UNIX INPUT / OUTPUT

PART - 1

LAB # 05

Fall 2023

CSE-302L

Systems Programming Lab

Submitted by: **AIMAL KHAN**

Registration No.: **21PWCSE1996**

Class Section: **A**

"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. Abdullah Hamid

Sunday, January 28, 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

CSE 302L: SYSTEMS PROGRAMMING LAB

LAB ASSESSMENT RUBRICS

Criteria & Point Assigned	Outstanding 2	Acceptable 1.5	Considerable 1	Below Expectations 0.5	Score
Attendance and Attentiveness in Lab PLO08	Attended in proper Time and attentive in Lab	Attended in proper Time but not attentive in Lab	Attended late but attentive in Lab	Attended late not attentive in Lab	
Capability of writing Program/Algorithm/Drawing Flow Chart PLO1, PLO2, PLO3, PLO5	Right attempt/ no errors and well formatted	Right attempt/ no errors but not well formatted	Right attempt/ minor errors and not well formatted	Wrong attempt	
Result or Output/ Completion of target in Lab PLO9	100% target has been completed and well formatted.	75% target has been completed and well formatted.	50% target has been completed but not well formatted.	None of the outputs are correct.	
Overall, Knowledge PLO10,	Demonstrates excellent knowledge of lab	Demonstrates good knowledge of lab	Has partial idea about the Lab and procedure followed	Has poor idea about the Lab and procedure followed	
Attention to Lab Report PLO4,	Submission of Lab Report in Proper Time i.e., in next day of lab, with proper documentation.	Submission of Lab Report in proper time but not with proper documentation.	Late Submission with proper documentation.	Late Submission very poor documentation	

Instructor:

Name: ___________________________	Signature: __________________________

Unix I/O Part 1

Objectives:

Learn about these systems calls

- Read
- Write
- Close
- Open

Tasks:

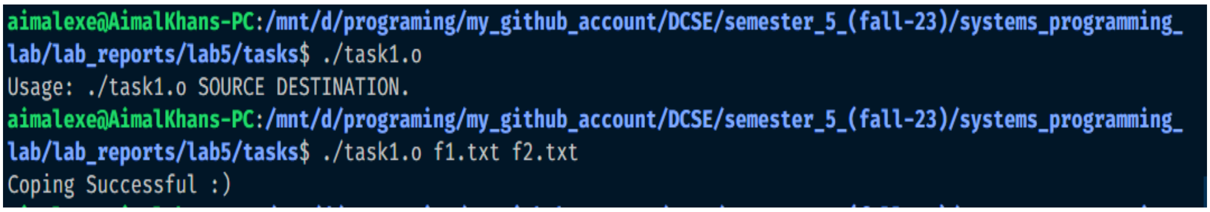
Task 1: Implement the **cp** command.

Code in C:

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>
#include <sys/stat.h>
#include "../../reusable_code_snippets/readWrite.h"

int main(int argc, char *argv[])
{
    if (argc != 3)
    {
        fprintf(stderr, "Usage: %s SOURCE DESTINATION.\n", argv[0]);
        return 1;
    }

    // 1. open sourc file (for reading only)
    int srcFile = open(argv[1], O_RDONLY);
    if (srcFile == -1)
    {
        fprintf(stderr, "Something went wrong while opening the
source file: %s due to %s\n", argv[1], strerror(errno));
        return 1;
    }

    // 2. open destination file if not present create it (for
writing only)
    int destFile = open(argv[2], O_WRONLY | O_CREAT | O_APPEND,
S_IRWXU | S_IRWXG | S_IRWXO);
    if (destFile == -1)
    {
        fprintf(stderr, "Something went wrong while opening the
destination file: %s due to %s\n", argv[2], strerror(errno));
```

```
        return 1;
    }

    // 3. Read from src file
    // 4. write to destination from src file
    if (readWriteOnly(&srcFile, &destFile) < 0)
    {
        perror("Something went wrong while reading or writing a
file.\n");
        return 1;
    }

    // 5. Close destination and src files
    int closeSrc = close(srcFile);
    int closeDest = close(destFile);
    if (closeSrc == -1 || closeDest == -1)
    {
        fprintf(stderr, "Error while closing the file. %s\n",
strerror(errno));
        return 1;
    }

    printf("Coping Successful :)\n");

    return 0;
}
```

Output:



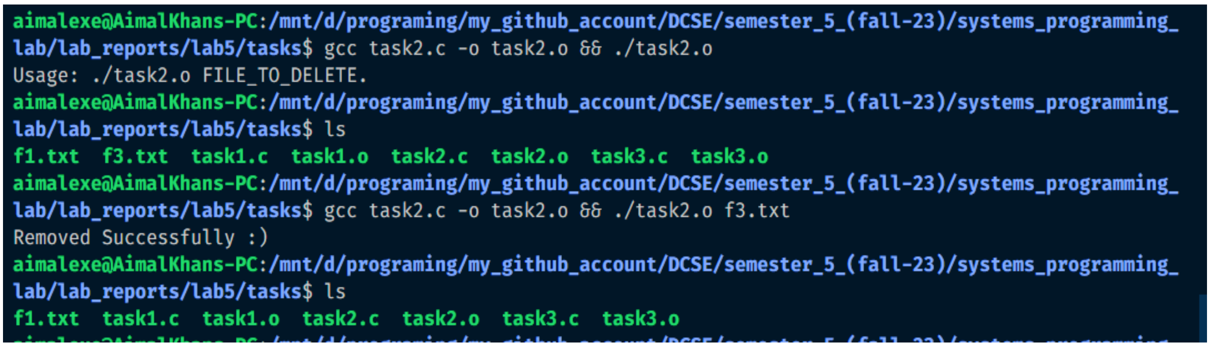
```
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ ./task1.o
Usage: ./task1.o SOURCE DESTINATION.
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ ./task1.o f1.txt f2.txt
Coping Successful :)
```

Task 2: Implement the **rm** command.

Code in C:

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

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        fprintf(stderr, "Usage: %s FILE_TO_DELETE.\n", argv[0]);
        return 1;
    }

    int deleteFile = unlink(argv[1]);
    if (deleteFile == -1)
    {
        fprintf(stderr, "Something went wrong while deleting the
file: %s due to %s\n", argv[1], strerror(errno));
        return 1;
    }
    printf("Removed Successfully :)\n");

    return 0;
}
```

Output:



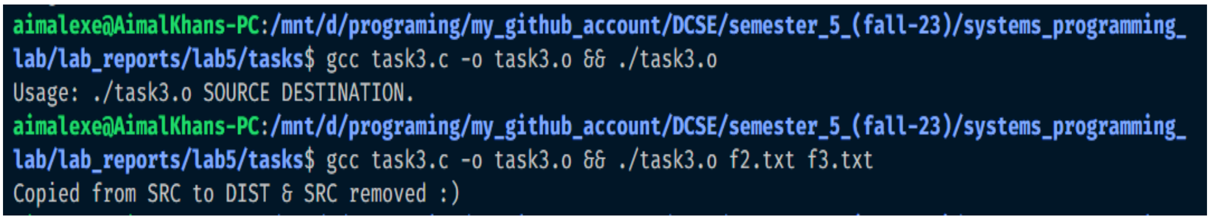
```
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ gcc task2.c -o task2.o && ./task2.o
Usage: ./task2.o FILE_TO_DELETE.
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ ls
f1.txt  f3.txt  task1.c  task1.o  task2.c  task2.o  task3.c  task3.o
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ gcc task2.c -o task2.o && ./task2.o f3.txt
Removed Successfully :)
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ ls
f1.txt  task1.c  task1.o  task2.c  task2.o  task3.c  task3.o
```

Task 3: Implement the **mv** command.

Code in C:

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>
#include <sys/stat.h>
#include "../../reusable_code_snippets/readWrite.h"

int main(int argc, char *argv[])
{
    if (argc != 3)
    {
        fprintf(stderr, "Usage: %s SOURCE DESTINATION.\n", argv[0]);
        return 1;
    }

    if (readWrite(argv[1], argv[2]) < 0)
    {
        perror("Something went wrong while reading or writing a
file.\n");
        return 1;
    }
    // 6. delete the source file
    int removeSrc = unlink(argv[1]);
    if (removeSrc == -1)
    {
        fprintf(stderr, "Something went wrong while deleting the
file: %s due to %s\n", argv[1], strerror(errno));
        return 1;
    }

    printf("Copied from SRC to DIST & SRC removed :)\n");

    return 0;
}
```

Output:



```
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ gcc task3.c -o task3.o && ./task3.o
Usage: ./task3.o SOURCE DESTINATION.
aimalexe@AimalKhans-PC:/mnt/d/programing/my_github_account/DCSE/semester_5_(fall-23)/systems_programming_
lab/lab_reports/lab5/tasks$ gcc task3.c -o task3.o && ./task3.o f2.txt f3.txt
Copied from SRC to DIST & SRC removed :)
```

Reference:

To view my codes, please refer to my GitHub account: https://github.com/aimalexe/DCSE/tree/main/semester_5_(fall-23)/systems_programming_lab/lab_reports .

Conclusion:

In conclusion, I have learned in depth about these system calls like read, open, close, write, buffers, rm, cp mv and much more. Now I am able to use these in future projects.

The End.