

DEPARTMENT OF INFORMATION TECHNOLOGY, NITK, SURATHKAL

Parallel Computing

Lab-1-6th August 2024

Note: Execute the following programs given below. Observe the results of each program, write the analysis of the results and take the screenshot of the results. Upload in the Moodle along with the code and screenshots of the results.

Total marks: 1*10 = 10

1. Write an OpenMP program to print Hello World by setting the required number of threads using the *Environment Variable* in the program.
2. Write a simple OpenMP program to print Hello World by adding appropriate *clause* to the parallel directive in order to set the number of threads.
3. Demonstrate the *shared variable concept* using a simple OpenMP program.
4. Demonstrate the *fork-join* parallel execution model in an OpenMP program.
5. Write an OpenMP program to extract the thread identifiers from total number of threads using *private clause*.
6. Write an OpenMP program to set the number of OpenMP threads and retrieve the thread ID number at runtime.
7. Write a simple OpenMP program to demonstrate the parallel loop construct.
 - (i) Use OMP_SET_THREAD_NUM() and OMP_GET_THREAD_NUM() to find the number of processing unit.
 - (ii) Use function invoke to print 'Hello World'.
 - (iii) To examine the above scenario, the functions such as omp_get_num_procs(), omp_set_num_threads(), omp_get_num_threads(), omp_in_parallel(), omp_get_dynamic() and omp_get_nested() are listed and the explanation is given below to explore the concept practically.

omp_set_num_threads() - takes an integer argument and requests that the Operating System provide that number of threads in subsequent parallel regions.

omp_get_num_threads() (integer function) - returns the actual number of threads in the current team of threads.

omp_get_thread_num() (integer function) - returns the ID of a thread, where the ID ranges from 0 to the number of threads minus 1. The thread with the ID of 0 is the master thread.

omp_get_num_procs() - returns the number of processors that are available when the function is called.

omp_get_dynamic() - returns a value that indicates if the number of threads available in subsequent parallel region can be adjusted by the run time.

omp_get_nested() returns a value that indicates if nested parallelism is enabled.

8. Write a simple OpenMP program to illustrate the usage of *shared clause*.
9. Demonstrate the use of *private clause* in a simple OpenMP program.
10. Write a simple OpenMP program that uses *if clause* to get the desired output.
