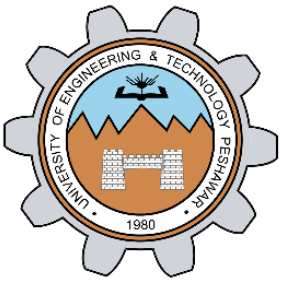
LAB NO 09

TRAVERSING DIRECTORIES



Fall 2024

CSE-302L Systems Programming Lab

Submitted by:

Name: **Hassan Zaib Jadoon**

Reg no.**: 22PWCSE2144**

Class Section**: A**

Signature: _____

Submitted to:

Engr. Abdullah Hamid

December 29, 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

LAB NO 09 TRAVERSING DIRECTORIES

Task 1

Traverse directory tree in depth-first order.

Code:

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

void dp1(const char *dir) {
  struct dirent *dp;
  struct stat buffer;
  DIR *d = opendir(dir);
   chdir(dir);
  while ((dp = readdir(d)) != NULL) {
       if (dp->d_name[0] == '.') {
      continue;
     }
     printf("%s\n", dp->d_name);
     stat(dp->d_name, &buffer);
      if (S_ISDIR(buffer.st_mode)) {
    dp1(dp->d_name);
        chdir("..");
      }
    }
}
int main(int argc, char *argv[]) {
    dp1(argv[1]);
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 09$ ./task1.o ~/Desktop/"SP Lab"
SP Lab 06
task2.o
f3.txt
f4.txt
f2.txt
task1.c
task1.o
f1.txt
task2.c
SP Lab 09
task2.o
task1.c
task1.o
task2.c
SP Lab 10
task2.o
task1.c
task1.o
task3
fifo
task3.o
task3.c
task2.c
SP Lab 07
task2.o
src.txt
task1.c
dest2.txt
task1.o
task4.c
task3.o
src2.txt
src3.txt
dest3.txt
```

Task 2

Traverse directory tree in breadth-first order.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/stat.h>
#include <string.h>
#include <unistd.h>
typedef struct QueueNode {
    char path[1024];
    struct QueueNode *next;
} QueueNode;
typedef struct {
    QueueNode *front;
    QueueNode *rear;
} Queue;
void enqueue(Queue *q, const char *path) {
    QueueNode *node = malloc(sizeof(QueueNode));
    strcpy(node->path, path);
    node->next = NULL;
    if (q->rear) {
        q->rear->next = node;
    } else {
        q->front = node;
    }
    q->rear = node;
}

int dequeue(Queue *q, char *path) {
    if (!q->front) return 0;
    QueueNode *node = q->front;
    strcpy(path, node->path);
    q->front = node->next;
    if (!q->front) q->rear = NULL;
    free(node);
    return 1;
}

void dp1(const char *root) {
    Queue q = {NULL, NULL};
    enqueue(&q, root);
    char currentPath[1024];
    struct dirent *dp;
    struct stat buffer;

    while (dequeue(&q, currentPath)) {
        DIR *d = opendir(currentPath);
        if (!d) continue;
        printf("%s\n", currentPath);
        while ((dp = readdir(d)) != NULL) {
            if (dp->d_name[0] == '.') continue;

            char fullPath[1024];
            snprintf(fullPath, sizeof(fullPath), "%s/%s", currentPath, dp->d_name);

            stat(fullPath, &buffer);
            if (S_ISDIR(buffer.st_mode)) {
                enqueue(&q, fullPath);
            } else {
                printf("%s\n", fullPath);
            }
        }
        closedir(d);
    }
}

int main(int argc, char *argv[]) {
    dp1(argv[1]);
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 09$ ./task2.o ~/Desktop/"SP Lab"
/home/hassan/Desktop/SP Lab
/home/hassan/Desktop/SP Lab/SP Lab 06
/home/hassan/Desktop/SP Lab/SP Lab 06/task2.o
/home/hassan/Desktop/SP Lab/SP Lab 06/f3.txt
/home/hassan/Desktop/SP Lab/SP Lab 06/f4.txt
/home/hassan/Desktop/SP Lab/SP Lab 06/f2.txt
/home/hassan/Desktop/SP Lab/SP Lab 06/task1.c
/home/hassan/Desktop/SP Lab/SP Lab 06/task1.o
/home/hassan/Desktop/SP Lab/SP Lab 06/f1.txt
/home/hassan/Desktop/SP Lab/SP Lab 06/task2.c
/home/hassan/Desktop/SP Lab/SP Lab 09
/home/hassan/Desktop/SP Lab/SP Lab 09/task2.o
/home/hassan/Desktop/SP Lab/SP Lab 09/task1.c
/home/hassan/Desktop/SP Lab/SP Lab 09/task1.o
/home/hassan/Desktop/SP Lab/SP Lab 09/task2.c
/home/hassan/Desktop/SP Lab/SP Lab 10
/home/hassan/Desktop/SP Lab/SP Lab 10/task2.o
/home/hassan/Desktop/SP Lab/SP Lab 10/task1.c
/home/hassan/Desktop/SP Lab/SP Lab 10/task1.o
/home/hassan/Desktop/SP Lab/SP Lab 10/task2.c
/home/hassan/Desktop/SP Lab/SP Lab 07
/home/hassan/Desktop/SP Lab/SP Lab 07/task2.o
/home/hassan/Desktop/SP Lab/SP Lab 07/src.txt
/home/hassan/Desktop/SP Lab/SP Lab 07/task1.c
/home/hassan/Desktop/SP Lab/SP Lab 07/dest2.txt
```

CSE 302L: SYSTEMS PROGRAMMING LAB

LAB ASSESSMENT RUBRICS

Criteria & Point Assigned	Outstanding 2	Acceptable 1.5	Considerable 1	Below Expectations 0.5	Score
Attendance and Attentiveness in Lab PLO08	Attended in proper Time and attentive in Lab	Attended in proper Time but not attentive in Lab	Attended late but attentive in Lab	Attended late not attentive in Lab	
Capability of writing Program/ Algorithm/Drawing Flow Chart PLO1, PLO2, PLO3, PLO5,	Right attempt/ no errors and well formatted	Right attempt/ no errors but not well formatted	Right attempt/ minor errors and not well formatted	Wrong attempt	
Result or Output/ Completion of target in Lab PLO9,	100% target has been completed and well formatted.	75% target has been completed and well formatted.	50% target has been completed but not well formatted.	None of the outputs are correct	
Overall, Knowledge PLO10,	Demonstrates excellent knowledge of lab	Demonstrates good knowledge of lab	Has partial idea about the Lab and procedure followed	Has poor idea about the Lab and procedure followed	
Attention to Lab Report PLO4,	Submission of Lab Report in Proper Time i.e., in next day of lab., with proper documentation.	Submission of Lab Report in proper time but not with proper documentation.	Late Submission with proper documentation.	Late Submission Very poor documentation	

Instructor:

Name: _____ Signature: _____