

程式設計專題

作品名稱：Cian Jhen Online Judge
前鎮高中程式解題系統

作者：201 04 李佳樺

指導老師：蔣寶鑫老師

```
問題 1 輸出 偵錯主控台 終端機 篩選 (例如 text、lexclude)
=thread-group-added,id="i1"
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Warning: Debuggee TargetArchitecture not detected, assuming x86_64.
=cmd-param-changed,param="pagination",value="off"
=cmd-param-changed,param="args",value="/C c:\\\\project\\\\b.exe & pause"
[New Thread 8452.0x2b98]
[New Thread 8452.0x24e0]
[New Thread 8452.0x22c8]
[New Thread 8452.0x73c]
[New Thread 8452.0x295c]
The program 'C:\Windows\system32\cmd.exe' has exited with code -1 (0xffffffff).
```

(gdb) Launch (project) 第 7 行, 第 2 欄 空格:4 UTF-8 CRLF C++ Win32

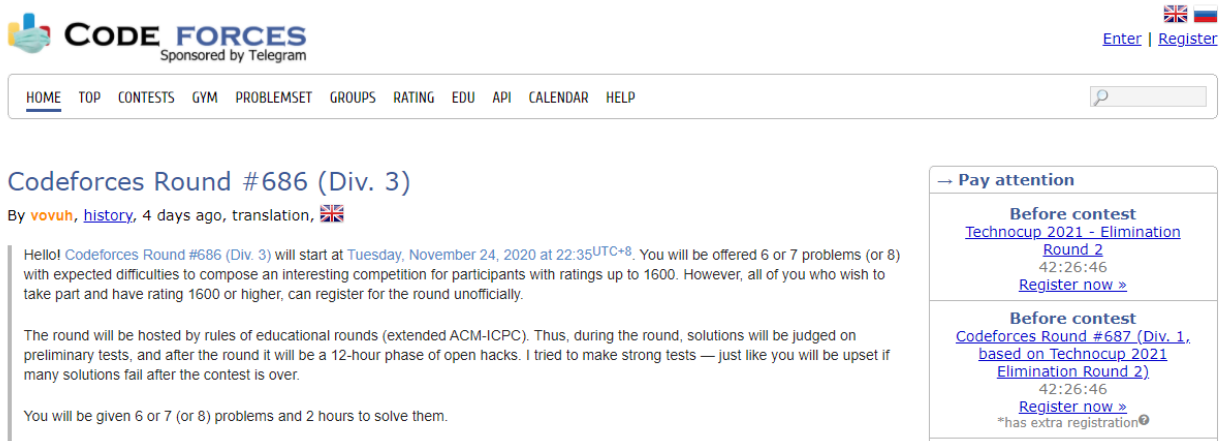
目錄

題目：Cian Jhen Online Judge

一、前言	2
二、創作動機	3
三、介紹 Online Judge	3
四、開源 OJ 系統選擇	5
五、研究 QDUOJ 系統架構	5
六、安裝系統部屬所需指令	7
七、Docker 部屬 OJ 系統	9
八、升級 OJ 系統	10
九、QDUOJ 修改 root 密碼以及操作與設定	11
十、進行前端二次開發	15
十一、統整操作方式與部屬方式	24
十二、結論	25
十三、參考資料及其他	26

一、前言

我是一名前鎮高中的學生，目前高二，是校隊的資訊競賽選手，也有接觸了一些競賽，在網路上的解題系統刷了一些題目，解題系統就是可以在上面舉行程式解題競賽，即時回饋以及記分板都能讓你模擬真實的 ACM 競賽情況，除了模擬競賽以外，也可以透過解題系統來練習題目，不斷的練習題目不只可以提升自身的邏輯思考，更能節省比賽的耗時！對於我來說，在打 Codeforces(國外知名 Online Judge)的競賽時，通常看到第一題(最簡單的 pA)都可以在不到一分鐘的情況想出解法，正是因為我平時有足夠的練習，所以第一題不用五分鐘就可以解決，但是你如果缺乏練習的話，你可能會在理解題目花上不少時間，理解完之後還必須想解法，想完後就要用程式碼來將你的解題想法呈現。但這樣時間夠嗎？很顯然是不夠的！所以每個程式競賽選手都有一定的刷題量，就是為了在競賽時節省時間，以及讓腦內充滿解題思維。



The screenshot shows the Codeforces website interface. At the top, there's a navigation bar with links like HOME, TOP, CONTESTS, GYM, PROBLEMSET, GROUPS, RATING, EDU, API, CALENDAR, and HELP. Below this, the main content area displays information for "Codeforces Round #686 (Div. 3)". It mentions the contest is by "vovuh" and was held 4 days ago. The text describes the contest rules, including the number of problems (6 or 7) and the time limit (2 hours). On the right side, there's a sidebar with a "Pay attention" section, which lists upcoming contests like "Technocup 2021 - Elimination Round 2" and "Codeforces Round #687 (Div. 1, based on Technocup 2021 Elimination Round 2)".

圖 1 Codeforces 首頁示意圖

二、創作動機

在前鎮高中，會寫程式的人並不多，對程式設計有興趣的同學也是極少數，所以我希望透過架設 Online Judge 的方式來提升大家對程式設計的熱愛，也可以讓想要透過個人申請 APCS 組來升學的同學提升自身能力，使其考取 APCS 更加順利！而且透過 Online Judge 的 Contest System，我也可以帶著大家體驗程式競賽的快感，享受腦力激盪的滋味。

三、介紹 Online Judge

在網路上，已經有好幾個 Online Judge(以下簡稱 OJ)的成品，所以我決定挑出幾個比較常見且親民的 OJ 來介紹給大家，並統整出他們各自的優缺點，分析完優缺點後再選擇出自己所要選用的開源系統來進行二次開發。

(一) UVa Online Judge(UVa/ACM)

最古老，最有名的 OJ，是西班牙 Valladolid 大學的 OJ，裡面有許多經典的題目，但缺點就是太過老舊，介面以及操作方式很不直覺。



(二) Temporary Infor Online Judge(TIOJ)

TIOJ 是建國中學的 OJ，题目的難度偏高，但也收錄了很多台灣資訊奧林匹亞的考古題，還有開放建中校內培訓講義供大家參考，介面還算不錯，很簡單明瞭，但缺點就是題目難度太高，不適合新手。



圖 4 TIOJ 首頁示意圖

(三) 高中生程式解題系統(ZeroJudge)

高雄師範大學附屬高中的 OJ，題目數量龐大，是目前國內最多人使用的 OJ，但是也被高手們戲稱水題庫，因為題目偏易，但在基礎題庫當中也能看到一些難題，這對於不會判斷題目難易度的新手來說非常不友好，但優點是容易架設，使用 VM 虛擬機一鍵部署就可以完成了。



圖 5 zerojudge 首頁示意圖

(四) Peking University Online Judge(PKU/POJ)

北京大學的 OJ，是中國程式競賽選手最常使用的平台，題目的類型都是演算法競賽，難度一樣偏高，對新手來說母湯。



圖 6 POJ 首頁示意圖

(五) Qingdao University Online Judge(QDUOJ)

青島大學的 OJ，上面的題目幾乎都是他們舉辦比賽所留下來的考古題，難度偏高，也是一樣不適合新手，但是他的介面是目前介紹的這幾個內最好看的一個，而且標籤系統也可以明確的分類題目類型，題目難易度也會呈現出來，新手不需要擔心自己陷入了惡魔題。

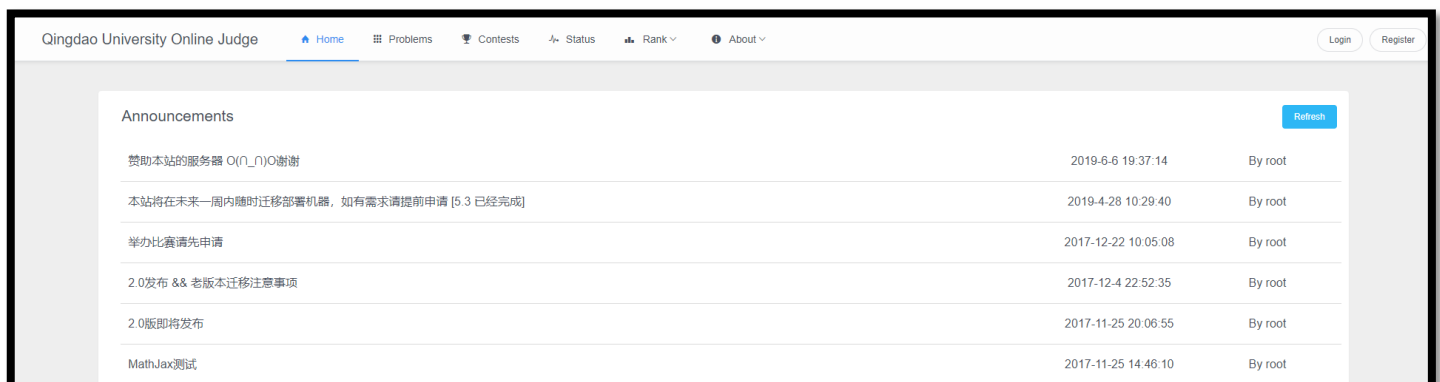


圖 7 QDUOJ 首頁示意圖

四、開源 OJ 系統選擇

介紹了這麼多個 OJ，最後我決定使用 QDUOJ 來進行部屬以及二次開發，選擇這套 OJ 的原因是因為他的介面討喜，而且新增題目也很輕鬆，不需要額外撰寫.json 檔。難易度分界明顯，也讓新手不再誤入地雷區。標籤系統可以輕鬆的進行題目分類。Docker 部屬開源也算蠻輕鬆的，不會消耗太多心力以及時間。

五、研究 QDUOJ 系統架構

從 github 上下載 Deploy 並部屬完成後，我們可以直接點開 docker-compose.yml 來看內部程式碼，會發現他分成四個部分：Services:Redis、Postgres、Judge-Sever、oj-backend。

```
oj-backend:
  image: registry.cn-hangzhou.aliyuncs.com/onlinejudge/oj_backend
  container_name: oj-backend
  restart: always
  depends_on:
    - oj-redis
    - oj-postgres
    - judge-server
  volumes:
    - ./data/backend:/data
  environment:
    - POSTGRES_DB=onlinejudge
    - POSTGRES_USER=onlinejudge
    - POSTGRES_PASSWORD=onlinejudge
    - JUDGE_SERVER_TOKEN=CHANGE_THIS
    # - FORCE_HTTPS=1
    # - STATIC_CDN_HOST=leejiahua.myftp.org
  ports:
    - "0.0.0.0:80:8000"
    - "0.0.0.0:443:1443"
```

圖 8 oj-backend 程式碼區塊

```
judge-server:
  image: registry.cn-hangzhou.aliyuncs.com/onlinejudge/judge_server
  container_name: judge-server
  restart: always
  read_only: true
  cap_drop:
    - SETPCAP
    - MKNOD
    - NET_BIND_SERVICE
    - SYS_CHROOT
    - SETFCAP
    - FSETID
  tmpfs:
    - /tmp
  volumes:
    - ./data/backend/test_case:/test_case:ro
    - ./data/judge_server/log:/log
    - ./data/judge_server/run:/judger
  environment:
    - SERVICE_URL=http://judge-server:8080
    - BACKEND_URL=http://oj-backend:8000/api/judge_server_heartbeat/
    - TOKEN=CHANGE_THIS
    # - judger_debug=1
```

圖 9 Judge-Sever 程式碼區塊

```
oj-postgres:
  image: postgres:10-alpine
  container_name: oj-postgres
  restart: always
  volumes:
    - ./data/postgres
  environment:
    - POSTGRES_DB=onlinejudge
    - POSTGRES_USER=onlinejudge
    - POSTGRES_PASSWORD=onlinejudge
```

圖 10 Postgres 程式碼區塊

```
oj-redis:
  image: redis:4.0-alpine
  container_name: oj-redis
  restart: always
  volumes:
    - ./data/redis:/data
```

圖 11 Services:Redis 程式碼

這樣子也有利於後續的二次開發，比如說如果要進行前端修改的話就必須要把 oj-backend 的 volumes 增加一行前端的文件路徑，之後再 docker-compose up -d 就可以了。

那系統的架構呢？Github 的文件上有寫一共分成四個模組：

1. Backend(Django): <https://github.com/QingdaoU/OnlineJudge>
2. Frontend(Vue): <https://github.com/QingdaoU/OnlineJudgeFE>
3. Judger Sandbox(seccomp): <https://github.com/QingdaoU/Judger>
4. JudgeSever(A wrapper for judger): <https://github.com/QingdaoU/JudgeServer>

整理到這裡，我們大概已經知道了 QDUOJ 的流程了，大概是：

1. 進入前端(Vue)
2. 送出程式碼，呼叫後端的 API(Python)
3. 後端 API 在呼叫 Judge Sever API(Go)
4. Judge Sever API 呼叫 Judge 執行指令(C,execeve+seccomp 執行)

那什麼是 seccomp 呢？seccomp 是 OJ 最核心的一部分，你想看，如果有人寫了重開機的程式碼怎麼辦？寫了一個無窮迴圈？seccomp 就是用來避免這些問題，QDUOJ 的 seccomp 是用 C 寫的，fork 一個新的 process，設定 rule 後用 execeve 來執行指令。

六、安裝系統部屬所需指令

先介紹一下我們這次所使用的硬體設備以及系統環境！

硬體設備為：

硬體名稱	規格型號	數量	備註
CPU	AMD R7 3700x	1	3.6Ghz 8C16T
RAM	Kingston DDR4 3200 FURY 8G	2	共 16G
SSD	Micron MX500 250G	1	
GPU	AMD RX5700XT	1	

系統為： Windows 10 專業版 64 位元

由於這次架設並沒有要放置到網路上，所以才會採用 Windows 系統，如果要放到網路上提供使用的話建議使用 Linux 系統，這次的操作只是要進行二次開發，部屬等動作，日後才有打算申請虛擬機、網域，並將 OJ 發放到網路上供大家使用。

那因為使用的是 Windows 系統，所以部屬會比較麻煩一點，需要安裝許多的指令，要安裝的有：pip、git、docker-compose、python。那先從 pip 跟 python 開始，只需要到 Python 的官網，選擇下載 Python 3.7，下載完成之後打開 cmd，試著輸入 python

看看是否有成功切換到 python 的編譯器，接著就是到網路上下載 get-pip.py，然後透過 cmd 安裝。

```
python get-pip.py
```

安裝成功就會看到 Successfully installed 的訊息囉！

```
Collecting pip
  Downloading pip-20.1-py2.py3-none-any.whl (1.5 MB)
    |#####| 1.5 MB 819 kB/s
Collecting wheel
  Downloading wheel-0.34.2-py2.py3-none-any.whl (26 kB)
Installing collected packages: pip, wheel
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
  Successfully installed pip-20.1 wheel-0.34.2
```

圖 12 pip 安裝訊息
之後在 cmd 輸入 `pip --version` 確認是否安裝成功。

```
pip --version
```

成功的話，會顯示出 pip 的位置路徑和 Python 版本。

```
C:\Users\LJH54>pip --version
pip 20.2.4 from C:\Users\LJH54\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\site-packages\pip (python 3.7)
```

圖 13 pip 路徑以及 Python 版本
在安裝完 pip 與 python 後，接著就是要來安裝 git，安裝方

式就是到 git 的官網，選擇完版本後下載，之後安裝就一直無腦的點擊下一步就好了，之後打開 cmd 輸入 `git --version` 驗證

是否安裝成功，以及版本是否正確。

```
C:\Users\LJH54>git --version
git version 2.29.2.windows.2
```

圖 14 git 版本

Git 安裝完成後就只剩下 docker-compose 了！docker-

compose 指令安裝方式非常簡單，只需要到官網下載 docker 並

安裝就完成了！安裝完 docker 後就可以打開 cmd 輸入 docker-

compose version 驗證是否安裝成功。

```
C:\Users\LJH54>docker-compose version
docker-compose version 1.27.4, build 40524192
docker-py version: 4.3.1
CPython version: 3.7.4
OpenSSL version: OpenSSL 1.1.1c 28 May 2019
```

圖 15 docker-compose、docker-py、CPython、OpenSSL 版本

七、Docker 部屬 OJ 系統

在安裝完所有部屬所需的指令之後，就可以來進行系統部屬啦！

第一步，就是先用系統管理員身分執行 powershell，並選擇一個空間充裕的地方進行安裝步驟。

第二步，輸入以下指令：

```
git clone -b 2.0 https://github.com/QingdaoU/OnlineJudgeDeploy.git ; cd OnlineJudgeDeploy
```

注意！原本文檔裡的指令是 `git clone.....Deploy && cd Online...`，之所以把 `&&` 更換成 `;` 是因為目前的 powershell 版本無法使用 `&&` 作為分隔符號，但替換成 `;` 就可以解決問題了。

第三步，啟用服務(輸入以下指令)：

```
docker-compose up -d
```

等待一段時間後，就會顯示部屬完成了！但打開 docker 會發現有個問題，就是 `oj-postgres` 一直啟用失敗，我們可以點擊旁邊的 logs 查看紀錄，發現是 `volumes` 路徑出了問題，一直找不到目標檔案，所以我們利用編譯器 visual studio code 來開啟 `docker-compose.yml` 並修改 `volumes` 路徑，修改成：

```
volumes:  
  - ./data/postgres
```

就完成了，因為她原本的設置是

```
./data/postgres:/var/lib/postgresql/data
```

但是我們到\OnlineJudgeDeploy\data\postgres 就能發現，他缺少了那些文件，所以把後綴的那些路徑去掉就可以正常運行了。

在完成這些步驟後，就可以打開瀏覽器，輸入 127.0.0.1，就可以看到 OnlineJudge 的首頁了！到這裡就已經部屬完成了，之後就是進行二次開發的動作！

八、升級 OJ 系統

我們點開 Deploy 的 README.md，可以發現下面有一個他們自己的文檔，點進去看就可以發現裡面有一個“代碼升級說明”，由於我們部屬的是最一開始釋出的版本，而如今已經有經過許多的更新改版，所以我們需要使用代碼來升級系統，那升級的方式就是用系統管理員身分執行 Powershell，然後移動到 OnlineJudgeDeploy 的資料夾，之後輸入以下的指令：

```
git pull  
docker-compose pull ; docker-compose up -d
```

我們一樣把文檔裡原本的指令分隔號&&換成；，原因在前面的部屬系統時有提到，這裡就不多做解釋了。

九、QDUOJ 修改密碼以及 操作與設定

在升級完 OJ 之後，我們要做的第一步就是到 admin 修改 root 的密碼。打開瀏覽器，輸入 127.0.0.1/admin，就會進入到 admin 的登入畫面，帳號輸入 root，密碼輸入 rootroot，就可以登入進去了，登進去之後點左邊的 General，之後點右邊 root 的 Edit，就可以修改 root 的資料了，在 new password 輸入你要的新密碼後就更改完成了！

那如果要導入帳號的話，可以選擇使用 Excel 檔來一次大量導入，如果只是要新增一個的話可以到一開始的首頁，然後點右上角的 Register，進行註冊的動作。

註冊完帳號之後，接著就是要來新增題目，那新增題目的方式就是到 admin，點左邊的 Problem List，之後點擊 Create，就可以進行新增題目的動作啦~那題目敘述跟範例測資都打完了之後，就是要來上傳題目測資了，點擊 Test Case 下方的 Choose File，選擇你的測資檔案上傳，這樣就完成了。那測資要怎麼做呢？是單純的 txt 檔嗎？並不是，測資分別為.in 和.out 檔，一樣是由文字編輯器所編輯，那第一組測資的檔名就要命名為 1.in、1.out，第二組為 2.in、2.out，

以此類推，測資都處理好以後必須要把他們壓縮起來，變成一個.zip 檔，我們可以用樹的方式來呈現一下資料格式：

```
→ testcase pwd
/tmp/testcase
→ testcase tree
.
├── 1.in
└── 1.out

0 directories, 2 files
```

圖 16 資料格式圖

然後把它壓縮到一個 zip 中：

```
→ testcase zip testcase.zip ./{*in,*out}
adding: 1.in (stored 0%)
adding: 1.out (stored 0%)
```

圖 17 壓縮測資到 zip

這樣題目就新增完成啦！

我也試著新增了四道題目，用於測試待會的 Contest System。

Problem List			
#	Title	Level	Total
1	pA 欸幹 🤔 穿山甲欸 🤔 🤔	Mid	0
2	pB 欸幹 🤔 國小數學欸 🤔 🤔	Low	0
3	pC 欸幹 🤔 裝子賢欸 🤔 🤔	Low	0
4	pD 欸幹 🤔 大合照欸 🤔 🤔	Low	1

圖 18 題目列表

pA 欸幹 😱 穿山甲欸 😱 😱

Description

在寶鑫國裡，穿山甲是一個很稀有的生物，東佑是一位生物學家，他希望能透過基因改造來解決此問題。

在改造完成後寶鑫國內只剩下了一對小穿山甲(一雄一雌)，但這些被改造後的穿山甲不會死亡，一對穿山甲每月能生一對小穿山甲(一雄一雌)，每對小穿山甲在出生後的第三個月裡，又能開始生一對小穿山甲。

東佑改造完穿山甲後，很想知道 n 個月後一共有幾對穿山甲，所以請書恆幫忙計算，但是書恆太懶了，他想請你幫忙寫一個程式來計算，但是因為計算出來的數字太大，書恆可能記不住，所以他希望你可以把答案 $\text{mod } 1000000007$ 後再傳給他

Input

每行有一整數 n ， $-2 < n < 2e31$

輸入可能有多行，每一行有一個整數是一筆測資，最後一行以-1 代表結束，不需要處理該筆測資。

Output

請輸出 n 個月後共有幾對穿山甲，並把結果 $\text{mod } 1000000007$

圖 19 第一題的題目敘述

這題就單純在考費式數列和矩陣快速冪，敘述中的穿山甲其實就是所謂的月兔，但這都不是重點！重點是我們要測試他的 Judge System 是否能正常使用。所以我們可以嘗試送出 AC 程式碼來確認他的評測系統。

ID	Status	Problem	Time	Memory
2c8049711a28	Accepted	1	0ms	3MB

圖 20 評分狀態

可以發現，他的評測系統沒有問題，而且點擊 ID 就可以看到你送出的程式碼，完整地紀錄下來。

在簡單的進行完新增題目的操作以及測試 JudgeSystem 之後，就可以來操作 Contest System 啦~

那新增競賽的方式其實很簡單，比新增題目還簡單，你只需要到 admin，點左邊的 Create Contest，就可以新增競賽了！是不是非常的方便！

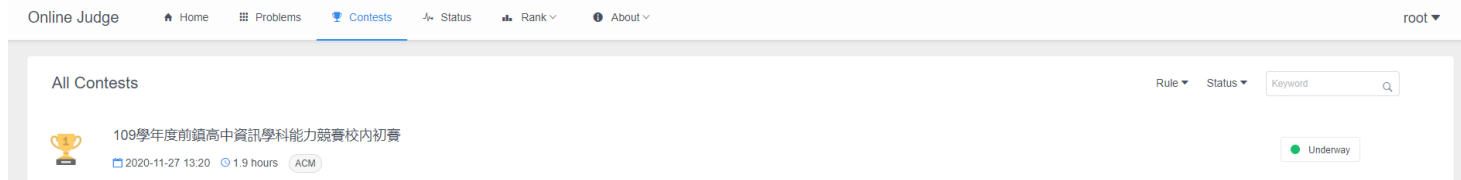


圖 21 競賽列表

新增完之後，就可以到 Contest List 新增比賽的題目，點擊右邊 Operation 的 Problem，就可以進入到 Contest Problem List 了，之後點擊下面的 Create 或 Add From Public Problem 就可以進行新增題目到競賽當中的動作了，那新增完題目之後，就可以來進行比賽了！為了測試他的 Rankings，我新增了兩隻帳號，而權限都是設定為 Regular User，如果權限為 admin 或是 Super admin 的話就無法參與競賽排名，可能是因為怕影響比賽公平性。那新增完之後就可以解題來測試排名系統了。

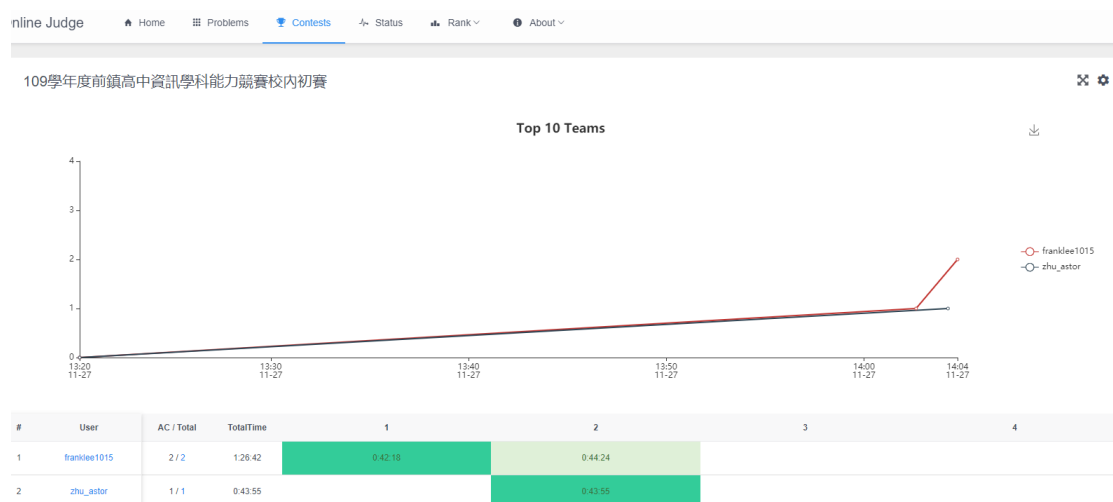


圖 22 競賽記分板

那沒問題之後我們可以來進一步研究排名系統，那綠色就代表 AC，深綠代表首殺，紅色代表錯誤，可能是 CE、RE、TLE、MLE 等等的狀況，反正不是 AC 的就是紅色。

十、進行前端二次開發

那在我們分析完整個初始系統後，就要來進行 OJ 外觀的改造了。

QDUOJ 的前端模組：<https://github.com/QingdaoU/OnlineJudgeFE>

在前面研究 QDUOJ 的系統架構時，我們也有提到這個 OJ 的前端是用 Vue 來寫的，那 Vue 是什麼呢？Vue 是一套用於建構用戶介面的漸進式框架，與其他大型框架不同的是，Vue 被設計為可以自底向上逐層應用。Vue 的核心庫只關注視圖層，不僅易上手，還便於與第三方庫整合。

二次開發內容：

1. 修改介面語言(從英文改為繁體中文)
2. Admin management 左上角 logo 修改
3. FAQ 修改
4. Favicon 修改

在二次開發前端之前，我們先來了解一下前端模組結構：

build	項目建構(webpack)相關代碼
config	配置目錄，包括端口號等。
node_modules	npm 加載的項目依賴區塊。
src	<p>主要開發區塊，基本上要修改的東西都在這裡。</p> <p>裡面包含了幾個目錄以及文件：</p> <ul style="list-style-type: none">● assets：放了一些圖片，如 logo 等。● components：目錄裡放了一個組件文件，可以不用。● App.vue：項目入口文件，我們也可以直接將組件寫在這裡，而不使用 components 目錄。● main.js：項目的核心文件
static	靜態資源目錄，如圖片、字體等。
index.html	首頁入口文件，你可以添加一些 meta 信息或統計代碼之類的。
package.json	項目配置文件。
README.md	項目的說明檔，markdown 格式。

了解完前端的結構以及我們要二次修改的內容之後，我們就可以來開始進行修改的動作了！那我們前面有提到，QDUOJ 的前端是使用 Vue 所撰寫的，所以想要修改前端文件的話我們就必須先在電腦安裝 Vue 的開發環境。

那要安裝那些東西呢？首先就是設定環境變數啦！我們先到 Node.js 的官網來安裝 Node.js，下載網址：<https://nodejs.org/en/> 點選左邊穩定版本來進行下載，那下載完並且安裝完成之後就可以設定環境變數了，設定完成之後可以打開 cmd，輸入指令確認版本。

```
# print Node.js version
node -v

# print NPM version
npm -v
```

如果成功的話就會顯示出目前 npm 和 node 的版本了。

```
C:\Users\LJH54>npm -v
6.14.8

C:\Users\LJH54>node -v
v14.15.1
```

圖 23 npm 與 node 版本

在安裝完 Node.js 之後，我們就可以開始安裝 Vue 了，首先打開 cmd，輸入以下指令來進行安裝：

```
npm install -g vue
```

或者你也可以用 cnpm 來進行安裝：

```
cnpm install -g vue
```

cnpm 是什麼呢？由於 npm 的伺服器位於國外，所以進行安裝插件時易受網路環境影響，所以淘寶就在中國弄了個伺服器，為了改善這種情況。

該如何下載 cnpm？輸入以下指令：

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

看到此圖就代表 Vue 安裝完成了：



圖 24 Vue 安裝成功

然後就是安裝 Vue CLI，打開 cmd，輸入以下指令：

```
npm install -g vue-cli
```

安裝成功後會顯示下圖：

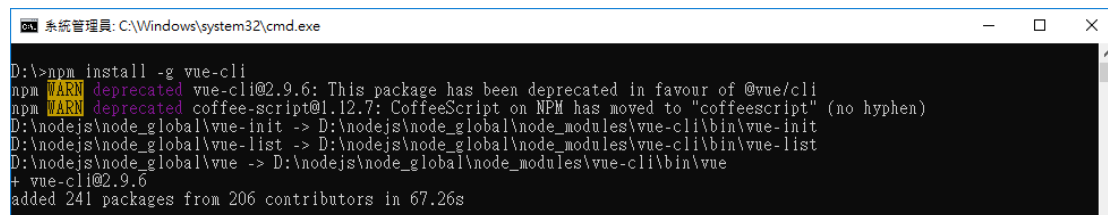


圖 25 Vue CLI 安裝成功

這些都安裝成功後，就可以開始進行前端模組修改了！
我們先把前端模組下載下來，在空間充裕的地方輸入指令：

```
git clone https://github.com/QingdaoU/OnlineJudgeFE.git
```

下載完成之後我們就可以對照著模組結構來進行修改，
由於我們這次只是進行細微的調整，所以會稍微輕鬆一點。

我們第一個要做的就是修改介面語言，從預設的英文改為繁體中文，做法就是找到以下檔案並修改其內容：

```
\OnlineJudgeFE\src\i18n\index.js
```

使用 Visual Studio Code 打開並修改：

```
29 // load language packages
30 export default new VueI18n({
31   locale: 'zh-TW',
32   messages: messages
33 })
```

圖 26 介面語言設定程式碼區塊

把第 31 行的 en-US 語言改為 zh-TW，如上圖所示。

之後別急著打包，我們還有剩下三個地方要修改。

第二個要修改的就是 Admin management 左上角的 logo，由於他的檔案是 svg 檔，所以無法使用 png 等常見格式了，那圖檔選擇完畢之後就是要到以下路徑進行替換：

```
\OnlineJudgeFE\src\assets
```

到了這個資料夾後就可以找到一個名為 logo.svg 的檔案了，我們把事先準備好的圖案替換上去，就完成啦～

第三個要修改的就是 FAQ 內容，FAQ 就是常見問題，我們可以自行添增或刪減其內容，只需要找到以下檔案修改：

```
\OnlineJudgeFE\src\pages\oj\views\help\FAQ.vue
```

新增內容如下：

```
28 </p>
29 </li>
30 <li>{{t('開發者是誰?')}}
31 <p>{{t('是一名前鎮高中的學生，目前高二，熱愛程式設計，是數理班的學生也是許多數理資訊學科競賽的校隊選手。')}}
32 </p>
```

圖 27 FAQ 新增內容

最後一個修改內容就是更換 Favicon，這個就簡單多了，完全不需要修改到前端的模組內容，只需要找到下面的檔案：

```
\OnlineJudgeDeploy\data\backend\public\website\favicon.ico
```

然後我們再準備一個要替換的 ico 檔就行了，並使用圖片編輯軟體將大小調整為 128*128 的大小就行了，如下圖：



圖 28 欲更換之圖示

把要修改的都修改完了之後就可以開始進行打包的動作了，那要如何打包呢？前面有提到說我們的前端是用 vue 來寫的，所以打包就會用到 vue,node,npm 等指令，還好我們事先已經安裝完畢了，我們只需要打開 PowerShell(系統管理員身分)，到前端模組的資料夾內，輸入指令：

```
npm install
set NODE_ENV=development
npm run build
set TARGET=http://Your-backend
npm run dev
```

輸入完這五行的指令之後就已經打包完成啦～你可能會想說第 4 行的指令是不是怪怪的？真的要輸入 Your-backend？沒錯就是 Your-backend，不用懷疑。

那打包完成之後我們就會看到一個叫做 `dist` 的資料夾，裡面的結構分布如同下圖所示：

```
→ OnlineJudgeFE git:(master) X tree dist
dist
├── admin
│   └── index.html
├── index.html
└── static
    ├── css
    │   ├── admin.127f3da5b09451926728de2829ebb32e.css
    │   ├── loader.css
    │   ├── oj.0ba722f43ddb758cde2f9dc804455e.css
    │   └── vendor.f033d6c4c74b6b40e92ca86f168fd0be.css
    ├── fonts
    │   ├── KaTeX_AMS-Regular.3d8245d.woff2
    │   └── KaTeX_AMS-Regular.ac1d46d.woff
    ....
    ....
```

圖 29 `dist` 內結構樹狀圖

那打包完成之後，我們就要把 `dist` 資料夾拉到原本開源的資料夾內(`OnlineJudgeDeploy`)，我這裡是選擇放到 `data` 資料夾內，路徑如下：

```
\OnlineJudgeDeploy\data\dist
```

放完之後就完事了？並沒有，我們需要到 `docker-compose.yml` 檔案內新增前端檔案路徑。

```
54 volumes:
55   - ./data/backend:/data
56   - ./data/dist:/app/dist
```

圖 30 `oj-backend` 中的 `volumes` 程式碼區塊

在 `oj-backend` 的 `volumes` 底下新增實際路徑。

修改完成之後，打開 PowerShell，移動到 Online JudgeDeploy 資料夾內，然後輸入以下指令進行部屬：

```
docker-compose up -d
```

輸入完成之後就可以打開 127.0.0.1 來確認修改內容。



圖 31 OnlineJudge 首頁示意圖

如果未登入的時候也是繁中介面的話就代表修改成功了！

Admin management 左上角 logo 確認：

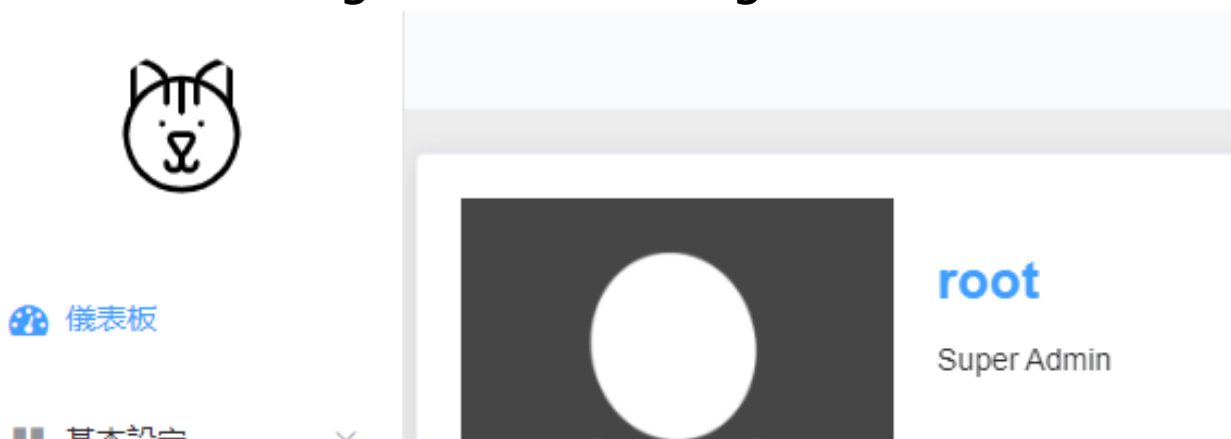


圖 32 admin 頁面示意圖

左上角的 logo 也修改完畢了！

FAQ 修改確認：

常見問題

- 輸入與輸出在哪邊？

你的程式會從 `stdin` ('標準輸入') 讀取輸入，並且將結果輸出到 `stdout` ('標準輸出') 例如，你可以在 C 中使用 `scanf` 和 `printf` 來讀取和輸出。如果你使用 `std::cin` 和 `std::cout` 來讀取和輸出，則會導致 `Runtime Error`。

- 提交執行時間是什麼意思？

線上解題系統可能會使用不同的輸入檔案來多次測試你的程式碼。如果你的程式碼在每個輸入檔案的時間限制內給

- 我要如何使用 C++ `long long`？

你應該宣告為 `long long` 並且與 `cin/cout` 或 `%lld` 一起使用，使用 `__int64` 將導致 `Compile Error`。

- Java 規範？

所有的程式都必須以 `Main` 類別的靜態 `main` 方法開始執行。不要使用 `public` 類別：即使 `Main` 也必須是非 `public` 類

- 關於輸出格式錯誤？

此 OJ 中沒有輸出格式錯誤。Judge 系統將自動整理輸出然後包裹在輸出的最後一行中。如果仍然與正確的輸出不同

- 如何回報關於此 OJ 的錯誤？

此線上解題系統是開源的，你可以到 [Github](#) 提交問題。請提供有關錯誤的詳細訊息(如 `env`，版本...)，這將極大地幫助

- 開發者是誰？

是一名前鎮高中的學生，目前高二，熱愛程式設計，是數理班的學生也是許多數理資訊學科競賽的校隊選手。

圖 33 FAQ 內容

可以看到最後一行的內容就是我前面程式碼新增的內容，FAQ 也很順利地修改成功！

Favicon 修改確認：



圖 34 OnlineJudge Favicon 圖示

十一、統整操作方式與部屬方式

如果以後學校或者是個人想架設 OJ 的話，使用 qduoj 來進行部屬是一件不錯的事，Windows 的部屬方式統整如下：

1. 安裝所需指令：git、docker-compose、python、pip、node、npm、vue。
2. 使用 docker 部屬開源。
3. git clone (前端模組)。
4. 修改前端模組，進行二次開發。
5. npm run build 打包前端模組。
6. 修改 docker-compose.yml 檔案內容，新增前端模組文件路徑。
7. 重新部屬，docker-compose up -d。
8. 打開瀏覽器，在網址列輸入 127.0.0.1 或 localhost。
9. 盡情享用 OJ 系統~

十二、結論

由於時間不多，所以我在二次開發的方面沒有下太大的功夫，主要是因為整個製作時間不到五天，如果有充裕的時間，我一定會新增更多內容，以及申請虛擬機及網域，使 OJ 發部到網路上，供大家使用，架設 OJ 的動機很單純，想服務大家，想為前鎮的資訊教育盡一份心力，我們學校的校隊選拔是採用筆試的方式，我希望之後這套 OJ 有機會能派上用場，讓校內賽也能用上機考的方式去進行，也能讓校隊的選手去練習以及切磋交流。

使用 windows 系統來進行部屬的動作的確會有許多的問題，網路上也沒有人能替你解答，你只能靠著程式的報錯來進行 debug，過程也極其血尿，但是架設完了你就會覺得，這一切都值得。

但我還是建議使用 Linux 系統啦！尤其是以後要放到網路上的，在安裝過程就會方便很多了。

十三、參考資料及其他

參考資料：

1. 前端二次開發：

https://docs.onlinejudge.me/#/onlinejudge/guide/update_fe

https://blog.csdn.net/qq_45792080/article/details/105968762

2. Vue、node.js 環境安裝：

<https://dotblogs.com.tw/skyfu160/2019/09/04/165504>

<https://dotblogs.com.tw/skyfu160/2019/09/05/162843>

QDUOJ 網址：<https://qduoj.com/>

Favicon 圖源：



由前鎮高中 2 年 11 班 黃馨儀 同學所畫

真的很謝謝他><