REPORT ON UNIX PROGRAMMING

MEMBER: Vũ Bình Dương - USTHBI5-030
Trần Trọng Khánh - USTHBI5-062

Work distribution

Mainly part 1: Vũ Bình Dương.
Mainly part 2: Trần Trọng Khánh.
We discussed to each other if any members got problems.

PART 1: Shell Script and C/C++ Programming

I/ Requirements

Project4: Write a program with the following features:
1. List all the video files from a folder (including sub-folders).
2. Sort files by file name, filesize, created date.
3. Detect and highlight all the video file that have the same name (possible redundancy)
4. Play a video by clicking on the video path.
5. Search video files by keyword.

The Graphical User Interface is written in C/C++.
All commands are written in Shell Script.

II/ Process

- The picture below is the User Interface when user run program

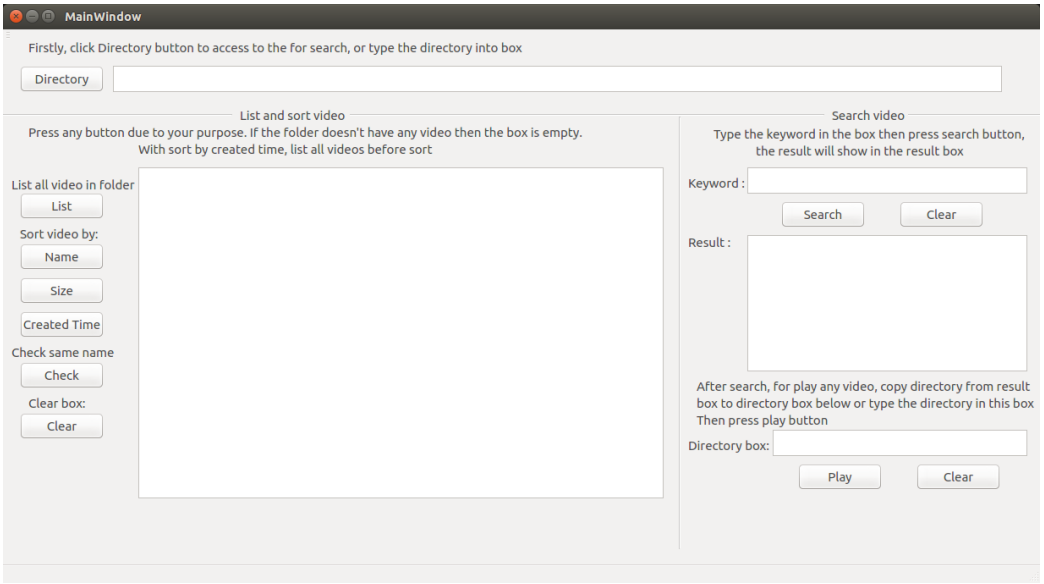


Figure 1.1: Initial User Interface

- First of all, the users must choose the directory of folder which users want to list all video file or search video by clicking Directory button then find directory in file dialog, or typing directly directory in box.

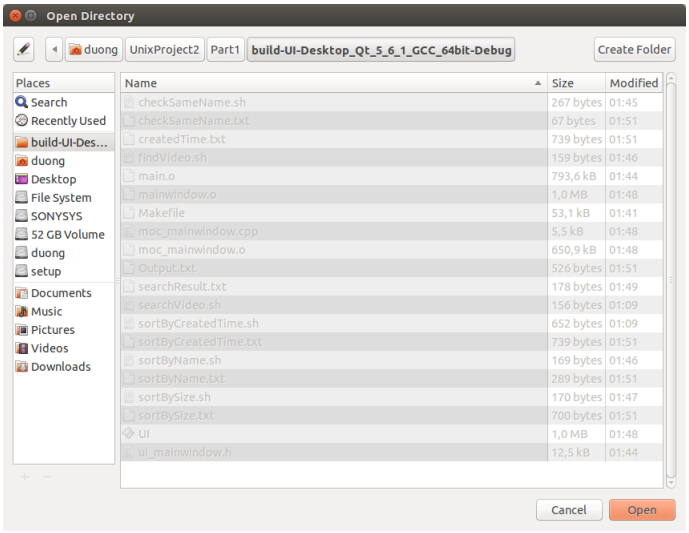


Figure 1.2: File dialog

- Then, due to users' purpose:
    + List all video in folder: clicking “List” button
    + Sort video by name: clicking “Name” button
    + Sort video by size: clicking “Size” button
    + Sort video by created time: clicking “Created Time” button
    + Check videos which have same name: clicking “Check” button

=> See the in the box next to buttons. To clear text in box, clicking “Clear” button

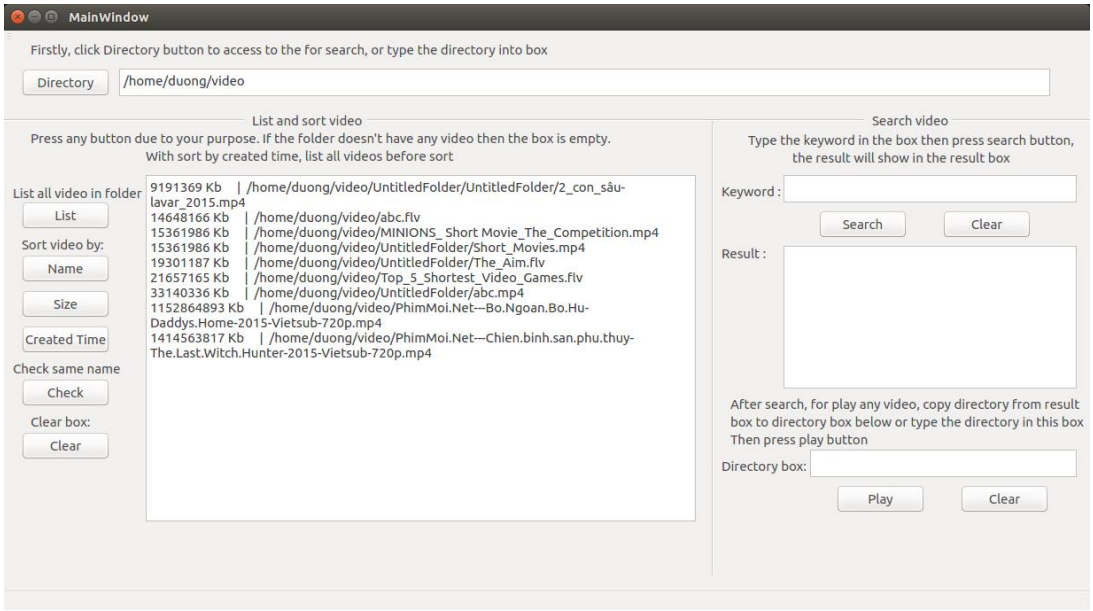


Figure 1.3: Result of sorting video in /home/duong/video

- To search or play video, users use the right part. After fill directory, users type their keyword then click “Search” button, the search result will be shown in the Result box.

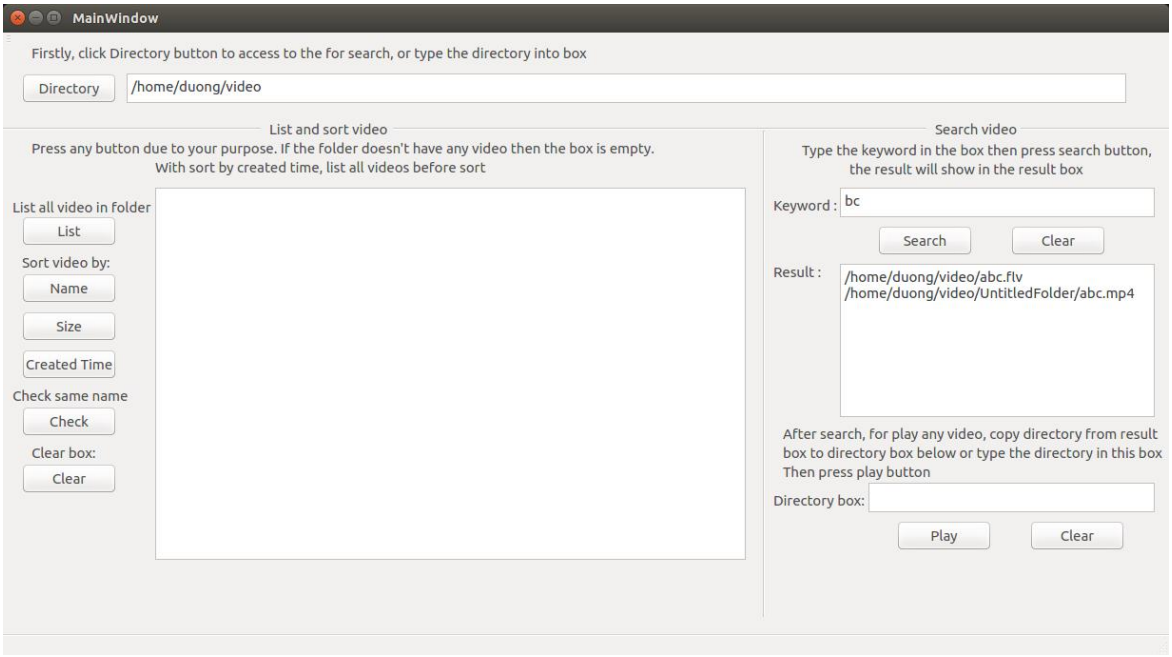


Figure 1.4: Search results

- To play any video after search, users provide directory of this video in Directory box then click “Play” button

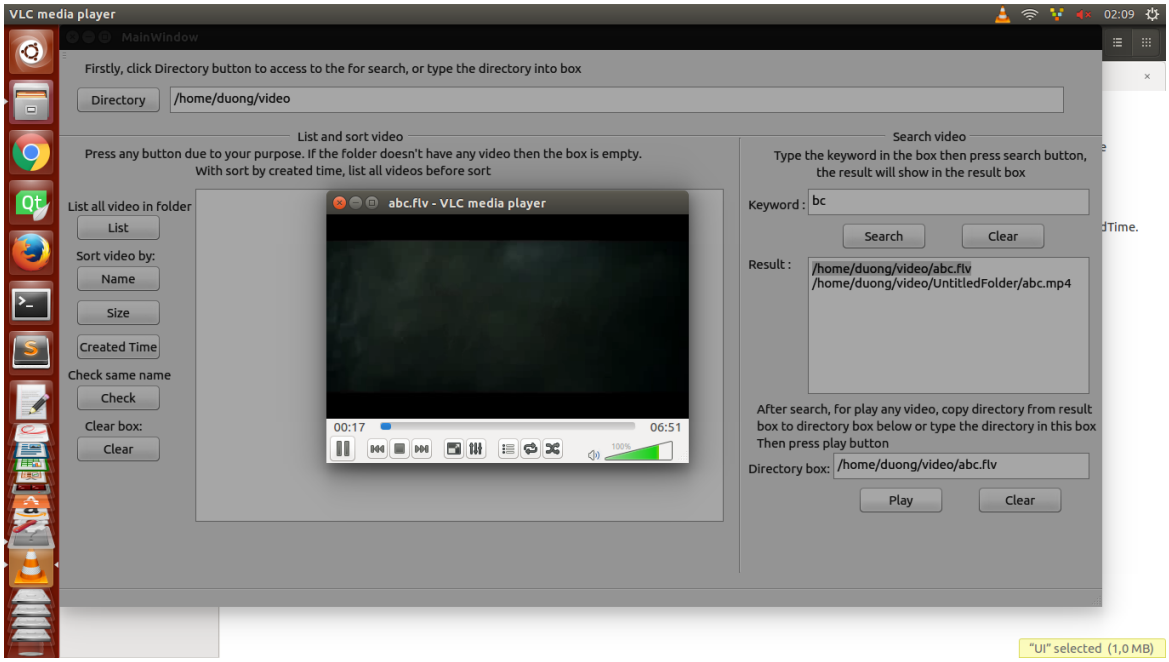


Figure 1.5: Play video abc.flv

III/ Difficulty

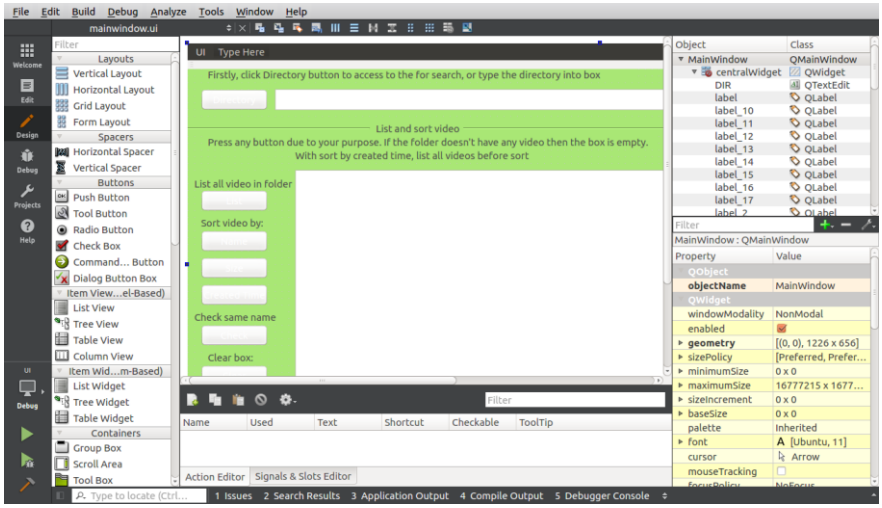
- During working process in part 1, the syntax of shell script is quite hard to write down. I need more time to understand more about this.
- New framework namely qt5 is other problem. We got many error when defining function in qt to create UI so it's necessary to study about qt. We don't have enough time to create UI for part 2.

IV/ Achievement

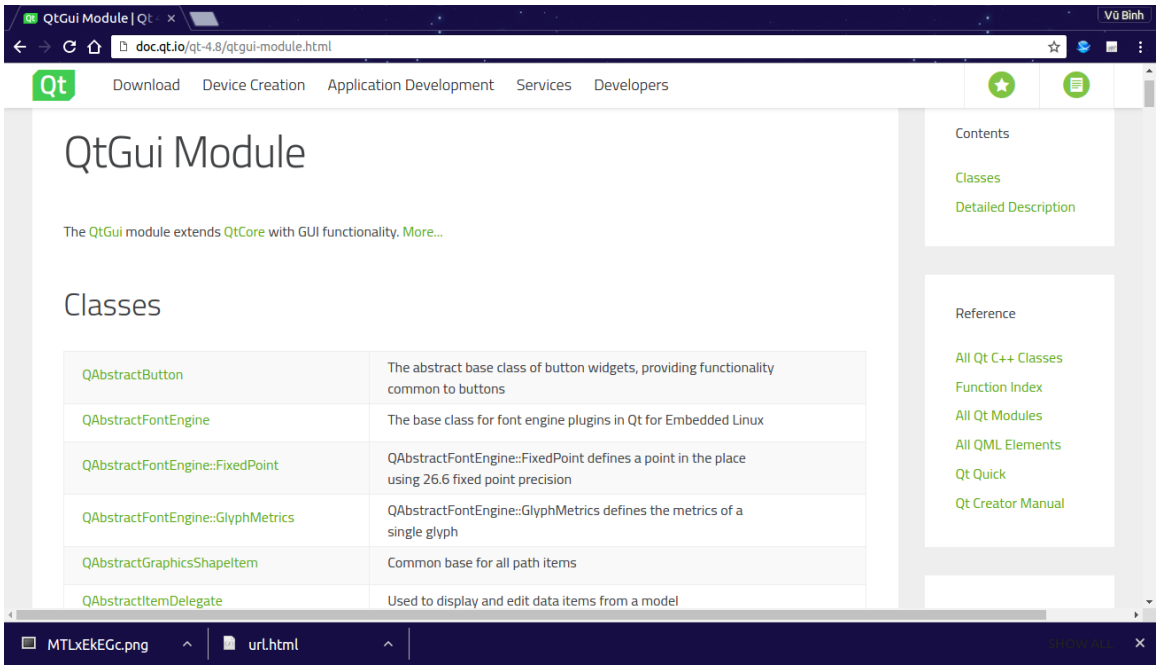
- Understand more about shell script
- After project, we become amateur in qt and know a new framework to create UI

IV, Support Tools

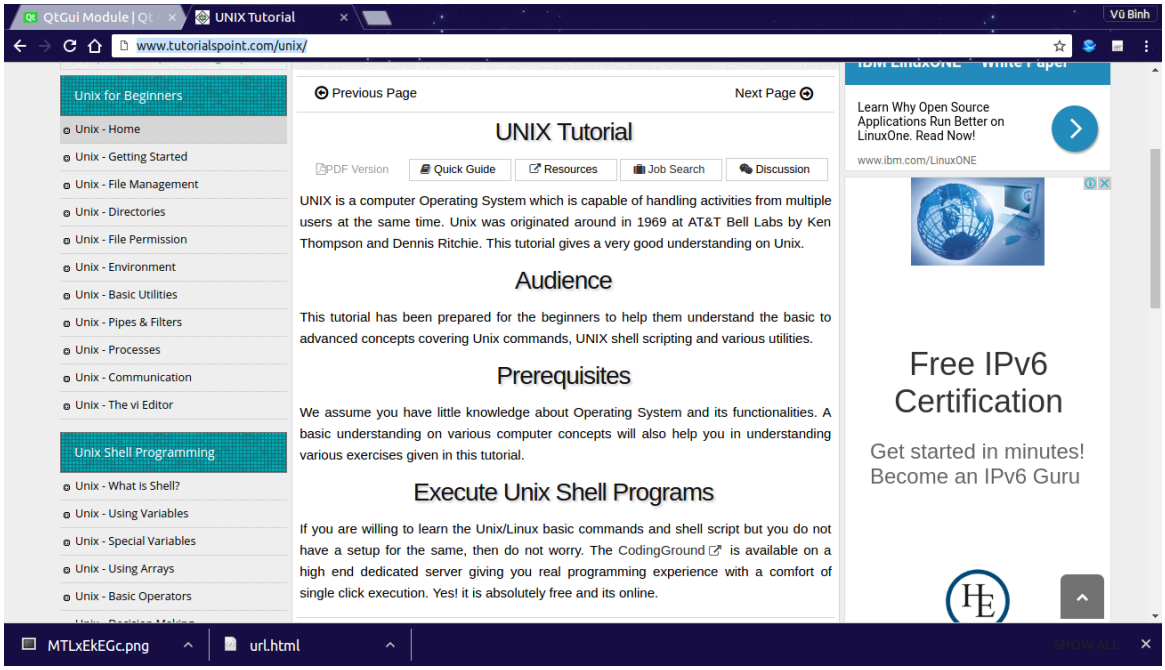
- QT creator



- web doc.qt.io



- web http://www.tutorialspoint.com/unix/



PART 2: Network Programming and SQL

I/ Requirement

Project2: Chat room with multiple users. Chat history is stored in a database

II/ Process

+ Database system: SQLite version 3

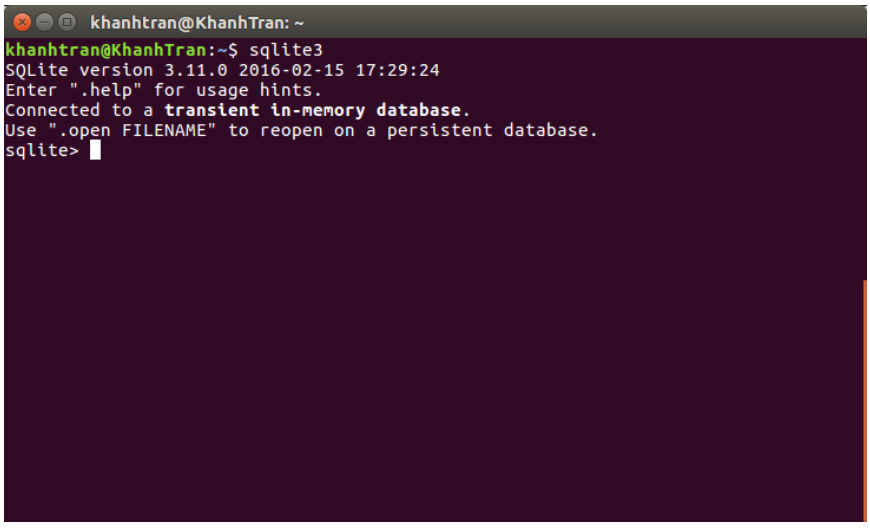


Figure 2.1: SQLite Version

1. Compile files

- Compile these files: ***client.c, server.c*** and ***create_table.c***.
- ***server.c*** and ***create_table.c*** requires ***sqlite3*** library to run.

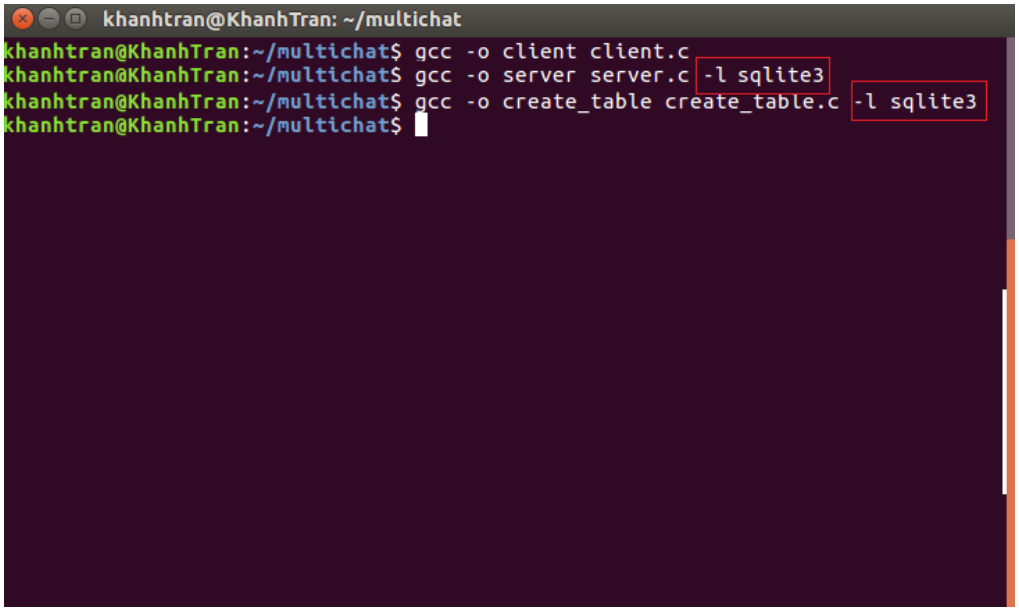


Figure 2.2: Compile files in terminal

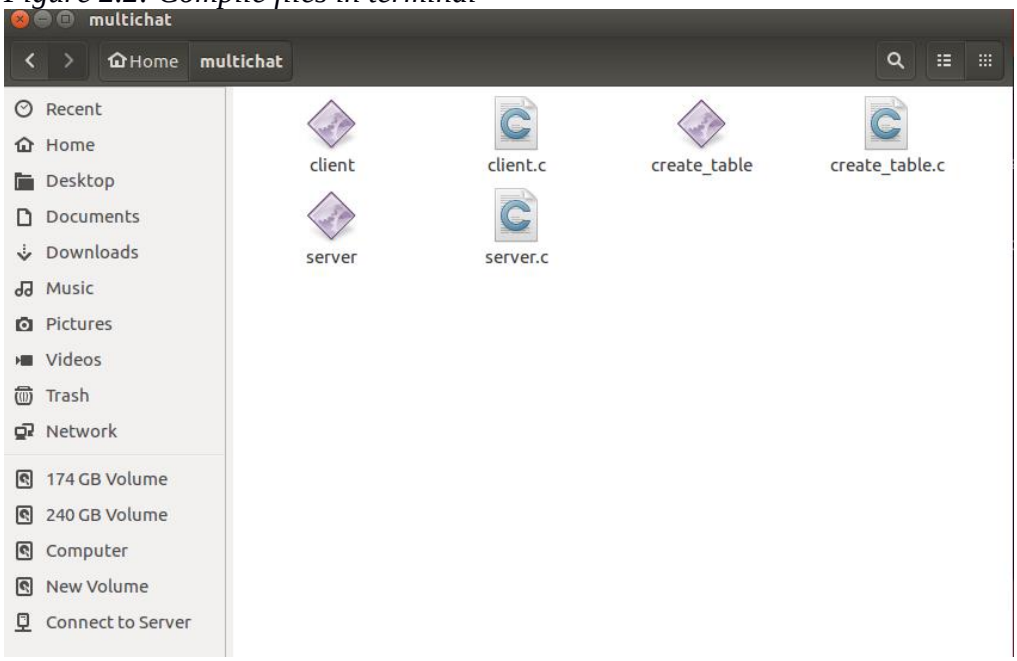


Figure 2.3: Repository after compiling

2. Execute files

- Execute ***server***.
- Execute ***create_table*** to create database.

Database file: ***"my_database.db"***.
my_database.db creates table storing information named: ***"my_table"***

- Execute ***clients***.

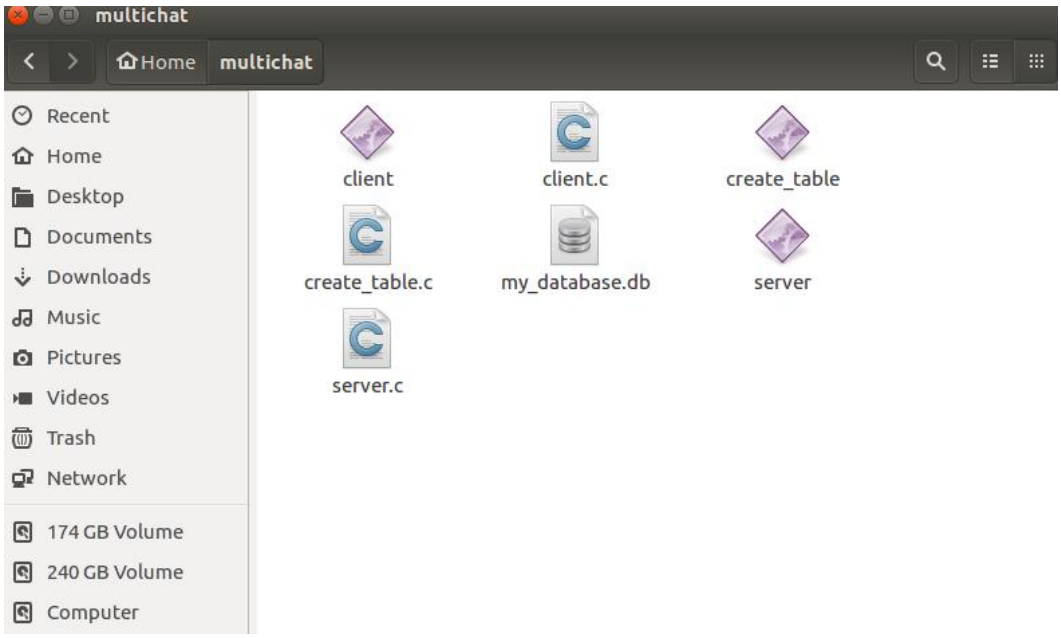


Figure 2.4: Repository after executing

Whenever the client is executed, server will notice the message "New connection from /***IP Address***/ on port /***port number***/ ".

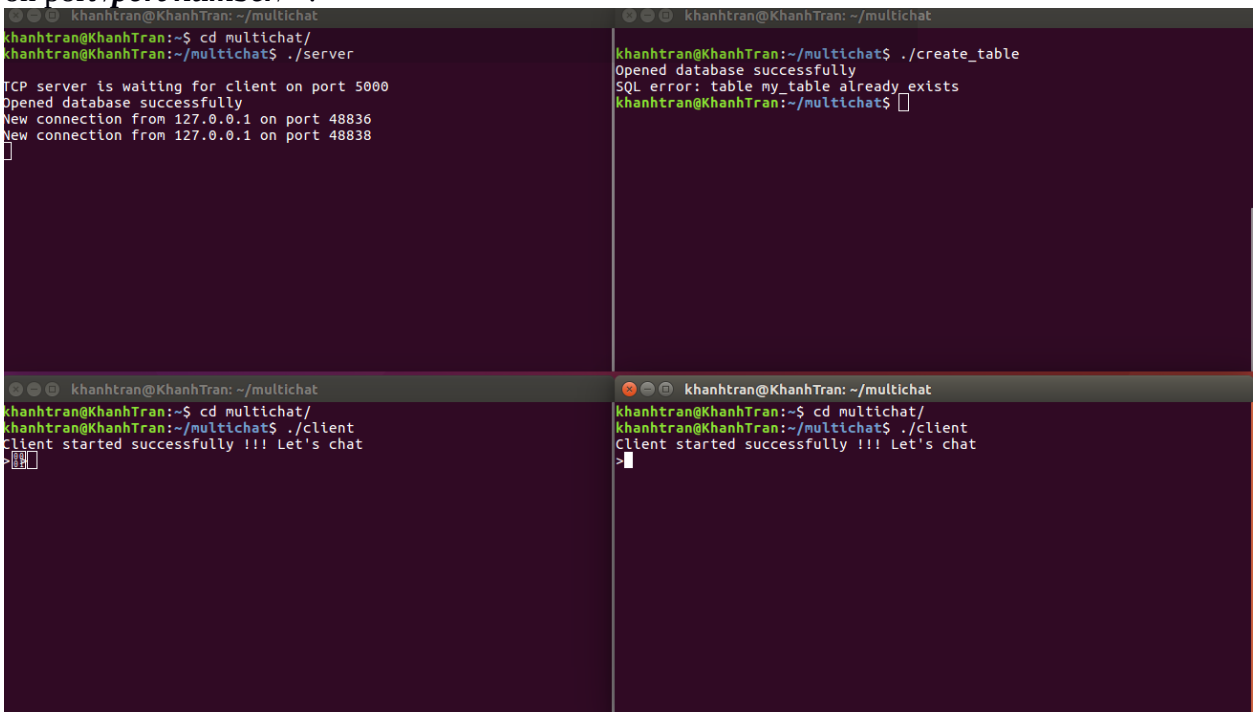
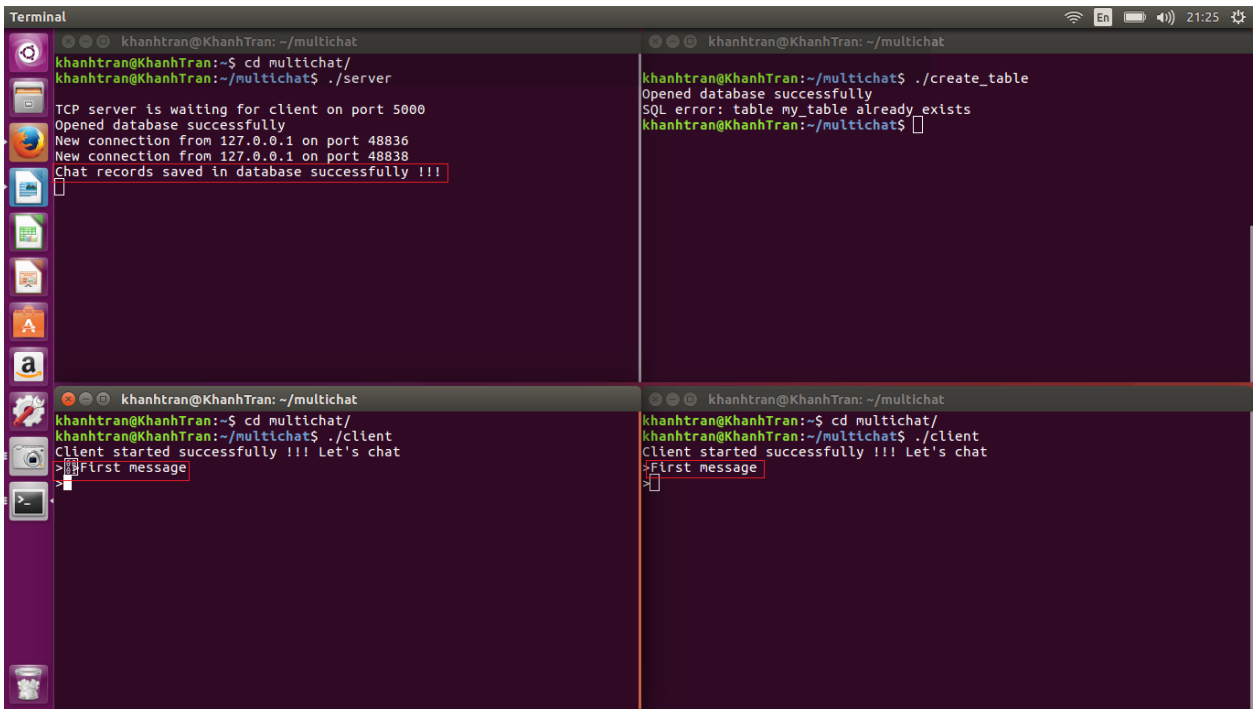


Figure 2.5: Socket programming | Chatting server-clients system

First client starts chatting, then other client will receive message. After each chat, server will notice the message "Chat records saved in database successfully".



3. Check the database to see the message history

+ Using command "***sqlite3 my_database.db***" where ***my_database.db*** stored information in database.
+ Then, type command ***sqlite>select * from my_table*** to show table.
(* symbol: show all attributes of table).
Table attributes: Order number | Date | IP Address | Port number | Message content.

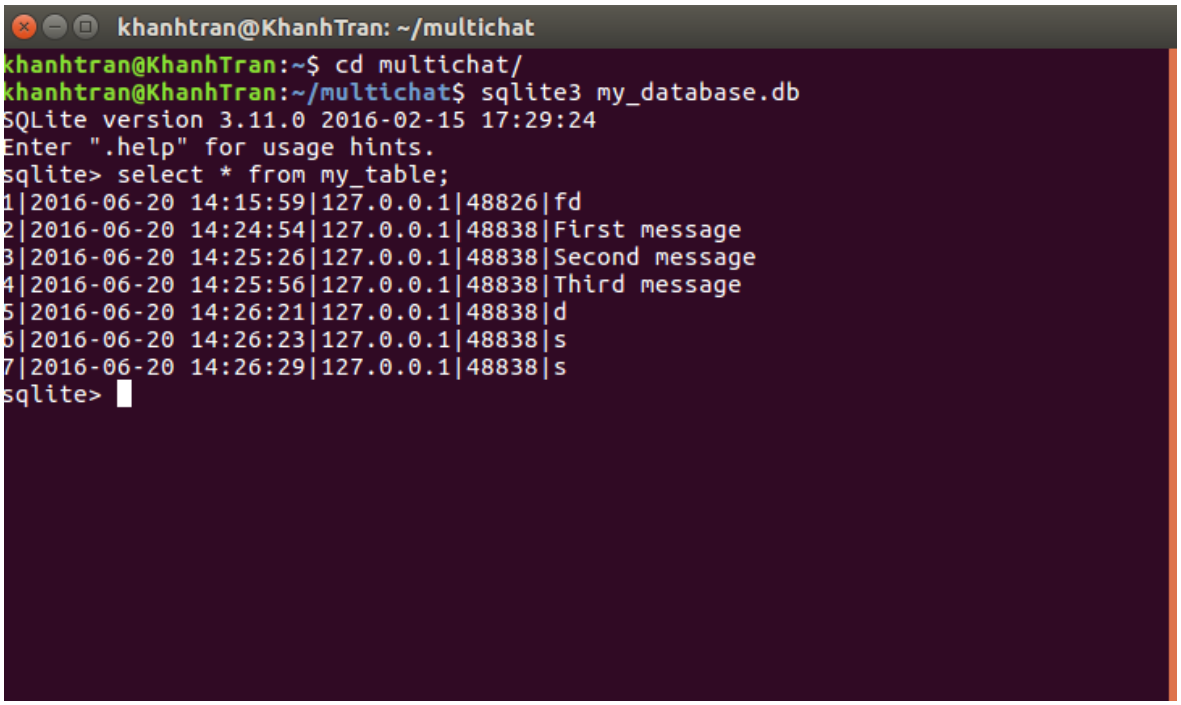


Figure 2.6: SQLite 3 database history

III/ Difficulty

It took much time to figure out what Structured Query Language should be used. I've decided to choose the database system SQLite version 3. Then I found it hard to use SQLite command, code. I had to refer some SQLite codes on Internet, and read to understand them.

IV/ Achievement

+ Socket Programming: is the connection between server and clients through the socket.
+ How server and clients contact to each other: Whenever client sends message, it will be move through server and saved in database using SQLite3 system.

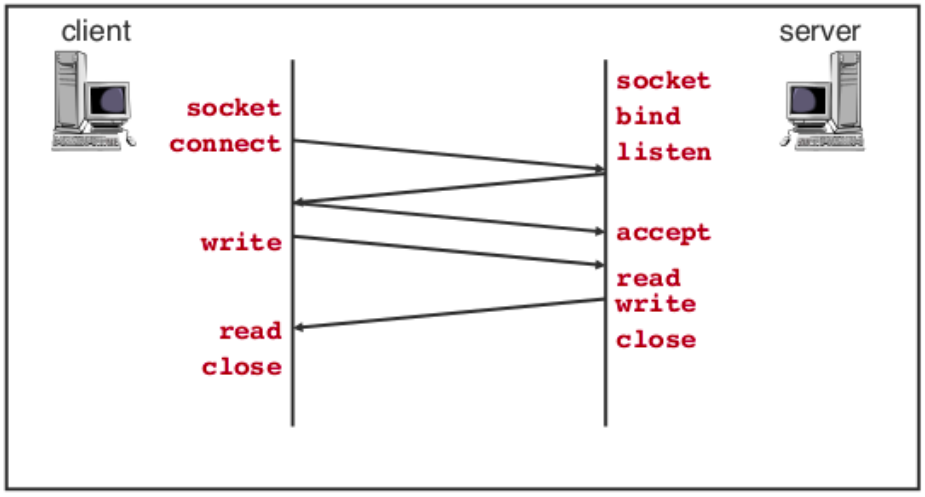


Figure 2.7: Server-client working mechanism

+ Client: Create function ***socket()***, connect to server by function ***connect()***, send and receive data by function ***read()*** and ***write()***, and finish by function ***close()***.
+ Server: Create the socket by function ***socket()***, bind socket to server address using function ***bind()***, listen the connection from clients by function ***listen()***, ***accept()*** agree the connection from clients, send and receive data by function ***read()*** and ***write()***, an finish by function closed.
+ Database system: SQLite 3 is a relational database management system contained in a C programming library. It must be included the ***#include <sqlite3.h>***. SQLite is available to Ubuntu and many operating systems. SQLite is light, confidential, open source and easy to use.