Parallel Processing – task 4

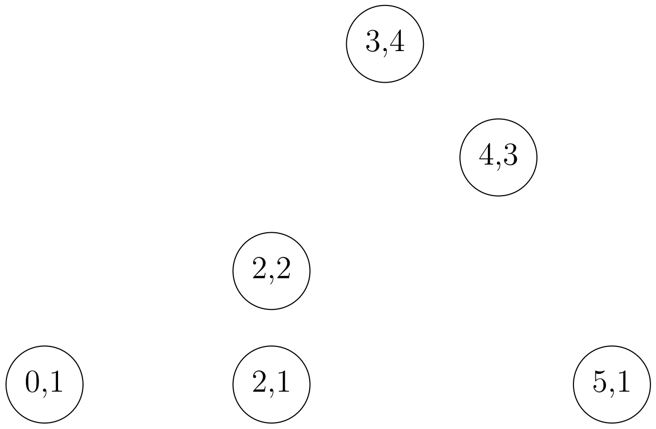
03.12.2015

Design and implement the synchronization protocol for the following problem. Use the MPI library. Allow only local communication (each process can communicate only with the neighboring processes). No global synchronization is allowed.

Problem.

1. A group of street musicians occupy a city square. Their positions are given in the file `positions`. The first row contains the number of musicians, the next rows describe their coordinates (in meters). An exemplary file and the corresponding arrangement of musicians are given below.

```
6
0  1
2  1
3  4
2  2
5  1
4  3
```



2. Each musician wants to give a concert. The concert lasts for 2 seconds. After giving a concert, the musician becomes inactive.

3. Each musician can be heard in the radius of 3 meters. As musicians are gentle people, they will not play if they can hear someone playing.

4. Observe that not all musicians can play at the same time. Thus they play in rounds. Then the musicians finish playing, the musicians who will play in the next round are selected.

5. The only thing that may stop a musician from playing is hearing another musician (or having finished before).

6. When the musicians are discussing who will play, they can communicate only with the people within the distance of 3m.

Wymagania techniczne.

1. Your solution should work on Omega.

2. There is one process for each musician.

3. Provide some comments explaining your algorithm.

4. Use MPI for communication.

5. Print the current state of processes to the console.

6. Submit the source files and the makefile. Running `make compile` should compile your application. running `make run file=[file]` should run your program for the positions described in `file`. Assume that the file is in the proper format and the number of processes given in the hosts file is correct.

Submission deadline: 15.12.2015 (23:59)