


Manual

make sure you have gcc compiler installed

This project includes files to implement 4 peers and 2 indexing servers but you can have additional peers and servers the environment. If additional servers are added, changes are required in 'server_list.txt' file at each peer. You need a separate terminal for each one.

(instructions given in this manual are for 4 peers and 2 indexing servers)

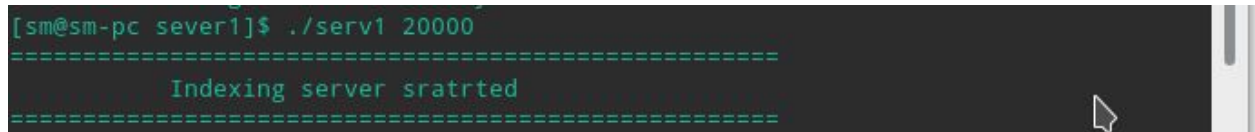
1. Open terminals in each directory of 4 peers and 2 servers
2. Use make command to compile code on terminals of 'p1' and 'server1'



```
p1 : bash — Konsole
[sm@sm-pc p1]$ make
g++ -pthread -o peer1 peer.cpp
g++ -pthread -o ../p2/peer2 peer.cpp
g++ -pthread -o ../p3/peer3 peer.cpp
g++ -pthread -o ../p4/peer4 peer.cpp
[sm@sm-pc p1]$

server1 : bash — Konsole
[sm@sm-pc sever1]$ make
g++ -pthread -o ../server2/serv2 server.cpp
g++ -pthread -o serv1 server.cpp
[sm@sm-pc sever1]$
```

3. On terminal in server1 Enter './serv1 20000' to start the server1 at port 20000. On terminal in server2 Enter './serv2 30000' to start the server2 at port 30000. You can use any available port number as long as same port number is used in client arguments and same is listed in server list of each relevant client too.



```
[sm@sm-pc sever1]$ ./serv1 20000
=====
Indexing server started
=====
```

4. On terminals of p1, p2, p3 and p4 enter './peer1 20000 9010', './peer2 20000 9020', './peer3 30000 9030', './peer4 30000 9040' to connect peer1 and peer2 to server1 and remaining to server2.



```
[sm@sm-pc p3]$ clear
[sm@sm-pc p3]$ ./peer3 30000 9030
port to act as server for this Peer :>9030

Choose action:
1.index existing file
2.serch file on servers
```

5. 90X0 are the port number at which that peer listens for connections from other peers. It should be unique for each peer as it is used to differentiate between two peers.
6. On Client terminal, you can choose between 2 options, make selections accordingly.
 - a. To register a file, enter its full name as well as extension. Eg."7.txt"

```
1
enter file name to register :
7.txt
7.txt registered with server

Choose action:
1.index existing file
2.serch file on servers
```

- b. To lookup a file, enter its full name as well as extension. Eg."7.txt"
7. After you have made a lookup query if the file is not found, appropriate message is displayed.

If the file is found, on local server, peer does not search for file on other servers. list of ports on which the file is available.

```
2
enter file name to search:
7.txt
file available on ports:
9010
enter peer number to download file or 0 to exit.
```

If file is not found on local server, peer searches for it on other servers

```
2
enter file name to search:
7.txt
file not found on this original server. connecting to other servers:
connected to server at: 20000
file available on ports:
9010
enter peer number to download file or 0 to exit.
```

At this point you can select a port to download from or to exit from the lookup query without downloading.

If not found, appropriate message is displayed

8. After selecting a port to download file from, message is displayed if the file is successfully downloaded and you can check it in the respective peer's directory.

```
enter peer number to download file or 0 to exit.  
9010  
connected to file source peer at: 9010  
file transfer complete  
7.txt of size 8219 bytes downloaded in 0.245 seconds!
```

9. You can repeat the register and lookup queries for as long as required.