Ayush Goyal
190905522 CSE D Roll 62

Operating Systems Lab 2: Working with Directory Structures

1. Write a C program to emulate the ls -l UNIX command that prints all files in a current directory and lists access privileges, etc. DO NOT simply exec ls -l from the program.

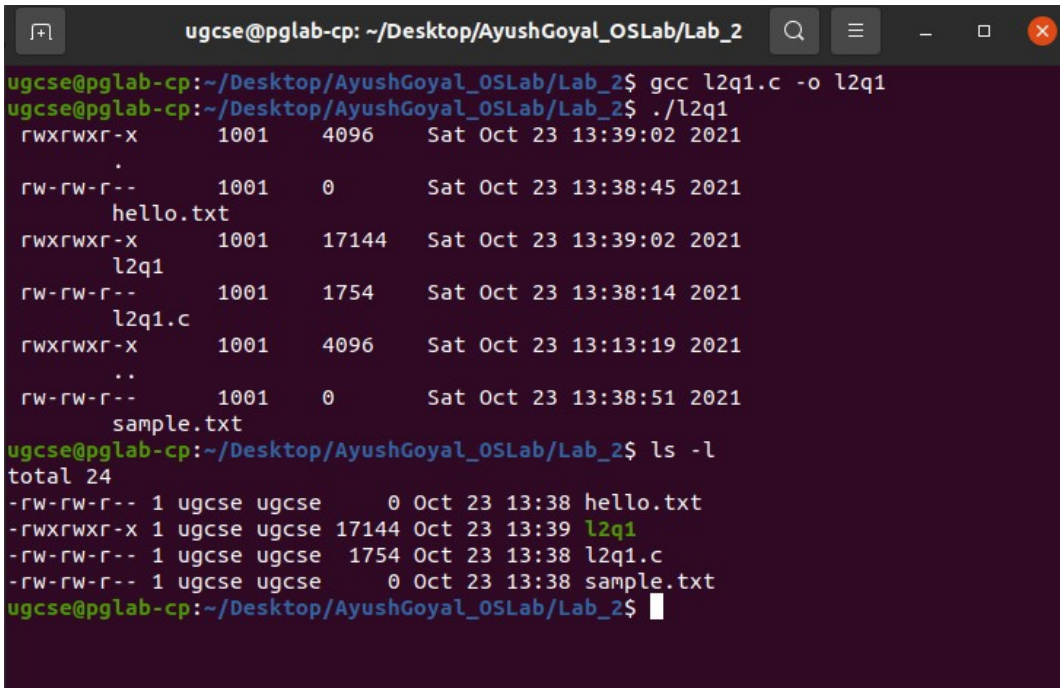
Code:

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <time.h>
#include <string.h>
#include <pwd.h>
#include <grp.h>
#include <locale.h>
#include <langinfo.h>
#include <stdint.h>

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

	return local_buff;
}

int main(int argc, char *argv[]){
	DIR *mydir;
	struct dirent *myfile;
	struct stat mystat;
	if(argc == 1)
		mydir = opendir(".");
	else
		mydir = opendir(argv[1]);
```

```
        while((myfile = readdir(mydir)) != NULL){
                bzero(&mystat, sizeof(mystat));
                stat(myfile->d_name, &mystat);
                printf("%10.10s", perms(mystat.st_mode));
                printf("\t%d", mystat.st_uid);
                printf("\t%ld\t",mystat.st_size);
                printf("%s\t", ctime(&mystat.st_mtime));
                printf("%s\n", myfile->d_name);
        }
        closedir(mydir);
}
```

The output of the above executed program resembles that of the UNIX command ls -l as shown in the screenshot below:

Output:



2. Write a program that will list all files in a current directory and all files in subsequent subdirectories.

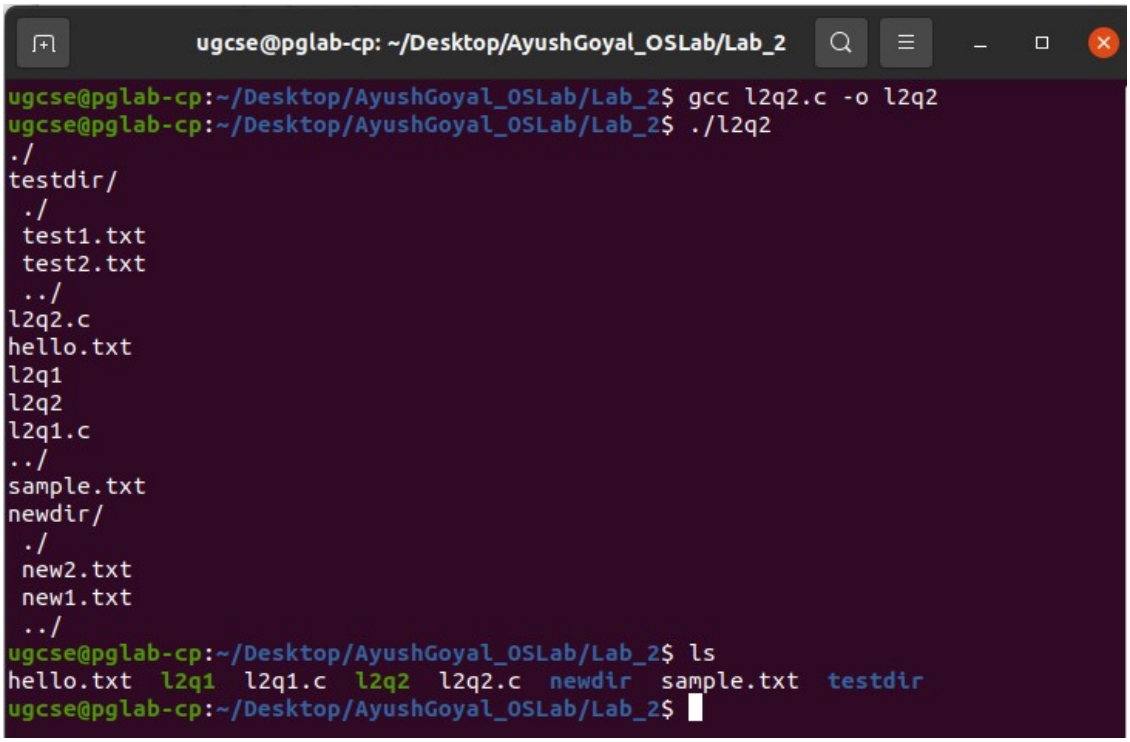
Code:

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

void executeFunc(char* dir, int depth){
```

```
        struct dirent *myfile;
        struct stat mystat;
        DIR *mydir = opendir(dir);
        chdir(dir);
        while((myfile = readdir(mydir)) != NULL){
                bzero(&mystat, sizeof(mystat));
                stat(myfile->d_name, &mystat);
                for(int i=0;i<depth;i++)
                        printf(" ");
                printf("%s", myfile->d_name);
                if(S_ISDIR(mystat.st_mode))
                        printf("/");
                printf("\n");
                if(S_ISDIR(mystat.st_mode) && strcmp(myfile->d_name, ".") != 0 &&
strcmp(myfile->d_name, "..") != 0){
                        executeFunc(myfile->d_name, depth+1);
                }
        }
        chdir("..");
        closedir(mydir);
}

int main(){
        executeFunc(".", 0);
        return 0;
}
```

I have created two directories “testdir” and “newdir” to test and added text files in them to test the above program. The output shown below lists all files and directories as required:

Output:

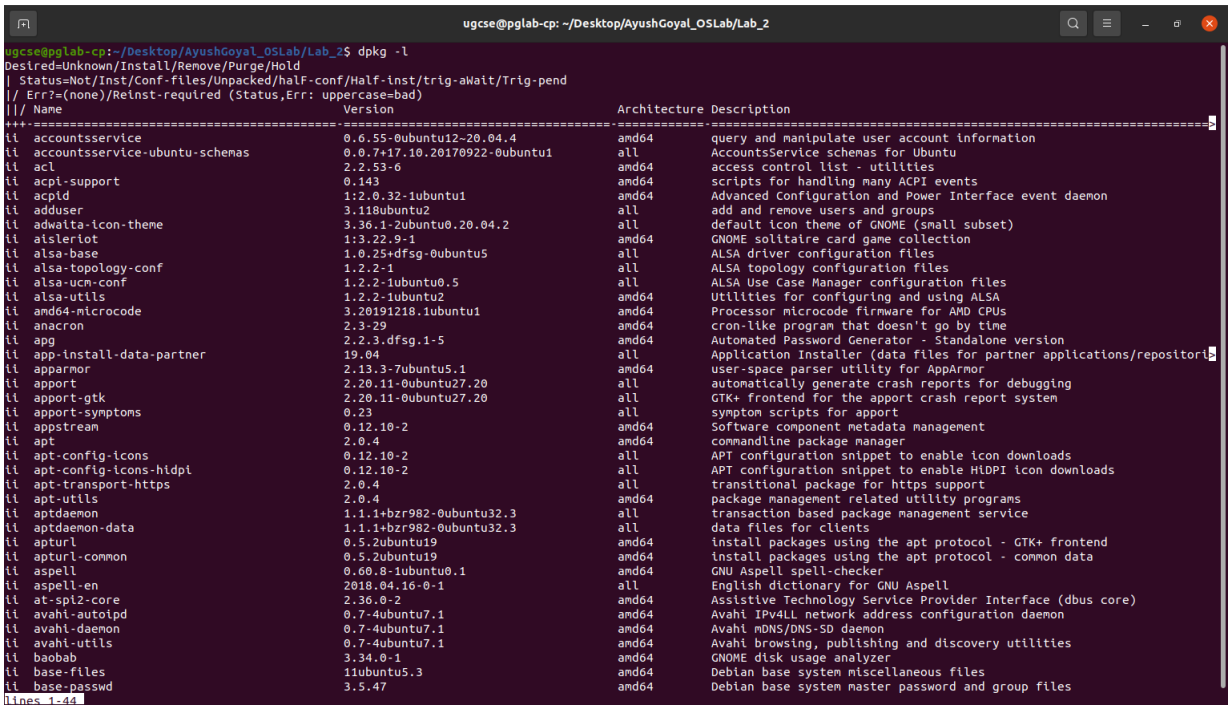


3. How do you list all installed programs in Linux?

Code:

$ dpkg -l

Output:

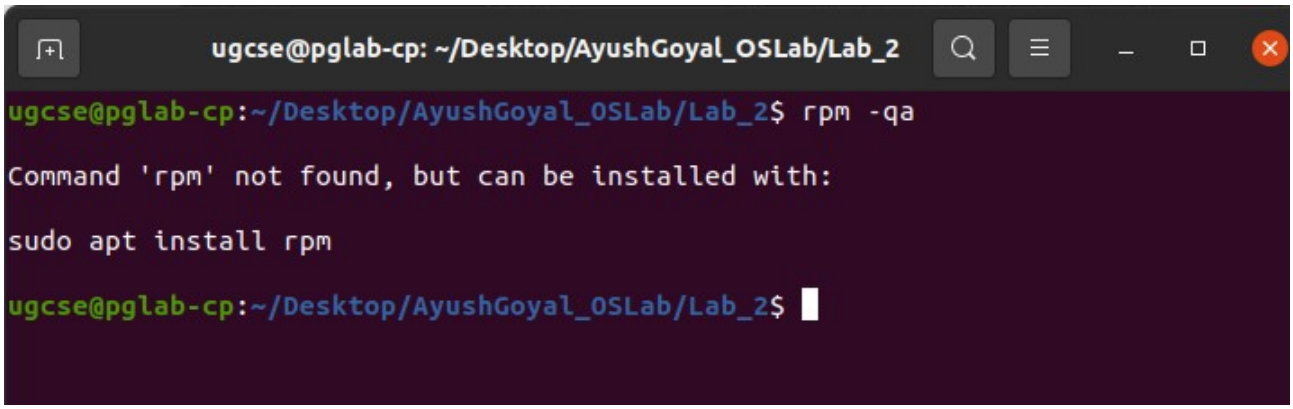


4. How do you find out what RPM packages are installed on Linux?

Code:
$ rpm -qa

RPM is not installed on Ubuntu by default. Can be installed by following command:
$ sudo apt install rpm

Output:



THE END