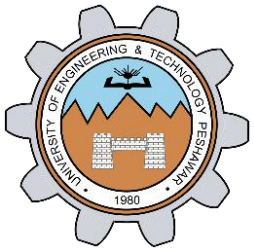
Systems Programming LAB
Lab 9



Fall 2023

Submitted by: **Hamza Mateen**

Registration No. : **21PWCSE2013**

Class Section: **C**

"As student of University of Engineering & Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: Hamza

Submitted to:

Engr. Abdullah Hamid

Jan 31, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Q NO 1: Breadth First Directory Search

Code:

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

void breadth (char* directoryName) {
// lab 9
// SP lab f4
// f1 f2 f3
```

```c
struct stat myStat;
DIR* directoryPtr = opendir(directoryName);
int r = chdir(directoryName); // change to lab 9
struct dirent* myDirectoryEntry;
while ((myDirectoryEntry = readdir(directoryPtr)) != NULL)
{
if (strcmp(myDirectoryEntry->d_name, ".") == 0 ||
strcmp(myDirectoryEntry->d_name, "..") == 0 ) {
continue;
}
// print the contents of current or changed to diretory
printf("%s\t", myDirectoryEntry->d_name);
}
```

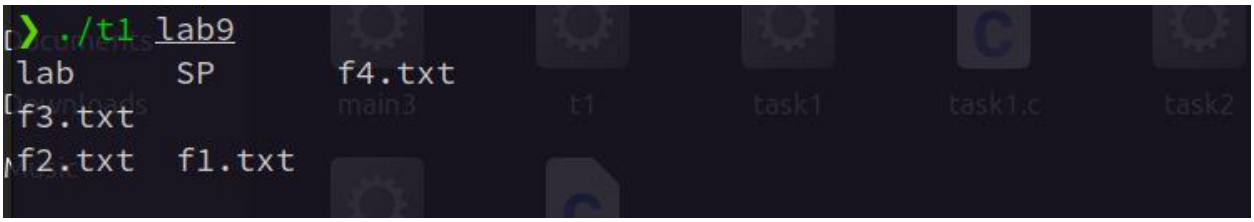
```c
printf("\n");
rewinddir(directoryPtr);
```

```c
while ((myDirectoryEntry = readdir(directoryPtr)) != NULL) {
if (strcmp(myDirectoryEntry->d_name, ".") == 0 ||
strcmp(myDirectoryEntry->d_name, "..") == 0 ) {
continue;
}
// get the stats of the file or directory
int retValue = stat(myDirectoryEntry->d_name, &myStat);
if (S_ISDIR(myStat.st_mode)) {
// call for this directoyr this function recursively now
breadth(myDirectoryEntry->d_name);
chdir("..");
}
```

```
}
}
```

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

Ouput



Q NO 2: Depth First Directory Search

Code:

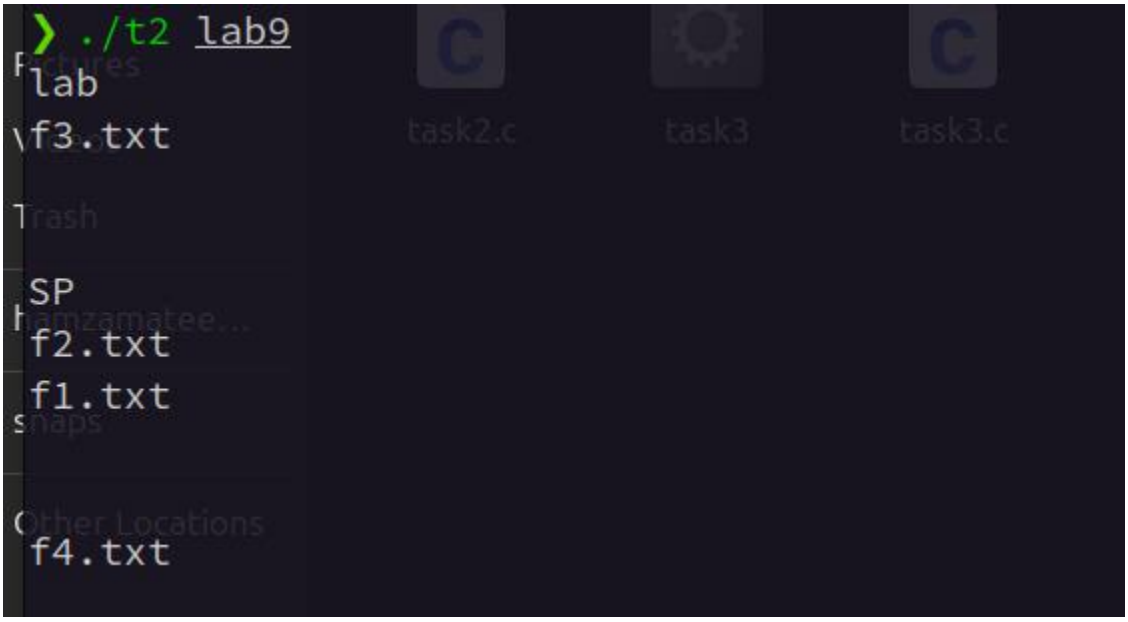
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/stat.h>
#include <time.h>
#include <fcntl.h>
#include <dirent.h>

void depth (char* directoryName) {
// lab 9
// SP lab f4
// f1 f2 f3
struct stat myStat;
struct dirent* myDirectoryEntry;
// open the directry whose name is passed as argument
```

```
DIR* directoryPtr = opendir(directoryName);
int r = chdir(directoryName); // change to lab 9
```

```
while((myDirectoryEntry = readdir(directoryPtr)) != NULL) {
if (strcmp(myDirectoryEntry->d_name, ".") == 0 ||
```

```
strcmp(myDirectoryEntry->d_name, "..") == 0 ) {
continue;
}

int ret = stat(myDirectoryEntry->d_name, &myStat);

// check if it is a directory
printf("%s\n", myDirectoryEntry->d_name);
if (S_ISDIR(myStat.st_mode)) {
// open this directory, chdir into it
// which translates to simply call this directory and that's it
depth(myDirectoryEntry->d_name);
chdir("..");
}

// else just print the content (non dir) files in the directory
}

printf("\n\n");
}

int main(int argc, char** argv)
{
depth("lab9");
return 0;
}
```

Ouput



Q NO 3: Find Utility

Code:

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

#include <libgen.h>
```

```c
void find(char *directoryName, char *filename, char *rootDirName) {
// lab 9
// SP lab f4
// f1 f2 f3
```

```c
struct stat myStat;
char path[100];
```

```c
// open the direcotry whose name is passed as argument
getcwd(path, 100);
```

```
// printf("before: %s\n", path);

DIR *directoryPtr = opendir(directoryName);
int r = chdir(directoryName); // change to lab 9

getcwd(path, 100);
// printf("after: %s\n", path);

struct dirent *myDirectoryEntry;
while ((myDirectoryEntry = readdir(directoryPtr)) != NULL) {
if (strcmp(myDirectoryEntry->d_name, ".") == 0 ||
strcmp(myDirectoryEntry->d_name, "..") == 0) {
continue;
}
}

// printf("\n");
rewinddir(directoryPtr);

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

if (strcmp(myDirectoryEntry->d_name, filename) == 0) {
getcwd(path, 100);
printf("found at %s\n", path);
closedir(directoryPtr);
return;
}

// get the stats of the file or directory
int retValue = stat(myDirectoryEntry->d_name, &myStat);
if (S_ISDIR(myStat.st_mode)) {
// call for this directoyr this function recursively now
find(myDirectoryEntry->d_name, filename, rootDirName);
chdir("..");
}
}

closedir(directoryPtr);
}

int main(int argc, char **argv) {
// sharing state in recursion
find(argv[1], argv[2], argv[2]);
```

```
return 0;
}
```

Ouput

```
❯ gcc task3.c -o t3
❯ ./t3 lab9 f3.txt
found at /home/ycombinator/University Semester 5/semester5/SP/LAB/lab09/lab9/lab

~/University Semester 5/semester5/SP/LAB/lab09
```

********** THE END **********