

OpenMP- Hello world!

Let's print the statement "Hello world!" using all the threads in your computer:

A screenshot of a video player interface. The video frame shows a dark background with green and white text representing C code. The code includes OpenMP headers, a main function, and a parallel region where threads print their IDs. A play button is centered over the video. In the top right corner of the video frame, the text '!OpenMP Hello world' is displayed next to a blue circular icon with a white 'S'. The bottom left of the video frame shows 'lo.c* 25L, 550C' and the bottom right shows '1,1'.

```
<omp.h>
<stdio.h>

int main(int argc, char *argv[]) {
    int nthreads, tid;

    /* Fork a team of threads with each thread having a private tid variable */
    #pragma omp parallel private(tid)
    {
        /* Obtain and print thread id */
        tid = omp_get_thread_num();
        printf("Hello World! from thread = %d\n", tid);

        /* Only master thread does this */
        if (tid == 0)
        {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);

            /* All threads join master thread and terminate */
        }
    }
}
```

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

int main(int argc, char *argv[]) {

    int nthreads, tid;

    /* Fork a team of threads with each thread having a private tid variable */
    #pragma omp parallel private(tid)
    {

        /* Obtain and print thread id */
        tid = omp_get_thread_num();
        printf("Hello World! from thread = %d\n", tid);

        /* Only master thread does this */
        if (tid == 0)
        {
            nthreads = omp_get_num_threads();
```



```
    nthreads = omp_get_num_threads();  
    printf("Number of threads = %d\n", nthreads);  
}  
  
} /* All threads join master thread and terminate */  
  
}
```

To compile this code, run the following:

```
gcc -o omp_helloc -fopenmp omp_hello.c  
$ export OMP_NUM_THREADS=2  
$ ./omp_helloc  
Hello World from thread = 0  
Hello World from thread = 1  
Number of threads = 2
```



Note that the environment variable `OMP_NUM_THREADS` was used to specify number of threads.