**程序自动评测系统**

姓 名：

课题组的分工或贡献：

课 程 名 称：

指 导 教 师：

2017年6月

项程序自动测评系统

（燕山大学 信息科学与工程学院）

1. 为实现软件工程老师对学生平时编写的程序进行测评，尽可能的实现自动化管理学生信息，题目信息和学生的题目的通过情况的需求，为了进一步促进教学水平，本项目采用Linux shell编程，在Linux系统环境下完成该程序自动测评系统。

# 需求分析

**1.1问题描述**

帮助教师开发一个程序自动评测系统（教师用）。教师有多道C语言题目要求及测试用例。学生完成程序设计后，提交到系统。系统自动评测学生提交的C程序是否通过测试。

**1.2功能需求**

(1)存储需求。存储学生的基本信息（ID号，姓名，班级，学号）。可以逐条输入，可以批量导入。

存储题目信息（题目ID号，题目名称，题目描述，题目测试案例1，2，3等输入，输出文件，题目要求的程序运行时间、内存大小）。可以逐条输入，可以批量导入。

存储学生提交的程序（成功通过测试的程序，应该保存。不成功的程序，删除）；

存储学生提交程序的过程信息（提交的题目编号，提交时间，是否通过？不通过的原因（编译错误；运行时间超时，内存超出））。

(2)控制流程。系统为单机版运行，由系统管理员一人操作。系统的功能分为两个方面。①基本信息维护（设置）。管理员导入或输入学生基本信息，题目信息。

②评测程序的过程。学生的程序被复制到某个目录下面，系统管理员调用自动评测脚本，评测程序。如果通过，则存入数据库，如果没有通过，则给出程序的出错信息，但在数据库中保存一次记录。

# 系统架构和程序设计

## 系统架构

整个系统分为两个脚本程序和一个数据库管理系统。用J1和J2分别表示两个脚本程序。J1负责基本信息维护（设置），J2负责评测程序。数据库存储数据。

Mysql数据库

J1

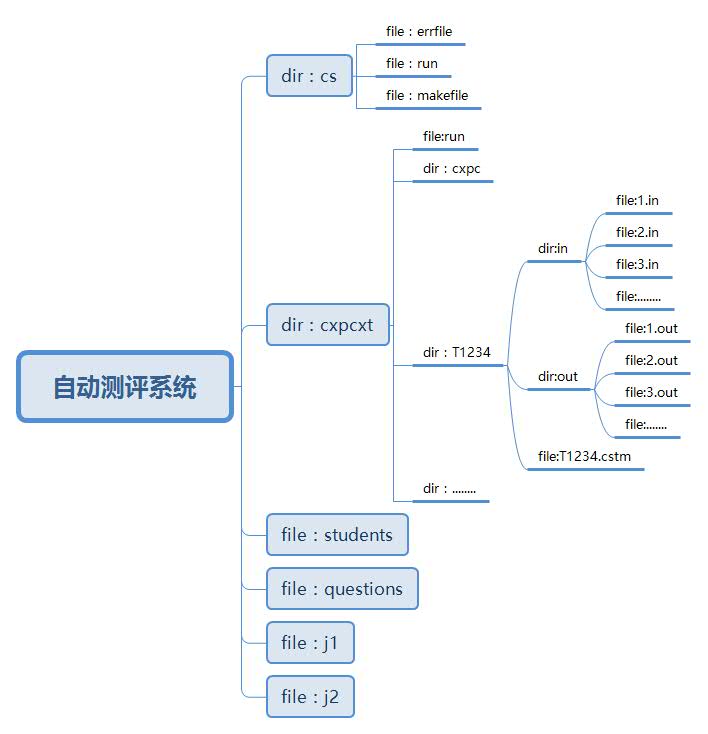
J2

图1.总体设计图

**2.2 开发工具**

开发工具采用：Linux+ Bash脚本语言+MySQL数据库。

**2.3 程序使用目录介绍**

****

**目录介绍:**

***cxpcst***目录：原始题目（包括自定义的测试数据）和学生提交的.c文件夹

* cxpc ：负责存放学生提交的.c文件
* T1234：题目名为T1234的存放文件夹，包含题目说明信息T1234.cstm，存放多个测试输入信息in文件夹，存放多个测试题目输出信息out文件夹。可以有多个类似该文档结构的题目文件夹存在

***cs***目录:

* 存放负责编译学生提交的.c文件的makfile文件
* 负责捕获错误信息的errfile
* 负责捕获运行输出的run文件

***students***文件：用于存放学生信息，和学生信息的批量导入。文件必须具有一定的格式才会 有效，其有效格式为一个学生的信息占据一行，每个数据用英文格式的逗号

“,”间隔开，用例：

112,xiao,201777,0036

222,mi,201502,77

***questions***文件：用于存放题目信息，和题目信息的批量导入。文件必须具有一定的格式才 会有效，其有效格式为一个题目的信息占据一行，每个数据用英文格式的逗 号“,”间隔开，用例：

1,ppp/22

2,taoey/test

***j1***文件：

实现基本信息维护（设置）。管理员导入或输入学生基本信息，题目信息。

J1的功能：新增、删除、修改、查找学生基本信息（学生信息表）；新增、删除、 修 改、查找题目信息（题目信息表）

***j2***文件：

负责评测程序。学生（教师）将学生程序复制到 cxpcxt/cxpc目录下。学生提交的 程序命名规则：XSID.TMID.C。

J2自动（可以）访问/cxpcxt/cxpc。如果 cxpcxt/cxpc有.c程序。则自动评测，并将 评测结果写入到提交过程表。

**2.4 数据库设计**

设计多张mysql数据表。

(1)学生信息表

表1 学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文描述 | | 学生信息表 | 备 注 | | |
| 表名 | | XSXXB |  | | |
| 字段名 | 中文描述 | 字段类型 | 是否可以空值 | 长度 | 备注 |
| XSID | ID号 | 数值 自增 | 否 | 4位 | 主键 |
| XM | 姓名 | Char | 是 | 20 |  |
| BJ | 班级 | Char | 是 | 20 |  |
| XH | 学号 | Char | 是 | 20 |  |

MySQL建表语句：

create table `XSXB`(`XSID` int(4) auto\_increment   primary key not null , `XM` char(20), `BJ` char(20), `XH` char(20)) ;

(2)题目信息表。存储题目信息（题目ID号，题目名称，题目描述，题目测试案例1，2，3等输入，输出。题目要求的程序运行时间、内存大小）。可以逐条输入，可以批量导入。

表2 题目信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文描述 | | 学生信息表 | 备 注 | | |
| 表名 | | TMXXB |  | | |
| 字段名 | 中文描述 | 字段类型 | 是否可以空值 | 长度 | 备注 |
| TMID | 题目编号 | 字符 | 否 | 5位 | 主键 |
| CCLJ | 测试题目路径 | 字符 | 是 | 1000 |  |

MySQL建表语句：

create table `TMXXB`(`TMID` varchar(5) primary key not null, `CCLJ` varchar(1000));

(3)提交过程表。存储学生提交程序的过程信息（提交的题目编号，提交时间，是否通过？不通过的原因（编译错误；运行时间超时，内存超出）。

表3提交过程表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中文描述 | | 学生信息表 | 备 注 | | |
| 表名 | | TJGCB |  | | |
| 字段名 | 中文描述 | 字段类型 | 是否可以空值 | 长度 | 备注 |
| TJID | ID号 | 数值 自增 | 否 | 4位 | 主键 |
| XSID | 学生id | 数值 | 是 | 4 | 必须是学生信息表中的XSID。否则提示首先注册。 |
| TJSJ | 提交的时间 | Date | 是 |  |  |
| TMID | 此次提交的题目ID | 字符 | 是 | 5 | 题目信息表中TMID。 |
| PCJG | 提交后，系统自动评测，评测的结果 | 字符  （接受，编译错误，测试错误，运行超时，内存超出） | 是 | 8 |  |

MySQL建表语句：

 create table `process`(`TJID` int(4) auto\_increment primary key not null, `XSI` int(4), `TJSJ` Date,`TMID` varchar(5), `PCJG` varchar(8));

**2.5 j1和j2 shell程序功能分析与设计**

**2.5.1 j**1需要实现的具体功能：对于数据库的增删改查操作，学生信息表和题目信息表的数据维护工作。

需要shell对MySQL数据库进行直接的操作，来完成用户的相应需求

**2.5.2** j2需要实现的功能:

(1)定时。每60秒访问一次cxpcxt/cxpc。判断/cxpcxt/cxpc目录下是否有.c程序。如果有.c程序，则for循环，对每一个.c程序处理。

(2).c程序处理过程。将.c文件名分成两个部分，学生学号和题目编号。将.c程序复制到cxpcxt/cs/temp.c。调用make文件，编译链接成为可执行文件（可以用临时文件名temp，以后删除）。根据题目编号，在题目子目录如cxpcxt/T1234， 中找到输入测试文件的数量（date.in文件的个数）。调用每个输入测试输入文件，每个输出测试文件，测试temp。temp的输出文件与输出测试文件比对，比对结果写入一个文件，如果该文件长度大于0，则测试错误。根据测试结果在提交过程表中插入相应的记录。

(3).在测评学生的相应程序时，会在程序中进行各种错误判断，如果出现一种程序错误，shell程序将会直接跳出，等待60秒，程序将会再次相应。

# 详细功能展示及核心源码分析

## j1程序分析

J1程序的整体架构采用了用select语句搭建起的交互式界面，详细的功能抽象出来，写为一个函数，从而简化了整个主函数的逻辑，同时便于程序的维护。下因为程序对学生信息表的操作和对题目信息表的操作类似，所以本文将以操作学生表详细分析具体功能的实现。

**3.1.1数据库的连接和MySQL语句的执行**

在j1程序中每次执行一次数据库操作必须要进行数据库的连接和关闭两个操作。本程序采用的基本框架是：

#!/bin/bash

user=root

passwd=123456

mysql -u$user -p$passwd -e "

相应的MySQL执行语句

quit

"

**3.1.2学生表信息的增加函数：**

包括从程序主目录中读取students文件导入数据库中。

stuCre(){

select var in "insert by keyboard" "insert by files" "quit";

do

if [ "$var" == "insert by keyboard" ]

then

echo "Please input the order,name,class,id in order "

read \_order \_name \_class \_id

mysql -u$user -p$passwd -e "

use LinuxDatas;

insert into XSXXB(XSID,XM,BJ,XH)values('$\_order','$\_name','$\_class','$\_id');

select \* from XSXXB where XH="$\_id";

quit

"

echo -e "1.insert by keyboard\n2.insert by files\n3.quit"

elif [ "$var" == "insert by files" ]

then

while read LINE

do

XSID=`echo $LINE |cut -d , -f 1`

XM=`echo $LINE |cut -d , -f 2`

BJ=`echo $LINE |cut -d , -f 3`

XH=`echo $LINE |cut -d , -f 4`

mysql -u$user -p$passwd -e"

use LinuxDatas;

insert into XSXXB(XSID,XM,BJ,XH)values('$XSID','$XM','$BJ','$XH');

select \* from XSXXB where XSID='$XSID';

quit"

done < students >/dev/null 2>&1

echo -e "1.insert by keyboard\n2.insert by files\n3.quit"

elif [ "$var" == "quit" ]

then

break

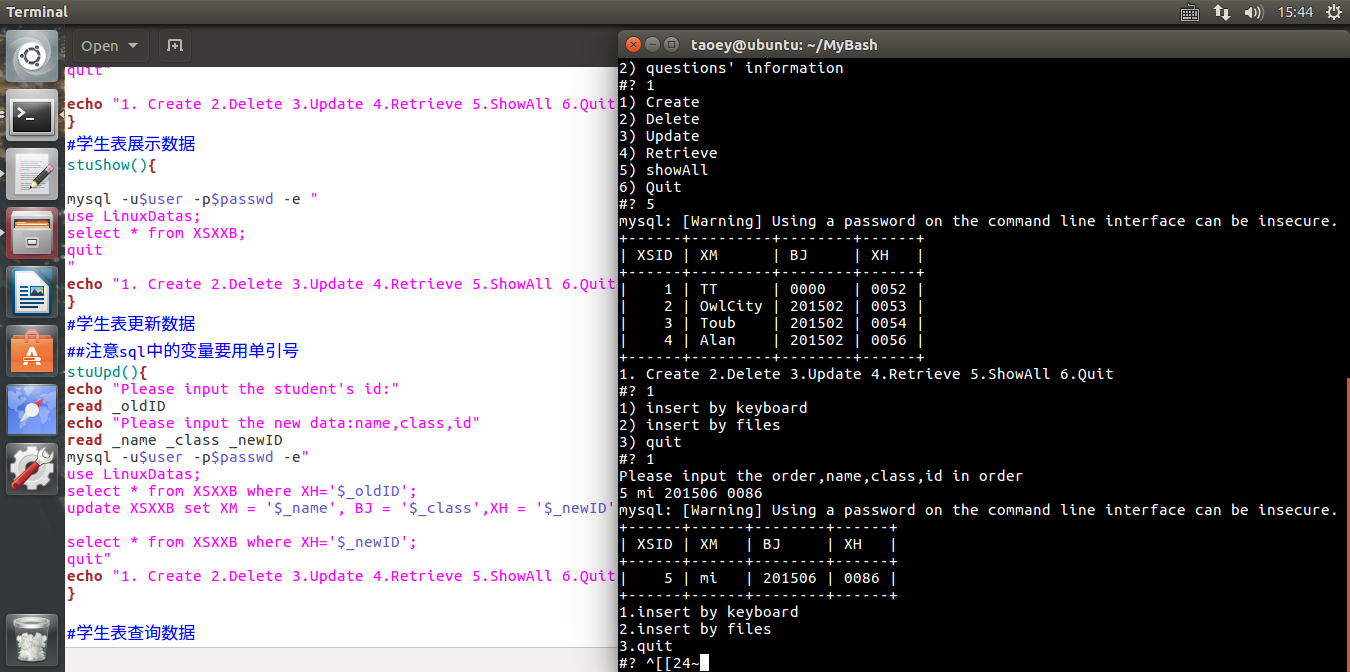
else echo "Please input the right operation number"

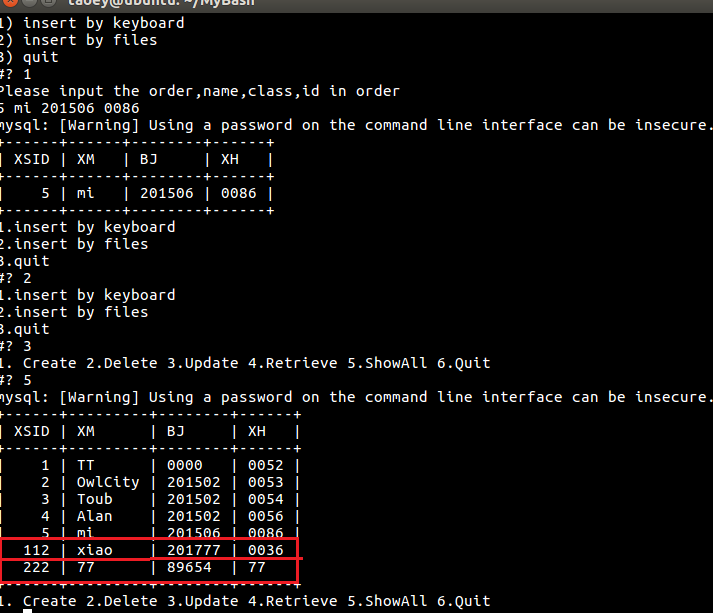
fi

done

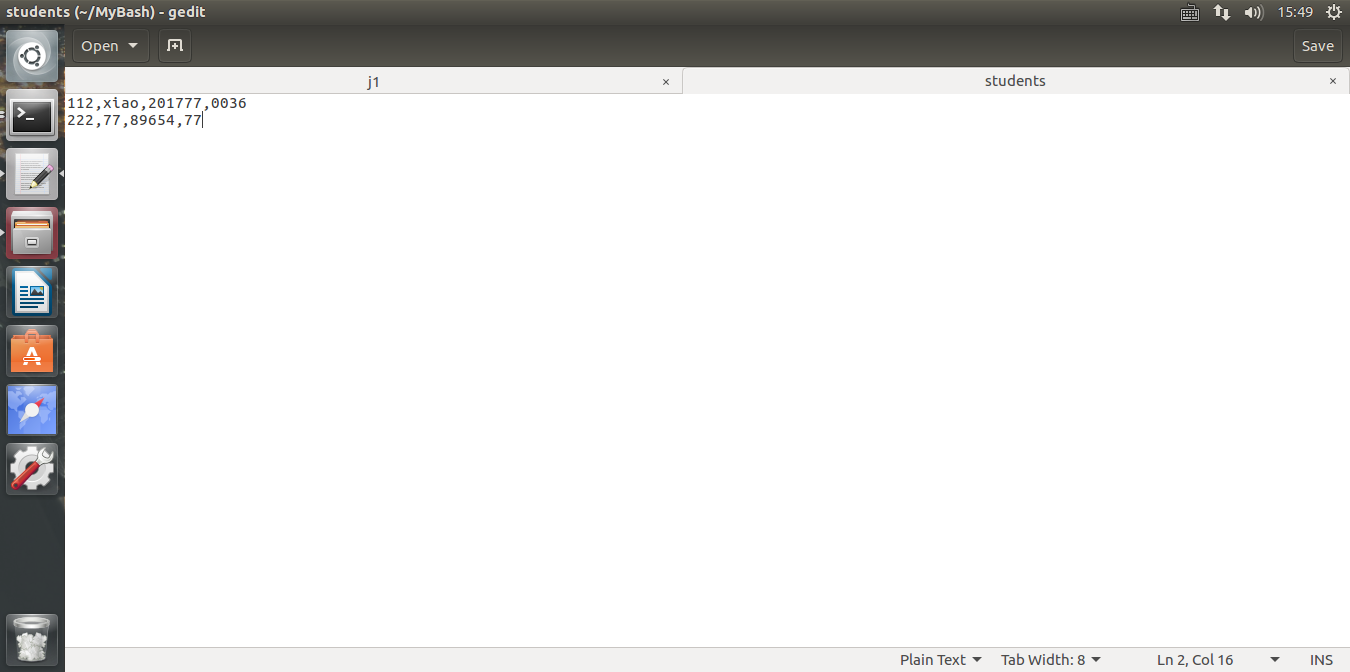
echo "1. Create 2.Delete 3.Update 4.Retrieve 5.ShowAll 6.Quit"

}





Students中的数据



**3.1.3 学生信息表数据的删除函数**

stuDel(){

echo "please input the student's id:"

read \_id

mysql -u$user -p$passwd -e "

use LinuxDatas;

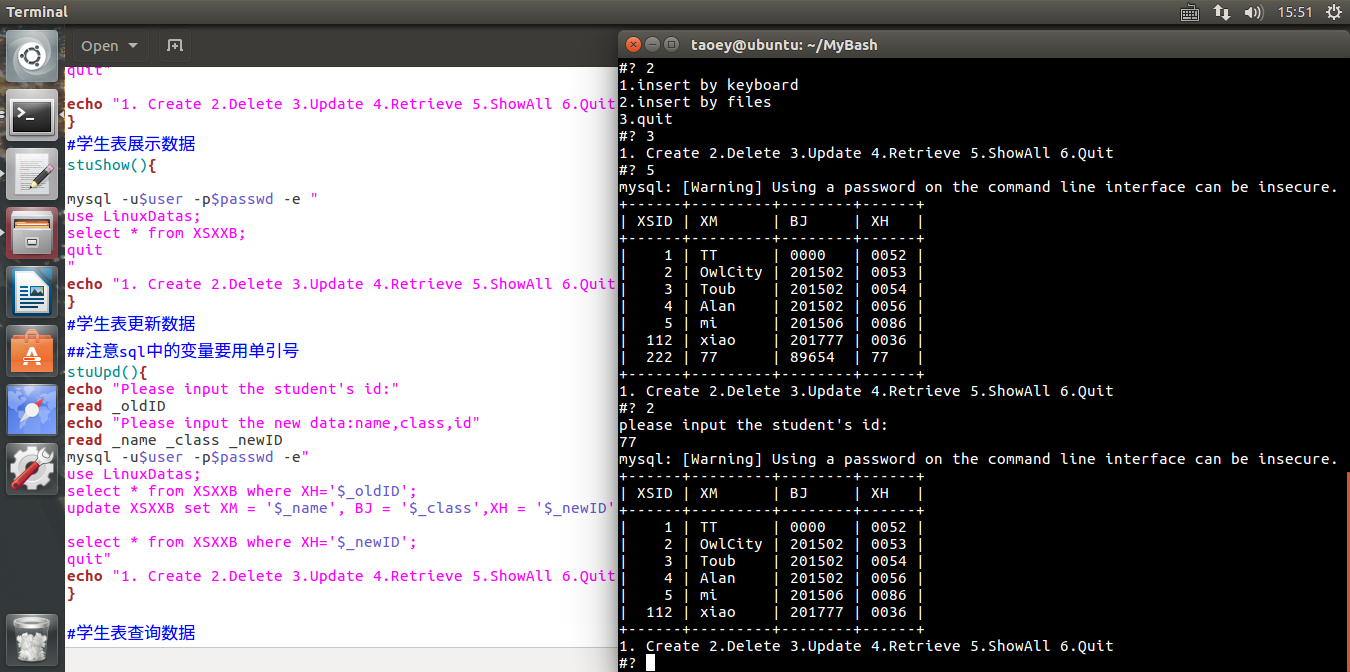
delete from XSXXB where XH = '$\_id';

select \* from XSXXB;

quit"

echo "1. Create 2.Delete 3.Update 4.Retrieve 5.ShowAll 6.Quit"

}



**3.1.4 学生信息表数据的更新函数**

stuUpd(){

echo "Please input the student's id:"

read \_oldID

echo "Please input the new data:name,class,id"

read \_name \_class \_newID

mysql -u$user -p$passwd -e"

use LinuxDatas;

select \* from XSXXB where XH='$\_oldID';

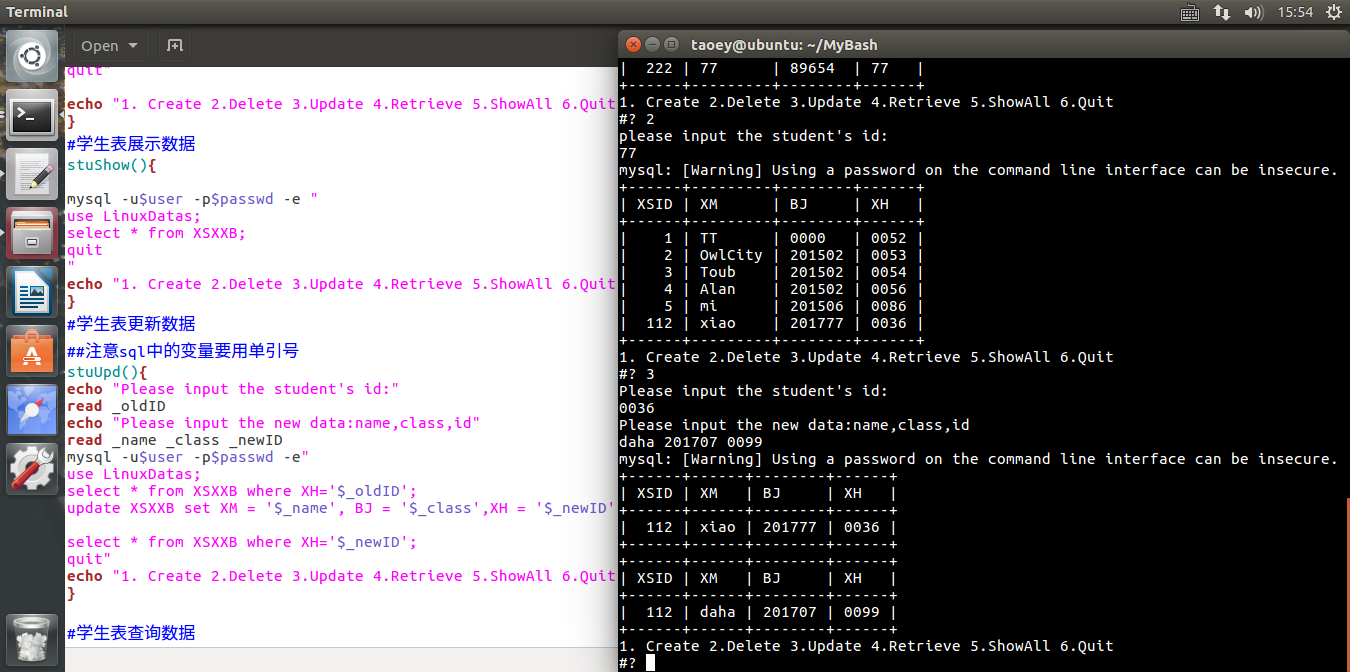
update XSXXB set XM = '$\_name', BJ = '$\_class',XH = '$\_newID' where XH = '$\_oldID';

select \* from XSXXB where XH='$\_newID';

quit"

echo "1. Create 2.Delete 3.Update 4.Retrieve 5.ShowAll 6.Quit"

}



**3.1.5 学生数据的查询函数**

可以根据已知的学生任意部分信息进行查询，然后将查询结果展示到交互界面

stuRet(){

echo "Please select the student's information what you know:"

select var in "name" "class" "id" "quit";

do

if [ "$var" == "name" ]

then

echo 'Please input the name'

read name

echo "The student's information is as follows:"

mysql -u$user -p$passwd -e "

use LinuxDatas;

select \* from XSXXB where XM='$name';

quit

"

echo "1.name 2.class 3.id 4.quit"

elif [ "$var" == "class" ]

then

echo 'Please input the class'

read class

echo "The student's information is as follows:"

mysql -u$user -p$passwd -e "

use LinuxDatas;

select \* from XSXXB where BJ='$class';

quit

"

echo "1.name 2.class 3.id 4.quit"

elif [ "$var" == "id" ]

then

echo 'Please input the id'

read id

echo "The student's information is as follows:"

mysql -u$user -p$passwd -e "

use LinuxDatas;

select \* from XSXXB where XH='$id';

quit

"

echo "1.name 2.class 3.id 4.quit"

elif [ "$var" == "quit" ]

then

break

else

echo "Please input the right operation number"

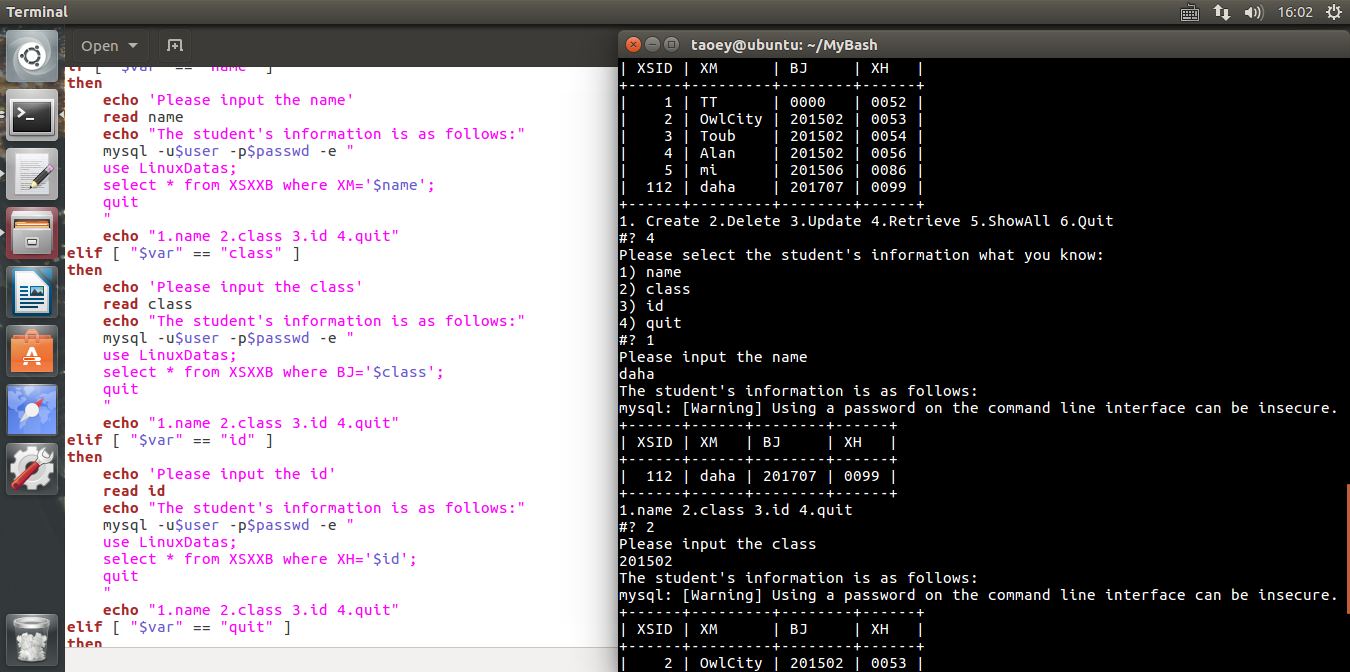
echo "1.name 2.class 3.id 4.quit"

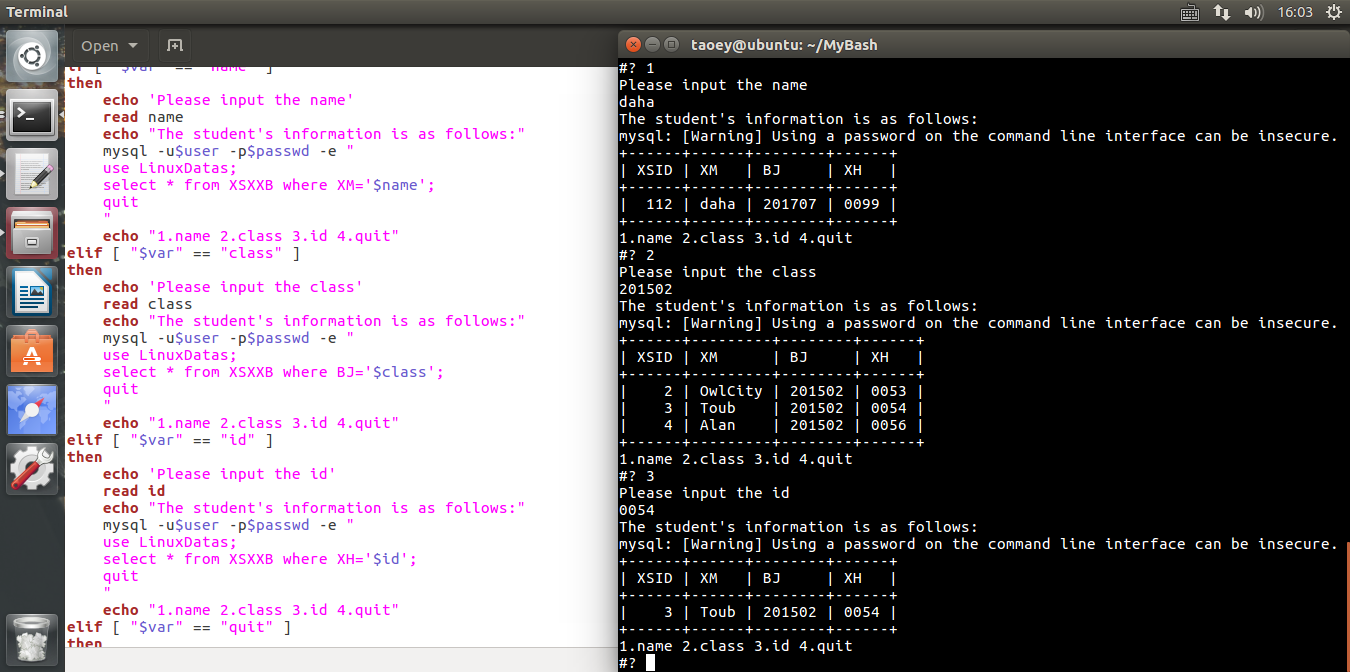
fi

done

echo "1. Create 2.Delete 3.Update 4.Retrieve 5.ShowAll 6.Quit"

}





**3.1.6 学生数据的展示函数**

展示全部学生数据：

stuShow(){

mysql -u$user -p$passwd -e "

use LinuxDatas;

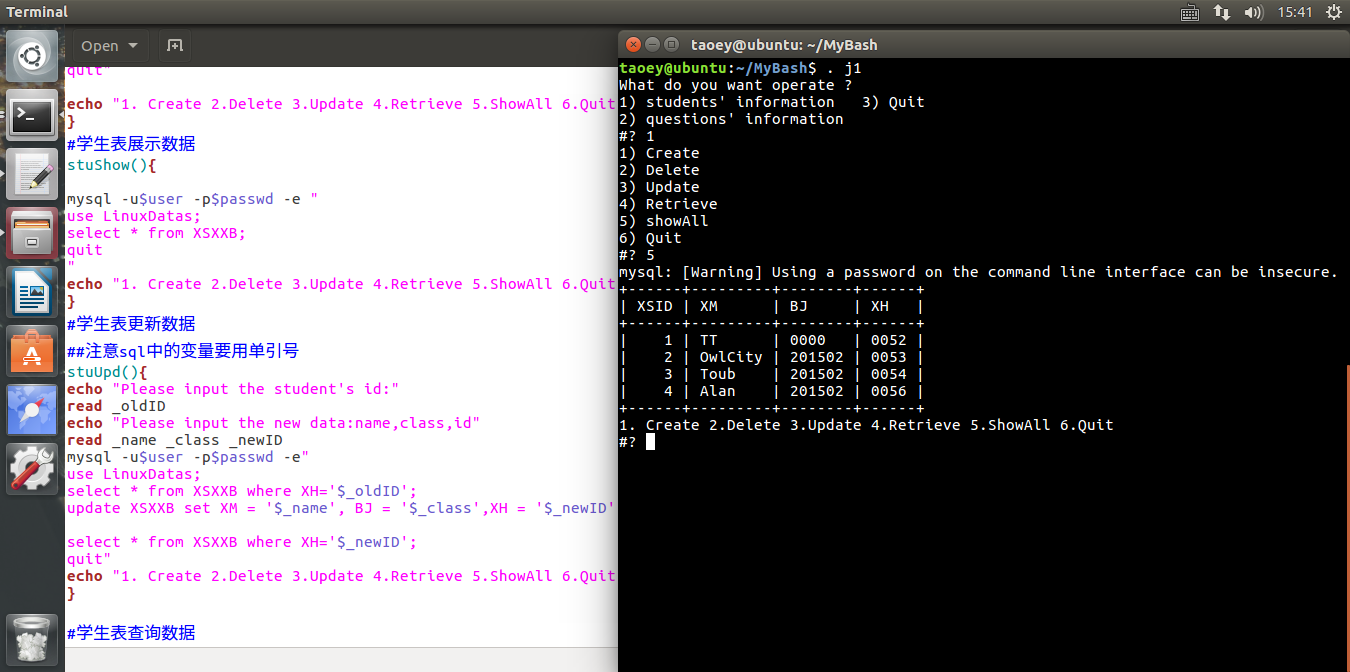
select \* from XSXXB;

quit

"

echo "1. Create 2.Delete 3.Update 4.Retrieve 5.ShowAll 6.Quit"

}



**3.1.7 主界面程序**

主要采用select语句，配合每个函数的echo语句，组成比较友好的交互界面

###############

#主程序

###############

echo "What do you want operate ?"

select var in "students' information" "questions' information" "Quit";

do

if [ "$var" == "students' information" ]

then

select stu in "Create" "Delete" "Update" "Retrieve" "showAll" "Quit";

do

if [ "$stu" == "Create" ]

then

stuCre

elif [ "$stu" == "Delete" ]

then

stuDel

elif [ "$stu" == "Update" ]

then

stuUpd

elif [ "$stu" == "Retrieve" ]

then

stuRet

elif [ "$stu" == "showAll" ]

then

stuShow

elif [ "$stu" == "Quit" ]

then

echo -e "1.students' information\n2.questions' information\n3.Quit\n"

break

else echo "Please input the right operation number"

fi

done

elif [ "$var" == "questions' information" ]

then

select que in "Create" "Delete" "Update" "Retrieve" "showAll" "Quit";

do

if [ "$que" == "Create" ]

then

queCre

elif [ "$que" == "Delete" ]

then

queDel

elif [ "$que" == "Update" ]

then

queUpd

elif [ "$que" == "Retrieve" ]

then

queRet

elif [ "$que" == "showAll" ]

then

queShow

elif [ "$que" == "Quit" ]

then

echo -e "1.students' information\n2.questions' information\n3.Quit\n"

break

else echo "Please input the right operation number"

fi

done

elif [ "$var" == "Quit" ]

then

break

else

echo "Please input the right operation number"

fi

done

**3.2 j2程序设计**

**3.2.1 程序定时扫描**

为实现程序定时扫描文档的功能，利用永真循环和sleep语句实现简单的程序测试

其程序的基本框架为：

#！/bin/bash

while true

do

#主体程序

sleep 60

done

**3.2.2 程序拷贝和运行**

程序的拷贝主要依靠 cp命令，.c文件的运行靠make语句调用程序运行目录下的Makefile文件

Makefile文件编译.c的代码为：

temp:temp.c

gcc temp.c 2>errfile

clean:

rm temp \*.o

其中如果程序运行出错，会将错误信息写入同文件目录下的errfile文件中。之后会在主程序中判断errfile的大小，如果保持0kb则程序运行成功，反之运行失败。判断完一个.c程序之后会将errfile文件置空。

**3.2.3 程序运行时间的测量**

程序运行时间的测量只要通过一个自行编写的函数实现，其主要内容为：

function getTiming() {

start=$1

end=$2

start\_s=$(echo $start | cut -d '.' -f 1)

start\_ns=$(echo $start | cut -d '.' -f 2)

end\_s=$(echo $end | cut -d '.' -f 1)

end\_ns=$(echo $end | cut -d '.' -f 2)

time=$(( ( 10#$end\_s - 10#$start\_s ) \* 1000 + ( 10#$end\_ns / 1000000 - 10#$start\_ns / 1000000 ) ))

echo "$time"

}

该函数的调用：

start=$(date +%s.%N)

make >run 2>&1

end=$(date +%s.%N)

#获取运行时间

runTime=`getTiming $start $end`

**3.2.4 运行结果的提交**

主要通过shell 和数据库进行交互以编译失败为例：

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','BYerr');

quit

" >run 2>&1 #编译错误

**3.2.5 正确的.c程序**

如何判断一个程序的运行结果是否正确？在编译成功的基础上本程序通过对in文件夹中的数据进行输入，每一次和.out 文件夹中的内容进行对比，然后把对比结果写入dif文件中，当把每一个.in文件对比结束后，如果dif的文件大小仍然为0，就说明该程序运行正确。

核心代码：

for inFile in `ls`

do

ID=`echo $inFile | cut -d . -f 1`

MyOutput=`$testpath/a.out < $inFile`

echo $MyOutput > $TMpath/$TM/run

echo $TMpath/$TM/out/$ID.out

#和.out文件对比

diff $TMpath/$TM/run $TMpath/$TM/out/$ID.out >>$TMpath/$TM/dif

#清空相关文件

: >$TMpath/$TM/run

done

**J2程序的主要代码为：**

#!/bin/bash

myfile="cxpcxt/cxpc/\*.c"

mypath="`pwd`/cxpcxt/cxpc/" #程序存放目录

TMpath="`pwd`/cxpcxt"

testpath="`pwd`/cs" #程序测试目录

opath=`pwd` #shell 目录

#时间测试

# arg1=start, arg2=end, format: %s.%N

function getTiming() {

start=$1

end=$2

start\_s=$(echo $start | cut -d '.' -f 1)

start\_ns=$(echo $start | cut -d '.' -f 2)

end\_s=$(echo $end | cut -d '.' -f 1)

end\_ns=$(echo $end | cut -d '.' -f 2)

time=$(( ( 10#$end\_s - 10#$start\_s ) \* 1000 + ( 10#$end\_ns / 1000000 - 10#$start\_ns / 1000000 ) ))

echo "$time"

}

##########################

#主程序

##########################

while true

do

if ls $myfile >/dev/null 2>&1; #目录下有.c文件

then

echo "runing..."

cd $mypath

for filename in `ls`

do

TM=`echo $filename | cut -d . -f 2`

XS=`echo $filename | cut -d . -f 1`

TJSJ=`date +%Y-%m-%d`

#复制到测试目录

cp $mypath/$filename $testpath/temp.c #-f 强制复制

#判断文件内存大小

tempSize=`du -b $testpath/temp.c | awk '{print $1}'`

TMSize=`cat $TMpath/$TM/$TM.cstm | grep size | cut -d : -f 2`

if [ $tempSize -gt $TMSize ]

then

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','DXerr');

quit

" >/dev/null 2>&1 #大小错误

else

cd $testpath

#调用makefile文件编译

start=$(date +%s.%N)

make >run 2>&1

end=$(date +%s.%N)

#获取运行时间

runTime=`getTiming $start $end`

TMTime=`cat $TMpath/$TM/$TM.cstm | grep time | cut -d : -f 2`

if [ $runTime -gt $TMTime ]

then

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','SJerr');

quit

" >/dev/null 2>&1 #时间错误

else

#获取errfile文件大小判断是否编译错误

size=`du -b errfile | awk '{print $1}'`

if [ 0 -eq $size ]

then

cd $TMpath/$TM/in

for inFile in `ls`

do

ID=`echo $inFile | cut -d . -f 1`

MyOutput=`$testpath/a.out < $inFile`

echo $MyOutput > $TMpath/$TM/run

echo $TMpath/$TM/out/$ID.out

#和.out文件对比

diff $TMpath/$TM/run $TMpath/$TM/out/$ID.out >>$TMpath/$TM/dif

#清空相关文件

: >$TMpath/$TM/run

done

difSize=`du -b $TMpath/$TM/dif | awk '{print $1}'`

if [ ! $difSize -eq 0 ]

then

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','YXerr');

quit

" >/dev/null 2>&1 #运行错误

else

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','pass');

quit

" >/dev/null 2>&1

fi

#总体对比结束,#清空相关文件

rm "$testpath/a.out" "$testpath/temp.c"

: > $TMpath/$TM/dif

else

mysql -uroot -p123456 -e "

use LinuxDatas;

insert into TJGCB(XSI,TJSJ,TMID,PCJG)values('$XS','$TJSJ','$TM','BYerr');

quit

" >run 2>&1 #编译错误

: >errfile

fi

fi

fi

done

else echo "no file"

fi

#sleep 60

cd $opath

break

Done

**参考文献：**

1. 孟庆昌，牛欣源. Linux教程 第四版，北京：电子工业出版社
2. 鸟哥 ，鸟哥的Linux私房菜基础学习篇第三版 北京：人民邮电出版社