System and Device Programming

Examination Test – Programming Part
10 September 2013

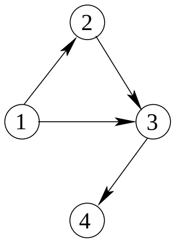
Examination Time: 1h 45min. Evaluation. 18 marks.
Textbooks and/or course material allowed.
The final mark is the sum of the 1st and the 2nd parts; it cannot be refused (no retry for marks ≥ 18*).*

A binary file stores the adjacent list (i.e., a list of list) of a directed graph. The file includes:

- A value indicating the number of vertices n of the graph.
- For each of the n vertex
  - A value as vertex identifier
  - A value reporting the length of the adjacent list
  - A list ov values reporting the adjacent vertices.

Notice that each value in the file is a 32 bit integer. The following figure reports a correct example of the file (left-end side) and the corresponding graph (center).

```
4
1 2 2 3
2 1 3
3 1 4
4 0
```



```
4
1 3 2 3 4
2 2 3 4
3 1 4
4 0
```

A program is run with three parameters on the command line
`inputFileName n outputFileName`
where `n` is an integer value, and the other two parameters (`inputFileName` and `outputFileName`) are strings (of maximum 30 characters) indicating file names.

The program has to run `n` threads. Each thread has to visit the graph starting from one of its vertices, and has to generate the list of reachable vertices (from the starting one). Nevertheless, the file must not be stored into main memory completely (so it is not possibile to yuse a list of lists, for instance). As a concequence all threads have to access the `inputFileName` file in a direct way to visit the corresponding graph.
The list of vertices reachable from each vertex has to be stored in the output file with the same format (see the right-end side of the previous figure).

Notice that each thread corresponds to a recursive function visiting the graph in depth-first way. Moreover, n is usually much smaller than the number of vertices of the graph. Finally, obviously, the output file has to be accessed in mutual exclusion by the threads generating the visited lists of vertices.