NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

DEPARTMENT OF INFORMATION TECHNOLOGY

IT 301 Parallel Computing LAB 2

19th August 2020

Faculty: Dr. Geetha V and Mrs. Tanmayee

-------------------------------------------------------------------------------------------------------------------------------

Execute following programs and put screen shots of the output. Write analysis of the result before uploading in IRIS as a single pdf file. for programming exercises, write the code and also put screenshot of the results.

**1. Program 1**

**Aim: To understand and analyze shared clause in parallel directive.**

/\*shared.c\*/

#include<omp.h>

int main()

{

int x=0;

#pragma omp parallel shared(x)

{

int tid=omp\_get\_thread\_num();

x=x+1;

printf(“Thread [%d]\n value of x is %d”,tid,x);

}

}

**2. Program 2**

**Learn the concept of private(), firstprivate()**

/\*learn.c\*/

#include<stdio.h>

#include<omp.h>

int main()

{

int i=10;

printf("Value before pragma i=%d\n",i);

#pragma omp parallel num\_threads(4) private(i)

{

printf("Value after entering pragma i=%d tid=%d\n",i, omp\_get\_thread\_num());

i=i+omp\_get\_thread\_num(); //adds thread\_id to i

printf("Value after changing value i=%d tid=%d\n",i, omp\_get\_thread\_num());

}

printf("Value after having pragma i=%d tid=%d\n",i, omp\_get\_thread\_num());

}

**\* Note down the result by changing private() to firstprivate().**

**3. Program 3**

**Learn the working of lastprivate() clause:**

#include<stdio.h>

#include<omp.h>

void main()

{ int x=0,i,n;

printf("Enter the value of n");

scanf("%d",&n);

#pragma omp parallel

{

int id=omp\_get\_thread\_num();

#pragma omp for lastprivate(i)

for(i=0;i<n;i++)

{

printf("Thread %d: value of i : %d\n",id,i);

x=x+i;

printf("Thread %d: x is %d\n",id,x);

}

}

printf("x is %d\n",x);

printf("i IS %d\n",i);

}

**4. Demonstration of reduction clause in parallel directive.**

#include<stdio.h>

#include<omp.h>

void main()

{

int x=0;

#pragma omp parallel num\_threads(6) reduction(+:x)

{

int id=omp\_get\_thread\_num();

int threads=omp\_get\_num\_threads();

x=x+1;

printf("Hi from %d\n value of x : %d\n",id,x);

}

printf("Final x:%d\n",x);

}

**5. Programming exercise**

1. Write a parallel program to calculate the sum of elements in an array

2. Write a parallel program to calculate the a[i]=b[i]+c[i], for all elements in array b[] and c[]

3. Write a parallel program to find the largest among all elements in an array.