```
// # CS532_Lab5

// $gcc lab5_v2.c
// $./a.out projects/

// (out)
// [1] . (directory)
// [2] .. (directory)
// [3] .DS_Store (regular file)
// [3] fread.c (regular file)
// [3] fwrite.c (regular file)
// [3] project1 (directory)
//           [3] . (directory)
//           [4] .. (directory)
//           [5] project1.docx (regular file)
//           [5] README (regular file)
// [4] project2 (directory)
//           [4] . (directory)
//           [5] .. (directory)
//           [6] project2.docx (regular file)
//           [6] README (regular file)
// [5] project3 (directory)
//           [5] . (directory)
//           [6] .. (directory)
//           [7] project3.docx (regular file)
//           [7] README (regular file)
// [6] project4 (directory)
//           [6] . (directory)
//           [7] .. (directory)
//           [8] project4.docx (regular file)
//           [8] README (regular file)
// [7] read.c (regular file)
// [7] write.c (regular file)


#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>

char *filetype(unsigned char type);

// Read Directory
void test(char *parentDir, int count, int c){
    struct dirent *dirent;
    int i=0;
    char dirname[1024];

    memset(dirname, ’\0’,sizeof(dirname)); // directory name

    DIR *Dir = opendir(parentDir);
    if (Dir == NULL){
        printf("Error\n");
        exit(EXIT_FAILURE);
    }

    while((dirent = readdir(Dir)) != NULL){
        for (i = 0; i < c; i++){
            printf ("          ");  // files in "------" directory
        }

        printf ("[%d] %s (%s)\n", count, dirent->d_name, filetype(dirent->d_type));
        if (dirent->d_type != DT_DIR){     // check if directory or not
```

```
            continue;
        }
        // if in directory, check the files in directory and read file again
        if (strcmp(dirent->d_name, ".") != 0 && strcmp(dirent->d_name, "..") != 0){
            sprintf(dirname, "%s/%s", parentDir, dirent->d_name);
            test(dirname, count, c + 1);    // in a directory, so count+1
        }
        count++;   // in a directory, so count+1
    }
    closedir (Dir);
}

char *filetype(unsigned char type) {
  char *str;
  switch(type) {
  case DT_BLK: str = "block device"; break;
  case DT_CHR: str = "character device"; break;
  case DT_DIR: str = "directory"; break;
  case DT_FIFO: str = "named pipe (FIFO)"; break;
  case DT_LNK: str = "symbolic link"; break;
  case DT_REG: str = "regular file"; break;
  case DT_SOCK: str = "UNIX domain socket"; break;
  case DT_UNKNOWN: str = "unknown file type"; break;
  default: str = "UNKNOWN";
  }
  return str;
}

int main (int argc, char **argv){
    DIR *parentDir;

    if (argc < 2) {
        printf ("Usage: %s <dirname>\n", argv[0]);
        exit(-1);
    }
    parentDir = opendir (argv[1]);
    if (parentDir == NULL) {
        printf ("Error opening directory ’%s’\n", argv[1]);
        exit (-1);
    }

    char filename[1024];

    memset(filename,’\0’,sizeof(filename));
    getcwd(filename, sizeof(filename));
    strcat(filename,"/");
    strcat(filename,argv[1]);
    printf("%s\n", filename);

    int count = 1;
    int c = 0;
    test(argv[1], count, c);
    return 0;
}
```