Tutorial 3 –Threads and Concurrency (Chapters 4)

Operating Systems Comp Sci 3SH3, Winter 2024
Prof. Neerja Mhaskar

Q1) Give 3 examples of data and task parallelism.

Q2) Assume we have an application that is 70 percent parallel and 30 percent serial. If we run this application on a system with 4 processing cores, what is the speed-up?

Q 3) A system with two dual-core processors has four processors available for scheduling. A CPU-intensive application is running on this system. All input is performed at program start-up when a single file must be opened. Similarly, all output is performed just before the program terminates, when the program results must be written to a single file. Between start-up and termination, the program is entirely CPU-bound. Your task is to improve the performance of this application by multithreading it. The application runs on a system that uses the one-to-one threading model (each user thread maps to a kernel thread).

1. How many threads will you create to perform the input and output? Explain.
2. How many threads will you create for the CPU-intensive portion of the application? Explain.

Q4) a) What are the different multithreading models?
b) An operating system that adopts the many-to-one model is installed on a dual core machine. Suppose a process P is created in this system. When P executes it creates a total of three threads T1, T2 and T3. Suppose T2 makes a system call requesting an I/O operation. Do threads T1 and T2 execute, while T1 is waiting? Explain your answer.

Q5) Consider a multicore system and a multithreaded program written using the many-to-many threading model. Let the number of user-level threads in the program be greater than the number of processing cores in the system. Discuss the performance implications of the following scenarios:

a) The number of kernel threads allocated to the program is less than the number of processing cores.
b) The number of kernel threads allocated to the program is equal to the number of processors.
c) The number of kernel threads allocated to the program is greater than the number of processors.

Q6) The program shown in Figure 4.16 (page 194 of the textbook) uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
pid_t pid;
pthread_t tid;
pthread_attr_t attr;

 pid = fork();

 if (pid == 0) { /* child process */
   pthread_attr_init(&attr);
   pthread create(&tid,&attr,runner,NULL);
   pthread_join(tid,NULL);
   printf("CHILD: value = %d",value); /* LINE C */
 }
 else if (pid gt; 0) { /* parent process */
   wait(NULL);
   printf("PARENT: value = %d",value); /* LINE P */
 }

 void *runner(void *param) {
    value = 5;
    pthread_exit(0);
 }
```