CS 5500 HW 2

Zac Johnson

January 2020

1 Implementation

This implementation is heavily influenced by the pachinko example we were shown in class. I changed the way `srand()` was seeded so that my numbers would be different every time the program ran. Rank 3 creates the bomb, which is just an `int` that holds a randomly generated number 1-20. The location of the bomb is shown just like pachinko except the timer is shown instead of the asterisk. Once the bomb is passed, the value is decremented and the next recipient is randomly selected. Once the timer reaches zero, the bomb explodes, the rank holding the bomb is declared The Loser, and a terminate message is sent to all the processes.

2 Compile and Run Commands

```
$ mpic++ timeBomb.cpp
$ mpirun -np 8 -oversubscribe a.out
```

Note: Commands are run from the same directory as timeBomb.cpp

3 Code

```cpp
#include <iostream>
//#include <mpi.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>

#include "/usr/local/include/mpi.h"
#define MCW MPI_COMM_WORLD

using namespace std;

int main(int argc, char **argv) {

    int rank, size;
    int timer;
    int dest;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MCW, &rank);  // unique number 0 to size-1
    MPI_Comm_size(MCW, &size);  // number of processes
    srand(time(nullptr));       // seed random so it's different every time

    if (rank==3) {
        timer = rand()%20;      // randomly set timer length
        cout << "The bomb has been planted with a " << timer << " second timer." << endl;

        for(int i=0; i < size; ++i)cout<<" "<<i<<" ";
        cout<<endl;

```

```
        MPI_Send(&timer,1,MPI_INT,3,0,MCW);
    }

    while(1) {
        // Wait to receive timer
        MPI_Recv(&timer,1,MPI_INT,MPI_ANY_SOURCE,0,MCW,MPI_STATUS_IGNORE);

        if(timer<0)break;
        sleep(1);
        // print location of the bomb with time value
        for (int i = 0; i < rank; ++i) cout << "   ";
        cout << " " << timer << " " <<endl;
        if (timer == 0) {
            cout << "Rank " << rank << " tried to defuse the bomb and did not succeed. F."
    << endl << endl;
        }

        // End and send the poison pill to everyone
        if(timer == 0){
            timer=-1;
            for(int i=0;i<size;++i) MPI_Send(&timer,1,MPI_INT,i,0,MCW);
        }

        timer--;                // decrement timer
        dest = rand() % size;   // randomly select next destination

        MPI_Send(&timer,1,MPI_INT,dest,0,MCW);
    }

    cout<<"rank "<<rank<<" left the game.\n";

    MPI_Finalize();

    return 0;
}
```

4 Desired Output

```
The bomb has been planted with a 13 second timer.
 0  1  2  3  4  5  6  7
          13
                      12
    11
    10
                      9
                      8
                   7
    6
                   5
                      4
                3
    2
                1
                      0
Rank 7 tried to defuse the bomb and did not succeed. F.

rank 0 left the game.
rank 1 left the game.
rank 2 left the game.
rank 3 left the game.
rank 7 left the game.
rank 4 left the game.
rank 5 left the game.
rank 6 left the game.
```