Network Programming Lab 6

Aims

1. To familiar with concurrency.
2. Practice on sockets, server sockets and threads.

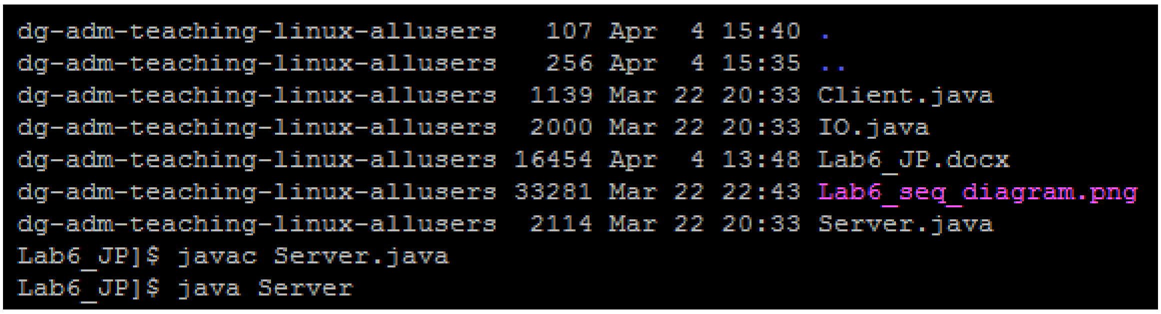
Tasks

Use Sockets, ServerSockets and Threads in the following tasks when possible.

1. Write a simple FTP clone using the Client/Server architecture:
   - Using the Java classes `Socket`, `ServerSocket` and `Thread`, write a client program that reads the 100 megabyte uncompressed file (*see canvas to download the compressed file*) and sends that file to the server. The server will then write that file to disk. The reading of the file within the client must be done asynchronous and is thread safe. Evidence of concurrency must be present i.e. that at least two threads are in a running state (*not sleeping or waiting)* at least some of the same time.
   - Create a sequence diagram based on the FTP clone.

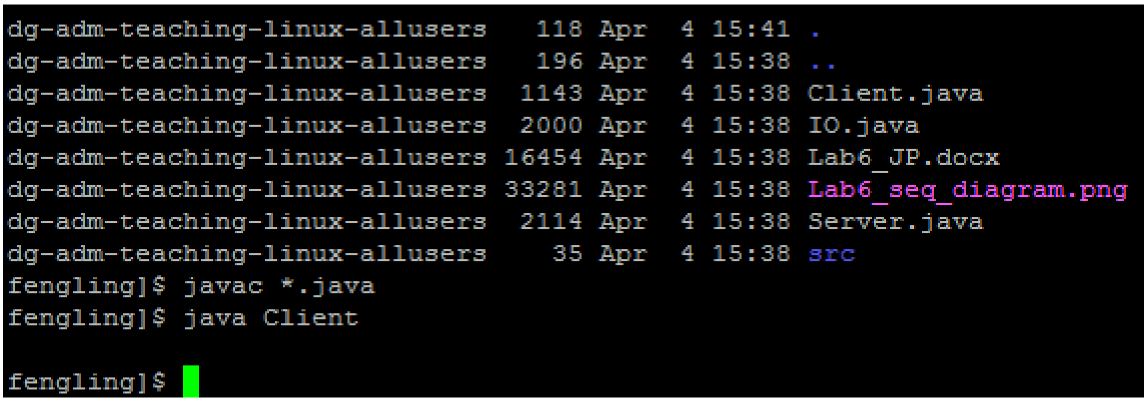
Hint: The server code and the 100_megabyte_file.zip file is provide, it is your task to program the Client and the IO classes. Furthermore, it's not mandatory to use the server code if you wish to supplement it with own server code.

Step 1: Run in netprog01 server



```
dg-adm-teaching-linux-allusers   107 Apr  4 15:40 .
dg-adm-teaching-linux-allusers   256 Apr  4 15:35 ..
dg-adm-teaching-linux-allusers  1139 Mar 22 20:33 Client.java
dg-adm-teaching-linux-allusers  2000 Mar 22 20:33 IO.java
dg-adm-teaching-linux-allusers 16454 Apr  4 13:48 Lab6_JP.docx
dg-adm-teaching-linux-allusers 33281 Mar 22 22:43 Lab6_seq_diagram.png
dg-adm-teaching-linux-allusers  2114 Mar 22 20:33 Server.java
Lab6_JP]$ javac Server.java
Lab6_JP]$ java Server
```

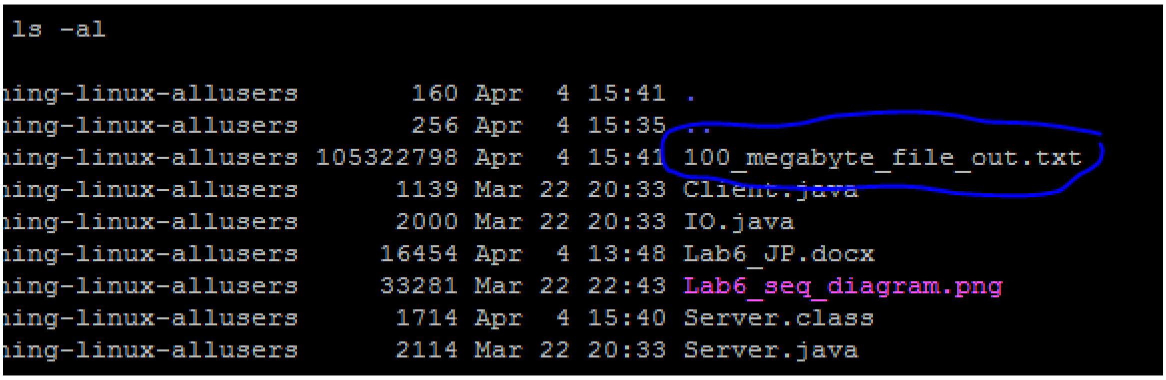
Step 2: Run in netprog02 server



```
dg-adm-teaching-linux-allusers   118 Apr  4 15:41 .
dg-adm-teaching-linux-allusers   196 Apr  4 15:38 ..
dg-adm-teaching-linux-allusers  1143 Apr  4 15:38 Client.java
dg-adm-teaching-linux-allusers  2000 Apr  4 15:38 IO.java
dg-adm-teaching-linux-allusers 16454 Apr  4 15:38 Lab6_JP.docx
dg-adm-teaching-linux-allusers 33281 Apr  4 15:38 Lab6_seq_diagram.png
dg-adm-teaching-linux-allusers  2114 Apr  4 15:38 Server.java
dg-adm-teaching-linux-allusers    35 Apr  4 15:38 src
fengling]$ javac *.java
fengling]$ java Client

fengling]$ 
```

Step 3: check the passed in file in netprog01 server



```
 ls -al

ing-linux-allusers       160 Apr  4 15:41 .
ing-linux-allusers       256 Apr  4 15:35 ..
ing-linux-allusers 105322798 Apr  4 15:41 100_megabyte_file_out.txt
ing-linux-allusers      1139 Mar 22 20:33 Client.java
ing-linux-allusers      2000 Mar 22 20:33 IO.java
ing-linux-allusers     16454 Apr  4 13:48 Lab6_JP.docx
ing-linux-allusers     33281 Mar 22 22:43 Lab6_seq_diagram.png
ing-linux-allusers      1714 Apr  4 15:40 Server.class
ing-linux-allusers      2114 Mar 22 20:33 Server.java
```

Week 6 tutorial questions

1. What is the asynchronous method to overwrite in the Thread class?
2. What is the best method for a thread to die?
3. How does a thread determine its Thread ID?
4. What the difference between extends and implements in regards to the Thread class?
5. What does “thread-safe” mean? What are the thread-safe alternatives?