

Assignment 2 report

Cory Christensen

January 20, 2020

1 Implementation

The code is created by creating an initial if (rank == 0) statement that enters into the game. The bomb starts with process zero with a randomly generated timer value between zero and fifty then a random number between zero and size is created to send the bomb to. This then enters a while true loop where each process is just waiting to receive a message. If that process receives a message it will receive a countdown number.

If the countdown is greater than zero it will print a statement about receiving the bomb with however many seconds left, subtract one from the countdown, and send the countdown to another random process that is not itself.

If the countdown is equal to zero a statement shows that the process that received the zero lost. The countdown is then set to negative one and sent to all processes.

If the countdown is less than zero the process will break out of the loop. Thus after the countdown is zero all processes will break out of their respective loops and the program will end.

1.1 A Run Commands

1. Navigate to directory where file is stored
2. Run command: `mpic++ Assn2.cpp`
3. Run command: `mpirun -np 8 -oversubscribe a.out`

2 Code

```
#include <iostream>
#include <mpi.h>
#include <unistd.h>
#include <stdlib.h>

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

using namespace std;

int main(int argc, char **argv){
    int rank, size;
    int countdown = rand()%50;
    int initCount = countdown;
    int process;//= rand()%size;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MCW, &rank);
    MPI_Comm_size(MCW, &size);
    srand(countdown*rank);
```

```

bool start = true;

if(rank == 0 && start) {
    cout << "Process " << rank << " started the bomb with " << countdown << " time left\n";
    start = false;
    countdown--;
    sleep(1);
    process = rand() % size;
    while (process == rank) {
        process = rand() % size;
    }
    MPI_Send(&countdown, 1, MPI_INT, process, 0, MCW);
}
while (1) {
    MPI_Recv(&countdown, 1, MPI_INT, MPI_ANY_SOURCE, 0, MCW, MPI_STATUS_IGNORE);
    srand(countdown*rank);

    if(countdown > 0){
        cout << "Process " << rank << " recieved the bomb with " << countdown << " time left\n";
        countdown--;
        sleep(1);
        process = rand() % size;
        while (process == rank) {
            process = rand() % size;
        }
        //cout << "Process " << rank << " is sending the bomb to process " << process << "\n";
        MPI_Send(&countdown, 1, MPI_INT, process, 0, MCW);
    } else if(countdown == 0){
        cout << "Process " << rank << " recieved the bomb with " << countdown << " time left\n";
        cout << "The bomb has exploded. Process " << rank << " has lost" << endl;
        countdown = -1;
        //MPI_Bcast(&countdown, 1, MPI_INT, 0, MCW);
        for(int i = 0; i < size; i++){
            MPI_Send(&countdown, 1, MPI_INT, i, 0, MCW);
        }
        break;
    } else if(countdown < 0) {
        break;
    } else{
        process = rand()%size;
        MPI_Send(&countdown, 1, MPI_INT, process, 0, MCW);
    }
}

MPI_Finalize();

return 0;
}

```

3 Output

```
Process 0 started the bomb with 33 time left
Process 7 recieved the bomb with 32 time left
Process 3 recieved the bomb with 31 time left
Process 6 recieved the bomb with 30 time left
Process 1 recieved the bomb with 29 time left
[Cory-C:01419] 7 more processes have sent help message help-btl-vader.txt / cma-permission-denied
[Cory-C:01419] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
Process 7 recieved the bomb with 28 time left
Process 6 recieved the bomb with 27 time left
Process 7 recieved the bomb with 26 time left
Process 1 recieved the bomb with 25 time left
Process 5 recieved the bomb with 24 time left
Process 1 recieved the bomb with 23 time left
Process 6 recieved the bomb with 22 time left
Process 4 recieved the bomb with 21 time left
Process 5 recieved the bomb with 20 time left
Process 0 recieved the bomb with 19 time left
Process 7 recieved the bomb with 18 time left
Process 1 recieved the bomb with 17 time left
Process 5 recieved the bomb with 16 time left
Process 3 recieved the bomb with 15 time left
Process 2 recieved the bomb with 14 time left
Process 3 recieved the bomb with 13 time left
Process 4 recieved the bomb with 12 time left
Process 7 recieved the bomb with 11 time left
Process 2 recieved the bomb with 10 time left
Process 7 recieved the bomb with 9 time left
Process 3 recieved the bomb with 8 time left
Process 1 recieved the bomb with 7 time left
Process 5 recieved the bomb with 6 time left
Process 3 recieved the bomb with 5 time left
Process 5 recieved the bomb with 4 time left
Process 7 recieved the bomb with 3 time left
Process 0 recieved the bomb with 2 time left
Process 7 recieved the bomb with 1 time left
Process 5 recieved the bomb with 0 time left
The bomb has exploded. Process 5 has lost
```