Project2: RWG System

NP TA 王瑞渝

4/13 23:59

Project 2 Deadline

Demo: 4/14 Mon.

Project 2

In this project, you are asked to design 2 kinds of servers:

- np_simple (Single user)
  - **Project 1**
  - **Concurrent connection-oriented**

- np_single_proc (Multiple users)
  - **Project 1** + **User pipe** + **4 functions** + **Broadcast message**
  - **Single-process concurrent**

Project 2: Info

- You are **HIGHLY** encouraged to publish your questions on Project 2 討論區
  - Check the spec and other questions first.
- You can contact TAs by e3. (Mails sent to other addresses will **NOT** be replied)
- TA hours (Tuesday: 15:00 ~ 17:00) on **4/8** will be held online.
  - You **MUST** make a reservation by email in advance.
- TAs will **NOT** debug for you.

Project 2: Demo

- You are **NOT** allow to demo if we are unable to compile your project with a single make command.
- 4/14 Mon. 17:30 ~ 21:20.
- We will announce demo slots 1 ~ 3 days before.
- Tasks:
  - Correct format and compile.
  - QA.
  - Pass testcases.
  - Implement extra function with limit time.

Scenario

Server 1

```
bash$ telnet nplinux1.cs.nctu.edu.tw 7001
% ls | cat
bin test.html
% ls |1
% cat
bin test.html
% exit
bash$
```

Server 2

- Chat-like system
- Provide all functions in **project 1**
- New functions
  - **Login/Logout message**
  - **who – get information of all users**
  - **name – rename yourself**
  - **tell – send message to someone**
  - **yell – broadcast message**
  - **User pipe**

Server 2: scenario example

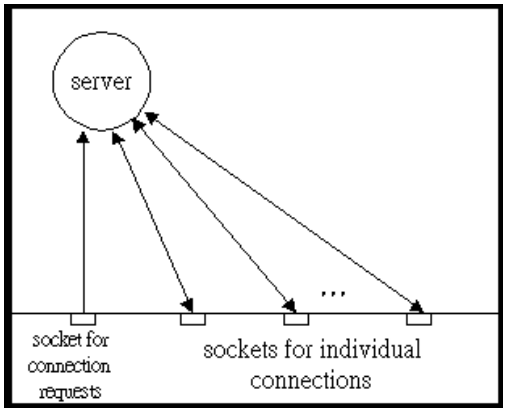
- Text only ver.
  - https://hackmd.io/@q12we34rt5/rkM8WMCeC

- Video demonstration
  - https://youtu.be/o_d1kY-j_BM

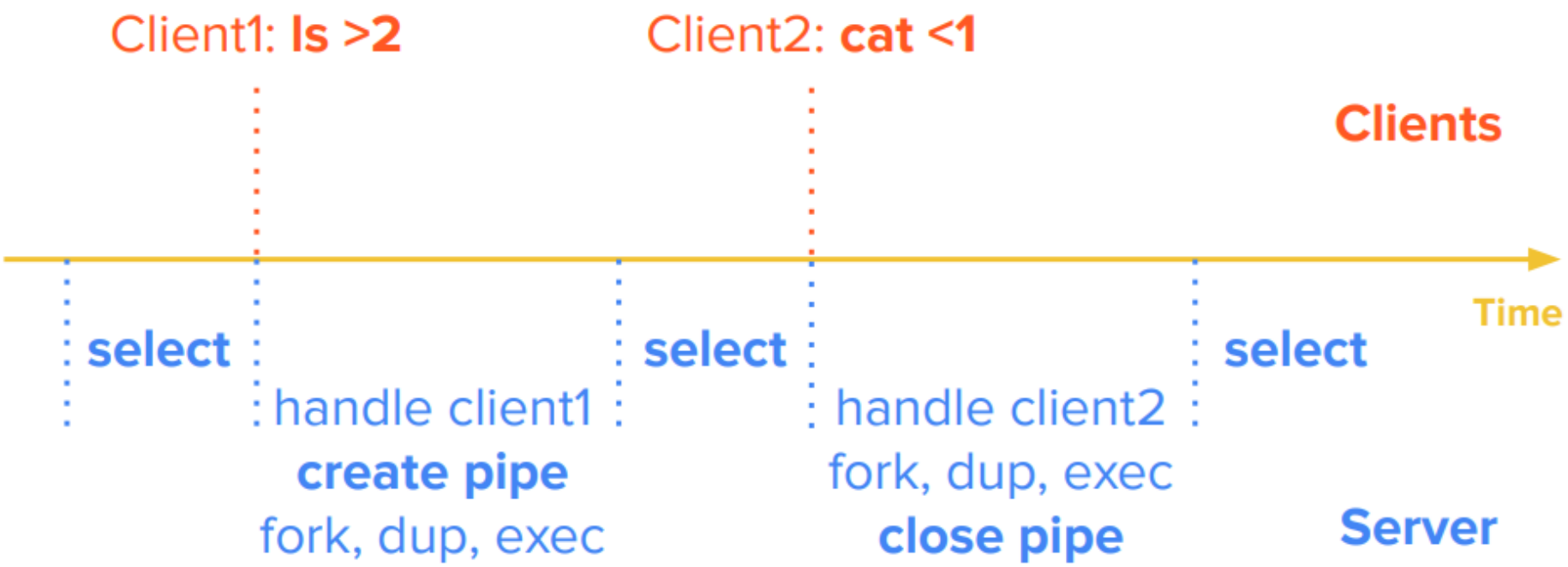
Implementation

Server 2 (np_single_proc)

- **Single-process concurrent** (use **select**)
- Use **pipe** to implement user pipe
  - **DO NOT** use FIFO or temporary files
- Use socket to send messages directly
- Maintain environment variables for every user



Server 2 (np_single_proc) - User Pipe



User Pipe Detail

- Pipe **stdout** only
- Whole command line should be printed in broadcast message

[terminal of user1]
% cat test.html | removetag0 >2
*** user1 (#1) just piped 'cat test.html | removetag0 >2' to user2 (#2) ***
Error: illegal tag "!test.html" // error message from removetag0
%
[terminal of user2]
% cat <1
*** user2 (#2) just received from user1 (#1) by 'cat test.html | removetag0 >2' ***
Test ...
%

User Pipe - Error Handling

- When user pipe error, each command should still be executed
  - Some command prints something itself
  - Prevent stuck when pipe large file

```
% cat test.html | removetag0 >999
*** Error: user #999 does not exist yet. ***
Error: illegal tag “!test.html”              // error message from removetag0
% Pikachu <999                               // Pikachu prints input message and Pika!Pika!
*** Error: user #999 does not exist yet. ***
Pika!Pika!
% cat LargeFile | cat | cat >999
*** Error: user #999 does not exist yet. ***
```

User Pipe - Error Handling

% cat <2 | Pikachu

pipe(#2->#1)

% cat LargeFile >2

pipe(#1->#2)

% cat <999 | Pikachu // user pipe error

% cat LargeFile >999 // user pipe error

User Pipe - Error Handling

- Redirect stdin/stdout to **/dev/null**
  - stdin: EOF
  - stdout: dump everything

% cat <999 | Pikachu // user pipe error

% cat LargeFile >999 // user pipe error

Issues

Handle Function Failures !!

- **Fork** may failed (Project 1)
- **Create pipe** may failed (Project 1)
- **Select** may failed
- **Read** may failed

Select May Failed

```
if (select(maxfd + 1, &read_set, NULL, NULL, NULL) < 0) {
    // may be interrupted by signal or other errors
    // handle error
}
for (fd = 0; fd < maxfd; ++fd) {
    if (FD_ISSET(fd, &read_set)) {
        // handle fd
    }
}
```

Read May Failed

```
if (read(cli_fd, buf, BUF_SIZE) < 0) {
    // may be interrupted by signal or other errors
    // handle error
}
```

Remember to set the flag

You should set flag **SO_REUSEADDR** in server socket.
Hint: use function setsocketopt

Q&A