

DAS

Implementation

The task is fully completed. The approach taken during the completion process was to use multithreading to process each UDP Datagram separately to improve performance. Because of this concurrency, it may happen that protocol commands that change state of the Master, such as sending -1 which must terminate the process, may produce double output in case if 2 or more datagrams with -1 are received and first corresponding thread isn't fast enough to stop execution of the other ones. This may produce more lines with "-1", exceptions are, however, handled.

Implementation Details

There are 3 main classes: DAS, Master, Slave. DAS - used in execution, tries to open socket and decides in which mode application works. Then delegates the job to either Master or Slave.

Master - runs in a loop receiving datagram packets. When packets are received - executes handling function in a separate thread, managed by the ExecutorService. ExecutorService here used to have more control over executed threads, to be able to shutdown them in the future, when we need to terminate the process. Values received by the master are stored in ConcurrentLinkedQueue, which makes cross-thread read/write access safe.

Slave - simple one-function program to send specified value on the specified port in local host.

Protocol

Protocol is defined by the task description:

- Both slave and master are executed on localhost.
- Slave communicate with master by sending Datagrams with numbers, sending things that are not integer numbers is prohibited by both Master and Slave.
- Master doesn't communicate with Slaves directly, but it sends datagrams on the local network broadcast address.
- On receiving 0 - Master calculates average and broadcasts it. On receiving -1 - Master broadcasts -1 and terminates. Any other value is stored in memory for future computations of the average.

How to run

To run the application it is sufficient to download java and compile the program in the project directory:

```
javac src/*
```

Then go into out/production/DAS directory and use it with:

```
java DAS <port> <number>
```