第五周周报

汪子龙

本周用 golang 实现了一个模拟交易系统，用户可以在系统进行充值和提币等操作，下面介绍了系统的功能及实现思路。

用户

一、注册

注册时用户需要绑定一个地址（银行卡），系统从账户地址池中分配一个账户地址（支付宝），并将对应的助记词反馈给用户。

```
please choose your option:
0: log in       1: register     2: log in as admin      3: exit
1
please input your ethereum address:
0xb05c4d5cb3679b1b9e2a3e937806c84c06bb7df7
succeeded to register, your mnemonic is:
bright best hello addict year awful salmon liar hurdle proof else today b
rick rose front dress digital base upon unaware foot gold crucial divorce
```

地址池信息以 json 格式存放在 addresses.txt 中，列表中各键含义如下

address：系统中的账户地址

mnemonic：助记词

status：账户地址状态（-1 为主地址，0 为空闲地址，1 为已分配地址）


```
{
"accounts": [
{
"address": "0xc0093215beC3Cbb9522352DCB4E3fa8fd5b665D1",
"mnemonic": "inmate energy transfer mesh cinnamon educate slide latin veteran excess dance say scissors provide reveal link have caught orphan fruit can clog degree damp",
"status": "-1"
},
{
"address": "0x8bf8fb9Cfd048dE2645539F5007D12Ae6465623B",
"mnemonic": "bright best hello addict year awful salmon liar hurdle proof else today brick rose front dress digital base upon unaware foot gold crucial divorce",
"status": "1"
},
{
"address": "0x297fCA8450553A3C9103Dd410B59e79CDf6F4a0A",
"mnemonic": "bottom theme muscle dune trigger blossom apple wing inform weekend tourist lava rate seminar tribe twist help reduce enlist window reform shy sport update",
"status": "0"
},
```


用户信息存放在 users.txt 中，列表中各键含义如下

address：用户在系统中的账户地址（支付宝）

ethaddress：用户绑定的地址（银行卡）

```
{
"users": [
{
"address": "0x8bf8fb9Cfd048dE2645539F5007D12Ae6465623B",
"ethaddress": "0xb05c4d5cb3679b1b9e2a3e937806c84c06bb7df7"
}
]
}
```


二、登录

用户输入助记词，系统查询 addresses.txt 和 users.txt 文件，获取用户的账户地址（支付宝）和绑定地址（银行卡）；若助记词无记录，则提示用户不存在。

```
please choose your option:
0: log in       1: register     2: log in as admin      3: exit
0
please input your mnemonic:
bright best hello addict year awful salmon liar hurdle proof else today br
ick rose front dress digital base upon unaware foot gold crucial divorce
welcome, 0x8bf8fb9cfd048de2645539f5007d12ae6465623b
```

三、查询余额（交易监控）

每个账户地址的交易信息以 json 格式存放在相应的文本文件中，各键含义如下

balance：上次刷新后 账户的余额

start：上次刷新后 扫描到的区块号

transactions：上次刷新后 所有的充值/提币交易记录


```
{
"balance": "10000000000000000000",
"start": "1962",
"transactions": [
{
"amount": "5000000000000000000",
"from": "0xb05c4d5cb3679b1b9e2a3e937806c84c06bb7df7",
"hash": "0xa0f9bc3a7cf0a041fb3745f22f15412461c8e110d7cf6668ab6683bdbbb3786d",
"to": "0x8bf8fb9cfd048de2645539f5007d12ae6465623b"
},
{
"amount": "5000000000000000000",
"from": "0xb05c4d5cb3679b1b9e2a3e937806c84c06bb7df7",
"hash": "0x71b623a5903978a02433abcee3f4fc1364fb8d82ed19d5889fd7353d63ab1379",
"to": "0x8bf8fb9cfd048de2645539f5007d12ae6465623b"
},
{
"amount": "5000000000000000000",
"from": "0xb05c4d5cb3679b1b9e2a3e937806c84c06bb7df7",
"hash": "0x1cbba028e40e36fd952d39a6296645a32823ded11fb71d7751dcac00ac4a1cf2",
"to": "0x8bf8fb9cfd048de2645539f5007d12ae6465623b"
},
```


用户查询余额时，先读取信息文件，获取上次刷新账户的余额、结束区块号和交易记录。从上次结束的区块号开始扫描直至最新一个区块，获取每个区块上的交易，若交易满足：

1．发起地址是用户的绑定地址，接受地址是用户的账户地址，则记录该交易

为充值，账户余额加上此次交易的面额

2．发起地址是系统主地址，接受地址是用户的绑定地址，则记录该交易为提币，账户余额减去此次交易的面额

结束后将最新的余额、区块号和交易记录保存到相应的文件中。

```
please choose your option:
0: check balance        1: recharge     2: withdraw     3: exit
0
your balance is:
12500000000000000000
```

四、充值

用户提供 keystore 文件路径和口令，输入需要充值的面额，系统生成交易并提交；交易的发起地址为用户的绑定地址，交易的接受地址为用户的账户地址；交易提交成功后，返回此次交易的 hash。

```
please choose your option:
0: check balance        1: recharge     2: withdraw     3: exit
1
please input your key file path:
D:\Projects\GethProgram\data\keystore\UTC--2020-07-31T04-44-41.315101100Z-
-b05c4d5cb3679b1b9e2a3e937806c84c06bb7df7
please input your password:
testuser
please input the value:
500000000000000000
0xa4b56d709ffcdccce595aeefc1e61f0e88f60acf519daa6680c9bcbe8215310d
transaction submitted:0xa4b56d709ffcdccce595aeefc1e61f0e88f60acf519daa6680
c9bcbe8215310d
```

五、提币

用户输入需要提取的面额，系统生成交易并提交；交易的发起地址为系统主地址，交易的接受地址为用户的绑定地址；交易提交成功后，返回此次交易的 hash。

```
please choose your option:
0: check balance        1: recharge     2: withdraw     3: exit
2
please input the value:
500000000000000000
0x95113347a01e15e9fe5481f6d94ab33e3f9a6312d701465c7f6acb07804c79e3
transaction submitted: 0x95113347a01e15e9fe5481f6d94ab33e3f9a6312d701465c7
f6acb07804c79e3
```

管理员

管理员选择以管理员身份登录，输入密码 admin，进入管理系统。

```
please choose your option:
0: log in       1: register     2: log in as admin      3: exit
2
please input admin password:
admin
```

一、回笼

从 users.txt 中获取系统所有用户的信息，对每个账户地址都进行刷新操作，具体机制见用户部分第三节；获取所有账户地址的最新余额后，每个账户地址发起交易，接收方为系统主地址，面额为账户地址的余额。

```
please choose your option:
0: centralize   1: exit
0
refresh accounts done
0x414734d37be14a5ca0a5f97b4a7059e5c4298865ce8b198750fd9f9993f92dba
[0x8bf8fb9Cfd048dE2645539F5007D12Ae6465623B] transaction submitted: 0x4147
34d37be14a5ca0a5f97b4a7059e5c4298865ce8b198750fd9f9993f92dba
```

二、生成账号

随机生成一个助记词，并通过助记词生成账户地址，保存到系统的账户地址池中。