Course: Operating Systems

Instructor: Dr. Raheel Ahmed Memon

Lab-10

Objectives

1. Understanding the policies of CPU Scheduling by using simulation technique
2. Understanding the purpose of memory barrier
3. Impact of increasing number of cores
4. Observe the key Quality of Service Parameters

Task # 01

Simulate a system scheduling with memory barrier, this memory barrier can hold the tasks for a limited time and then release them in the form of a package. See the Figure 1 for reference

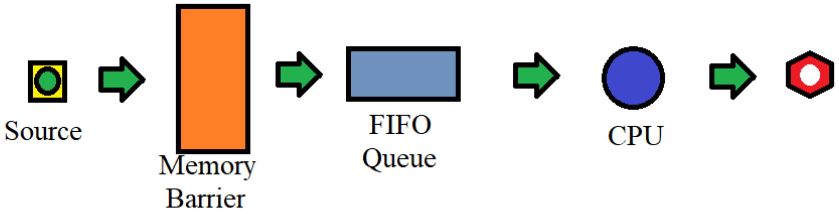

Figure 1 CPU Scheduling with Memory Barrier – Simulation

Following are the required features of Task:

- There can be any policy FIFO/RR/SJF queuing policy
- The jobs should come to the memory barrier, wait for 10 jobs to arrive
- Once the memory barrier is full that batch of job should be shifted to the FIFO queue.

Calculate following results:

- Average waiting in the queue
- Average Turnaround time
- Average System Throughput

Task # 02

Vary the number of cores in Task # 1, calculate the same results for this task also, and compare with the resulting graphs of both Task # 1 and Task # 2. Please see the Figure 2 for reference

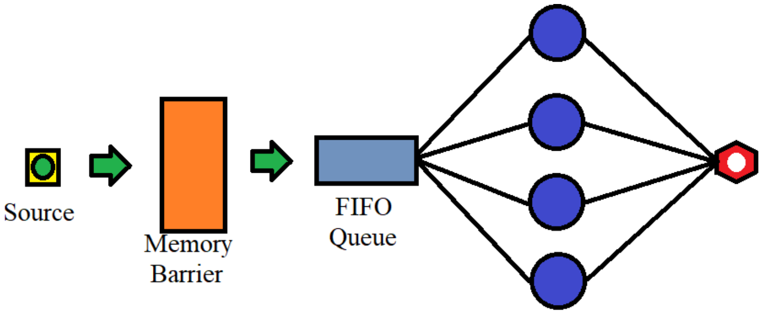

Figure 2 CPU Scheduling with Memory Barrier and multiple cores– Simulation

Named Pipes

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
int main()
{
int res = mkfifo("/tmp/my_fifo", 0777);
if (res == 0)
printf("FIFO created\n");
exit(EXIT_SUCCESS);
}
```

Accessing FIFO file

```
$ cat < /tmp/my_fifo &
$ echo “Hello World” > /tmp/my_fifo
```

Task 3 :

Use the named pipe to communicate between two programmes. Send and receive message to each other.