# CSC 305

Prof. Jonathan Parziale

**Lab. Assignment # 5.**

***Inter-process Communication Using Pipes.***

Write a c++ program which implements inter-process communication between parent and child processes.

Note: Make sure to include the c library unistd.h is required for most of the functions

**Goals:**

* **Create two unidirectional pipes to implement two-way communication between the parent and a child.**
* **(Extra Credit)Use one-way communication to compute the summation of 1 through N(N is a real number provided by the user), parent will print the final result from the partial results.**
* **Your program should implement the following steps:**

1. The parent and child process create 2 unique pipes using ***pipe ( ) system call***.

2. The parent process forks (use **fork () system call to spawn the child process**)

3) Both child and parent create 2 File descriptors, one for each pipe()

**Parent sends Message to child process**

a) The writer (parent) closes its read end of the first pipe, and the reader (child) closes its write end of the first pipe.

b) The parent and child communicate by using **write ( )** and **read ( )** calls respectively.

c) Parent sends a message to child, Child will print the message from the Parent(specifying that it is the child printing the parents message)

**Child sends Message to Parent process**

a) The writer (child) closes its read end of the second pipe, and the reader (Parent) closes its write end of the second pipe.

b) The parent and child communicate by using **write ( )** and **read ( )** calls respectively.

c) Child sends a message to Parent, Parent will print the message from the child(specifying that it is the child printing the parents message)

4) Each process closes its active pipe descriptor when it’s finishes with it.

**Computing the sums(Extra Credit 5 points towards grade)**

Note: Once a pipe is used it cannot be reused, you only need to send data in one direction.

1. Parent computes the partial sum of 1 through 5
2. child computes the partial sum of 6 through 10
3. Child sends its result to the Parent using a pipe(use the same technique you used in previous parts)
4. Parent sums the Partial sums together
5. Parent prints the total sum.

Deliverables: Submit on blackboard your complete c++ code, and screen shots of your code running, showing the results.

**10 points**

**Deadline: 12/5/2022 at midnight**

In this lab you will write a program in c++ that will communicate between system processes to accomplish a task.

***NOTE: This lab will only work on linux as the unistd.h library is for linux system commands.***

The c library unistd.h needs to be included in order for the functions, "pipe", "fork", "read", "write", "close"

Pipe documentation, <http://man7.org/linux/man-pages/man2/pipe.2.html>

another example, using pipes and forks

<http://hzqtc.github.io/2012/07/linux-ipc-with-pipes.html>

References:

unistd.h, "write" function documentation, <http://man7.org/linux/man-pages/man2/write.2.html>

unistd.h "read" function documentation, <http://man7.org/linux/man-pages/man2/read.2.html>

unistd.h "close" function documentation, <http://man7.org/linux/man-pages/man2/close.2.html>

unistd.h "fork" function documentation, <http://man7.org/linux/man-pages/man2/fork.2.html>

[LAB5\_interprocessn.docx](https://bbhosted.cuny.edu/bbcswebdav/pid-72234647-dt-content-rid-550515825_1/xid-550515825_1)