

IT301 Assignment 6

NAME: SUYASH CHINTAWAR

ROLL NO.: 191IT109

TOPIC: OPENMP-TASKS DIRECTIVE

NOTE:

(a) The code for problem 1 is not attached as it is already given in the problem statement.

(b) In problem 1, as many printf statements will get printed for larger values of 'n', the printf statements have been commented for larger values of 'n'. (n=25)

Q1. Execute following code and observe the working of task directive. Check the result by removing if() clause with task.

SOLUTION:

Executing **with** 'if(n>5)':

Case 1 Output: n = 3

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ gcc -fopenmp task.c
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter the value of n:
3
Task Created by Thread 2
Task Created by Thread 2
Task Executed by Thread 2      a=1
Task Created by Thread 2
Task Executed by Thread 2      b=0
Task Executed by Thread 2      a=1
Task Created by Thread 2
Task Executed by Thread 2      b=1
Fib is 2
Time taken is 0.000986 s
```

Fig 1

Case 2 Output: n=25

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter the value of n:
25
Fib is 75025
Time taken is 0.067538 s
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

Fig 2

Executing **without** 'if(n>5)':

Case 1 Output: n = 3

```

ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ gcc -fopenmp task.c
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter the value of n:
3
Task Created by Thread 5
Task Created by Thread 2
Task Created by Thread 3
Task Executed by Thread 3      b=1
Task Executed by Thread 2      a=1
Task Created by Thread 1
Task Executed by Thread 1      b=0
Task Executed by Thread 5      a=1
Fib is 2
Time taken is 0.003680 s
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ █

```

Fig 3

Case 2 Output: n=25

```

ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter the value of n:
25
Fib is 75025
Time taken is 0.211459 s
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ █

```

Fig 4

Observations: The if statement states that do not do parallel execution when $n \leq 5$. In that case we must do serial execution. Now, to see the effects of this if statement, let's see case 1. In case 1, the purely parallel execution(fig 4) takes more time than partially parallel execution(fig 1) as parallel execution has a lot of overhead on smaller inputs. In case 2 though, when the if statement is not used(fig 6), it means that we have to do parallel execution for small inputs also, which in turn imposes a lot of time overhead as many calls to smaller inputs will be there. That's why the execution time in purely parallel execution increases(fig 4) than partially parallel execution(fig 2). Basically when the if statement becomes false, an undeferred task is generated.

Q2. Design a parallel program to find a given element in an unsorted array using Binary Search. Take a large number of elements up to the maximum possible size. Make use of openmp task directive. Use random function to initialise values.

SOLUTION:

(a) Serial Execution:

```
1  #include<bits/stdc++.h>
2  #include <omp.h>
3  using namespace std;
4
5  bool found = false;
6  int index_found = -1;
7
8  void binary_search(int a[],int l,int r,int target)
9  {
10     if(l<=r)
11     {
12         int mid = l + (r-l)/2;
13         if(a[mid]==target)
14         {
15             found = true;
16             index_found = mid;
17             return;
18         }
19         binary_search(a,l,mid-1,target);
20         binary_search(a,mid+1,r,target);
21     }
22 }
23
24 int main()
25 {
26     int n;
27     cout<<"Enter size of array: ";
28     cin>>n;
29
30     int a[n];
31     for(int i=0;i<n;i++)
32     {
33         a[i]=rand();
34     }
35     char c;
36     cout<<"Do you want element to be found in array?(y/n): ";
37     cin>>c;
38     int target,idx;
39     if(c=='y')
40     {
41         idx = rand()%n;
42         target = a[idx];
43     }
44     else
45     {
46         target = rand();
47     }
48     double t1,t2;
49     t1=omp_get_wtime();
50     binary_search(a,0,n-1,target);
51     t2=omp_get_wtime();
52     if(found) cout<<"Found at index "<<index_found<<"!\n";
53     else cout<<"Not found!!\n";
54     printf("Time taken is %f seconds \n",t2-t1);
55 }
```

Output:

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_serial.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 200
Do you want element to be found in array?(y/n): y
Found at index 176!
Time taken is 0.000005 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

For best, average and worst cases, target element has been fixed to $a[(n-1)/2]$, $a[(n-1)/4]$ and $a[n-1]$ respectively.

Now in best case, the element will be in the middle, output in that case is,

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_serial.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 499999!
Time taken is 0.000007 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

In average case, consider the element to be somewhere at the end of first quarter of array,

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_serial.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 249999!
Time taken is 0.008142 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

In worst case element will be at any of the extreme in array,

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_serial.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 999999!
Time taken is 0.016164 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

(b) Parallel Execution:

(continued...)

```

1  #include<bits/stdc++.h>
2  #include <sys/time.h>
3  #include <omp.h>
4  using namespace std;
5
6  bool found = false;
7  int index_found = -1;
8
9  void binary_search_serial(int a[],int l,int r,int target)
10 {
11     if(l<=r)
12     {
13         int mid = l + (r-l)/2;
14         if(a[mid]==target)
15         {
16             found = true;
17             index_found = mid;
18             return;
19         }
20         binary_search_serial(a,l,mid-1,target);
21         binary_search_serial(a,mid+1,r,target);
22     }
23 }
24
25 void binary_search_parallel(int a[],int l,int r,int target)
26 {
27     if(l<=r)
28     {
29         int mid = l + ((r-l)/2);
30         if(a[mid]==target)
31         {
32             found = true;
33             index_found = mid;
34             return;
35         }
36         if(r-l<1000) |
37         {
38             binary_search_serial(a,l,r,target);
39             return;
40         }
41         #pragma omp task shared(a,target) firstprivate(l,mid,r)
42         {
43             binary_search_parallel(a,l,mid-1,target);
44         }
45         #pragma omp task shared(a,target) firstprivate(l,mid,r)
46         {
47             binary_search_parallel(a,mid+1,r,target);
48         }
49         #pragma omp taskwait
50     }
51 }
52
53 int main()
54 {
55

```

```

53
54  int main()
55  {
56      int n;
57      cout<<"Enter size of array: ";
58      cin>>n;
59      int a[n];
60
61      for(int i=0;i<n;i++)
62      {
63          a[i]=rand();
64      }
65      char c;
66      cout<<"Do you want element to be found in array?(y/n): ";
67      cin>>c;
68      int target,idx;
69      if(c=='y')
70      {
71          idx = rand()%n;
72          target = a[idx];
73      }
74      else
75      {
76          target = rand();
77      }
78      double t1,t2;
79      t1=omp_get_wtime();
80      #pragma omp parallel shared(n,a,target)
81      {
82          #pragma omp single
83          {
84              binary_search_parallel(a,0,n-1,target);
85          }
86      }
87      t2=omp_get_wtime();
88      if(found) cout<<"Found at index "<<index_found<<"!\n";
89      else cout<<"Not found!!\n";
90      printf("Time taken is %f seconds \n",t2-t1);
91  }

```

Output:

```

ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 300
Do you want element to be found in array?(y/n): y
Found at index 190!
Time taken is 0.003742 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$

```

Best case:

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_parallel.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 499999!
Time taken is 0.002973 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

Average Case:

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_parallel.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 249999!
Time taken is 0.004441 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

Worst Case:

```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ g++ -fopenmp bin_search_parallel.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$ ./a.out
Enter size of array: 1000000
Found at index 999999!
Time taken is 0.007331 seconds
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT301/Assignment 6$
```

THANK YOU