Spring'16: CS415/515 Parallel Programming 4/13/16

Prof. Jingke Li (FAB 120-06, li@cs.pdx.edu); Class: MW 2:00-3:50pm @ FAB 40-07; Office Hr: MW 1-2pm & by appt.

Lab 3: Memory Consistency and OpenMP

Download and unzip the file `lab3.zip` from D2L. You'll see a `lab1` directory with some program files.

An Exercise on Memory Consistency Models

Consider the following two groups of threads. Assume that all variables have an initial value of 0 when the execution starts.

```
Group A:                                Group B:
 P1:       P2:       P3:                 P1:        P2:
  a = 1     x = a     y = b               a = 1      x = a
            b = 1     z = a               y = b      b = 1
                                          synch      synch
                                                     z = a
```

Decide which of the value combinations are possible at the end of the execution: (a) Group A under the sequential consistency (SC) model. (b) Group B under the weak consistency (WC) model.

x	y	z	Possible?
0	0	0	
0	0	1	
0	1	0	
0	1	1	

x	y	z	Possible?
1	0	0	
1	0	1	
1	1	0	
1	1	1	

OpenMP Exercises

1. Consider the following OpenMP program (in `demo.c`):

```
int main() {
  int x = 2;
  #pragma omp parallel num_threads(2) shared(x)
  {
    if (omp_get_thread_num() == 0)
      x = 5;
    else
      printf("1: Thread %d: x = %d\n", omp_get_thread_num(), x);
    #pragma omp barrier
    printf("2: Thread %d: x = %d\n", omp_get_thread_num(), x);
  }
  return 0;
}
```

(a) What do you think the two `printf` statements will print?

(b) Compile and run the program to confirm your answer.

2. Parallelizing the following programs by adding OpenMP directives. If there are multiple ways of parallelizing the program, try several different versions. Compile and test the resulting programs.

   (a) `arraysum.c` — a simple array sum program.

   (b) `mtxmul.c` — a matrix-multiplication program.