由于 Linux 系统具有安全性高、 稳定等优点， 目前大部分服务器都是使用 Linux 系统。本文提出一个基于 Linux 的 ACM 在线评测系统，详述该系统中编译、 测试、 获取计算机消耗情况、 返回测试结果这几种功能的具体实现。并分析该系统中可能出现恶意占用资源、 恶意调用系统等不安全因素， 提出相应的解决方案。该系统具有实用性强、 安全性好等优点。

ACM 评测系统核心是接受用户提交的程序， 把程序的运行结果返回给用户，运行结果为程序正确或程序不正确。程序不正确包括: 运行错误( run-time error) 、 运行超时( time-limit exceeded) 、 运行结果错误( wrong answer) 以及运行结果输出格式错误( presentation error) 。

服务器后台运行守候程序，每一个时间间隔查询一次数据库，如发现有提交纪录，

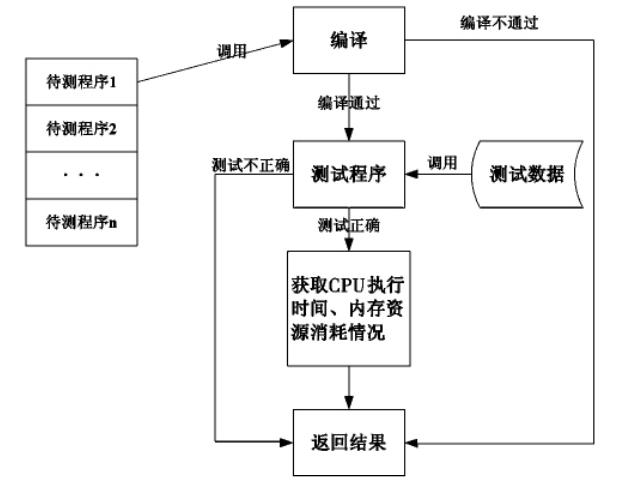
这启动评判进程，进行相关题目的评判。每道题目均对应一个数据输入输出文件，本次

设计采用 Linux 管道处理，让评判进程启动一个子进程编译运行用户提交上来的程序，

当程序启动时计时，在时间允许范围之内，如果其得到了正确答案，则修改数据库，令

其 status 为 Accepted，否则，如果答案错，或者表达错，运行错等，则向该进程发出

SIGKILL 信号，强行中止该进程，把数据库记录进程相应的修改。



编译：ACM 评测系统首先需要对提交的待测程序进行编译。对于 C、 C ++ 程序的编译链接，采用 GNU 的旗舰产品 GCC。但是由于 GCC 没有提供应用程序接  
口，因而需要使用 system 函数。在本系统中， 以调用shell 命令的方式来调用 GCC 编译， 链接编译用户提交的 C、 C ++ 程序，

1，WIFEXITED(status) 这个宏用来指出子进程是否为正常退出的，如果是，它会返回一个非零值。

（请注意，虽然名字一样，这里的参数status并不同于wait唯一的参数--指向整数的指针status，而是那个指针所指向的整数，切记不要搞混了。）

2，WEXITSTATUS(status) 当WIFEXITED返回非零值时，我们可以用这个宏来提取子进程的返回值，如果子进程调用exit(5)退出，WEXITSTATUS(status) 就会返回5；如果子进程调用exit(7)，WEXITSTATUS(status)就会返回7。请注意，如果进程不是正常退出的，也就是 说，WIFEXITED返回0，这个值就毫无意义。

3,WIFSIGNALED(status)若为异常结束子进程返回的状态,则为真

当然，处理进程退出状态的宏并不止这两个，但它们当中的绝大部分在平时的编程中很少用到。

3、WTERMSIG宏测试

WTERMSIG：测试作用通过信号来决定那个子线程被退出。  
**命令格式：**

#include <sys/wait.h>

int WTERMSIG(int status);

调用情况：与wait()和 waitpid()函数结合使用。  
返回值：WTERMSIG宏测试被执行后，若成功返回被终止的子进程的信号值。返回的信号值被定义在 sys/signals.h头文件中。

system与exec的区别：

（1）system与exec都可执行进程外的命令，但是system是在原进程上新开辟一个进程，而exec是用新进程（命令）覆盖了原来的进程。

（2）二者都会产生返回值，system的返回值不影响原进程，exec的返回值影响原进程。

1. 进程控制wait3和wait4

wait3和wait4提供的功能比wait、waitpid和waitid所提供的功能要多一个，这与附加参数rusage有关。该参数要求内核返回由终止进程及其所有子进程使用的资源汇总。

pid\_t wait3(int \*statloc, int options, struct rusage \*rusage);

pid\_t wait4(pid\_t pid, int \*statloc, int options, struct rusage \*rusage);

两个函数返回值:若成功返回进程ID，若出错则返回-1

      资源统计信息包括用户CPU时间总量、系统CPU时间总量、页面出错次数、接收到信号的次数等。有关细节参阅getrusage(2)手册页。

 ls /proc 能看到一些数字命名的文件夹，这个就不多说了，什么pid、fd的知识还是容易弄到的  
拿init开刀，  
[root@localhost proc]# cat /proc/1/statm  
487 185 133 31 0 67 0  
很简单地返回7组数字，每一个的单位都是一页 （常见的是4KB）  
分别是  
size:任务虚拟地址空间大小  
Resident：正在使用的物理内存大小  
Shared：共享页数  
Trs：程序所拥有的可执行虚拟内存大小  
Lrs：被映像倒任务的虚拟内存空间的库的大小  
Drs：程序数据段和用户态的栈的大小  
dt：脏页数量