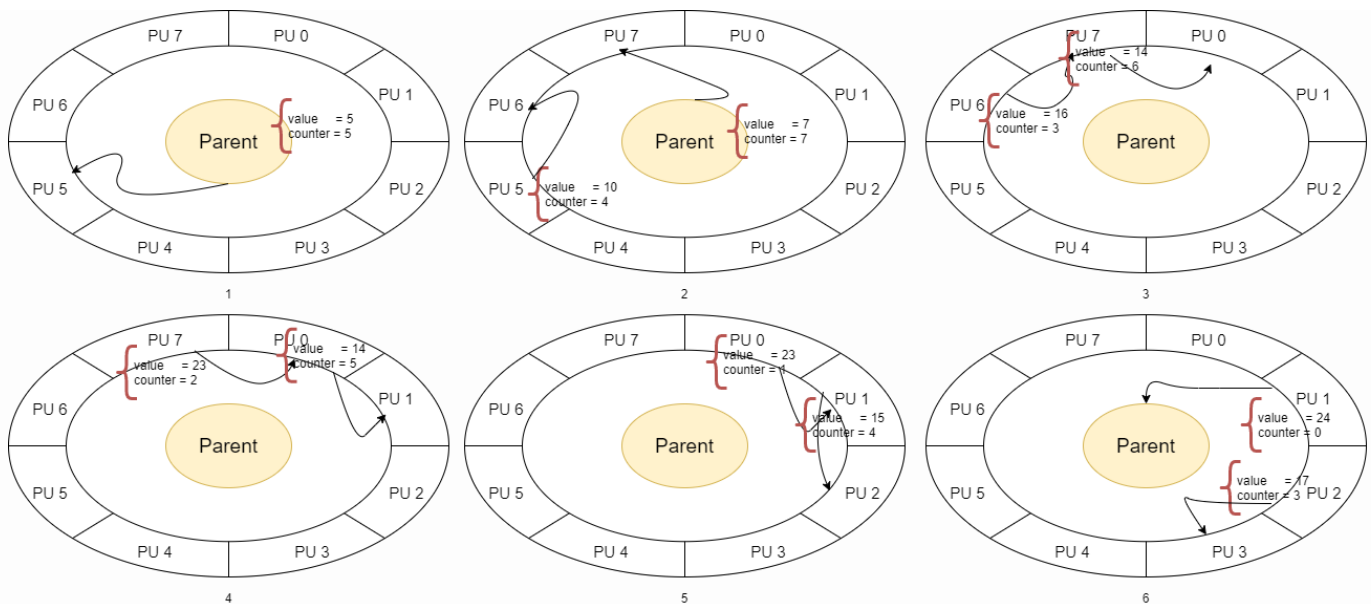# #4 Assignment - Inter-Process Communication

Question 1

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)
}
```

An example scenario is illustrated in the following figure.

PU 7 | PU 0
PU 6 | PU 1
value = 5
counter = 5
Parent
PU 5 | PU 2
PU 4 | PU 3

1

PU 7 | PU 0
PU 6 | PU 1
value = 7
counter = 7
Parent
PU 5 | value = 10
counter = 4 | PU 2
PU 4 | PU 3

2

PU 7 | PU 0
value = 14
counter = 6
PU 6 | value = 16
counter = 3 | PU 1
Parent
PU 5 | PU 2
PU 4 | PU 3

3

PU 7 | PU 0 | value = 14
counter = 5
value = 23
counter = 2
PU 6 | PU 1
Parent
PU 5 | PU 2
PU 4 | PU 3

4

PU 7 | PU 0
value = 23
counter = 4
PU 6 | PU 1
value = 15
counter = 4
Parent
PU 5 | PU 2
PU 4 | PU 3

5

PU 7 | PU 0
PU 6 | PU 1
value = 24
counter = 0
Parent
PU 5 | value = 17
counter = 3 PU 2
PU 4 | PU 3

6

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

Question 2

Write a report about the following subjects:

1. The difference between Cross Memory Attach and other models of IPC methods introduced in the class.
2. The safety of the pipe mechanism.
3. Can `pthread_mutex_lock` be shared among processes (just like semaphores) instead of threads? Why or why not?

Question 3

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.

Some reading materials are:
https://man7.org/linux/man-pages/man2/process_vm_readv.2.html
https://lwn.net/Articles/405284/
https://stackoverflow.com/questions/17324564/cross-memory-attach-how-do-i-get-the-remote-address-from-a-child-process-to-a-p

Deadline

Your midterm is considered for determining the deadline.

- **Tuesday** 8th Dec. 23:00

Submission

Submit just a zip file ,containing your codes and a report, in LMS. The file should be named as [`9752xxxx.zip`]. For example `97521234.zip` .