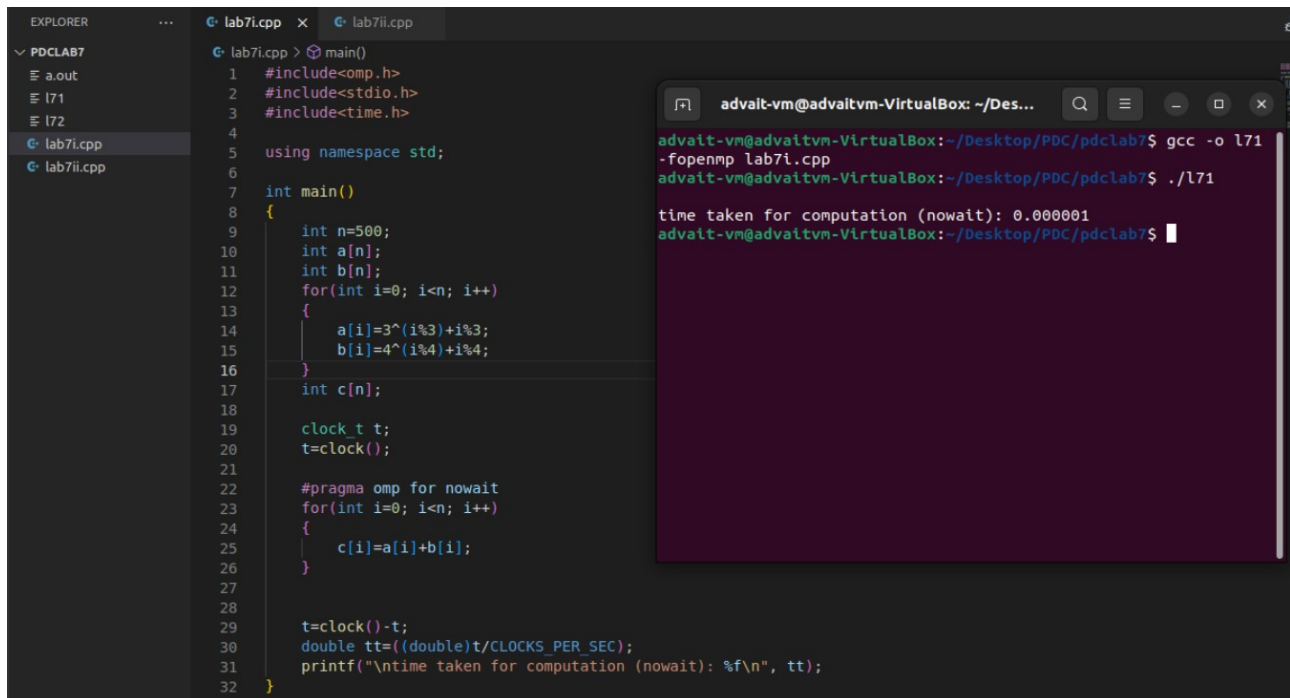


# PDC Lab 7

Advait Deochakke

20BCE1143

Q1 ) Parallel – nowait



The screenshot shows a code editor with two files: lab7i.cpp and lab7ii.cpp. The lab7i.cpp file contains the following code:

```
1 #include<omp.h>
2 #include<stdio.h>
3 #include<time.h>
4
5 using namespace std;
6
7 int main()
8 {
9     int n=500;
10    int a[n];
11    int b[n];
12    for(int i=0; i<n; i++)
13    {
14        a[i]=3^(i%3)+i%3;
15        b[i]=4^(i%4)+i%4;
16    }
17    int c[n];
18
19    clock_t t;
20    t=clock();
21
22    #pragma omp for nowait
23    for(int i=0; i<n; i++)
24    {
25        c[i]=a[i]+b[i];
26    }
27
28    t=clock()-t;
29    double tt=((double)t/CLOCKS_PER_SEC);
30    printf("\ntime taken for computation (nowait): %f\n", tt);
31 }
32
```

The terminal window shows the compilation and execution of the program:

```
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$ gcc -o l71 -fopenmp lab7i.cpp
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$ ./l71
time taken for computation (nowait): 0.000001
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$
```

Code:

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

```
using namespace std;
```

```
int main()
{
    int n=500;
    int a[n];
    int b[n];
    for(int i=0; i<n; i++)
    {
        a[i]=3^(i%3)+i%3;
        b[i]=4^(i%4)+i%4;
```

```

    }
    int c[n];

    clock_t t;
    t=clock();

    #pragma omp for nowait
    for(int i=0; i<n; i++)
    {
        c[i]=a[i]+b[i];
    }

    t=clock()-t;
    double tt=((double)t/CLOCKS_PER_SEC);
    printf("\ntime taken for computation (nowait): %f\n", tt);
}

```

Q2 ) Parallel – Barrier

The screenshot shows a code editor with a file named `lab7ii.cpp` and a terminal window. The code in the editor implements a parallel barrier using OpenMP. It defines two arrays `a` and `b` of size `n=500`, initializes them with specific values, and then computes the sum `c[i] = a[i] + b[i]` in parallel. A barrier is placed after the parallel loop to ensure all threads complete their work before the final time calculation. The terminal shows the compilation and execution of the program, resulting in a time taken for computation (barrier) of 0.000002 seconds.

```

EXPLORER
...
lab7i.cpp
lab7ii.cpp

lab7ii.cpp > main()
1  #include<omp.h>
2  #include<stdio.h>
3  #include<time.h>
4
5  using namespace std;
6
7  int main()
8  {
9      int n=500;
10     int a[n];
11     int b[n];
12     for(int i=0; i<n; i++)
13     {
14         a[i]=3^(i%3)+i%3;
15         b[i]=4^(i%4)+i%4;
16     }
17     int c[n];
18
19     clock_t t;
20     t=clock();
21
22     #pragma omp unroll full
23     for(int i=0; i<n; i++)
24     {
25         c[i]=a[i]+b[i];
26         #pragma omp barrier
27     }
28
29
30     t=clock()-t;
31     double tt=((double)t/CLOCKS_PER_SEC);
32     printf("\ntime taken for computation (barrier): %f\n", tt);
33 }

```

```

advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$ gcc -o l72
-fopenmp lab7ii.cpp
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$ ./l72

time taken for computation (barrier): 0.000002
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab7$

```

Code:

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

using namespace std;

int main()
{
    int n=500;
    int a[n];
    int b[n];
    for(int i=0; i<n; i++)
    {
        a[i]=3^(i%3)+i%3;
        b[i]=4^(i%4)+i%4;
    }
    int c[n];

    clock_t t;
    t=clock();

    #pragma omp unroll full
    for(int i=0; i<n; i++)
    {
        c[i]=a[i]+b[i];
        #pragma omp barrier
    }

    t=clock()-t;
    double tt=((double)t/CLOCKS_PER_SEC);
    printf("\ntime taken for computation (barrier): %f\n", tt);
}
```