Operating Systems Lab

IPC – Shared Memory

Instructor: Muhammad Ahsan

Lab Task

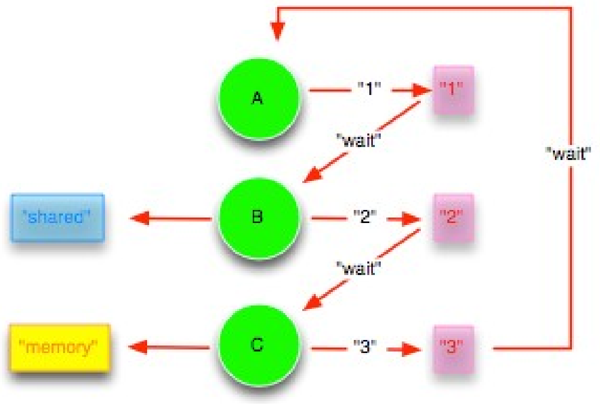
Task-1

Write a C program that creates a shared memory segment, writes an array of integers to it from the parent process, and then reads the array of integers from the shared memory segment in the child process. The child process then performs a series of operations on the array and writes the results back to the shared memory segment. The parent process then reads the results from the shared memory segment and outputs them to the console.

Task-2

Write a program where processes synchronize such that a process **A** prints out the strings of two other separate writing processes (**B** first and then **C** second) from shared memory. process **A** needs to 'wait' by polling until **B** and **C** finish writing their strings to memory.
Here is the sequence of events that needs to be implemented:

1. Process **A** writes to position **1** in memory and then waits until **B** and **C** completes
2. Process **B** writes the string "shared" into memory then signals **A** & **C** that it is complete by writing into memory position **1** (note process **B** should wait to write into position 1 until after process **A** writes into position 1 in memory)
3. Process **C** writes the string "memory" into memory and then signals to process **A** that it is complete by writing into memory position 1 (note process **C** should wait until process **B** writes into position 1 in memory).


A
"1"
"1"
"wait"
"wait"
"shared"
B
"2"
"2"
"wait"
"memory"
C
"3"
"3"