```
/* File:     omp_ppc.c
 *
 * Purpose:  A parallel program that uses OpenMP
 *
 * Compile:  g++ -g -Wall -fopenmp -o omp_ppcX omp_ppc.c
 * Run:      ./omp_ppc <number of threads>
 *
 * Input:    Number of Threads
 * Output:   A message from each thread
 *
 * PPC:      Novembro 2014
 */
#include <iostream>
#include <stdlib.h>

#ifdef _OPENMP
    #include <omp.h>
#else
    #define omp_get_num_threads() 0
    #define omp_get_thread_num() 0
#endif

using namespace std;
/*--------------------------------------------------------------------*/
int main(int argc, char* argv[]) {
  int thread_count = strtol(argv[1], NULL, 10);

  //#  pragma omp parallel num_threads(thread_count)
  //#  pragma omp for ordered
  // #  pragma omp
  //#  pragma omp single
  //#  pragma omp master
   for(int i=0;i<8;i++) {
     int id = omp_get_thread_num();
     #pragma omp critical
     cout << "id " << id << "->" << i << endl;
     //#  pragma omp barrier
   }

   return 0;
}  /* main */
```