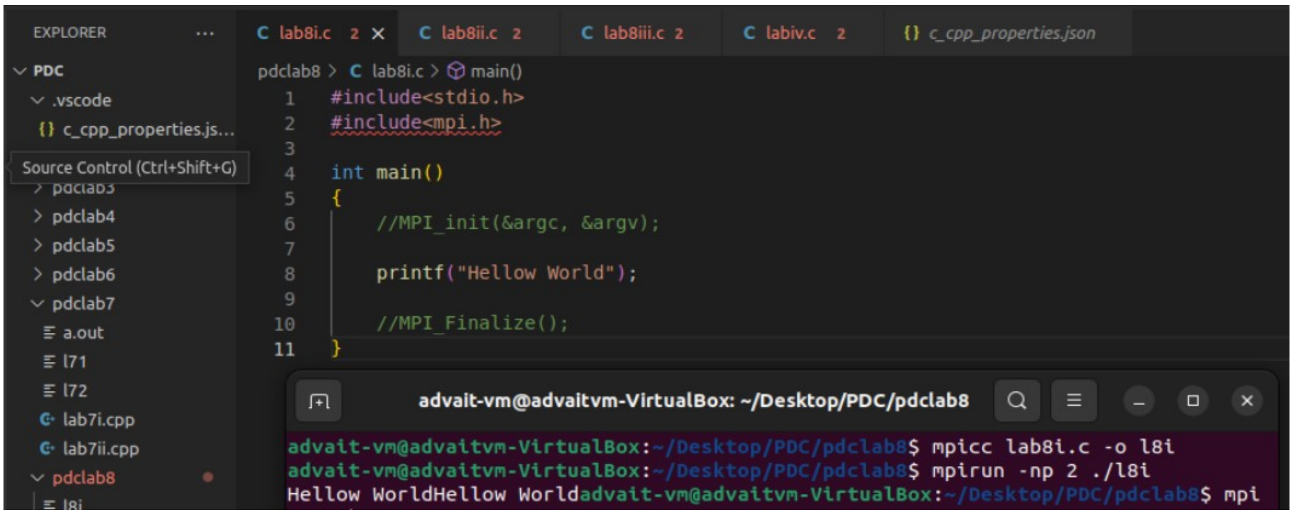
PDC Lab 8

Advait Deochakke

20BCE1143

Sample Hello World



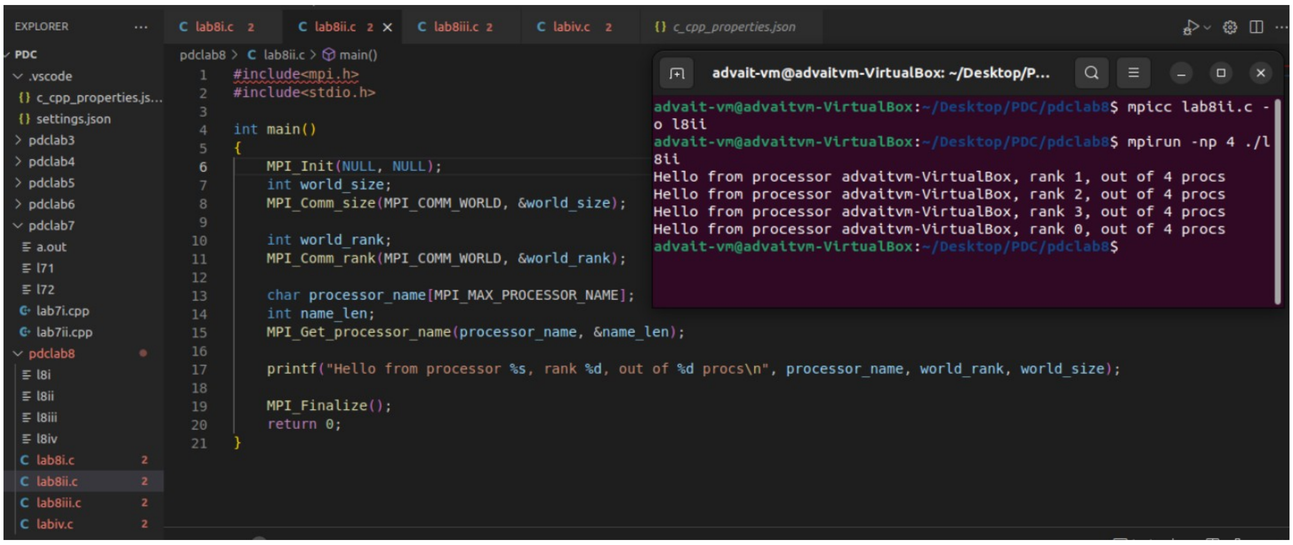
```
#include<stdio.h>
#include<mpi.h>

int main()
{
    //MPI_init(&argc, &argv);

    printf("Hellow World");

    //MPI_Finalize();
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpicc lab8i.c -o l8i
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpirun -np 2 ./l8i
Hellow WorldHellow Worldadvait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpi
```

Code:

```
#include<stdio.h>
#include<mpi.h>

int main()
{
    //MPI_init(&argc, &argv);

    printf("Hellow World");

    //MPI_Finalize();
}
```

Prink rank, world size and processor name



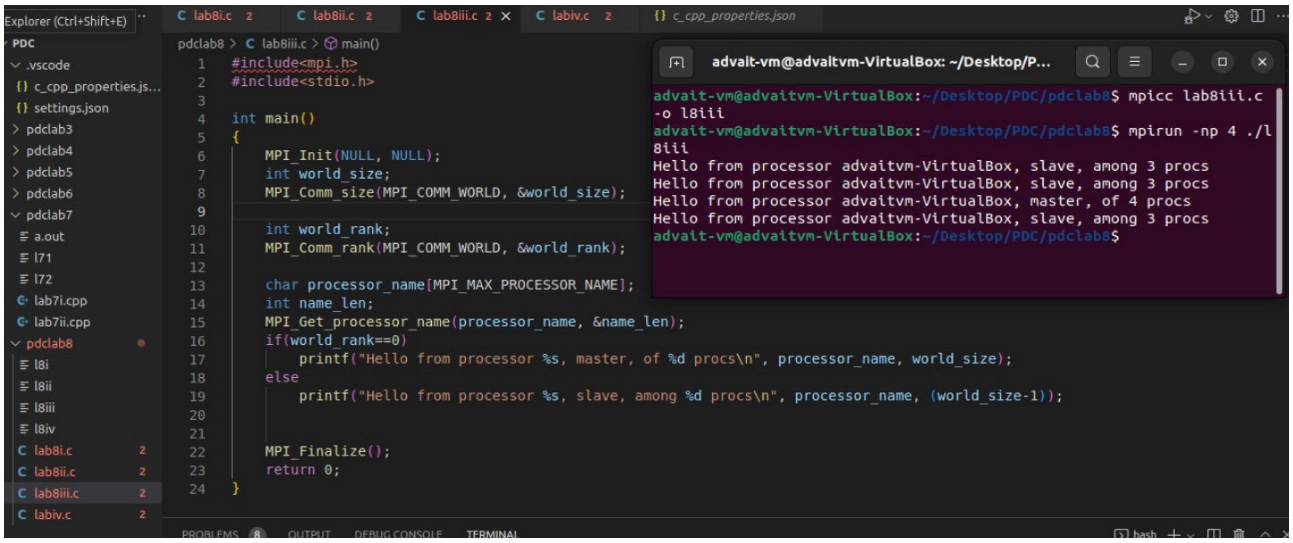
```
#include<mpi.h>
#include<stdio.h>

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    printf("Hello from processor %s, rank %d, out of %d procs\n", processor_name, world_rank, world_size);

    MPI_Finalize();
    return 0;
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpicc lab8ii.c -
o l8ii
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpirun -np 4 ./l
8ii
Hello from processor advaitvm-VirtualBox, rank 1, out of 4 procs
Hello from processor advaitvm-VirtualBox, rank 2, out of 4 procs
Hello from processor advaitvm-VirtualBox, rank 3, out of 4 procs
Hello from processor advaitvm-VirtualBox, rank 0, out of 4 procs
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$
```

Code:

```
#include<mpi.h>
#include<stdio.h>

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    printf("Hello from processor %s, rank %d, out of %d procs\
n", processor_name, world_rank, world_size);

    MPI_Finalize();
    return 0;
}
```

Master prints "I am Master", Worker prints "I am worker"



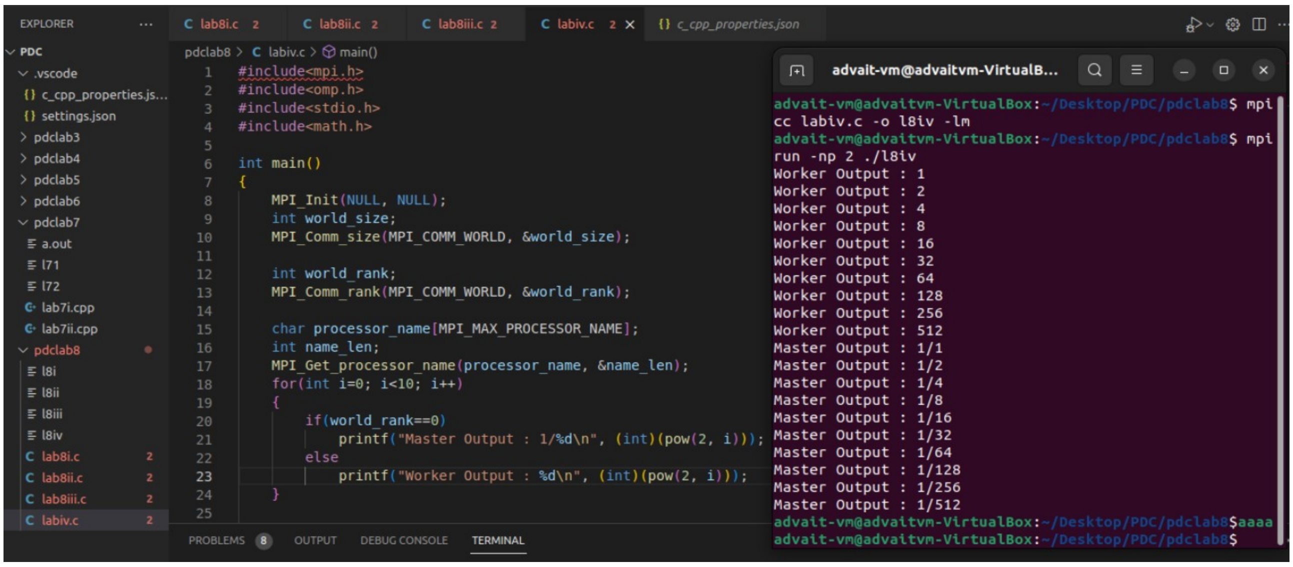
```c
#include<mpi.h>
#include<stdio.h>

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    if(world_rank==0)
        printf("Hello from processor %s, master, of %d procs\n", processor_name, world_size);
    else
        printf("Hello from processor %s, slave, among %d procs\n", processor_name, (world_size-1));


    MPI_Finalize();
    return 0;
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpicc lab8iii.c
-o l8iii
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpirun -np 4 ./l
8iii
Hello from processor advaitvm-VirtualBox, slave, among 3 procs
Hello from processor advaitvm-VirtualBox, slave, among 3 procs
Hello from processor advaitvm-VirtualBox, master, of 4 procs
Hello from processor advaitvm-VirtualBox, slave, among 3 procs
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$
```

Code:

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

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    if(world_rank==0)
        printf("Hello from processor %s, master, of %d procs\n", 
processor_name, world_size);
    else
        printf("Hello from processor %s, slave, among %d procs\n", 
processor_name, (world_size-1));


    MPI_Finalize();
    return 0;
}
```

Master generates 1/2,1/4,1/8,1/16...1/n; Worker generates 2,4,8,16...n



```
#include<mpi.h>
#include<omp.h>
#include<stdio.h>
#include<math.h>

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    for(int i=0; i<10; i++)
    {
        if(world_rank==0)
            printf("Master Output : 1/%d\n", (int)(pow(2, i)));
        else
            printf("Worker Output : %d\n", (int)(pow(2, i)));
    }
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpicc labiv.c -o l8iv -lm
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$ mpirun -np 2 ./l8iv
Worker Output : 1
Worker Output : 2
Worker Output : 4
Worker Output : 8
Worker Output : 16
Worker Output : 32
Worker Output : 64
Worker Output : 128
Worker Output : 256
Worker Output : 512
Master Output : 1/1
Master Output : 1/2
Master Output : 1/4
Master Output : 1/8
Master Output : 1/16
Master Output : 1/32
Master Output : 1/64
Master Output : 1/128
Master Output : 1/256
Master Output : 1/512
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$aaaa
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab8$
```

Code:

```
#include<mpi.h>
#include<omp.h>
#include<stdio.h>
#include<math.h>

int main()
{
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    for(int i=0; i<10; i++)
    {
        if(world_rank==0)
            printf("Master Output : 1/%d\n", (int)(pow(2, i)));
        else
            printf("Worker Output : %d\n", (int)(pow(2, i)));
    }

    MPI_Finalize();
    return 0;
}
```
