CS451 Homework 2

Name: **Batkhishig** Dulamsurankhor
CWID: A20543498
Collaborator: None

The game server is deployed at 3.16.22.216:80(aws) and you can play using the client application.

Questions:

1. **Explain how this fits into the client server architecture.**
   First, there is a server application that handles the game logic for all users connected to the server and processes guesses by them and responds to each of them accordingly. Clients on the other hand connect to the server, send user guesses and receive game updates from the server. They communicate through socket connection. Each app has their own purposes.

2. **How would this work differently without threads? How would it work differently going from Linux to Windows, if there are differences?**
   Without threads, the gameplay would be much slower to start, especially when there are multiple players, because each game must be run sequentially. Players would have to wait for their previous player to finish his/her game.
   The application should work similarly on both Linux and Windows as long as they both have the same JDK installed(since I used java to write the application). From the user 's perspective, there shouldn't be any differences. However, system libraries for socket, threading etc might differ.

3. **Explain the flow of your implementation and what design choices you made**
   I have made the server to handle all the game logic, instructions to the client, while the client only shows messages sent by the server and delivers responses to it. Whenever a new client connects, the server spawns a new thread class called "BdulamsurankhorClientHandler" which handles game logic for the current connected client until the client decides to stop playing the game or the connection is broken. For example, when there are 3 players playing the game, the server will spawn 3 threads of "BdulamsurankhorClientHandler" class and each of the threads handles their own respective games. When the client connection terminates, BdulamsurankhorClientHandler class closes all resources and exits gracefully.
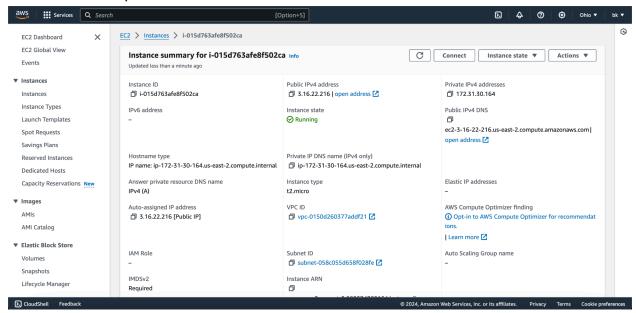
4. **I asked for TCP – what is the difference between tcp and udp and what additional logic would you need for UDP.**

The primary difference between tcp and udp is that tcp is connection oriented, with handshake ensures reliable connection, while udp is connectionless and just sends data without guarantee of delivering. Even though udp is lightweight and faster compared to tcp, it is less reliable.

If I switch from tcp to udp, I need to implement a logic to acknowledge when a packet is received by the client since there is no guarantee. I will also have to design a recovery mechanism when a packet is lost or packets are received out of order.

Screenshots:

1. AWS setup:

I have used AWS EC2 to deploy the hangman server application.

2. Deployment

```
[admin@ip-172-31-30-164:~$ ./deploy.sh
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [38 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [47 B]
Hit:2 https://cdn-aws.deb.debian.org/debian bookworm InRelease
Hit:3 https://cdn-aws.deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://cdn-aws.deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://cdn-aws.deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.39.2-1.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openjdk-17-jdk is already the newest version (17.0.12+7-2~deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ant is already the newest version (1.10.13-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Cloning into 'cs451-hw2'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 40 (delta 14), reused 35 (delta 9), pack-reused 0 (from 0)
Receiving objects: 100% (40/40), 237.78 KiB | 4.85 MiB/s, done.
Resolving deltas: 100% (14/14), done.
Buildfile: /home/admin/cs451-hw2/build.xml

compile:
    [mkdir] Created dir: /home/admin/cs451-hw2/bin
    [javac] /home/admin/cs451-hw2/build.xml:4: warning: 'includeantruntime' was not set, defaulting to
    [javac] Compiling 3 source files to /home/admin/cs451-hw2/bin
     [copy] Copying 1 file to /home/admin/cs451-hw2/bin
     [copy] Copying 1 file to /home/admin/cs451-hw2/bin

BUILD SUCCESSFUL
Total time: 2 seconds
[admin@ip-172-31-30-164:~$ systemctl status hangman.service
● hangman.service - Hangman Game Server
     Loaded: loaded (/etc/systemd/system/hangman.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-09-13 05:08:12 UTC; 29min ago
   Main PID: 12368 (sudo)
      Tasks: 15 (limit: 1137)
     Memory: 30.2M
        CPU: 1.647s
```

I have written a "deploy.sh" script to automate my deployment process. The file can be found in the zip file.

3. Gameplay, client loses the game and exits gracefully:

```
[batkhishig@dhcp124 bin % java src.BdulamsurankhorClient
Connected to the server!

 "_____"
 You have 10 lives left.
q

 "_____"
 You have 9 lives left.
w

 "_____"
 You have 8 lives left.
t

 "_____"
 You have 7 lives left.
u

 "_____"
 You have 6 lives left.
i

 "_____"
 You have 5 lives left.
o

 "_____"
 You have 4 lives left.
l

 "_____"
 You have 3 lives left.
k

 "_____"
 You have 2 lives left.
j

 "_____"
 You have 1 lives left.
h
Sorry, you have used up all 10 of your lives. The correct word was "spermary".
Do you want to play again? y or n.
n
java.io.EOFException
        at java.base/java.io.DataInputStream.readUnsignedShort(DataInputStream.java:346)
        at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:595)
        at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:570)
        at src.BdulamsurankhorClient.<init>(Unknown Source)
        at src.BdulamsurankhorClient.main(Unknown Source)
Resources closed, exiting gracefully...
batkhishig@dhcp124 bin %
```

Server console:

```
Sep 13 07:07:59 ip-172-31-30-164 sudo[13267]: [Server]: Client accepted
Sep 13 07:07:59 ip-172-31-30-164 sudo[13267]: Created a new game: spermary
Sep 13 07:08:45 ip-172-31-30-164 sudo[13267]: Client lost the game!
Sep 13 07:08:47 ip-172-31-30-164 sudo[13267]: Resources closed, thread terminating gracefully...
lines 582-637/637 (END)
```

4. Another gameplay, client wins the game, plays again and the client exits ungracefully(ctrl+c). The gameplay includes invalid characters and multiple character errors.

```
● ● ●                    📁 bin — -zsh — 89×57
[batkhishig@dhcp124 bin % java src.BdulamsurankhorClient
Connected to the server!

 "----------"
 You have 10 lives left.
a

 "a-a--------"
 You have 10 lives left.
n

 "ana-------n"
 You have 10 lives left.
c

 "anac------n"
 You have 10 lives left.
o

 "anaco----on"
 You have 10 lives left.
l

 "anacol---on"
 You have 10 lives left.
u

 "anacolu--on"
 You have 10 lives left.
t

 "anacolut-on"
 You have 10 lives left.
h
Congratulations! The correct word was indeed "anacoluthon"
Want to play again? y or n.
y

 "------"
 You have 10 lives left.
as
Invalid input!
!
Invalid character!
a

 "------"
 You have 9 lives left.
a
You have already chosen letter a.
^C%
batkhishig@dhcp124 bin %
```

Server console:

```
Sep 13 07:10:51 ip-172-31-30-164 sudo[13267]: [Server]: Client accepted
Sep 13 07:10:51 ip-172-31-30-164 sudo[13267]: Created a new game: anacoluthon
Sep 13 07:11:14 ip-172-31-30-164 sudo[13267]: Client won the game!
Sep 13 07:11:16 ip-172-31-30-164 sudo[13267]: Created a new game: stride
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]: java.io.EOFException
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.io.DataInputStream.readUnsignedShort(DataInputStream.java:337)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:583)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:558)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at src.BatkhishigClientHandler.run(Unknown Source)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]:           at java.base/java.lang.Thread.run(Thread.java:840)
Sep 13 07:11:33 ip-172-31-30-164 sudo[13267]: Resources closed, thread terminating gracefully...
lines 595-650/650 (END)
```