PEER TO PEER FILE SHARING SYSTEMS COURSEWORK

NAME	REGISTRATION	STUDENT NO
LUBWAMA ISAAC	**17/U/5739/PS**	**217017396**
NAKAYENGA VIOLA	**17/U/7565/PS**	**217003981**
MUKWATSE COLLIN	**17/U/6509/PS**	**217012303**
KAAYA MARVIN	**17/U/4253/EVE**	**217017391**

Entire project is designed using Python 3.8 where we used some network programming concepts such as sockets, multi-threading for establishing connections between peers. Major components of this project are:

- Indexing Server (Hash Tables & Distributed Hash Tables)
- Peer (Which acts as client and a server)

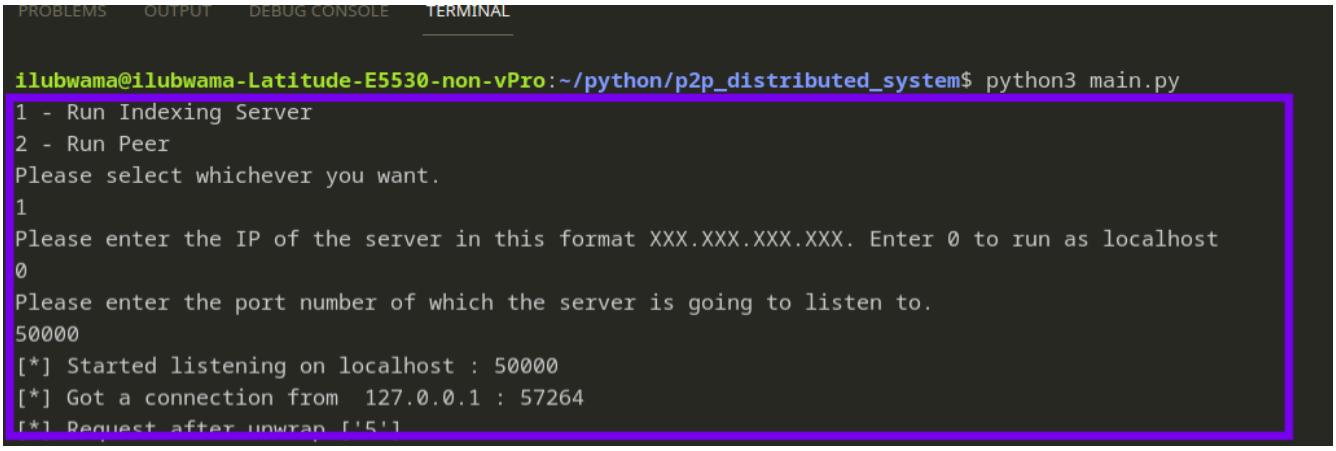
Indexing server(Hash Tables & Distributed Hash Tables)

The indexing server acts as Hash Tables & Distributed Hash Tables Indexing server indexes the content of all peers (Which have registered with it) using dictionary with a peer id attached to each peer. It provides two functions which are **register** and **search**.

Peer

Major function of the peer: Register and Listen to clients that wants to download (As a server). Search for a filename and ask to download it(As a client). Firstly, as a client, user can request a file name to the indexing server. The indexing server returns a peer data which shares this file with all of its details. The user then connects to this peer and downloads the file. Secondly, as a server,the peer waits for requests from other peers and sends the requested file when receiving a request. So peers here, act as both the client and the server. This server is different from the central index server which only indexes the files. But, the server functionality of the peer can be used to download the files from its directory. The peer acts a client to download the files from other peers into its directory.

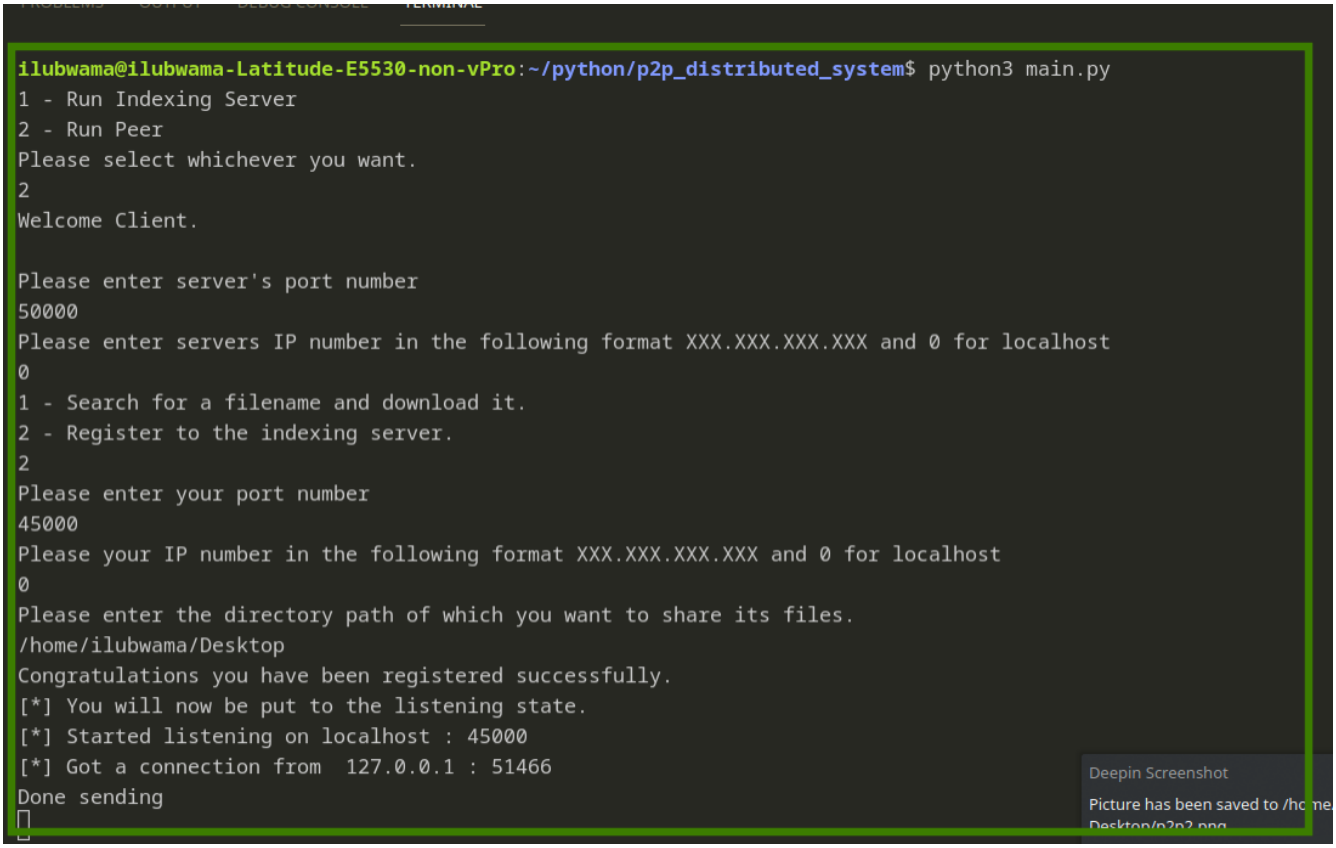
How-to



- Run main.py. Output should look like this:

You must run the indexing server first. So you choose "1"

In the above running example, we choose "0" to run a localhost (You can enter your machine IP), and for the listening port we choose 45000. Now server is waiting(listening) for incoming peers' requests. We run main.py one more time to run peer. This time we choose "2".



(Case that one or more peer has the file)

```
File isaac.py was found in the following one or more peers. Peer/s details are:

Peer ID: 1

Peer port: 45000

Peer host: localhost

File shared at: 2022-01-03 19:51:33

-----------------------------------
Peer ID: 2

Peer port: 45001

Peer host: localhost

File shared at: 2022-01-03 20:16:27

-----------------------------------
Do you want to download it (Y/N):
```

The screen shot below shows when a file has been downloaded successfully

```
Peer port: 45001

Peer host: localhost

File shared at: 2022-01-03 20:16:27

-----------------------------------
Do you want to download it (Y/N):
y
Please specify Peer ID
2
Successfully get the file
connection closed
ilubwama@ilubwama-Latitude-E5530-non-vPro:~/python/p2p_distributed_system$
```