

LAB Assignment Week #3

Topic: Interprocess communication

Name/ID of participant(s)

Instructions:

- You will be working in a group of two. Choosing your own team.
- You are allowed to discuss freely within your group. Avoiding seeking solution from other groups.
- Two computers are provided per group.
- Turn in the result by the end of class period.

Activity 1: Learn how to use interprocess communication in Linux and Windows

- Compile and run lab2_3.cpp(Linux) and lab2_4.cpp(Windows)
- Observe how we implement shared memory in Linux and memory-mapped file in Windows for the consumer-producer problem.

Activity 2: Producer-Consumer Problem among three processes

- Using knowledge from lab2_1.cpp to lab2_4.cpp, you modify these examples to create a program that forks two more processes. Thus, we have 3 concurrent processes, one acts as producer and other two as consumers
- You will be required to solve the problems for:
 - a) Process A: Running in a loop, waiting for two integer values from user, store them in shared variables. Process A terminates (break out of loop) when user enter a negative integer number as the first value.
Process B: Running in a loop, reading the two shared variables, add them together and show the result to the user. Wait for two seconds before starting the next loop.
Process B terminates if the first variable contains a negative integer number.
Process C: Running in a loop, reading the two share variable, multiply them together and show the result to the user. Wait for three seconds before starting the next loop.
Process C terminates if the first variable contains a negative integer number.
 - b) Process A: Running in a loop, waiting for two floating point values from user, store them in shared variables. Process A terminates (break out of loop) when user enter the first value which is larger than 100.
Process B: Running in a loop, reading the two shared variable, subtract the first variable with the second variable, then show the result on the screen. Wait for two

seconds before starting the next loop. Process B terminates if the first variables contains a value that is larger than 100.

Process C: Running in a loop, reading the two shared variable, then show the values stored in both variables on the screen. Wait for three seconds before starting the next iteration. Process C terminates if the first variables contains a value that is larger than 100.