Assignment 2
CS3500 - Operating Systems

Deadline: 22 September 2024

Inter Process Communication Implementations

In this assignment you will be asked to implement Producer process that generates stream of messages which is then read by a consumer which does stream processing on these inputs

Functional Requirements

1. The main program should fork 2 processes - `Producer` & `Consumer` with the following description
2. Producer process produces a stream of characters(lowercase alphabets). The characters need to be generated randomly (use inbuilt `PRNG`)
3. Consumer performs stream processing on this and counts the number of vowels read till present.
4. Every 1 min each producer writes the characters produced to a file `prod_<thread_number>` (each character in new line) and similarly every 1 min each consumer writes the number of vowels read from the start into file `cons_<thread_number>` (think of it as counter metric that gets read periodically).
5. Proper process synchronisation is needed to avoid loss of any characters. The program needs to be thread-safe.
6. You are expected to use IPC constructs like `pipes`, `message queues`, `mutexes`, `semophores` etc.
7. Coding must be done is `C` language.

Additional Restrictions

1. A producer thread produces minimum 1 Million characters in a minute (Incase of resource constraints that the 1 million limit is not possible to reach, proper justification with proof should be added to the README file justifying your limit. There will be penality if any student has not added proof for the same or wantedly went with very low production rate.)
2. The producer runs only for 5 mins(i.e, drive the producer to completion and consumer to completion once all characters are processed).
3. Only one shared IPC channel should be used in the problem(common channel to be polled by consumers)

Sub-Problem 1 - Single Producer, Single Consumer

1. In this problem there is only one producer thread and one consumer thread in each of the processes.

Sub-Problem 2 - Single Producer, Multiple Consumer

1. In this problem there is only one producer thread but there are `5` consumer threads.

Sub-Problem 3 - Multiple Producer, Multiple Consumer

1. In this problem there are `5` producer threads and `5` consumer threads.

Submission Instructions

Zip all code files and `README` into `ROLLNO.zip`. `README` should contain instructions to run the assignment, explanation for different code files present and all the assumptions made while solving the assignment.

1. Each subproblem's main C file should be named as `q_<sub_problem_no>.c`
2. Makefile should also be made for the assignment with supporting following commands
    1. `make clean` should remove all the executables
    2. `make q_<sub_problem_no>` should compile the subproblem's code(main executable must be named as `q_<sub_problem_no`)
    3. `make all` should compile all code files.

Assignment 2
CS3500 - Operating Systems

Bonus(Additonal 10 Marks)

1. The producer and consumer process are created in 2 different namespaces.
2. The IPC communication needs to be across these 2 processes that belong to different namespace.
3. Please add a section in `README` for the bonus part explaining the solution
4. Name the bonus main C code as `bonus.c` and also add rule in make file `make bonus` to compile the code. Let the main executable be named as `bonus`