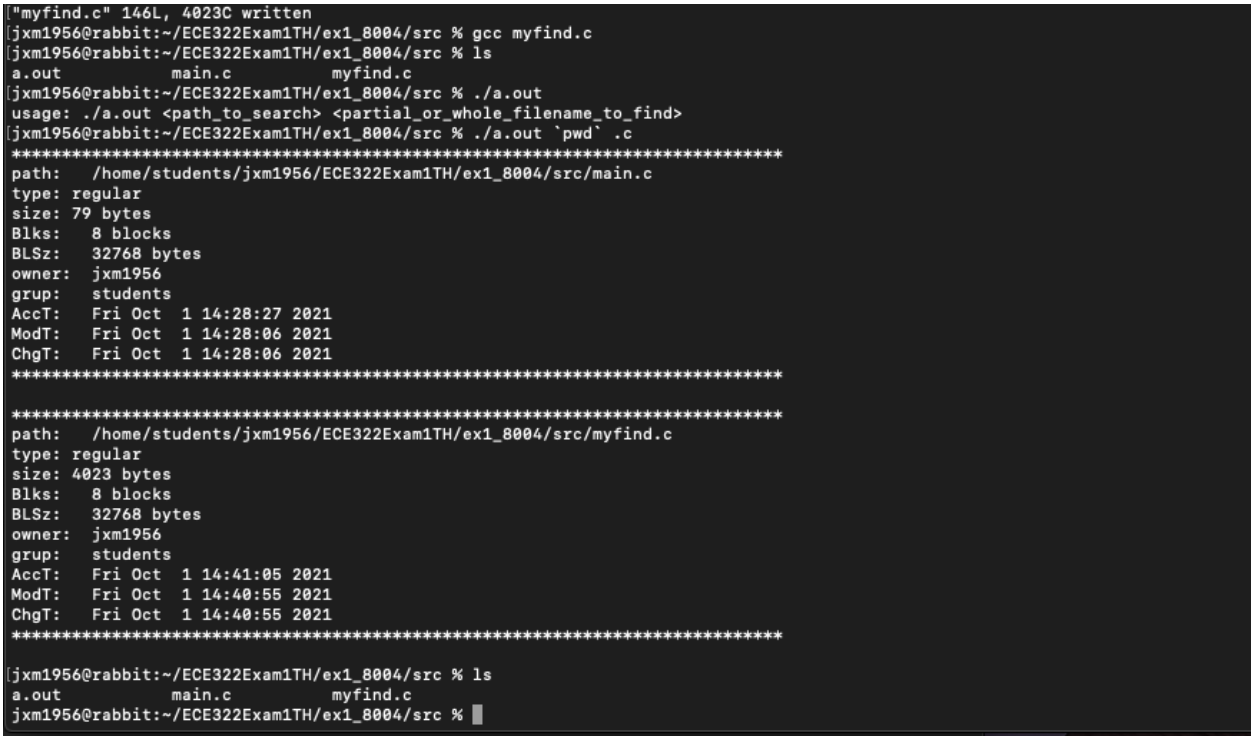
ECE 322: Systems Programing
Exam 1: Take Home

Jing Ma
C12108004
October 1, 2021

Screenshots:



This screenshot shows the output of our myfind.c working and running.

Code:

cr_project.csh

```
#!/bin/csh

#Phase 1: Setup your environment ?^`^s directory: ex1_<last_four_digits_of_your_C#>
#Command: cr_project <project_name>
#C12108004

#globals

set MKD=/bin/mkdir
set TOUCH=/usr/bin/touch
set CP=/bin/cp
set RM=/bin/rm
set PROJNAME=$argv[1]
set PROJPATH=/home/students/jxm1956/ECE322Exam1TH

echo "you entered ${#argv} arguments"

if (${#argv} < 1) then
        echo "usage: $0  <projectName>"
        exit
endif

#directory to store project
if (! -d $PROJPATH) then
        echo "$PROJPATH does not exist"
        $MKD $PROJPATH
else
```

```
    echo "$PROJPATH exits"
    echo  " "
endif

echo "Making $PROJNAME inside main directory: $PROJPATH"
$MKD $PROJPATH/$PROJNAME
echo "Creating bin and src direcotries in $PROJNAME"
$MKD $PROJPATH/$PROJNAME/bin
$MKD $PROJPATH/$PROJNAME/src
$TOUCH $PROJPATH/$PROJNAME/src/main.c
exit
```

build_project.csh

```
#!/bin/csh

#Phase 2: Create a compiling script
#Command: build_project <project_name>

#set globals

set GCC=/usr/local/bin/gcc
set GCCOPTS="-c"
set LD=/usr/local/bin/gcc
set LDOPTS="-o"
set PROJNAME=$argv[1]
set PROJPATH=/home/students/jxm1956/ECE322Exam1TH
set MV=/bin/mv
set STAT=/usr/bin/stat
set CP=/bin/cp
set SRCPATH=/home/students/jxm1956/ECE322Exam1TH/ex1_8004/src
set BINPATH=/home/students/jxm1956/ECE322Exam1TH/ex1_8004/bin

if(${#argv}<1) then
   echo "usage $0 <project_name>"
   exit
endif

cd $SRCPATH

foreach file(*.c)
   if(! -e ${file:r}.o) then
```

```
        $GCC $file -c
        if(! -e ${file:r}.o) then
            echo "$file did not compile"
            exit
        else
            echo "$file compiled"
            $LD ${file:r}.o -o $PROJNAME
            $CP $PROJNAME $BINPATH
        endif
    else
        if(`$STAT -f %m $file` >`$STAT -f %m ${file:r}.o` ) then
            $GCC -c $file
            if(! -e ${file:r}.o) then
                echo "$file did not compile"
                exit
            else
                echo "$file compiled"
                $LD ${file:r}.o -o  $PROJNAME
                $CP $PROJNAME $BINPATH
            endif
        else
            echo "$file cannot compile"
        endif
    endif
end
```

myfind.c

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

//Phase 3: Write a C Program to search for files

int printFInfo(char *path, char *fname) {

        char *filename = NULL;
        struct stat finfo;
        struct passwd *pw = NULL;
        struct group *gr = NULL;

        filename = (char *)malloc(sizeof(char)*strlen(path)+strlen(fname)+2);
        sprintf(filename,"%s/%s", path, fname);

        if (lstat(filename, &finfo) == 0){

        char *dname = (char *)malloc(sizeof(char)*strlen(fname)+2);
        char *ch = "regular";
        if (S_ISDIR(finfo.st_mode)){
```

```
    ch = "directory";
}
else if (S_ISREG(finfo.st_mode)){
    if ((finfo.st_mode & S_IXUSR) == S_IXUSR){

        ch = "executable";
    }
}
else if(S_ISLNK(finfo.st_mode)){

    ch= "link";
}
else if(S_ISFIFO(finfo.st_mode)){

    ch= "fifo";
}
sprintf(dname,"%s%c", fname, ch);

printf("********************************************************************************\n");
printf("path: ");
printf("\t%s", path);
printf("/");
printf("%s", fname);
printf("\n");
printf("type: ");
printf("%s", ch);
printf("\n");
printf("size: ");
printf("%ld ", finfo.st_size);
```

```
printf("bytes");
printf("\n");
printf("Blks: ");
printf("\t%ld", finfo.st_blocks);
printf(" blocks");
printf("\n");
printf("BLSz: ");
printf("\t%ld", finfo.st_blksize);
printf(" bytes");
printf("\n");
printf("owner: ");
pw = getpwuid(finfo.st_uid);
gr = getgrgid(finfo.st_gid);
printf("\t%s", pw->pw_name);
printf("\n");
printf("grup: ");
printf("\t%s", gr->gr_name);
printf("\n");
printf("AccT: ");
printf("\t%s", ctime(&finfo.st_atime));
printf("ModT: ");
printf("\t%s", ctime(&finfo.st_mtime));//for rabbit: st_mtim    ctime(&finfo.st_mtim.tv_sec)); //
printf("ChgT: ");
printf("\t%s", ctime(&finfo.st_ctime));
printf("*******************************************************************************\n\n");
//printf("%ld", finfo.st_mtime);
//printf("\n");
//printf("\n");
free(dname);
}
```

```
        free(filename);
        return 0;
}

int searching(char *path, char *keyword){
DIR *dirp=NULL;
struct dirent *d=NULL;
//char*path=strdup(folder);
dirp=opendir(path);
d = readdir(dirp);
   if(dirp==NULL){
        fprintf(stderr,"Bad path: %s\n", path);
        exit(0);
              return -1;
   }
   else{
        int counter = 0;
        while((d=readdir(dirp))!= NULL){
            char *filename = NULL;
            struct stat finfo;
            filename = (char *)malloc(sizeof(char)*strlen(path)+strlen(d->d_name)+2);

            char *temp=NULL;
            temp = (char *)malloc(sizeof(char)*strlen(path)+strlen(d->d_name)+2);
            sprintf(temp,"%s/%s", path, d->d_name);
            lstat(temp, &finfo);
            if(S_ISDIR(finfo.st_mode)){
                if(d->d_name[0]!='.'){
                    printf("%s \n",temp);
                    searching(temp,keyword);
```

```
            }
        }
        else{
            if(strstr(d->d_name,keyword)!=NULL){
                printFInfo(path,d->d_name);
            }
        }
    }
  }
return 0;
}

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

  char * path;
  char * filename;
  if(argc<3){
    //fprintf(stderr,"usage: %s <path_to_search> <filename>\n",argv[0]);
    fprintf(stderr,"usage: %s <path_to_search>
<partial_or_whole_filename_to_find>\n",argv[0]);
    exit(0);
    return -1;
  }
  path=strdup(argv[1]);
  filename=strdup(argv[2]);
  searching(path,filename);

  return 0;
}
```

```
            if(S_ISDIR(finfo.st_mode)){
                if(d->d_name[0]!='.'){
                    printf("%s \n",temp);
                    searching(temp,keyword);
                }
            }
            else{
                if(strstr(d->d_name,keyword)!=NULL){
                    printFInfo(path,d->d_name);
                }
            }
        }
    }
return 0;
}

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

    char * path;
    char * filename;
    if(argc<3){
        //fprintf(stderr,"usage: %s <path_to_search> <filename>\n",argv[0]);
        fprintf(stderr,"usage: %s <path_to_search>
<partial_or_whole_filename_to_find>\n",argv[0]);
        exit(0);
        return -1;
    }
    path=strdup(argv[1]);
    filename=strdup(argv[2]);
    searching(path,filename);
```

```
    return 0;
}
```

Conclusion

Overall, I learned a lot while doing this program. It was hard but interesting. I totally do not recommend doing this the night before it was due, because it does take some time to figure out. In the end my program did compile and work, and I'm very proud of that. But definitely for the next take home I will start earlier and finish it earlier.