Ex07_p01

Below is the program I wrote for this exercise:

p01.cpp

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

int main(int argc, char** argv)
{
    MPI_Init(&argc, &argv);

    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    if (world_size < 2) {
        std::cerr << "World size must be greater than 1 for
" << argv[0] << std::endl;
        MPI_Abort(MPI_COMM_WORLD, 1);
    }
    int msg[2];
    if (world_rank == 0) {
        for (int j = 0; j < 100; j++) {
            MPI_Recv(&msg, 2, MPI_INT, MPI_ANY_SOURCE,
MPI_ANY_TAG, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
            std::cout << "Recieved message " << msg[1] << "
from process " << msg[0] << std::endl;
        }
    } else if (world_rank >= 1 && world_rank <= world_size)
{
        msg[0] = world_rank;
        for (int i = 0; i < 100; i++) {
            msg[1] = i;
            MPI_Send(&msg, 2, MPI_INT, 0, 0,
MPI_COMM_WORLD);
        }
    }

    MPI_Finalize();
}
```

After running the program multiple times and examining the output I came to the conclusion that the message passing implementation is mostly fair. The order in which messages are received is different every time and seems to be, for the most part, random.