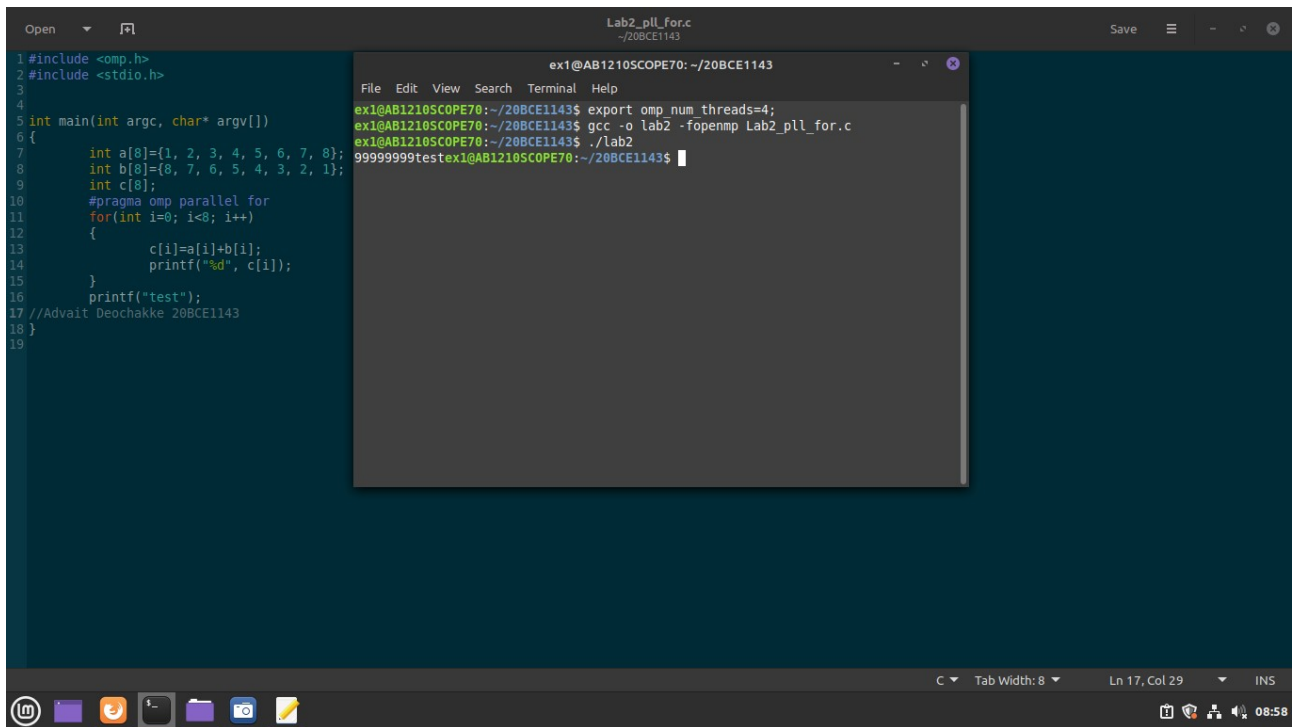


PDC Lab 2

Advait Deochakke

20BCE1143 – 2 August '22

Question 1: Array Addition using parallel for



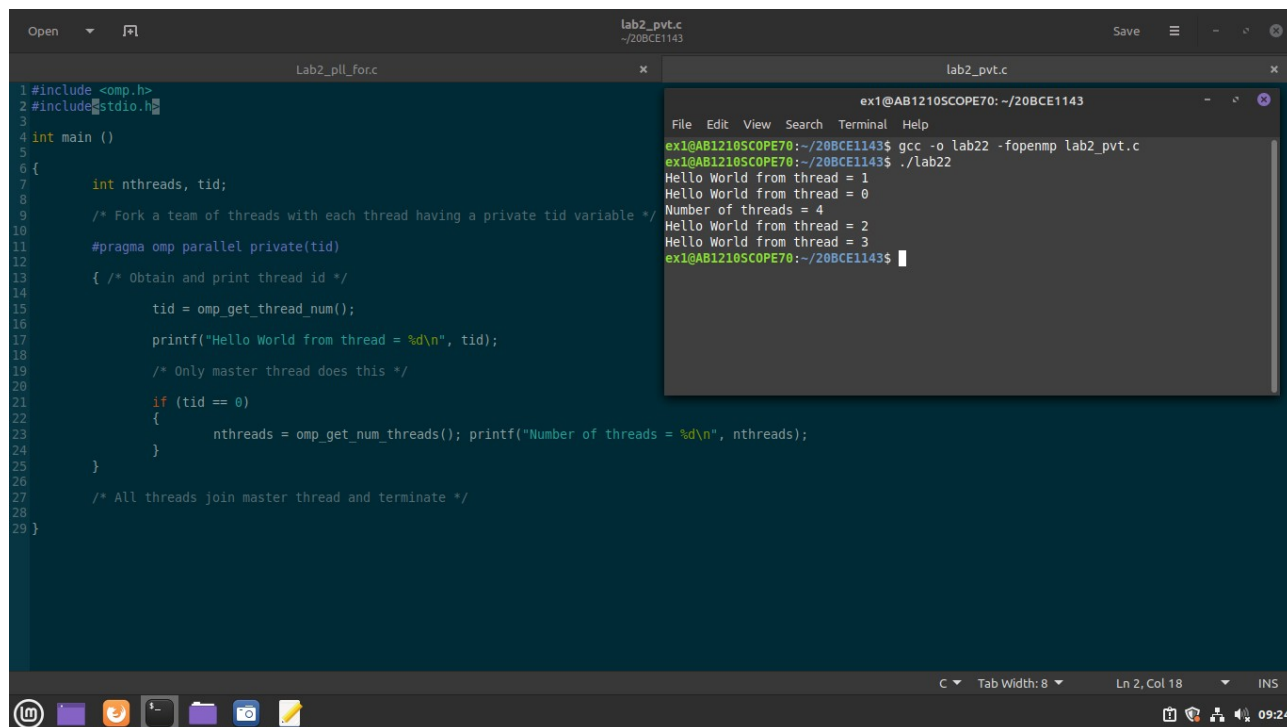
The screenshot shows a code editor with a C program and a terminal window. The C program, `Lab2_pll_for.c`, includes `<omp.h>` and `<stdio.h>`. It defines two arrays, `a` and `b`, and a third array `c`. A parallel for loop is used to calculate the sum of corresponding elements in `a` and `b` and store the result in `c`. The terminal window shows the execution of the program, where the output is `test` followed by the values of the array `c`.

```
1 #include <omp.h>
2 #include <stdio.h>
3
4
5 int main(int argc, char* argv[])
6 {
7     int a[8]={1, 2, 3, 4, 5, 6, 7, 8};
8     int b[8]={8, 7, 6, 5, 4, 3, 2, 1};
9     int c[8];
10    #pragma omp parallel for
11    for(int i=0; i<8; i++)
12    {
13        c[i]=a[i]+b[i];
14        printf("%d", c[i]);
15    }
16    printf("test");
17    //Advait Deochakke 20BCE1143
18 }
19
```

```
ex1@AB1210SCOPE70: ~/20BCE1143
ex1@AB1210SCOPE70:~/20BCE1143$ export omp_num_threads=4;
ex1@AB1210SCOPE70:~/20BCE1143$ gcc -o lab2 -fopenmp Lab2_pll_for.c
ex1@AB1210SCOPE70:~/20BCE1143$ ./lab2
99999999testex1@AB1210SCOPE70:~/20BCE1143$
```

```
#pragma omp parallel for
for(int i=0; i<8; i++)
{
    c[i]=a[i]+b[i];
    printf("%d", c[i]);
}
```

Question 2 & 3: Sample for Shared and Private Variable



The screenshot shows a code editor with two tabs: 'Lab2_pll_for.c' and 'lab2_pvt.c'. The 'lab2_pvt.c' tab is active, displaying the following C code:

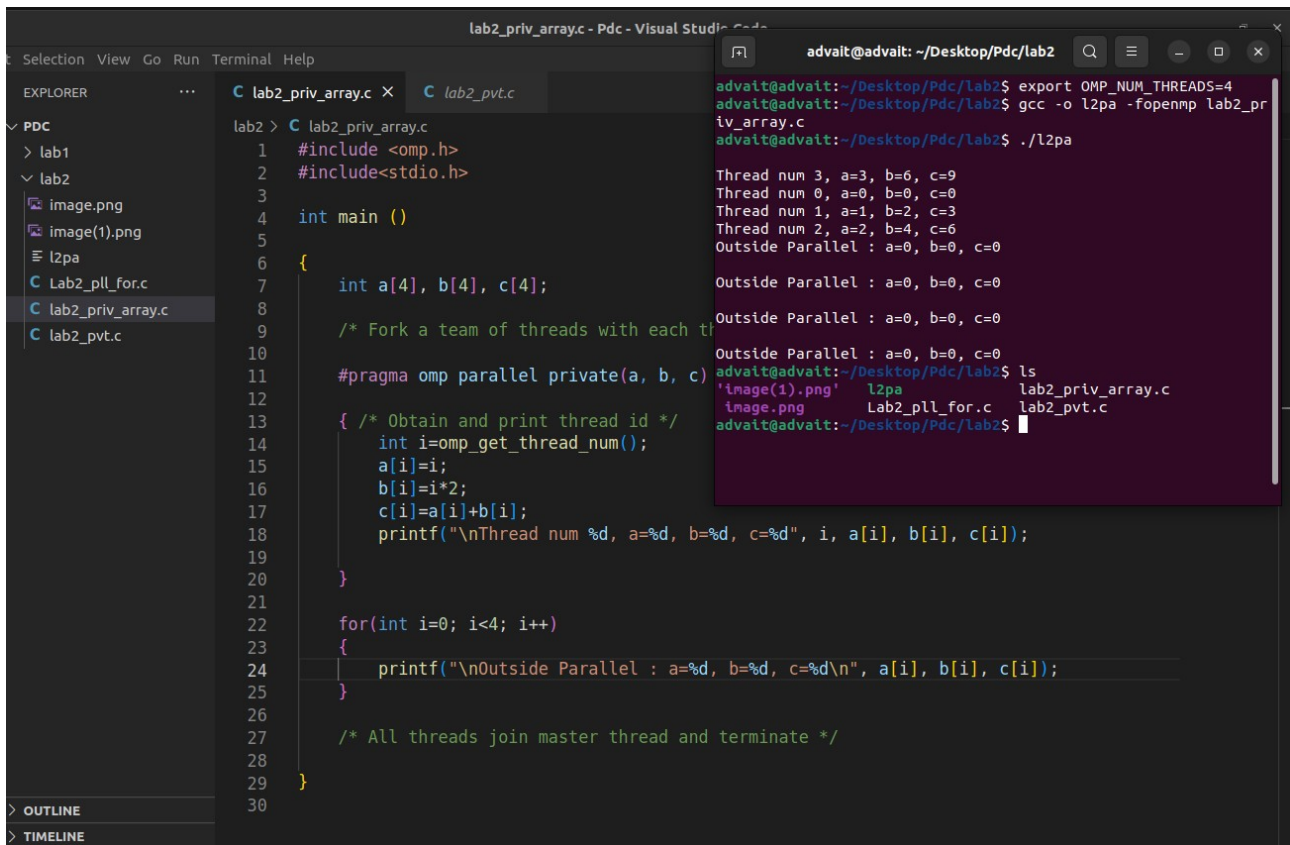
```
1 #include <omp.h>
2 #include <stdio.h>
3
4 int main ()
5 {
6     int nthreads, tid;
7
8     /* Fork a team of threads with each thread having a private tid variable */
9
10    #pragma omp parallel private(tid)
11    { /* Obtain and print thread id */
12
13        tid = omp_get_thread_num();
14
15        printf("Hello World from thread = %d\n", tid);
16
17        /* Only master thread does this */
18
19        if (tid == 0)
20        {
21            nthreads = omp_get_num_threads(); printf("Number of threads = %d\n", nthreads);
22        }
23    }
24
25    /* All threads join master thread and terminate */
26
27 }
```

Below the code editor, a terminal window is open, showing the execution of the program. The terminal output is as follows:

```
ex1@AB1210SCOPE70: ~/20BCE1143
ex1@AB1210SCOPE70:~/20BCE1143$ gcc -o lab22 -fopenmp lab2_pvt.c
ex1@AB1210SCOPE70:~/20BCE1143$ ./lab22
Hello World from thread = 1
Hello World from thread = 0
Number of threads = 4
Hello World from thread = 2
Hello World from thread = 3
ex1@AB1210SCOPE70:~/20BCE1143$
```

```
#include <omp.h>
#include <stdio.h>
int main ()
{
    int nthreads, tid;
    /* Fork a team of threads with each thread having a private tid
    variable */
    #pragma omp parallel private(tid)
    { /* Obtain and print thread id */
        tid = omp_get_thread_num();
        printf("Hello World from thread = %d\n", tid);
        /* Only master thread does this */
        if (tid == 0)
        {
            nthreads = omp_get_num_threads(); printf("Number of threads = %d\
n", nthreads);
        }
    }
    /* All threads join master thread and terminate */
}
```

Question 4: Array addition using parallel for with a, b, c as private arrays



```
lab2_priv_array.c - Pdc - Visual Studio Code
Selection View Go Run Terminal Help
EXPLORER
PDC
lab1
lab2
image.png
image(1).png
l2pa
Lab2_pll_for.c
lab2_priv_array.c
lab2_pvt.c
lab2 > C lab2_priv_array.c
1 #include <omp.h>
2 #include <stdio.h>
3
4 int main ()
5 {
6     int a[4], b[4], c[4];
7
8     /* Fork a team of threads with each thread having a private tid
9     variable */
10
11     #pragma omp parallel private(a, b, c)
12     { /* Obtain and print thread id */
13         int i=omp_get_thread_num();
14         a[i]=i;
15         b[i]=i*2;
16         c[i]=a[i]+b[i];
17         printf("\nThread num %d, a=%d, b=%d, c=%d", i, a[i], b[i], c[i]);
18     }
19
20     for(int i=0; i<4; i++)
21     {
22         printf("\nOutside Parallel : a=%d, b=%d, c=%d\n", a[i], b[i], c[i]);
23     }
24
25     /* All threads join master thread and terminate */
26 }
27
28
29
30
advalit@advalit: ~/Desktop/Pdc/lab2$ export OMP_NUM_THREADS=4
advalit@advalit:~/Desktop/Pdc/lab2$ gcc -o l2pa -fopenmp lab2_priv_array.c
advalit@advalit:~/Desktop/Pdc/lab2$ ./l2pa
Thread num 3, a=3, b=6, c=9
Thread num 0, a=0, b=0, c=0
Thread num 1, a=1, b=2, c=3
Thread num 2, a=2, b=4, c=6
Outside Parallel : a=0, b=0, c=0
Outside Parallel : a=0, b=0, c=0
Outside Parallel : a=0, b=0, c=0
advalit@advalit:~/Desktop/Pdc/lab2$ ls
'lab2_priv_array.c'  l2pa  lab2_pvt.c
'lab2_pll_for.c'  lab2_priv_array.c
'lab2_pvt.c'  lab2_pll_for.c
advalit@advalit:~/Desktop/Pdc/lab2$
```

```
#include <omp.h>
#include <stdio.h>
int main ()
{
    int a[4], b[4], c[4];
    /* Fork a team of threads with each thread having a private tid
    variable */
    #pragma omp parallel private(a, b, c)
    { /* Obtain and print thread id */
        int i=omp_get_thread_num();
        a[i]=i;
        b[i]=i*2;
        c[i]=a[i]+b[i];
        printf("\nThread num %d, a=%d, b=%d, c=%d", i, a[i], b[i], c[i]);
    }
    for(int i=0; i<4; i++)
    {
        printf("\nOutside Parallel : a=%d, b=%d, c=%d\n", a[i], b[i], c[i]);
    }

    /* All threads join master thread and terminate */
}
```

Question 5: Parallelize addition and subtraction of two integer variables a and b

The screenshot shows the Visual Studio Code interface with the file explorer on the left, the source code editor in the center, and a terminal window on the right.

Source Code (lab2_as.c):

```
1 int main ()
2 {
3     int a=2, b=2;
4     printf("\na=%d", a);
5     printf("\nb=%d", b);
6     printf("\nParallel adding subtracting")
7
8     #pragma omp parallel
9     {
10         int i=omp_get_thread_num();
11         if(i%2)
12         {
13             a=a+b;
14             printf("\na=%d", a);
15         }
16         else
17         {
18             b=b-a;
19             printf("\nb=%d", b);
20         }
21     }
22 }
```

Terminal Output:

```
advait@advait: ~/Desktop/Pdc/lab2$ export OMP_NUM_THREADS=2
advait@advait:~/Desktop/Pdc/lab2$ gcc -o l2a -fopenmp lab2_as.c
advait@advait:~/Desktop/Pdc/lab2$ ./l2a
a=2
b=2
Parallel adding subtracting
a=4
b=-2advait@advait:~/Desktop/Pdc/lab2$ ./l2a
a=2
b=2
Parallel adding subtracting
a=4
b=-2advait@advait:~/Desktop/Pdc/lab2$ ./l2a
a=2
b=2
Parallel adding subtracting
a=4
b=-2advait@advait:~/Desktop/Pdc/lab2$ ./l2a
a=2
b=2
Parallel adding subtracting
a=4
b=0
```

```
#pragma omp parallel
```

```
{
    int i=omp_get_thread_num();
    if(i%2)
    {
        a=a+b;
        printf("\na=%d", a);
    }
    else
    {
        b=b-a;
        printf("\nb=%d", b);
    }
}
```