CS425 MP1

In this project, unicast, causal order multicast and total order multicast process.

Basic schema

- Each process is a java process
- Each process has one thread to handle user Input, one thread to send message, and one thread for receiving message from its peers.
- There is a master node to handle total order multicast.
- Each process maintain an internal vector clock

Requirement

- MacOs or Linux (gnome desktop preferred)
- Java 9 installed
- Gradle installed

Project structure

```
.
 BlockingProcess.java              # Unicast process
CausalMulticastDemo.java         # Causal order multicast startup
CausalOrderProcess.java          # Causal order multicast processs
 Config.java                      # Configuration file parser
DelayParser.java                # Delay parser for causal order "dealy command"
 DeliverThread.java               # The thread for deliver
Master.java                     # Master process for total order multicast
 MasterUp.java                    # Master process startup
Message.java                    # Message object for total order multicast
Packet.java                     # Packet object for causal order multicast
TotalOrderDemo.java              # "Ordinary" total order multicast startup
TotalOrderProcess.java           # "Ordinary" total order multicast process
 UnicastDemo.java                 # Unicast startup
 VectorClock.java                 # Vecterclock object
```

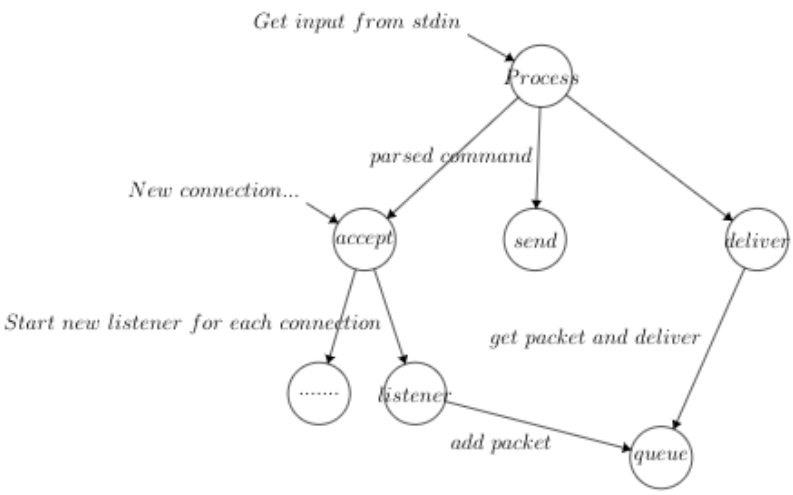

Figure 1:

Overview of a process

Build

```
gradle jar #(at root dir)
```

Run

Run CausalMulticastDemo by .jar after build

```
java -cp build/libs/CS425MP1.jar Process.CausalMulticastDemo <id> CausalConfiguration [scrip
```

Run UnicastDemo by .jar after build

```
java -cp build/libs/CS425MP1.jar Process.UnicastDemo <id> UnicastConfiguration
```

```
./UnicastRun.sh 4 #(the number of process, has to be consistent to configuration file)
```

```
./CausalMulticastRun.sh 4 (the number of process, has to be consistent to configuration
file) [script directory name] #(read startup command from ./script directory)
```

Available command

Unicast

```
send <id: int> <message>
```

Causal multicast

```
sleep (sleep for 1000ms)

clock (check current vector clock)

msend <message> [delay id=delay,id=delay...]
(the last delay will be used to other unspecified processes, current thread is always 0 (I think
If not specified delay explicitly, random delay will be used)
```

Exit

Just press Ctrl+C

Script format

Same as available command (script is only available for causal multicast)