CS3516 A15

Help Session 1

Presented by Oleksandr Narykov
onarykov@wpi.edu

Outline

- Project 1 Overview
- Processing commands
- How to find help and other tips.

CS3516 Project1

- Your programs should compile and work on ccc.wpi.edu computers, which are running Linux.
- Programs should be written in C or C++.
- If your program is developed on another platform or machine, you should test the software on ccc before turning in the assignment.
- Make sure you have the correct #include commands in your program.

Project 1 missions (in handout)

- The Client:
  1. Reading a command from a script file "*LClient.txt*" or from console.
  2. Sending the command to the server.
  3. Receiving and displaying the information from the server.
  4. Writing the results to the log file *LClient.log*.

Project 1 missions (in handout)

- Server:
  1. Processing the command from the client and return the result to the client.
  2. Maintaining the records to keep the location information.
  3. Writing the complete database to the file *LDatabase.txt* when the server received the "quit EOF" command.

Outline

- Project 1 Overview
- Processing commands
- How to find help and other tips.

Processing commands

- Each command triggers a communication conversion, between client and server. Then, we have
  - login
  - add
  - update
  - remove
  - find
  - locate
  - quit
  - *list (attn: this one is different from above commands, most complex one).*

Commands

- In the *login, add, remove*, and *quit* commands:

  The server only returns one message to the client.

- In the *list, find, locate commands,* the server may return multiple messages to the client.

  "Each entry, which meets the search condition, is sent as a separate TCP message back to the Client."

Login Command

- Login Command Format.

login name

- Login Command Handling
  - For The Client: When the Client reads a login command, the client establishes a TCP connection to the Server.
  - For The Server: When the Server receives a "login name", it replies "Hello, name!" to the client.

Add Command

- **Add Command Format:**

  add id_number first_name last_name gender location

  Notes:

  - first_name, last_name, gender and location are nonblank ASCII string. For example:

  add 321654987 Tony Smith M 12_Institute_rd_worcester

  - id_number is 9 digital number similar to SSN number. (example shown: 321654987)

- **For the Client:**

  reads and sends the add command to the server, and displays the result returned from server.

Add Command (cont'd)

- **For the Server:**

  When the server gets the *Add* command, it will

  - add the five items as an entry into the location database in the proper location, and return a successful message to client.
  - If a duplicate *id_number* is received, the server sends an error message back to the client.
  - If the command's parameter is not valid, the server returns an Error message to the client.

    For example,

    Add 120335 Tony Smith M worcester, MA

    returns "an invalid add command".

Find Command

- Find Command Format:

  find *first_name last_name*

  Notes:

  - first_name, last_name are nonblank ASCII string. For example

    find Donald Trump

- For the Client:

  reads and sends the find command to the server, and displays the result returned from server.

Find Command (cont'd)

- **For the Server:**

  When the server gets the Find command, it will

  - Search database entries
  - Return matched results. One TCP packet per result.
  - If nothing was found, return message indicating this.

Locate Command

- Locate Command Format:

  locate *location*

- For the Client:

  reads and sends the locate command to the server, and displays the result returned from server.

Find Command (cont'd)

- **For the Server:**

  When the server gets the Locate command, it will

  - Search database entries.
  - All entries with matching location would be sent to the client. One TCP packet per entry. Entries should be sent in **alphabetical** order by last name!
  - If nothing was found, return message indicating this.

  - **Note – location has to be an exact match but case-insensitive.**

Update Command

- Update Command Format:

  update *id_number first_name last_name gender location*

  Notes:

  - Requirements are the same as for add command

- For the Client:

  reads and sends the update command to the server, and displays the result returned from server.

- For the Server:

  sends back an error of id_number is not in the database.

Remove Command

- Remove command format

  remove *id_number*

  example: "remove 123456789" *is a valid command.*

- For the Client,

  sends the remove command to the server, and displays the result returned from server.

Remove command (cont'd)

- **For the Server,**

  When the server receives remove command, the server searches the database for a match on *id_number*.

  - If the *id_number* entry exists in the database for a person, that entry is removed from the location database and a **success** message that contains the first and last name of the person removed is sent back to the Client.
  - If there is not a match in the database, the server does not modify the database and sends an appropriate **error** message back to the Client.

Quit Command

- **Quit Command format:**

quit [EOF]

For example, quit and quit EOF are valid commands.

- **For the Client**
  - **sends the quit command to the server, and when the client received the response message from the server, the client knows the connection will be closed.**
  - **If EOF is specified, the client will close the log file, and terminate.**

Quit Command (Cont'd)

- For the Server,
  - When server received quit command, it sends a response back to the Client indicating that the connection will be closed and including a count of the number of commands that are issued by *name*. The server returns to wait for a new connection triggered by a subsequent login request.
  - If quit EOF is received, the Server additionally writes out the complete database to the file *LDatabase.txt* and sends back to the Location Client a *count* of the number of clients processed, then terminates.

List Command

- List Command format

list *start [finish]*

Notes: start – one character
finish – one character but optioal

Examples:

- **list**

  Find and list all the entries in the database.

- **list *A***

  Find and list the entries, whose last_name starts with *A*

- **list *A d***

  Find the entries, whose *last_name* starts with any letter between *A* and D including both *A* and D.

Note – finish must be later in the alphabet than start.

List Command (cont'd)

- **For the Client:**

  Sends the command to the server, and displays the response messages from the server.

- **For the Server:**

  When it receives the list command:

  - sends all location entries satisfying the list limits.
  - sends "no such records" if there are no entries satisfying the list request.
  - sends "invalid command" if the list command is in illegal format.
    - example
    - list Aa
    - list Aa B
    - list A Bb
    - list Bb Aa

Outline

- Project 1 Overview
- Unix Network Programming
  - TCP Client
  - TCP Server
- Processing a command
- How to find help and other tips.

Server Database

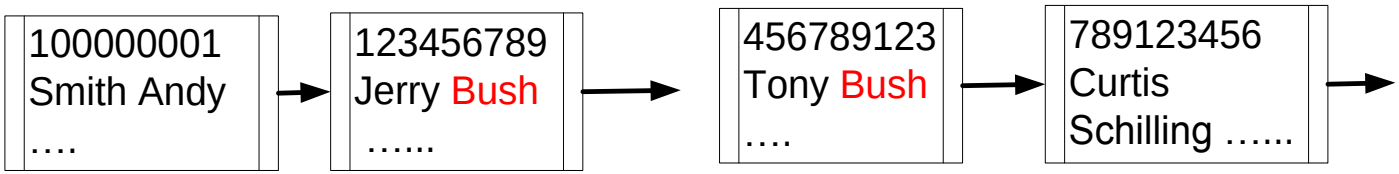
There are many possible data structure choices for implementing the server data base. Two of them are:

- Linked list:
  Easy to add/remove an entry.
- Array:
  The simplest data structure.

Sorting in Database

- The server's database is sorted ascending by *last_name AND in chronological order.*

For example, (based on a linked list)



Ldatabase.txt

last_name	first_name	id,	location
Andy	Smith	100000001	…
Bush	Jerry	123456789	…

String comparison

- The case insensitive string compare functions in Linux.
  - int strcasecmp(const char *s1, const char *s2);
  - int strncasecmp(const char *s1, const char *s2, size_t n);
  - Their usage is similar to strcmp() function.
- An Alternative method.

  Storing the information in upper case letters in server's database. (Smith -> SMITH)

HELP

- Bring printouts to office hours.
- Email questions to Prof.+TAs (cs3516-ta "at" cs.wpi.edu), but do NOT expect immediate results, better to attend office hours.
  - My Office Hours: Mon, 4:00-6:00pm; Fri, 4-6pm
  - Dongqing Xiao's Office Hours: Tue, 2-4pm; Thu, 2-4pm
- We do have a class mailing list that could be used as a last resort.

Questions?

More Tips: file and stdio

- In Linux, a device could be treated as a file.

  For example, the standard input device could be handled as a file.

```
/* fgets() will read a line from the keyboard. */
        fp=stdin;
            fgets(buffer, buffer_len, fp);

/* next  fgets() will read a line from the file named "script.txt". */
        fp=fopen("script.txt", "r");
            fgets(buffer, buffer_len, fp);
```

References

- Beej's Guide to Network Programming
- The GNU C Library
- IBM iSeries Information Center
- The Open Group Base Specifications
- Wikipedia