



Sebastian Hack
Christoph Weidenbach

February 03, 2009

Tutorials for “Advanced C”
Exercise sheet 11

Exercise 11.1: (4 P)

Consider the following program:

```
1 #include <stdio.h>
2
3 int main() {
4     int i;
5
6     #pragma omp parallel private(i) num_threads(2)
7     {
8         for(i = 0; i < 2; ++i) {
9             fprintf(stderr, "%d ", i);
10        }
11    }
12
13    fprintf(stderr, "\n");
14    return 0;
15 }
```

1. What are the possible outputs of this program? Explain what happens. (2P)
2. What happens if you change the number of threads? (1P)
3. What effect would *i* not being private have? (1P)

Exercise 11.2: (4 P)

Implement a function `void matrix_multiply(int* a, int* b, int size);` to perform matrix multiplication on two arrays a and b containing square matrices. Columns of a row shall be adjacent, so $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ shall be $[a, b, c, d]$ in memory. The array a shall hold the result of the multiplication, the array b shall be unchanged after the computation finishes.

Test this function on a sufficiently large problem, parallelize it with OpenMP, modifying the original sequential code as little as possible, and compare the performance (in terms of total execution user time) with OpenMP enabled and disabled.

Exercise 11.3: (2 P)

```
1 void foo(void) {
2     #pragma omp parallel
3     {
4         #pragma omp single
5         { fprintf(stderr, "first part\n"); }
6         #pragma omp barrier
7         #pragma omp single
8         { fprintf(stderr, "second part\n"); }
9     }
10 }
11
12 void bar(void) {
13     #pragma omp parallel
14     {
15         #pragma omp sections
16         {
17             fprintf(stderr, "first part\n");
18             #pragma omp section
19             fprintf(stderr, "second part\n");
20         }
21     }
22 }
```

1. Are `foo()` and `bar()` equivalent? If they are not equivalent, explain the difference. If they are equivalent, explain what they both do. (2P)

Submit your solution until the lecture on February 10.

Note: Joint solutions are not permitted.