Programming Assignment 3:

Prerequisite:

- The working environment should have Java-1.8 installed.
- Apache Ant-1.9.3 and above.

Steps to Execute:

- Extract PROG3_KALE_ADITYA.tar to a particular location in your Linux environment. Go to above extracted location from terminal. Go to project **DistributedPeerToPeerFileTransfer**.
- Execute **command** / **DistributedPeerToPeerFileTransfer** $ **ant**
  Build.xml gets execute and generates a jar. A **dist** folder is generated which contains **DistributedP2PFileTransfer.jar.**

Steps to start Multiple Clients.

- Go to folder resources and open **config.properties** update noOfservers variable value. It is the number of Clients that will act as a server to put values in hashtable.

  Ex: **noOfServers** = 8

- Update replica1 and replica2 value to true/false. It is used for replication of files in system. System will replicate files only if you set value of this parameter to true.

  Ex: **replica1** = true
  replica2 = false

- Update all serverIp variable value with Ipaddress of those machine which will be used in this system.

  Ex: **serverIp1** = 192.168.239.131

  serverIp2 = 192.168.139.132

- Update all serverPort variable value with port numbers which are available for communication on above mentioned machines. Make sure that serverport1 is a port used by machine serverIp1 for communication.

  Ex: **srerverPort1** = 9991

  srerverPort2 = 8787

- Update all fileTransferPortServer variable value with port numbers which are available for file transfer communication on above mentioned machines. Make sure that serverport1 is a port used by machine serverIp1 for file transfer communication.

  Ex: **fileTransferPortServer1** = 9001

  fileTransferPortServer2 = 9002

  Note: If you want to start more than 8 clients then you need to add serverIp and serverPort variables similar to given variables in config.properties.

Steps to run multiple Clients.

- Copy the same project structure and configuration to multiple Linux machine which are connected in a single network.
- Before starting every client you need to update the variable in config.properties **currentMachineIp** to an IP of that particular machine.
  Ex: **currentMachineIp** = 192.168.26.60

  currentMachinePort to a port number of that particular machine.
  Ex: **currentMachinePort** = 9991

  currentMachinePortForSendingFile to a port number used for sending files of that particular machine.
  Ex: **currentMachinePortForSendingFile** = 9001

- Go to above project location from terminal and execute following commands
- For Linux Environment
  - Execute **command** / **DistributedPeerToPeerFileTransfer** $ **./runClient.sh**
    - Clients gets Start and ready to accept requests from another clients.

Note: Before starting any operations you need to check that all clients are up and ready to listen request from another clients. Otherwise system will not work.

  - Now you can give input from client.
    1. **Register Files**→ It will register all files of "ClientData" folder and will store their locations in a distributed hash table of some machine
    2. **Search for a file name**→ It will take file name as an input from user and will search that file in distributed hash table. And will display on which server that value is present.
    3. **Get all file names to download** →It will display all file names which user can download.
    4. **Download a file** → It will take file name as an input from user and will search that file in distributed hash table and will download that file in "Downloaded" folder if found on any server.
    5. **Exit** → Will terminate the program.
    6. Also can Use **ctrl + C** to terminate the program.

For Windows Environment

  - Execute **command** \ **DistributedPeerToPeerFileTransfer** > **java –cp dist\ DistributedP2PFileTransfer.jar edu.distributedFileSystem.Client**
    - Client gets start and ready to listen other peer request.