NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

DEPARTMENT OF INFORMATION TECHNOLOGY

**IT 301 Parallel Computing (Minor) LAB 6**

08 Feb 2021

Faculty: Dr. Geetha V

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

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.

Total Marks =10

**1. Program 1**

**Execute following code and observe the working of threadprivate directive and copyin clause: [ 2 Marks]**

#include<stdio.h>

#include<omp.h>

int tid,x;

#pragma omp threadprivate(x,tid)

void main()

{

x=10;

#pragma omp parallel num\_threads(4) copyin(x)

{

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

#pragma omp master

{

printf("Parallel Region 1 \n");

x=x+1;

}

#pragma omp barrier

if(tid==1)

x=x+2;

printf("Thread % d Value of x is %d\n",tid,x);

}//#pragma omp barrier

#pragma omp parallel num\_threads(4)

{

#pragma omp master

{

printf("Parallel Region 2 \n");

}

#pragma omp barrier

printf("Thread %d Value of x is %d\n",tid,x);

}

printf("Value of x in Main Region is %d\n",x);

}

**DO the following:**

**1. Remove copyin clause and check the output.**

**2. Remove copyin clause and initialize x globally.**

**Note the observation about threadprivate directive and copyin clause.**

**------------------------------------------------------------------------------------**

**2. Program 2**

**Learn the concept of firstprivate() and threadprivate() [2 Marks]**

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

int count=0;

#pragma omp threadprivate(count)

int main (void) {

int x=10, y=20,a[10],b[10],c[10],i;

//int count=0;

for(i=0;i<10;i++)

b[i]=c[i]=i;

printf("1. count=%d\n",count);

#pragma omp parallel num\_threads(2) copyin(count)

{

#pragma omp for schedule(static,5) firstprivate(x)

for(i=0;i<10;i++)

{

int tid1=omp\_get\_thread\_num();

a[i]=b[i]+c[i];

count++;

x++;

printf("tid=%d,a[%d]=%d, count=%d x=%d\n",tid1,i,a[i],count,x);

}

#pragma omp barrier

printf("2. before copyprivate count=%d x=%d tid=%d\n",count,x,omp\_get\_thread\_num());

#pragma omp single copyprivate(count)

{

count=count+20;

}

printf("3. after copyprivate count=%d x=%d tid=%d\n",count,x,omp\_get\_thread\_num());

#pragma omp for schedule(static,5) firstprivate(x)

for(i=0;i<10;i++)

{

int tid1=omp\_get\_thread\_num();

a[i]=b[i]\*c[i];

count++;

x++;

printf("tid=%d,a[%d]=%d, count=%d, x=%d\n",tid1,i,a[i],count,x);

}

}

#pragma omp barrier

printf("4. count=%d x=%d\n",count,x);

printf("\n");

return 0;

}

**Analyse the results for variable count and x.**

**---------------------------------------------------------------------------------------------**

**3. Program to understand the concept of collapse() [6 Marks]**

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

int main (void) {

int i,j;

#pragma omp parallel

{

#pragma omp for schedule(static,3) private(i,j)

for(i=0;i<6;i++)

for(j=0;j<5;j++)

{

int tid2=omp\_get\_thread\_num();

printf("tid=%d, i=%d j=%d\n",omp\_get\_thread\_num(),i,j);

}

}

return 0;

}

**Consider three for loops and check the result with no collapse(), collapse(2) and collapse(3). [Marks 2+2+2=6 with explanation]**