UNIX I/O

LAB # 05

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

Sunday, December 27th, 2020

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objective(s):

- Understand and implement read, write, open, close and unlink function calls.

Task # 01:

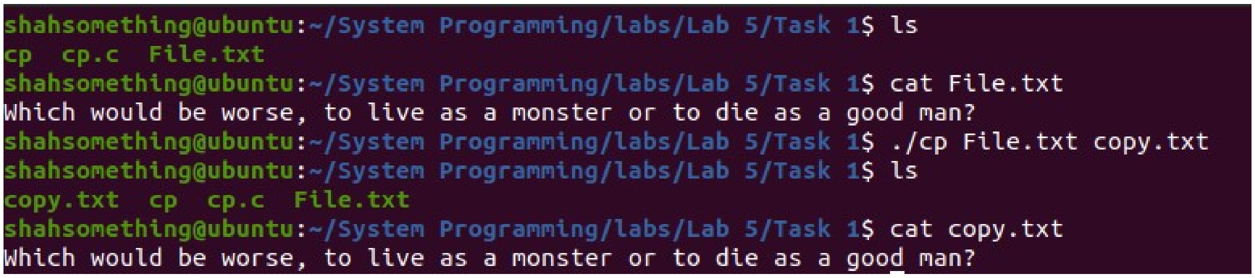
Implement the cp command.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>

int main(int argc, char *argv[])
{
	if(argc<3)
	{
		printf("Required Arguments not Provided\n");
		return -1;
	}
	int fd1 = open(argv[1],O_RDONLY);
	if(fd1==-1)
	{
		perror("Failed to open Source file");
		return -1;
	}

	int fd2 = open(argv[2],O_WRONLY | O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO);
	if(fd2==-1)
	{
		perror("Failed to open destination file");
		return -1;
	}

	int bytesread;
	char buffer[100];
	do
	{
		bytesread = read(fd1,buffer,100);
		if(bytesread==-1)
		{
			perror("Error Occured while reading");
			return -1;
		}
		int byteswriten = write(fd2,buffer,bytesread);
```

```
        if(byteswriten==-1)
        {
            perror("Error Occured while writing");
            return -1;
        }
    }while(bytesread!=0);

    int cfd1 = close(fd1);
    if(cfd1==-1)
    {
        perror("Failed to close Source file");
        return -1;
    }
    int cfd2 = close(fd2);
    if(cfd2==-1)
    {
        perror("Failed to close destination file");
        return -1;
    }
    return 0;
}
```

Output/Results:

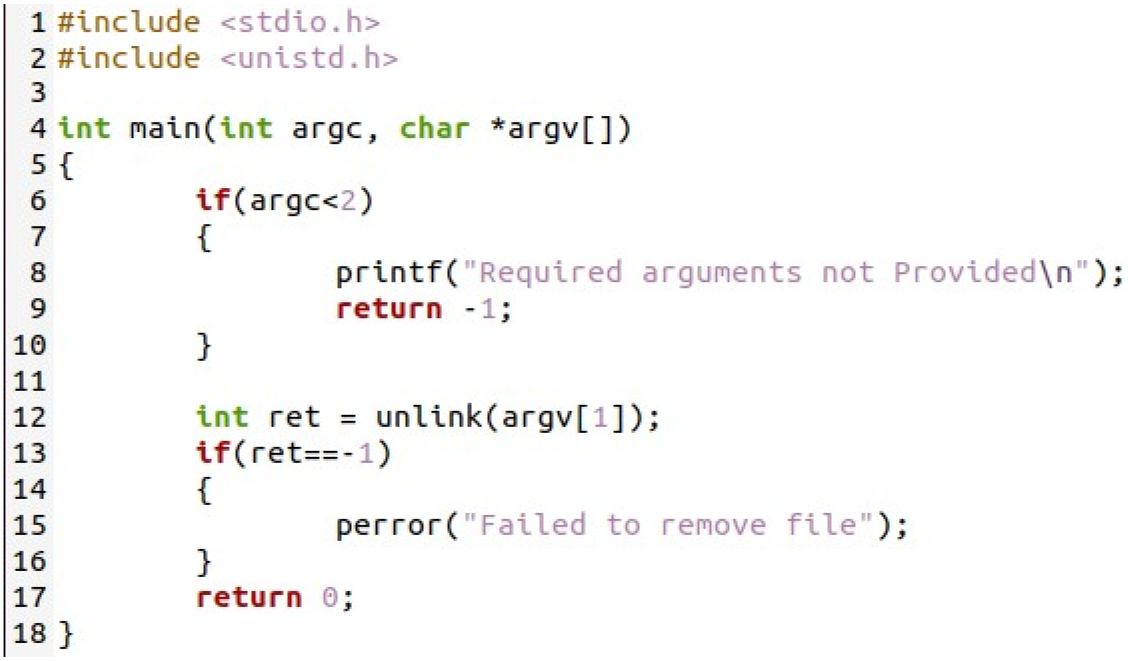


```
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 1$ ls
cp  cp.c  File.txt
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 1$ cat File.txt
Which would be worse, to live as a monster or to die as a good man?
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 1$ ./cp File.txt copy.txt
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 1$ ls
copy.txt  cp  cp.c  File.txt
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 1$ cat copy.txt
Which would be worse, to live as a monster or to die as a good man?
```

Task # 02:

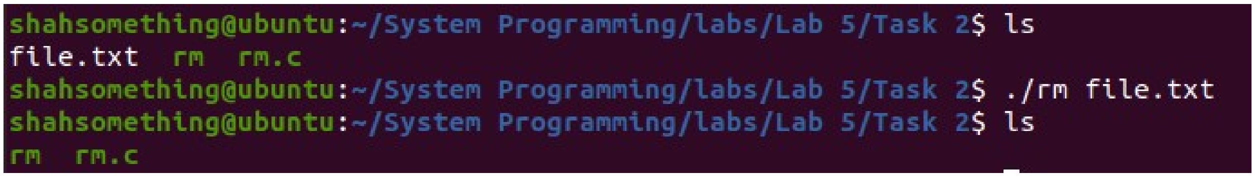
Implement rm command.

Code:



```c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
        if(argc<2)
        {
                printf("Required arguments not Provided\n");
                return -1;
        }

        int ret = unlink(argv[1]);
        if(ret==-1)
        {
                perror("Failed to remove file");
        }
        return 0;
}
```

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 2$ ls
file.txt  rm  rm.c
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 2$ ./rm file.txt
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 2$ ls
rm  rm.c
```

Task 3:

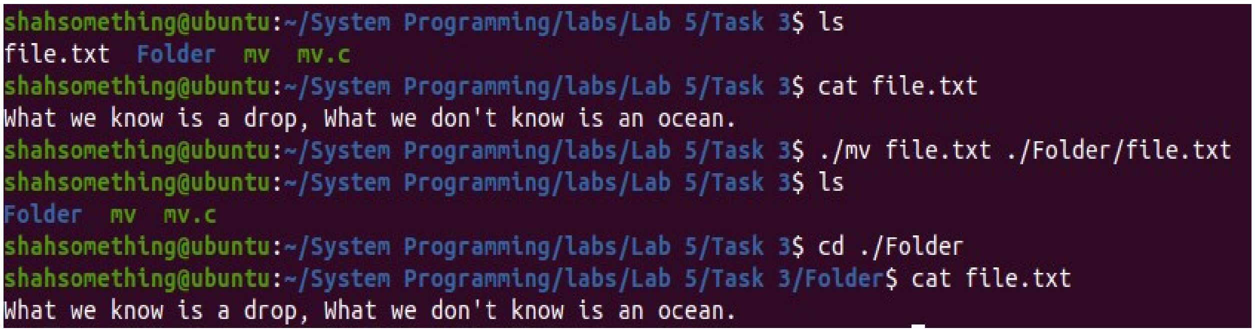
Implement the mv command.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>

int main(int argc, char *argv[])
{
	if(argc<3)
	{
		printf("Required arguments not provided\n");
		return -1;
	}

	int fd1 = open(argv[1],O_RDONLY);
	if(fd1==-1)
	{
		perror("Failed to open Source file");
		return -1;
	}

	int fd2 = open(argv[2],O_WRONLY | O_CREAT,S_IRWXU | S_IRWXG | S_IRWXO);
	if(fd2==-1)
	{
		perror("Failed to open destination file");
		return -1;
	}

	int bytesread;
	char buffer[100];
	do
	{
		bytesread = read(fd1,buffer,100);
		if(bytesread==-1)
		{
			perror("Error Occured while reading");
			return -1;
		}
		int byteswriten = write(fd2,buffer,bytesread);
```

```
                if(byteswriten==-1)
                {
                        perror("Error Occured while writing");
                        return -1;
                }
        }while(bytesread!=0);

        int cfd1 = close(fd1);
        if(cfd1==-1)
        {
                perror("Failed to close Source file");
        }

        int ret = unlink(argv[1]);
        if(ret==-1)
        {
                perror("Failed to remove Source file");
                return -1;
        }
        int cfd2 = close(fd2);
        if(cfd2==-1)
        {
                perror("Failed to close destination file");
        }
        return 0;
}
```

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3$ ls
file.txt  Folder  mv  mv.c
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3$ cat file.txt
What we know is a drop, What we don't know is an ocean.
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3$ ./mv file.txt ./Folder/file.txt
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3$ ls
Folder  mv  mv.c
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3$ cd ./Folder
shahsomething@ubuntu:~/System Programming/labs/Lab 5/Task 3/Folder$ cat file.txt
What we know is a drop, What we don't know is an ocean.
```