

1 Shared Memory II; Threads, OpenMP Examples

1. Producer-consumer using condition variables; complete the following [code](#) segment.

Assignment (due to Nov 6): Implement a multi-access threaded queue with multiple threads inserting and multiple threads extracting from the queue. Use mutex-locks to synchronize access to the queue. Document the time for 1000 insertions and 1000 extractions each by 64 insertion threads (producers) and 64 extraction threads (consumers).

2. Composite Synchronization Constructs; complete the following [code](#) segment. Plot the Execution time vs Number of threads for nproc=2.
3. OpenMP; two examples are given; [hello](#), [workshare](#)

- *Hello world*;

- In this simple example, the master thread forks a parallel region.
- All threads in the team obtain their unique thread number and print it.
- The master thread only prints the total number of threads. Two OpenMP library routines are used to obtain the number of threads and each thread's number.

```
export PGI=/usr/local/pgi
export PATH=$PGI/linux86/6.2/bin:$PATH
export MANPATH=$MANPATH:$PGI/linux86/6.2/man
export LD_LIBRARY_PATH=/usr/local/pgi/linux86/6.2/liblbf:
/usr/local/pgi/linux86/6.2/lib:$LD_LIBRARY_PATH
```

```
export OMP_NUM_THREADS=4 (This line is optional.)
pgcc -mp -o omp_hello omp_hello.c
```

- *Loop work-sharing*;

- The iterations of a loop are scheduled dynamically across the team of threads.
- A thread will perform CHUNK iterations at a time before being scheduled for the next CHUNK of work.

```
pgcc -o omp_workshare omp_workshare1.c -mp
```

- Increase the number of threads and the value of CHUNK, study the output.