Back pressure

Distributed Systems Paradigms
Lab Guide 4

Use end-to-end back pressure in the the reactive streams framework.

Steps

1. Test back pressure stratgies. Suggestion: Use `Flowable.interval()` to generate data and `Thread.sleep()` in the subscriber.
2. Improve the reactive main loop to accept a flowable to be written to a connected socket.
3. Implement a simple client/server application using the reactive main loop, for instance, echoing back the string in uppercase. (Initially, use some back pressure strategy to convert incoming observable to a flowable.)
4. Implement a back pressure-aware custom reactive operator to split lines in byte buffers. Test with:

```
Flowable.just("xyz\n", "abc\n123", "456\n", "def")
    .map(s -> StandardCharsets.UTF_8.encode(s))
    .lift(new LineSplitOperator())
    .map(bb -> StandardCharsets.UTF_8.decode(bb))
    .subscribe(s -> System.out.println(s));
```

5. Improve the reactive main loop to consider back pressure when reading from connected sockets (i.e., make `read()` return a `Flowable`.)

Questions

1. Can you describe the sequence of invocations in reactive interface methods in the pipelines used for answering Steps 3 and 5?
2. What if business logic itself would block (e.g., waiting for a database operation)?

Learning Outcomes Understand back pressure in reactive streams. Construct efficient and resilient servers using reactive streams.