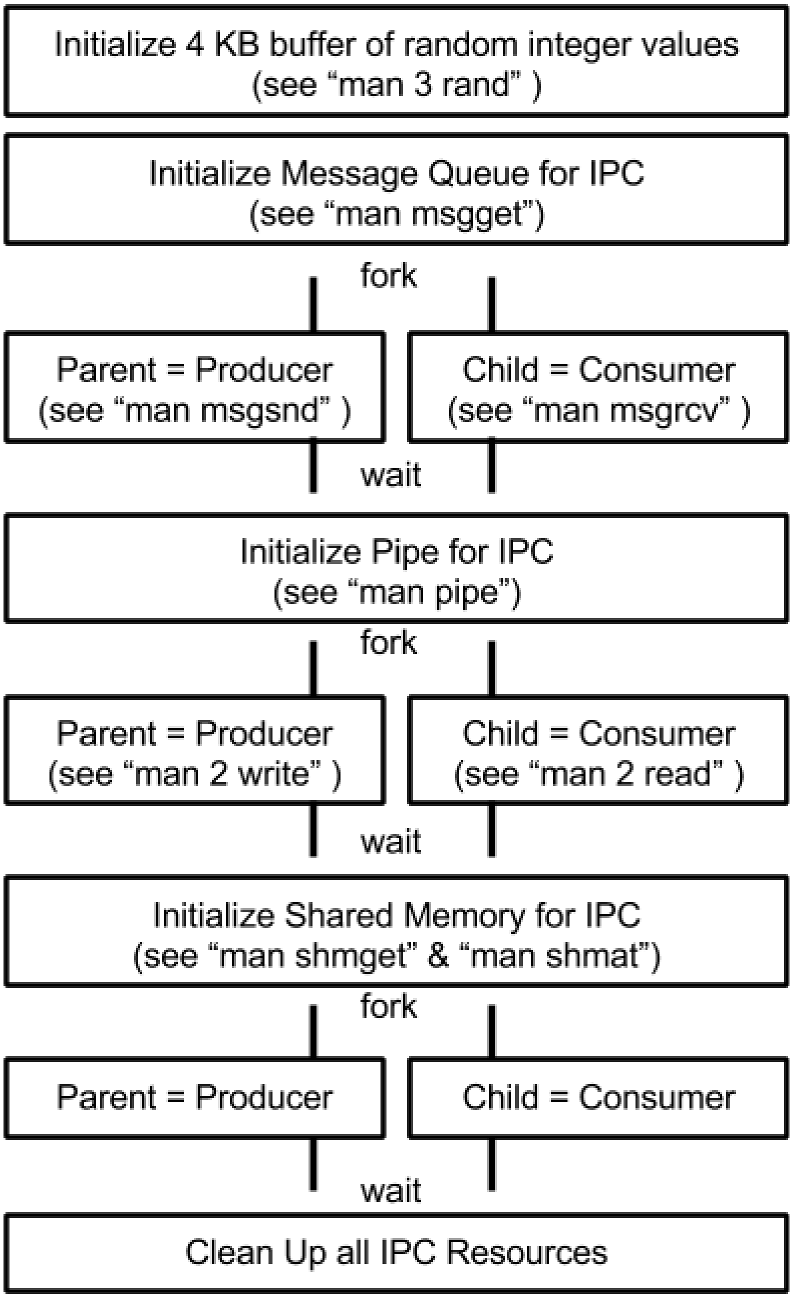
Operating Systems : Assignment 3 : IPC

Due: Wednesday May 4th, 11:59 AM (right before class)

Base tag: A3_BASE **Submission tag: A3**

Interprocess Communication - Program Design:

This assignment will give you experience using **three varied IPC mechanisms** to implement a *Producer-Consumer* multiprocess cooperation. You will construct a program that has the following flow.



In each phase of the program (Producer-Consumer) the parent will send 256 Bytes at a time and the child will receive 256 Bytes at a time. Certain IPC operations must be mutexed using a Semaphore (see “man semget” & “man semctl”).

Interprocess Communications Programming:

You will implement the above flowchart using C / C++ as a CMake project. Please include a README for the TAs. During each phase:

1. The parent process should produce (copy sequentially) 256 bytes from the randomly initialized 4K buffer and communicate that data to the child for consumption using the appropriate IPC mechanism.
2. The child should consume the 256 byte messages and accumulate them into a second buffer
3. Once the whole 4K buffer has been received, the child should perform a byte-by-byte verification of the received buffer against the original buffer. Any errors should be reported. The final submission should have no errors.

All reads and writes from/to the Pipe and the Shared Memory Segment must be mutexed using IPC Semaphores.

Note: Pthread mutexes are not allowed! We are dealing with processes not threads, and these will not work

Rubric:

This assignment is out of 100 points.

Insufficient Parameter Validation -20% of rubric score
Insufficient Error Checking -20% of rubric score
Insufficient Block and Inline Comments -20% of rubric score
Submission compiles with warnings (with -Wall -Wextra -Wshadow) -80%
Submission does not compile -100% of rubric score
Submission has any memory leaks or leaves IPC resources on the system -50% of rubric score