Ex6

s1260119 Yuta Tomiyama

Task 6.1

Bcast is more efficient. It can send values from root to all processes easily.

Task 6.2

1. Faster by not requiring communication between processors.
2. Deadlock may happen.
3. TCP

Task 6.3

```
import mpi.Intracomm;
import mpi.MPI;

import java.io.*;
import java.util.Arrays;
import java.util.stream.Collectors;

public class w06_wordCount {
    public static void main(String[] args) throws Exception {
        MPI.Init(args);
        Intracomm comm = MPI.COMM_WORLD;
        int[] sum = new int[1];
        int[] max = new int[1];
        int rank = comm.Rank();
        File file = new File("file_" + rank + ".txt");
        String[] content = new BufferedReader(new InputStreamReader(new
FileInputStream(file)))
                .lines().collect(Collectors.joining(" ")).split(" ");
        int wordCount = content.length;
        int longestWordLength =
Arrays.stream(content).mapToInt(String::length).max().orElse(0);
        comm.Reduce(new int[] { wordCount }, 0, sum, 0, 1, MPI.INT,
MPI.SUM, 0);
        comm.Reduce(new int[] { longestWordLength }, 0, max, 0, 1,
MPI.INT, MPI.MAX, 0);
        if (rank == 0) {
            System.out.println(sum[0] + " " + max[0]);
        }
        MPI.Finalize();
    }
}
```

Task 6.4

```
import mpi.Intracomm;
import mpi.MPI;

import java.util.Random;

public class w06_RSP {
    public static void main(String[] args) {
        MPI.Init(args);
        Intracomm comm = MPI.COMM_WORLD;
        Random rand = new Random();
        int p1Wins = 0;
        int p2Wins = 0;
        while (true) {
            int[] hand = { rand.nextInt(3) };
            int[] hands = new int[2];
            comm.Gather(hand, 0, 1, MPI.INT, hands, 0, 1, MPI.INT, 0);
            if (comm.Rank() == 0) {
                int p1 = hands[0];
                int p2 = hands[1];
                int result = (p1 - p2 + 3) % 3;
                System.out.println("P1: " + getHand(p1) + ", P2: " +
getHand(p2));
                if (result == 0) {
                    System.out.println("Draw!");
                } else if (result == 1) {
                    p2Wins++;
                    System.out.println("P2 wins!");
                } else {
                    p1Wins++;
                    System.out.println("P1 wins!");
                }
                comm.Send(new boolean[] { p1Wins == 3 || p2Wins == 3 },
0, 1, MPI.BOOLEAN, 1, 0);
                if (p1Wins == 3) {
                    System.out.println("Winner: P1");
                    break;
                } else if (p2Wins == 3) {
                    System.out.println("Winner: P2");
                    break;
                }
            } else {
                boolean[] result = new boolean[1];
                comm.Recv(result, 0, 1, MPI.BOOLEAN, MPI.ANY_SOURCE,
MPI.ANY_TAG);
                if (result[0]) {
```

```
                    break;
                }
            }
        }
        MPI.Finalize();
    }

    private static String getHand(int hand) {
        String str;
        switch (hand) {
            case 0:
                str = "rock";
                break;
            case 1:
                str = "scissors";
                break;
            case 2:
                str = "paper";
                break;
            default:
                str = "";
                break;
        }
        return str;
    }
}
```