

# PARALLEL AND DISTRIBUTED COMPUTING

L – 19,20

DATE: 11.7.19

DHRUBANKA DUTTA, 17BCE1019

## 1. HELLO WORLD

```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>

int main (int argc, char *argv[])
{
    int nthreads, tid;

    /*Fork a team of threads giving them their own copies of variables*/
    #pragma omp parallel private (nthreads, tid)
    {
        /*Obtain thread number*/
        tid = omp_get_thread_num();
        printf("Hello world from thread = %d\n", tid);
        /*Only master thread does this*/
        if (tid==0)
        {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        }
    }
    /*All threads join the master and disband*/
}
```

```
gcc -fopenmp pdclabHelloWorld.c && time ./a.out
```

Hello world from thread = 0

Number of threads = 4

Hello world from thread = 2

Hello world from thread = 1

Hello world from thread = 3

real 0m0.003s

user 0m0.005s

sys 0m0.000s

## 2. ADDITION

```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>
```

```
int main (int argc, char *argv[])
```

```

{
int nthreads, a=4, b=5, c, tid;

/*Fork a team of threads giving them their own copies of variables*/
#pragma omp parallel private (nthreads, tid)
{
/*Obtain thread number*/
tid = omp_get_thread_num();
c = a+b;
printf("Result of addition from thread = %d\n", tid);
printf("Result of addition = %d\n", c);
/*Only master thread does this*/
if (tid==0)
{
nthreads = omp_get_num_threads();
printf("Number of threads = %d\n", nthreads);
}
}
/*All threads join the master and disband*/
}

```

gcc -fopenmp pdclabSum.c && time ./a.out

Result of addition from thread = 0

Result of addition = 9

Number of threads = 4

Result of addition from thread = 3

Result of addition = 9

Result of addition from thread = 2

Result of addition = 9

Result of addition from thread = 1

Result of addition = 9

real 0m0.006s

user 0m0.010s

sys 0m0.000s

### 3. SUBTRACTION

```

#include<omp.h>

```

```

#include<stdio.h>

```

```

#include<stdlib.h>

```

```

int main (int argc, char *argv[])

```

```

{

```

```

int nthreads, a=4, b=5, c, tid;

```

```

/*Fork a team of threads giving them their own copies of variables*/

```

```

#pragma omp parallel private (nthreads, tid)

```

```

{

```

```

/*Obtain thread number*/
tid = omp_get_thread_num();
c = a-b;
printf("Result of addition from thread = %d\n", tid);
printf("Result of addition = %d\n", c);
/*Only master thread does this*/
if (tid==0)
{
nthreads = omp_get_num_threads();
printf("Number of threads = %d\n", nthreads);
}
}
/*All threads join the master and disband*/
}

```

gcc -fopenmp pdclabDifference.c && time ./a.out

```

Result of addition from thread = 1
Result of addition = -1
Result of addition from thread = 3
Result of addition = -1
Result of addition from thread = 0
Result of addition = -1
Number of threads = 4
Result of addition from thread = 2
Result of addition = -1

```

```

real    0m0.004s
user    0m0.010s
sys     0m0.000s

```