System and Device Programming

Examination Test – Programming Part
06 February 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).

Write a multi-threaded application in the Windows environment with the following specifications.

The application receives three parameters on the command line:

- An integer value **n**.
- A string named **source**, representing a directory tree name which includes only text files.
- A string named **destination**, representing an empty or a non-existing directory tree.

The application performs the following operations:

- It makes a copy of each single file included in the **source** directory tree in a corresponding sub-tree (with the same name) of the **destination** directory tree. The **destination** directory and sub-directory have to be created if they do not exist.
- Each file copied from the **source** to the **destination** directory has to be modified as follows:
  - Each line of the file includes at the beginning of the line itself the number of words on the entire line. The lines of each file are ordered in ascending order of the words count. The number of lines, words, and characters of the file are added as a first line of the file (three integer values separated by one single space).
- It creates in the current (working) directory a single file containing the content of all destination files (but the first line added by the application itself) merged together, i.e., it creates a single file whose content is equal to the same set of lines of the original file ordered by increasing number of words with the number of words at the beginning of each line.

The application has to used **n+1** threads to perform its duty. The first **n** threads are in charge of modifying and copying all single files from the **source** to the **destination** directory. The last thread merges the content off all files in a single file and it writes that file on the current (working) directory.