

Operating Systems Lab 4: File System

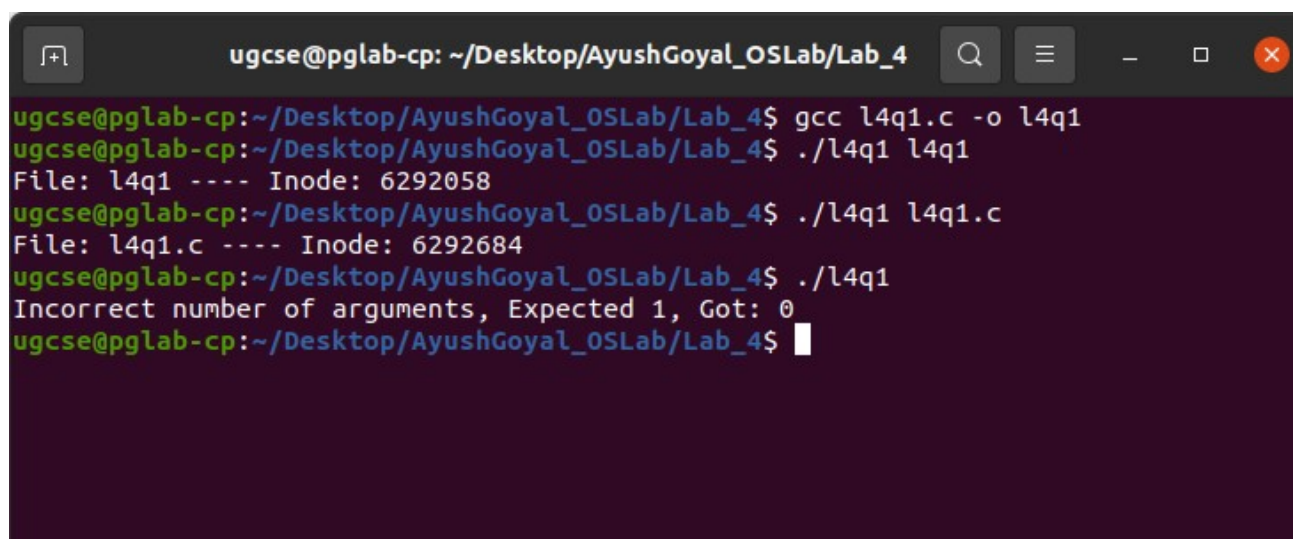
1. Write a program to find the inode number of an existing file in a directory. Take the input as a filename and print the inode number of the file.

Code:

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

int main(int argc, char const *argv[]){
    if(argc < 2){
        printf("Incorrect number of arguments, Expected 1, Got: %d\n",argc-1);
        return 0;
    }
    struct stat file_stat;
    int return_value = stat(argv[1], &file_stat);
    if(return_value != 0){
        perror("stat");
        return 1;
    }
    printf("File: %s ---- Inode: %ld\n", argv[1], file_stat.st_ino);
    return 0;
}
```

Output:

A terminal window with a dark background and light-colored text. The window title is "ugcse@pglab-cp: ~/Desktop/AyushGoyal_OSLab/Lab_4". The terminal shows the following commands and output:

```
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ gcc l4q1.c -o l4q1
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q1 l4q1
File: l4q1 ---- Inode: 6292058
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q1 l4q1.c
File: l4q1.c ---- Inode: 6292684
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q1
Incorrect number of arguments, Expected 1, Got: 0
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$
```

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

Code:

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

const char *perms(__mode_t mode){
    static char buf[16] = {0};
    int i = 0;
    //user permissions
    buf[i++] = ((mode & S_IRUSR) ? 'r' : '-');
    buf[i++] = ((mode & S_IWUSR) ? 'w' : '-');
    buf[i++] = ((mode & S_IXUSR) ? 'x' : '-');
    //group permissions
    buf[i++] = ((mode & S_IRGRP) ? 'r' : '-');
    buf[i++] = ((mode & S_IWGRP) ? 'w' : '-');
    buf[i++] = ((mode & S_IXGRP) ? 'x' : '-');
    //other permissions
    buf[i++] = ((mode & S_IROTH) ? 'r' : '-');
    buf[i++] = ((mode & S_IWOTH) ? 'w' : '-');
    buf[i++] = ((mode & S_IXOTH) ? 'x' : '-');
    return buf;
}

int main(int argc, char const *argv[]){
    if(argc < 2){
        printf("Incorrect number of arguments, Expected 1, Got: %d", argc-1);
        return 0;
    }
    struct stat file_stat;
    int return_value = stat(argv[1], &file_stat);
    if(return_value != 0){
        perror("stat");
        return 1;
    }
    printf("File: %s\nInode: %ld\nDevice Container ID: %ld\nMode: %10.10s\nHard Links: %ld\nUser ID: %d\nGroup ID: %d\nDevice ID: %ld\nTotal Size: %ld\nBlock Size: %ld\nNumber of blocks: %ld\nLast Access Time: %s\nLast Modified: %s\nLast Status: %s", argv[1], file_stat.st_ino, file_stat.st_dev, perms(file_stat.st_mode), file_stat.st_nlink, file_stat.st_uid, file_stat.st_gid, file_stat.st_rdev, file_stat.st_size, file_stat.st_blksize, file_stat.st_blocks, ctime(&file_stat.st_atime), ctime(&file_stat.st_mtime), ctime(&file_stat.st_ctime));
    return 0;
}
```

Output:

```
ugcse@pglab-cp: ~/Desktop/AyushGoyal_OSLab/Lab_4
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ gcc l4q2.c -o l4q2
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q2 l4q1.c
File: l4q1.c
Inode: 6292684
Device Container ID: 2057
Mode:  rw-rw-r--
Hard Links: 1
User ID: 1001
Group ID: 1001
Device ID: 0
Total Size: 449
Block Size: 4096
Number of blocks: 8
Last Access Time: Sat Oct 30 13:27:29 2021
Last Modified: Sat Oct 30 13:27:29 2021
Last Status: Sat Oct 30 13:27:29 2021
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ stat l4q1.c
  File: l4q1.c
  Size: 449          Blocks: 8          IO Block: 4096   regular file
Device: 809h/2057d  Inode: 6292684    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1001/   ugcse)   Gid: ( 1001/   ugcse)
Access: 2021-10-30 13:27:29.317127736 +0530
Modify: 2021-10-30 13:27:27.965127679 +0530
Change: 2021-10-30 13:27:27.965127679 +0530
 Birth: -
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$
```

3. 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 new path.

Code:

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

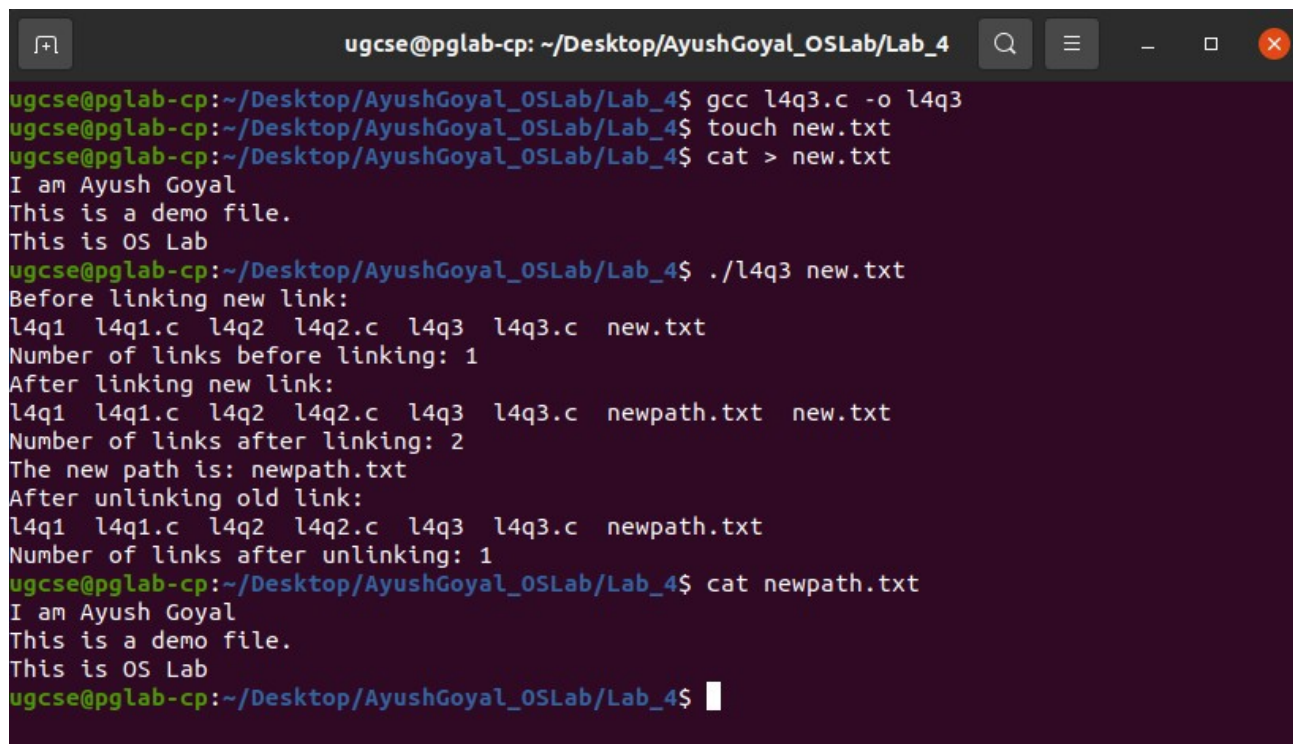
int main(int argc, char *argv[])
{
    if(argc < 2){
        printf("Incorrect number of arguments, Expected 1, Got: %d", argc-1);
        return 0;
    }
    char *oldpath = argv[1];
    char *newpath = "newpath.txt";
    struct stat file_stat;
    printf("Before linking new link:\n");
    system("ls");
    int return_value = stat(oldpath, &file_stat);
```

```

        if(return_value != 0){
            perror("stat");
            return 1;
        }
        printf("Number of links before linking: %ld\n", file_stat.st_nlink);
        link(oldpath, newpath);
        printf("After linking new link:\n");
        system("ls");
        int return_value_2 = stat(newpath, &file_stat);
        if(return_value_2 != 0){
            perror("stat");
            return 1;
        }
        printf("Number of links after linking: %ld\nThe new path is: %s\n", file_stat.st_nlink,
newpath);
        unlink(oldpath);
        printf("After unlinking old link:\n");
        system("ls");
        int return_value_3 = stat(newpath, &file_stat);
        if(return_value_3 != 0){
            perror("stat");
            return 1;
        }
        printf("Number of links after unlinking: %ld\n", file_stat.st_nlink);
        return 0;
    }
}

```

Output:



```

ugcse@pglab-cp: ~/Desktop/AyushGoyal_OSLab/Lab_4
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ gcc l4q3.c -o l4q3
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ touch new.txt
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ cat > new.txt
I am Ayush Goyal
This is a demo file.
This is OS Lab
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q3 new.txt
Before linking new link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c new.txt
Number of links before linking: 1
After linking new link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c newpath.txt new.txt
Number of links after linking: 2
The new path is: newpath.txt
After unlinking old link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c newpath.txt
Number of links after unlinking: 1
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ cat newpath.txt
I am Ayush Goyal
This is a demo file.
This is OS Lab
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$

```

4. 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 new path.

Code:

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

int main(int argc, char *argv[])
{
    if(argc < 2){
        printf("Incorrect number of arguments, Expected 1, Got: %d", argc-1);
        return 0;
    }
    char *oldpath = argv[1];
    char *newpath = "newpath.txt";
    struct stat file_stat;
    printf("Before linking new link:\n");
    system("ls");
    int return_value = stat(oldpath, &file_stat);
    if(return_value != 0){
        perror("stat");
        return 1;
    }
    printf("Number of links before linking: %ld\n", file_stat.st_nlink);

    symlink(oldpath, newpath);

    printf("After linking new link:\n");
    system("ls");
    int return_value_2 = stat(newpath, &file_stat);
    if(return_value_2 != 0){
        perror("stat");
        return 1;
    }
    printf("Number of links after linking: %ld\nThe new path is: %s\n", file_stat.st_nlink,
newpath);
    unlink(newpath);
    printf("After unlinking new soft link:\n");
    system("ls");
    int return_value_3 = stat(oldpath, &file_stat);
    if(return_value_3 != 0){
        perror("stat");
        return 1;
    }
    printf("Number of links after unlinking: %ld\n", file_stat.st_nlink);
    return 0;
}
```

Output:

```
ugcse@pglab-cp: ~/Desktop/AyushGoyal_OSLab/Lab_4
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ gcc l4q4.c -o l4q4
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ cat new.txt
Hello this is 4th question
This is a new file.
I am Ayush Goyal.
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$ ./l4q4 new.txt
Before linking new link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c l4q4 l4q4.c new.txt
Number of links before linking: 1
After linking new link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c l4q4 l4q4.c newpath.txt new.txt
Number of links after linking: 1
The new path is: newpath.txt
After unlinking new soft link:
l4q1 l4q1.c l4q2 l4q2.c l4q3 l4q3.c l4q4 l4q4.c new.txt
Number of links after unlinking: 1
ugcse@pglab-cp:~/Desktop/AyushGoyal_OSLab/Lab_4$
```

THE END