评测系统剖析与选手参赛策略

清华大学 王逸松
（wangyisong1996@gmail.com）

摘要

本课程主要介绍评测系统的主要原理和实现，并对信息学竞赛相关的几种评测系统做一个简单剖析，以及分析在不同评测系统下选手的参赛策略。

关于本讲义

本讲义只对本课程的内容进行简单介绍。欲下载完整课件，请访问
https://wys.life/201901-Judge-Lecture.html

或扫描下图中的二维码：

目录

1 引言

信息学竞赛的题目通常是，给定一个任务，要求编写一个程序，实现该任务的要求。而“评测”就是对一个程序进行各方面评价的过程。为了给选手一个较为客观的评价，评测的过程通常由“评测系统”自动完成。可见，评测是信息学竞赛中的重要环节，而评测系统对评测起到了关键作用。研究评测系统，对于选手参赛来说，也是非常重要的。

在本课程的第一部分，我们将基于现有的竞赛题目，分析和讨论如何在 Linux 操作系统上实现评测。在本课程的第二部分，我们将对现有的信息学竞赛相关的几类评测系统进行简单剖析，并据此讨论不同评测系统下选手的参赛策略。

接下来的内容为课程各部分的简要介绍。

2 评测系统的分析与实现

为了对评测系统进行分析，首先需要了解评测的定义。信息学竞赛的“评测”通常是，给定一个源程序和若干组输入文件，要求在某种操作系统下，编译并运行该程序，得到对应的输出文件。在程序成功运行的前提下，测量其时间、内存的使用量，并将输出文件与标准答案进行比对。

基于评测的定义，我们接下来介绍评测系统的各方面内容。

2.1 测什么

评测系统接受一份选手源程序和一组数据（包括输入文件、答案文件）作为输入，编译并运行选手源程序，如果成功运行，则将选手程序的运行时间、内存使用量，和输出是否正确作为评测系统的输出，否则判定选手程序运行出错。接下来我们详细列举需要评测的内容。

- **编译**：使用评测系统中的编译器，是否能将选手源程序编译成评测系统的机器代码。若在此过程中若出错，则返回错误信息。
- **运行时间**：从选手程序启动到结束，这一段过程中选手使用的时间。需要注意的是，不同操作系统可能提供了不同的时间定义。
- **内存用量**：在选手程序的整个运行周期，最多同时使用了多少内存。这也需要根据具体的操作系统来进行定义。
- **输出正确性**：选手程序读取输入文件，产生输出文件。我们将其与答案文件进行某种方式的比对，来确定选手的输出是否正确。

2.2 怎么测

评测系统通常运行在 x86-64 架构的 GNU/Linux 操作系统上。我们可以用下列工具或命令来帮助评测。

- **编译**：使用 <编译器> <选手程序> <编译选项>，如对于 C++语言，开启 O2 优化开关，可用 `g++ main.cpp -O2`。如果编译器输出了错误信息，则认为编译失败，否则编译成功。
- **运行时间**：使用 `time` 命令，可以测量一个程序（即进程）的运行时间。如 `time ./a.out` 即可测试可执行文件 `a.out` 的运行时间。

- **内存用量**：在 Linux 操作系统中，可以通过/proc/<pid>/status 来查看各项状态，其中包括了内存使用量。其中<pid>为进程号，可以在启动一个进程时获得。
- **输出正确性**：使用 Linux 的重定向功能，如./a.out < in.txt > out.txt，即可将 a.out 的标准输入重定向到 in.txt，标准输出重定向到 out.txt。再使用 diff out.txt ans.txt 即可比较 out.txt 和 ans.txt 是否相同。
- **时间/内存限制**：题目可能要求选手程序的运行时间/内存不能超过某个上限，这时可用 ulimit 命令来限制这些资源的使用。如 ulimit –t 5 表示限制时间不超过 5s，ulimit –v 524288 表示限制内存使用不超过 512MB。

2.3 精确性

在使用任何一种测量工具时，都应考察其精确性。

对于 Linux 操作系统的 time 工具，可以测量一个进程的 real/user/sys time，分别精确到 0.001s。对于评测而言，通常使用 user time 作为选手程序的运行时间。

对于/proc/<pid>/status 中关于内存的测量，可以精确到 4KB，也就是操作系统的页（物理内存分配的最小单位）的大小。在评测中，我们通常使用 VmPeak 的值作为选手程序使用的内存，它表示该进程使用的物理内存的最大值。

2.4 稳定性

在多次评测同一个程序和同一个输入时，容易发现运行时间并非确定，甚至有较大的波动。这主要是因为在每次运行时，物理内存的布局不一致导致缓存的效果不一样，以及其他进程的运行也会给选手进程带来不确定的干扰。为了增强稳定性，通常可以减少评测时的其他干扰因素，如鼠标、键盘的输入，其他进程的运行。

2.5 安全性

安全性是评测系统的重要特性。一个安全的评测系统，应能够防止选手对其进行攻击，如阻碍评测系统的正常运行，或窃取答案文件。在 Linux 系统中运行选手程序时，可以使用 ptrace，来“监视”选手进程的每个系统调用，从而阻止可疑的行为；也可以使用虚拟化技术，将选手程序放在一个完全隔离的环境中运行。

2.6 高效性

评测系统通常为比赛服务，或作为在线评测网站的后端。为了尽快完成选手提交的评测任务，高效性也是评测系统的一大特点。由于多个评测任务之间完全独立，使用多台评测机是个常见的解决的方案。这时需要一个管理多台评测机的机制。

2.7 易用性

评测系统作为一个直接面向选手的软件，拥有良好的界面或功能是必要的。很多评测系统有图形界面，容易进行操作；另有很多评测系统为在线评测系统，它们通常以网站的形式呈现。

3 现有评测系统与选手参赛策略

这部分主要介绍在信息学竞赛相关的领域现有的评测系统，并讨论选手在使用这些评测系统时的参赛策略。这些评测系统主要包括：

- 离线评测系统，以 Arbiter 为例，多用于 NOI 等比赛。
- 在线评测系统，限制每道题的提交次数，多用于 IOI 等比赛。
- 在线评测系统，不限制提交次数，但计算“罚时”，多用于 ICPC 等比赛。

3.1 离线评测系统

离线评测系统不依赖网络即可运行，可以在本地进行测试。这类评测系统适合选手的训练和测试。有一些离线评测系统也支持局域网内的联机，可用于考后统一评测的比赛。

3.2 在线评测系统

在线评测系统通常是一个网站，需要网络的支持。当选手通过网页提交程序时，后台的评测机将自动完成评测任务，并在网页上给予反馈。在线评测系统适合提供网络题库服务，或者网络比赛，即选手可以在任何地方参赛；也适合在同一地理位置举办的比赛，此时在线评测系统能给选手更好的体验。

从功能来看，有些评测系统，选手在比赛时只能提交一定次数（如 100 次），每道题的得分为每次提交得分的最大值。另外一些评测系统，选手可以提交任意多次，但是每次提交会产生一定的“罚时”，最终按照通过题目数和罚时进行双关键字排序。

3.3 选手参赛策略

在不同评测系统下，选手除了有足量的知识储备和技能训练外，也要有合适的参赛策略。竞赛中的题目通常设有“部分分”，在不能解出一道题时，可以选做难度较低的部分分。但是，竞赛的规则（如是否在线评测、提交次数限制的多少）决定了对部分分的正确处理。选手应该尽可能利用评测系统的特点，在有限时间的竞赛中得到更多的分数。

参考文献

[1] 中国计算机学会，“NOI 竞赛规则”，
http://www.noi.cn/newsview.html?id=68&hash=CDD941&type=6
[2] International Olympiad in Informatics，
“International Olympiad in Informatics Regulations”，
https://ioinformatics.org/files/regulations18.pdf
[3] icpc.foundation，“World Finals Rules for 2019”，
https://icpc.baylor.edu/worldfinals/rules
[4] 吕凯风，“下一代测评系统”，CCF NOI2016 冬令营授课