COL100: Lab 10
Threads

March 25, 2017

Part 1: Threads

1. **What is a Thread?**
   A thread is a single sequence stream within in a process. Because threads have some of the properties of processes, they are sometimes called lightweight processes.

2. **What are the differences between process and thread?**
   Threads are not independent of one other like processes. As a result threads shares with other threads their code section, data section and OS resources like open files and signals. But, like process, a thread has its own program counter (PC), a register set, and a stack space.

3. **Why Multithreading?**
   Threads are popular way to improve application through parallelism. For example, in a browser, multiple tabs can be different threads. MS word uses multiple threads, one thread to format the text, other thread to process inputs, etc.

4. **Threads operate faster than processes due to following reasons:**

   (a) Thread creation is much faster.

   (b) Context switching between threads is much faster.

   (c) Threads can be terminated easily

   (d) Communication between threads is faster.

Part 2: Questions

1. Write a program to find the sum of first n numbers using k threads. (The ith thread will add numbers in the range [i*n/k + 1 to (i+1)*n/k + 1) i = 0 to k-1). Do the cost analysis of the work done by each thread assuming the following models of computation: a) Adding two numbers takes constant time b) Adding two numbers is proportional to the size (number of bits) of the numbers. (The work done by each thread would be monotonically increasing)

2. Write a program to find primes in the range of 1 to n using k threads. Does the above cost analysis help you in distributing the range of numbers on which each thread should operate?

3. Write a program to print numbers from 1 to n in order using k threads (n % k = 0) such that the ith thread prints all j such that j % k = i - 1. **Note:** You will have to use mutex in order to achieve this.

4. Write a program containing two threads. One thread reads a file and stores the content in an internal buffer. The other thread writes the content of the buffer into another file.

Part 3: Practice Questions

1. If one thread opens a file with read privileges then
   (a) other threads in the another process can also read from that file
   (b) other threads in the same process can also read from that file
   (c) any other thread can not read from that file
   (d) all of the mentioned

2. Read about preemptive and non preemptive multitasking, if all processes trust each other which one would you prefer? What about the case when processes don't trust one another?

3. What is a mutex and critical section?

4. An operating system uses Shortest Remaining Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

Process	Execution time	Arrival time
P1	20	0
P2	25	15
P3	10	30
P4	15	45

   What is the total waiting time for process P2?

5. Following is how Round Robin Scheduling works: (Read about First come first serve scheduling (FCFS)). Round robin scheduling is similar to FCFS scheduling, except that CPU bursts are assigned with limits called time quantum. When a process is given the CPU, a timer is set for whatever value has been set for a time quantum. If the process finishes its burst before the time quantum timer expires, then it is swapped out of the CPU just like the normal FCFS algorithm. If the timer goes off first, then the process is swapped out of the CPU and moved to the back end of the ready queue. The ready queue is maintained as a circular queue, so when all processes have had a turn, then the scheduler gives the first process another turn, and so on. RR scheduling can give the effect of all processors sharing the CPU equally, although the average wait time can be longer than with other scheduling algorithms. Find average wait time of a process for the following example?

Process	Burst time
P1	24
P2	3
P3	3

   When will RR scheduling not be a good idea?