NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL
DEPARTMENT OF INFORMATION TECHNOLOGY

IT 301 Parallel Computing LAB 4

24th August 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 =3+2+2+3=10 marks

1. Understanding concept of schedule. Write the observation using schedule (static, 5), schedule (dynamic, 5) and schedule (guided, 5) [Marks: 1+1+1=3]

```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
int main (void) {
int i;
#pragma omp parallel num_threads(4)
{
  #pragma omp for schedule(guided,5) private(i)
    for(i=0;i<27;i++)
     {
       printf("tid=%d, i=%d \n",omp_get_thread_num(),i);
     }
   }
  return 0;
}
```

2. Execute following code and observe the working of threadprivate directive and copyin clause:

```
#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: [Marks: 1+1=2]
a. Remove copyin clause and check the output.
b. Remove copyin clause and initialize x globally.
Note the observation about threadprivate directive and copyin clause.

---

3. Learn the concept of firstprivate() and threadprivate()

```
#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. write your observation [Marks: 1+1=2]

---

4. Program to understand the concept of collapse()

```
#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) collapse(2)
    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 <u>three for loops</u> and check the result with no collapse(), collapse(2) and collapse(3). [1+1+1=3 Marks]
