#### Operating system experiment report

1. **Name:** Luoyu Mei **Number:** 71117408 **Date:** 05/05/2019
2. **Working target:** Handle consumer producer problem in Windows and Linux operating system by using and .

**Working environment:** as basic operating system with virtual machine and running on it. Using GCC version 8.2.0 on Windows and version 7.4.0 on Ubuntu.

1. **Steps:**
2. In order to make the program satisfy , I set three semaphore and a char buffer array to be the critical region.
3. Producer produce into buffer using signal and make buffer turn to ‘A’. Consumer consume buffer by changing it to ‘B’. Either producer or consumer access , I print buffer out to show changes.
4. Create multi-thread which include 5 producer and 5 consumer，the maximum frequency for each thread to access is 10.

**4. Flow chart:**

begin

Apply for three semaphore: mutex，empty，full

Create multi-thread to be producer and consumer

Access critical region and print out the value int buffer

end

**5. Main data structure:**

I use three semaphores, counter semaphore empty and full, which were respectively set initial value 10 and 0, exclusive semaphore mutex to control the process accession. I define a buffer in type of char to save the status. Whenever consumer or producer access the buffer, I change the value in it and print it out.

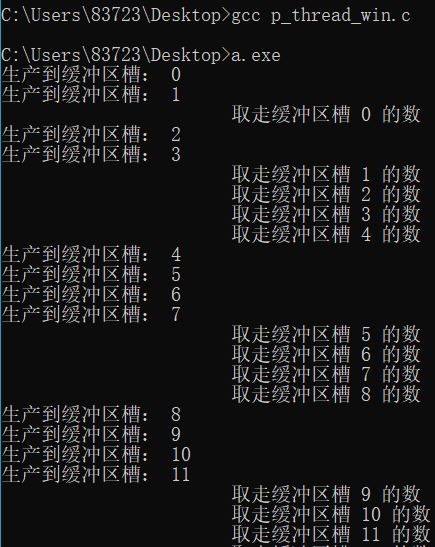
**The code file of my program will also be bale together with this report.**

**6. Experiment result:**

Ubuntu：



Windows：



**7. Filling of experiment:**

It isn’t difficult to design my program, however the realization of it in Windows operating system maybe difficult for me as a beginner. Windows API is power but difficult to learn, I spend half of my coding time to search for reference of it. The most acquisition I got in this experiment is the using of Windows API to comply multi thread.

**8. Code for Linux (如果您在看PDF，代码会在下一页出现):**

****

**Code for Windows (如果您在看PDF，代码会在下一页出现):**

****