CS425 MP1

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

Group: *Chuchao Luo(chuchao2), Wenhan Zhao(wenhanz3)*

(check README.md for better format)

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

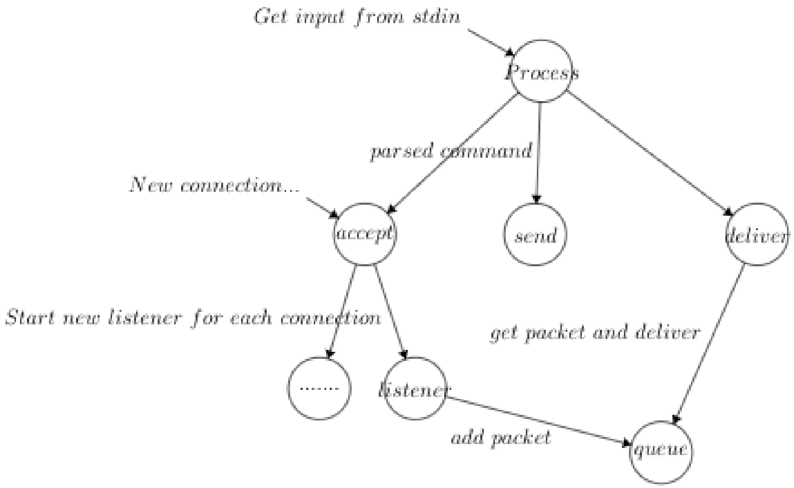
Requirement

- 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
```

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 TotalOrderDemo by .jar after build

```
java -cp build/libs/CS425MP1.jar Process.TotalOrderDemo <id> TotalConfiguration [script]
```

- Run UnicastDemo by .jar after build

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

- Startup 4 Unicast processes

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

- Startup 4 Total order multicast processes

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

- Startup 4 Causal order multicast processes

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

Available command

Unicast

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

Total multicast

```
msend <message> [delay]
(The delay parameter is optional, if you want to customize the delay you can add
the delay, otherwise there will be a randomized delay within the range specified
in the configuration file)
```

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 t
If not specified delay explicitly, random delay will be used)
```

Exit

Just press Ctrl+C

Script format

Same as available command