Department of Computer Science and Engineering

S.G.Shivanirudh , 185001146, Semester IV

13 February 2020

UCS1411 - Operating Systems Laboratory

Exercise – 2- Simulation of system commands using system calls

Objective:

To develop a C program to implement the cp, ls, grep commands (with some options) using system calls.

Code:

Q1. To develop a C program to implement the cp command using system calls

```c
//Implementing cp command
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<fcntl.h>
#include<string.h>
#include<unistd.h>

```

```
void main (int argc,char *argv[]){
    if(argc<3)
        printf("\n Insufficient arguments \n");
    else{
        if(argc==3){
        //Non-interactive
            int sourcefd=open(argv[1],O_RDWR);
            //Non-existent source file
            if(sourcefd==-1){
                printf("\n Source file does not exist \n");
            }
            else{
                //Reading source file
                char *tmpline=(char*)calloc(1000,sizeof(char));
                int readfd=read(sourcefd,tmpline,100);

                tmpline[readfd]=’\0’;

                //Creating destination file if non-existent
                int destfd=open(argv[2],O_CREAT|O_RDWR);
                if(destfd==-1)
                    printf("\n Destination file could not be created \n");
                else{
                    write(destfd,tmpline,strlen(tmpline));
                    printf("\n Content copied successfully\n");
                    close(destfd);
                }
                close(sourcefd);
            }
        }
        else{
            //Interactive
            int sourcefd=open(argv[2],O_RDWR);
            //Non-existent source file
            if(sourcefd==-1){
                printf("\n Source file does not exist \n");
            }
            else{
                //Reading source file
                char *tmpline=(char*)calloc(1000,sizeof(char));
                int readfd=read(sourcefd,tmpline,100);

                tmpline[readfd]=’\0’;

                //Creating destination file if non-existent
                int destfd=open(argv[3],O_CREAT|O_RDWR);
                if(destfd==-1)
                    printf("\n Destination file could not be created \n");
                else{
                    char opt;
                    if(strcmp(argv[1],"-i")==0){
                        printf("\n Copy contents? y/n ");
                        scanf(" %c",&opt);
                        if(opt=='Y'||opt=='y'){
                            write(destfd,tmpline,strlen(tmpline));
                            printf("\n Content copied successfully\n");
                        }
                        else{
                            printf("\n Manual Abort. \n");
                        }
                    }
                    else{
                        write(destfd,tmpline,strlen(tmpline));
                        printf("\n Content copied successfully\n");
                    }
                    close(destfd);
                }
                close(sourcefd);
            }
        }
    }
}
```

Output:

```
Output:
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./mycp

 Insufficient arguments
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./mycp -i source.txt

 Source file does not exist
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./mycp -i source.txt dest.txt

 Copy contents? y/n y

 Content copied successfully
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./mycp source.txt dest.txt

 Content copied successfully
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ cat source.txt
shiva
sharvan
shashuuuu
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ cat dest.txt
shiva
sharvan
shashuuuu
```

Q2. To develop a C program to implement the ls command using system calls

```c
//Implementing ls command
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>
#include<unistd.h>
#include<dirent.h>

void Recurse_ls(char *dirname){
    DIR *dirstream=NULL;
    struct dirent *dirpointer=NULL;

    if(dirname[0]=='.'){
        getcwd(dirname,1000);
    }
    dirstream=opendir(dirname);
    if(dirstream==NULL){
        printf("\n Unable to open directory \n");
    }
    else{
        struct stat dirstat;
        while(NULL != (dirpointer=readdir(dirstream))){
            printf("%s\t",dirpointer->d_name);
        }
        printf("\n\n");
        dirpointer=NULL;
        while(NULL != (dirpointer=readdir(dirstream))){
            int dirfd=stat(dirpointer->d_name,&dirstat);
            if(dirfd<0)
                printf("\nUnable to locate file\n");
            else{
                int check_dir=S_ISDIR(dirstat.st_mode);
                if(check_dir){
                    Recurse_ls(dirpointer->d_name);
                }
            }
        }
    }
}
void main (int argc,char *argv[]){

    char *dirname=(char*)calloc(1000,sizeof(char));

    printf("\nEnter path of directory: ");
    scanf(" %s",dirname);
    if(argc==1){
        DIR *dirstream=NULL;
        struct dirent *dirpointer=NULL;

        if(dirname[0]==’.’){
            getcwd(dirname,1000);
        }
        dirstream=opendir(dirname);
        if(dirstream==NULL){
            printf("\n Unable to open directory \n");
        }
        else{
            while(NULL != (dirpointer=readdir(dirstream))){
                printf("\n %s",dirpointer->d_name);
            }
            printf("\n");

            closedir(dirstream);
        }
    }
    else if(strcmp(argv[1],"-l")==0){
        DIR *dirstream=NULL;
        struct dirent *dirpointer=NULL;

        if(dirname[0]==’.’){
            getcwd(dirname,1000);
        }
        dirstream=opendir(dirname);
        if(dirstream==NULL){
            printf("\n Unable to open directory \n");
        }
        else{
            struct stat dirstat;
            while(NULL != (dirpointer=readdir(dirstream))){

                int dirfd=stat(dirpointer->d_name,&dirstat);
                if(dirfd<0)
                    printf("\nUnable to locate directory\n");
                else{
                    printf((S_ISDIR(dirstat.st_mode))?"d":"-");
                    printf( (dirstat.st_mode & S_IRUSR) ? "r" : "-");
                    printf( (dirstat.st_mode & S_IWUSR) ? "w" : "-");
                    printf( (dirstat.st_mode & S_IXUSR) ? "x" : "-");
                    printf( (dirstat.st_mode & S_IRGRP) ? "r" : "-");
                    printf( (dirstat.st_mode & S_IWGRP) ? "w" : "-");
                    printf( (dirstat.st_mode & S_IXOTH) ? "x" : "-");
                    printf( (dirstat.st_mode & S_IXGRP) ? "x" : "-");
                    printf( (dirstat.st_mode & S_IROTH) ? "r" : "-");
                    printf( (dirstat.st_mode & S_IWOTH) ? "w" : "-");
                    printf(" %ld ",dirstat.st_nlink);
                    printf(" %6ld ",dirstat.st_size);
                    printf(" %8ld ",dirstat.st_ino);
                    printf(" %15s",dirpointer->d_name);
                }
                printf("\n");
            }
        }
    }
    else if(strcmp(argv[1],"-R")==0){
        Recurse_ls(dirname);
    }
}
```

Output:

```
Output:
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./myls

```

```
Enter path of directory: .

 mycp.c
 2-ex2-cmds.pdf
 mygrep.c
 .
 dest.txt
 CPUScheduling
 SYSTEM_CALLS_MANUAL.pdf
 a
 grep.c
 mycp
 myls
 ls
 mygrep
 myls.c
 source.txt
 ..
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./myls -l

Enter path of directory: .
-rw-rw--r- 1    2625   2363958           mycp.c
-rw-rw--r- 1   44937   3670881   2-ex2-cmds.pdf
-rw-r---r- 1    1358   2363973         mygrep.c
drwxr-xxr- 3    4096   2363810                .
-r--rw-xr- 1      24   2360766         dest.txt
drwxr-xxr- 2    4096   2368128    CPUScheduling
-rw-rw--r- 1   19896   3670880  SYSTEM_CALLS_MANUAL.pdf
-rwxrwxxr- 1   12920   2360763                a
-rw-r---r- 1    2506   2360758           grep.c
-rwxrwxxr- 1   12840   2360764             mycp
-rwxrwxxr- 1   12992   2360767             myls
-rwxrwxxr- 1   12992   2368119               ls
-rwxrwxxr- 1    8560   2363966           mygrep
-rw-r---r- 1    2864   2363972           myls.c
-rw-rw--r- 1      24   2363959       source.txt
drwxr-xxr- 6    4096   2363963               ..
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/SystemCalls$ ./myls -R

Enter path of directory: .
mycp.c  2-ex2-cmds.pdf  mygrep.c    .   dest.txt    CPUScheduling   SYSTEM_CALLS_MANUAL.pdf a   grep.c  mycp    myls    ls  mygrep  myls.c  source.txt  ..
```

Q3. To develop a C program to implement the grep command using system calls

```c
//Implementing grep command
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<fcntl.h>
#include<string.h>
#include<unistd.h>

int isSubstring(char* s1, char* s2)
{
    int M = strlen(s1);
    int N = strlen(s2);

    //Loop to slide pat[] one by one
    for (int i = 0; i <= N - M; i++) {
        int j;

        // For current index i, check for pattern match
        for (j = 0; j < M; j++)
            if (s2[i + j] != s1[j])
                break;

        if (j == M)
            return i;
    }

    return -1;
}

void main (int argc,char *argv[]){
    if(argc<2)
        printf("\n Insufficient arguments \n");
    else{

        int filefd;
        //printf("\n%d\n", argc);

        if(argc==3)
            filefd=open(argv[2],O_RDWR);
        else
            filefd=open(argv[3],O_RDWR);

        //Non-existent source file
        if(filefd==-1){
            printf("\n Source file does not exist \n");
        }
        else{
            //Reading source file
            char *tmpline=(char*)calloc(1000,sizeof(char));
            int readfd=read(filefd,tmpline,100);

            tmpline[readfd]='\0';

            char *lines[100];
            for(int i=0;i<100;i++){
                lines[i]=(char*)malloc(sizeof(100));
            }
            int lctr=0;
            char* token=strtok(tmpline,"\n");
            while(token!=NULL){
                strcpy(lines[lctr++],token);
                token=strtok(NULL,"\n");
            }

            if(argc==3){
                for(int i=0;i<lctr;i++){
                    if(isSubstring(argv[1],lines[i])!=-1)
                        printf("\n%s\n",lines[i]);
                }
            }
            else{
                if(strcmp(argv[1],"-c")==0){
                    int ctr=0;
                    for(int i=0;i<lctr;i++){
                        if(isSubstring(argv[2],lines[i])!=-1)
                            ctr++;
                    }
                    printf("\n%d\n",ctr);
                }
                else if(strcmp(argv[1],"-v")==0){
                    for(int i=0;i<lctr;i++){
                        if(isSubstring(argv[2],lines[i])==-1)
                            printf("\n%s\n",lines[i]);
                    }
                }
                else if(strcmp(argv[1],"-n")==0){
                    int ctr=0;
                    for(int i=0;i<lctr;i++){
                        if(isSubstring(argv[2],lines[i])!=-1)
    {
                            ctr++;
                            printf("\n%d %s\n",ctr,lines[i]);
                        }
                    }
                }
            }

```

```
            close(filefd);
        }
    }
}
```

Output:

```
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/
    SystemCalls$ ./a hi source.txt

shiva
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/
    SystemCalls$ ./a -c hi source.txt

1
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/
    SystemCalls$ ./a -v hi source.txt

sharvan

shashuuuu
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/
    SystemCalls$ ./a -n hi source.txt

1 shiva
shivanirudh@shiva-ideapad:~/Desktop/Semester4/OSLAB/
    SystemCalls$ cat source.txt
shiva
sharvan
shashuuuu
```