Project #2: remote working ground (rwg)

In this homework, you are asked to design chat-like systems, called

remote working systems (server only). In this system, users can

meet with/work with/talk to/make friends. Basically, this system

supports all functions, as you did in the first project. In addition,

all clients can see what all on-line users are working.

Command Summary:

\* All commands in project 1.

\* Add more built-in commands (which cannot be used together with pipes), listed as follows.

\* who: show all the online users. E.g.,

<id> <nickname> <IP/port> <indicate me>

1 Iam3 140.113.215.62/1201 <- me

2 (No Name) 140.113.215.63/1013

3 student5 140.113.215.64/1302

\* tell <id> <message>: send only to <user> the message <message>. E.g.,

% tell 5 Hello. # from client #3 (Iam3)

The user "student5" will receive the following.

\*\*\* Iam3 told you \*\*\*: Hello.

\* yell <message>: broadcast the message <message> to all users. E.g.,

% yell Hi...

All users, including me, will receive the following.

\*\*\* student1 yelled \*\*\*: Hi...

\* name <name>: set my name to <name> and broadcast to all users. E.g.,

% name student5

All users, including me, will receive the following.

\*\*\* User from 140.113.215.62/1201 is named 'student5'. \*\*\*

\* Whenever a client comes in, broadcast as follows.

\*\*\* User '(no name)' entered from 140.113.215.63/1013. \*\*\*

Whenever a client leaves, broadcast as follows.

\*\*\* User 'student5' left. \*\*\*

\* A client, say student7, can pipes the standard output/error into a public pipe, say public pipe 3.

% cat test.html >3

All clients see the following message.

\*\*\* student7 (#7) just piped 'cat test.html >3' \*\*\*

If the public pipe exists already, show the following error message.

\*\*\* Error: public pipe 3 already exists. \*\*\*

\* A client, say student4, can receive from a public pipe, say 3, as the standard input.

% cat <3

All clients see the following message.

\*\*\* student4 (#4) just received via 'cat <3' \*\*\*

After closed by both sender/receiver, the public pipe is gone (does not exist).

If public pipe 3 does not exists, show the following error message.

\*\*\* Error: public pipe 3 does not exist yet. \*\*\*

The following is a scenario of using the system.

csh> telnet myserver.nctu.edu.tw 7001 # the server port number

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* Welcome to the information server, myserver.nctu.edu.tw. \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% # All commands in Project 1 must still work!!!

% \*\*\* User '(no name)' entered from 140.113.215.62/1201. \*\*\*

% who # All IDs must be distinct between 1 ~ 30.

1 (no name) 140.113.215.62/1201 <- me

% name 曾佣躬

\*\*\* User from 140.113.215.62/1201 is named '曾佣躬'. \*\*\*

% ls

bin/ test.html test1.txt test2.txt

\*\*\* User '(no name)' entered from 140.113.215.63/1013. \*\*\*

% who

1 曾佣躬 140.113.215.62/1201 <- me

2 (no name) 140.113.215.63/1013

\*\*\* User from 140.113.215.63/1013 is named 'Ｘ神'. \*\*\*

% who

1 曾佣躬 140.113.215.62/1201 <- me

2 Ｘ神 140.113.215.63/1013

\*\*\* User '(no name)' entered from 140.113.215.64/1302. \*\*\*

% who

1 曾佣躬 140.113.215.62/1201 <- me

2 Ｘ神 140.113.215.63/1013

3 (no name) 140.113.215.64/1302

% yell 隨諸到Project #2怎麼做？教教我！

\*\*\* (no name) yelled \*\*\*: 歹勢，我也步知稻。 :-(

\*\*\* Ｘ神 yelled \*\*\*: 我豬盜！混容易的啦！

% tell 2 神阿！我粉認真，請告訴我啦！

\*\*\* Ｘ神 told you \*\*\*: 好阿！仔細看偶如何把檔案送給你地！

\*\*\* Ｘ神 (#2) just piped 'cat test.html >7' \*\*\*

\*\*\* Ｘ神 told you \*\*\*: 阿你可以用'cat <7'把他給秀出來看！

% cat <5 #不小心打錯

\*\*\* Error: public pipe 5 does not exist yet. \*\*\*

% cat <7

<!test.html>

<TITLE>Test<TITLE>

<BODY>This is a <b>test</b> program

for rwg.

</BODY>

\*\*\* 曾佣躬 (#1) just received via 'cat <7' \*\*\*

% tell 2 挖！蒸的耶！好棒喔！

\*\*\* Ｘ神 told you \*\*\*: 阿偶還可以把程式結果送給你喔！

\*\*\* Ｘ神 (#2) just piped 'removetag0 test.html >8' \*\*\*

\*\*\* Ｘ神 told you \*\*\*: 阿你還可以用程式收喔！素素看'number <8'

% number <8

1 Error: illegal tag "!test.html"

2

3 Test

4 This is a test program

5 for ras.

6

\*\*\* 曾佣躬 (#1) just received via 'number <8' \*\*\*

% tell 2 挖！你好神喔！３Ｑ！

\*\*\* Ｘ神 told you \*\*\*: 不客氣！偶本來就是～神～

\*\*\* User 'Ｘ神' left. \*\*\*

% exit

csh>

Now, let us see what happened to the second user.

csh> telnet myserver.nctu.edu.tw 7001 # the server port number

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* Welcome to the information server, myserver.nctu.edu.tw. \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% \*\*\* User '(no name)' entered from 140.113.215.63/1013. \*\*\*

% # All commands in Project 1 must still work!!!

% name Ｘ神

\*\*\* User from 140.113.215.63/1013 is named 'Ｘ神'. \*\*\*

\*\*\* User '(no name)' entered from 140.113.215.64/1302. \*\*\*

% who

1 曾佣躬 140.113.215.62/1201

2 Ｘ神 140.113.215.63/1013 <- me

3 (no name) 140.113.215.64/1302

\*\*\* 曾佣躬 yelled 隨諸到Project #2怎麼做？教教我！

\*\*\* (no name) yelled \*\*\*: 歹勢，我也步知稻。 :-(

% yell 我豬盜！混容易的啦！

\*\*\* 曾佣躬 told you \*\*\*: 神阿！我粉認真，請告訴我啦！

% tell 1 好阿！仔細看偶如何把檔案送給你地！

% cat test.html >7

\*\*\* Ｘ神 (#2) just piped 'cat test.html >7' \*\*\*

% tell 1 阿你可以用'cat <7'把他給秀出來看！

\*\*\* 曾佣躬 (#1) just received via 'cat <7' \*\*\*

\*\*\* 曾佣躬 told you \*\*\*: 挖！蒸的耶！好棒喔！

% tell 1 阿偶還可以把程式結果送給你喔！

% removetag0 test.html >8

\*\*\* Ｘ神 (#2) just piped 'removetag0 test.html >8' \*\*\*

% tell 1 阿你還可以用程式收喔！素素看'number <8'

\*\*\* 曾佣躬 (#1) just received via 'number <8' \*\*\*

\*\*\* 曾佣躬 told you \*\*\*: 挖！你好神喔！３Ｑ！

% tell 1 不客氣！偶本來就是～神～

% exit

csh>

Requirements:

\* Write two different server programs.

(1) Use the single-process concurrent paradigm.

(2) Use the concurrent connection-oriented paradigm with shared memory.

\* For each client, all behaviors required by project 1 are still required

in this project.

Hints:

\* You can assume that the # of public pipes is <= 100.

\* ">3" or "<4" has no space between them. So, you can distinct them

from "> filename" easily.

\* For the second program (2),

\* One chat buffer has at most 10 unread messages, each of which has

at most 1024 bytes.

\* For each public channel, need to support a lock on it to prevent from others to access it.

\* For each pipe for ">3", use FIFO instead of pipe.

\* If a message has more than 1024 bytes, simply truncate it to 1024 bytes.

\* If you use BBS-like method, you can use signal "SIGUSR1" or "SIGUSR2" to help.

\* For "who", the master process maintain an id for each forked process.

Since there are no more than 30 processes, id <= 30 and let the id be allocated

in the way similar to FD table.

If you find some commands confusing or not workable, please let us know.

Due date: 11/22 (Sunday)