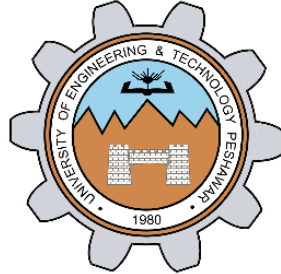


LAB NO 06
UNIX I/O (CONTINUED)



Fall 2024
CSE-302L Systems Programming Lab

Submitted by:

Name: **Hassan Zaib Jadoon**

Reg no: **22PWCSE2144**

Class Section: **A**

Signature: _____

Submitted to:

Engr. Abdullah Hamid

December 29, 2024

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

LAB NO 06 UNIX I/O (CONTINUED)

Task 1

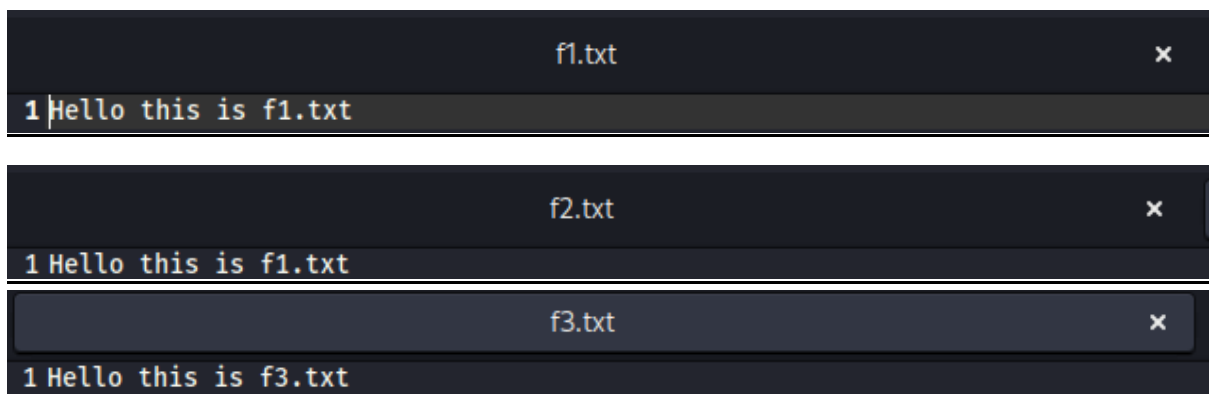
Write a program for parallel file copying using multiple processes. (First check if the no of files entered are even and if yes then create a child process for each pair and read from one and write to other)

Code:

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>
#include <string.h>
int main(int argc, char *argv[]) {
    if ((argc - 1) % 2 != 0) {
        perror("The number of files must be even.\n");
        return -1;
    }
    int processId;
    for (int i = 1; i < argc; i += 2) {
        processId = fork();
        if (processId < 0) {
            perror("Process creation failed");
            return -1;
        } else if (processId == 0) {
            int srcFd, destFd;
            char buffer[1024];
            int bytesRead, bytesWritten;
            srcFd = open(argv[i], O_RDONLY);
            destFd = open(argv[i + 1], O_WRONLY | O_CREAT | O_TRUNC, 0644);
            while ((bytesRead = read(srcFd, buffer, sizeof(buffer))) > 0) {
                bytesWritten = write(destFd, buffer, bytesRead);
            }
            close(srcFd);
            close(destFd);
            exit(0);
        }
    }
    for (int i = 1; i < argc; i += 2) {
        wait(NULL);
    }
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ nano task1.c
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ gcc task1.c -o task1.o
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ ./task1.o f1.txt f2.txt f3.txt f4.txt
```



```
f4.txt
1Hello this is f3.txt
```

Task 2

Implement the “Cat” command utility.

Code:

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

int main(int argc, char const *argv[]) {
    int fd1, fd2, br, bw;
    char buffer[1024];
    if (argc == 2) {
        fd1 = open(argv[1], O_RDONLY);
        dup2(fd1, STDIN_FILENO);
        close(fd1);
    } else if (argc == 3) {
        fd1 = open(argv[2], O_WRONLY | O_CREAT | O_APPEND, S_IRWXU|S_IRWXG|S_IRWXO);
        dup2(fd1, STDOUT_FILENO);
        close(fd1);
    } else if (argc == 4) {
        fd1 = open(argv[1], O_RDONLY);
        dup2(fd1, STDIN_FILENO);
        close(fd1);
        fd2 = open(argv[3], O_WRONLY | O_CREAT | O_APPEND, S_IRWXU|S_IRWXG|S_IRWXO);
        dup2(fd2, STDOUT_FILENO);
        close(fd2);
    }
    while ((br = read(STDIN_FILENO, buffer, sizeof(buffer))) > 0) {
        bw = write(STDOUT_FILENO, buffer, br);
    }
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ gcc task2.c -o task2.o
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ ./task2.o
^C
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ ./task2.o ">" f1.txt
^X^C
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 06$ ./task2.o f1.txt
Hello this is f1.txt
Hello i am writing to f1.txt
```

```
f1.txt
1Hello this is f1.txt
2Hello i am writing to f1.txt
```

```
dest.txt
1Hello this is my source file and i am copying its data.
```

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: _____