size - %ld\n", sb.st_size);
printf("Blocksize - %ld\n", sb.st_blksize);
printf("Number of Blocks - %ld\n", sb.st_blocks);
printf("Last access time - %s\n", formatdate(date, sb.st_atime));
printf("Last modification time - %s\n",formatdate(date,
sb.st_mtime));
printf("Last change time - %s\n", formatdate(date, sb.st_ctime));
DIR *dp;
struct dirent *entry;
struct stat statbuf;
if ((dp = opendir(".")) == NULL)
{
printf("Cannot open directory \n");
return 0;
}
chdir(".");
while ((entry = readdir(dp)) != NULL)
{
lstat(entry->d_name, &statbuf);
if (!S_ISDIR(statbuf.st_mode))
{
if (strcmp(entry->d_name, argv[1]) == 0)
{
printf("Permissions\t");
printf((S_ISDIR(statbuf.st_mode)) ? "d" : "-");
printf((statbuf.st_mode & S_IRUSR) ? "r" : "-");
printf((statbuf.st_mode & S_IWUSR) ? "w" : "-");
printf((statbuf.st_mode & S_IXUSR) ? "x" : "-");
printf((statbuf.st_mode & S_IRGRP) ? "r" : "-");
printf((statbuf.st_mode & S_IWGRP) ? "w" : "-");
printf((statbuf.st_mode & S_IXGRP) ? "x" : "-");
printf((statbuf.st_mode & S_IROTH) ? "r" : "-");
printf((statbuf.st_mode & S_IWOTH) ? "w" : "-");
printf((statbuf.st_mode & S_IXOTH) ? "x" : "-");
printf("\n\n");
}
```

```
}
}}
```

Output :



P3) Write a program to create a new hard link to an existing file and unlink the same. Accept the old path as input and print the newpath.

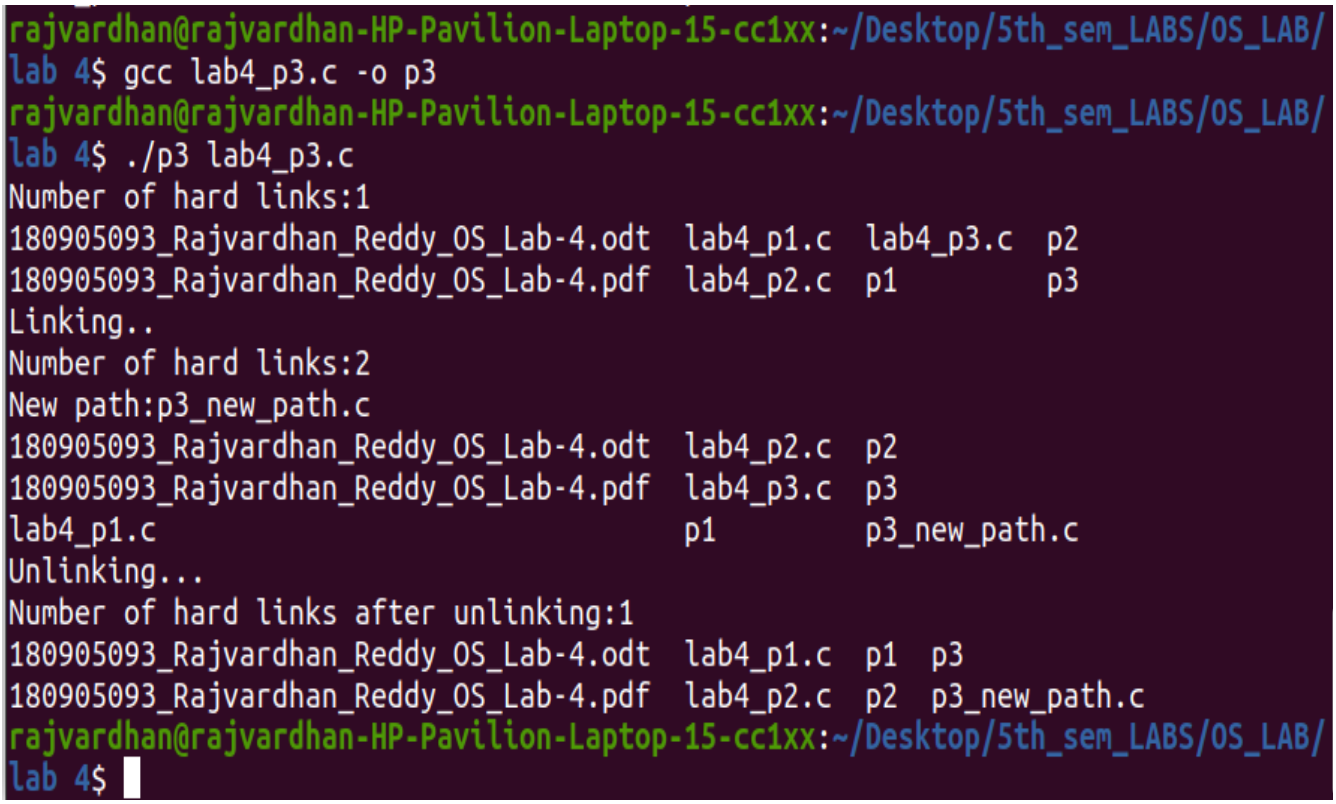
Program :

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <inttypes.h>
#include <stdlib.h>

void main(int argc, char *argv[])
{
if (argc < 2)
{
printf("Insufficient arguments\n");
```

```
return;
}
char new_path[100] = "p3_new_path.c";
struct stat start;
int ret1 = stat(argv[1], &start);
printf("Number of hard links:%ld\n", start.st_nlink);
system("ls");
printf("Linking..\n");
int ret2 = link(argv[1], new_path);
struct stat intermediate;
int ret3 = stat(argv[1], &intermediate);
printf("Number of hard links:%ld\n", intermediate.st_nlink);
printf("New path:%s\n", new_path);
system("ls");
int ret4 = unlink(argv[1]);
struct stat ending;
int ret5 = stat(new_path, &ending);
printf("Unlinking...\n");
printf("Number of hard links after unlinking:%ld\n", ending.st_nlink);
system("ls");
}
```

Output :



P4) Write a program to create a new soft link to an existing file and unlink the same. Accept the old path as input and print the newpath.

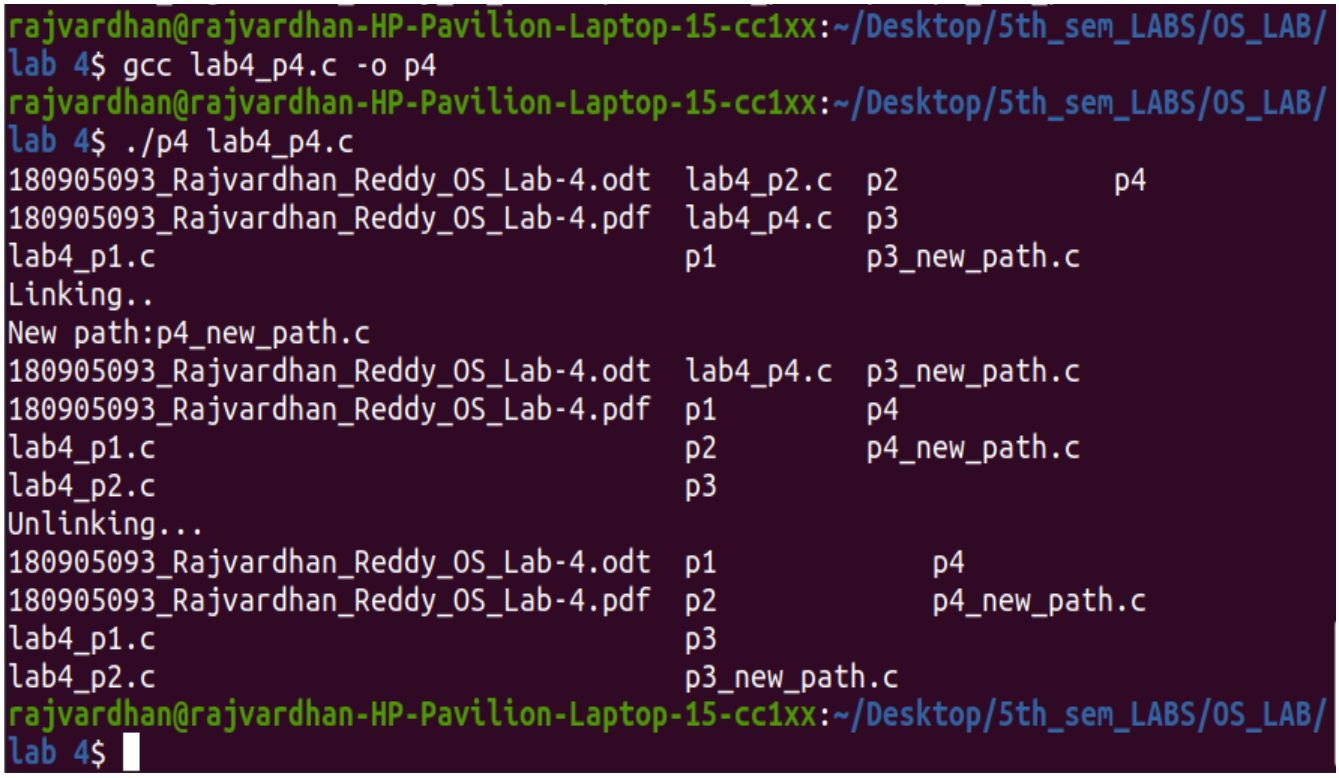
Program :

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <inttypes.h>
#include <stdlib.h>

void main(int argc, char *argv[])
{
if (argc < 2)
{
printf("Insufficient arguments\n");
return;
}

char new_path[100] = "p4_new_path.c";
struct stat start;
int ret1 = stat(argv[1], &start);
system("ls");
printf("Linking..\n");
int ret2 = symlink(argv[1], new_path);
struct stat intermediate;
int ret3 = stat(argv[1], &intermediate);
printf("New path:%s\n", new_path);
system("ls");
int ret4 = unlink(argv[1]);
struct stat ending;
int ret5 = stat(new_path, &ending);
printf("Unlinking...\n");
system("ls");
}
```

Output :


rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab 4$ gcc lab4_p4.c -o p4
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab 4$ ./p4 lab4_p4.c
180905093_Rajvardhan_Reddy_OS_Lab-4.odt lab4_p2.c p2 p4
180905093_Rajvardhan_Reddy_OS_Lab-4.pdf lab4_p4.c p3
lab4_p1.c p1 p3_new_path.c
Linking..
New path:p4_new_path.c
180905093_Rajvardhan_Reddy_OS_Lab-4.odt lab4_p4.c p3_new_path.c
180905093_Rajvardhan_Reddy_OS_Lab-4.pdf p1 p4
lab4_p1.c p2 p4_new_path.c
lab4_p2.c p3
Unlinking...
180905093_Rajvardhan_Reddy_OS_Lab-4.odt p1 p4
180905093_Rajvardhan_Reddy_OS_Lab-4.pdf p2 p4_new_path.c
lab4_p1.c p3
lab4_p2.c p3_new_path.c
rajvardhan@rajvardhan-HP-Pavilion-Laptop-15-cc1xx:~/Desktop/5th_sem_LABS/OS_LAB/lab 4$