区块链 大作业热身报告

> 姓名：唐瑞怡
>
> 学号：18340159
>
> 专业：计算机科学与技术
>
> 时间：2020/11/26

一、完成私有链的搭建以及新节点的加入

1.1 准备环境

安装ubuntu依赖

```
sudo apt install -y openssl curl
```

```
shelly@shelly-virtual-machine:~$ sudo apt install -y openssl curl
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
  snapd-login-service
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件:
  libcurl3-gnutls
下列【新】软件包将被安装:
  curl
下列软件包将被升级:
  libcurl3-gnutls openssl
升级了 2 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 173 个软件包未被升级。
需要下载 815 kB 的归档。
解压缩后会消耗 340 kB 的额外空间。
获取:1 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 libcurl3-gnutls amd64 7.47.0-1ubuntu2.16 [184 kB]
获取:2 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 openssl amd64 1.0.2g-1ubuntu4.17 [492 kB]
获取:3 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 curl amd64 7.47.0-1ubuntu2.16 [139 kB]
已下载 815 kB，耗时 1秒 (526 kB/s)
(正在读取数据库 ... 系统当前共安装有 216767 个文件和目录。)
正准备解包 .../libcurl3-gnutls_7.47.0-1ubuntu2.16_amd64.deb  ...
正在将 libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.16) 解包到 (7.47.0-1ubuntu2.15) 上 ...
正准备解包 .../openssl_1.0.2g-1ubuntu4.17_amd64.deb  ...
正在将 openssl (1.0.2g-1ubuntu4.17) 解包到 (1.0.2g-1ubuntu4.16) 上 ...
正在选中未选择的软件包 curl。
正准备解包 .../curl_7.47.0-1ubuntu2.16_amd64.deb  ...
正在解包 curl (7.47.0-1ubuntu2.16) ...
正在处理用于 libc-bin (2.23-0ubuntu11) 的触发器 ...
正在处理用于 man-db (2.7.5-1) 的触发器 ...
正在设置 libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.16) ...
正在设置 openssl (1.0.2g-1ubuntu4.17) ...
正在设置 curl (7.47.0-1ubuntu2.16) ...
正在处理用于 libc-bin (2.23-0ubuntu11) 的触发器 ...
```

创建操作目录

```
cd ~ && mkdir -p fisco && cd fisco
```

```
shelly@shelly-virtual-machine:~$ cd ~ && mkdir -p fisco && cd fisco
shelly@shelly-virtual-machine:~/fisco$ curl -#LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.7.0/build_chain.sh && chmod u+x build_chain.sh
######################################################################## 100.0%
```

1.2 搭建单群祖4节点联盟链

```
bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545
```

```
shelly@shelly-virtual-machine:~/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/v2.7.0/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      : /home/shelly/fisco/nodes
[INFO] CA Path         : /home/shelly/fisco/nodes/cert/
============================================================
[INFO] Execute the download_console.sh script in directory named by IP to get FISCO-BCOS console.
e.g.  bash /home/shelly/fisco/nodes/127.0.0.1/download_console.sh -f
============================================================
[INFO] All completed. Files in /home/shelly/fisco/nodes
```

1.3 启动FISCO BCOS链

```
bash nodes/127.0.0.1/start_all.sh
```

```
shelly@shelly-virtual-machine:~/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
 node0 start successfully
 node3 start successfully
 node1 start successfully
 node2 start successfully
```

1.4 检查进程

```
ps -ef | grep -v grep | grep fisco-bcos
```

```
shelly@shelly-virtual-machine:~/fisco$ ps -ef | grep -v grep | grep fisco-bcos
shelly    50707   1895  2 21:31 pts/6    00:00:00 /home/shelly/fisco/nodes/127.0.0.1/node0/../fisco-bcos -c config.ini
shelly    50709   1895  2 21:31 pts/6    00:00:00 /home/shelly/fisco/nodes/127.0.0.1/node1/../fisco-bcos -c config.ini
shelly    50711   1895  2 21:31 pts/6    00:00:00 /home/shelly/fisco/nodes/127.0.0.1/node3/../fisco-bcos -c config.ini
shelly    50713   1895  2 21:31 pts/6    00:00:00 /home/shelly/fisco/nodes/127.0.0.1/node2/../fisco-bcos -c config.ini
```

进程数=4，启动成功。

1.5 检查日志输出

查看节点node0链接的节点数

```
tail -f nodes/127.0.0.1/node0/log/log*  | grep connected
```

```
shelly@shelly-virtual-machine:~/fisco$ tail -f nodes/127.0.0.1/node0/log/log*  | grep connected
info|2020-11-26 21:35:16.568005|[P2P][Service] heartBeat,connected count=3
info|2020-11-26 21:35:26.568493|[P2P][Service] heartBeat,connected count=3
info|2020-11-26 21:35:36.569454|[P2P][Service] heartBeat,connected count=3
info|2020-11-26 21:35:46.569797|[P2P][Service] heartBeat,connected count=3
^Z
[1]+  已停止               tail -f nodes/127.0.0.1/node0/log/log* | grep --color=auto connected
```

检查是否在共识

```
tail -f nodes/127.0.0.1/node0/log/log*  | grep +++
```

```
shelly@shelly-virtual-machine:~/fisco$ tail -f nodes/127.0.0.1/node0/log/log*  | grep +++
info|2020-11-26 21:36:33.534165|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=2ed9
f4af...
info|2020-11-26 21:36:37.676358|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=7650
1a7b...
info|2020-11-26 21:36:41.848805|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=132e
9ebb...
info|2020-11-26 21:36:45.992394|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=9596
0893...
info|2020-11-26 21:36:50.109053|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=d363
1658...
info|2020-11-26 21:36:54.281826|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=eeab
b14a...
info|2020-11-26 21:36:58.400382|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=1,tx=0,nodeIdx=0,hash=1be5
a9e3...
^Z
[2]+  已停止               tail -f nodes/127.0.0.1/node0/log/log* | grep --color=auto +++
```

不停输出++++Generating seal，表示共识正常。

二、配置及使用控制台

2.1 准备依赖

安装java

```
sudo apt install -y default-jdk
```

安装成功：

```
done.
done.
shelly@shelly-virtual-machine:~/fisco$
```

获取控制台并回到fisco目录

```
cd ~/fisco && curl -#LO https://github.com/FISCO-
BCOS/console/releases/download/v2.7.0/download_console.sh && bash
download_console.sh
```

```
shelly@shelly-virtual-machine:~/fisco$ cd ~/fisco && curl -#LO https://github.com/FISCO-BCOS/console/releases/download/v2.7.0
/download_console.sh && bash download_console.sh
######################################################################## 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    266      0  0:00:02  0:00:02 --:--:--   266
  5 38.1M    5 1970k    0     0  49735      0  0:13:24  0:00:40  0:12:44 21473
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-BCOS/console/releases/v2.7.0
/console.tar.gz
######################################################################## 100.0%
[INFO] Download console successfully
[INFO] unzip console successfully
```

拷贝控制台配置文件

```
# 最新版本控制台使用如下命令拷贝配置文件
cp -n console/conf/config-example.toml console/conf/config.toml
```

```
shelly@shelly-virtual-machine:~/fisco$ cp -n console/conf/config-example.toml console/conf/config.toml
```

配置控制台证书

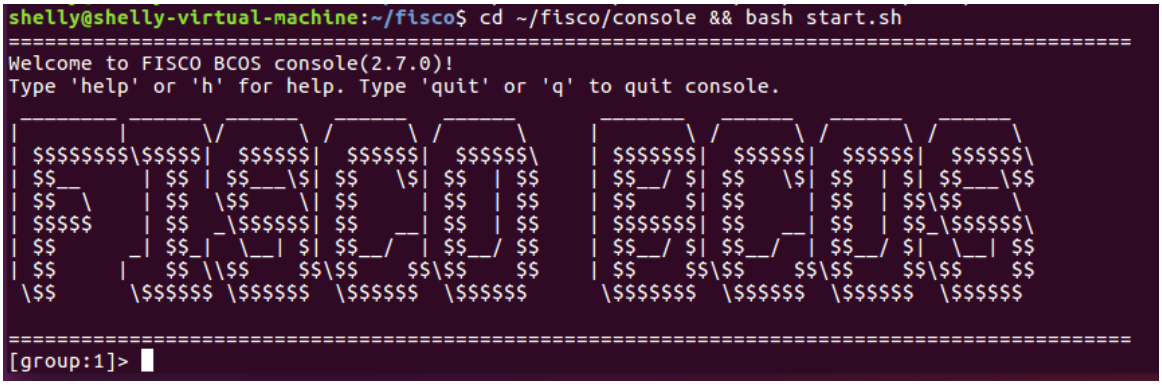
```
cp -r nodes/127.0.0.1/sdk/* console/conf/
```

```
shelly@shelly-virtual-machine:~/fisco$ cp -r nodes/127.0.0.1/sdk/* console/conf/
shelly@shelly-virtual-machine:~/fisco$
```

2.2 启动并使用控制台

启动控制台

```
cd ~/fisco/console && bash start.sh
```



使用控制台获取信息

```
getNodeVersion
```

```
[group:1]> getNodeVersion
ClientVersion{
    version='2.7.0',
    supportedVersion='2.7.0',
    chainId='1',
    buildTime='20201126 08:00:45',
    buildType='Linux/clang/Release',
    gitBranch='HEAD',
    gitCommitHash='0bc47666979df4766723adf1ab9c5d80f5e40537'
}
```

```
getPeers
```

```
[group:1]> getPeers
[
    PeerInfo{
        nodeID='6054fc55b6684e683e7430df0483902384b9c4eaeb4e3d5adc626d41b78b8d7991effef47ccff85d366270b78445
e8aaa92d7808791bfcc4e0abe57d7a8d0ae7',
        iPAndPort='127.0.0.1:48290',
        node='node2',
        agency='agency',
        topic='[

        ]'
    },
    PeerInfo{
        nodeID='ff30a9bc9eb4d0aebf696254e00063bc1e9ac6a71d951fb5006928c963c8ef175984182eb9ff49d9cc0da229f13f
5a4c6f350daa23b49584d36db43c96a2d6f7',
        iPAndPort='127.0.0.1:48288',
        node='node1',
        agency='agency',
        topic='[
            _block_notify_1
        ]'
    },
    PeerInfo{
        nodeID='efa79105aba8e6f4b83ff0c2c92566050c9a8a884a10bf757f443def07365ec9ce09c08acba4ee43020280d842ba
6eb87921e8aa44a96a0bb14955347c613c0c',
        iPAndPort='127.0.0.1:48292',
        node='node3',
        agency='agency',
        topic='[

        ]'
    }
]
```

三、新节点的加入

在**fisco/nodes/127.0.0.1**目录下：

- **获取证书生成脚本**

```
curl -#LO https://raw.githubusercontent.com/FISCO-
BCOS/FISCOBCOS/master/tools/gen_node_cert.sh
```

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ curl -#LO https://raw.githubusercontent.com/FISCO-BCO
S/FISCOBCOS/master/tools/gen_node_cert.sh
######################################################################## 100.0%
```

然而，进行下一步的时候**一直报错**：

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ bash gen_node_cert.sh -c ../cert/agency -o newNode
gen_node_cert.sh: 行 1: 404:: 未找到命令
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ bash gen_node_cert.sh -c ../cert/agency -o newNode
gen_node_cert.sh: 行 1: 404:: 未找到命令
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ bash gen_node_cert.sh -c ../cert/agency -o newNode
```

后来，打开gen_node_cert.sh文件发现是**“404 Not found”，存在网络问题。**

因此，按照网站上的提示，使用如下语句：

```
curl -#LO https://gitee.com/FISCO-BCOS/FISCO-
BCOS/raw/master/tools/gen_node_cert.sh
```

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ curl -#LO https://gitee.com/FISCO-BCOS/FISCO-BCOS/raw/master/tools/gen_node_cert.sh
######################################################################## 100.0%
```

成功下载！

- **生成新节点的私钥证书**

```
bash gen_node_cert.sh -c ../cert/agency -o newNode
```

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ bash gen_node_cert.sh -c ../cert/agency -o newNode
============================================================
[INFO] Cert Path    : ../cert/agency
[INFO] Output Dir   : newNode
============================================================
[INFO] All completed. Files in newNode
```

此时新创建文件夹 newNode。

- **拷贝群组1中节点node0的配置文件和工作脚本等到newNode**

```
cp node0/config.ini newNode/config.ini
cp node0/conf/group.1.genesis newNode/conf/group.1.genesis
cp node0/conf/group.1.ini newNode/conf/group.1.ini
cp node0/*.sh newNode/
cp -r node0/scripts newNode/
```

- **修改新节点的配置文件config.ini**

  对于`[rpc]`模块，修改`listen_ip`、`channel_listen_port`和`jsonrpc_listen_port`；对于`[p2p]`模块，修改`listen_port`。并在node.4中增加自身节点信息：

```
[rpc]
    channel_listen_ip=127.0.0.1
    channel_listen_port=20204
    jsonrpc_listen_ip=127.0.0.1
    jsonrpc_listen_port=8549
[p2p]
    listen_ip=0.0.0.0
    listen_port=30304
    ; nodes to connect
    node.0=127.0.0.1:30300
    node.1=127.0.0.1:30301
    node.2=127.0.0.1:30302
    node.3=127.0.0.1:30303
    node.4=127.0.0.1:30304
```

- **启动新节点**

```
newNode/start.sh
```

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ newNode/start.sh
 newNode start successfully
```

- **查看新节点的id**

```
more conf/node.nodeid
```

```
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1$ cd newNode
shelly@shelly-virtual-machine:~/fisco/nodes/127.0.0.1/newNode$ more conf/node.nodeid
b16c128bab7d259474bfb2889ca89dae59c10342e626c4919cff4a0d9716f2fc34e165a1f311fc7e52dac9b2788584c4274f03f2d58711f4e8b5028cfd5e5867
```

- **启动控制台将新节点设置为共识节点**

```
[group:1]> addSealer
1b1e9a7243549a38b82fe6eef8021382bea5597a0194fa129721cdc56c61576a17a1a46a61109405
744beef73c6546105fb3770d4f18319935610c9ea10bb4b8
```

```
shelly@shelly-virtual-machine:~/fisco$ cd ~/fisco/console && bash start.sh
=====================================================================================================
Welcome to FISCO BCOS console(2.7.0)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
 ________ ______ ______  ______  ______      ________  ______  ______  ______
|        |      \/      \ /      \ /      \    |        \ /      \ /      \ /      \
| $$$$$$$$\$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\   | $$$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\
| $$__       | $$ | $$___\$| $$   \$| $$  | $$   | $$__/ $| $$   \$| $$  | $| $$___\$$
| $$  \      | $$  \$$    \| $$      | $$  | $$   | $$    $| $$      | $$  | $$\$$    \
| $$$$$      | $$  _\$$$$$$| $$   __ | $$  | $$   | $$$$$$$| $$   __ | $$  | $$_\$$$$$$\
| $$        _| $$_|  \__| $| $$__/  \| $$__/ $$   | $$__/ $| $$__/  \| $$__/ $|  \__| $$
| $$       |   $$ \\$$    $$\\$$    $$\\$$    $$   | $$    $$\\$$    $$\\$$    $$\\$$    $$
 \$$        \$$$$$$ \$$$$$$  \$$$$$$  \$$$$$$     \$$$$$$$  \$$$$$$  \$$$$$$  \$$$$$$

=====================================================================================================
[group:1]> addSealer b16c128bab7d259474bfb2889ca89dae59c10342e626c4919cff4a0d9716f2fc34e165a1f311fc7e52dac9b2788584c4274f03f2d58711f4e8b5028cfd5e5867
{
    "code":1,
    "msg":"Success"
}
```

- **检查连接和共识**

```
[group:1]> getSealerList
```

```
[group:1]> getSealerList
[
    2a558043e4cfc98ccbec7d584aad737b2faaf9e06a2b652cee523355c3500e82d57dcd89f4792265a9f31c9cb7d7698e9c628f1a1d8873273044ad0ea3af6e78,
    40c8f68fe64efdb6eead09931ace24e67a67d0ec60ca00e9588a330bcd5cfa10c8cfadab8272ebe704c3c31fc9ee4035c2ae4a5692762d705d754dc3a245ced2,
    f27cd291057887e6a71cc32e11db6b6f16cb7ee0f80a9571a0f5237965372383234494d13d61929fd36062cdf202782fd54afce827d92b70a6ec75822ab9ca0f,
    fe3b57e200290d802ed9c755f570ec31db165654dcd2767bd9595b2ebdbc5c98a3a73d31ce441654d8e12609c3d278333a9ff2eeb241ac4a4dd797ce09d2ae21,
    b16c128bab7d259474bfb2889ca89dae59c10342e626c4919cff4a0d9716f2fc34e165a1f311fc7e52dac9b2788584c4274f03f2d58711f4e8b5028cfd5e5867
]
```

可以发现，新节点的确加入到了共识列表中。

- **查看node0的连接：**

```
tail -f nodes/127.0.0.1/node0/log/log* | grep connected
```

```
shelly@shelly-virtual-machine:~/fisco$ tail -f nodes/127.0.0.1/node0/log/log* | grep connected
info|2020-11-27 10:59:59.007549|[P2P][Service] heartBeat,connected count=3
info|2020-11-27 12:59:57.902013|[P2P][Service] heartBeat,connected count=4
info|2020-11-27 13:07:17.929344|[P2P][Service] heartBeat,connected count=4
info|2020-11-27 13:07:27.929874|[P2P][Service] heartBeat,connected count=4
```

可以发现，连接数的确从3变到了4，新节点插入成功。

四、编写智能合约、部署到私有链、合约调用

4.1 编写合约

在`fisco/console/contracts/solidity`目录下，新建`Mycontract.sol`。`Mycontract`合约提供`get()`和`set()`两个接口，用来获取/设置合约变量`name`。合约内容如下：

```
pragma solidity>=0.4.24 <0.6.11;
contract Mycontract {
    string name;

    constructor() public{
        name = "Welcome to my cotract!";
    }
    function get() public view returns (string memory){
        return name;
    }
    function set(string memory n) public{
        name = n;
    }
}
```

4.2 部署合约

把`Mycontract`合约放在控制台目录下`contracts/solidity/Mycontract.sol`下：

```
[group:1]> deploy Mycontract
transaction hash: 0x61c3c65b4184aa3257252640277d1ec57663c04f3a2bd1700ec052986a38aef7
contract address: 0xd3d2e026b060371cd0819baea9bf909d8a179484
```

返回了合约地址，部署成功。

4.3 调用合约

```
#部署合约，得到合约地址
[group:1]> deploy Mycontract
transaction hash:
0x61c3c65b4184aa3257252640277d1ec57663c04f3a2bd1700ec052986a38aef7
contract address: 0xd3d2e026b060371cd0819baea9bf909d8a179484
#查看当前块高（由于之前新建过一个，所以当前为2）
[group:1]> getBlockNumber
2
#调用合约get接口，得到name变量
```

```
[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 get
-------------------------------------------------------------------------------
-------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------
-------------
Return values:
[
    "Welcome to my cotract!"
]
-------------------------------------------------------------------------------
-------------
#查看当前块高，因为get接口不更改账本状态，所以块高不变，仍为2
[group:1]> getBlockNumber
2
#调用set设置变量name
[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 set
"Welcome"
transaction hash:
0xcb2ba2f274d36b6fdfa40ad3f42362303f833a989b38876f42adda98676d03d7
-------------------------------------------------------------------------------
-------------
transaction status: 0x0
description: transaction executed successfully
-------------------------------------------------------------------------------
-------------
Output
Receipt message: Success
Return message: Success
Return value: []
-------------------------------------------------------------------------------
-------------
Event logs
Event: {}
##因为账本状态改变，所以块高增加为3，表示已出块
[group:1]> getBlockNumber
3
#调用get接口查看更改是否生效
[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 get
-------------------------------------------------------------------------------
-------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------
-------------
Return values:
[
    "Welcome"
]
-------------------------------------------------------------------------------
---------
```

```
[group:1]> deploy Mycontract
transaction hash: 0x61c3c65b4184aa3257252640277d1ec57663c04f3a2bd1700ec052986a38aef7
contract address: 0xd3d2e026b060371cd0819baea9bf909d8a179484

[group:1]> getBlockNumber
2
```

```
[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 get
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "Welcome to my cotract!"
]
---------------------------------------------------------------------------------------------

[group:1]> getBlockNumber
2

[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 set "Welcome"
transaction hash: 0xcb2ba2f274d36b6fdfa40ad3f42362303f833a989b38876f42adda98676d03d7
---------------------------------------------------------------------------------------------
transaction status: 0x0
description: transaction executed successfully
---------------------------------------------------------------------------------------------
Output
Receipt message: Success
Return message: Success
Return value: []
---------------------------------------------------------------------------------------------
Event logs
Event: {}

[group:1]> getBlockNumber
3
```

```
[group:1]> call Mycontract 0xd3d2e026b060371cd0819baea9bf909d8a179484 get
---------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
---------------------------------------------------------------------------------------------
Return values:
[
    "Welcome"
]
---------------------------------------------------------------------------------------------
```

五、查看区块并解释

```
[group:1]> getBlockHeaderByNumber 0
BlockHeader{
    # number:区块号
    number=0x0,
    #hash：当前区块的哈希串
    hash='0x711cefead40fd9bee4ce57a33c4175f92b11d55f9312a1ad10c9fe4d568b3b3a',
    #parentHash：父区块的哈希值

 parentHash='0x0000000000000000000000000000000000000000000000000000000000000000'
,
    #logsBloom：区块日志的布隆过滤器

logsBloom='0x000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000',
```

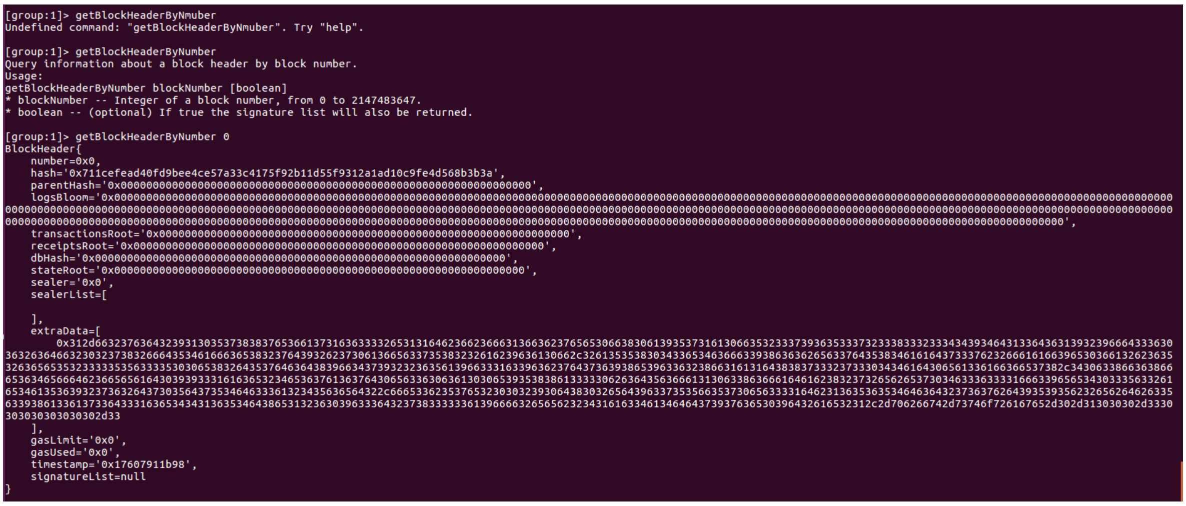
```
#transactionRoot：区块的交易前缀树的根
transactionsRoot='0x0000000000000000000000000000000000000000000000000000000000000000',
    #receiptsRoot：收据的根节点

 receiptsRoot='0x0000000000000000000000000000000000000000000000000000000000000000',
    #dbHash：分布式存储通过计算哈希值来记录一个区块中写入的数据，是FISCO
    dbHash='0x0000000000000000000000000000000000000000000000000000000000000000',
    #stateRoot：区块状态树的根哈希

 stateRoot='0x0000000000000000000000000000000000000000000000000000000000000000',
    #sealer：打包区块的节点在共识节点列表中的索引，FISCO BCOS新增字段
    sealer='0x0',
    #sealerList：区块的共识节点列表（不含观察节点），FISCO BCOS新增字段
    sealerList=[

    ],
    #extraData：当前块的其他信息
    extraData=[

0x312d663237636432393130353738383765366137316363333265313164623662366631366362376565306638306139353731613066353233373936353337323338333233343439346431336436313932396664333630363263646632303237383266643534616663653832376439326237306136656337353832326162396361306 2c326135353830343365346366633938636362656337643538346161643733376232666161663965303661326236353263656535323333353563333530306538326435376463643896634373932323635613966333163396362376437363938653963363238663161316438383733323733303434616430656133616636653738 2c343063386636386665363465666462366565616430393933316163653234653637613637643065633630636130306539353838613333306263643563666131306338636661646162383237326562653730346333633331666339656534303335633261653461353639323736326437303564373534646333613234356365643 22c666533623537653230323930643830326564396337353566353730656333316462313635363534646364323736376264393535623265626462633563393861336137336433316365343431363534643865313236303963336432373833333361396666326565623234316163346134646437393763653039643261653231 2c2d706266742d73746f726167652d302d313030302d333030303030303030302d33
    ],
    #gasLimit：当前区块允许的最大gas
    gasLimit='0x0',
    #gasUsed：当前区块累计使用的总gas
    gasUsed='0x0',
    #timestamp：时间戳
    timestamp='0x17607911b98',
    #签名
    signatureList=null
}
```



解释如下：

- **number**：区块号
- **hash**：当前区块的哈希串
- **parentHash**：父区块的哈希值
- **logsBloom**：区块日志的布隆过滤器
- **transactionRoot**：区块的交易前缀树的根
- **receiptsRoot**：收据的根节点
- **dbHash**：分布式存储通过计算哈希值来记录一个区块中写入的数据，是FISCO
- **stateRoot**：区块状态树的根哈希
- **sealer**：打包区块的节点在共识节点列表中的索引，FISCO BCOS新增字段
- **sealerList**：区块的共识节点列表（不含观察节点），FISCO BCOS新增字段
- **extraData**：当前块的其他信息
- **gasLimit**：当前区块允许的最大gas
- **gasUsed**：当前区块累计使用的总gas
- **timestamp**：时间戳
- **signatureList**：签名