CS 378 Computer Networks Lab

Socket Programming: Distributed File System

Project Readme

Astha Agarwal (110050018)

Anmol Garg (110050020)

Rahul Singhal (110050023)

File System

1. **Folders**:
    a. **server**: contains the source files for running the server.
        i. FileMeshNode.cpp
        ii. node.cpp
        iii. node.h
        iv. nodeData.cpp
        v. nodeData.h
    b. **client**: contains the source files for running the client.
        i. client.cpp
        ii. nodeData.cpp
        iii. nodeData.h

2. **files:**
    a. setup.sh: A bash script to setup server nodes, client and cleaning the directory.
    b. FileMesh.cfg: A sample configuration file.
    c. Readme.pdf : This file.

3. **Files and folders that are created after running the client and server:**
    a. All the server nodes will create their own storage folder according to the specifications given in the configuration file. If the folder already exists, nothing happens else a new folder is created.
    b. The client will produce a "**received**" folder. It will store all the data and files received in this folder.

Compilation & Running Instructions

- **Go to the root folder of the project which contains above listed folder and files.**
- **Make the bash script executable by running the command "chmod +x setup.sh"**

- **Server:**
    a. Run the command:
    " ./setup.sh server <path_to_configuration_file> "
    ➢ This will generate a binary named "**server**" in the **bin** folder, automatically provide it the configuration file and fire it up.

- **Client:**
    a. Run the command:
    " ./setup.sh client <path_to_configuration_file> "
    ➢ This will generate a binary named "**client**" in the **bin** folder, automatically provide it the configuration file and fire it up.

- **Clean the folder:**
    a. Run the command:
    " ./setup.sh clean "

Manual Compilation instructions:

1. We have provided a bash script "setup.sh" that automatically compiles and runs the executables that are needed.
2. In case you want to manually compile the source files, run the commands below(Make sure you are in the root folder of the project):
    a. mkdir bin [make directory to store the produced binaries]
    b. g++ client/*.cpp -lcrypto -o bin/client [client compilation]
    c. g++ server/*.cpp -o bin/server [server compilation]
3. These commands will produce the required binaries in the **bin** folder.

Configuration File Format:

The format of configuration file is exactly the same as that given in the problem statement.

"<IP address1>:<Port_no1> <path_to_storage_folder1>"
"<IP address2>:<Port_no2> <path_to_storage_folder2>"
"<IP address3>:<Port_no3> <path_to_storage_folder3>"
so on..

See the sample file "FileMesh.cfg".