OpenMP - Programming and execution model

Matic Brank

LECAD laboratory, FS UL

OpenMP – **Exercise 1**

- Goal: To write an OpenMP program which outputs „Hello world from thread 1-N“ by each MP process (help yourself with OpenMP runtime functions)
- Instructions:
  - Open *e1.c* file.
  - Include omp header file in the beginning of the file (should be above main function)
  - Define parallel region with pragma
  - Use printf function to print “Hello World from thread n of N”, where n is the ID number of thread and N is the number of all threads. Use runtime function that returns the ID of each thread.
  - Compile and run it in serial on a single processor (is there an error?)
  - Run it on several processors in parallel by using environment variable `export OMP_NUM_THREADS=n`
- Expected result:
  - The code outputs “Hello World from thread 1 of N” for each thread.

OpenMP – Clauses and directives format

- Directive format
  - Format:

    `#pragma omp` *directive_name* [clause[clause]…]

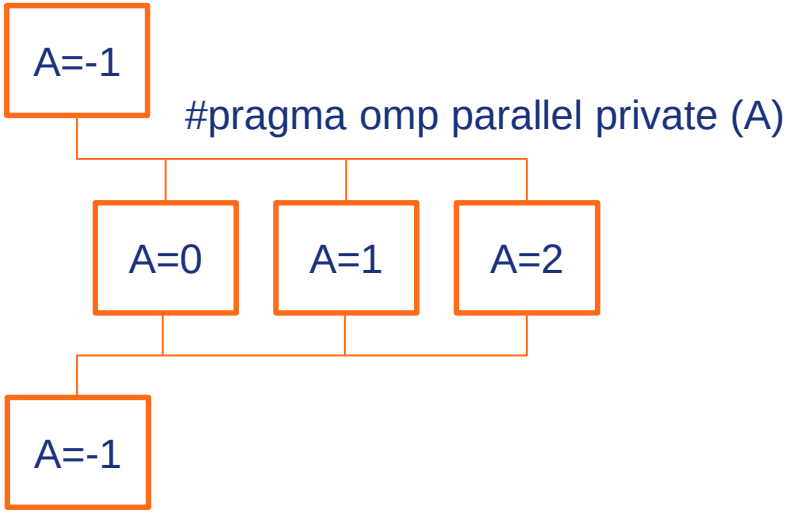
  - Conditionals:

```
#ifdef _OPENMP
        block of code to be executed if code was compiled with OpenMP, for example
        printf("Number of threads: %d", omp_get_num_threads);
#else
        block of code to be executed if code was compiled without OpenMP
#endif
```

OpenMP – Clauses and directives format

- Clauses
  - In C/C++ clauses can be:
    - **private** (list) – in this case the variable is private to each thread
    - **shared** (list) – in this case the variable is shared between threads
  - C/C++:

```
int A;
#pragma omp parallel private(A)
{
A=omp_get_thread_num();
…
}
```



OpenMP – **Exercise 2**

- Goal: To use if/else clauses
- Instructions
  - Open file *e2.c* and move to comment `/* Begin of SPACE for second exercise */` and add an `if` clause if omp is used. Inside add a parallel region that prints the ID of each thread (with `omp_get_thread_num()`) and the number of all threads (with `omp_get_num_threads()`).
  - Compile and run on 4 CPUs and then on 12 CPUs. Observe output.
  - Add an `else` clause if program is not compiled with OpenMP. Add a print statement that says: „The program is not compiled with OpenMP“. Compile the program without OpenMP and run it. Observe the output.

OpenMP – **Exercise 2**

- Define a new variable and assign `omp_get_thread_num()` to it
- Add a `private` clause to `omp parallel` directive for the variable associated with `omp_get_thread_num()` and observe the difference in the output.
- Check if you get race condition:
  - Two threads access the same shared variable and at least one thread modifies the variable and the accesses are unsynchronized
  - Outcome of the program depends on timing of the threads in the team
  - This is caused by unintended shared of data

```
thread 0 of 4
thread 0 of 4
thread 0 of 4
thread 0 of 4
```

Don‘t worry if you always get correct output, because the compiler may use a private register on each thread instead of writing into shared memory

OpenMP – **Exercise 2**

- Expected output:
  - If compiled with OpenMP, the program should output and ID of each thread and number of all threads
  - If not compiled with OpenMP, the program should output “The program was not compiled with OpenMP“

THANK YOU FOR YOUR ATTENTION

www.prace-ri.eu