Dennis Johnson
180905025
CSE - B
Batch 3

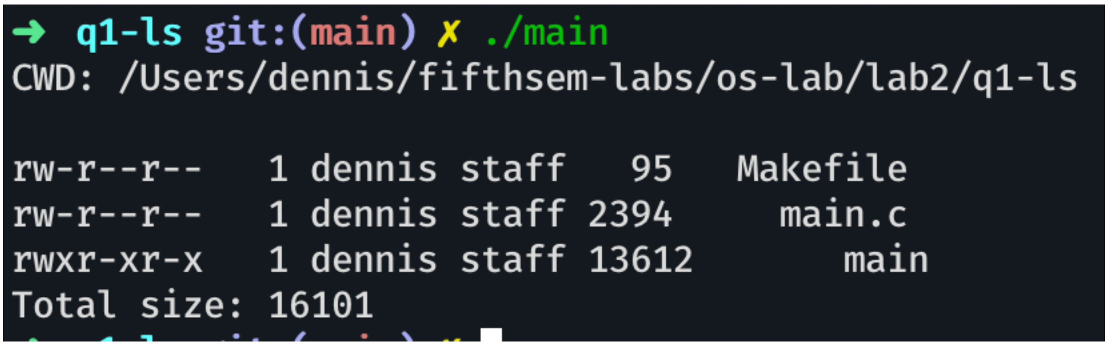
OS – Lab

LAB-2

Q1.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.

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

#define LEN_BUFFER 256
/*
 Program to emulate behaviour of ls -l command
*/

long int displayFile(struct dirent* entry, struct stat stat_buff);
void formatID(const mode_t permissions, char *buffer);

int main(){
  char cwd[LEN_BUFFER];

  getcwd(cwd, sizeof(cwd));
  printf("CWD: %s\n\n", cwd);

  DIR* dp;
  struct dirent* entry;

  if((dp = opendir(cwd)) == NULL){
    fprintf(stderr, "Could not open directory %s", cwd);
    perror(" ");
    exit(EXIT_FAILURE);
  }

  struct stat stat_buff;
  long int totalSize = 0;

  while((entry = readdir(dp)) != NULL){
    //Skip printing info of . and .. dirs
    if(strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0)
      continue;

    lstat(entry->d_name, &stat_buff);
    totalSize += displayFile(entry, stat_buff);
  }

  closedir(dp);
  printf("Total size: %li\n", totalSize);
  return 0;
}

long int displayFile(struct dirent* entry, struct stat stat_buff){
  mode_t permissions = stat_buff.st_mode;
  nlink_t numLinks = stat_buff.st_nlink;
  uid_t userID = stat_buff.st_uid;
  gid_t groupID = stat_buff.st_gid;
  off_t size = stat_buff.st_size;
```

```
}
  char username[LEN_BUFFER], groupname[LEN_BUFFER], perm_formatted[32];
  memset(username, '\0', sizeof(username));
  memset(groupname, '\0', sizeof(groupname));

  struct passwd* uid = getpwuid(userID);
  struct group* gid = getgrgid(groupID);

  strcpy(username, uid->pw_name);
  strcpy(groupname, gid->gr_name);
  formatID(permissions, perm_formatted);

  printf("%s %3lu %s %s %4li %10s\n", perm_formatted, numLinks, username, groupname, size, entry->d_name);

  return size;
}

#define sc(ch) strcat(buffer,(ch))
void formatID(const mode_t permissions, char *buffer){
  memset(buffer, '\0', 32 * sizeof(char));
  if(permissions & S_IRUSR) sc("r"); else sc("-");
  if(permissions & S_IWUSR) sc("w"); else sc("-");
  if(permissions & S_IXUSR) sc("x"); else sc("-");
  if(permissions & S_IRGRP) sc("r"); else sc("-");
  if(permissions & S_IWGRP) sc("w"); else sc("-");
  if(permissions & S_IXGRP) sc("x"); else sc("-");
  if(permissions & S_IROTH) sc("r"); else sc("-");
  if(permissions & S_IWOTH) sc("w"); else sc("-");
  if(permissions & S_IXOTH) sc("x"); else sc("-");
}
```



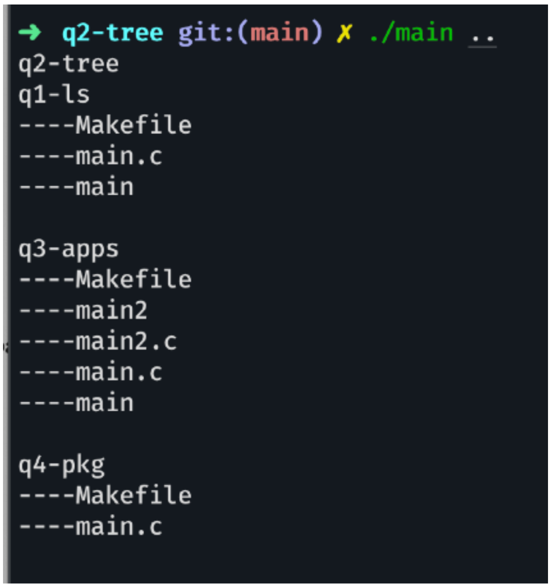
```
➜ q1-ls git:(main) ✗ ./main
CWD: /Users/dennis/fifthsem-labs/os-lab/lab2/q1-ls

rw-r--r--   1 dennis staff   95   Makefile
rw-r--r--   1 dennis staff 2394     main.c
rwxr-xr-x   1 dennis staff 13612       main
Total size: 16101
```

Q1

Q2. Write a program that will list all files in a current directory and all files in subsequent subdirectories.

```
#include <stdio.h>

#include <stdlib.h>

#include <dirent.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <string.h>

#define LEN_BUFFER 256

int recurse(char* path, int depth);


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

 if(argc != 2){

  fprintf(stderr, "Usage: arg1 - src of directory");

  exit(EXIT_FAILURE);

 }


 recurse(argv[1], 0);

 return 0;

}
```

```
int recurse(char *path, int depth){
  DIR* dp;
  if((dp = opendir(path)) == NULL){
    //Invalid dir or path specified is not of a dir
    return -1;
  }

  struct dirent* entry;
  struct stat stat_buff;

  while((entry = readdir(dp)) != NULL){
    if(strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0)
      continue;

    int temp = depth * 2;
    while(temp--) printf("--");
    printf("%s\n", entry->d_name);

    lstat(entry->d_name, &stat_buff);
    if (S_ISDIR(stat_buff.st_mode) == 0){
      //Found a nested directory
      char new_path [LEN_BUFFER];
      strcat(new_path, path);
      strcat(new_path, "/");
      strcat(new_path, entry->d_name);

      recurse(new_path, depth + 1);
      memset(new_path, '\0', sizeof(new_path));
    }
  }
  printf("\n");

  closedir(dp);

  return 0;

}
```

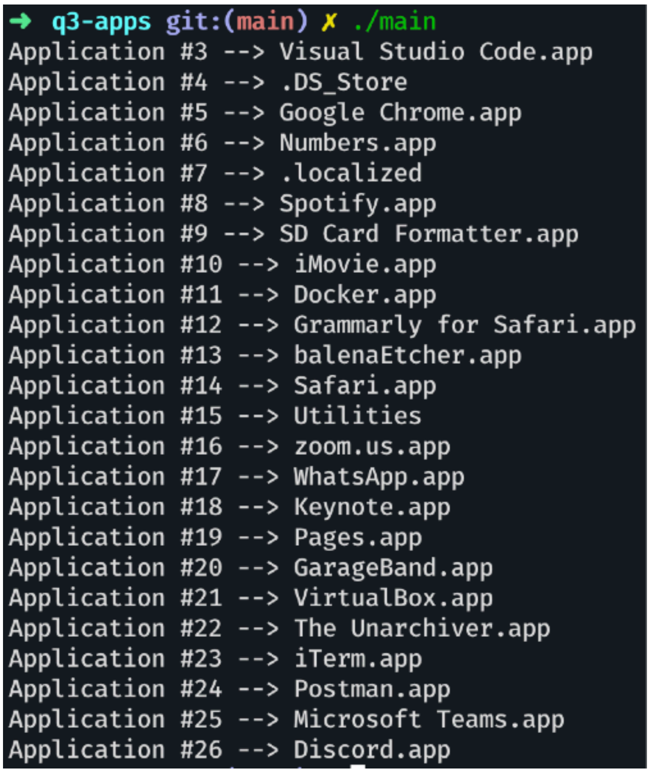


```
➜  q2-tree git:(main) ✗ ./main ..
q2-tree
q1-ls
----Makefile
----main.c
----main

q3-apps
----Makefile
----main2
----main2.c
----main.c
----main

q4-pkg
----Makefile
----main.c
```

Q2

Q3. How do you list all installed programs in Linux?

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

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <string.h>

#include <dirent.h>

#define LEN_BUFFER 256

/*

 Program to list all installed Applications

*/

void displayFile(struct dirent* entry, struct stat stat_buff);


int main(){

  //Path to Apps folder, here I'm using OS X
  char* path = "/Applications";
  int appCount = 0;
  DIR* dp;
  struct dirent* entry;
  if((dp = opendir(path)) == NULL){
   fprintf(stderr, "Could not open directory %s", path);
   perror(" ");
   exit(EXIT_FAILURE);
  }
  while((entry = readdir(dp)) != NULL){
   appCount ++;
   //Skip printing info of . and .. dirs
   if(strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0)
    continue;
   // Print Application names
   printf("Application #%d --> %s\n", appCount, entry->d_name);
  }
  closedir(dp);
  return 0;
}
```

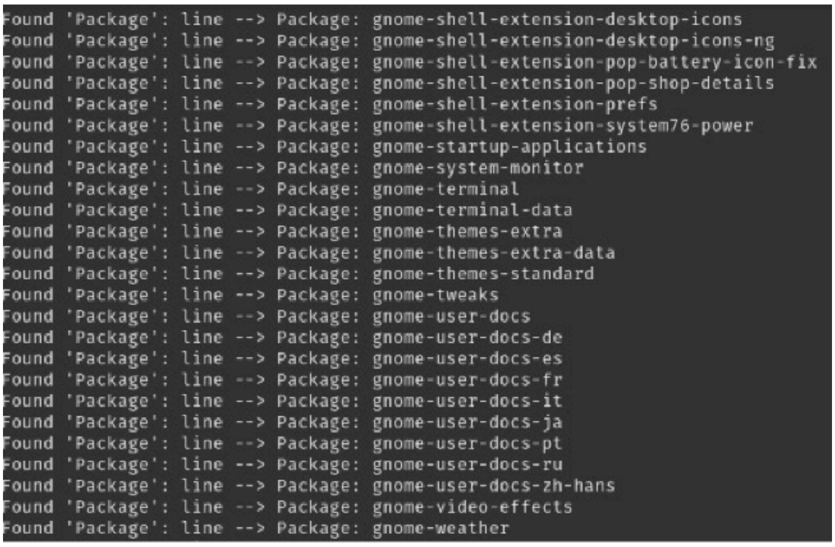


```
➜ q3-apps git:(main) ✗ ./main
Application #3 --> Visual Studio Code.app
Application #4 --> .DS_Store
Application #5 --> Google Chrome.app
Application #6 --> Numbers.app
Application #7 --> .localized
Application #8 --> Spotify.app
Application #9 --> SD Card Formatter.app
Application #10 --> iMovie.app
Application #11 --> Docker.app
Application #12 --> Grammarly for Safari.app
Application #13 --> balenaEtcher.app
Application #14 --> Safari.app
Application #15 --> Utilities
Application #16 --> zoom.us.app
Application #17 --> WhatsApp.app
Application #18 --> Keynote.app
Application #19 --> Pages.app
Application #20 --> GarageBand.app
Application #21 --> VirtualBox.app
Application #22 --> The Unarchiver.app
Application #23 --> iTerm.app
Application #24 --> Postman.app
Application #25 --> Microsoft Teams.app
Application #26 --> Discord.app
```

Q3

Q4. How do you find out what DPKG packages are installed on Linux?

```
#include<fcntl.h>
#include<string.h>
#include<stdlib.h>

#define LEN_BUFFER 256
/*
 * Similar to Lab1/Q1 grep tool
   Here we look for Packages in /var/lib/dpkg/status file
 * */
int main(){
  int in;
  char ch, buffer[LEN_BUFFER];

  char *path = "/var/lib/dpkg/status";
  char *key  = "Package";

  if((in = open(path, O_RDONLY)) < 0){
    fprintf(stderr, "Error opening file %s", path);
    perror(" ");
    exit(EXIT_FAILURE);
  }

  size_t len_read, char_count = 0;
  while((len_read = read(in, &ch, sizeof(char))) > 0){
    if (ch == '\n'){
      buffer[char_count] = '\0';

      if(strstr(buffer, key) != NULL)
        printf("Found '%s': line --> %s\n", key, buffer);
      char_count = 0;
      memset(buffer, '\0', sizeof(buffer));
    }

    else {
      buffer[char_count++] = ch;
    }
  }

  close(in);
            return 0;
}
```



```
Found 'Package': line --> Package: gnome-shell-extension-desktop-icons
Found 'Package': line --> Package: gnome-shell-extension-desktop-icons-ng
Found 'Package': line --> Package: gnome-shell-extension-pop-battery-icon-fix
Found 'Package': line --> Package: gnome-shell-extension-pop-shop-details
Found 'Package': line --> Package: gnome-shell-extension-prefs
Found 'Package': line --> Package: gnome-shell-extension-system76-power
Found 'Package': line --> Package: gnome-startup-applications
Found 'Package': line --> Package: gnome-system-monitor
Found 'Package': line --> Package: gnome-terminal
Found 'Package': line --> Package: gnome-terminal-data
Found 'Package': line --> Package: gnome-themes-extra
Found 'Package': line --> Package: gnome-themes-extra-data
Found 'Package': line --> Package: gnome-themes-standard
Found 'Package': line --> Package: gnome-tweaks
Found 'Package': line --> Package: gnome-user-docs
Found 'Package': line --> Package: gnome-user-docs-de
Found 'Package': line --> Package: gnome-user-docs-es
Found 'Package': line --> Package: gnome-user-docs-fr
Found 'Package': line --> Package: gnome-user-docs-it
Found 'Package': line --> Package: gnome-user-docs-ja
Found 'Package': line --> Package: gnome-user-docs-pt
Found 'Package': line --> Package: gnome-user-docs-ru
Found 'Package': line --> Package: gnome-user-docs-zh-hans
Found 'Package': line --> Package: gnome-video-effects
Found 'Package': line --> Package: gnome-weather
```

Q4