System Programming
hw2

江建德
連彥傑
侯廷璋
林家瑞

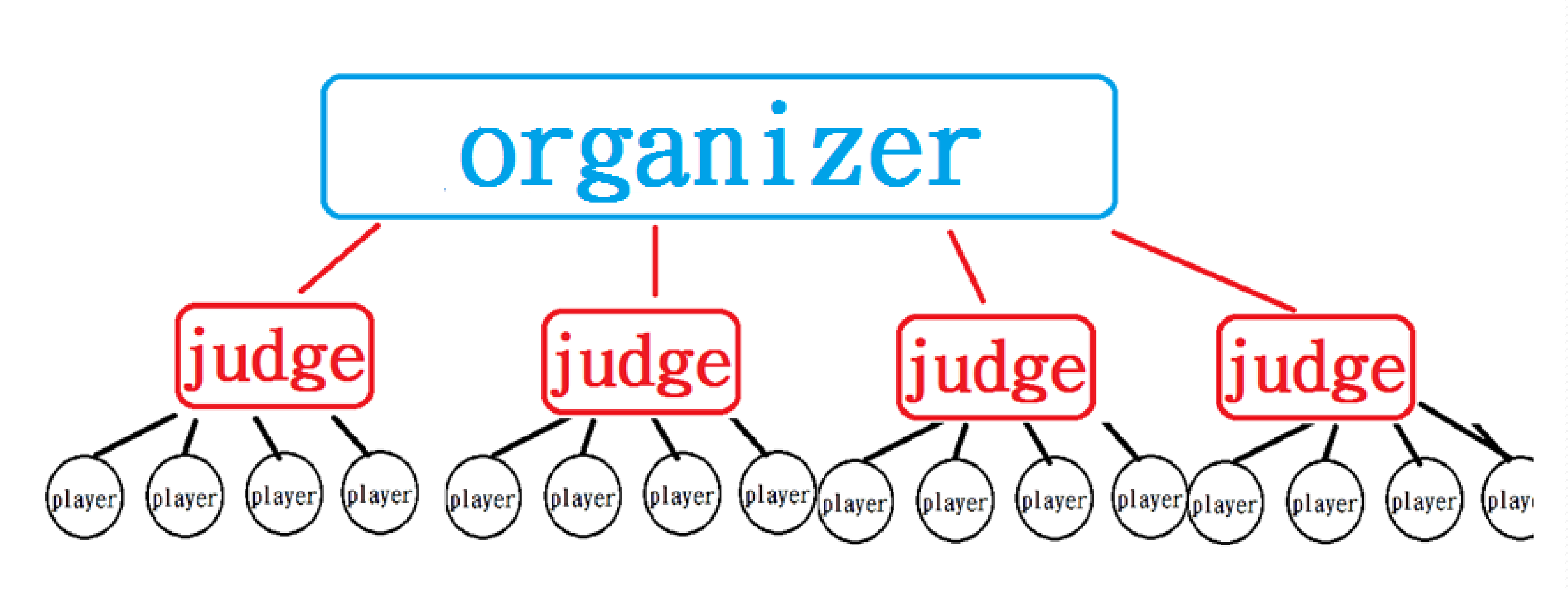
Introduction

- In this assignment, you are going to practice communicating among processes.
- The goal of this assignment is to practice multiple-processes control via ***pipe*** and ***FIFO*** , and understand the use of ***fork()*** and **exec**().

A Game

- It’s a Old maid(抽鬼牌) contest.
- We want to find the joker king.
- There are a organizer to hold this contest.
- Several judges in this contest.
- Many players join in this contest.
- 52 poker + 1 joker

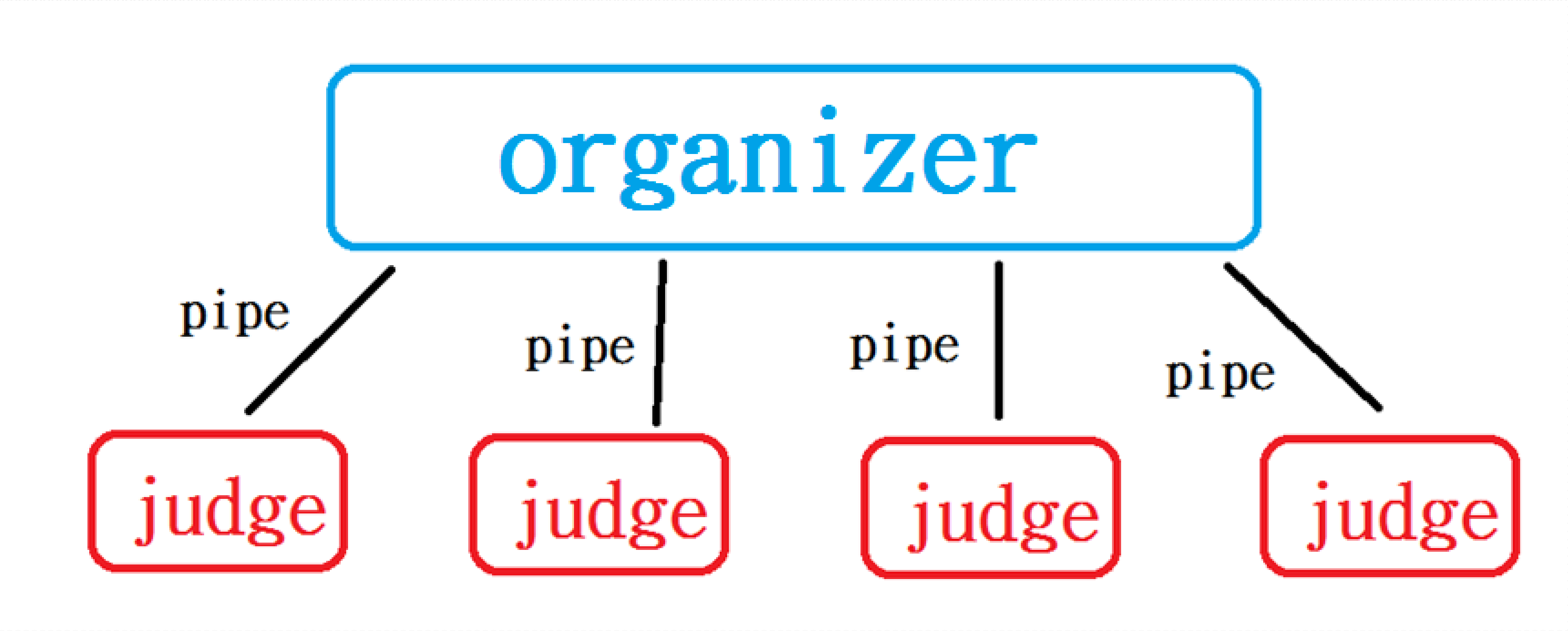
A Game


organizer
judge
judge
judge
judge
player
player
player
player
player
player
player
player
player
player
player
player
player
player
player
play

Task

- organizer.c
  - **$./organizer [judge_num] [player_num]**
  - Organizer should fork and execute the number of judges specified by the argument (**judge_num**)
  - The organizer must build pipes to communicate with each of them before executing them.
  - There will be C(player_num, 4) competitions needed to be assigned.
  - Loser get **-1** , and others get **0**
  - outputs all players' ID sorted by their scores, from the lowest to the highest, separated by spaces.

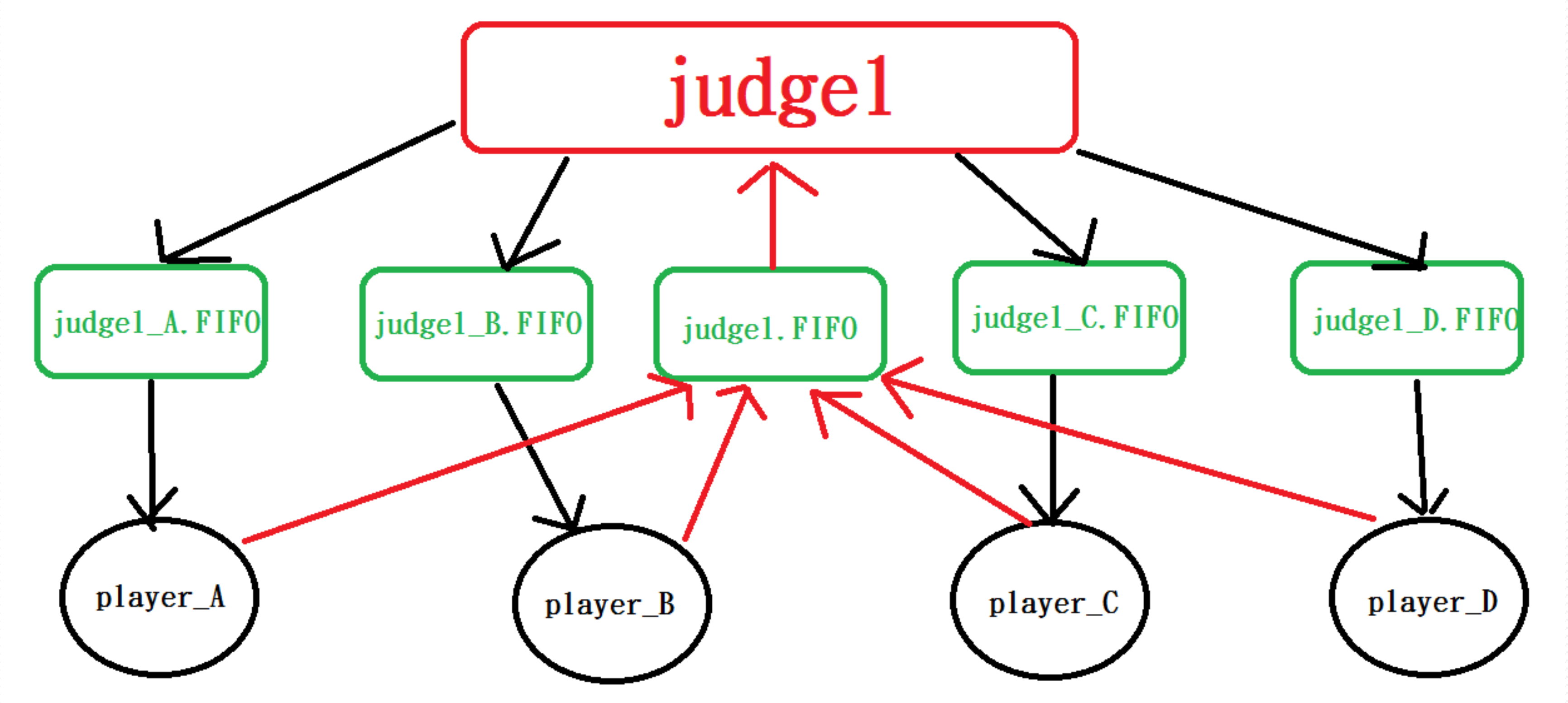
Task



Task

- judge.c
  - **$./judge [judge_id]**
  - **judge.c** should fork and execute 4 players.
  - The judge should create a FIFO named judge[judge_id].FIFO, such as judge1.FIFO, to read responses from the players
  - create four FIFOs named judge[judge_id]_A.FIFO, judge[judge_id]_B.FIFO,judge[judge_id]_C.FIFO, judge[judge_id]_D.FIFO, to write messages to the players in the competition held by this judge.

Task


judge1
judge1_A.FIFO
judge1_B.FIFO
judge1.FIFO
judge1_C.FIFO
judge1_D.FIFO
player_A
player_B
player_C
player_D

Task

- player.c
  - **$./player [judge_id] [player_index] [random_key]**
  - **judge_id** is the judge of this competition.
  - **player_index** would be a character in {'A', 'B', 'C', 'D'}.
  - **random_key** would be an integer in [0, 65536). It is used to verify if a response really comes from that player.

Game flow (judge.c <-> player.c)

- the judge send 14 cards to player A, 13 cards to player B, C, D in the following format:

  [card_1] [card_2] [card_3] … [card_14] (if it sends to player A)

  [card_1] [card_2] [card_3] … [card_13] (if it sends to player B, C, D)

- And then, players should send the number of cards to judge through judge[judge_id].FIFO in the following format:

  [player_index] [random_key] [number_of_cards]

Game flow (judge.c <-> player.c)

- the judge send the number of cards of player B to player A in the following format:

  [type] [number_of_cards]

  The value of type” is “<” to indicate that it is the player A’s turn to draw.

- the player A should send the card ID that it wants to draw to judge in the following format:

  [player_index] [random_key] [card_ID]

- The judge send the card ID had been drawn (by random choosed) to player B in the following format:

  [type] [card_ID]

  The value of type” is “>” to indicate that the card_ID of player B have been drawn.

- player B send the card of the card ID to judge in the following format:

  [player_index] [random_key] [the_card_number]

Game flow(judge.c <-> player.c)

- The judge send the card of the card ID to player A in the following format:

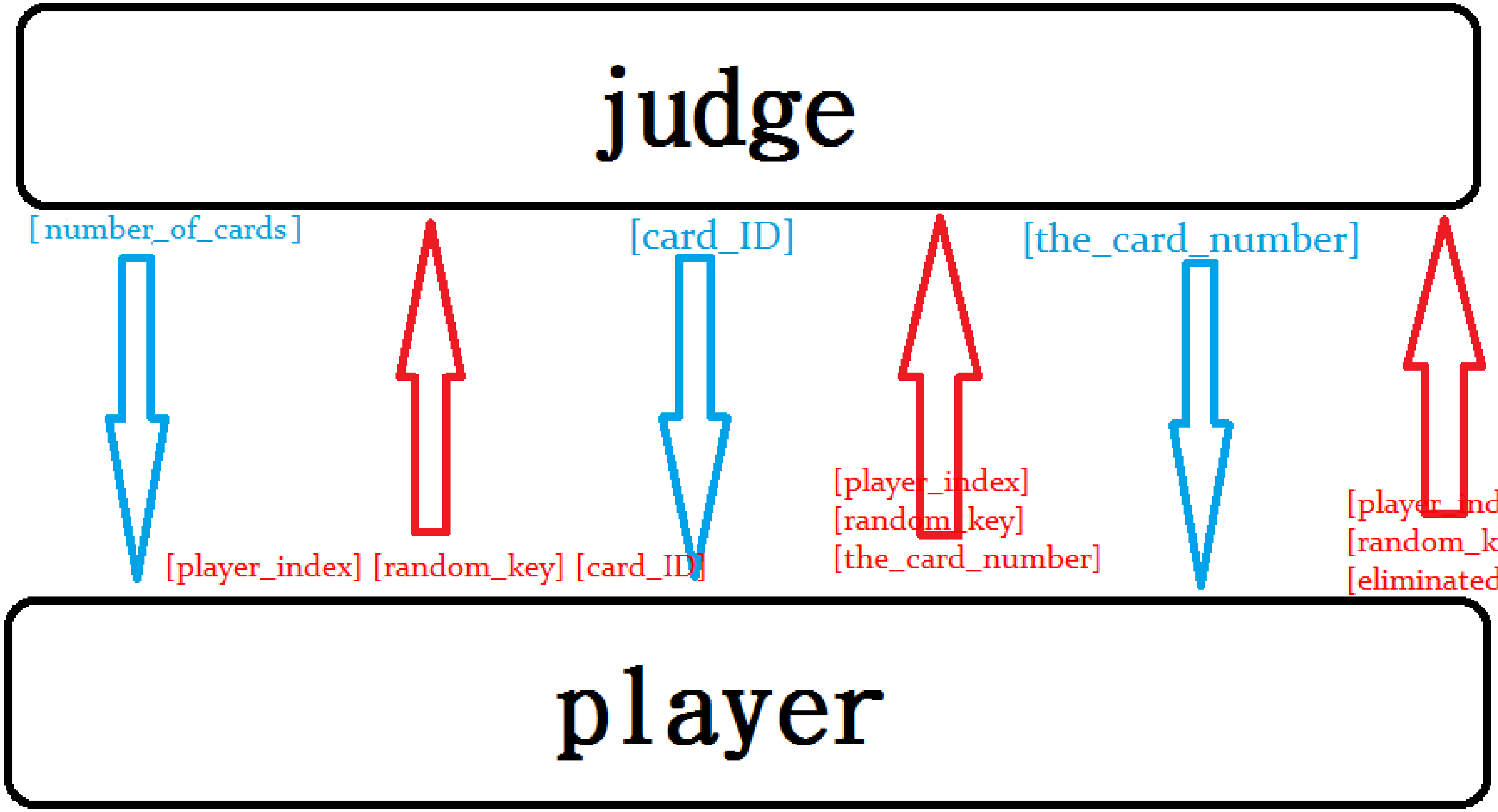
  [the_card_number]

- player A told judge whether he/she eliminated the two card with the same number, that is, whether the number of his/her cards decreased by one in the following format:

  [player_index] [random_key] [eliminated]

  - if **eliminated** is 0, indicated player A doesn’t have two cards with the same number
  - if **eliminated** is 1, indicated player A have two cards with the same number

Game flow


judge
[number_of_cards]
[card_ID]
[the_card_number]
[player_index] [random_key] [card_ID]
[player_index]
[random_key]
[the_card_number]
[player_index]
[random_key]
[eliminated]
player

Execution format

- **$ ./organizer 1 4**
  - This will run 1 judge and 4 players. The organizer will fork and execute:
- **$ ./judge 1**
  - The judge will create:

```
judge1.FIFO
judge1_A.FIFO
judge1_B.FIFO
judge1_C.FIFO
judge1_D.FIFO
```

Execution format

- The organizer sends judge 1 (judge 1 reads from standard input):
  - **1 2 3 4**
- The judge executes:
  - **$ ./player 1 A 9**
  - **$ ./player 1 B 2014**
  - **$ ./player 1 C 10000**
  - **$ ./player 1 D 65535**

Execution format

- The judge 1 sends 14 or 13 cards to every players through judge1_{A, B, C, D}.FIFO
  - **0 3 5 6 6 7 7 8 9 11 12 12 12** (ex: sends player B through judge1_B.FIFO)

- each player send the number of cards to judge through judge1.FIFO
  - **A 9 5**
  - **B 2014 7**
  - **C 10000 10**
  - **D 65535 8**

Execution format

- In round 1, player A's turn:
  - judge 1 sends player A through judge1_A.FIFO:
    - **< 7** (player B has 7 cards)
  - player A sends judge 1 through judge1.FIFO:
    - **A 9 6** (player A wants the 6th card of player B)
  - judge 1 sends player B through judge1_B.FIFO:
    - **> 4** (the 4th card of yours is picked)
  - player B sends judge 1 through judge1.FIFO:
    - **B 2014 11** (the card is J)
  - judge 1 sends player A through judge1_A.FIFO:
    - **11** (player A gets J)
  - player A sends judge 1 through judge1.FIFO:
    - **A 9 0** (player A doesn't have the same cards)

Execution format

- All competitions are over.
  - The organizer sends judge 1:
    - **0 0 0 0**
    - The judge 1 terminates.
- The organizer outputs the result:
  - **2 1 3 4** (player with ID 2 got most negative score)

Scoring (7 points)

- Make to generate three execution file. (1 point)
- organizer.c correctness (1 point)
- judge.c correctness (2 points)
- player.c correctness (1 point)
- Processes handling (1 point)
- Player's cheating detection (1 point)

Submission

- Submit **SPHW2_[student id].tar.gz** (only .tar.gz) to Ceiba

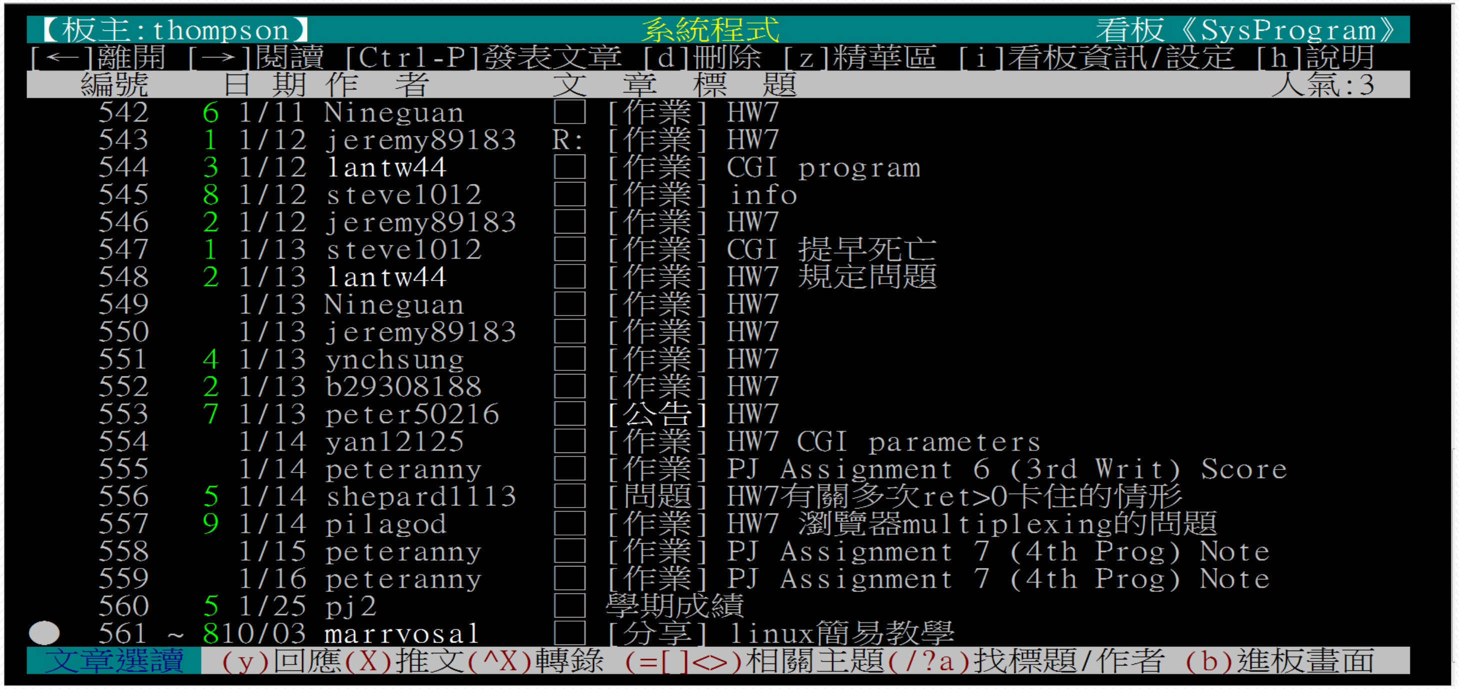
  (ex : SPHW2_b02902000.tar.gz)
- The file should include three files.
  - organizer.c
  - judge.c
  - player.c
  - Makefile
  - README.txt
    - please briefly state how to compile your program or something you have to explain

Punishment

- Plagiarism punishment
- Late punishment
  - 5% for each day delay
- File error punishment
  - Submission file format is .tar.gz
  - Include FIVE files

Ask Question

- If you have any question on homework, post a article on SysProgram@PTT2. TAs will answer your question on SysProgram.
- Send email to spfall2014@gmail.com



```
【板主:thompson】                     系統程式                      看板《SysProgram》
[←]離開 [→]閱讀 [Ctrl-P]發表文章 [d]刪除 [z]精華區 [i]看板資訊/設定 [h]說明
   編號       日 期 作   者          文  章  標  題                              人氣:3
    542    6 1/11 Nineguan          □ [作業] HW7
    543    1 1/12 jeremy89183       R: [作業] HW7
    544    3 1/12 lantw44           □ [作業] CGI program
    545    8 1/12 steve1012         □ [作業] info
    546    2 1/12 jeremy89183       □ [作業] HW7
    547    1 1/13 steve1012         □ [作業] CGI 提早死亡
    548    2 1/13 lantw44           □ [作業] HW7 規定問題
    549      1/13 Nineguan          □ [作業] HW7
    550      1/13 jeremy89183       □ [作業] HW7
    551    4 1/13 ynchsung          □ [作業] HW7
    552    2 1/13 b29308188         □ [作業] HW7
    553    7 1/13 peter50216        □ [公告] HW7
    554      1/14 yan12125          □ [作業] HW7 CGI parameters
    555      1/14 peteranny         □ [作業] PJ Assignment 6 (3rd Writ) Score
    556    5 1/14 shepard1113       □ [問題] HW7有關多次ret>0卡住的情形
    557    9 1/14 pilagod           □ [作業] HW7 瀏覽器multiplexing的問題
    558      1/15 peteranny         □ [作業] PJ Assignment 7 (4th Prog) Note
    559      1/16 peteranny         □ [作業] PJ Assignment 7 (4th Prog) Note
    560    5 1/25 pj2               □ 學期成績
●   561 ~ 810/03 marrvosal         □ [分享] linux簡易教學
 文章選讀  (y)回應(X)推文(^X)轉錄 (=[]<>)相關主題(/?a)找標題/作者 (b)進板畫面
```

TA hour

- Tue 10:00 – 12:00 連彥傑 @R302
- Wed 14:00 – 16:00 江建德 @R302
- Fri 10:00 – 12:00 侯廷璋 @R302
- Fri 12:00 – 14:00 林家瑞 @R302