Fall Sem 2021-22

Assignment: 5

Date: 5/10/21

Name: Ishan Sagar Jogalekar

Reg no: 19BCE2250

Course: Parallel and distributed computing LAB - CSE4001

Slot: L55+L56

Aim: Write a simple OpenMP program to demonstrate the use of pattern generation in schedule clause.

Ans –

1. **static** - Statically assign the loop iterations to threads

SOURCE CODE:

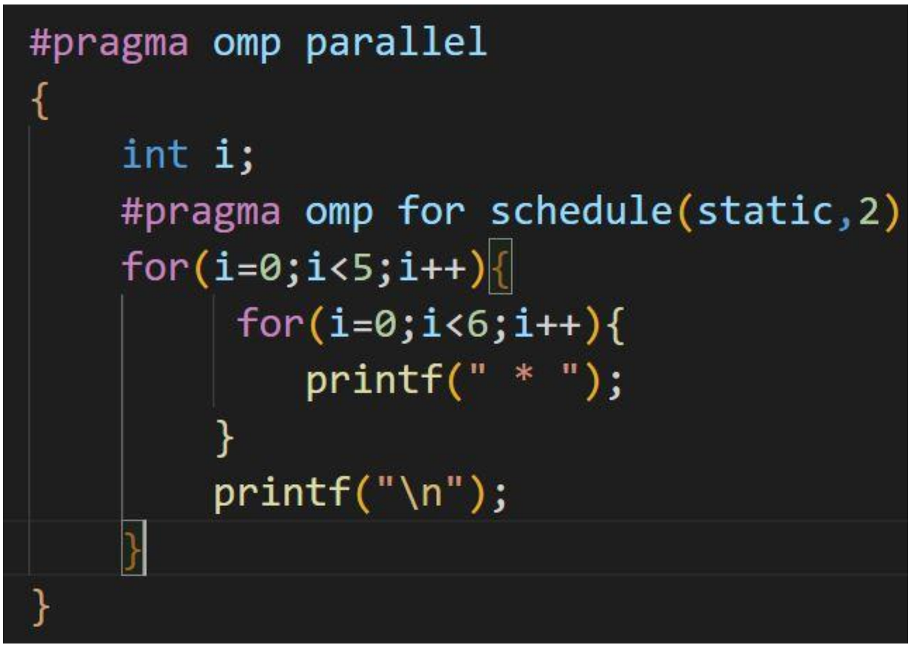
```
#include <stdio.h>
#include <omp.h>
void main(){
    printf("19BCE2250 - Ishan Jogalekar\n");
    printf("--Star pattern using Openmp : Scheduling(Static)--\n");
  #pragma omp parallel
    {
        int i;
        #pragma omp for schedule(static,2)
        for(i=0;i<5;i++){
             for(i=0;i<6;i++){
                printf(" * ");
            }
            printf("\n");
        }
    }
}
```

EXECUTION:

- <omp.h> is header file used to access Openmp functions within program.
- #pragma omp parallel starts parallel execution of process using OpenMP.
- Schedule is used as #pragram omp for schedule.
- Scheduling is a method in OpenMP to distribute iterations to different threads in for loop.
- Inside schedule clause, static is used to assign loops to threads statically.
- Also, size of chunk gives as 2 to divide iterations into chunks.

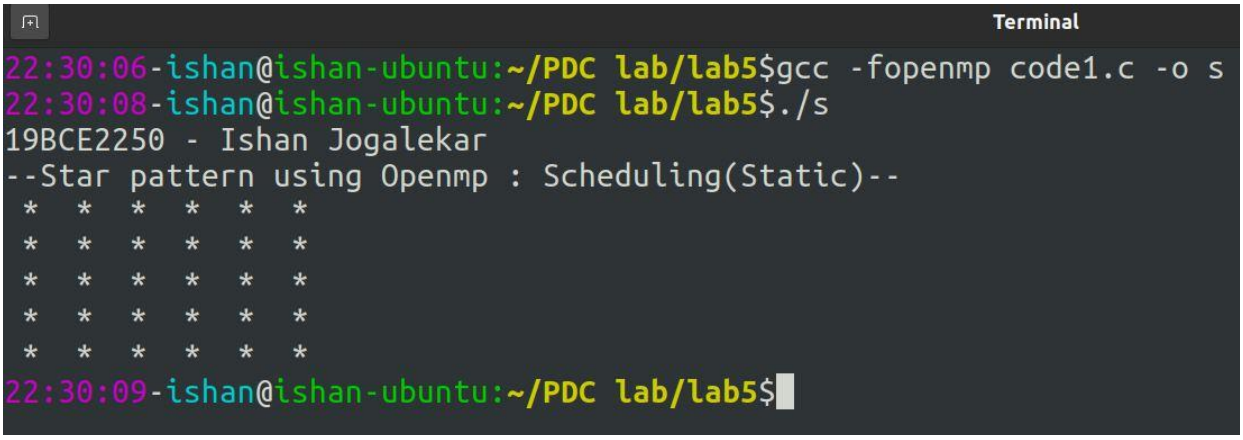
RESULTS:

1. Main logic:



```
#pragma omp parallel
{
    int i;
    #pragma omp for schedule(static,2)
    for(i=0;i<5;i++){
        for(i=0;i<6;i++){
            printf(" * ");
        }
        printf("\n");
    }
}
```

2.Output:



```
22:30:06-ishan@ishan-ubuntu:~/PDC lab/lab5$gcc -fopenmp code1.c -o s
22:30:08-ishan@ishan-ubuntu:~/PDC lab/lab5$./s
19BCE2250 - Ishan Jogalekar
--Star pattern using Openmp : Scheduling(Static)--
 *  *  *  *  *  * 
 *  *  *  *  *  * 
 *  *  *  *  *  * 
 *  *  *  *  *  * 
 *  *  *  *  *  * 
22:30:09-ishan@ishan-ubuntu:~/PDC lab/lab5$
```

2.**dynamic** - Dynamically assign one iteration to each threads

SOURCE CODE:

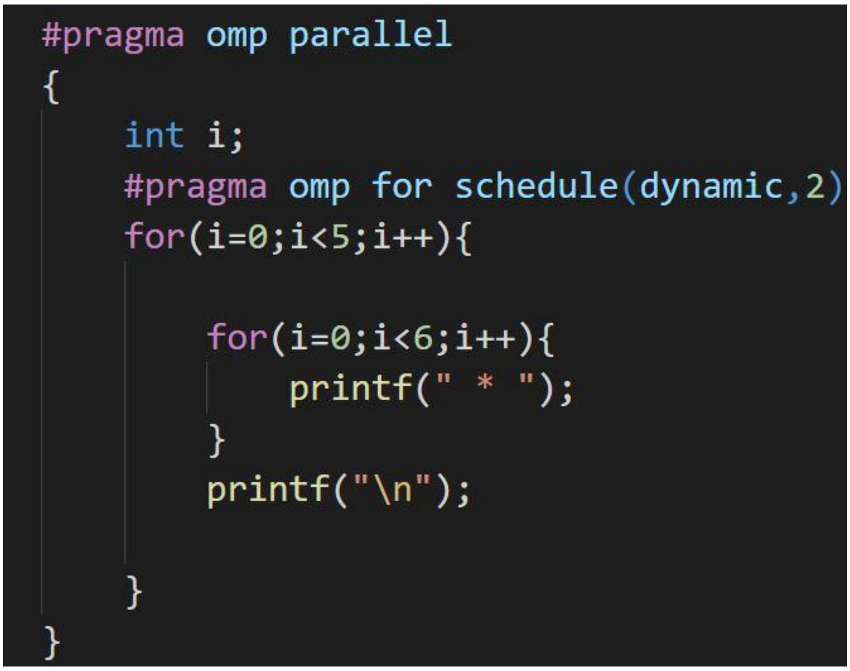
```
#include <stdio.h>
#include <omp.h>
void main(){
    printf("19BCE2250 - Ishan Jogalekar\n");
    printf("--Star pattern using Openmp : Scheduling(Dynamic)-\n");

    #pragma omp parallel
    {
        int i;
        #pragma omp for schedule(dynamic,2)
        for(i=0;i<5;i++){

            for(i=0;i<6;i++){
                printf(" * ");
            }
            printf("\n");

        }
    }
}
```

EXECUTION:

- <omp.h> is header file used to access Openmp functions within program.
- #pragma omp parallel starts parallel execution of process using OpenMP.
- Schedule is used as #pragram omp for schedule.
- Scheduling is a method in OpenMP to distribute iterations to different threads in for loop.
- Inside schedule clause, dynamic is used to assign loops to threads dynamically.
- Also, size of chunk gives as 2 to divide iterations into chunks.

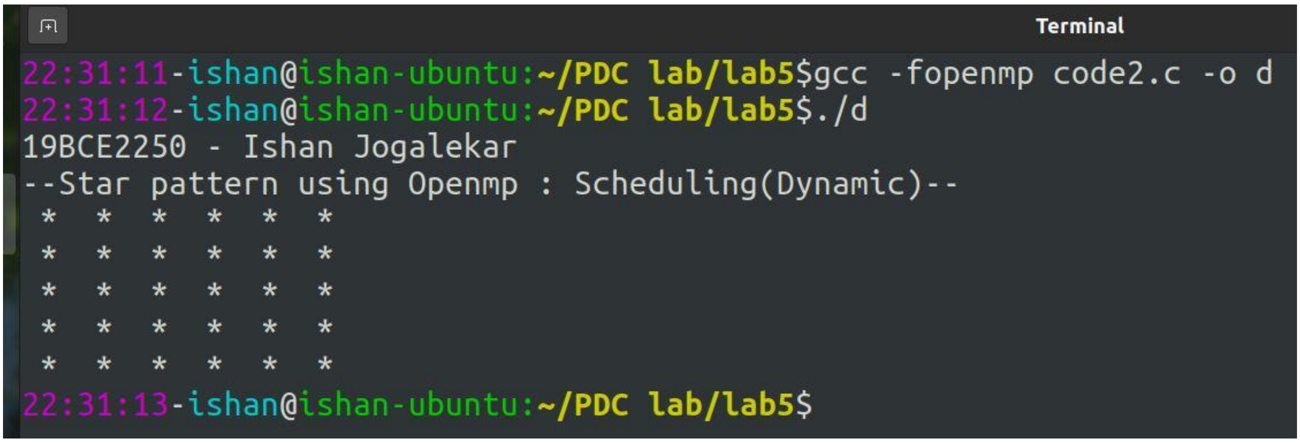
RESULTS:

1.Main logic:



```
#pragma omp parallel
{
    int i;
    #pragma omp for schedule(dynamic,2)
    for(i=0;i<5;i++){

        for(i=0;i<6;i++){
            printf(" * ");
        }
        printf("\n");

    }
}
```

2.Output:



```
22:31:11-ishan@ishan-ubuntu:~/PDC lab/lab5$gcc -fopenmp code2.c -o d
22:31:12-ishan@ishan-ubuntu:~/PDC lab/lab5$./d
19BCE2250 - Ishan Jogalekar
--Star pattern using Openmp : Scheduling(Dynamic)--
 *  *  *  *  *  *
 *  *  *  *  *  *
 *  *  *  *  *  *
 *  *  *  *  *  *
 *  *  *  *  *  *
22:31:13-ishan@ishan-ubuntu:~/PDC lab/lab5$
```