**SSN College of Engineering, Kalavakkam**

**Department of Computer Science and Engineering**

**Semester - IV**

**UCS1411 – Operating Syatems Lab**

**Academic Year: 2019-2020 Batch: 2018-2022**

**Name : Rahul Ram M**

**Class : CSE – B**

**Register Number : 185001121**

**Lab Exercise 2 Implementation of System calls**

**CODE:**

**mycp.c:**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <errno.h>

#include <unistd.h>

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

{

int src, dest, rd, wrt;

char c;

char \*buff[1024];

if(argc<3 || argc>4)

{

printf("Insufficient Arguments !!!\n");

exit(0);

}

if(argc==3)

{

src = open(argv[1], O\_RDONLY);

if(src == -1)

{

printf("\n Error opening source file");

exit(0);

}

dest = open(argv[2], O\_WRONLY | O\_CREAT);

if(dest == -1)

{

printf("\n Error opening Destination file");

exit(0);

}

while((rd = read(src, buff, 1024))>0)

{

if(write(dest, buff, rd)!=rd)

{

printf("Error in writing data");

}

}

if(rd == -1)

printf("\nError in reading data from %s\n",argv[1]);

if(close(src) == -1)

printf("\nError in closing file %s\n",argv[1]);

if(close(dest) == -1)

printf("\nError in closing file %s\n",argv[2]);

printf("FILE COPIED !!!\n");

}

else

{

if(argv[1][1]=='i')

{

src = open(argv[2], O\_RDONLY);

if(src == -1)

{

printf("\n Error opening source file");

exit(0);

}

dest=open(argv[3], O\_RDONLY);

if(dest != -1)

{

printf("Do you Want to overwrite ? (y/n) : ");

scanf("%c",&c);

if(c=='n')

exit(0);

}

dest = open(argv[3], O\_WRONLY | O\_CREAT );

if(dest == -1)

{

printf("\n Error opening Destination file");

exit(0);

}

while((rd = read(src, buff, 1024))>0)

{

if(write(dest, buff, rd)!=rd)

{

printf("Error in writing data");

}

}

if(rd == -1)

printf("\nError in reading data from %s\n",argv[2]);

if(close(src) == -1)

printf("\nError in closing file %s\n",argv[2]);

if(close(dest) == -1)

printf("\nError in closing file %s\n",argv[3]);

printf("FILE OVERWRITTEN !!!\n");

}

else if(argv[1][1]=='v')

{

src = open(argv[2], O\_RDONLY);

if(src == -1)

{

printf("\n Error opening source file");

exit(0);

}

dest = open(argv[3], O\_WRONLY | O\_CREAT);

if(dest == -1)

{

printf("\n Error opening Destination file");

exit(0);

}

while((rd = read(src, buff, 1024))>0)

{

if(write(dest, buff, rd)!=rd)

{

printf("Error in writing data");

}

}

if(rd == -1)

printf("\nError in reading data from %s\n",argv[2]);

if(close(src) == -1)

printf("\nError in closing file %s\n",argv[2]);

if(close(dest) == -1)

printf("\nError in closing file %s\n",argv[3]);

printf(" '%s' -> '%s' \n",argv[2],argv[3]);

}

else if(argv[1][1]=='f')

{

src = open(argv[2], O\_RDONLY);

if(src == -1)

{

printf("\n Error opening source file");

exit(0);

}

dest = open(argv[3], O\_WRONLY | O\_CREAT);

if(dest == -1)

{

remove(argv[3]);

}

dest = open(argv[3], O\_WRONLY | O\_CREAT);

while((rd = read(src, buff, 1024))>0)

{

if(write(dest, buff, rd)!=rd)

{

printf("Error in writing data");

}

}

if(rd == -1)

printf("\nError in reading data from %s\n",argv[2]);

if(close(src) == -1)

printf("\nError in closing file %s\n",argv[2]);

if(close(dest) == -1)

printf("\nError in closing file %s\n",argv[3]);

}

}

exit(EXIT\_SUCCESS);

}

**mygrep.c:**

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include<string.h>

#include <fcntl.h>

void match\_pattern(char \*argv[] ,int argc ,int op)

{

char argu[argc][100];

int flag=0;

for (int i=1;i<argc;i++)

{

strcpy(argu[i],argv[i]);

}

if(argv[1][0]!='-')

{

strcpy(argu[3],argu[2]);

strcpy(argu[2],argu[1]);

flag=0;

}

if(argv[1][1]=='i')

{

flag=1;

}

int fd,r,j=0,count=0,co=1;

char temp,line[100];

if((fd=open(argu[3],O\_RDONLY)) != -1)

{

while((r=read(fd,&temp,sizeof(char)))!= 0)

{

if(temp!='\n')

{

line[j]=temp;

j++;

}

else

{

line[j]='\0';

if(strstr(line,argu[2])!=NULL && op ==0)

{ printf("%s\n",line);

}

else if(strstr(line,argu[2])!=NULL && op ==4)

printf("%d - %s\n",co,line);

else if(strstr(line,argu[2])==NULL && op ==2)

printf("%s\n",line);

else if(strstr(line,argu[2])!=NULL && op ==3)

count++;

memset(line,0,sizeof(line));

j=0;

co++;

}

}

if(op==3)

printf("The given word occurs in %d lines\n",count);

}

}

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

{

struct stat stt;

if(argc==3)

{

if(stat(argv[2],&stt)==0)

match\_pattern(argv,argc,1);

else

{

perror("stat()");

exit(1);

}

}

else if(argc==4)

{

if(stat(argv[3],&stt)==0 && argv[1][0]=='-')

if(argv[1][1]=='v')

match\_pattern(argv,argc,2);

else if(argv[1][1]=='c')

match\_pattern(argv,argc,3);

else if(argv[1][1]=='n')

match\_pattern(argv,argc,4);

else

{

perror("stat()");

exit(1);

}

}

return 0;

}

**myls.c:**

#include <stdio.h>

#include <stdlib.h>

#include <dirent.h>

#include <errno.h>

#include <string.h>

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

{

struct dirent \*de;

char option='n';

DIR \*dr;

char name[50][100];

int n=0;

if(argc > 3)

{

printf(" Error! Incorrect number of Arguments Provided");

return 0;

}

if(argc==1)

dr = opendir(".");

if(argc==2)

dr = opendir(argv[1]);

if(argc==3)

{

dr = opendir(argv[2]);

option=argv[1][1];

}

if(dr == NULL)

{

printf("\n Could not open directory");

return 0;

}

while((de = readdir(dr)) != NULL)

{

if(option=='r')

{

strcpy(name[n],de->d\_name);

n++;

continue;

}

else if(option=='i')

printf("%ld\t", de->d\_ino);

//if(option=='s')

//printf("%d\t", de->d\_reclen);

//printf("%d\t", de->d\_type);

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

}

if(option=='r')

{

for(int i=n-1;i>=0;i--)

{

printf("%s \n", name[i]);

}

}

closedir(dr);

return 0;

}