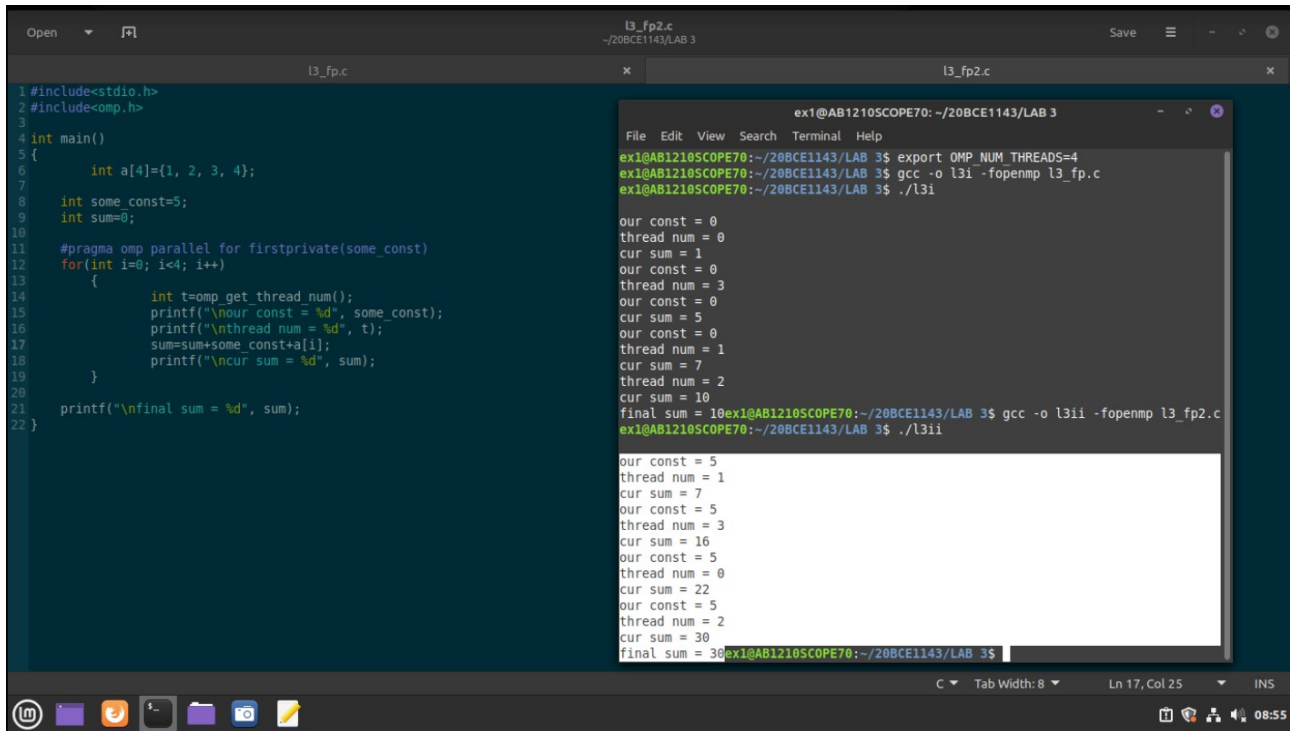


PDC Lab 3 Report

Advait Deochakke
20BCE1143

1) First Private



The screenshot shows a code editor with a C program named `l3_fp2.c` and a terminal window displaying its execution output. The C program uses OpenMP to parallelize a loop with the `firstprivate` clause. The terminal shows the output for 4 threads, with the final sum being 10.

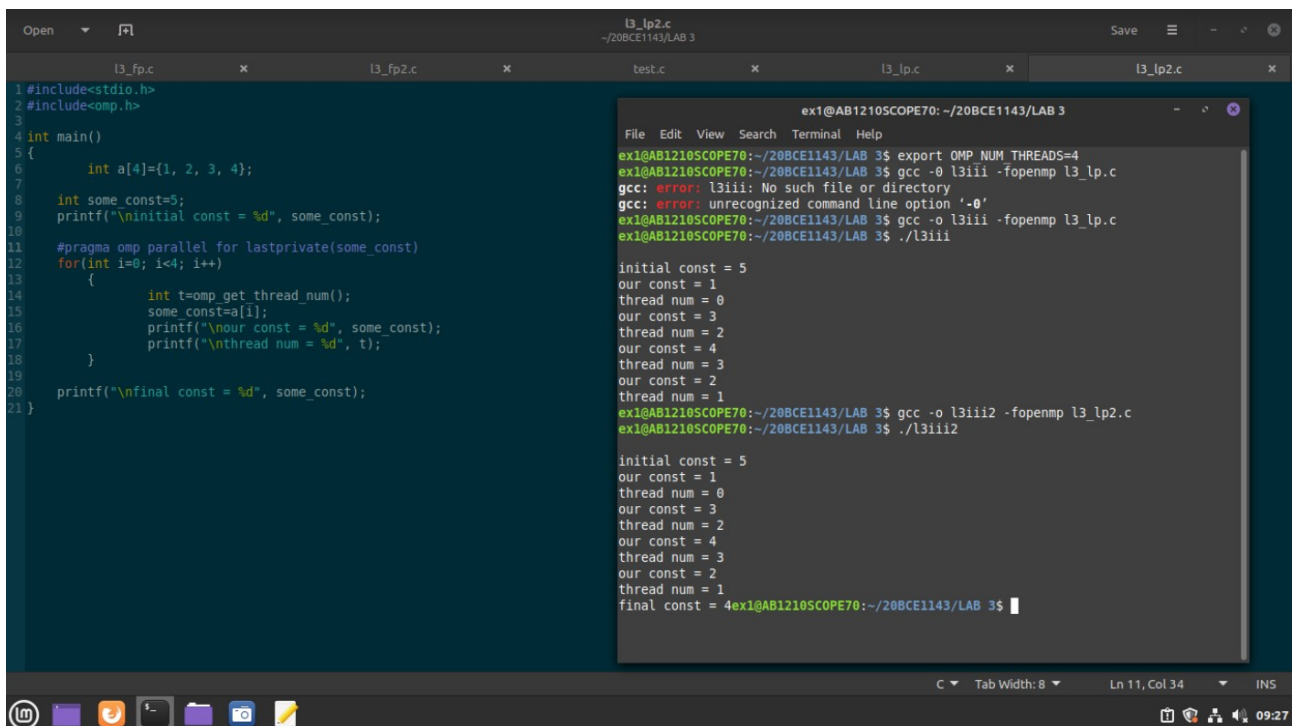
```
1 #include<stdio.h>
2 #include<omp.h>
3
4 int main()
5 {
6     int a[4]={1, 2, 3, 4};
7
8     int some_const=5;
9     int sum=0;
10
11     #pragma omp parallel for firstprivate(some_const)
12     for(int i=0; i<4; i++)
13     {
14         int t=omp_get_thread_num();
15         printf("\nour const = %d", some_const);
16         printf("\nthread num = %d", t);
17         sum=sum+some_const+a[i];
18         printf("\ncur sum = %d", sum);
19     }
20
21     printf("\nfinal sum = %d", sum);
22 }
```

```
ex1@AB1210SCOPE70: ~/20BCE1143/LAB 3
File Edit View Search Terminal Help
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ export OMP_NUM_THREADS=4
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ gcc -o l3i -fopenmp l3_fp.c
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ ./l3i

our const = 0
thread num = 0
cur sum = 1
our const = 0
thread num = 3
our const = 0
cur sum = 5
our const = 0
thread num = 1
cur sum = 7
our const = 0
thread num = 2
cur sum = 10
final sum = 10ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ gcc -o l3ii -fopenmp l3_fp.c
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ ./l3ii

our const = 5
thread num = 1
cur sum = 7
our const = 5
thread num = 3
cur sum = 16
our const = 5
thread num = 0
cur sum = 22
our const = 5
thread num = 2
cur sum = 30
final sum = 30ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$
```

2) Last Private



The screenshot shows a code editor with a C program named `l3_lp2.c` and its execution output in a terminal window.

```
#include<stdio.h>
#include<omp.h>
int main()
{
    int a[4]={1, 2, 3, 4};
    int some_const=5;
    printf("\ninitial const = %d", some_const);
    #pragma omp parallel for lastprivate(some_const)
    for(int i=0; i<4; i++)
    {
        int t=omp_get_thread_num();
        some_const=a[i];
        printf("\nour const = %d", some_const);
        printf("\nthread num = %d", t);
    }
    printf("\nfinal const = %d", some_const);
}
```

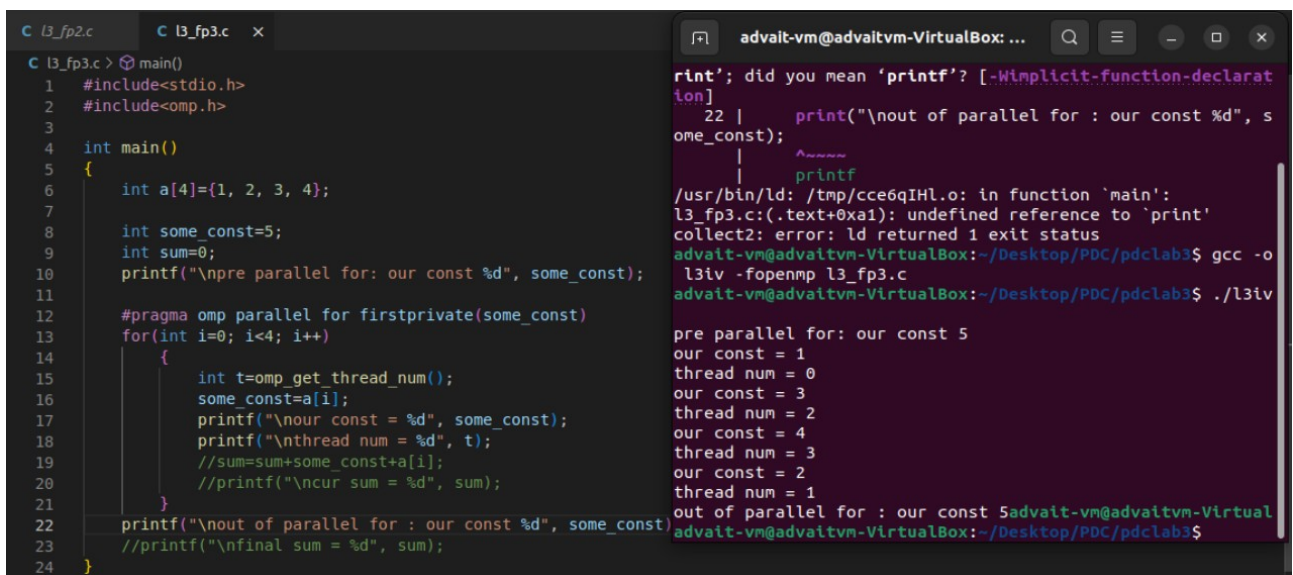
The terminal output shows the execution of the program with 4 threads. The initial constant is 5. Each thread prints its own value of `some_const` (which is `a[i]`) and its thread number. The final constant is 4.

```
ex1@AB1210SCOPE70: ~/20BCE1143/LAB 3
File Edit View Search Terminal Help
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ export OMP_NUM_THREADS=4
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ gcc -O3 l3iii -fopenmp l3_lp.c
gcc: error: l3iii: No such file or directory
gcc: error: unrecognized command line option '-O'
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ gcc -O3 l3iii -fopenmp l3_lp.c
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ ./l3iii

initial const = 5
our const = 1
thread num = 0
our const = 3
thread num = 2
our const = 4
thread num = 3
our const = 2
thread num = 1
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ gcc -O3 l3iii2 -fopenmp l3_lp2.c
ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$ ./l3iii2

initial const = 5
our const = 1
thread num = 0
our const = 3
thread num = 2
our const = 4
thread num = 3
our const = 2
thread num = 1
final const = 4ex1@AB1210SCOPE70:~/20BCE1143/LAB 3$
```

3) First Private with modification to privated variable



The screenshot shows a code editor with a C program named `l3_fp3.c` and its execution output in a terminal window.

```
#include<stdio.h>
#include<omp.h>
int main()
{
    int a[4]={1, 2, 3, 4};
    int some_const=5;
    int sum=0;
    printf("\npre parallel for: our const %d", some_const);
    #pragma omp parallel for firstprivate(some_const)
    for(int i=0; i<4; i++)
    {
        int t=omp_get_thread_num();
        some_const=a[i];
        printf("\nour const = %d", some_const);
        printf("\nthread num = %d", t);
        //sum=sum+some_const+a[i];
        //printf("\nour sum = %d", sum);
    }
    printf("\nout of parallel for : our const %d", some_const);
    //printf("\nfinal sum = %d", sum);
}
```

The terminal output shows the execution of the program with 4 threads. The initial constant is 5. Each thread prints its own value of `some_const` (which is `a[i]`) and its thread number. The final constant is 5.

```
advait-vm@advaitvm-VirtualBox: ...
rint'; did you mean 'printf'? [-Wimplicit-function-declarat
ion]
22 |     printf("\nout of parallel for : our const %d", s
some_const);
    |     printf
/usr/bin/ld: /tmp/cce6qIHL.o: in function 'main':
l3_fp3.c:(.text+0xa1): undefined reference to 'print'
collect2: error: ld returned 1 exit status
advait-vm@advaitvm-VirtualBox: ~/Desktop/PDC/pdclab3$ gcc -O3
l3iv -fopenmp l3_fp3.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab3$ ./l3iv

pre parallel for: our const 5
our const = 1
thread num = 0
our const = 3
thread num = 2
our const = 4
thread num = 3
our const = 2
thread num = 1
out of parallel for : our const 5advait-vm@advaitvm-Virtual
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab3$
```

4) Sum of array c[i]

```
C l3_fp2.c C l3_fp3.c C l3_s.c X  
C l3_s.c > main()  
1 #include<stdio.h>  
2 #include<omp.h>  
3  
4 int main()  
5 {  
6     int a[40]={1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1,  
7  
8     int sum=0;  
9  
10    #pragma omp parallel for reduction(+:sum)  
11    for(int i=0; i<sizeof(a)/sizeof(int); i++)  
12        {  
13            sum=sum+a[i];  
14        }  
15  
16    printf("\nfinal sum = %d", sum);  
17 }  
18
```

adavt-virtual-machine@adavtvm-VirtualBox: ... 🔍 ☰ ⌵ □ ✕

```
adavt-vm@adavtvm-VirtualBox:~/Desktop/PDC/pdcclab3$ gcc -o  
l3v -fopenmp l3_s.c  
adavt-vm@adavtvm-VirtualBox:~/Desktop/PDC/pdcclab3$ ./l3v  
  
final sum = 100adavt-vm@adavtvm-VirtualBox:~/Desktop/PDC/  
pdcclab3$
```