Real Time Systems, January 2024

Implementing a variable shared between a child process and its parent process

Opened: Thursday, 25 January 2024, 12:00 AM
Due: Friday, 9 February 2024, 11:59 PM

✔ Done

Let there be a (parent) process which goes on putting a **random** (integer) number in a shared variable, say **n**, in a regular interval. There is a child process that goes on putting the **factorial** of the integer (that it finds in **n**) back into the shared variable (**n**), in regular intervals. The parent process prints the factorial that it receives from the child process. Device your own mechanism(s) so that the child process "knows" that a new number has been put in **n** by the parent process and the parent process "knows" that a the factorial of a number has been put in **n** by the child process,

Write a complete C program to implement the above. The program should make arrangements for releasing shared memory that has been created during execution.

The processes must print meaningful output so that the user understands what is happening.

Note:

1. For generation for random numbers you may use the "**rand()**" function. "**man 3 rand**" will show you the manual.
2. Since the parent and child processes run independently at their own speed, there may be inconsistency in the results printed by them. For example, the parent process may manage to put numbers in a faster manner than the child process can handle. **You may even try to demonstrate that such inconsistency appears in the output**. How to avoid such inconsistencies will be discussed later.

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 2 mins 3 secs early
Last modified	Friday, 9 February 2024, 11:56 PM
File submissions	⚙ program1.c 9 February 2024, 11:56 PM
Submission comments	▶ Comments (0)

◄ Sample program involving shared memory