

# OpenMP Tutorial

## OpenMP's "Hello World"

Name this little "Hello World" program `hello.c`:

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

int main(int argc, char *argv[]) {
    int iam = 0, np = 1;

    #pragma omp parallel default(shared) private(iam, np)
    {
        #if defined (_OPENMP)
            np = omp_get_num_threads();
            iam = omp_get_thread_num();
        #endif
        printf("Hello from thread %d out of %d\n", iam, np);
    }
}
```

## Compiling and Linking OpenMP Programs

Once you have your OpenMP example program, you can compile and link it with

- Linux:

```
/afs/slac.stanford.edu/package/intel_tools/compiler9.0/@sys/cc/bin/icc
-openmp hello.c -o hello
```

- Solaris:

```
/afs/slac/package/sunworkshop/10/SUNWspro/bin/cc -xopenmp=noopt
hello.c -o hello
```

## Running OpenMP Programs

The OpenMP runtime environment needs an environment variable to tell it how many threads you want to use for your program. In `bash` syntax, this looks like this

```
export OMP_NUM_THREADS=4
```

Now you can start your program and it will execute with 4 parallel threads:

```
alfw@rhel6-64> ./hello
Hello from thread 0 out of 4
Hello from thread 1 out of 4
Hello from thread 2 out of 4
Hello from thread 3 out of 4
```

Note that if the computer you are executing your OpenMPI program on has fewer CPUs or cores than the number of threads you have specified in `OMP_NUM_THREADS`, the OpenMP runtime environment will still spawn as many threads but the operating system will sequentialize them.

## References

- [www.openmp.org](http://www.openmp.org)
- [OpenMP Tutorial from LLNL](#)

- [Getting Started with OpenMP -- C version](#)
- Sun Microsystem's Studio 11 [OpenMP API User's Guide](#)

---

[Alf Wachsmann](#). *Last Modified: Mar 01, 2006.*