

McCrae Smith
Professor Ghosh
CMSC 312
4/23/2021

Team Sixteen Homework Three, Task Two

Usage Instructions:

The code files consist of the Makefile, the main SJFFull.c, and the two support files, SJFGlobalQueue.c and CSem2.c, all files must be in the same directory. To compile, simply call the Makefile with 'make', and the executable will be compiled.

When running the executable, include command line arguments for the number of producer processes, and the number of consumer threads, the syntax is './SJFull {producers} {consumers}'.

Upon execution, the program will initialize the buffer, consumers, and producers, at which point they will begin interacting with the buffer, printing their status as they enqueue and dequeue jobs from the buffer. Once the producer processes have completed, the main will then clean up the threads and buffer, returning the average waiting time for the jobs on the queue, and the total execution time of the program.

At any point after initialization, calling a keyboard interrupt (ctrl+C) will call the signal handler, which upon catching the signal will gracefully exit, cleaning up the processes, threads, and buffer.

Description:

As part of Homework Three, this is the Task Two implementation of the buffer, where jobs are consumed in SJF or 'Shortest Job First' fashion. This program is designed to simulate a global queue of print jobs, where the producer processes are users submitting jobs to be printed, and the consumer threads are networked printers removing jobs from the queue.

SJFFull.c contains implementation of the main control process, the producer processes, consumer threads, and the basic signal handler. SJFGlobalQueue.c contains implementation of the buffer itself and its functions. CSem2.c contains correct implementation of the counting semaphores using only binary semaphores as given for Homework Two.

For further description, refer to the Task Two section in the project report documents.