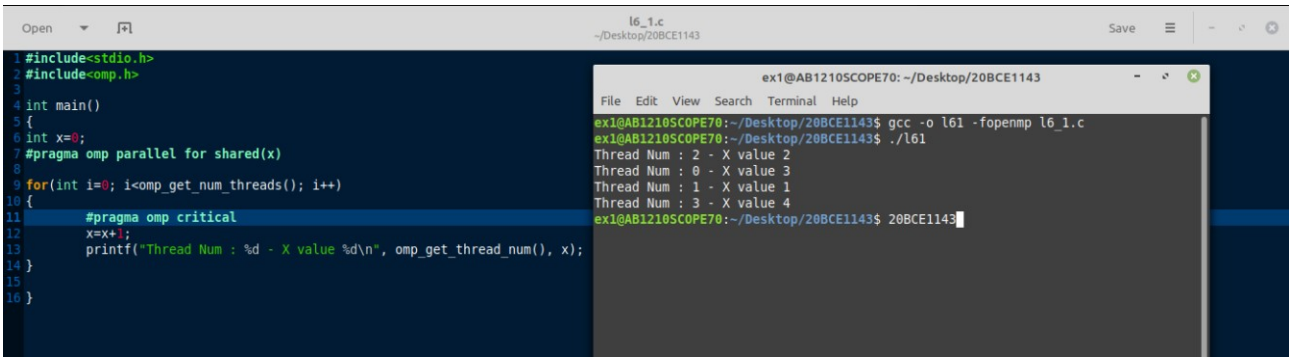
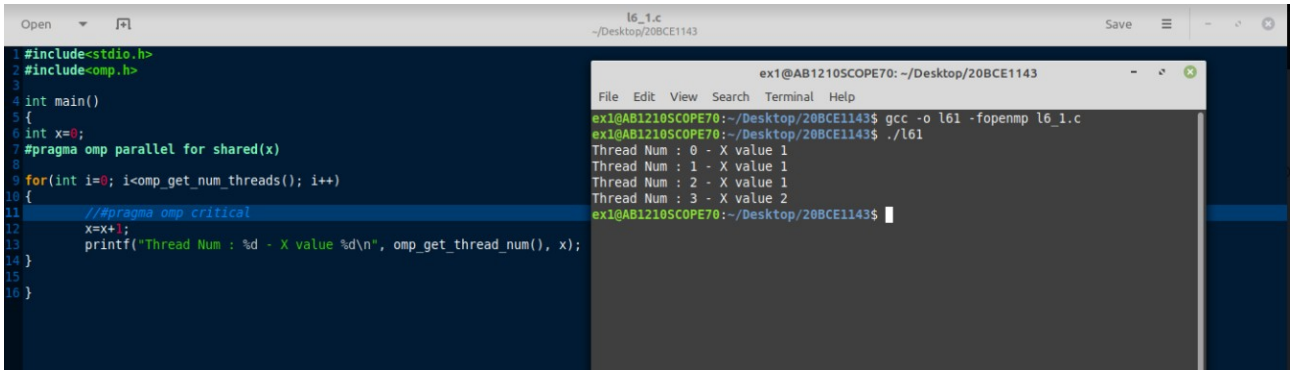
PDC Lab 6

Advait Deochakke

20BCE1143

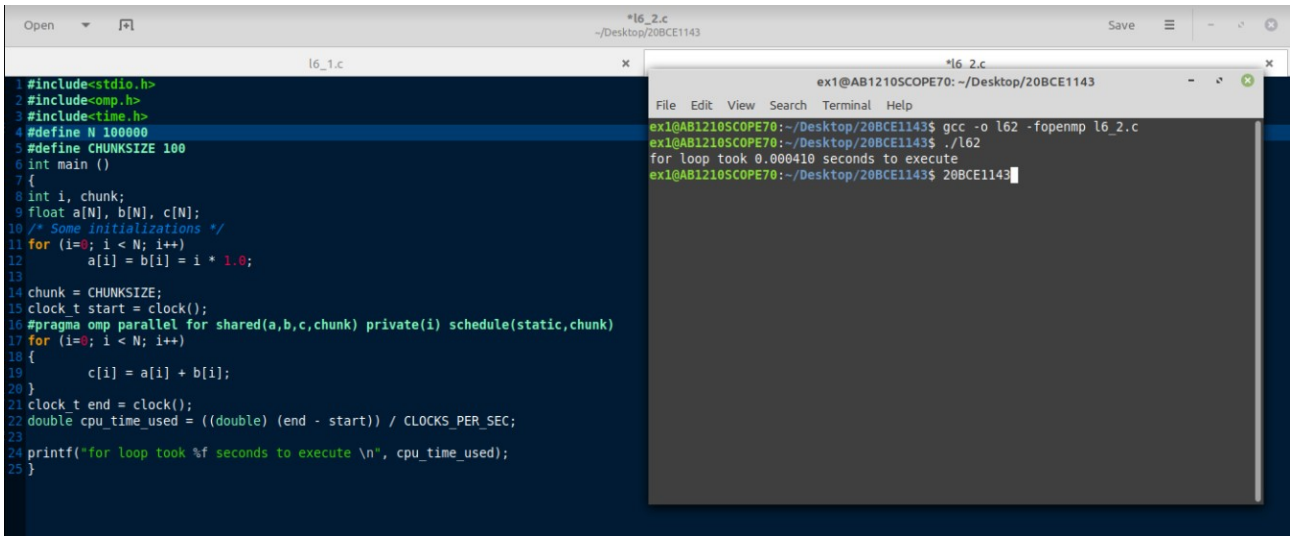
Sample critical without parallel program





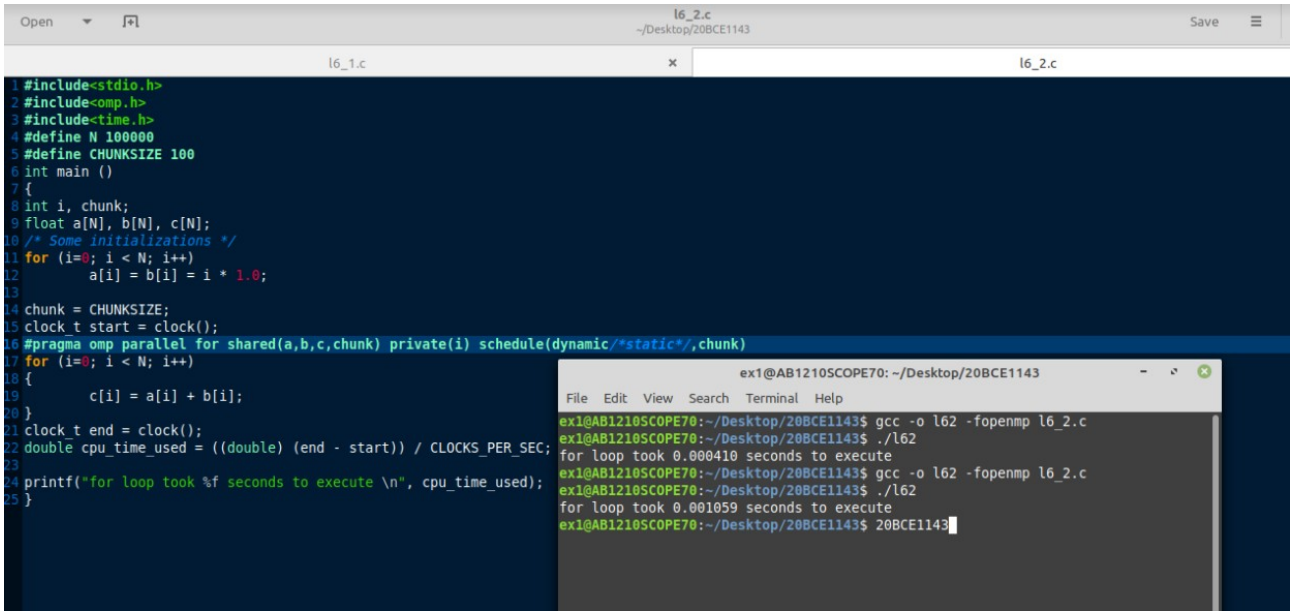

```c
#include<stdio.h>
#include<omp.h>

int main()
{
int x=0;
#pragma omp parallel for shared(x)

for(int i=0; i<omp_get_num_threads(); i++)
{
   #pragma omp critical
   x=x+1;
   printf("Thread Num : %d - X value %d\n", omp_get_thread_num(), x);
}

}
```

Sample static schedule program



(Code combined with next 4 questions)

Sample dynamic schedule program



(Code combined with next 3 questions)

Sample profile program with omp_get_wtime()

```c
#include<stdio.h>
#include<omp.h>
#include<time.h>
#define N 100000
#define CHUNKSIZE 100
int main ()
{
int i, chunk;
float a[N], b[N], c[N];
/* Some initializations */
for (i=0; i < N; i++)
    a[i] = b[i] = i * 1.0;

chunk = CHUNKSIZE;
//clock_t start = clock();
double start1 = omp_get_wtime();
#pragma omp parallel for shared(a,b,c,chunk) private(i) schedule(dynamic/*static*/,chunk)
for (i=0; i < N; i++)
{
    c[i] = a[i] + b[i];
}
//clock_t end = clock();
//double cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
double end1 = omp_get_wtime();

printf("for loop took %f seconds to execute \n", end1-start1);
}
```

```
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab6
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$ gcc -o l6iii -fopenmp l6_3.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$ ./l6iii
for loop took 0.000614 seconds to execute
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$
```

(Code combined with next 2 questions)

Sample program to set threads using directive clause
Sample program to set threads using omp_set_threads()
(Combined Together, as that is legal programming)

```c
#include<stdio.h>
#include<omp.h>
#include<time.h>
#define N 100000
#define CHUNKSIZE 100
int main ()
{
int i, chunk;
float a[N], b[N], c[N];
/* Some initializations */
for (i=0; i < N; i++)
    a[i] = b[i] = i * 1.0;

chunk = CHUNKSIZE;
#pragma omp sections
{
    omp_set_num_threads(8);
}
//clock_t start = clock();
double start1 = omp_get_wtime();
#pragma omp parallel for shared(a,b,c,chunk) private(i) schedule(dynamic/*static*/,chunk) num_threads(8)
for (i=0; i < N; i++)
{
    c[i] = a[i] + b[i];
}
//clock_t end = clock();
//double cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
double end1 = omp_get_wtime();

printf("for loop took %f seconds to execute \n", end1-start1);
}
```

```
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab6
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$ gcc -o l6iv -fopenmp l6_4.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$ ./l6iv
for loop took 0.005555 seconds to execute
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab6$
```

Code: Combined

```
#include<stdio.h>
#include<omp.h>
#include<time.h>
#define N 100000
#define CHUNKSIZE 100
int main ()
{
int i, chunk;
float a[N], b[N], c[N];
/* Some initializations */
for (i=0; i < N; i++)
  a[i] = b[i] = i * 1.0;

chunk = CHUNKSIZE;
#pragma omp sections
{
  omp_set_num_threads(8);
}
//clock_t start = clock();
double start1 = omp_get_wtime();
#pragma omp parallel for shared(a,b,c,chunk) private(i)
schedule(dynamic/*static*/,chunk) num_threads(8)
for (i=0; i < N; i++)
{
  c[i] = a[i] + b[i];
}
//clock_t end = clock();
//double cpu_time_used = ((double) (end - start)) /
CLOCKS_PER_SEC;
double end1 = omp_get_wtime();

printf("for loop took %f seconds to execute \n", end1-start1);
}
```