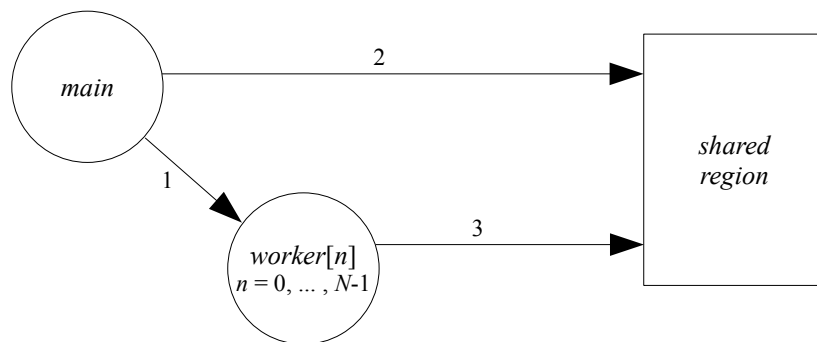


Assignment 1

Take the general problems, which have been discussed in the lab classes and for which you have developed a single-threaded solution. The aim now is to convert them into a multithreaded application running under Linux.

The decomposition in both cases can be described by the following diagram.



OPERATIONS

1. create *worker* threads, wait for them to terminate
2. present the data file names for processing, print results
3. get a piece of data and process it, save partial results.

So, the role of thread *main* will be getting the data file names by processing the command line and storing them in the shared region, creating the *worker* threads and waiting for their termination, and printing the results of the processing. On the other hand, the role of threads *worker* will be carrying out the processing itself: they request in succession pieces of data to process, process it and deliver the results of their processing. They terminate when there are no more data pieces to process.

GRADING

- development and validation of a multithreaded version of one of the general problems according to specification – 13 points
- development and validation of a multithreaded version of both general problems according to specification – 20 points.

DELIVARABLES

- an archive, named `CLE1_T$G#.zip` (where \$, equal to 1, ... , 3, means the lab number, and #, equal to 1, ... , 10, means the group number) containing both the source files your solution to the two problems and a pdf file, named `present.pdf`, up to 6 pages (power point like), where the main ideas of the design of the solutions to the two problems and the timing results that were obtained on runs with 1, 2, 4, 8 *worker* threads, are discussed
- the archive should result from the compression of a directory, named `CLE1_T$G#`, containing two subdirectories, named `prog1` and `prog2`, and the file `present.pdf`.

DEADLINE

- April, 24, at midnight.