

# HW02 - Time Bomb

## CS5500

Chelsi Gupta

September 12, 2018

## 1 Steps for Implementation

- First I initialized the timer by using the rand function. The timer is a random integer between 1 to 10.
- Then the process 0 generates the time bomb and sends it to randomly selected process. Which is done through "rand()%size".
- The process that receives this timer, decrements it and then sends it to other processes by using MPI\_Send().
- The process which receives 1 as the timer and then decrements it to zero, declares itself as the loser. This process then sends the timer(which is 0) to all other processes.
- If a process receives a 0 timer value, it will simply stop its execution and display "bomb exploded".
- I compiled and ran the program using "mpic++ timebomb.cpp -o a.out" and "mpirun -np 4 ./a.out" commands respectively.

## 2 Code

```
#include <iostream>
#include <stdlib.h>
#include <mpi.h>
#define MCW MPLCOMM_WORLD

using namespace std;

int main(int argc, char **argv){
    int rank, size;
    int timer=rand()%10+1;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MCW, &rank);
    MPI_Comm_size(MCW, &size);
```

```

    if(rank==0){
        MPI_Send(&timer,1,MPI_INT,rand()%size,0,MW);
    }

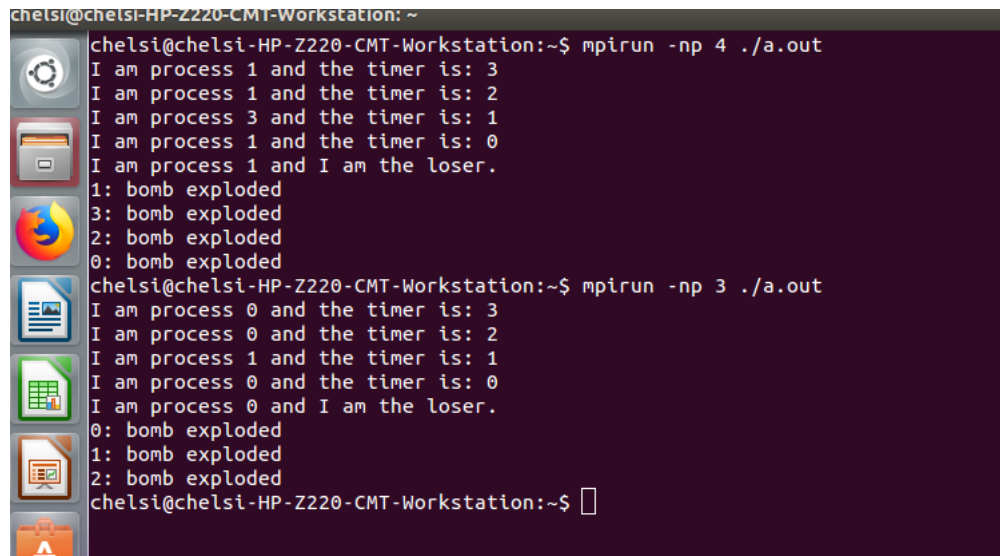
    while(1){
        MPI_Recv(&timer,1,MPI_INT,MPI_ANY_SOURCE,0,MW,MPI_STATUS_IGNORE);
        if(!timer){
            cout<<rank<<": bomb exploded"<<endl;
            break;
        }
        timer--;
        cout<<"I am process " <<rank<<" and the timer is: " <<timer<<endl;
        if(!timer){
            cout<<"I am process " <<rank<<" and I am the loser." <<endl;
            for(int i=0;i<size;++i){
                MPI_Send(&timer,1,MPI_INT,i,0,MW);
            }
        }
        MPI_Send(&timer,1,MPI_INT,rand()%size,0,MW);
    }

    MPI_Finalize();

    return 0;
}

```

### 3 Output



The image shows a terminal window on a system named 'chelsi@chelsi-HP-Z220-CMT-Workstation'. The user runs the command 'mpirun -np 4 ./a.out'. The output shows four processes (ranks 0, 1, 2, 3) each printing 'I am process X and the timer is: 3', then 'I am process X and the timer is: 2', then 'I am process X and the timer is: 1', then 'I am process X and the timer is: 0', and finally 'I am process X and I am the loser.'. After this, each rank prints 'X: bomb exploded'. Then the user runs 'mpirun -np 3 ./a.out'. The output shows three processes (ranks 0, 1, 2) each printing 'I am process X and the timer is: 3', then 'I am process X and the timer is: 2', then 'I am process X and the timer is: 1', then 'I am process X and the timer is: 0', and finally 'I am process X and I am the loser.'. After this, each rank prints 'X: bomb exploded'. The terminal ends with a prompt 'chelsi@chelsi-HP-Z220-CMT-Workstation:~\$'.

```

chelsi@chelsi-HP-Z220-CMT-Workstation:~$ mpirun -np 4 ./a.out
I am process 1 and the timer is: 3
I am process 1 and the timer is: 2
I am process 3 and the timer is: 1
I am process 1 and the timer is: 0
I am process 1 and I am the loser.
1: bomb exploded
3: bomb exploded
2: bomb exploded
0: bomb exploded
chelsi@chelsi-HP-Z220-CMT-Workstation:~$ mpirun -np 3 ./a.out
I am process 0 and the timer is: 3
I am process 0 and the timer is: 2
I am process 1 and the timer is: 1
I am process 0 and the timer is: 0
I am process 0 and I am the loser.
0: bomb exploded
1: bomb exploded
2: bomb exploded
chelsi@chelsi-HP-Z220-CMT-Workstation:~$

```

Figure 1: My Output