

CS550 Advanced Operating Systems
Programming Assignment 1
Design Documentation

Directory structure:

```
/CS550PA1-----|
|
|/files-----file 1
|               file 2
|               :
|               file n
IndexServer.java
Peer.java
IdGenerator.java
output.txt
build.xml
```

1. IndexServer.java

IndexServer class has a parameterized constructor `IndexServer(int)`. It accepts `port_number` and starts `IndexServer`.

`IndexServer` starts new thread for each client that connects.

All the major interaction with client is handled in separate new thread. Client is ready to accept new requests.

In `run` method, there is a switch case to serve different request from client. Right now there are two request that can be responded from Server, 1. Register and 2. Search,Download.

In `IndexServer` I have created two `HashMaps`,

1. `PeerList< Integer,String>`
2. `PeerFileList<Integer, File[]>`

`PeerList` `HashMap` stores unique ID for each client that connects and `Peer_port_number` which is obtained from client while registering.

so `PeerList` has a `<key ,value>` pair.

PeerFileList has unique ID for the client and File array mapping.

So ID serves as a kind of primary key through which we can find file belonging to peer.

So while searching for peer we need find the ID corresponding to required field and we can get the port numbers of peers from PeerList HashMap.

2. Peer.java

Peer class has two threads that run concurrently,

1. peer
2. client

peer thread acts as a server and continuously listens for other peers to connect. Peer thread mainly provides send file facility.

client thread has major work to do. client thread provides interactive menu to user. I have created switch case again. switch case is inside a continuous loop that makes it possible to have menu always available.

Register option in menu does two tasks,

1. creating new client socket. connection to server port.
2. registering files in /files directory to server

Register is one time task.

Search File actually server two big tasks,

1. Search for file
2. Download file

Right now I have combined these two tasks under one menu option.

When server returns list of peers with port number, the port number is automatically passed to socket and connection is established with corresponding peer.

IdGenerator.java

IdGenerator uses AtomicInteger class to generate unique IDs for clients.

build.xml

build.xml is right now very primitive it provides basic compile and clean tasks.

output.txt

I tried to create output.txt as kind of user friendly log of activities. Right now it records, Server start, client connection, file registration, number of files registered with each client, search query, search response, download cancellation, file sent status.

Tradeoffs/known Problems:

1. I could have used more modular approach.
2. The menu is not working properly. There is some problem with 'newline' character. One or more extra 'newline' character are causing menu to load more than one time before allowing user to enter choice.
3. If user selects to cancel download the client menu doesn't accept further inputs. Need to reconnect client
4. Multiple time registration is not handled.
5. Most validations are not done.

Improvements/Extension:

1. Need to make this more modular
 - a. Registration module
 - b. Search module
 - c. Download module

d. connection module – want to separate the code of connection from other activities.

2. Nested menu:

like when user selects to search file, user can be given option to 1. download 2. continue

3. Replication

4. Existing file check. This will warn user that he/she already has same file.

5. change directory option:

user can change shared directory or use multiple directories.