

区块链大作业：热身报告

18308045 Zhengyang Gu

2020 年 11 月 27 日

目录

1 使用已有的开源区块链系统FISCO-BCOS，完成私有链的搭建以及新节点的加入。（截图说明搭建流程）	3
1.1 单群组FISCO BCOS联盟链的搭建	3
1.1.1 准备环境	3
1.1.2 搭建单群组4节点联盟链	4
1.1.3 启动FISCO BCOS链	4
1.1.4 检查进程	4
1.1.5 检查日志输出	4
1.2 配置及使用控制台	5
1.2.1 准备依赖	5
1.2.2 启动控制台	6
1.2.3 使用控制台获取信息	6
1.3 群组新增节点	7
1.3.1 为新节点生成私钥证书	7
1.3.2 准备配置文件	7
1.3.3 启动新节点并加入群组	8
2 自行编写一个智能合约并部署到私有链上，同时完成合约调用。（截图说明部署流程）	9
2.1 Upvote合约	9
2.1.1 提供接口	9
2.1.2 合约内容	9
2.2 部署Upvote合约	12
2.3 调用Upvote合约	12

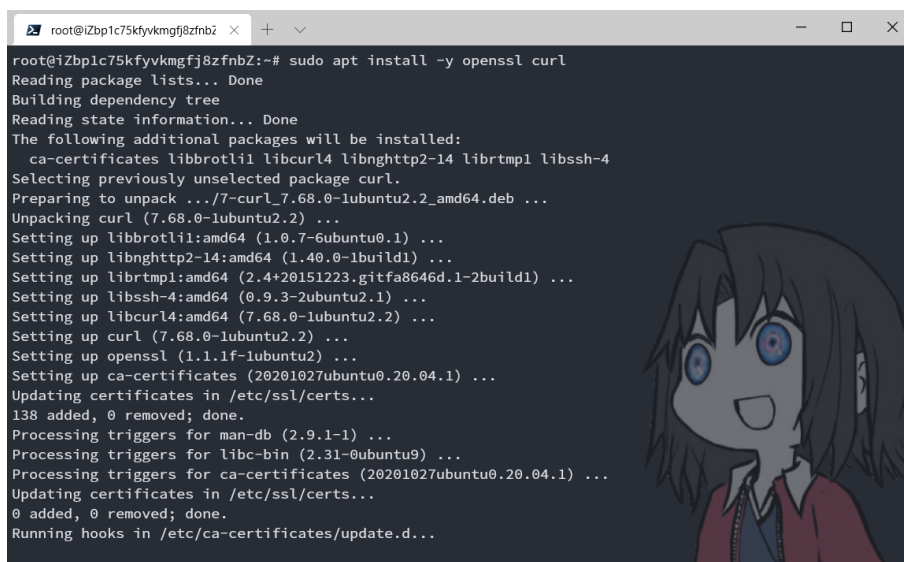
2.3.1	创建2个新的账户	12
2.3.2	调用Upvote合约的接口	12
3	使用命令查看一个区块，并对各个字段进行解释。	14

1 使用已有的开源区块链系统FISCO-BCOS，完成私有链的搭建以及新节点的加入。（截图说明搭建流程）

1.1 单群组FISCO BCOS联盟链的搭建

1.1.1 准备环境

1. 安装依赖



```
root@izbp1c75kfyvkmgfj8zfnbZ:~# sudo apt install -y openssl curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates libbrotli1 libcurl4 libnghttp2-14 librtmp1 libssh-4
Selecting previously unselected package curl.
Preparing to unpack .../7-curl_7.68.0-1ubuntu2.2_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.2) ...
Setting up libbrotli1:amd64 (1.0.7-6ubuntu0.1) ...
Setting up libnghttp2-14:amd64 (1.40.0-1build1) ...
Setting up librtmp1:amd64 (2.4+20151223.gitfa8646d.1-2build1) ...
Setting up libssh-4:amd64 (0.9.3-2ubuntu2.1) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.2) ...
Setting up curl (7.68.0-1ubuntu2.2) ...
Setting up openssl (1.1.1f-1ubuntu2) ...
Setting up ca-certificates (20201027ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs...
138 added, 0 removed; done.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for ca-certificates (20201027ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
```

图 1: 安装ubuntu依赖

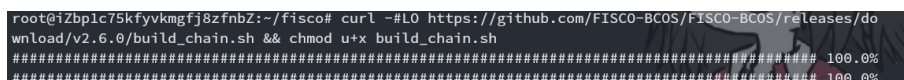
2. 创建操作目录



```
root@izbp1c75kfyvkmgfj8zfnbZ:~# cd ~ && mkdir -p fisco && cd fisco
root@izbp1c75kfyvkmgfj8zfnbZ:~/fisco#
```

图 2: 创建操作目录

3. 下载build_chain.sh脚本



```
root@izbp1c75kfyvkmgfj8zfnbZ:~/fisco# curl -#LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.6.0/build_chain.sh && chmod u+x build_chain.sh
##### 100.0%
##### 100.0%
```

图 3: 下载build_chain.sh脚本

1.1.2 搭建单群组4节点联盟链

```
root@iZbp1c75kfyvkmfj8zfnbZ:~/fisco# bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545
[INFO] Downloading fisco-bcos binary from https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.6.0/fisco-bcos.tar.gz ...
##### 100.0%
# 1.8%
curl: (28) Operation too slow. Less than 102400 bytes/sec transferred the last 20 seconds

[INFO] Download speed is too low, try https://osp-1257653870.cos.ap-guangzhou.myqcloud.com/FISCO-BCOS/FISCO-BCOS/releases/v2.6.0/fisco-bcos.tar.gz
##### 100.0%
=====
Generating CA key...
=====
Generating keys and certificates ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
=====
Generating configuration files ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
=====
[INFO] Start Port      : 30300 20200 8545
[INFO] Server IP       : 127.0.0.1:4
[INFO] Output Dir        : /root/fisco/nodes
[INFO] CA Path            : /root/fisco/nodes/cert/
=====
[INFO] Execute the download_console.sh script in directory named by IP to get FISCO-BCOS console.
e.g.  bash /root/fisco/nodes/127.0.0.1/download_console.sh -f
=====
[INFO] All completed. Files in /root/fisco/nodes
```

图 4: 搭建单群组4节点联盟链

1.1.3 启动FISCO BCOS链

```
root@iZbp1c75kfyvkmfj8zfnbZ:~/fisco# bash nodes/127.0.0.1/start_all.sh
try to start node0
try to start node1
try to start node2
try to start node3
node3 start successfully
node1 start successfully
node2 start successfully
node0 start successfully
```

图 5: 启动FISCO BCOS链

1.1.4 检查进程

```
root@iZbp1c75kfyvkmfj8zfnbZ:~/fisco# ps -ef | grep -v grep | grep fisco-bcos
root      18239      1  1 21:50 pts/0    00:00:01 /root/fisco/nodes/127.0.0.1/node3/./fisco-bcos
-c config.ini
root      18241      1  1 21:50 pts/0    00:00:01 /root/fisco/nodes/127.0.0.1/node1/./fisco-bcos
-c config.ini
root      18243      1  1 21:50 pts/0    00:00:01 /root/fisco/nodes/127.0.0.1/node2/./fisco-bcos
-c config.ini
root      18245      1  1 21:50 pts/0    00:00:01 /root/fisco/nodes/127.0.0.1/node0/./fisco-bcos
-c config.ini
```

图 6: 检查进程

1.1.5 检查日志输出

1. 查看节点node0链接的节点数

```
root@iZbp1c75kfyvkmfj8zfnbZ:~/fisco# tail -f nodes/127.0.0.1/node0/log/log* | grep connected
info|2020-11-26 21:59:06.314063|[P2P][Service] heartBeat,connected count=3
info|2020-11-26 21:59:16.314194|[P2P][Service] heartBeat,connected count=3
info|2020-11-26 21:59:26.314334|[P2P][Service] heartBeat,connected count=3
```

图 7: 查看节点node0链接的节点数

2. 检查是否在共识

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco# tail -f nodes/127.0.0.1/node0/log/log* | grep +
info|2020-11-26 21:59:58.006058|[g:1][CONSENSUS][SEALER]+++++++ Generating seal on,blkNum=1
,tx=0,nodeIdx=0,hash=36d3fbf2...
info|2020-11-26 22:01:26.732441|[g:1][CONSENSUS][SEALER]+++++++ Generating seal on,blkNum=1
,tx=0,nodeIdx=0,hash=9f9f843f...
info|2020-11-26 22:01:30.765033|[g:1][CONSENSUS][SEALER]+++++++ Generating seal on,blkNum=1
,tx=0,nodeIdx=0,hash=d4b5f012...
```

图 8: 检查是否在共识

1.2 配置及使用控制台

1.2.1 准备依赖

1. 安装java

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco# sudo apt install -y default-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  at-spi2-core ca-certificates-java default-jdk-headless default-jre default-jre-headless
  fontconfig-config fonts-dejavu-core fonts-dejavu-extra java-common libatk-bridge2.0-0
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libfontenc1 libgif7
  libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libice-dev libice6
  libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm10 libpciaccess0 libpcsclite1 libpthread-stubs0-dev
  libsm-dev libsm6 libvulkan1 libwayland-client0 libx11-dev libx11-xcb1 libxau-dev libxaw7
  libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-sync1
  libxcb1-dev libxcomposite1 libxdamage1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1
  libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6 libxtst6
  libxv1 libxxf86dgal libxxf86vml mesa-vulkan-drivers openjdk-11-jdk openjdk-11-jdk-headless
  openjdk-11-jre openjdk-11-jre-headless x11-common x11-utils x11proto-core-dev x11proto-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc liblcms2-utils pcsd libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic mesa-utils
The following NEW packages will be installed:
  at-spi2-core ca-certificates-java default-jdk default-jdk-headless default-jre
  default-jre-headless fontconfig-config fonts-dejavu-core fonts-dejavu-extra java-common
  libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
  libatspi2.0-0 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1
  libfontenc1 libgif7 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0
```

图 9: 安装java

2. 获取控制台并回到fisco目录

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco# cd ~/fisco && curl -#L0 https://github.com/FISCO-BCOS/console/
releases/download/v2.6.1/download_console.sh && bash download_console.sh
##### 100.0%
##### 100.0%
[INFO] Downloading console 2.7.0 from https://github.com/FISCO-BCOS/console/releases/download/v2.7.0
/console.tar.gz
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 640 100 640 0 0 134 0 0:00:04 0:00:04 --:--:-- 134
0 38.1M 0 194k 0 0 5498 0 2:01:14 0:00:36 2:00:38 4170
curl: (28) Operation too slow. Less than 102400 bytes/sec transferred the last 30 seconds
[WARN] Download speed is too low, try https://osp-1257653870.cos.ap-guangzhou.myqcloud.com/FISCO-BCO
S/console/releases/v2.7.0/console.tar.gz
##### 100.0%
```

图 10: 获取控制台并回到fisco目录

3. 拷贝控制台配置文件


```
[group:1]> getPeers
[
  PeerInfo{
    nodeID='b6a6b46e56db48e482ffa1a4073e2b2b45b20c62896c150d036da3ff6ea752b9cbb091b8c5c99f1c504b
c7952da9178c8f335d81a6d1d2ee39d266ce6af7398',
    iPAndPort='127.0.0.1:48486',
    node='node3',
    agency='agency',
    topic='[
      ]'
    },
    PeerInfo{
      nodeID='67102de0d8a858e244f2455935dfd5f4545cbb5c54e6bf36021aa77150a332e6895f13ade153b9e12d4e
2f5f6638d9f8c20b1085f39a66cea2cdc815f6e548b8',
      iPAndPort='127.0.0.1:48478',
      node='node2',
      agency='agency',
      topic='[
        ]'
      },
      PeerInfo{
        nodeID='5a934a69c4e9c7ccb13cb6e0ca5e16293a183f77270d2b397e0f156a4446b91e8ff1f969764e46e7a39f
0fb5ab4452ed43cfd05dad781642f8ad522c62384ee',
        iPAndPort='127.0.0.1:48502',
        node='node1',
        agency='agency',
        topic='[
          _block_notify_1
        ]'
      }
    ]
  }
]
```

图 15: 获取节点链接信息

1.3 群组新增节点

1.3.1 为新节点生成私钥证书

1. 获取证书生成脚本

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# curl -#L0 https://raw.githubusercontent.com/FISCO-BCOS/FISCO-BCOS/master/tools/gen_node_cert.sh
##### 100.0%
```

图 16: 获取证书生成脚本

2. 生成新节点私钥证书

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# bash gen_node_cert.sh -e ../cert/agency -o new
Node
=====
[INFO] Cert Path : ../cert/agency
[INFO] Output Dir : newNode
=====
[INFO] All completed. Files in newNode
```

图 17: 生成新节点私钥证书

1.3.2 准备配置文件

1. 拷贝群组1中节点node0配置文件与工具脚本

```
root@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# cp node0/config.ini newNode/config.ini
cp node0/conf/grouproot@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# cp node0/conf/group.1.genesis
s newNode/conf/group.1.genesis
cp node0/conf/root@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# cp node0/conf/group.1.ini newNod
e.inif/group.1
cp node0/*.shroot@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# cp node0/*.sh newNode/
p -r node0root@izbp1c75kfyvkmfj8zfnbZ:~/fisco/nodes/127.0.0.1# cp -r node0/scripts newNode/
```

图 18: 拷贝群组1中节点node0配置文件与工具脚本

2. 更新newNode/config.ini中监听的IP和端口

```
root@izbp1c75kfyvkmgfj8zfnbz ~/f/n/1/n/config.ini
1 [rpc]
2   channel_listen_ip=0.0.0.0
3   channel_listen_port=20204
4   jsonrpc_listen_ip=127.0.0.1
5   jsonrpc_listen_port=8549
6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30304
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14
```

图 19: 更新newNode/config.ini中监听的IP和端口

3. 将新节点的P2P配置中的IP和Port加入原有节点的config.ini中的[p2p]字段

```
6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30304
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14  node.4=127.0.0.1:30304
15

6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30300
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14  node.4=127.0.0.1:30304
15

6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30301
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14  node.4=127.0.0.1:30304
15

6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30302
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14  node.4=127.0.0.1:30304
15

6 [p2p]
7   listen_ip=0.0.0.0
8   listen_port=30303
9   ; nodes to connect
10  node.0=127.0.0.1:30300
11  node.1=127.0.0.1:30301
12  node.2=127.0.0.1:30302
13  node.3=127.0.0.1:30303
14  node.4=127.0.0.1:30304
15
```

图 20: 将新节点的P2P配置中的IP和Port加入原有节点的config.ini中的[p2p]字段

1.3.3 启动新节点并加入群组

1. 启动新节点，执行newNode/start.sh

```
root@izbp1c75kfyvkmgfj8zfnbz:~/fisco/nodes/127.0.0.1# bash newNode/start.sh
newNode start successfully
```

图 21: 启动新节点，执行newNode/start.sh

2. 通过console将新节点加入群组1

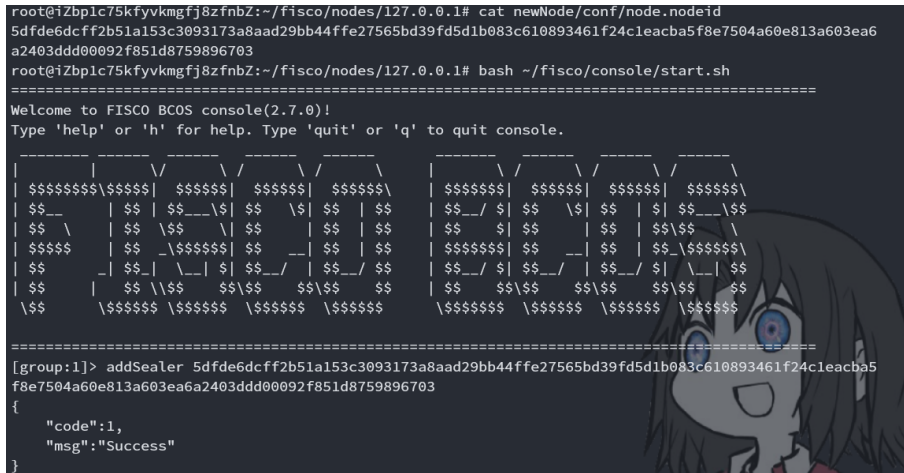


图 22: 通过console将新节点加入群组1

3. 检查连接和共识

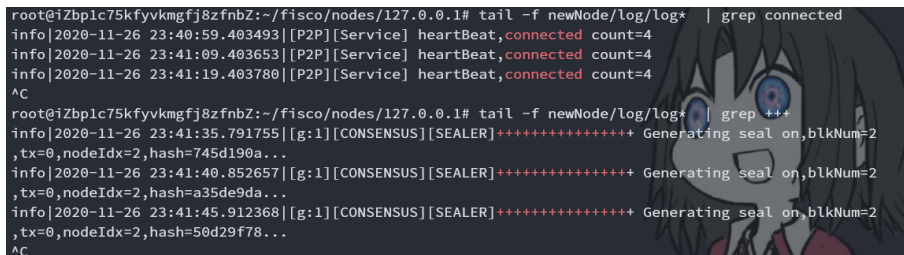


图 23: 检查连接和共识

2 自行编写一个智能合约并部署到私有链上，同时完成合约调用。（截图说明部署流程）

2.1 Upvote合约

2.1.1 提供接口

1. **publish(content_names):** 参数是内容名字的数组，发布内容，只有合约中唯一的发布者可以调用。
2. **giveVotesToUser(user):** 参数是用户地址，给该用户10个点赞机会，只有合约中唯一的发布者可以调用，每个用户只能被授予1次。
3. **upvote(name, votes):** 参数是内容名和点赞数，给某个内容点赞若干次。
4. **winner():** 返回当前票数最高的内容名。

2.1.2 合约内容

```
1 pragma solidity >=0.4.24;
```

```

2
3 contract Upvote
4 {
5     struct User
6     {
7         uint votes;
8         bool given;
9     }
10
11     struct Content
12     {
13         string name;
14         uint votes;
15     }
16
17     address public publisher;
18     mapping(address => User) public users;
19     Content[] public contents;
20
21     constructor() public
22     {
23         publisher = msg.sender;
24     }
25
26     function publish(string memory name) public
27     {
28         require(msg.sender == publisher, "Only the publisher can
29         publish contents.");
30         for (uint i = 0; i < contents.length; i++)
31         {
32             require(keccak256(name) != keccak256(contents[i].
33             name), "The content already exists.");
34         }
35         contents.push(Content({name: name, votes: 0}));
36     }
37
38     function giveVotesToUser(address user) public
39     {
40         require(msg.sender == publisher, "Only the publisher can
41         give votes to a user.");

```

```

39         require(users[user].given == false, "The user has been
given votes.");
40         users[user].votes = 10;
41     }
42
43     function upvote(string memory name, uint votes) public
44     {
45         User storage sender = users[msg.sender];
46         require(sender.votes >= votes, "Don't have enough votes.
");
47         uint i = 0;
48         for (; i < contents.length; i++)
49         {
50             if (keccak256(name) == keccak256(contents[i].name))
51             {
52                 sender.votes -= votes;
53                 contents[i].votes += votes;
54                 break;
55             }
56         }
57         require(i < contents.length, "The content doesn't exist.
");
58     }
59
60     function winner() public view returns(string memory)
61     {
62         uint max_votes = 0;
63         string storage name;
64         for (uint i = 0; i < contents.length; i++)
65         {
66             if (max_votes < contents[i].votes)
67             {
68                 max_votes = contents[i].votes;
69                 name = contents[i].name;
70             }
71         }
72         return name;
73     }
74 }

```

2.2 部署Upvote合约

```
[group:1]> deploy Upvote
transaction hash: 0x1d8c21f8a2e04ca9e9bffa79dbf699ec9c97ce7cc04d15c0f61c48cf32cce0
contract address: 0x03fa8f8e9a53dda337ee8b2a810e3610030fc72
```

图 24: 部署Upvote合约

2.3 调用Upvote合约

2.3.1 创建2个新的账户

```
[group:1]> newAccount
AccountPath: account/ecdsa/0x879fdb8f4abbc695b9ae31350ef17c0cd9a3ce6.pem
newAccount: 0x879fdb8f4abbc695b9ae31350ef17c0cd9a3ce6
AccountType: ecdsa

[group:1]> newAccount
AccountPath: account/ecdsa/0x79d2cb46c30a0f8d959a4bad9903febb5faefa2.pem
newAccount: 0x79d2cb46c30a0f8d959a4bad9903febb5faefa2
AccountType: ecdsa
```

图 25: 创建2个新的账户

2.3.2 调用Upvote合约的接口

1. 使用当前新用户调用giveVotesToUser，失败

```
[group:1]> listAccount
0x79d2cb46c30a0f8d959a4bad9903febb5faefa2(current account) <-
0x879fdb8f4abbc695b9ae31350ef17c0cd9a3ce6
0xdb0e858cce076c6dba949a9f4500df5fcae56a0

[group:1]> call Upvote 0x83fa8f8e9a53dda337ee8b2a810e3610030fc72 giveVotesToUser 0x879fdb8f4abbc695b9ae31350ef17c0cd9a3ce6
transaction hash: 0x743204a79ac521e8cfff9a9649dc5ed75d51ebe0e13db57f41b435019934af7
-----
transaction status: 0x16
-----
Output
Receipt message: Only the publisher can give votes to a user.
Return message: Only the publisher can give votes to a user.
Return value: null
-----
```

图 26: 使用当前新用户调用giveVotesToUser

2. 使用部署合约用户调用giveVotesToUser，成功，说明部署合约的用户是publisher。

```
[group:1]> loadAccount 0xdb0e858cce076c6dba949a9f4500df5fcae56a0
Load account 0xdb0e858cce076c6dba949a9f4500df5fcae56a0 success!

[group:1]> call Upvote 0x83fa8f8e9a53dda337ee8b2a810e3610030fc72 giveVotesToUser 0x879fdb8f4abbc695b9ae31350ef17c0cd9a3ce6
transaction hash: 0xc2301be77159c9fa51d11721c375004840e0a5cf270e566b598537a883e5a0b
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}

[group:1]> call Upvote 0x83fa8f8e9a53dda337ee8b2a810e3610030fc72 giveVotesToUser 0x79d2cb46c30a0f8d959a4bad9903febb5faefa2
transaction hash: 0x127f7b22e6de71ca0fcee448a4201bb9ebf55b59409c7240931e30c9204e
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}
```

图 27: 使用部署合约用户调用giveVotesToUser

3. publisher调用publish，如果遇到相同内容会报错

```
[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 publish "Nice Boat"
transaction hash: 0x5cde404bc5bb7ff4cc0e672a57b7d94ab965ee5393464743bb7fd858bbd3c2a
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}

[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 publish "Pluck!"
transaction hash: 0x811b5ab8d123a7c8441285b1cfeb1712f24baf8cd9c23ff88ac96f01cac07d4
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}

[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 publish "Nice Boat"
transaction hash: 0x10feb5a0a1a25c8854ca299006ac796c4b2a1d52ff6dead30be2434528a577
-----
transaction status: 0x16
-----
Output
Receipt message: The content already exists.
Return message: The content already exists.
Return value: null
-----
Event logs
Event: {}
```

图 28: publisher调用publish

4. 切换用户调用upvote

- 当票数足够时，成功运行

```
[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 upvote "Nice Boat" 5
transaction hash: 0x842903adc59738a0915ef369e89e2e63df60039f6b51e5f9f2ae7d51d6e42fba
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}

[group1]: loadAccount 0x879fdbc8f4abbce95b9ae31350ef170cd9a3ce6
load account 0x879fdbc8f4abbce95b9ae31350ef170cd9a3ce6 success!

[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 upvote "Pluck!" 3
transaction hash: 0x1590514990e1588d17a3ed6d86f7a233134521a20c7f161981ab001a23ee8279
-----
transaction status: 0x0
description: transaction executed successfully
-----
Output
Receipt message: Success
Return message: Success
Return value: []
-----
Event logs
Event: {}
```

图 29: 调用upvote票数足够

- 当票数不够时，报错

```
[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 upvote "Pluck!" 8
transaction hash: 0x5f023ca8d28d41d4fce8f8048f1114a0f848c3cb2f0a4c824d8706280d0832d6
-----
transaction status: 0x16
-----
Output
Receipt message: Don't have enough votes.
Return message: Don't have enough votes.
Return value: null
-----
Event logs
Event: {}
```

图 30: 调用upvote票数不够

5. 调用winner

```
[group1]: call Upvote 0x83fa8fe9a53dda337ee8b2a810e3610030fc72 winner
-----
Return code: 0
description: transaction executed successfully
Return message: Success
-----
Return values:
{
  "Nice Boat"
}
```

图 31: 调用winner

3 使用命令查看一个区块，并对各个字段进行解释。

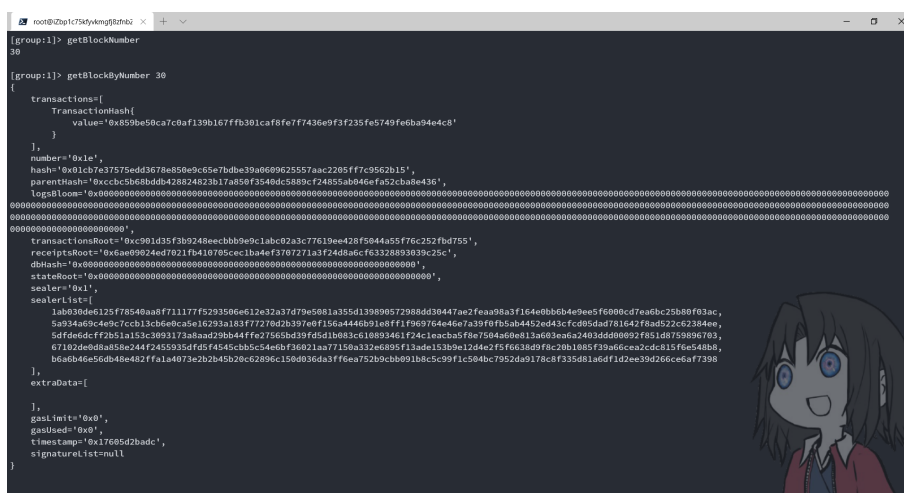


图 32: 调用upvote票数不够

- transactions: 交易列表
 - TransactionHash: 交易哈希
- number: 区块高度
- hash: 区块哈希
- parentHash: 父区块哈希
- logsBloom: log的布隆过滤器值
- transactionRoot: 区块内所有交易的merkle根
- receiptsRoot: 区块内所有交易回执的merkle根
- dbHash: 记录交易数据变更的哈希
- stateRoot: 状态根哈希
- sealer: 共识节点序号
- sealerList: 共识节点列表
- extraData: 交易内的extraData
- gasLimit: 区块中允许的gas最大值
- gasUsed: 区块中所有交易消耗的gas
- timestamp: 时间戳, 单位毫秒
- signatureList: PBFT共识的签名列表