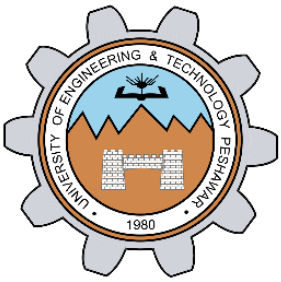
LAB NO 10

INTER PROCESS COMMUNICATION



Fall 2024

CSE-302L Systems Programming Lab

Submitted by:

Name: **Nouman Khan**

Reg no**: 22PWCSE2107**

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 10 INTER PROCESS COMMUNICATION

Task 1

A program in which a child writes a string to a pipe and the parent reads the string.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
#include <fcntl.h>

int main() {
    int pipefd[2];
    int pid;
    char message[] = "Hello from child!";
    char buff[256];
    pipe(pipefd);
    pid = fork();
    if (pid == 0) {
         close(pipefd[0]);
        write(pipefd[1], message, strlen(message) + 1);
        close(pipefd[1]);
        exit(0);
    } else {
        close(pipefd[1]);
        read(pipefd[0], buff, sizeof(buff));
        printf("Parent received message: %s\n", buff);
        close(pipefd[0]);
        wait(NULL);
    }
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ ls
task1.c  task1.o  task2.c  task2.o  task3
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ nano task1.c
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ gcc task1.c -o task1.o
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ ./task1.o
Parent received message: Hello from child!
```

Task 2

Write a program that creates a process fan. Parent process writes to the pipe and all the child processes read the message from pipe and display it on stdout.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
#include <fcntl.h>

int main() {
    int pipefd[2];
    int pid;
    char message[] = "Hello from child!";
    char buff[256];
    pipe(pipefd);
    pid = fork();
    if (pid == 0) {
         close(pipefd[0]);
        write(pipefd[1], message, strlen(message) + 1);
        close(pipefd[1]);
        exit(0);
    } else {
        close(pipefd[1]);
        read(pipefd[0], buff, sizeof(buff));
        printf("Parent received message: %s\n", buff);
        close(pipefd[0]);
        wait(NULL);
    }
    return 0;
}
```

Output:

```
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ nano task2.c
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ gcc task2.c -o task2.o
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10$ ./task2.o
child 1 message is Hello World
child 2 message is Hello World
child 3 message is Hello World
```

Task 3:

Write a Chatting application in which two processes can communicate using FIFO. Your program should satisfy the following specifications.

The program should take the name of FIFO, will create the FIFO (if not created yet) and should open it for reading and writing. Program should take input from standard input and write it to FIFO and should read from FIFO and write to standard output in another process. Both reading and writing shall be done concurrently.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/select.h>
#include <sys/types.h>
#include <sys/stat.h>
int main() {
    char buff[100];
    int fifofd;
    fd_set readset;
    struct timeval mytime;
    mytime.tv_sec = 2;
mytime.tv_usec =2;
    mkfifo("fifo", 0777);
    fifofd = open("fifo", O_RDWR);
    while (1) {
        FD_ZERO(&readset);
        FD_SET(STDIN_FILENO, &readset);
        FD_SET(fifofd, &readset);
        int maxfd = fifofd > STDIN_FILENO ? fifofd : STDIN_FILENO;
        select(maxfd + 1, &readset, NULL, NULL, &mytime);

        if (FD_ISSET(STDIN_FILENO, &readset)) {
            int br = read(STDIN_FILENO, buff, 100);
            write(fifofd, buff, br);
        }
        if (FD_ISSET(fifofd, &readset)) {
            int br = read(fifofd, buff, 100);
            write(STDOUT_FILENO, buff, br);
        }
    }
    close(fifofd);
    return 0;}
```

Output:

```
hassan@hassan-HP-ProBook-4740s: ~/Desktop/SP Lab/SP Lab 10/task3
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10/task3$ gcc task3.c -o
task3.o
hassan@hassan-HP-ProBook-4740s:~/Desktop/SP Lab/SP Lab 10/task3$ ./task3.o
Hello
Hello
Hassan Here
Hassan Here
```

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: _____