

# 程序编译运行步骤及结果简单展示

## 一、功能介绍

### 1. 总体功能介绍

源码目录结构：



总体功能图：`ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/tmp$ ./stu`

```
=====
欢迎使用学生成绩管理系统(链表练习版)
请输入如下指令实现相关操作：
a. 创建指定大小的空白链表；
b. 添加学生信息及成绩到空白链表；
c. 增加链表；
d. 插入链表；
e. 删除链表中学生信息及成绩；
f. 查询链表中学生信息及成绩；
g. 修改链表中学生信息及成绩；
h. 输出链表中学生信息及成绩；
i. 保存链表中全部学生信息及成绩到文件；
j. 下载文件中全部学生信息及成绩到链表；
k. 保存链表中全部学生信息及成绩到数据库；
l. 下载数据库中全部学生信息及成绩到链表；
m. 显示指令菜单栏；
n. 释放并退出；
=====
```

该学生成绩管理系统旨在练习链表操作，因此链表操作为总体功能中的一大块，链表中的数据需要存储，所以这里又提供了两大块内容，将链表数据存储到文件以及将链表数据存储到数据库，能存储当然需要能取出，所以在操作上分为三大块：链表操作、文件操作以及 mysql 数据库操作。功能上可以划分为五大块，链表处理、链表数据保存到文件、将文件内容载入到链表、链表数据保存到数据库以及载入数据库数据到链表。这样的话，其实可以将文件内容载入到链表，然后将链表内容保存到数据库，或者将数据库内容载入到链表，再将链表内容保存到文件。

网页部分仅提供按姓名查询功能，将查询结果以表格形式显示出来。

## 2.各个功能介绍

### (1) 创建指定大小的空白链表和添加学生信息及成绩到空白链表（a 和 b 指令）

这两个指令实际上是要一起使用的，图中的 a 指令仅仅按照要求创建一部分没有数据的空链表，b 指令则可以在这些链表中输入数据，使链表中有数据，否则毫无意义。

### (2) 添加和插入链表（c 和 d 指令）

添加是在链表尾部添加有数据的链表，插入则是根据 id 比较，按顺序插入有数据的链表。

### (3) 删除链表、修改链表以及查询链表以及输出链表信息（e、f、g 和 h 指令）

删除、修改和查询都是在学号的基础上进行的，即按照学号删除以及查询，修改时也要先查到学号，然后将输入的内容全部替换以达到修改的目的。输出链表信息，很多时候就是验证一下相关的指令是否成功。

### (4) 保存数据到文件以及载入文件数据到链表(i 和 j)

将链表中的数据保存到文件中，当然，文件大小是固定的，当达到一定大小后就会覆盖之前的内容。载入文件数据到链表即将文件中的数据读到链表中。

### (5) 保存数据到数据库以及载入数据库数据到链表（k 和 l）

同样，数据库操作和文件操作类似，处理数据的载体不同而已。

## 二、编译及运行

### 1.编译及简单运行可执行程序

编译提供两种，cmake 生成 makefile 来编译或者执行 scripts 下的 shell 脚本来编译，使用 cmake 是考虑到其良好的可移植性，当代码跨平台跨系统时只需要简单更改 CMakeLists 文件即可再次编译，使用 shell 脚本则是考虑该编译脚本的书写及使用的简单性。

两种方式使用的 mysql 头文件和库文件不同，shell 脚本使用手动编译生成的 mysql 头文件和库文件，而 cmake 则使用的是 apt-get 安装的 mysql 的系统头文件和库文件，所以在编译前需要更改源码中的头文件，分别在 database\_process.c 和 mysql\_fun.c 文件中，当使用 shell 脚本编译时采用 mysql\_include/mysql.h 而使用 cmake 时使用 mysql.h，这里如果错误的话，由于 mysql 版本的不同，造成生成的可执行程序在运行时会出现段错误。默认使用 shell 脚本编译，因为该项目较小，shell 脚本使用简单，更改简单。

其实在这个项目中使用系统自带的 mysql.h 最好，这里说明手动编译的 mysql 头文件和库文件主要用在交叉编译时，因为在交叉编译时编译是在 PC 机上进行的，如果链接 PC 机系统的头文件和库文件，由于 mysql 版本的差异，生成的可执行程序势必无法在目标机上正常运行。

shell 脚本：在 scripts 目录下直接运行 shell 脚本即可，编译成功后会直接进入 tmp 目录下，然后运行该目录下的可执行程序即可，shell 脚本默认使用手动编译的头文件和库文件，所以使用 shell 脚本时需要更改源码中头文件或更改 shell 脚本中链接的头文件和库文件位置。

```
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/tmp$ cd ../scripts/
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/scripts$ source compile.sh
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/tmp$ ./stu
=====
欢迎使用学生成绩管理系统(链表练习版)
请输入如下指令实现相关操作：
a.创建指定大小的空白链表；
b.添加学生信息及成绩到空白链表；
c.增加链表；
d.插入链表；
e.删除链表中学生信息及成绩；
f.查询链表中学生信息及成绩；
g.修改链表中学生信息及成绩；
h.输出链表中学生信息及成绩；
i.保存链表中全部学生信息及成绩到文件；
j.下载文件中全部学生信息及成绩到链表；
k.保存链表中全部学生信息及成绩到数据库；
l.下载数据库中全部学生信息及成绩到链表；
m.显示指令菜单栏；
n.释放并退出；
=====
```

结果：

```
j.下载文件中全部学生信息及成绩到链表；
k.保存链表中全部学生信息及成绩到数据库；
l.下载数据库中全部学生信息及成绩到链表；
m.显示指令菜单栏；
n.释放并退出；
=====
l
数据库数据下载完毕，如需查看请输出链表，然后也可将数据保存到文件
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1         jack      man       80           90           100
2         jack      man       80           90           100
输出完毕
=====
```

cmake：cmake 链接的头文件和库文件为系统 mysql 的头文件和库文件，分别在 /usr/include/mysql 下和 /usr/lib/mysql 下，所以如果你想要既使用 shell 脚本又使用系统的 mysql.h 的话，更改一下 shell 脚本也可以，将其中链接的头文件和库文件路径改成上面的两个路径即可。

cmake 使用步骤：

## (1)cmake

在 build 目录下 `cmake ..` 以生成 Makefile 脚本。

```
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/src/build$ cmake ..
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/MFC_ODBC/src/build
```

## (2) make

生成可执行程序。

```
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/src/build$ make
Scanning dependencies of target stu
[ 16%] Building C object CMakeFiles/stu.dir/main.c.o
/home/ubuntu/MFC_ODBC/src/main.c: In function 'main':
/home/ubuntu/MFC_ODBC/src/main.c:13:9: warning: variable 'logmask' set but not used [-Wunused-but-set-variable]
     int logmask      = 0;
     ^
[ 33%] Building C object CMakeFiles/stu.dir/link_list.c.o
[ 50%] Building C object CMakeFiles/stu.dir/file_process.c.o
/home/ubuntu/MFC_ODBC/src/file_process.c: In function 'save_data_to_file':
/home/ubuntu/MFC_ODBC/src/file_process.c:55:1: warning: control reaches end of non-void function [-Wreturn-type]
}
[ 66%] Building C object CMakeFiles/stu.dir/database_process.c.o
/home/ubuntu/MFC_ODBC/src/database_process.c: In function 'load_data_from_database':
/home/ubuntu/MFC_ODBC/src/database_process.c:129:12: warning: the address of 'name' will always evaluate as 'true' [-Waddress]
     if(!name)
        ^
/home/ubuntu/MFC_ODBC/src/database_process.c:138:12: warning: the address of 'sex' will always evaluate as 'true' [-Waddress]
     if(!sex)
        ^
/home/ubuntu/MFC_ODBC/src/database_process.c:83:11: warning: variable 'sex_point' set but not used [-Wunused-but-set-variable]
     char* sex_point    = NULL;
        ^
/home/ubuntu/MFC_ODBC/src/database_process.c:81:11: warning: variable 'name_point' set but not used [-Wunused-but-set-variable]
     char* name_point   = NULL;
        ^
[ 83%] Building C object CMakeFiles/stu.dir/mysql_fun.c.o
[100%] Linking C executable /home/ubuntu/MFC_ODBC/tmp/stu
[100%] Built target stu
```

## (3) 去 tmp 目录下运行可执行程序

```

ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/src/build$ cd ../../tmp/
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/tmp$ ./stu
=====
欢迎使用学生成绩管理系统(链表练习版)
请输入如下指令实现相关操作:
a.创建指定大小的空白链表;
b.添加学生信息及成绩到空白链表;
c.增加链表;
d.插入链表;
e.删除链表中学生信息及成绩;
f.查询链表中学生信息及成绩;
g.修改链表中学生信息及成绩;
h.输出链表中学生信息及成绩;
i.保存链表中全部学生信息及成绩到文件;
j.下载文件中全部学生信息及成绩到链表;
k.保存链表中全部学生信息及成绩到数据库;
l.下载数据库中全部学生信息及成绩到链表;
m.显示指令菜单栏;
n.释放并退出;
=====
l
数据库数据下载完毕，如需查看请输出链表，然后也可将数据保存到文件
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1         jack      man       80           90           100
2         jack      man       80           90           100
输出完毕
=====

```

### 三、详细运行展示

#### 1.链表操作

##### (1)创建空白链表及输入输出数据

按顺序操作，否则会出现错误



```

=====
a
请输入要创建的空数据的链表数量：
3
创建完成
=====
b
请输入：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1 jack man 80 90 100
3 jack man 80 90 100
4 jack man 80 90 100
输入完毕
=====

```

## (2) 添加链表

```

c
请输入要添加的链表的内容：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
5 jim man 80 90 100
增加完成
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1      jack      man      80      90      100
3      jack      man      80      90      100
4      jack      man      80      90      100
5      jim      man      80      90      100
输出完毕
=====

```

## (3) 插入链表

```

d
请输入要插入的链表的内容：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
2 jams man 80 90 100
插入完成
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1      jack      man      80      90      100
2      jams      man      80      90      100
3      jack      man      80      90      100
4      jack      man      80      90      100
5      jim      man      80      90      100
输出完毕
=====

```

## (4) 删除链表

```

e
暂只支持按学号删除，且只删除第一个
请输入学号：
1
删除完成
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
2         jans     man       80          90          100
3         jack     man       80          90          100
4         jack     man       80          90          100
5         jim      man       80          90          100
输出完毕
=====
e
暂只支持按学号删除，且只删除第一个
请输入学号：
2
删除完成
=====
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
3         jack     man       80          90          100
4         jack     man       80          90          100
5         jim      man       80          90          100
输出完毕
=====

```

## (5)查询

```

f
暂只支持按学号查找，请输入学号：
6
该学号不存在
查找结束
=====
f
暂只支持按学号查找，请输入学号：
4
查询结果：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
4         jack     man       80          90          100
查找结束
=====

```

```

请输入指令：
q
请输入更新内容：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
6 jine woman 90 90 90
输入的更新内容中学号不存在,更新失败
=====
请输入指令：
q
请输入更新内容：
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
4 jine woman 90 90 90
更新结束
=====
请输入指令：
n
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
3      jack      man      80      90      100
4      jine      woman    90      90      90
5      jim       man      80      90      100
输出完毕
=====

```

## 2.文件操作

文件位置为 doc 目录，文件名为 stu，保存链表数据后退出程序再将数据载入链表查看数据是否保存成功：

```

请输入指令：
i
保存完毕

```



```

ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/tmp$ ./stu
=====
欢迎使用学生成绩管理系统(链表练习版)
请输入如下指令实现相关操作:
a.创建指定大小的空白链表;
b.添加学生信息及成绩到空白链表;
c.增加链表;
d.插入链表;
e.删除链表中学生信息及成绩;
f.查询链表中学生信息及成绩;
g.修改链表中学生信息及成绩;
h.输出链表中学生信息及成绩;
i.保存链表中全部学生信息及成绩到文件;
j.下载文件中全部学生信息及成绩到链表;
k.保存链表中全部学生信息及成绩到数据库;
l.下载数据库中全部学生信息及成绩到链表;
m.显示指令菜单栏;
n.释放并退出;
=====
请输入指令:
j
下载完毕, 如需查看请输出链表
=====
请输入指令:
h

```

学号	姓名	性别	数学成绩	英语成绩	c语言成绩
3	jack	man	80	90	100
4	jine	woman	90	90	90
5	jim	man	80	90	100

```

输出完毕
=====

```

### 3.数据库操作

数据库中已有数据:

```

mysql> select* from class;
+----+-----+-----+-----+-----+-----+-----+
| id | name | sex | math | english | c_language | flag |
+----+-----+-----+-----+-----+-----+-----+
| 1  | jack | man | 80   | 90      | 100       | 0    |
| 2  | jack | man | 80   | 90      | 100       | 0    |
+----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

将其载入链表:

```

=====
请输入指令：
l
数据库数据下载完毕，如需查看请输出链表，然后也可将数据保存到文件
=====
请输入指令：
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
1         jack     man       80           90           100
2         jack     man       80           90           100
输出完毕
=====

```

将文件中已有数据载入链表然后上传到数据库：

```

=====
请输入指令：
j
下载完毕，如需查看请输出链表
=====
请输入指令：
h
学号      姓名      性别      数学成绩      英语成绩      c语言成绩
3         jack     man       80           90           100
4         jine    woman     90           90           90
5         jim     man       80           90           100
输出完毕
=====
请输入指令：
k
保存到数据库完毕
=====

```

```

mysql> select* from class;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | name | sex  | math | english | c_language | flag |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | jack | man  | 80   | 90      | 100       | 0    |
| 2  | jack | man  | 80   | 90      | 100       | 0    |
| 3  | jack | man  | 80   | 90      | 100       | 0    |
| 4  | jine | woman| 90   | 90      | 90        | 0    |
| 5  | jim  | man  | 80   | 90      | 100       | 0    |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

## 四、网页

网页部分由于技术限制仅提供按姓名查询功能，将结果以表格形式显示出来。

数据库必须提前建好，且数据库名为 MFC\_ODBC，表名为 class，用户名为 root，密码为 123，即需要和代码中保持一致。

将 php 目录下的 html 文件和 php 文件放到网站根目录下，即/var/www/html 目录下，然后在浏览器 localhost 登录即可。

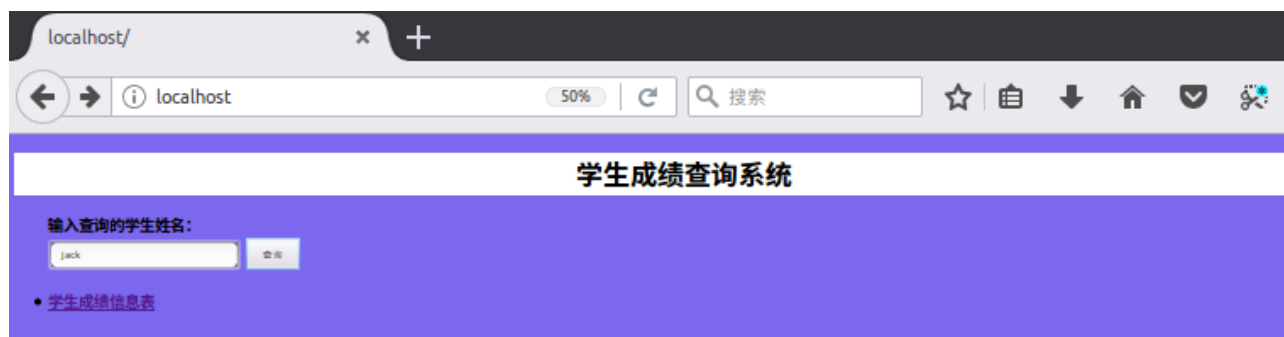
步骤：将 var/www/html 目录下的 index.html 备份，然后将源码 php 目录下的所有文件拷贝到/var/www/html 下，输入 localhost 即可进入主页，然后进行网页操作即可。

```
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/php$ cd /var/www/html/
ubuntu@ubuntu-virtual-machine:/var/www/html$ ls
index.html  phpmyadmin  test.php
ubuntu@ubuntu-virtual-machine:/var/www/html$ sudo mv index.html index.bak
[sudo] ubuntu 的密码:
ubuntu@ubuntu-virtual-machine:/var/www/html$ ls
index.bak  phpmyadmin  test.php
ubuntu@ubuntu-virtual-machine:/var/www/html$
```

```
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/php$ sudo cp * /var/www/html/
[sudo] ubuntu 的密码:
ubuntu@ubuntu-virtual-machine:~/MFC_ODBC/php$ cd /var/www/html/
ubuntu@ubuntu-virtual-machine:/var/www/html$ ls
display_table_class.php  index.html  search.php
index.bak                phpmyadmin  test.php
ubuntu@ubuntu-virtual-machine:/var/www/html$
```

效果如下：

主页：



输入 jack 后查询：

您查找的人员名为: jack  
查询结果:

id	name	sex	math	english	c_language	flag
1	jack	man	80	90	100	0

整表显示：

学生成绩信息表

id	name	sex	math	english	c_language	flag
1	jack	man	80	90	100	0
2	jim	man	90	90	100	0
3	jams	man	90	90	100	0

## 五、尾声

由于项目开发周期较短，所以程序中应该还存在很多 bug，最近我会将源码传到 github 中，欢迎大家提供建议，我会积极改正。