**Lab 2. Producer and Consumer problem**

Make C or C++ programs to illustrate the Producer and Consumer synchronization problem. You will have to create several processes to simulate the producers and consumers. Use shared memory to implement the shared buffer among producers and consumers. Use semaphore to synchronize the processes. Here are some constraints for the problem.

* A shared buffer with 3 slots, initially are all empty.
* Two producers
  + Randomly wait for a period of time and put product into the buffer.
  + Wait if the buffer slots are all full
  + Repeat 6 times.
* Three consumers
  + Randomly wait for a period of time and fetch a product from the buffer.
  + Wait if the buffer slots are all empty.
  + Repeat 4 times.

Notes：

* Display the status of the buffer and the time when a product has been put into or removed from the buffer.
* Use process (not thread) to simulate the consumer and producer.
* Use system calls like fork() and CreateProcess() to create new processes. Use system calls like shmget() to create shared memory and semget() to create semaphores in Linux.
* In windows, shared memory is implemented as memory-mapped files, you can use function CreateFileMapping to created the shared memory. You can find more information on <https://docs.microsoft.com/en-us/windows/win32/memory/creating-named-shared-memory?redirectedfrom=MSDN>.
* You can use CreateSemaphore and CreateMutex to create semaphore and mutex lock in Windows.
* **Implement a Windows version and a Linux version.**

