Lab 04: Inter-process communication - Pipes

Use of pipe(), fork(), exit() and if required open(), close(), read(), write()

In this lab students will learn about processes communicate among themselves. There are many mechanisms through which the processes communicate and in this lab we will discuss one such mechanism: Pipes. A pipe is used for one-way communication of a stream of bytes. In this lab we will learn how to create pipes and how processes communicate by reading or writing to the pipe.

Helpful documentation:
$man pipe
$man fork
$man open
$man close
$man read
$man write

<u>Tasks</u>

T1: Create a pipe between a parent process and a child process so that the child should write data to the pipe and parent should read the data from the pipe.
Logic flow

1. **Pipe creation**
2. **Process cloning**
3. **Closing the descriptor not used by each process**
4. **Read and write operation at pipe's ends**

T2: Write a C program that creates a pipe, write to the pipe, and then read from the pipe. You have to send a string to the child process from parent process and the child should return whether the string is a palindrome or not in the form of "Yes" or "Not a Palindrome". You do everything all within the same process.
You can use string functions or you can write all functions by yourself.

<u>Submission& Evaluation</u>

1. All the working against every task should be shown to the instructor within lab.
2. This is an individual assignment and both tasks carry equal marks.