

Project

In this project you are asked to demonstrate the process of converting one number from binary to decimal numbering system using Message Queues, Shared Memory, Semaphores and Signals.

More specifically, you need to implement two different programs, where the first, let's call it "Master", will be responsible for creating an one-dimensional table of random bits(binary digit) as well as to create and initialize all the necessary structures, communication and synchronization mechanisms. The second program, let's call it "worker", will be responsible for finding the multiplication(product) result of the bit (0 or 1) with the corresponding power of 2, which power comes from the position of the digit on the table. For each cell of the table a different "worker" will run.

The initial communication between the "master" and the "workers" will be achieved through a message queue while all other communication as well as synchronization between of processes will be achieved using shared memory, semaphores and signals.

More specifically, the master will create a message queue using one known key and then the workers will retrieve the queue ID using the same key. Also, the "master" will commit a section of shared memory and will connect to it and he will create and initialize a semaphore. It will then send messages of the shared memory key it previously committed through the queue to "workers", and will also send the semaphore key that has been created. The "workers" will be connected with the right order to shared memory and retrieve the semaphore ID.

In the next step the "master" will write the following information to the shared memory:

- The process identifier (pid),
- the number 0 (which then will be used

for assigning a unique ID to each "worker"),

- size of the bit table as well as the elements of the table.

At this point the “workers” should be blocked until the “master” writes the necessary informations in the shared memory and then they will be "warned" by the “master” to continue their execution. Synchronization at this point should be implemented by using the semaphore created by the “master”.

After the master“ notifies” workers to continue , the master will be blocked as soon as the workers complete the calculation that has been assigned to them and write the result in the shared memory. After that they will "notify" the master by sending a signal. The master should continue his execution only when he receives a signal from all workers. Then each worker will read from shared memory his unique id and will increase the value which has been previously read by one.

At this point synchronization is again been demanded with the use of a semaphore provided that only one worker can read at a time and increase his id value. Finally, each worker, based on his id will be assigned to calculate the value of the corresponding cell in the table. For example the id with value 0 will be assigned to a worker who will calculate the value located in cell 0 of the table. Secondly the id with value 1 will be assigned to a worker who will calculate the value that is located in cell 1 of the table, etc. Then, as mentioned above, each worker will write the result of his calculation to the shared memory and will notify the master. After the master receives a signal from all the workers he will read from the shared memory the individual results (products of the value of the bit with the corresponding power of 2) and will calculate the decimal number, by summing the individual products. Finally, it will display on the screen , both the binary digit number and decimal number.