**DC3580 PP2 Abdullah Alajaj**

**4/1/2018**

**Counter Servers**

In this program we are testing and modifying the Java codes from chapter 5 on figure 5.31, 5.32 and 5.33 in Liu’s textbook.

We describe three tasks individually:-

Task one, a client makes single connectionless (UDP) call to the server where the client provide the host name and port number using the presentation tier (CounterClient). The data move to the logic tier (CounterClientHelper) where the package get finalize using dynamic datagram socket and send it using the data tier (MyDatagramSocket). Also, it waits for the respond package and analyst back to the presentation tier.

On the other side, the counter server will be running on a loop waiting for packages to arrive. When a package arrives at the data tier (MyServerDatagramSocket), each message will be count as one and send back the total count to the client using same strategy of the three tier layers.

Task two, does the same thing as task one except the server count increment count by 2 for every package arrive.

Task three, does use streaming socket, since the program connection-oriented (TCP/ IP) where a client make request of connection before transmitting data. When the server accepts connection then the client will start sending and receiving packages. Logic layer not necessary on the server since we are only counting packages.

**Flow diagram:**

For **Task 1 and 2:**

Server Client

Create a Socket

Create a Socket

Bind the socket

Bind the socket

Data (request)

Send the request

Receive the socket

Data (reply)

Receive response

Send response

For **Task 3:**

Server Client

Create a Socket

Create a Socket

Request connection

Bind the socket

Establish Connection

Send the request

Accept connection

Data (request)

Receive response

Receive the socket

Send response

Data (reply)

**Functions Description:**

**Task 1:**

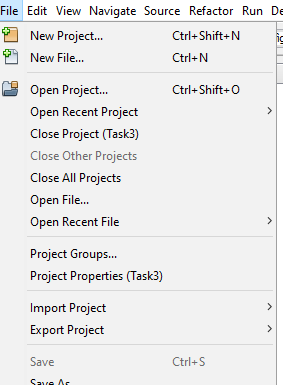
**Task2:**

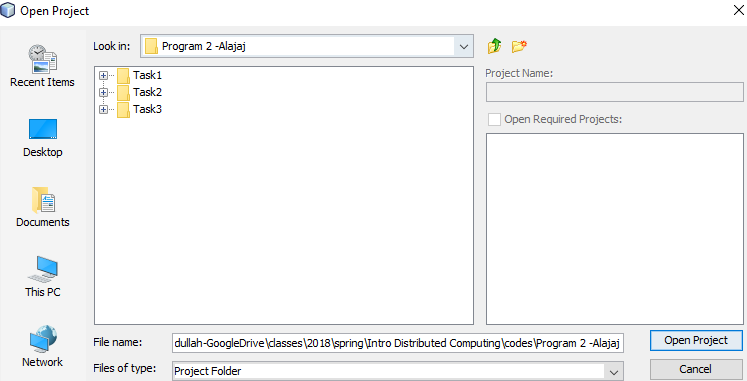
Same as Task 1 except the counter increases by 2 with each client

**Task 3:**

**Compiled and executed:**

* Unzip “program 2 – Alajaj” folder then open NetBean IDE (prefer 8.2 version)
  + From top Manu bar : file >> (file location)>>(choose a task)





**To compiled the code:**

* I test it through local network and on the same PC using windows PowerShell and command prompt in windows 10 Home Version 1709
  + The easy way to open with the correct path is shift + right click in src folder for the task you want to run << open PowerShell window here
  + For command prompt, Go to the desktop search bar and type “cmd”. To set correct path use cd = “ “ where inside “” the location of the src file.

Note:

* if the (classname).class does not exist then you can create one using this code Javac (the class name you want to compile).java
* You must run the server before the client.
* Make the client run on PowerShell and server run on command prompt or the other way around. Do not run both code on the same promot windows.

**Code test result:**

Let

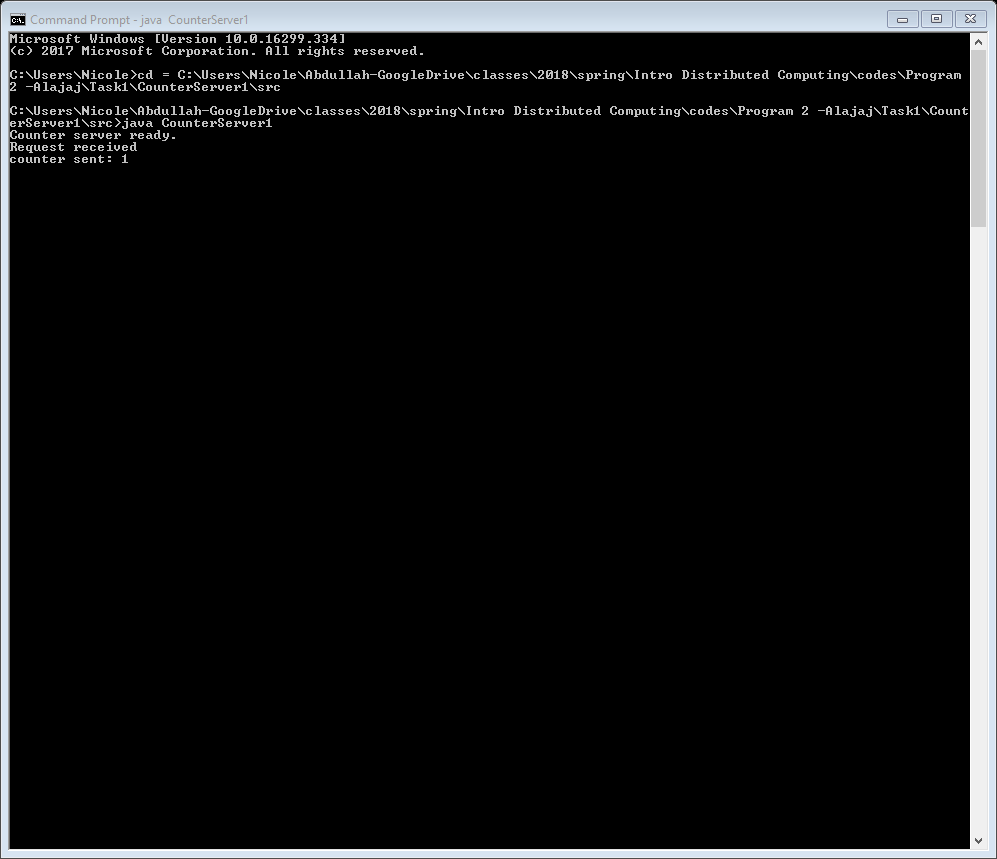
Client to be A

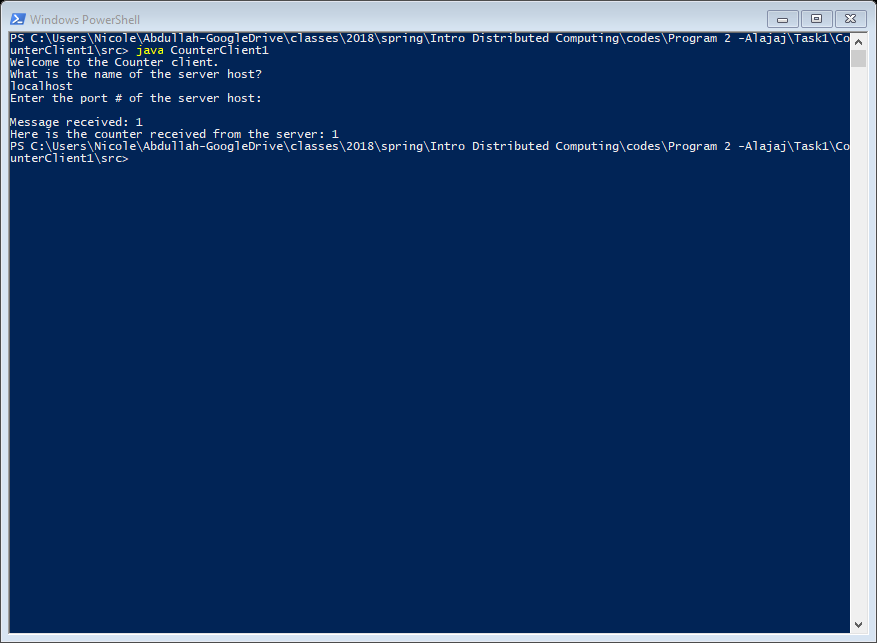
Server to be B

B First received package, A then package, the order in which they run is important, because in this routine, Using a connectionless transfer, sending data. One side cannot guarantee that the data must be received by the server, only the first run Receiver makes it into the blocked state, and then shipped row client In order to allow the smooth transfer of information. If the server has not yet opened the case to send data, then the number it will be lost the recipient will therefore remain blocked. Therefore, the server's service should be the first to open, In obstruction state, waiting for the client to send package.

**Task1:**

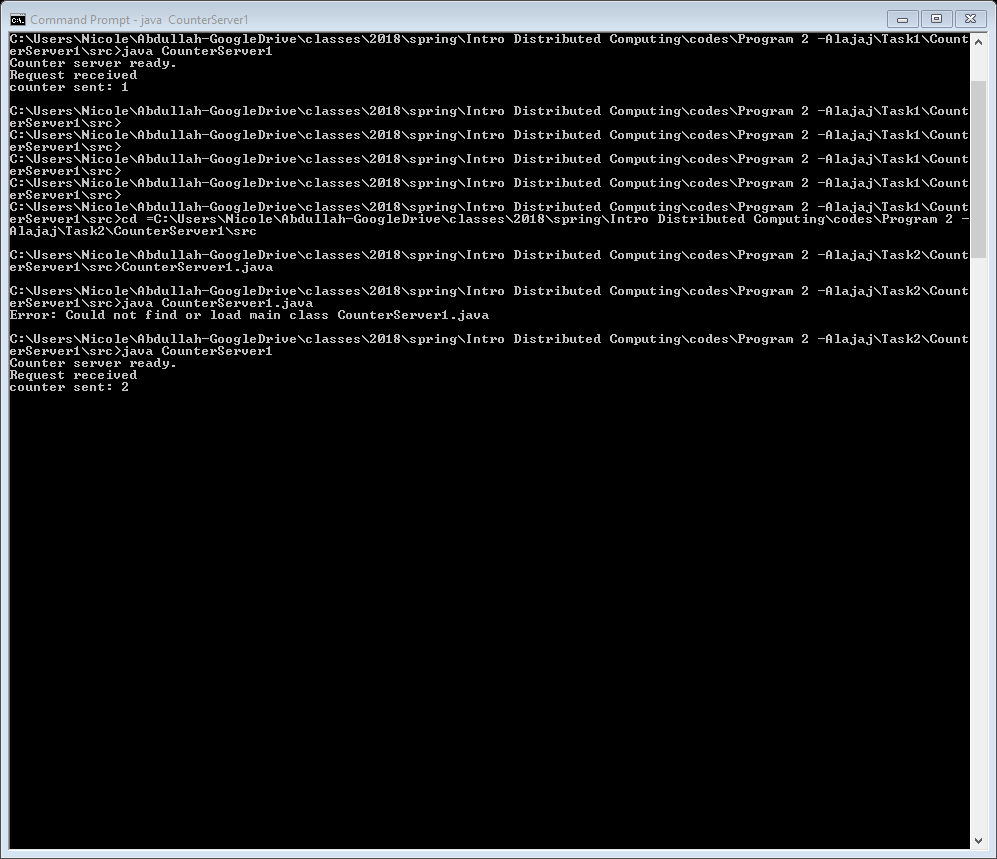
* Running B first and A second :- package arrived to B and package arrived to A with same number of total count.
* Running A first and B second: - none of the packages arrived and both API waiting for each other. When I stop B control c then run it again one package arrived at each console where the package from B got lost and did not deliver. When I stop A then it does not matter how many time I try to connect to B it does not send or receive anything.

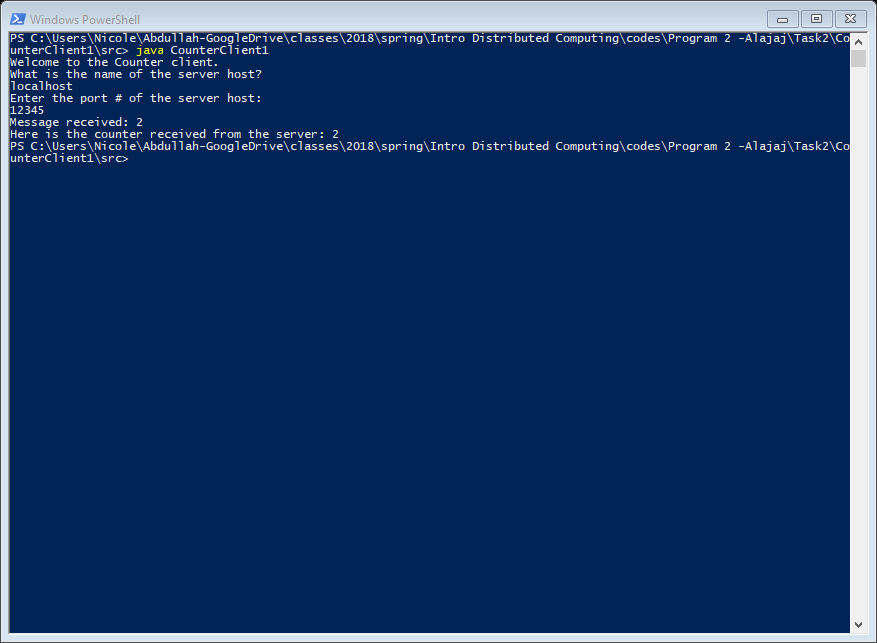




**Task2:**

* Running B first and A second: - package arrived to B and package arrived to A with same number of total count which is two for every package.
* Running A first and B second: - none of the packages arrived and both API waiting for each other. When I stop B control c then run it again one package arrived at each console where the package from B got lost and did not deliver. When I stop A then it does not matter how many time I try to connect to B it does not send or receive anything.

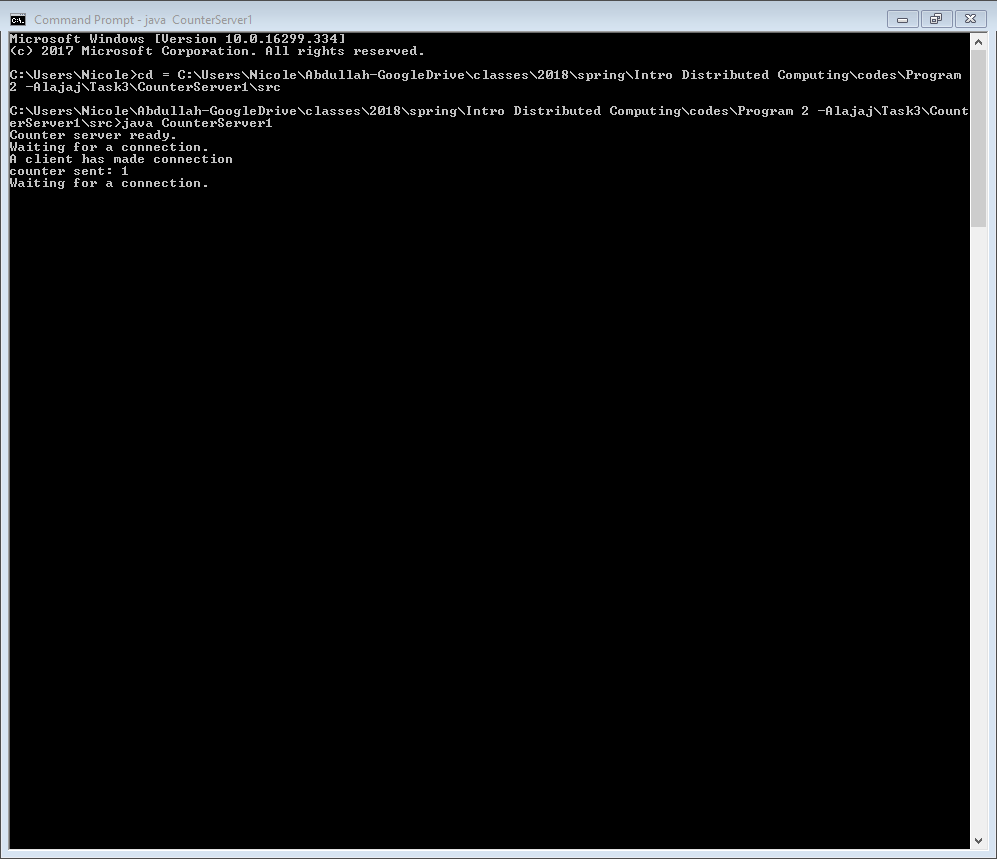


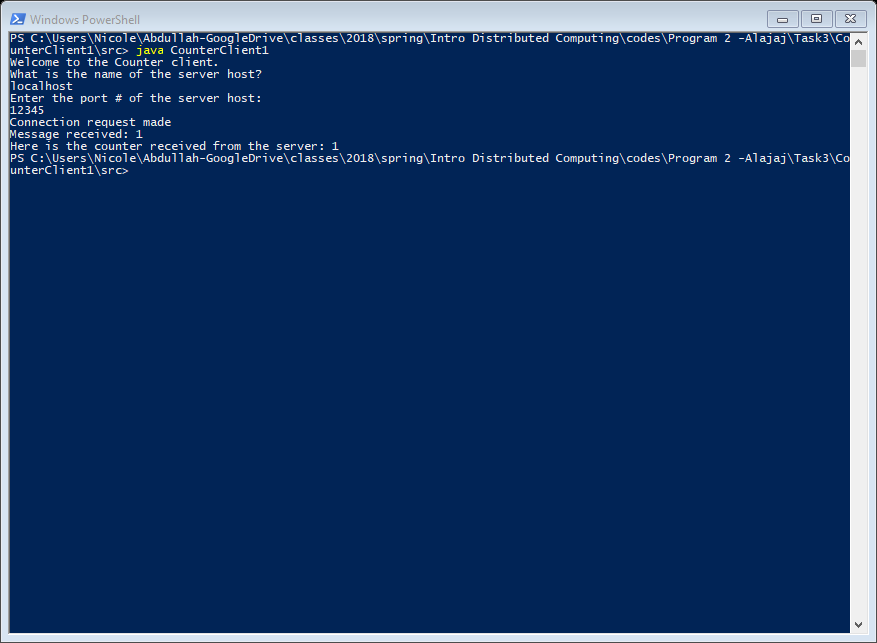


**Task3:**

First Executive B, its meeting Enter Into Resistance Plug shape state, It Rear Shipped Row A After sending the connection request to start streaming a file using streaming datagram socket, after conforming connection (accept). The program design to send one package and stop connection which makes it look like it is UDP, no it is not. We use TCP / IP (or localhost) which grantee package arrival on a specific order.

* Running B first and A second:- package arrived once to B and respond package arrived to A .
* Running A first and B second: - none of the packages arrived and A API will throw exception (connection refused) since B API run second.





Error when trying to connect to not open server

