CSE 5441 Homework 2

Neng Shi

Experiments

Start the computing node on Owens

```
-sh-4.2$ salloc -A PAS0027 -N 1 -t 1:00:00 -p serial
salloc: Pending job allocation 18024255
salloc: job 18024255 queued and waiting for resources
salloc: job 18024255 has been allocated resources
salloc: Granted job allocation 18024255
salloc: Waiting for resource configuration
salloc: Nodes o0242 are ready for job
sh-4.2$ ssh o0242
********************************************************************************

   This system is for the use of authorized users only.  Individuals using
   this computer system without authority, or in excess of their authority,
   are subject to having all of their activities on this system monitored
   and recorded by system personnel.  In the course of monitoring individuals
   improperly using this system, or in the course of system maintenance,
   the activities of authorized users may also be monitored.  Anyone using
   this system expressly consents to such monitoring and is advised that if
   such monitoring reveals possible evidence of criminal activity, system
   personnel may provide the evidence of such monitoring to law enforcement
   officials.

********************************************************************************
********************************************************************************
```

Running time

Please see “run_script.sh.o18024256”.

Program outputs

Please see the directory “output”.

Brief Explanation

Use mutex to wait

The consumer waits on the condition that “count > 0” and the producer waits on the condition that “count < MAX_BUF_SIZE”. If the condition is met, the consumer or the producer try to obtain the mutex. If it obtains the lock, start working, otherwise continue waiting.

Use mutexes and condtional variables

Both the consumer and the producer first try to get the mutex.
For the consumer, if “count == MAX_BUF_SIZE”, it waits on a conditional variable “cond_nonfull”, otherwise start working. After finishing the work, it operates “pthread_cond_broadcast(&cond_nonempty)”.
For the producer, if “count == 0”, it waits on a conditional variable “cond_nonempty”, otherwise start working. After finishing the work, it operates “pthread_cond_broadcast(&cond_nonfull)”.