**Directory Traversing**

**Lab no# 09**

****

**Fall 2021**

**CSE-302 System Programming Lab**

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr: Ma’am Madeha sheer**

**February** 20, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

##### Task 01: Traverse directory tree in depth-first order.

##### Source Code:

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <dirent.h>

#include <sys/stat.h>

int i=0;

int depthfirst(char \*name)

{

i++;

DIR \*dirp=opendir(name); //open directory

if(dirp==NULL)

{

printf("Sorry! directory can't openned successfuly\n");

perror("Reason");

return -1;

}

int x1=chdir(name); //change cwd to according to the argument directory.

//if we don't change directory then stat function will not work.

if(x1==-1)

{

printf("Sorry! directory can't changed successfuly\n");

perror("Reason");

return -1;

}

struct dirent \*direntp;

struct stat buffer;

while ((direntp=readdir(dirp))!=NULL)

{

if(direntp->d\_name[0]=='.') //if name start with '.' skip it.

continue; //if we don't skip (.) (..). then these two directories continusly passed to the

//depthfirst function and make a infinite loop.

for(int j=0; j<i; j++)

{

printf("\t"); //each directory entries will print at a tap distance.

}

printf("%s\n",direntp->d\_name);

int ret=stat(direntp->d\_name,&buffer);

if(ret==-1)

{

printf("Sorry! stat function can't executed successfuly\n");

perror("Reason");

return -1;

}

if(S\_ISDIR(buffer.st\_mode)) //if file is directory then again call depthfirst function.

{

depthfirst(direntp->d\_name); //recursively called depthfirst function.

int x2=chdir(".."); //come out from the directory to which we entered at the begining.

//if we don't change directory then stat function will not work.

if(x2==-1)

{

printf("Sorry! directory can't changed successfuly\n");

perror("Reason");

return -1;

}

i--; //during each recursive call of depthfirst() i value will incriment by 1.

//but when depthfirst() return i will decriment by 1 in each case.

}

}

};

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

{

if(argc>1) //if we give more than one argument it will ignore other and will select argv[1].

{

printf("%s\n",argv[1]);

depthfirst(argv[1]);

}

else //if no argument from command line it will traverse current working direcotry.

depthfirst(".");

return 0;

};

##### Output:

##### 

**Task 02: Traverse directory tree in breadth-first order.**

**Source code:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <dirent.h>

#include <sys/stat.h>

#include <pwd.h> //for getcwd() function.

void breadthfirst(char \*name)

{

DIR \*dirp=opendir(name);

if(dirp==NULL)

{

printf("Sorry! directory can't opened successfully\n");

perror("Reason");

exit(0);

}

int x1=chdir(name);//we change directory according to argument of breadthfirst() for stat fuction.

if(x1==-1) //if we don't change then it will give error.

{

printf("Sorry! directory can't changed successfully\n");

perror("Reason");

exit(0);

}

char path[200];

char \*p=getcwd(path,sizeof(path));//this is for getting complete path of current working directory.

if(p==NULL)

{

printf("Sorry! getcwd can't executed successfully\n");

perror("Reason");

exit(0);

}

struct dirent \*direntp;

while((direntp=readdir(dirp))!=NULL) //this is ls command implementation.

{

if(direntp->d\_name[0]=='.')

continue;

printf("%s/%s\n",path,direntp->d\_name); //first we will print all entries of directory.

}

rewinddir(dirp); //back dirp pointer to the first entry of directory.

struct stat buffer;

while((direntp=readdir(dirp))!=NULL) //this loop for checking directory.

{

if(direntp->d\_name[0]=='.') //if we don't continue files having name started with (.)

continue; //then it will make infinite loop.

int ret=stat(direntp->d\_name,&buffer);

if(ret==-1)

{

printf("Sorry! stat can't executed successfully\n");

perror("Reason");

exit(0);

}

if(S\_ISDIR(buffer.st\_mode)) //now we will check which entry is directory.

//if directory we will call breadthfirst() recursevily.

{

breadthfirst(direntp->d\_name); //Recursively called breadthfirst().

int x2=chdir(".."); //after changing directory we will come back from it.

if(x2==-1)

{

printf("Sorry! directory can't Rechanged successfully\n");

perror("Reason");

exit(0);

}

}

}

};

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

{

if(argc>1) //if we give more than one argument it will ignore other and will select argv[1].

{

printf("%s\n",argv[1]);

breadthfirst(argv[1]);

}

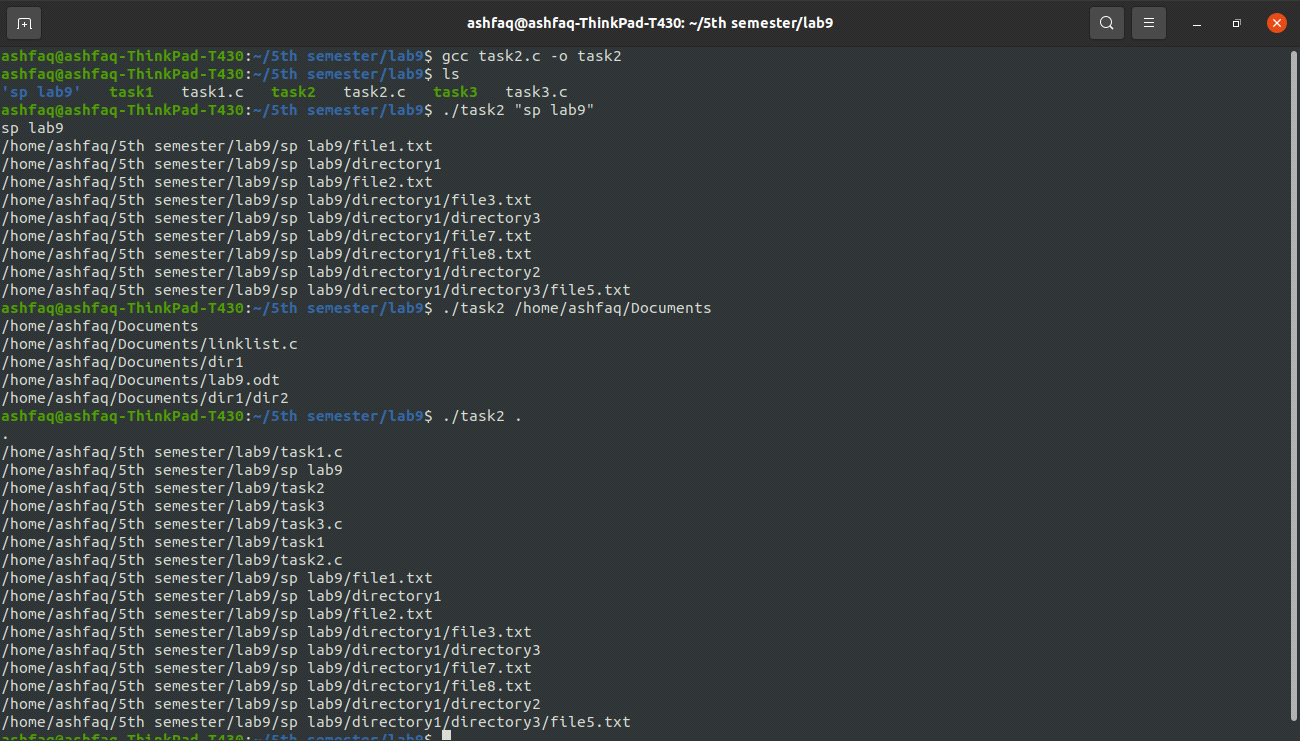
else

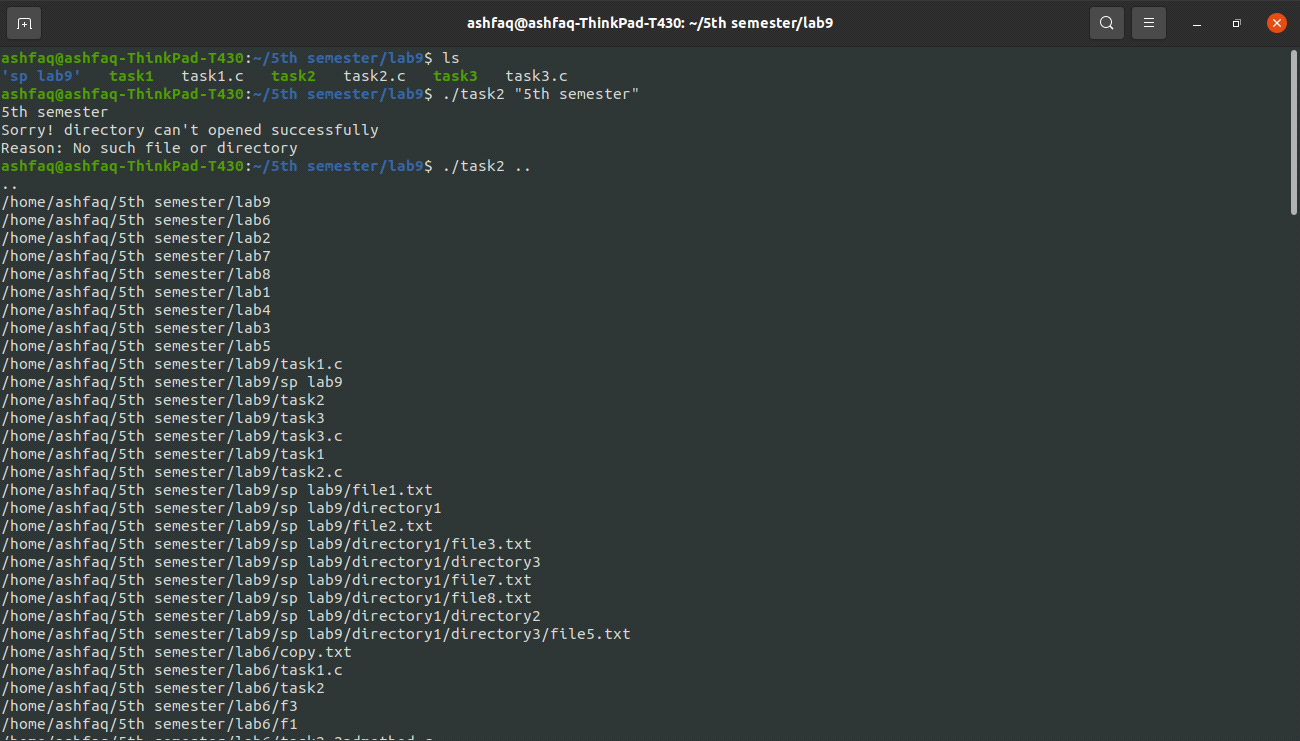
breadthfirst(".");

return 0;

}

**Output:**





**Task 03: Implement the pfind utility**

**Source Code:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <dirent.h>

#include <pwd.h>

#include <sys/stat.h>

#include <string.h>

int file\_count=0;

void pfind(char\* name1,char\* name2)

{

DIR \*dirp=opendir(name1); //open name1

if(dirp==NULL)

{

printf("Sorry! Directory can't openned successfuly\n");

perror("Reason");

exit(0);

}

int d1=chdir(name1); //change direcotry according to name1 otherwise stat() will give error.

if(d1==-1)

{

printf("Sorry! Directory can't changed successfuly\n");

perror("Reason");

exit(0);

}

char cwd[200]; //for getting file location of name2.

char \*ret=getcwd(cwd,sizeof(cwd));

if(ret==NULL)

{

printf("Sorry! path of CWD can't found successfuly\n");

perror("Reason");

exit(0);

}

struct dirent \*direntp;

struct stat buffer;

while((direntp=readdir(dirp))!=NULL)

{

if(direntp->d\_name[0]=='.')

continue;

int sc=strcmp(direntp->d\_name,name2);

//strcmp() compare two strings and return 3 types integers.

//if sc>0 its mean string 1 is greater than string 1.

//if sc<0 its mean string 1 is less than string 2.

//if sc=0 its mean string 1 is equal to string 1.

if(sc==0)

{

printf("File '%s' is Found in given directory:\n",name2);

printf("File Location: %s\n",cwd);

file\_count++; //if file found it will incriment and "file not found" in

//main function will not print.

//if we want to print such statement here it will print many time.

}

int s=stat(direntp->d\_name,&buffer); //getting file information

if(s==-1)

{

printf("Sorry! An error occured with stat function\n");

perror("Reason");

exit(0);

}

if(S\_ISDIR(buffer.st\_mode)) //either file is directory or not.

{

pfind(direntp->d\_name,name2); //recursively called in case of directory.

int d2=chdir(".."); //rechange the current working directory after changed it at

//the begining. if we don't change it then stat function will give error.

if(d2==-1)

{

printf("Sorry! Directory can't Rechanged successfuly\n");

perror("Reason");

exit(0);

}

}

}

}

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

{

if(argc<3) //minimum 2 argument.

{

printf("Sorry! invalid no of arguments\n");

return -1;

}

for (int i=2; i<argc; i++) //for loop for checking more than one file in given directory argv[1].

{

pfind(argv[1],argv[i]);

if(file\_count==0)

{

printf("Sorry! the file '%s' is not found in directory '%s':\n",argv[i],argv[1]);

}

else

file\_count--;

}

return 0;

}

**Output:**

