

基于 API 自动测试的程序设计在线判题系统的研究与实现

Programming online Judge System based on API Automatic Test

韩志科 王 贵 韩俊杰 (浙江大学城市学院计算学院 浙江 杭州 310015)

摘 要: 本文设计了一个能对程序设计题进行在线编译判题不需任何人工辅助的系统,把基于执行的动态 API 测试方法引入到程序设计在线编译判题中,系统综合运用了 Linux 平台下的 C、Shell 脚本语言技术、编译器实现在线判题,以及数据库技术和 PHP 等技术。

关键词: 程序设计 在线判题 API 自动测试 Linux 编译器

1 引言

随着计算机技术及网络技术的发展,以网络技术为支撑的现代计算机技术应用,已经具备较大的规模并在很大程度上得到普及。利用计算机网络组织实施大规模异地实时在线考试已成为当前各类教学和考试的一种重要方式。网上考试系统具有考试组织方便、考试不受时间和地域的限制,可以快速客观地判题评分和考试成本等相对较低等优点。但是考试系统中对于主观题的自动阅卷算法目前还没有太好的解决方法。主观题的自动评分需要解决很多技术上的问题,因而成为在线考试系统中的一个技术难点。国内外学者在这方面也进行了大量的研究:比如文献[1]利用模糊数学理论解决主观题评分问题,达到了一定的效果;文献[2]对 windows 操作系统环境下的操作题的判题提出了解决方法,它只关心最后答案是否正确,把最后答案与标准答案进行比对,实现了预期效果。本文研究用于程序设计题的判题和评分是一个更为复杂的过程,是一个能对程序设计题进行在线编译判题不需任何人工辅助的系统。在具体的开发过程中,引入了 API 自动测试技术。

2 系统设计

API 是 Application Program Interface 的缩写,它是

一组例程、协议的集合。从程序设计的角度来看,软件都由两种 API 构成:系统 API 和开发人员自定义 API。系统 API 是由系统环境和工具提供的,例如大多数的操作系统环境像 MS - Windows 通过提供一套完整的系统 API 使编程能够存取操作系统提供的各种服务;各种编程语言如 C/C + +、JAVA 等提供的函数库也属于系统 API。软件开发人员通过使用系统 API 来定制自己的 API(开发人员自定义 API),再由这些 API 构成软件模块进而构成完整的软件。API 测试就是运用软件测试技术与方法对 API 进行正确性测试,侧重于软件的正确性测试,包括逻辑正确性、功能正确性等等。目前国内外已经有了一些相对成熟自动化测试工具。

测试一般由下列过程组成^[3]: (a) 建立初始条件; (b) 用符合要求的参数调用 API; (c) 结果分析。通过对 API 测试过程和程序设计判题过程的对比研究,可以发现其中有很多重复性很强的工作,极为适合于自动化。所以可以把基于执行的动态 API 测试方法引入到程序设计在线编译判题中。但是系统要实现在线编译判题必须通过设计独特题库与编制合理的测试用例、设计执行控制脚本和严格的编译指令文件,进行测试执行结果分析,最后通过分析测试执行结果来评估 API 程序的正确性,给出评分。

图 1 是程序设计在线判题系统的系统总体结构

图,描述了在执行测试用例时各功能部件之间的关联。部件间的箭头表示通信和交互关系。从图中可见程序设计在线判题系统是 API 自动测试系统构框的子集。

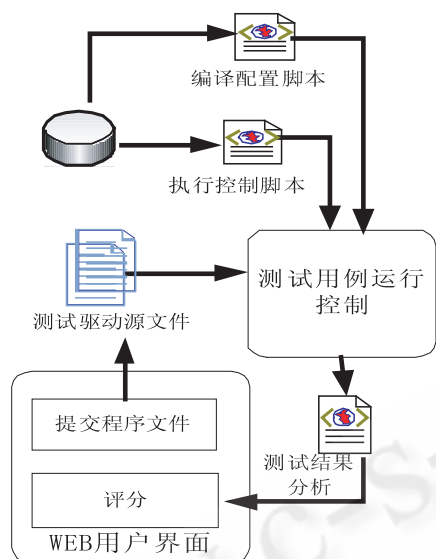


图1 系统总体结构图

3 系统实现

程序设计在线判题系统是一个基于网络的在线判题自动判题考试系统,核心是在于测试用例运行控制以及在线编译器和在线判题器的设计和实现。Linux 操作系统提供了一个良好的实现平台,系统可以通过 Linux 平台下的 Shell Script 脚本语言和 GCC、G++ 及其他编译器实现系统功能。

3.1 程序设计在线判题系统的题库设计

程序设计在线判题系统的题库由各类基本程序设计题组成,程序设计题目是进行判题和评分的单元,是测试驱动源文件。在本系统中,并不是所有的题目都能够进行在线编译判题,所以设计时必须针对程序设计题程序设计在线判题系统设计一个特定的题目结构,这种题目结构的目的在于考核学生对于一定的知识点的掌握并且能够被正确评估,一个切实可行的方案是限定程序题目的输入和输出,在程序中适当的留空,在这种条件下系统就能够有效实现程序正确性的判定。每一道题目由以下五部分组成:Input(标准输入),Output(标准输出),Fill.c(程序填空),Text(题目

的正文),Title(题目目标)。对于一个考核学生是否掌握求两个数和与差的题目示例如下:

(a) Text:题目的内容,是对题目的说明。如题:“程序填空,不要改变与输入输出有关的语句。输入整数 a 和 b,计算并输出 a、b 的和与差。例:输入 2 -8; 输出 The sum is -6;The difference is 10”。

(b) Title:题目标题,如上述的题目标题就是“求两个数的和与差”。

(c) Input:程序的标准输入。如上述的题目输入就是“2 -8”。

(d) Output:程序的标准输出,作为提交程序输出的语义匹配的模板。如上述的题目输出就是“The sum is -6;The difference is 10”。

(e) Fill.c:程序主体,空缺部分需要学生填写完整(以 C 语言为例),空缺部分可以是所要考核的知识点的关键词。比如:

```
#include <stdio.h>
int main() {
    int a,b,sum,diff;
    scanf("%d%d",&a,&b); /* fill_b */
    sum = a + b; //空缺部分,需要考生填入
    diff = a - b; /* fill_n */
    printf("The sum is %d\n",sum);
    printf("The difference is %d\n",diff);
}
```

程序题目作为记录存入数据库中,其中还包括其他的属性,例如题目类型,知识点,难度,程序进程运行的时间、空间限制。题目类型、知识点、难度用于出卷;程序进程运行的时间、空间限制用于控制判题时服务器资源。图2是程序设计在线判题系统的实现原理。

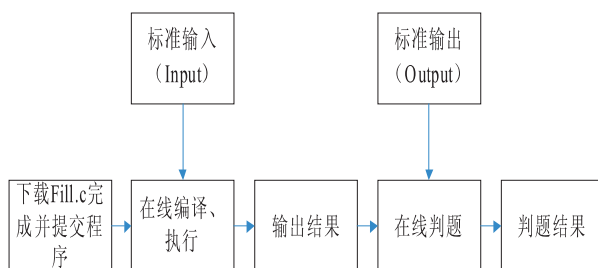


图2 判题原理

3.2 程序设计在线判题系统的测试用例运行控制的实现

程序设计在线判题系统的测试用例运行控制是整个系统运行控制中心,包含调度编译和判题两个功能。这部分的实现需要涉及操作系统的部分底层操作,如信号操作、进程间通信和控制等,所以系统测试用例运行控制主程序选择 Linux 平台下 C 实现。用户交互由 PHP 程序实现,后台数据库系统采用 MySQL。PHP 程序负责与用户交互,处理用户登录、查看题目、提交答案、比对输出和显示结果等请求。系统测试用例运行控制具体设计流程如下:考生首先在 PHP 页面下载程序题目,完成程序填空部分或者整个程序,然后提交题目到判题服务,并在数据库中插入相应的记录,提交要求判题信号,然后主程序收到信号后读取数据库记录,并且调用判题主程序。

判题主程序是判题系统的核心子系统,它主要功能是负责接收和分派判题请求,并且调用判题模块实现判题评分。判题主程序首先主要负责从数据库中读取所需要的题库信息 (Input、Output 和 Fill. c),为每一题目建立临时工作目录,并把这些信息保存到临时工作目录。一旦考生通过 WEB 页面提交要求被评估源代码,判题主程序接收到提交请求并且自动查询数据库,检查题库表中是否有对应题目存在,并同时查询题目状态是否为可提交到数据库判题运行表 (t_runs 表)。然后判题主程序从数据库中读取提交的源代码,把用户提交的源程写入到对应临时磁盘文件,写入判题请求到进程通信管道,请求中包括请求判题的 ID (runid),用来唯一标识。然后调用判题模块,先对提交的程序进行编译,接着把编译运行后的输出与对应的题目的 Output 进行比较,判断答案是否正确,将结果写入数据库,并删除临时文件。具体实现见图(3)。

3.3 在线编译器和在线判题器的设计和实现

判题模块负责处理判题的具体实现,包含在线编译器和在线判题器两个重要部件,是一个处理用户、操作系统、数据库、判题系统多方面交互的过程:判题模块首先查询数据库判题运行表 (t_runs 表) 获得提交详细信息,同时查询当场考试信息表获得当前考试时间,比较提交时间是否在考试时间内,如果不在考试时间

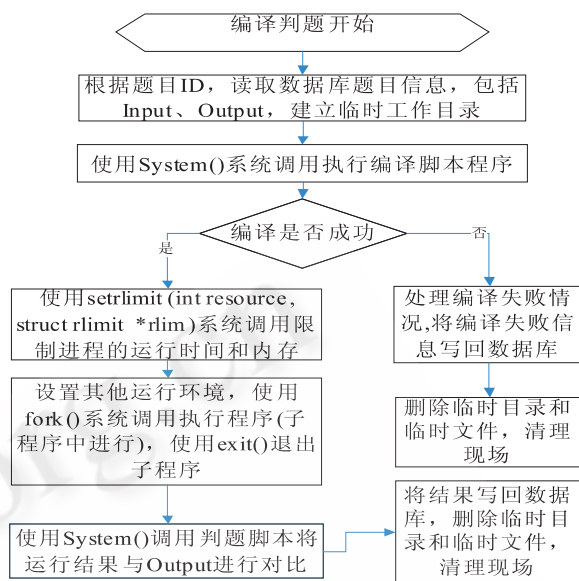


图3 在线编译判题流程

内则判定超时并返回。如果在考试时间内,调用编译配置脚本,传入参数题目号、使用语言,使用语言决定调用 GCC、G++ 或其他编译器。然后编译脚本对程序进行编译并执行,输出提交程序的结果文件。下面是编译 C、C++ 和 JAVA 为例编译配置脚本,使用 Bash Shell 脚本语言实现。

```

#!/bin/bash
# arguments:
# $1: runid
# $2: language string ( "C" , "CPP" , "JAVA" , etc )
# < Note: runid is combine with Contest ID ( / 1000000 ) , real runid is ( % 1000000 )
# return value:
# 0: script execute OK, 1: script error ( wrong arguments )
# output:
# $ runid.compile.result: compilation result ( " Success" or " Failed" )
# $ runid.compile.error: compilation error message generated by compiler
# Section 1: 设置运行环境
#设置工作路径
ZOJROOT = /opt/ZOJ
umask 002
  
```

```

# C special
CC = gcc
CFLAGS = " -ansi -fno -asm -Wall -lm - -static
-DONLINE_JUDGE"
# CPP special
CPPC = g + +
CPPFLAGS = " -ansi -fno -asm -Wall -lm - -
static -DONLINE_JUDGE"
#JAVA special
...
#Section 2: Check script parameters
if [ "$1" == "" ]; then
echo Run ID is empty
exit 1
fi
ID = $1
#处理每一题工作路径
( ( TEMP = $ID/1000000 ) )
TESTSROOT = `printf "%s/contests/%04d" $ZOJROOT
$TEMP`
cd $TESTSROOT/working
( ( ID = $ID % 1000000 ) )
TEMP = $ID
( ( P1 = $TEMP%100 ) )
( ( TEMP/=100 ) )
SOURCE = `printf "%02d/%02d/%02d.source" $P3
$P2 $P1`
#Section 3: begin to compile
case $2 in "C" )
Cp $TESTSROOT/runs/$SOURCE $TESTSROOT/work-
ing/$ID.c
$CC $TESTSROOT/working/$ID.c -o $ID.exe $C-
FLAGS 2 > $TESTSROOT/runs/$SOURCE.compile.er-
ror
if [ -f $ID.exe ]; then
echo `Success` > $ID.compile.result
strip $ID.exe
else

```

```

echo `Failed` > $ID.compile.result
fi
rm $TESTSROOT/working/$ID.c...
在线判题器使用 Bash Shell 脚本语言实现。首先
检查由编译脚本输出的结果文件,如果编译出错则判
定 Compile Error(编译错误)并返回。如果程序编译通
过,判题脚本 fork 出子进程,根据题目的相应限制设置
进程运行的时间、空间限制。同时判题脚本判断子进
程运行及返回状态,分别判定 Runtime Error(运行时错
误), Time limit exceeded(程序运行超时), Memory
limit exceeded(内存使用超限),并且返回。最后更新
数据库判题运行表(t_runs)写入判题结果和输出,删
除工作临时文件,进行语义匹配,给出分数。判题脚本
部分源代码如下:
# arguments:
# $1: run id
# $2: problem id
# output:
# $ runid. judge. result: judge result ( "Yes" or "No"
or "PE" )
#PE 为部分正确
if ( diff -a $CONTESTSROOT/working/$ID.output
$CONTESTSROOT/problems/$PID/output 1 >/dev/
null 2 >/dev/null ) then
echo `Yes` > $CONTESTSROOT/working/$ID.
judge.result
elif ( diff -a -b -B $CONTESTSROOT/working/
$ID.output $CONTESTSROOT/problems/$PID/output
1 >/dev/null 2 >/dev/null ) then
echo `PE` > $CONTESTSROOT/working/$ID.judge.re-
sult
else
echo `No` > $CONTESTSROOT/working/$ID.
judge.result

```

最后用户可以通过 PHP 页面用 WEB 方式查询数
据库,获取相关判题结果信息。

4 结束语

程序设计语言熟练程度和程序设计能力是计算机相关专业学生的最基本的技能。传统的上机练习往往依赖于教师的指导,在教师资源有限的情况下,上机指导教师的工作量普遍较大。设计一个能对程序设计题进行在线编译判题不需任何人工辅助的系统,不仅可以给学生提供一个训练程序设计能力的互动平台,也使教师能够通过系统更能把握学生对教学内容掌握程度并可作为监督学生学习的重要工具,还可以在一定程度上强化实践、激发自主性;其次,作为一个网络教学练习和考试系统,提供了在线判题功能,可以有效降低教师的阅卷工作量。系统在应用过程中适用于三类角色:管理员、教师和学生。系统主要实现:用户管理、题库管理、考试/练习管理、公告管理、学生进行考试/练习、在线判题、查看提交情况等。另外系统还包括辅助的系统出错处理,出错时,中止运行页面,并显示出错信息。在考试状态下,系统对于每个考生登陆后都被锁定于当前计算机的 IP 地址,考试中途在未经教师的允许下不能自行更换位置,利于有效地维持考场纪律和考试的公证性。

程序设计在线判题系统采用 IP 地址区分的方法抽题组卷的方式,即相邻 IP 地址(在实验室机房往往是物理上的相邻)抽取的试卷是不同的,同时生成多套试卷。试卷整体知识点覆盖面大,可基本反映考生的实际水平,可以有效防止突击复习或猜题、作弊得高分等不良现象。本系统基于大量的实践教学后设计而成,并在反复的实践教学试用和改进后,在计算机专业的本科教学中取得理想的效果。学生、教师以及管理员直接通过 WEB 浏览器使用本系统。

系统的设计所采用的特定软硬件环境:系统硬件环境基于 Intel CPU 的服务器或高性能 PC 机;采用

Redhat Linux 9.0 或更高版本本地操作系统,结合 Apache 1.3 + PHP 或更高版本 WEB 服务器和 MySQL 数据库服务器,如果数据库规模变大,可以考虑迁移到 Oracle。

本系统的开发目的是直接服务于程序设计语言课的程序设计实验教学。目前,已经在浙江大学、浙江大学城市学院投入使用,在实际使用当中系统在 HP P4 933Mhz 双 CPU,2G DDR RAM 的服务器上可以支持 600 人同时在线考试和判题,具有很好的使用价值和推广价值。系统开发了 GCC 支持编译的 C 语言,可对 Linux C,以及其他编译器所支持的设计语言如 C++,Java,PASCAL 进行成功判题。

参考文献

- 1 考试系统中主观题评分的算法研究. 东北财经大学 [硕士学位论文],2006-12.
- 2 windows 操作题自动阅卷的研究与实现. 计算机系统应用,2007,(10).
- 3 刘幕涛,张磊,王艳等. 基于 XML 的 API 自动化测试工具设计与实现. 计算机工程,2007,7(13): 96-98.
- 4 崔红军,饶若楠,劭培南. 一种 API 自动化测试工具的设计与实现. 计算机工程,2007,2(4): 270-274.
- 5 李玉波,朱自强,郭军. LINUX C 编程. 清华大学出版社,2005-9.
- 6 Arthur Griffith. GCC 技术参考大全. 清华大学出版社,2004-7.
- 7 Falk C K J. Nguyen H Q. 计算机软件测试. 2 版,王峰译,北京:电子工业出版社,2004.
- 8 Simon Cozens Advanced Perl Programming, 2/e. O'Reilly,2005-8.