GRADUATION DEFENSE TEMPLATE 2019

同济OJ 在线评测网站

网站设计汇报总结

计算机科学导论第二组

源码GitHub地址：https://github.com/Shanxce/Tongji-oj

CONTENT

GRADUATION DEFENSE TEMPLATE

Part 01

项目动机与设计思路

项目动机与设计思路

原同济OJ

缺乏动效，界面单调

缺少色彩和美术

同学需要对题目进行讨论

缺少相关版面

重写OJ

设计网站美术

设计大致框架

增加网站界面动效

增加错误提示信息

讨论版（论坛）

问题新增讨论

个人题问

个人回答解决问题

部分代码思路分享

论坛发帖全员共同参与讨论

OJ板块：正常OJ的功能基础上建立新的排版，在主站内以个人用户为中心的页面，建立个人题库与个人提问与讨论问题列表，评测机为核心技术，防止用户恶意提交进程卡评测机。

论坛版块：显示讨论的问题，并且可以对问题进行及时回复，方便同学随时查阅记录，更好地完成所有任务。

GRADUATION DEFENSE TEMPLATE

Part 02

技术路线和功能

前端功能

界面设计

界面设计包含众多同济元素，在美化界面的同时，给使用者更温馨的感觉

框架的使用

部分位置使用了bootstrap的框架模板，让前端界面看起来更加整洁，适配性更好

导航栏

导航栏功能齐全，无论是在OJ界面还是论坛界面，都能轻松到达想要去的界面

json模板的使用

使用了一些json的功能，比如canvas、看板娘等，让界面不再单调，同学们在做题之余也能有一些休闲

动态的下拉框

在论坛主界面，为了优化感官，仅展示了问题的标题，点击后才会下拉显示其他同学的回复

后端技术路线

Flask_login

Flask_mail

sqlite3

blueprint

login | verify | database | reconstruct

登录

登录的后端使用了flask_login，将前端表单输入的内容与数据库中存储的信息进行比对，如果比对到正确的信息，则将用户信息在User类中记录，供随后的前后端进行访问

注册验证码

为了保证注册的安全性，我们通过flask_mail模块设置了邮箱验证码，可向用户提供的邮箱中发送验证码，并将用户填入的验证码与发送的验证码进行比对，当比对正确时用户才能进行注册，将注册信息写进数据库

数据库

为了更好的完成OJ功能，我们使用了sqlite3模块来建立数据库，建立了用户信息、OJ题目、论坛提问、论坛回答等6张表，更好地对网站信息进行管理

代码重构

在完成OJ功能后，我们使用bluprint对代码进行了重构，将完成同一功能的代码放在一起进行管理，使项目的可维护性提高、代码的可读性提高

GRADUATION DEFENSE TEMPLATE

Part 03
关键技术点

关键技术

评测机关键技术

OJ网站的灵魂在评测机，既要精确检测出使用了多少内存和时间以检查是否满足时空限制，又要防止恶意代码破坏服务器——这涉及到安全问题。此外为了与数据库交互还需要将评测机封装为python模块以方便调用。

CPython封装

CPython库是一个可以将C语言编译成为Python模块的库。通过限定输入输出格式，注册函数等操作将评测机的接口封装成Python模块。之后便可以像调用普通Python库一样使用评测机。

资源限制和检测

在Linux系统中提供了一系列API以监测和限定资源。通过父进程fork子进程并使用setrlimit对子进程的时间、空间、堆栈等资源进行限制，当子进程超出限制就会被杀死，然后用wait4就可以获取子进程的CPU占用等信息以核查是否满足时空要求。

安全保护

用Linux系统提供的代码注入技术，监视被评测程序与内核交互部分，例如系统调用，内核向进程发出的信号，调用文件等等信息，如果调用了可能危害系统的函数、或访问了不该访问的文件，父进程会向子进程发出SIGKILL信号，子进程就会被杀死。

调试遇到的问题

在调试过程中，我们发现了一些具有学习意义的bug，这些bug往往是由于我们知识的缺失或者对知识没有深入理解所致，或者没有养成正确的工程习惯而花费了大量时间在务必要的调试过程中

setrlimit失效

我们从《UNIX系统编程》中得知setrlimit可以限制进程的各种资源，但是当程序写完之后我们发现程序并没有在预期超时之后被杀死。查了两天的错才发现，setrlimit设置了各种数值，却对系统不起作用。因为我们使用了微软提供的Ubuntu，其setrlimit部分有bug，因而官网注明了这一点。这让我们意识到，即使是大公司生产的工具也是可能有bug，我们不该盲目假设所用工具是无问题的。

database被占用

在最终代码整合调试过程中，我们发现正常操作会引起数据库被占用，然而我们并没有多线程同时修改数据库，最后在仔细审查代码后发现，有部分操作使用完数据库没有关闭，另一个人就无法修改数据库了。因而数据库应该在每次修改之后都要commit并关闭数据库。

接口不统一

在代码整合过程中往往会出现前后端接口不一致，导致无法整合的问题，这一问题源于我们小组沟通缺少，导致各自按照自己的想法写程序并潜意识假定对方也是这样规定的。例如数据库命名，路径约定等等。这些属于没有养成正确的工程习惯所带来的无必要的调试，应该要避免。

GRADUATION DEFENSE TEMPLATE 2019

THANKS for watching

源码GitHub地址：https://github.com/Shanxce/Tongji-oj