```c
/*
   OpenMP example program which computes the dot product of two arrays a and b
   (that is sum(a[i]*b[i]) ) using explicit synchronization with a critical region.
   Compile with gcc -fopenmp omp_critical.c -o omp_critical
*/

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

#define N 100

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

  double a[N], b[N];
  double localsum, sum = 0.0;
  int i, tid, nthreads;

#pragma omp parallel shared(a,b,sum) private(i, localsum)
  {
    /* Get thread number */
    tid = omp_get_thread_num();

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

    /* Initialization */
#pragma omp for
    for (i=0; i < N; i++)
      a[i] = b[i] = (double)i;

    localsum = 0.0;

    /* Compute the local sums of all products */
#pragma omp for
    for (i=0; i < N; i++)
      localsum = localsum + (a[i] * b[i]);

#pragma omp critical
    sum = sum+localsum;

  }  /* End of parallel region */

  printf("   Sum = %2.1f\n",sum);
  exit(0);
}

```