



دانشکده مهندسی کامپیوتر

دکتر رضا انتظاری ملکی

طراح: امیر محمد سهرابی

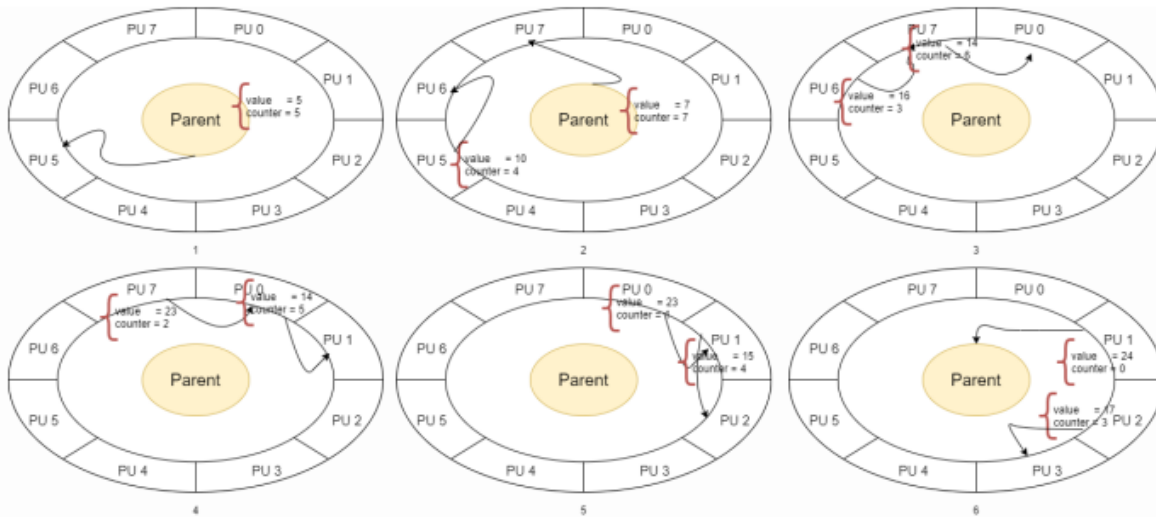
پاییز 1401

Q1

Write a program in C programming language using IPC primitives (shared memory or pipe) to satisfy the described following problem:

- The parent process creates eight children processes which are waiting for a data structure to assign. These processes are called Processing Units (PU).
- The parent process should generate the first hundred prime numbers, and store them in a struct which has a value and a counter attribute (value equals to counter at the beginning), then assign these structures to children based on the modular division of the prime number by 8.
- These structs are assigned to the PUs one by one, in a concurrent way.
- The PUs add their corresponding numbers (0 to 7) to the value and decrement the counter, and then pass the struct to next PU. This loop continues until the counter is zero.
- When the counter is zero, the PU passes the struct to the parent and parent prints it out.

```
struct prime {  
    int value; // value of the prime number  
    int counter; // a counter to decrement (at the start counter = value)  
}
```



In your implementation, you are free to use either pipe or shared memory or semaphores or a combination of these. This brings an expectation about submitting different codes. :DD

Q2

Cross Memory Attach (CMA) is one of the models for Inter-Process Communication. The system calls `process_vm_read()` and `process_vm_write()` are based on this model. Assume one of the two following scenarios based on your student number and implement two C programs to satisfy the desired scenario. The students with odd student number should consider the first scenario and the others consider second.

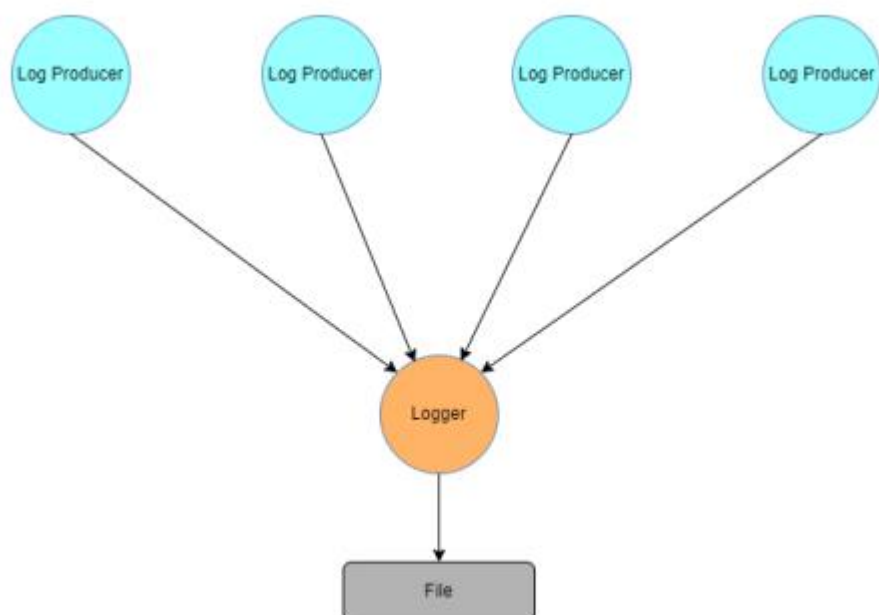
1. A producer program should produce 3 Gigabytes data. A consumer should read that data and prints out.
2. A producer produces 3 Gigabytes of data and then writes in the consumer's memory space.

Note that to synchronize the producer with the consumer you can use either `getchar()` or `sleep()` . Also, recall `memset()` from the class session to generate data.

در این سوال، قصد داریم به پیاده‌سازی یک **Logger** ساده بپردازیم.

در ابتدای کار، شما یک پردازنده را اجرا می‌کنید که وظیفه‌ی این پردازنده، ثبت لاگ‌هایی است که از سمت پردازنده‌های دیگر می‌آیند. (تمامی لاگ‌ها در یک فایل متنی نوشته می‌شوند.) سایر پردازنده‌ها به ارسال لاگ‌های تولیدشده توسط یک مموری مشترک می‌پردازند تا پردازنده‌ی تولیدکننده‌ی لاگ، آنها را داخل فایل مربوطه بنویسد.

نکته: برای پیاده‌سازی این سوال باید از **pipe** استفاده نمایید.



فرض کنید در یک پردازنده، یک پردازهی مادر و چند پردازهی فرزند داریم. (تعداد پردازها توانی از ۲ می باشد). در این سوال از شما انتظار می رود الگوریتم Merge Sort را به صورت توزیع شده پیاده سازی کنید.

حالت عملیاتی که باید انجام دهید، مرتب سازی یک آرایه است. به این صورت که ابتدا آرایهی داده شده را به طور برابر بین تمام پردازه های فرزند تقسیم می کنیم. (فرض کنید آرایهی ورودی به پردازهی مادر داده می شود). هرکدام از این آرایه های کوچکتر پس از مرتب سازی به پردازهی مادر ارسال می شوند. پردازهی مادر پس از دریافت تمام آرایه های مرتب شده، آنها را ادغام کرده و سپس چاپ می کند.

توجه کنید در حل این سوال حتما باید از روش POSIX برای عملیات Shared Memory استفاده کنید.