Eaton homework assignment

Ryan Dutton

3 March 2021

General instructions

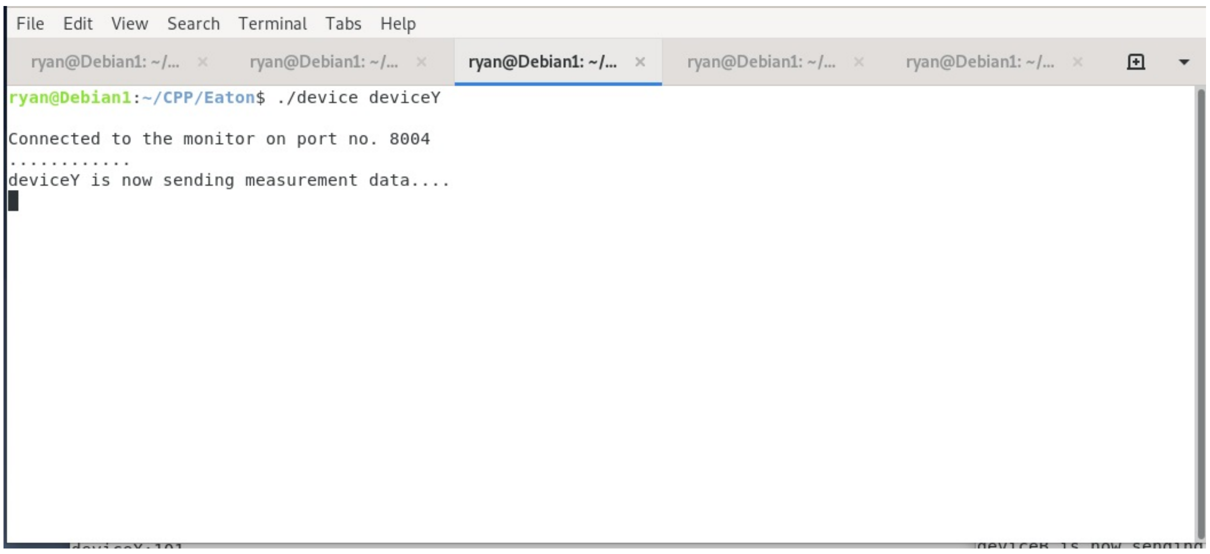
1. Go to the following link, https://github.com/pragyana/E-A-T-O-N-project, and download the following source code into one directory - server.cpp, client.cpp, device.cpp, ThreadWrapper.h, Makefile

2. Build the source code by running 'make' in the directory where files have been downloaded.

3. Open one terminal for the server, one terminal for the client and several terminals for running the device programs.

4. Start the server by typing './server'. Start the client on a separate terminal by typing './client'. Start different devices on different terminals by typing './device <device name>'. For example './device deviceA' will start a device that is uniquely identified as deviceA.

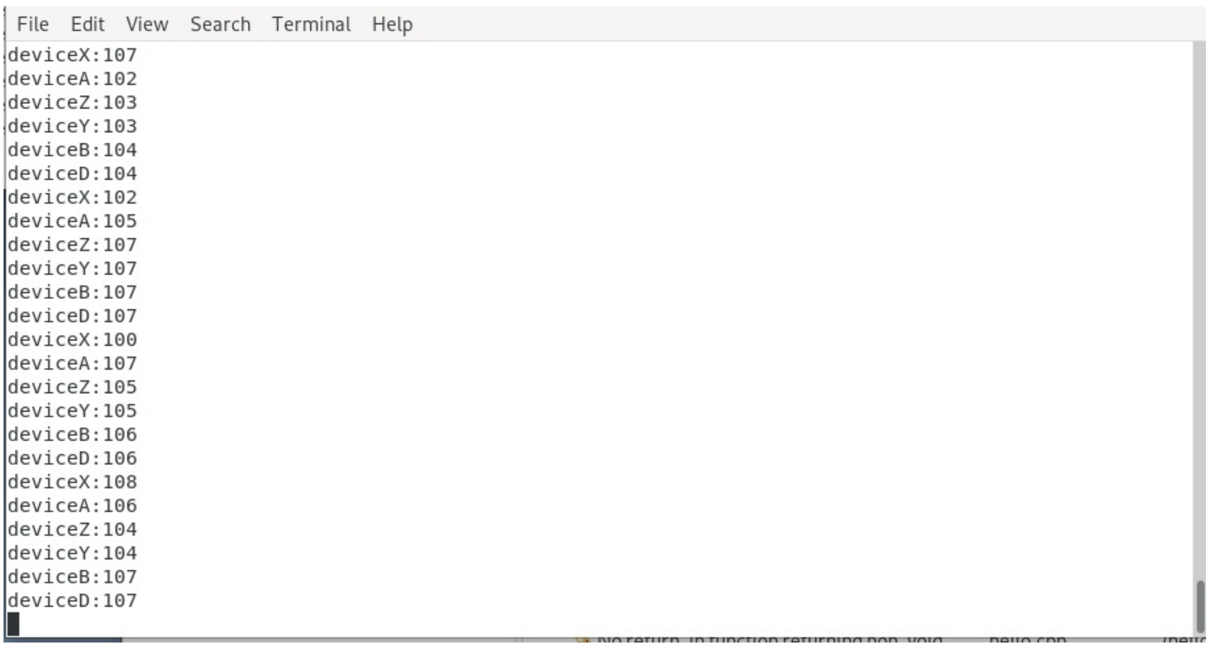
5. A device sends to the server measurements every 5 seconds. In our implementation these measurements are random numbers in the range of 100 to 109.



```
ryan@Debian1:~/CPP/Eaton$ ./device deviceY

Connected to the monitor on port no. 8004
............
deviceY is now sending measurement data....
```

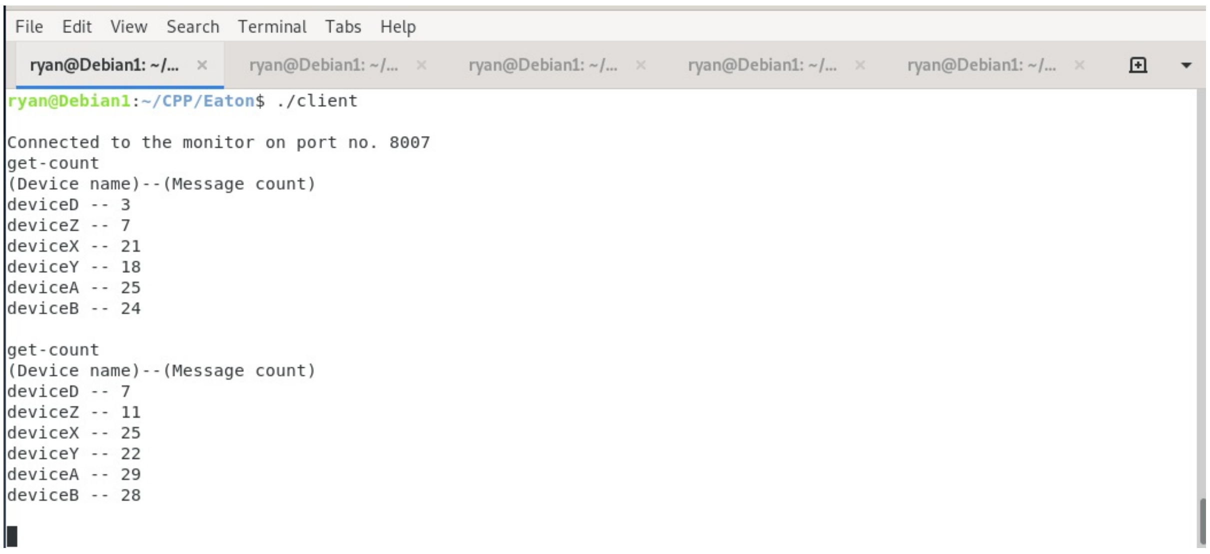
Fig. 1 – A device named 'deviceY' has been started and is sending measurements to the server



```
deviceX:107
deviceA:102
deviceZ:103
deviceY:103
deviceB:104
deviceD:104
deviceX:102
deviceA:105
deviceZ:107
deviceY:107
deviceB:107
deviceD:107
deviceX:100
deviceA:107
deviceZ:105
deviceY:105
deviceB:106
deviceD:106
deviceX:108
deviceA:106
deviceZ:104
deviceY:104
deviceB:107
deviceD:107
```

Fig. 2 – The server is displaying measurements coming from different devices.

6. Once all the devices have been started, the client can request a count of messages from different devices by typing ‘get-count’.



```
ryan@Debian1:~/CPP/Eaton$ ./client

Connected to the monitor on port no. 8007
get-count
(Device name)--(Message count)
deviceD -- 3
deviceZ -- 7
deviceX -- 21
deviceY -- 18
deviceA -- 25
deviceB -- 24

get-count
(Device name)--(Message count)
deviceD -- 7
deviceZ -- 11
deviceX -- 25
deviceY -- 22
deviceA -- 29
deviceB -- 28
```

Fig. 3 – The client displaying the message count for each device.

7. The server must be started before the devices and the client. The devices and the client must be shut down first before shutting down the server. They all can be shut down by pressing ctrl-C on the respective terminals.

Salient features

1. Port numbers 8000 to 10000 are reserved for the application. The main server is started on 8000. The devices and the client are each dynamically assigned a separate port. Thus, a maximum of 2000 device connections are possible without restarting the server.

2. By assigning a separate port to each device we are achieving better concurrency than having all the devices connecting on the same port.

3. Every time a device or the client connects to the server, the server starts a new thread that handles the individual connection. The new thread informs the device and the client of the new port number, and listens on that port number.

4. The devices and the client initially connect to the server on 8000. However, after receiving the new port number they connect on the new port number.

5. The main thread maintains a map that uses the unique device name as the key and the message count as the value. Every time a message arrives from a device, the count is incremented.

6. If a device shuts down, the server still maintains the message count. If the same device reconnects at a later time, the server will start incrementing the count that has already been preserved in the map.