

# 作业管理系统文档

章敏捷, 计算机科学, 3130102283

July 28, 2015

# 目 录

<b>1</b>	<b>设计思路</b>	<b>3</b>
1.1	数据模型 . . . . .	3
1.2	视图 . . . . .	3
1.3	控制器 . . . . .	3
<b>2</b>	<b>详细设计</b>	<b>4</b>
2.1	数据模型部分 . . . . .	4
2.1.1	数据存储 . . . . .	4
2.1.2	数据对象 . . . . .	4
2.2	视图部分 . . . . .	5
2.3	控制器部分 . . . . .	5
<b>3</b>	<b>其他</b>	<b>6</b>
3.1	开发小结 . . . . .	6
3.1.1	版本控制 . . . . .	6
3.1.2	版本迭代 . . . . .	6

# 章节 1

## 设计思路

总体采用MVC设计模式，代码分为模型、视图、控制器三个部分。将模型、视图用控制器连接减少耦合，方便扩展修改。

### 1.1 数据模型

所有数据存取都由数据模型完成。由于要求采用文件存储，不能使用数据库，所以需要新建一套数据存储体系。初步考虑采用类xml的方法结合正则表达式来处理。采用key-value的形式存储数据，单表作为一个文件，文件名为表名。并且在此基础上实现简单的ORM。

### 1.2 视图

保存于用户交互用的代码，所有的输入输出都由视图部分完成。主要完成接受用户输入以及对用户进行反馈。

### 1.3 控制器

通过调用视图来处理用户发送的请求，再调用数据模型完成数据的存取，最后调用视图对用户进行反馈。

## 章节 2

# 详细设计

### 2.1 数据模型部分

#### 2.1.1 数据存储

数据库操作包括在`data.sh`中，主要包括了：

用户:USER id 类型 工号/学号 姓名 PASSWD

课程:COURSE id 教师id 名称

作业:WORK id 课程id 名称

学生-课程关系:U-C id 学生id 课程id

学生-作业关系:S-W id 学生id 作业id 完成情况

为了满足ORM中主键自增的特性，每张k-v表需要自带一个int记录用于生成自增的主键。

#### 2.1.2 数据对象

对象部分的操作包括在`orm.sh`中。

数据库中每张表对应了一个类型的对象，每种对象都有自己的新建、删除、更新方法，还有通过id获取对象实例的方法。

为了简化对象之间的关系调用，在获取对象实例的时候会同时将与对象有唯一对应关系的其他对象也包含在该对象中成为子对象。如：作业对象中包含了其对应的课程信息。

对象通过一个字符串保存，每个字段用`\_`进行分割，有相应的方法将对象展开(unfold)成相应的属性，同时也会解析相应的子对象，

部分特殊的复杂操作需要同时对多表进行操作，因此单独提取作为新的函数，如新建作业时需要同时新建所有选课学生的S-W记录，因此使用了SYNC类函数对该类操作进行同步，以保证数据库结构的完整性，避免出现错误。

## 2.2 视图部分

视图部分包括在`view.sh`中，主要是文字的输入和输出，头部包含了对对象的展开函数，用于展开对象列表。

视图函数通过参数接受控制器信号，通过全局变量作为返回，其中list类视图需要根据不同的用户选择相应的命令提示。

多数视图函数都带有一个可选的msg参数，用于接收错误提示，在当前用户输入有误的情况下可以产生提醒。

## 2.3 控制器部分

控制器代码包括在`main.sh`中。

控制器由一个主循环作为状态机，起始状态要求用户登录，登陆后进入Index主页根据用户的不同权限显示不同的命令提示。

循环中根据不同的用户输入进入不同的状态函数，函数先调用视图接受用户的请求，之后根据需求的不同进入子状态或者完成请求返回主页继续新的请求。

## 章节 3

# 其他

### 3.1 开发小结

#### 3.1.1 版本控制

本轮开发采用git作为源代码管理工具，详细记录了开发的过程。虽然每次提交都有写备注但是并不十分符合标准，在正式开发中需要更加详细的描述每次提交的修改。

#### 3.1.2 版本迭代

第一版的产品仍然有许多不足，在第一次迭代中将进行修补，以优化用户体验。