

文章编号:1007-2985(2010)06-0037-03

Online Judge 评判内核的设计与实现*

何迎生, 罗 强

(吉首大学数学与计算机科学学院, 湖南 吉首 416000)

摘 要:针对目前基于 B/S 结构的 Online Judge 在线程序评判系统, 提出了一种评判内核的实现方法, 诠释了 OJ 系统如何通过评判内核处理用户提交的解题代码和实时获得评判结果的一系列过程。

关键词:Online Judge; 评判; 解题代码; 解题程序; 编译; 子进程

中图分类号:TP311

文献标志码:A

Online Judge 在线程序评判系统简称 OJ, 是一种基于 B/S 结构的多用户在线评判系统^[1], 允许用户在线提交解题代码, 系统自动进行编译运行并给出评判结果, 再根据用户的解题数目和用时进行排名. 评判内核是实现 OJ 的技术难点和关键, 它集成了获取解题代码, 获取测试数据、编译、试运行、操作限制、结果统计等一系列动作, 并且会因所支持语言的外部编译环境不同而使评判内核存在跨平台问题。

1 评判内核的设计原理

OJ 系统的体系结构如图 1 所示. 在该体系中, 评判内核处于业务层与持久层之间, 实现对多种编译器(如 GCC, JDK, Free Pascal 等)的控制. 评判内核首先接收来自业务层提交的评判任务, 将任务放入评判队列, 然后取队首的任务进行评判, 评判结束后将评判结果送持久层, 同时将本次评判的结果反馈给业务层. 其具体评判过程如下:

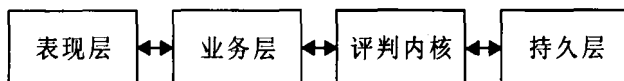


图 1 OJ 系统层次结构

后将评判结果送持久层, 同时将本次评判的结果反馈给业务层. 其具体评判过程如下:

(i) 根据任务编号获取解题代码并创建临时源代码文件。

(ii) 调用对应语言的编译器对源代码文件进行编译, 编译通过则继续(iii), 否则视为编译错误(Compilation Error), 保存编译错误信息并跳转到(vi)。

(iii) 试运行(ii)编译好的程序, 限制程序执行非法操作; 启动计时器并调入标准输入数据, 等待获取输出数据; 若运行异常(Runtime Error)则保存异常信息并跳转到(vi); 若内存消耗超出限制则跳转到(vi); 若时间消耗超出计时器限制则跳转到(vi); 否则将输出数据与调入的标准输出进行对比. 通常有 3 种对比结果, 即去掉前后空白和换行时完全匹配(Presentation Error), 去掉前后空白和换行时不匹配(Wrong Answer), 完全匹配(Accepted)。

(iv) 如果对比结果为完全匹配并且还有其他测试数据, 跳转到(iii)。

(v) 记录最大内存消耗、最大时间消耗, 更新用户解题数目、提交数目、排名等信息。

(vi) 保存并返回结果。

* 收稿日期: 2010-09-26

基金项目: 大学生研究性学习和创新性实验计划项目(湘教通[2009]320号)

作者简介: 何迎生(1974-), 男, 湖南浏阳人, 吉首大学数学与计算机科学学院副教授, 硕士, 主要从事面向对象程序设计研究; 罗强(1990-), 男, 湖南衡阳人, 吉首大学数学与计算机科学学院本科生, 主要从事面向对象程序设计研究。

评判流程如图 2 所示.

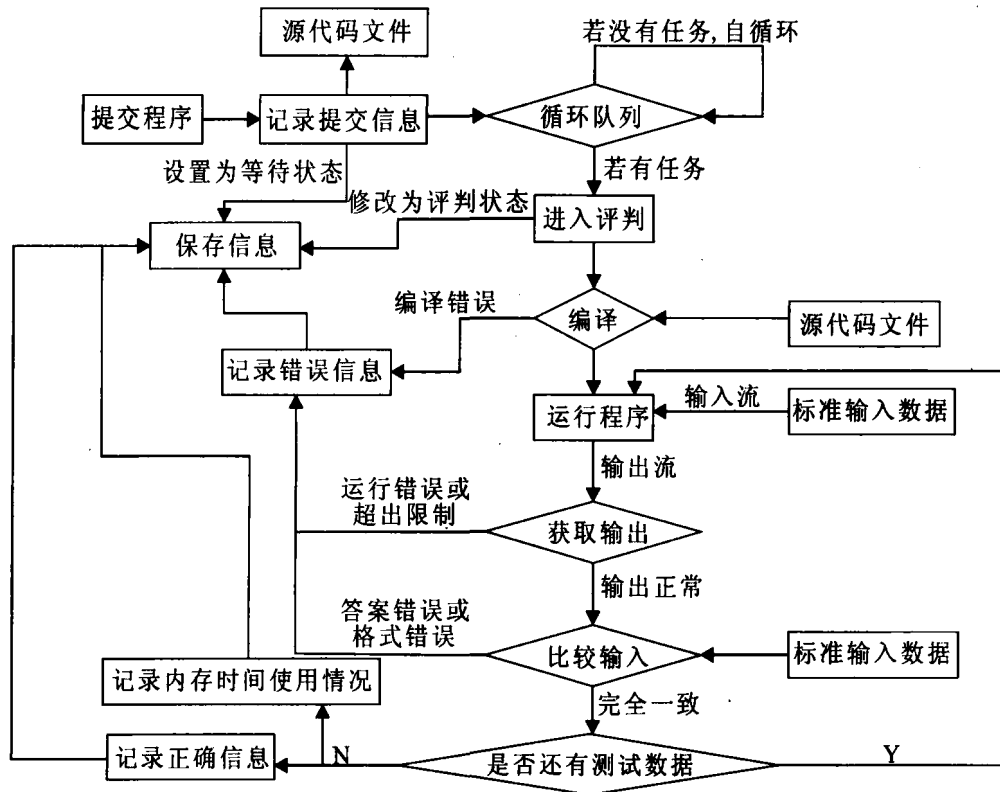


图 2 评判流程图

2 评判内核的实现

用户通过页面表单将题号、语言、解题代码提交后,由业务层做一些持久化工作,并新建一个唯一编号的评判任务,编号是为了让评判内核能够唯一确定某用户在某题上的一次提交.将每一个评判任务都设计为一个独立的线程,^[2]根据用户选择的语言生成对应类型的线程,如 GCCThread,GPPTThread,JavaThread 等,视 OJ 系统支持的编译系统而定,然后立即将其加入评判队列 JudgeQueue.为了监听评判队列的变化,设计一个评判内核启动器 Starter,自动地从评判队列中获取评判任务并启动评判过程. Starter 是一个 Listener,以保证在启动 OJ 服务时立即启动评判内核,由于 Starter 继承 Thread 类,因此只有当评判队列不为空时才进行评判,否则将一直处于休眠状态,这样就在很大程度上节省了系统开销,提高了系统的效率.

评判内核本身并不能充当编译器的角色,它是通过调用其他外部编译器来完成编译工作的,由此设计了一个通用的 Compiler 类来调用外部编译器来实现对解题代码的编译工作. Compiler 类设计如图 3 所示.

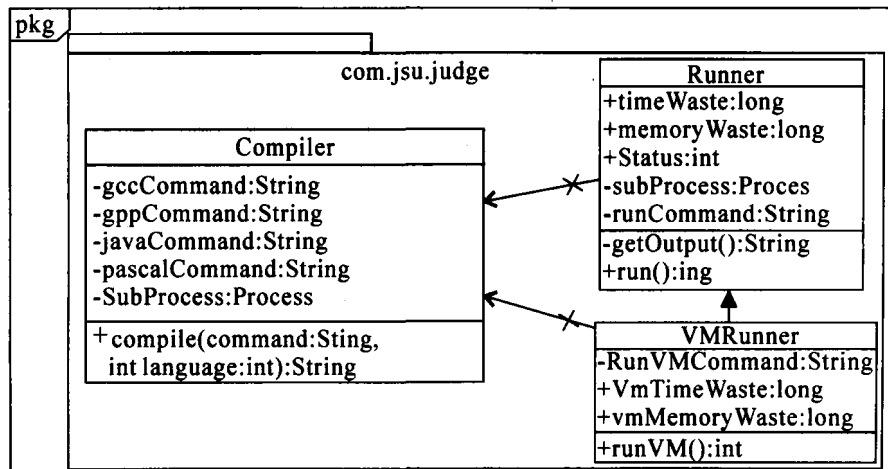


图 3 Compiler 类图

图 3 中 command 是 Java.lang.String 类型,一个可能的值是 gcc-fno-asm-w Main.c-o Main,^[3]它的含义是将 Main.c 源文件用 gcc 编译器编译生成 Main 可执行文件. Compiler 类调用外部编译器进行编译的时候首先生成子进程 subprocess,然后将编译命令 command 传给编译器,通过判断子进程的出口值来确定是否编译成功,当出口值为 0 时表示编译成功,否则视为编译失败,并通过获取子进程的错误流记录来提取错误信息.

按照编译系统的不同编译生成的文件分为可执行文件或字节码文件,前者运行在操作系统之上,后者运行在虚拟机之上. 评判内核自身无法提供这 2 种不同的运行平台,因此设计了 2 个类,Runner 和 VM-Runner,来分别模拟运行这 2 种程序,程序执行时总是作为 Runner 类或 VMRunner 类的一个子进程. Runner 和 VMRunner 类不仅用于执行程序,同时还担任了内存计算、时间消耗等任务. 由于评判任务必须在题目规定的时间内完成评判,当时限到达时即使没有获得运行结果也必须结束子进程,因此设计了一个计时器 TimerTask^[4]来控制子进程的执行时间. 先将标准输入数据通过子进程的输入流传递给编译好的程序,然后立即启动 TimerTask,若在题目规定的时限内仍未从子进程的输出流获得任何结果,则 TimerTask 强制销毁子进程,评判程序运行超时;若在题目规定时限内从子进程输出流获得了运行结果则立即停止 TimerTask 的计时,记算时间消耗和内存消耗,判断子进程的出口值,确定解题程序在执行过程中是否运行错误. 若出口值不等于 0 则表示子进程运行异常,再通过分析子进程的错误流来确定错误的种类,例如数组越界、除数为 0、数据溢出、内存溢出、栈溢出等,否则程序运行正常. 此时应将子进程的输出结果与标准测试数据对比,若去掉前后空白和换行时完全匹配则返回精度错误,若去掉前后空白和换行时也不匹配则返回答案错误,若完全匹配则返回答案正确.

程序经过编译、运行、返回结果之后,业务层将结果、时间消耗、空间消耗等发送到持久层进行保存,并将结果和相关数据(如耗时、内存使用、代码长度、使用语言等)发送给表现层呈现给用户,完成本次评判任务.

3 结语

在设计 OJ 系统时,评判内核可以作为一个单独的子系统来开发,通过定义良好的通信接口有利于降低系统的耦合度. 此外,由于无法预知用户提交的源代码是否嵌入非法操作,因此为提高系统的安全性有必要做出一些限制,可以从 2 个方面考虑:编译时限制和运行时限制. 编译时限制是指删除外部编译器中带有敏感操作的头文件和库函数,运行时限制则是通过 Java 安全机制让子进程运行在一个安全沙箱内,由此可以提高系统的安全性能.

参考文献:

- [1] 盛贤良,瞿有甜. 基于 Linux 的 Online Judge 的设计与实现 [J]. 南京大学学报,2005,41:570-574.
- [2] [美]埃克尔,著. 陈昊鹏,译. Thinking in Java [M]. 第 3 版. 北京:机械工业出版社,2007.
- [3] [美]ARTHUR GRIFFITH,著. 胡恩华,译. GCC 技术参考大全 [M]. 北京:清华大学出版社,2005.
- [4] [美]MESKER WAKE,著. 龚波,译. Java 设计模式 [M]. 北京:人民邮电出版社,2007.

Design and Implementation of Online Judge Kernel

HE Ying-sheng, LUO Qiang

(College of Mathematics and Computer Science, Jishou University, Jishou 416000, Hunan China)

Abstract: According to the present Online Judge System based on B/S structure, the authors put forward a complete evaluation method to realize the judge core, interpret how the OJ system handles user's submitted code and obtain real-time results for user.

Key words: Online Judge; judging; code; program; compile; sub-process

(责任编辑 向阳洁)