**密**

**码**

**货**

**币**

**与**

**区**

**块**

**链**

**技**

**术**

**作**

**业**

**报**

**告**

作业名称：简易比特币系统的实现

队名：C-moon

队员姓名：钱非凡、贾潇风、陈恺凡

队长：钱非凡

时间：2019/12/31

# 一、系统设计及实现

## 1.1 成员分工

钱非凡：交易和区块结构设计、脚本等功能的代码实现以及代码的合并和调试

贾潇风：数据库设计、操作数据库代码实现以及代码的合并和调试

陈恺凡：SM2签名、SM3哈希代码

## 1.2 数据库设计

共涉及了四个数据库，其中BLOCK为存放区块的数据库、TXS用于存放未被打包的交易、USER存放用户信息、UTXO存放UTXO：

表1：数据库设计

|  |  |  |
| --- | --- | --- |
| **表名** | **字段名** | **类型** |
| BLOCK | id | Int、主键 |
| BLOCK\_HEX | Text |
| BLOCK\_HEADER\_HASH | Text |
| TXS | Id | Int、主键 |
| TX\_HEX | Text |
| IF\_PACK | Char(1) |
| TX\_HASH | Text |
| USER | USERNAME | Char(30) |
| PASSWORD | Char(100) |
| USER\_PK | Text |
| USER\_SK | Text |
| UTXO | Id | Int、主键 |
| UTXO | Text |
| TX\_HASH | Text |
| IDX | Char(10) |
| OWNER | Text |
| VALUE | Char(20) |
| IF\_USE | Char(1) |

## 1.3 交易结构和区块结构

交易的结构分为交易输入和交易输出，其中交易的输入结构如下：

表2：交易输入结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 大小 | 含义 |
| tx\_id | 32字节 | 交易哈希 |
| idx | 4字节 | 输出索引 |
| lengthOfScriptSig | 8字节 | 脚本长度 |
| scriptSig | 长度不限 | 脚本内容 |

当交易为coinbase时，交易的输入结构如下：

表3：coinbase交易输入结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 大小 | 含义 |
| tx\_id | 32字节 | 交易哈希 |
| idx | 4字节 | 输出索引 |
| lengthOfCoinbase | 8字节 | 脚本长度 |
| coinbase | 长度不定 | 脚本内容 |

交易输出结构如下：

表4：交易输出结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 大小 | 含义 |
| value | 8字节 | 交易金额 |
| lengthOfScriptPubKey | 8字节 | 脚本长度 |
| scriptPubKey | 长度不定 | 脚本内容 |

交易结构如下：

表5：交易结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 大小 | 含义 |
| sum\_tx\_input | 4字节 | 交易哈希 |
| tx\_inputs | 长度不定 | 输出索引 |
| sum\_tx\_output | 4字节 | 脚本长度 |
| tx\_outputs | 长度不定 | 交易输出 |

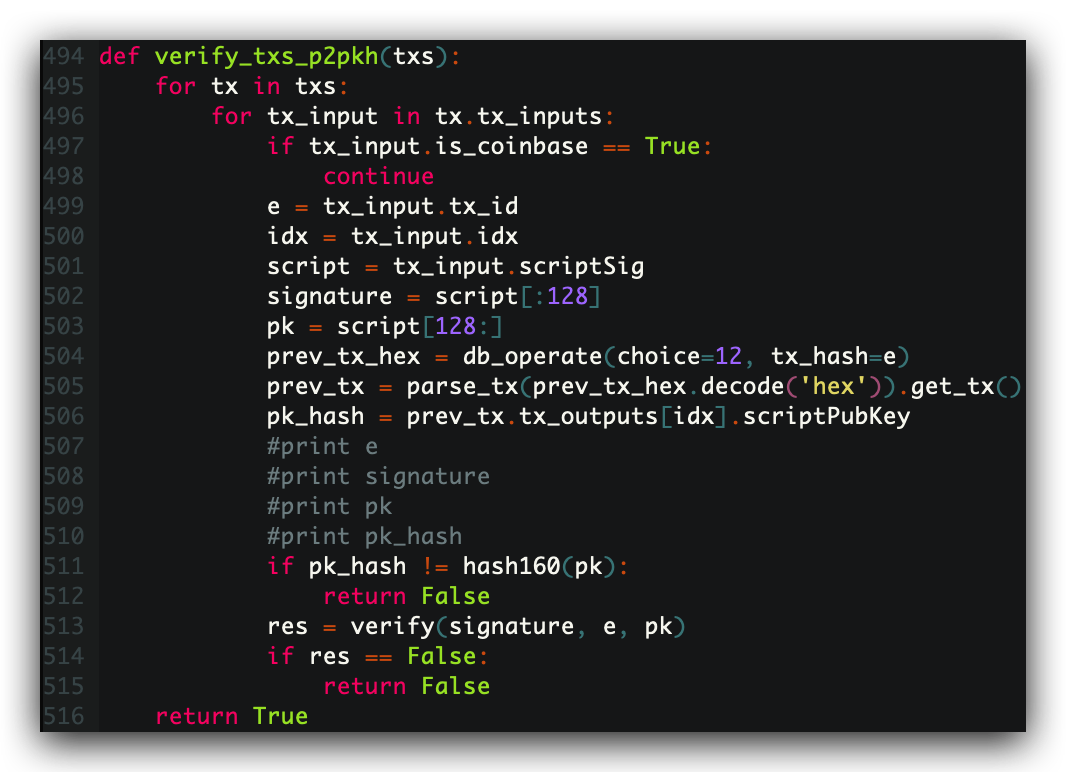
区块结构与比特币类似，其中区块头结构如下：

表6：区块头结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 大小 | 含义 |
| version | 4字节 | 版本号 |
| prev\_hash | 32字节 | 前一区块哈希 |
| merkle\_root | 32字节 | 梅克尔根 |
| timestamp | 4字节 | 时间戳 |
| nbits | 4字节 | 交易难度 |
| nonce | 4字节 | 工作量证明 |

## 1.4 P2PKH

交易输入的scriptSig部分（解锁脚本）中，前半部分为对交易的签名，后半部分为使用者的公钥；scriptPubKey（锁定脚本）中存储了公钥经过了hash160后的20字节信息。对交易的验证先进行对公钥的校验，然后对签名进行校验：



**图1：**P2PKH交易验证

# 二、系统使用

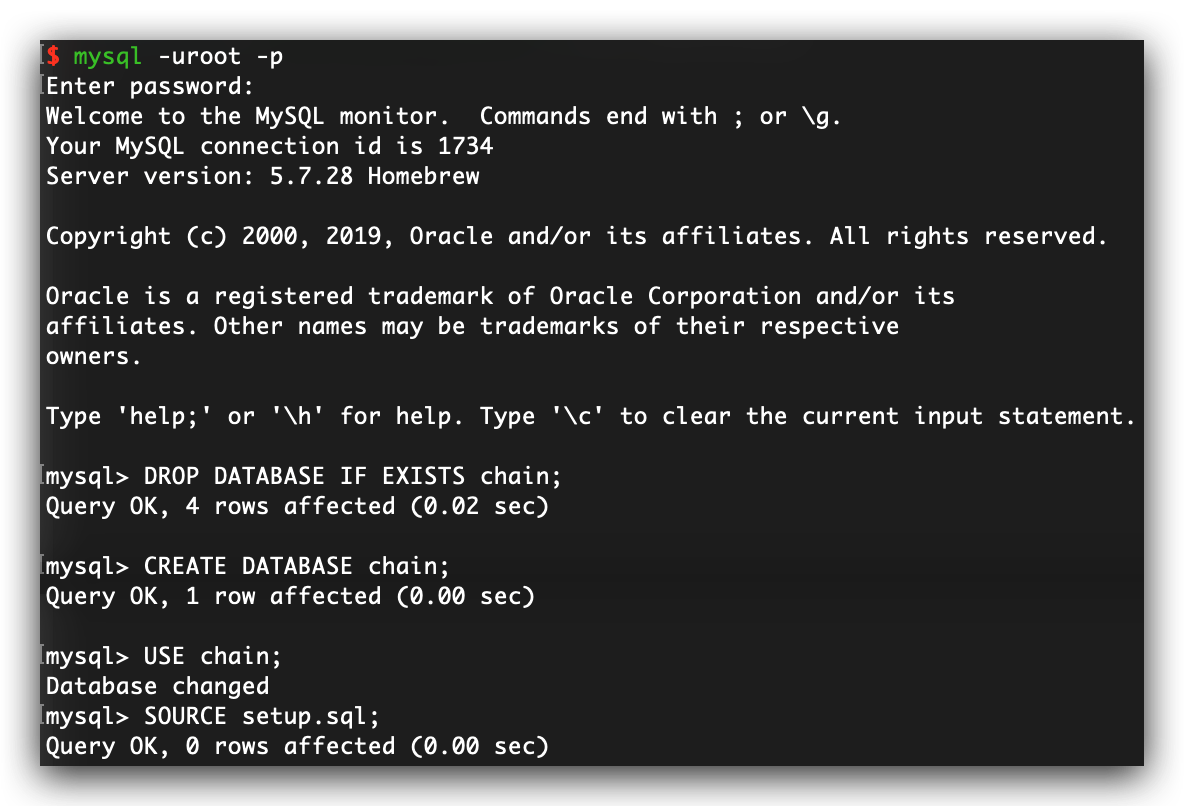
## 2.1 测试环境

系统：macOS Mojave 10.14.4

软件版本：Python 2.7.17（调用的库信息：bitcoin==1.1.42、MySQL-python==1.2.5）、mysql 5.7.28

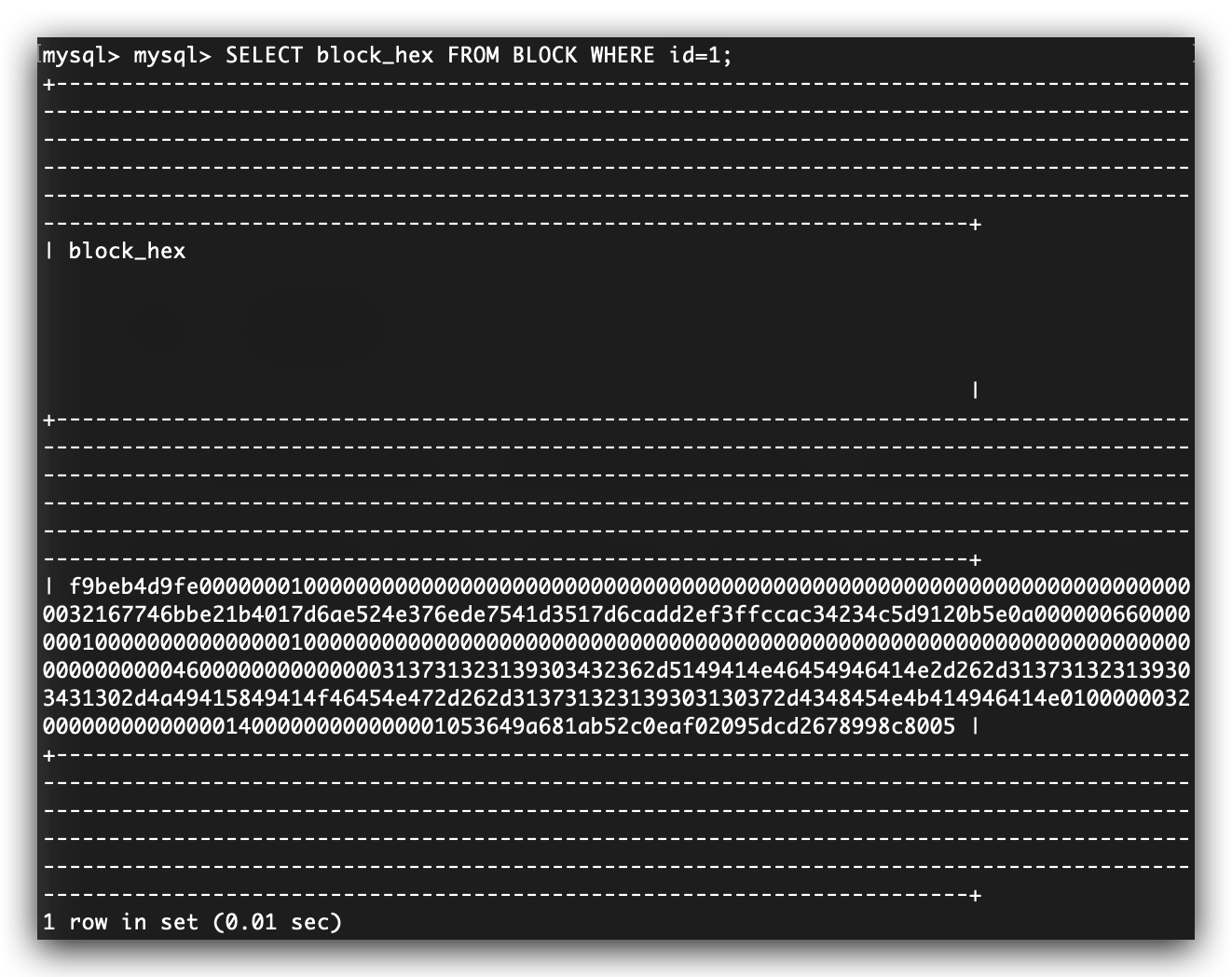
## 2.2 使用说明

附件中存储了一个初始系统的数据（setup.sql），其中有三个用户（admin/admin123、alice/123456、bob/123456），把setup.sql导入：



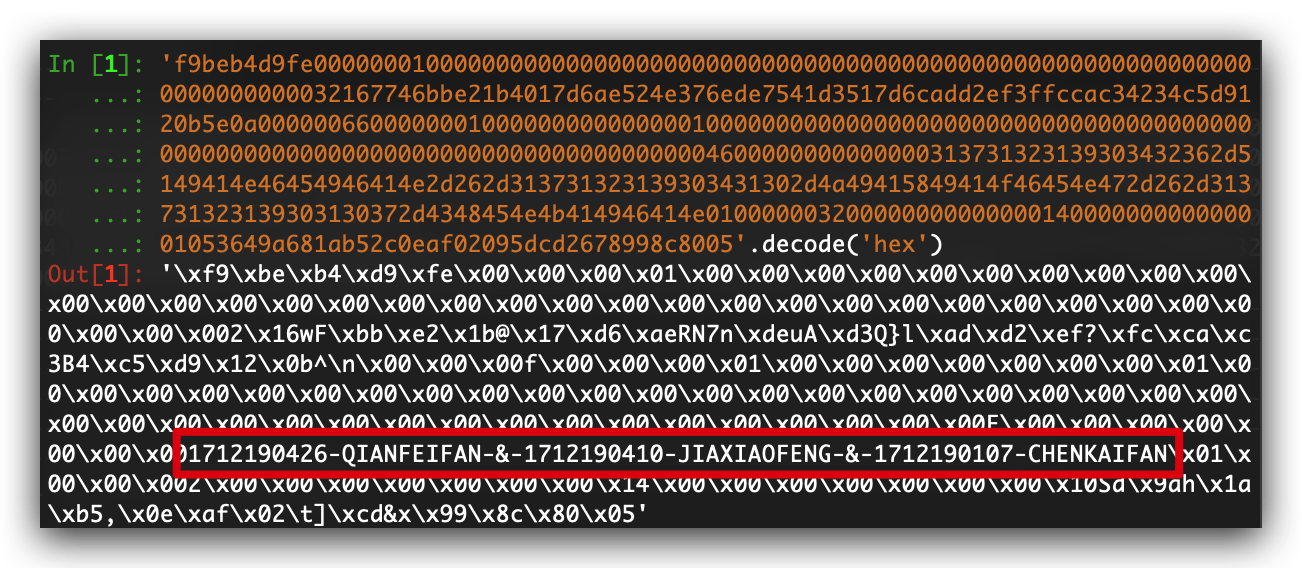
**图2：**导入setup.sql文件

在BLOCK表中，存储了区块的信息，第一个也就是创始区块中存储了相关的信息：



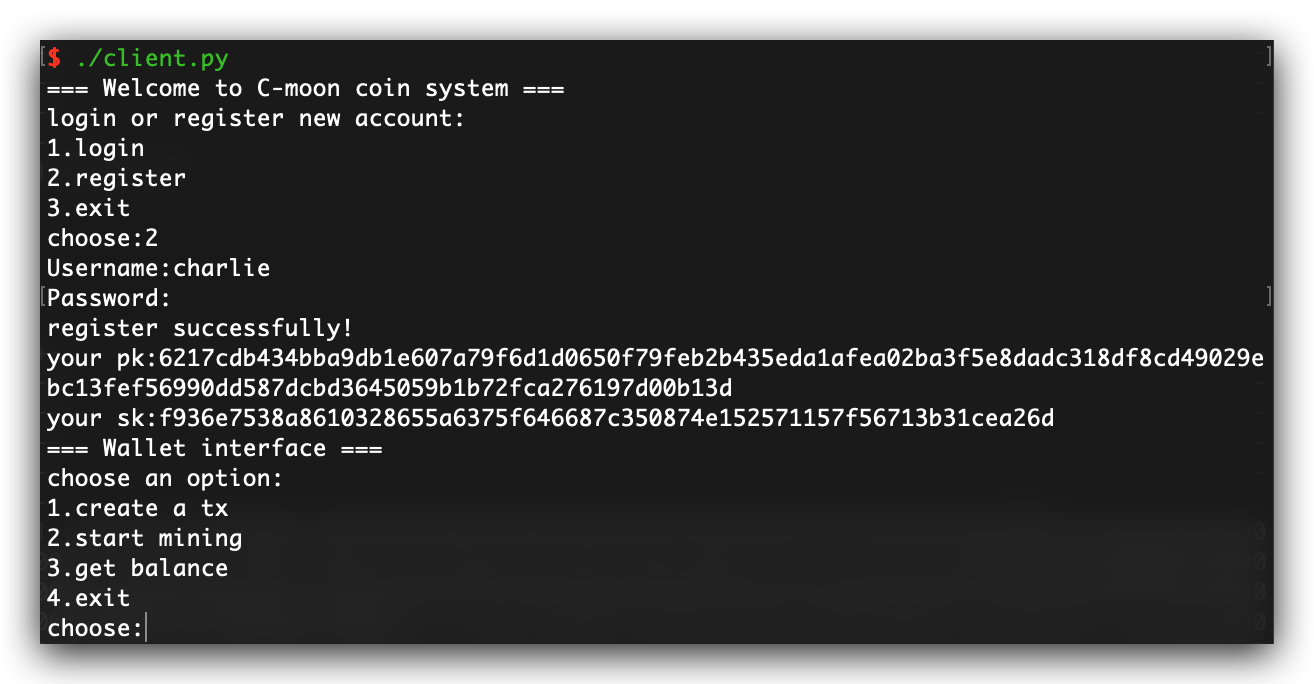
**图3：**查看数据库中的创始区块

将十六进制数据解码后得到字符串，在coinbase中存储了三个人的学号和姓名的拼音：



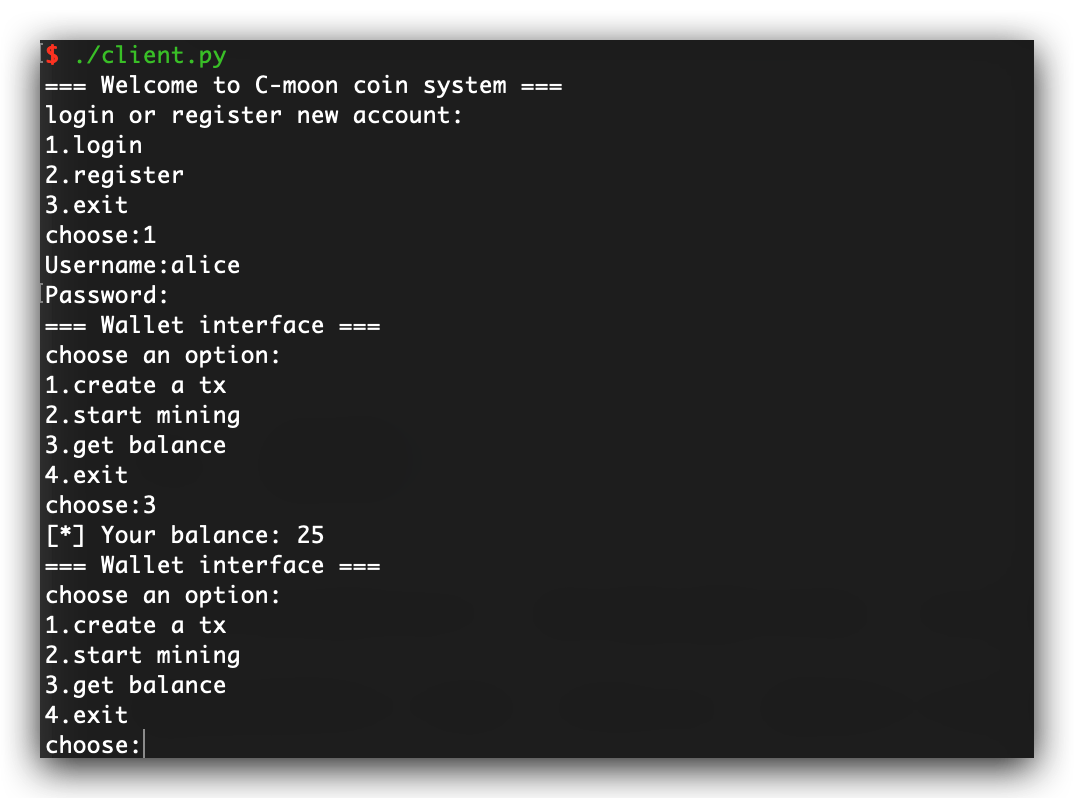
**图4：**查看创始区块解码后的信息

通过client.py进行登录或注册用户。注册一个用户charlie：



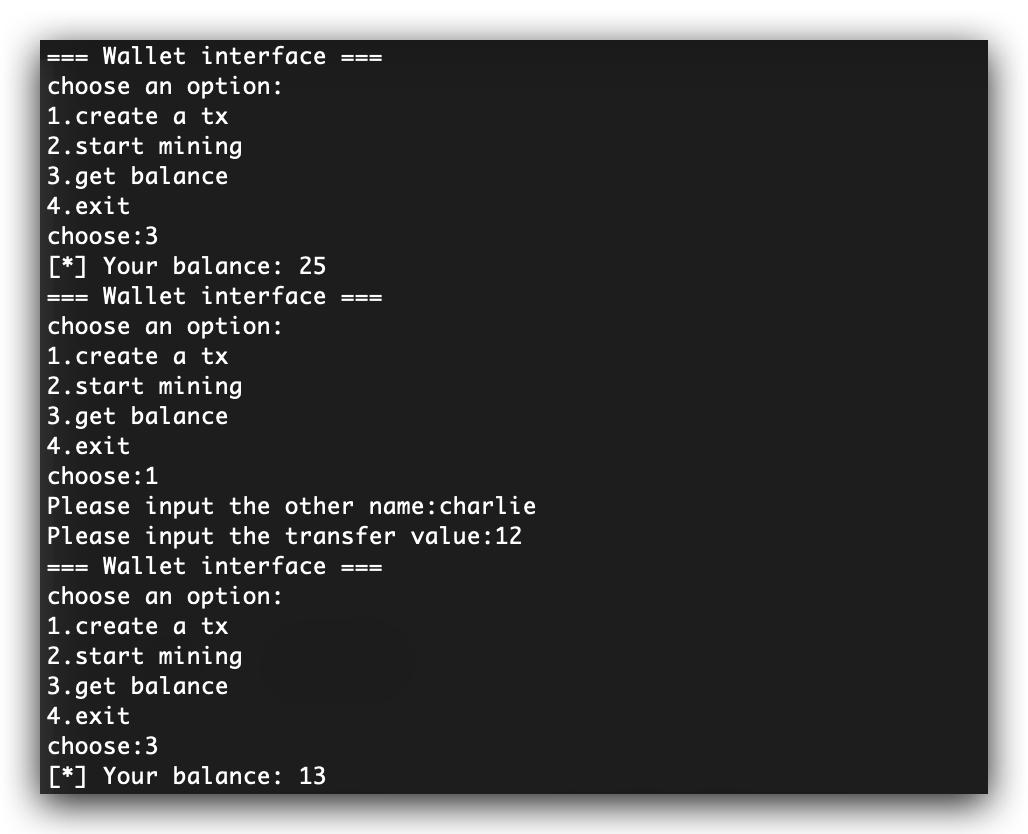
**图5：**注册charlie的账号

同样用client.py登录alice的账号：



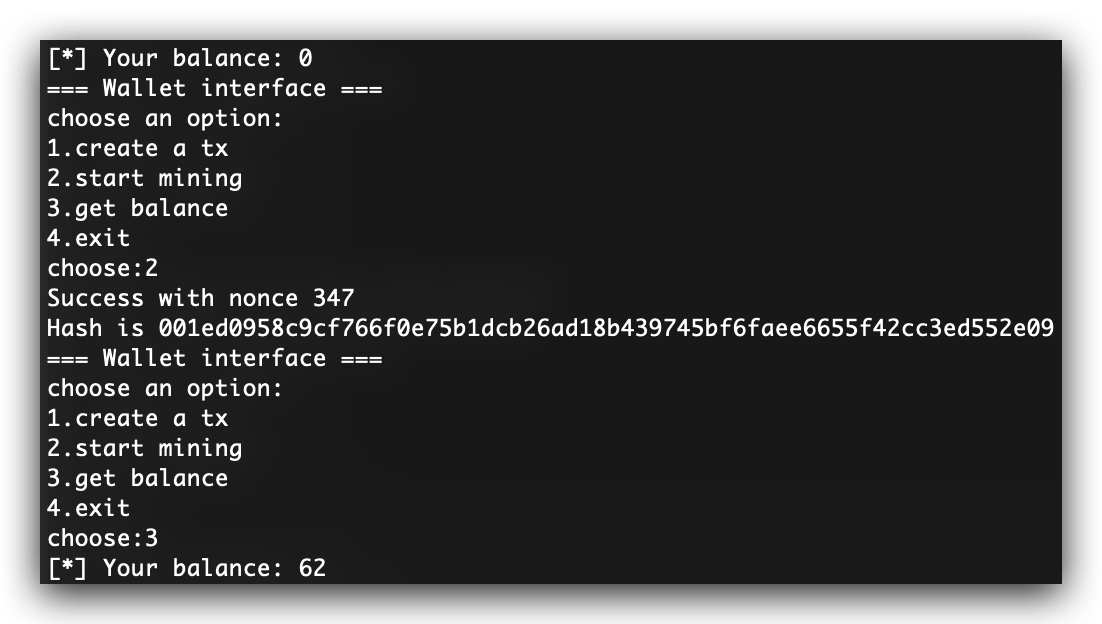
**图6：**登录alice的账号并查看余额

使用alice的账号创建一个向charlie转账12单位金额的交易。原本alice账户的余额为25，在产生交易并确认区块后，余额变为13：



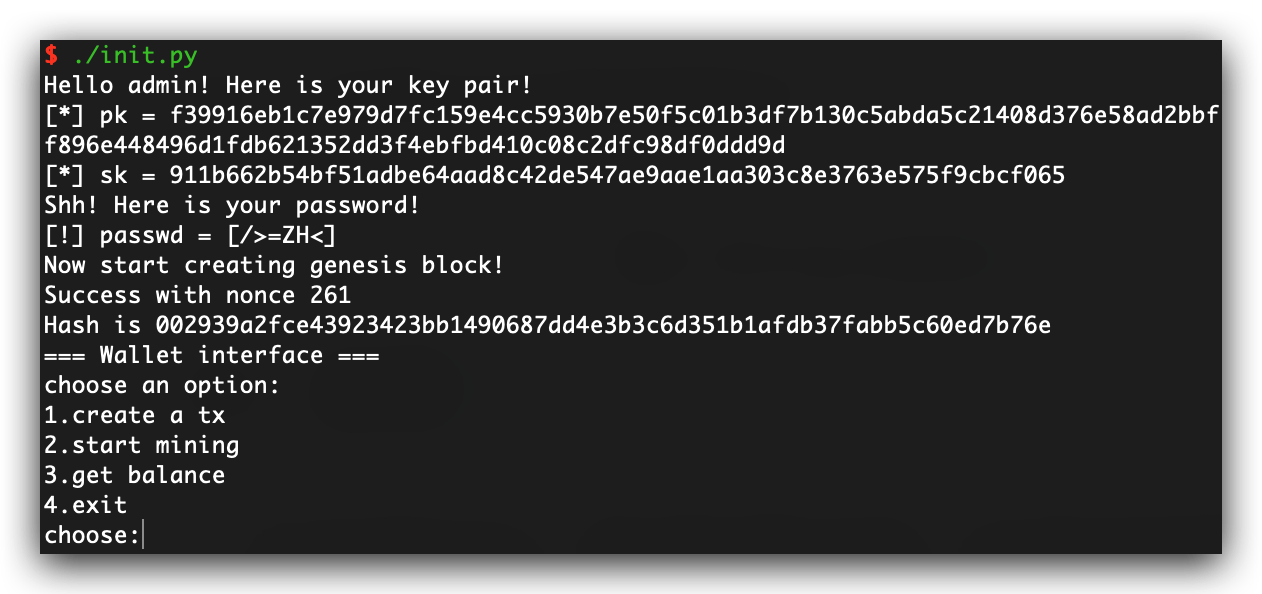
**图7：**alice创建一个向charlie转账12单位金额的交易

使用charlie的账号对这笔交易进行挖矿，挖矿成功后得到奖励费用50以及转账的金额12，即62：



**图8：**charlie对这笔交易交易进行验证、打包并挖矿成功

也可以直接通过init.py进行系统初始化，默认创建出一个admin用户，也同样可以注册其他新用户：



**图9：**使用init.py创建系统

# 三、总结

在本次的作业中，我们通过编写代码，对比特币的系统以及区块链的结构等有了更深一步的认识，感受到了比特币系统设计的精妙之处。