Assignment / Lab Experiment 04

Name : Kiran K. Patil Sem : 06
Reg ID : 21107090 4. Course : Parallel computing Lab.

Aim : Implementation using OpenMP.

1] Fork - Join model
2] Producer consumer problem.
3] Matrix multiplication
4] Find prime number
5] Largest Element in an array.
6] Pi calculation.

Theory : OpenMP is a standard programming API for shared memory environments, written in C, C++ or FORTRAN. It consists of a set of compiler directives with a "lightweight" syntax, library routines, and environment variables that influence run-time behaviour. OpenMP is governed by OpenMP architecture Review board (or OpenMP ARB), and is defined by several hardware and software vendors.

OpenMP identifies parallel regions as blocks of code that may run in parallel. Application developers insert compiler derictives into their code at parallel regions and these directives instruct the openMP run-time library to execute the region in parallel.

To create the new team of threads in program # pragma omp parallel is used ineide the program.

Whenever the OpenMP encounters the directive as.
(# pragma omp parallel)

It creates as many threads which are processing cores in the system. Thus, for a dual-core system, two threads are created, for a quad core system four threads are created; and so fourth. Then all the threads simultaneously execut the parallel region. When each thread exits the parallel region it is terminated. OpenMP provides several additional directives for running code regions in parallel including parallizing loops.

In addition to providing directives for personal OpenMP allows developers to choose among several levels of parallism. E.g. they can set the number of threads manually it also allows — developers to identify whether data are shared between threads or are private to the thread. OpenMP is available on several open-source and commersial compilers for linux, windows and mac os x systems.