Advanced OS
Assignment #3 - Peer-to-Peer Group Based File Sharing System
Hard - Deadline : 08/11/2021 (11:55 pm)

Guidelines:
Languages Allowed: C/C++
Submission format: <rollno>_a3.zip
ZERO tolerance towards any kind of code plagiarism. Plagiarism will fetch you a ZERO.

Pre-requisites:
Socket Programming, SHA1 hash, Multi-threading

Goal:
In this assignment, you need to build a group based file sharing system where users can share, download files from the group they belong to. Download should be parallel with multiple pieces from multiple peers.

Note:
- You have to divide the file into logical “pieces” , wherein the size of each piece should be 512KB
- SHA1: Suppose the file size is 1024KB, then divide it into two pieces of 512KB each and take SHA1 hash of each part, assume that the hashes are HASH1 & HASH2 then the corresponding hash string would be H1H2 , where H1 & H2 are starting 20 characters of HASH1 & HASH2 respectively and hence H1H2 is 40 characters
- Authentication for login needs to be done
- Error handling to be done(else marks will be deducted)
- You can implement your own version of torrent architecture i.e., you need not implement exact same protocols mentioned in the bit torrent paper. You can design your own algorithm of your own architecture.
- You need to design your own Piece Selection Algorithm.
- **Zip file should contain only 2 files(tracker.cpp and client.cpp).**

Architecture Overview:

The Following entities will be present in the network :

1. Synchronized trackers(2 tracker system) :
    a. Maintain information of clients with their files(shared by client) to assist the clients for the communication between peers
    b. Trackers should be synchronized i.e all the trackers if online should be in sync with each other
2. Clients:
    a. User should create an account and register with tracker
    b. Login using the user credentials
    c. Create Group and hence will become owner of that group
    d. Fetch list of all Groups in server
    e. Request to Join Group
    f. Leave Group
    g. Accept Group join requests (if owner)
    h. Share file across group: Share the filename and SHA1 hash of the complete file as well as piecewise SHA1 with the tracker
    i. Fetch list of all sharable files in a Group
    j. Download file
        i. Retrieve peer information from tracker for the file
        ii. Core Part: Download file from multiple peers (different pieces of file from different peers - piece selection algorithm) simultaneously and all the files which client downloads will be shareable to other users in the same group. Ensure file integrity from SHA1 comparison
    k. Show downloads
    l. Stop sharing file
    m. Stop sharing all files(Logout)
    n. Whenever client logins, all previously shared files before logout should automatically be on sharing mode

Working:

1. At Least one tracker will always be online.
2. Client needs to create an account (userid and password) in order to be part of the network.
3. Client can create any number of groups(groupid should be different) and hence will be owner of those groups
4. Client needs to be part of the group from which it wants to download the file
5. Client will send join request to join a group
6. Owner Client Will Accept/Reject the request
7. After joining group ,client can see list of all the shareable files in the group
8. Client can share file in any group (note: file will not get uploaded to tracker but only the <ip>:<port> of the client for that file)
9. Client can send the download command to tracker with the group name and filename and tracker will send the details of the group members which are currently sharing that particular file
10. After fetching the peer info from the tracker, client will communicate with peers about the portions of the file they contain and hence accordingly decide which part of data to take from which peer (You need to design your own Piece Selection Algorithm)
11. As soon as a piece of file gets downloaded it should be available for sharing
12. After logout, the client should temporarily stop sharing the currently shared files till the next login
13. All trackers need to be in sync with each other

Commands:

1. Tracker:
    a. Run Tracker: ./tracker tracker_info.txt tracker_no tracker_info.txt - Contains ip, port details of all the trackers
    b. Close Tracker: quit
2. Client:
    a. Run Client: `./client <IP>:<PORT> tracker_info.txt`
        tracker_info.txt - Contains ip, port details of all the trackers
    b. Create User Account: `create_user <user_id> <passwd>`
    c. Login: `login <user_id> <passwd>`
    d. Create Group: `create_group <group_id>`
    e. Join Group: `join_group <group_id>`
    f. Leave Group: `leave_group <group_id>`
    g. List pending join: `requests list_requests <group_id>`
    h. Accept Group Joining Request: `accept_request <group_id> <user_id>`
    i. List All Group In Network: `list_groups`
    j. List All sharable Files In Group: `list_files <group_id>`
    k. Upload File: `upload_file <file_path> <group_id>`
    l. Download File: `download_file <group_id> <file_name> <destination_path>`
    m. Logout: `logout`
    n. Show_downloads: `show_downloads`
        Output format:
        [D] [grp_id] filename
        [C] [grp_id] filename
        D(Downloading), C(Complete)
    o. Stop sharing: `stop_share <group_id> <file_name>`