**UNIX INPUT / OUTPUT**

**PART – 2**

**LAB # 0****6**

**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 2**

Objectives:

Learn about these systems concepts

* Redirections
* Pipelining
* Cat utility
* Multi processes

Tasks:

**Task 1**: Write a program for parallel file copying using multiple processes. (First check if the no of files entered are even and if yes then create a child process for each pair and read from one and write to other)

**Code in C:**

#*include* <stdlib.h>

#*include* <stdio.h>

#*include* <unistd.h>

#*include* <sys/types.h>

#*include* <sys/wait.h>

#*include* "../../reusable\_code\_snippets/readWrite.h"

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

{

*if* (argc < 3)

    {

*fprintf*(stderr, "Usage: %s FILE\_1 FILE\_2 [FILE\_N...]\n", argv[0]);

*return* 1;

    }

*if* (argc % 2 == 0)

    {

*perror*("Argument must be even in count.\n");

*return* 1;

    }

    pid\_t processId;

*for* (int i = 1; i < argc; i += 2)

    {

*if* ((processId = *fork*()) < 0)

        {

*perror*("Process Creation Failed!!\n");

*return* 1;

        }

*else* *if* (processId == 0)

        {

*if* (*readWrite*(argv[i], argv[i + 1]) < 0)

            {

*perror*("Something went wrong while reading or writing a file.\n");

*return* 1;

            }

*printf*("Copied from SRC: %s to DIST: %s.\n", argv[i], argv[i + 1]);

*return* 0;

        }

*else*

*wait*(NULL);

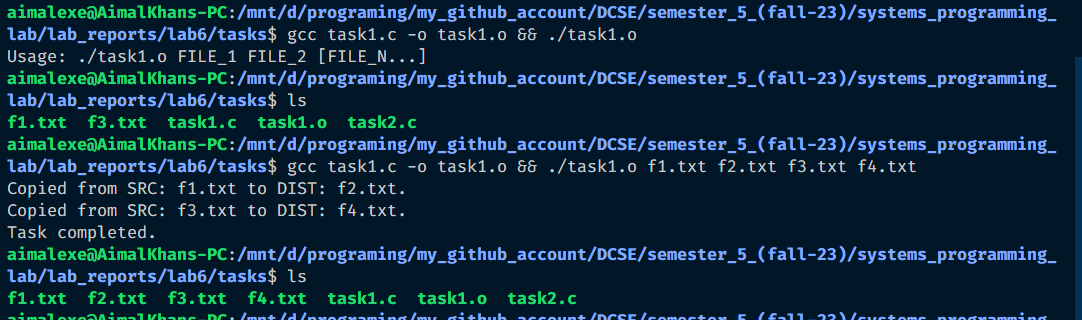
    }

*printf*("Task completed.\n");

*return* 0;

}

**Output:**

****

**Task 2**: Implement “**Cat**” utility.

**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*[]*)

{

    int readWriteReturnValue, keyboard = *STDIN\_FILENO*, screen = *STDOUT\_FILENO*, file;

*if* (argc == 1)

    {

        // *for 'cat' read from stdin and print on stdout.*

        readWriteReturnValue = *readWriteOnly*(&keyboard, &screen);

*if* (readWriteReturnValue < 0)

        {

*perror*("Something went wrong while reading from stdin or writing to stdout.\n");

*return* 1;

        }

    }

*else* *if* (argc == 2)

    {// *for 'cat file' read from file and write to stdout.*

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

*if* (file == -1)

        {

*fprintf*(*stderr*, "Something went wrong while opening the source file: %s due to %s\n", argv[1], *strerror*(*errno*));

*return* -1;

        }

        readWriteReturnValue = *readWriteOnly*(&file, &screen);

*if* (readWriteReturnValue < 0)

        {

*perror*("Something went wrong while reading from stdin or writing to stdout.\n");

*return* 1;

        }

*if* (*close*(file) == -1)

        {

*fprintf*(*stderr*, "Error while closing the file. %s\n", *strerror*(*errno*));

*return* 1;

        }

    }

*else* *if* (argc == 3)

    {// *for 'cat > file' read from stdin and write to file.*

        file = *open*(argv[3], *O\_CREAT* | *O\_TRUNC*, *S\_IRWXG* | *S\_IRWXO* | *S\_IRWXU*);

*if* (file == -1)

        {

*fprintf*(*stderr*, "Something went wrong while opening the source file: %s due to %s\n", argv[1], *strerror*(*errno*));

*return* -1;

        }

        readWriteReturnValue = *readWriteOnly*(&keyboard, &file);

*if* (readWriteReturnValue < 0)

        {

*perror*("Something went wrong while reading from stdin or writing to stdout.\n");

*return* 1;

        }

*if* (*close*(file) == -1)

        {

*fprintf*(*stderr*, "Error while closing the file. %s\n", *strerror*(*errno*));

*return* 1;

        }

    }

*else* *if* (argc == 4)

    {// *for 'cat file1 > file2' read from file1 and write to file2.*

        readWriteReturnValue = *readWrite*(argv[2], argv[4]);

*if* (readWriteReturnValue < 0)

        {

*perror*("Something went wrong while reading from stdin or writing to stdout.\n");

*return* 1;

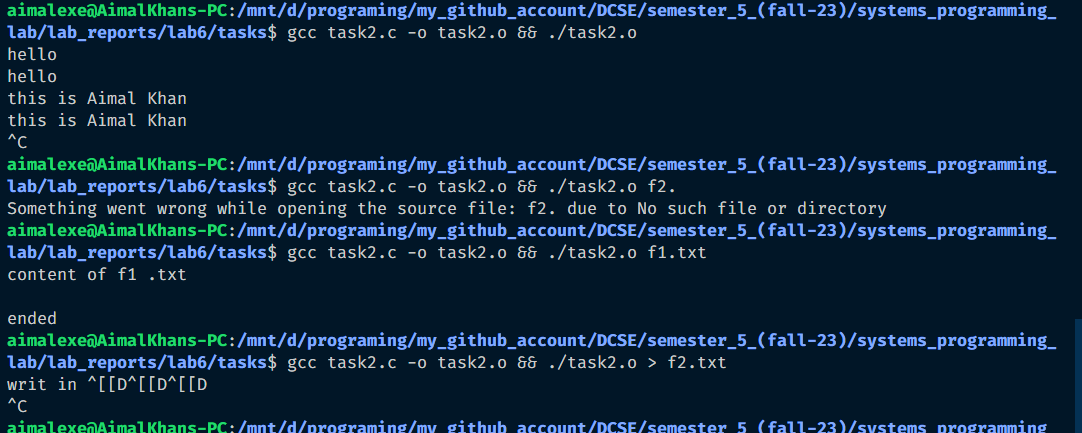
        }

    }

*return* 0;

}

**Output:**

****

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](%20https://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, cat, redirections, pipelining and much more. Now I am able to use these in future projects.

The End.