TRAVERSING DIRECTORIES

LAB # 09

Fall 2020

CSE302L System Programming Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

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. Madiha Sher

Saturday, February 6th, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

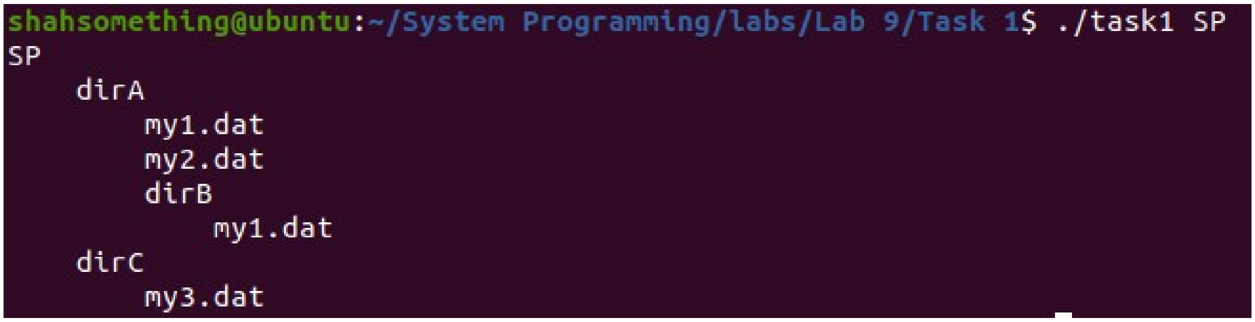
Task # 01:

Traverse directory tree in depth-first order.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <dirent.h>
#include <sys/stat.h>
#include <time.h>
#include <pwd.h>

int i=0;
void depthfirst(char *name)
{
   i++;
   DIR *dirp = opendir(name);
   if(dirp==NULL)
   {
      perror("Failed to open directory");
   }
   struct dirent *direntp;
   struct stat buffer;
   int retchd = chdir(name);
   if(retchd==-1)
          perror("Failed to change directory");
   while((direntp=readdir(dirp))!=NULL)
   {
          if(direntp->d_name[0]=='.')
                              continue;
          for(int j=0;j<i;j++)
                    printf("    ");
          printf("%s\n",direntp->d_name);
          int ret = stat(direntp->d_name,&buffer);
          if(ret==-1)
                    perror("Error using stat");
          if(S_ISDIR(buffer.st_mode))
          {
                    depthfirst(direntp->d_name);
                    chdir("..");
                    i--;
          }
   }
}
int main(int argc, char *argv[])
{
   printf("%s\n",argv[1]);
   depthfirst(argv[1]);
   return 0;
}
```

Output/Results:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 9/Task 1$ ./task1 SP
SP
    dirA
        my1.dat
        my2.dat
        dirB
            my1.dat
    dirC
        my3.dat
```

Task # 02:

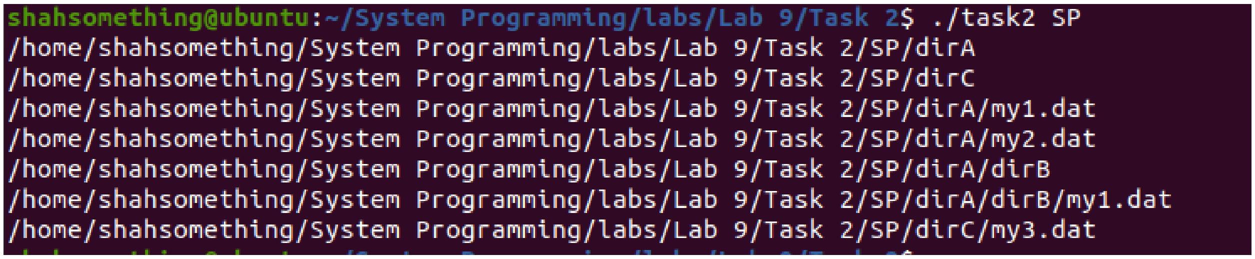
Traverse directory tree in breadth-first order.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <dirent.h>
#include <sys/stat.h>
#include <time.h>
#include <pwd.h>

void breadthfirst(char *name)
{
   DIR *dirp = opendir(name);
   if(dirp==NULL)
   {
      perror("Failed to open directory");
   }
   struct dirent *direntp;
   struct stat buffer;
   char cwd[400];
   int retchd = chdir(name);
   if(retchd==-1)
         perror("Failed to change directory");
   if(getcwd(cwd,sizeof(cwd))==NULL)
         perror("Failed to get cwd");
   while((direntp=readdir(dirp))!=NULL)
   {
         if(direntp->d_name[0]=='.')
                           continue;
```

```
            printf("%s/%s\n",cwd,direntp->d_name);
    }
    rewinddir(dirp);
    while((direntp=readdir(dirp))!=NULL)
    {
            if(direntp->d_name[0]=='.')
                                    continue;
            int ret = stat(direntp->d_name,&buffer);
            if(ret==-1)
                        perror("Error using stat");
            if(S_ISDIR(buffer.st_mode))
            {
                        breadthfirst(direntp->d_name);
                        chdir("..");
            }
    }
}

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

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 9/Task 2$ ./task2 SP
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirA
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirC
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirA/my1.dat
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirA/my2.dat
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirA/dirB
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirA/dirB/my1.dat
/home/shahsomething/System Programming/labs/Lab 9/Task 2/SP/dirC/my3.dat
```

Task # 03:

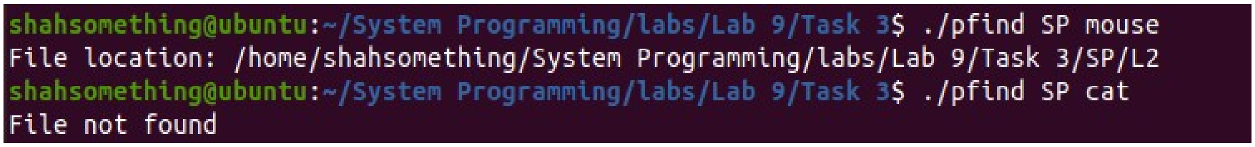
Implement the pfind utility

Code:

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

int found=0;

void pfind(char *dir, char *file)
{
   DIR *dirp = opendir(dir);
   if(dirp==NULL)
   {
      perror("Failed to open directory");
   }
   struct dirent *direntp;
   struct stat buffer;
   if(chdir(dir)==-1){
         perror("Unable to change directory");
   }
   char cwd[400];
   while((direntp=readdir(dirp))!=NULL)
   {
         int ret = stat(direntp->d_name,&buffer);
         if(ret==-1)
         {
                  perror("Stat Function Error");
         }
         if(direntp->d_name[0]=='.')
                  continue;
         if(strcmp(direntp->d_name,file)==0)
         {
                  if(getcwd(cwd,sizeof(cwd))==NULL)
```

```
            perror("Failed to get cwd");
        printf("File location: %s\n",cwd);
        found=1;
    }
    if(S_ISDIR(buffer.st_mode))
    {
        pfind(direntp->d_name,file);
        chdir("..");
    }
 }
}

int main(int argc, char *argv[])
{
    pfind(argv[1],argv[2]);
    if(!found)
        printf("File not found\n");
    return 0;
}
```

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 9/Task 3$ ./pfind SP mouse
File location: /home/shahsomething/System Programming/labs/Lab 9/Task 3/SP/L2
shahsomething@ubuntu:~/System Programming/labs/Lab 9/Task 3$ ./pfind SP cat
File not found
```