Parallel and Distributed Computing
CSE4001
Fall Semester 2020-21

Lab Assignment 5

ISHAAN OHRI
18BCE0265

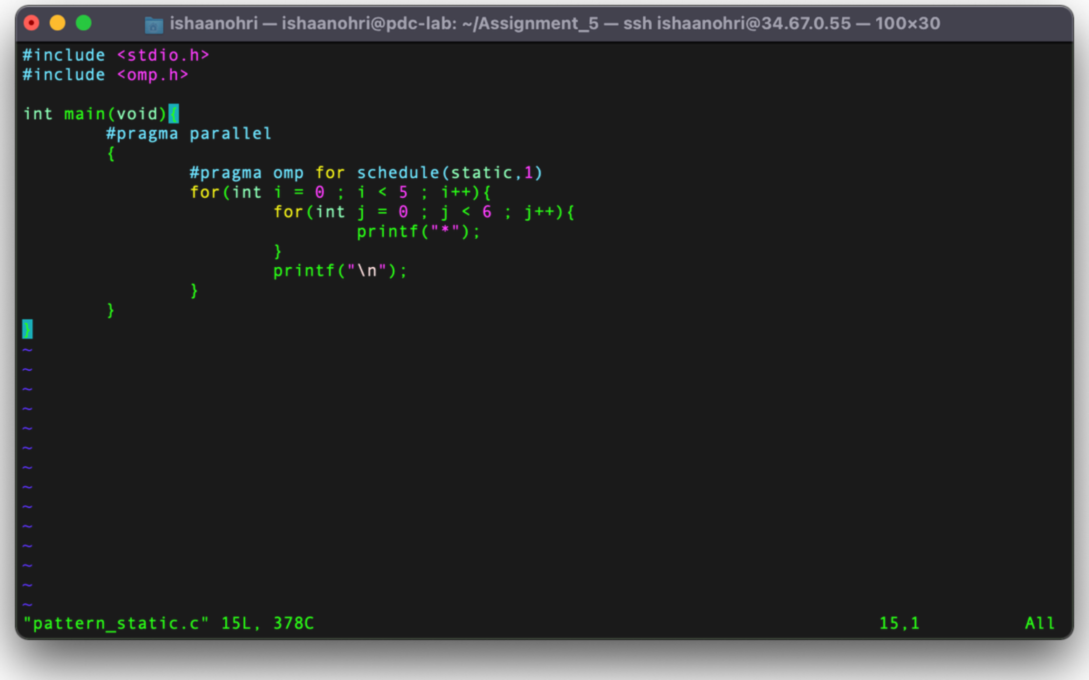
Aim:

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

```
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
```

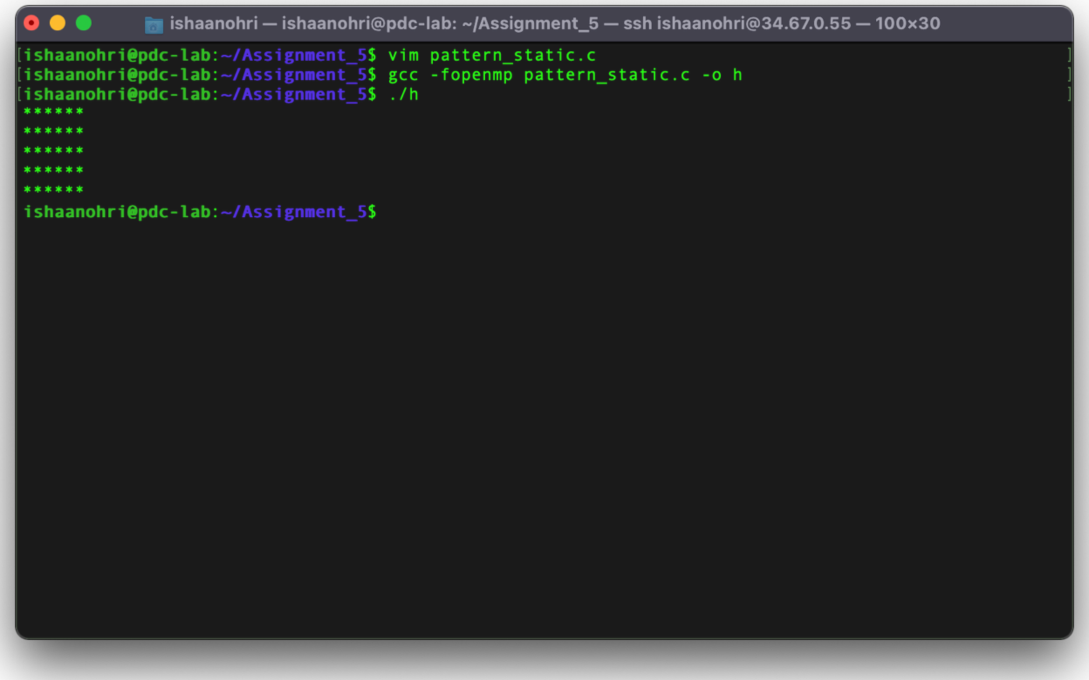
- Statically assign the loop iterations to threads
- Dynamically assign one iteration to each threads

Source Code:


ishaanohri — ishaanohri@pdc-lab: ~/Assignment_5 — ssh ishaanohri@34.67.0.55 — 100×30

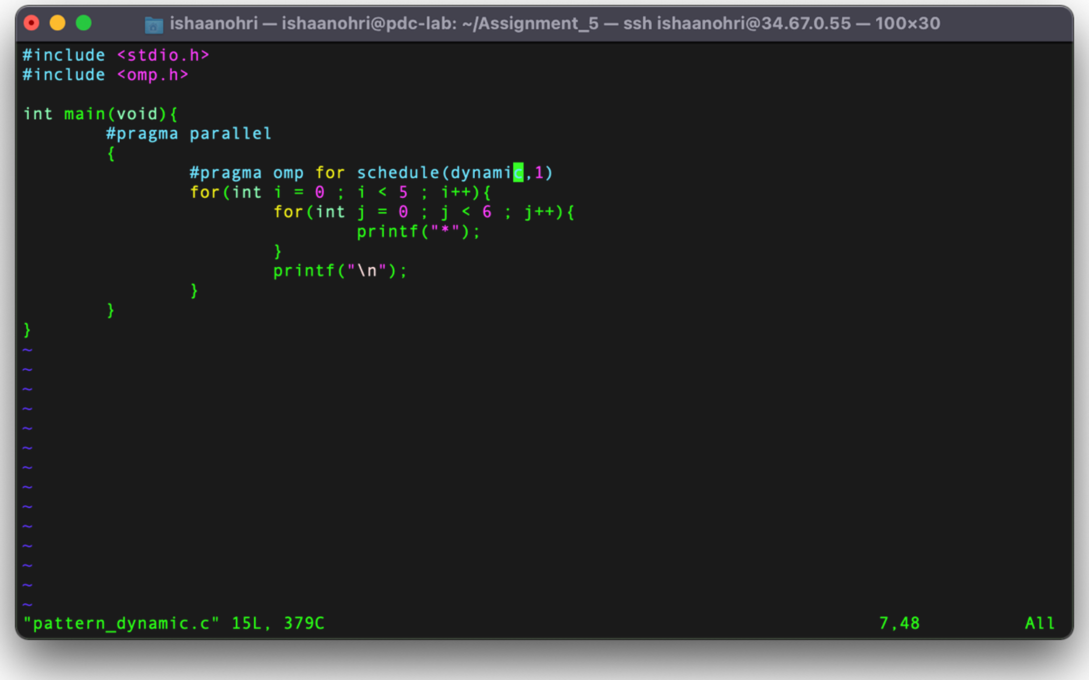
```
#include <stdio.h>
#include <omp.h>

int main(void){
        #pragma parallel
        {
                #pragma omp for schedule(static,1)
                for(int i = 0 ; i < 5 ; i++){
                        for(int j = 0 ; j < 6 ; j++){
                                printf("*");
                        }
                        printf("\n");
                }
        }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"pattern_static.c" 15L, 378C                                                    15,1          All
```

Execution:


ishaanohri — ishaanohri@pdc-lab: ~/Assignment_5 — ssh ishaanohri@34.67.0.55 — 100×30

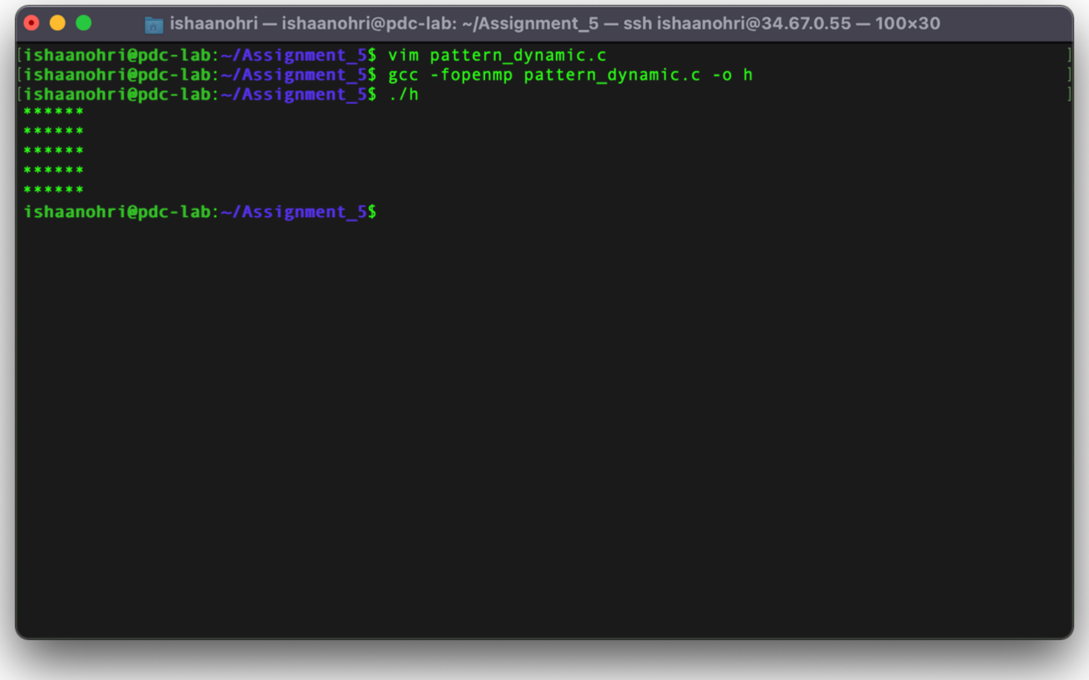
```
[ishaanohri@pdc-lab:~/Assignment_5$ vim pattern_static.c                                            ]
[ishaanohri@pdc-lab:~/Assignment_5$ gcc -fopenmp pattern_static.c -o h                              ]
[ishaanohri@pdc-lab:~/Assignment_5$ ./h                                                             ]
******
******
******
******
******
ishaanohri@pdc-lab:~/Assignment_5$
```

Source Code:


ishaanohri — ishaanohri@pdc-lab: ~/Assignment_5 — ssh ishaanohri@34.67.0.55 — 100×30

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

int main(void){
        #pragma parallel
        {
                #pragma omp for schedule(dynamic,1)
                for(int i = 0 ; i < 5 ; i++){
                        for(int j = 0 ; j < 6 ; j++){
                                printf("*");
                        }
                        printf("\n");
                }
        }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"pattern_dynamic.c" 15L, 379C                                         7,48          All
```

Execution:


ishaanohri — ishaanohri@pdc-lab: ~/Assignment_5 — ssh ishaanohri@34.67.0.55 — 100×30

```
[ishaanohri@pdc-lab:~/Assignment_5$ vim pattern_dynamic.c
[ishaanohri@pdc-lab:~/Assignment_5$ gcc -fopenmp pattern_dynamic.c -o h
[ishaanohri@pdc-lab:~/Assignment_5$ ./h
******
******
******
******
******
ishaanohri@pdc-lab:~/Assignment_5$
```

Result:

From this experiment, I was able to understand how threads are scheduled in OpenMP. There were two type of threads which were used in the experiment – *static* and *dynamic.*

In case of static scheduling, each thread is assigned chunk of iterations in round robin format. All iterations are equally divided among the threads. The integer specified as the 2nd argument will allocate chunk number of contiguous iterations to a particular thread.

In case of dynamic scheduling, each thread is assigned with a chunk of threads, then when each thread completes its iterations, it is assigned with the next set of iterations. The integer specified as the 2nd argument will allocate chunk number of contiguous iterations that are allocated to a thread at a time.

The above experiment was conducted and all results along with the source code have been attached above in the document. The experiment was assisted by Dr Deepak. I thank sir for his assistance.