**UNIX I/O**

**Lab no# 05**

****

**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**

**December** 25, 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Task 01:** Implement the ***cp*** command.

**Source code:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <string.h>

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

{

if(argc<3) //0 file is this program itself and 1,2 will passed as arguments. total=3

{

printf("Sorry! Required arguments are not provided:\n");

return -1;

}

int f1=open(argv[1],O\_RDONLY);

//open file 1 for reading. we can't create here. it must exist before.

//it will return a value greater than 2 because 0 1 and 2 files by default opened.

if(f1==-1)

{

perror("Sorry file 1 can't opened");

return -1;

}

Int f2=open(argv[2],O\_WRONLY|O\_CREAT|O\_TRUNC,S\_IRWXU|S\_IRWXG|S\_IRWXO);

//open file 2 for writing. if not exist create it first.

//it will return a value greater than 2 because 0 1 and 2 files by default opened.

if(f1==-1)

{

perror("Sorry file 2 can't opened");

return -1;

}

int bytesread;

char buff[50];

while(bytesread!=0)

//read-write will continue until buff reach to end of file i,e when reads become zero.

{

bytesread=read(f1,buff,sizeof(buff)); //read data from file 1 and store it in buff.

if(bytesread==-1)

{

perror("Sorry data can't read successfully");

return -1;

}

int byteswrite=write(f2,buff,bytesread); //write data in file 2 which stored in buff.

if(byteswrite==-1)

{

perror("Sorry data can't write successfully");

return -1;

}

}

int cf1=close(f1); //close file 1.

if(cf1==-1)

{

perror("Sorry file 1 can't closed succesfully");

return -1;

}

int cf2=close(f2); //close file 2.

if(cf2==-1)

{

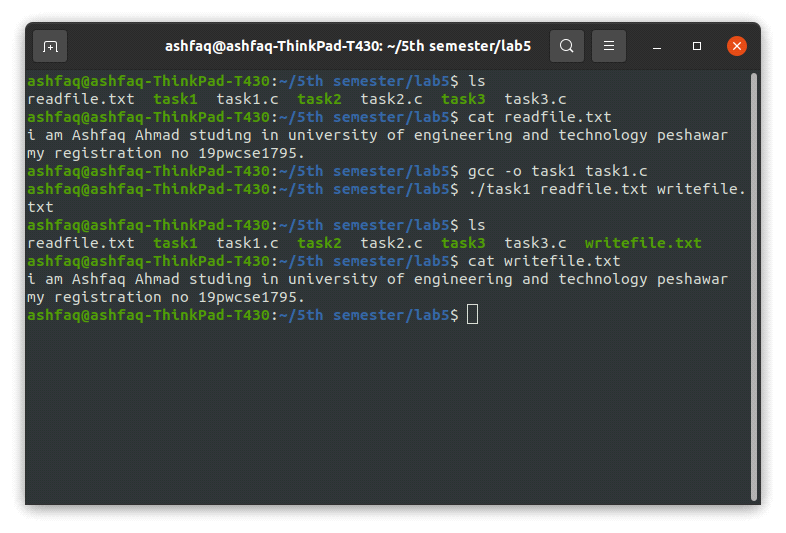
perror("Sorry file 2 can't close successfully");

return -1;

}

}

**Output:**



**Task02:** Implement ***rm*** command.

**Source Code:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

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

{

if(argc<2) //0 file is this program itself and 1 will passed as arguments. total=2

{

printf("Required Arguments not provided:\n");

return -1;

}

for (int i=1; i<argc; i++) //for loop is for if we have to remove more than one file.

{

int x=unlink(argv[i]); //it only remove file not a directory

if (x==0)

{

printf("File '%s' removed succesfully.\n",argv[i]);

}

else if(x==-1)

{

printf("Sorry! file '%s' can't removed succesfully.\n",argv[i]);

perror("Reason");

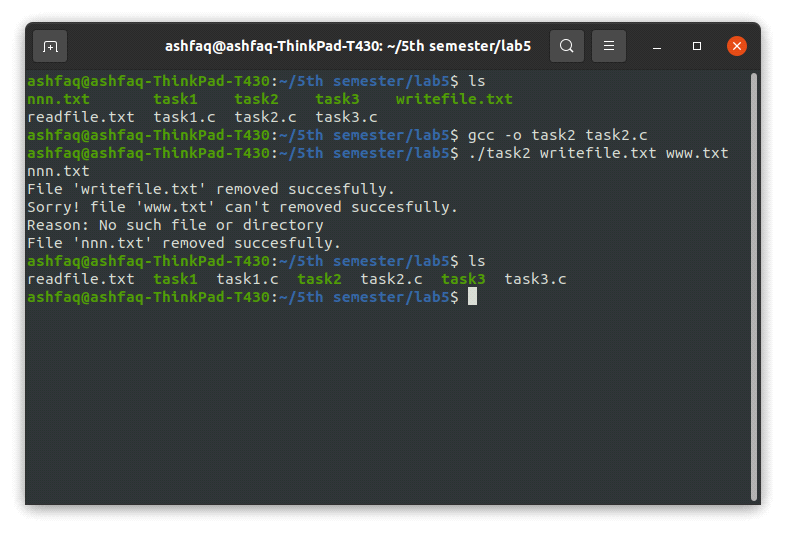
}

}

}

//unlink function return -1 in error case and 0 in success case.

**Output:**



**Task03:** Implement the ***mov*** command.

**Source Code:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <string.h>

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

{

if(argc<3) //0 file is this program itself and 1,2 will passed as arguments. total=3

{

printf("Sorry! Required arguments are not provided:\n");

return -1;

}

int f1=open(argv[1],O\_RDONLY);

//open file 1 for reading.we can't create here. it must exist before.

//it will return a value greater than 2 because 0 1 and 2 files by default opened.

if(f1==-1)

{

perror("Sorry file 1 can't opened");

return -1;

}

int f2=open(argv[2],O\_WRONLY|O\_CREAT|O\_TRUNC,S\_IRWXU|S\_IRWXG|S\_IRWXO); //open file 2 for writing. if not exist create it first.

//it will return a value greater than 2 because 0 1 and 2 files by default opened.

if(f1==-1)

{

perror("Sorry file 2 can't opened");

return -1;

}

int bytesread;

char buff[50];

do

//read-write will continue until buff reach to end of file i,e when reads become zero.

{

bytesread=read(f1,buff,sizeof(buff)); //read data from file 1 and store it in buff.

if(bytesread==-1)

{

perror("Sorry data can't read successfully");

return -1;

}

int byteswrite=write(f2,buff,bytesread); //write data in file 2 which stored in buff.

if(byteswrite==-1)

{

perror("Sorry data can't write successfully");

return -1;

}

}while (bytesread!=0);

int cf1=close(f1); //close file 1.

if(cf1==-1)

{

perror("Sorry file 1 can't closed succesfully");

return -1;

}

int x=unlink(argv[1]); //it only remove file not a directory

if (x==0)

{

printf("File '%s' Moved succesfully.\n",argv[1]);

}

else if(x==-1)

{

printf("Sorry! file '%s' can't Moved succesfully.\n",argv[1]);

perror("Reason");

}

int cf2=close(f2); //close file 2.

if(cf2==-1)

{

perror("Sorry file 2 can't close successfully");

return -1;

}

}

**Output:**

