

```

/*
  OpenMP example program that computes the value of Pi using the
  trapezoid rule (see Lecture#2).
  Compile with gcc -fopenmp -O3 omp_pi.c -o omp_pi
*/

#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>

void print_usage(char *s) {
    printf("Usage: %s -i <nr of intervals>\n", s);
    exit(0);
}

/* This is the function to integrate */
double f(double x) {
    return (4.0 / (1.0 + x*x));
}

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

    const double PI24 = 3.141592653589793238462643;
    int nthreads, tid;
    double d, x, sum=0.0, pi;
    double starttime, stoptime;
    int n=1000, i;
    char c;

    /* Check if we have at least one argument */
    if (argc <=1 ) {
        print_usage(argv[0]);
    }
    else {
        /* Parse the arguments for a -h or -i flag */
        while ((c=getopt(argc, argv, "hi:")) != EOF) {
            switch (c) {
                case 'h':
                    print_usage(argv[0]);
                case 'i':
                    n = atoi(optarg);    /* Get number of intervals */
                    break;
                default:
                    print_usage(argv[0]);
            }
        }
    }

    /* Compute the size of intervals */
    d = 1.0/(double) n;

    /* Start the threads */
    #pragma omp parallel default(shared) private(nthreads, tid, x)
    {
        /* Get the thread number */
        tid = omp_get_thread_num();

        /* The master thread checks how many there are */

```

```

#pragma omp master
{
    nthreads = omp_get_num_threads();
    printf("Number of threads = %d\n", nthreads);
    starttime = omp_get_wtime(); /* Measure the execution time */
}

/* This loop is executed in parallel by the threads */
#pragma omp for reduction(+:sum)
for (i=0; i<n; i++) {
    x = d*(double)i;
    sum += f(x) + f(x+d);
}
} /* The parallel section ends here */

/* The multiplication by d and division by 2 is moved out of the loop
*/
pi = d*sum*0.5;

stoptime = omp_get_wtime();
printf("The computed value of Pi is %2.24f\n", pi);
printf("The \"exact\" value of Pi is %2.24f\n", PI24);
printf("The difference is %e\n", fabs(PI24-pi));
printf("Time: %2.4f seconds \n", stoptime-starttime);

exit(0);
}

```