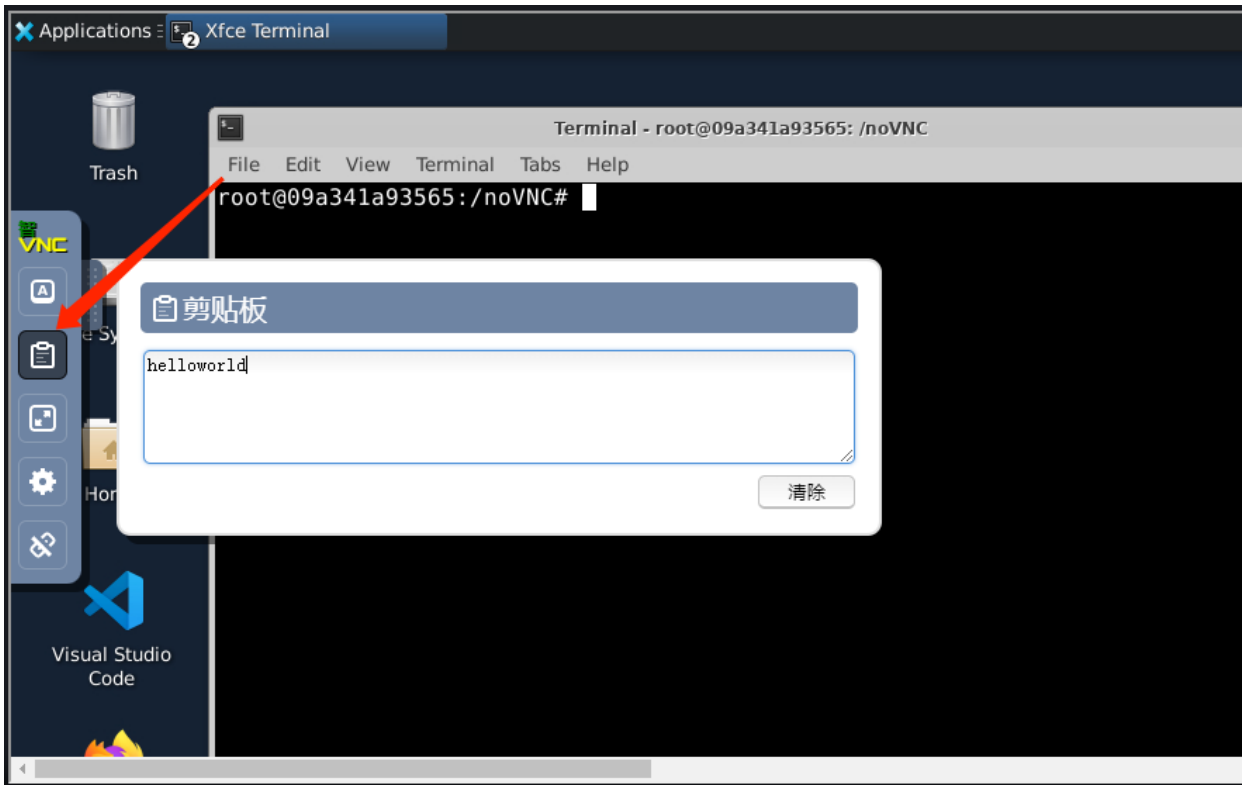
本章概述

熟悉FISCO BCOS控制台工具在应用开发中的使用场景

注：若想要把文字从镜像外复制到镜像内，需要将内容先复制粘贴到剪贴板，再在镜像中右键粘贴



一、联盟链搭建与启动

实验概述

掌握FISCO BCOS控制台搭建所需要进行的前置准备，包括搭建联盟链、启动联盟链、检查进程

任务步骤

1、进入实训界面
点击实训工具：

打开LX终端

2、搭建并启动FISCO BCOS联盟链

FISCO BCOS安装和状态检查

第一步. 安装依赖

开发部署工具 build_chain.sh 脚本依赖于openssl, curl，使用以下命令安装openssl和curl。

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

第二步. 进入FISCO BCOS目录

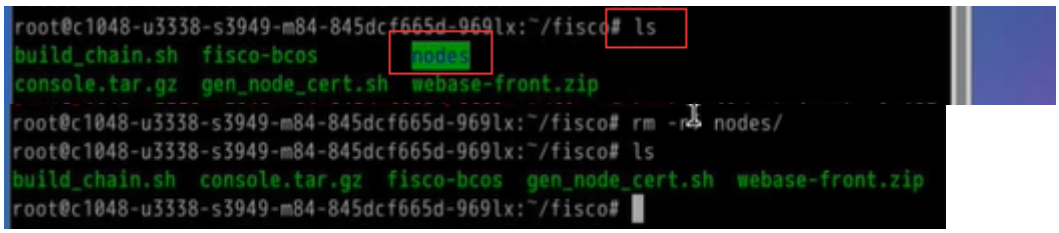
以下命令，回到root目录下的fisco目录，后续的操作都将在此目录下进行

```
cd /root/fisco
```

第三步. 搭建单群组4节点联盟链

在fisco目录下执行下面的指令，查看当前目录中是否存在nodes目录，如果存在，则删除此目录,如果不存在则跳过删除目录步骤。

```
ls
rm -rf nodes
```



在fisco目录下执行下面的指令，生成一条单群组4节点的FISCO链。 请确保机器的30300~30303，20200~20203， 8545~8548端口没有被占用。

```
bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545 -e ./fisco-bcos
```

其中-p参数指定起始端口，分别是p2p_port,channel_port,jsonrpc_port
出于安全性和易用性考虑，v2.3.0版本最新配置将listen_ip拆分成jsonrpc_listen_ip和channel_listen_ip，但仍保留对listen_ip的解析功能，详细请参考官方文档。

为便于开发和体验，channel_listen_ip参考配置是 0.0.0.0 ，出于安全考虑，请根据实际业务网络情况，修改为安全的监听地址，如：内网IP或特定的外网IP。

-e参数指定了本地二进制文件，可以避免从网络下载，加快部署速度。

命令执行成功会输出All completed，如下的信息。如果执行出错，请检查nodes/build.log文件中的错误信息。

```
Checking fisco-bcos binary...
Binary check passed.
================================================================
Generating CA key...
================================================================
Generating keys ...
Processing IP:127.0.0.1 Total:4 Agency:agency Groups:1
================================================================
Generating configurations...
Processing IP:127.0.0.1 Total:4 Agency:agency Groups:1
================================================================
[INFO] Execute the download_console.sh script in directory named by IP to get FISCO-
BCOS console.
e.g.  bash /fisco/nodes/127.0.0.1/download_console.sh
================================================================
[INFO] FISCO-BCOS Path    : bin/fisco-bcos
[INFO] Start Port         : 30300 20200 8545
[INFO] Server IP          : 127.0.0.1:4
[INFO] Output Dir         : /fisco/nodes
[INFO] CA Key Path        : /fisco/nodes/cert/ca.key
================================================================
[INFO] All completed. Files in /fisco/nodes
```

3、启动FISCO BCOS链

启动所有节点：

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

启动成功会输出：

```
root@2e0fd1663e5a:~/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
 node1 start successfully
 node0 start successfully
 node2 start successfully
 node3 start successfully
root@2e0fd1663e5a:~/fisco#
```

4、检查进程

检查进程是否启动：

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

正常情况会有类似下面的输出：

```
root@2e0fd1663e5a:~/fisco# ps -ef | grep fisco-bcos
root       746     1  0 09:38 pts/1    00:00:02 /root/fisco/nodes/127.0.0.1/node0/
../fisco-bcos -c config.ini
root       747     1  0 09:38 pts/1    00:00:02 /root/fisco/nodes/127.0.0.1/node1/
../fisco-bcos -c config.ini
root       750     1  0 09:38 pts/1    00:00:02 /root/fisco/nodes/127.0.0.1/node2/
../fisco-bcos -c config.ini
root       752     1  0 09:38 pts/1    00:00:01 /root/fisco/nodes/127.0.0.1/node3/
../fisco-bcos -c config.ini
root      1217   688  0 09:42 pts/1    00:00:00 grep --color=auto fisco-bcos
```

截至目前我们的FISCO BCOS联盟链已经启动完成了。

二、安装与启动控制台

实验概述

掌握控制台的安装与启动，为后续在控制台上进行智能合约的相关操作做准备

任务步骤

1. 第一步，安装依赖

为了同学们方便实验，已经为同学们安装了java，输入以下命令可以查看java版本：

```
java -version
```

```
root@317e15cc83a7:~/fisco# java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

2. 第二步，安装控制台

（1）回到 home 目录下的 fisco 子目录，安装控制台程序：

```
cd ~/fisco
tar -xzvf console.tar.gz
```

（3）解压完成后，拷贝控制台配置文件：

```
cp -n console/conf/config-example.toml console/conf/config.toml
```

```
root@317e15cc83a7:~/fisco# cp -n console/conf/config-example.toml console/conf/c
onfig.toml
```

（4）配置控制台证书，复制 BCOS 的证书到 console 的配置目录：

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

```
root@317e15cc83a7:~/fisco# cp -r nodes/127.0.0.1/sdk/* console/conf/
```

3. 第三步，启动控制台

（1）进入控制台程序所在目录，执行下面命令启动控制台：

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

（2）输出下述信息表明启动成功：

```
root@317e15cc83a7:~/fisco# cd ~/fisco/console && bash start.sh
=================================================================================
=====
Welcome to FISCO BCOS console(2.8.0)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
 _______  ______  ______  ______  ______    _______  ______  ______  ______
|       |      \/      \ /      \ /      \     |       \ /      \ /      \ /      \
| $$$$$$$\$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\    | $$$$$$$|  $$$$$$|  $$$$$$|  $$$$$$\
| $$__      | $$ | $$___\$| $$   \$| $$  | $$    | $$__/ $| $$   \$| $$  | $| $$___\$$
| $$  \     | $$  \$$    \| $$     | $$  | $$    | $$    $| $$     | $$  | $$\$$    \
| $$$$$     | $$  _\$$$$$$| $$   __| $$  | $$    | $$$$$$$| $$   __| $$  | $$_\$$$$$$\
| $$       _| $$_|  \__| $| $$__/  | $$__/ $$    | $$__/ $| $$__/  | $$__/ $|  \__| $$
| $$      |   $$ \\$$    $$\$$    $$\$$    $$    | $$    $$\$$    $$\$$    $$\$$    $$
 \$$       \$$$$$$ \$$$$$$  \$$$$$$  \$$$$$$      \$$$$$$$  \$$$$$$  \$$$$$$  \$$$$$$

=================================================================================
=====
```

三、应用开发场景下控制台常用命令展示

实验概述

熟悉应用开发场景下的控制台常用命令，如查看区块数量、生成新账户、部署与调用合约

任务步骤

1. 第一步，查看当前总区块数量

输入以下命令查看当前总区块数量，因为当前并没有产生交易，所以区块总数为 0：

```
getBlockNumber
```

```
[group:1]> getBlockNumber
0
```

2. 第二步，查看当前的账户列表

```
listAccount
```

可以看到系统的默认账户：

```
[group:1]> listAccount
0xecb8869fb18292c34164fc1f1c2016a5d828b9c6(current account) <=
```

3. 第三步，生成新账户

```
newAccount
```

生成了一个新账户：

```
[group:1]> newAccount
AccountPath: account/ecdsa/0x43713477e749f0d0175845fd5d729de0f6a926d3.pem
Note: This operation does not create an account in the blockchain, but only creates a local account,
 and deploying a contract through this account will create an account in the blockchain
newAccount: 0x43713477e749f0d0175845fd5d729de0f6a926d3
AccountType: ecdsa
```

4. 第四步，部署合约

为了方便用户快速体验，系统已经将HelloWorld合约内置于控制台中，位于控制台目录下contracts/solidity/HelloWorld.sol。

（1）HelloWorld合约的内容如下：

```
pragma solidity ^0.4.24;
contract HelloWorld {
     string name;
     function HelloWorld() {
         name = "Hello, World!";
     }
     function get()constant returns(string) {
         return name;
     }
     function set(string n) {
         name = n;
     }
}
```

（2）输入以下命令部署HelloWorld合约：

```
deploy HelloWorld
```

返回产生的交易哈希与合约地址：

```
[group:1]> deploy HelloWorld
transaction hash: 0xc2ca908ee9d41f52a479f0806b794b95277628f85b005d5e6f398caed7f64f6b
contract address: 0xcd0bc011c6c0fd5b2717a9ad337717fd1c887c6f
currentAccount: 0xecb8869fb18292c34164fc1f1c2016a5d828b9c6
```

5. 第五步，根据交易哈希获取交易详情

输入以下命令获取交易详情（注意：这里后面跟的交易哈希为上一步部署合约返回的交易哈希，每个人的都不一样）：

```
getTransactionByHash
0xc2ca908ee9d41f52a479f0806b794b95277628f85b005d5e6f398caed7f64f6b
```

返回交易详情：

```
[group:1]> getTransactionByHash 0xc2ca908ee9d41f52a479f0806b794b95277628f85b005d5e6f398caed7f64f6b
{
    blockHash='0x9a4df625a48b8b80398744198b2421e04a744d756595cbcbb13261de13e72b61',
    blockNumber='0x1',
    from='0xecb8869fb18292c34164fc1f1c2016a5d828b9c6',
    gas='0x419ce0',
    hash='0xc2ca908ee9d41f52a479f0806b794b95277628f85b005d5e6f398caed7f64f6b',
    input='0x6080604052348015610010576000080fd5b506040805190810160405280600d81526020017f48656c6c6f2c2
0576f726c6421000000000000000000000000000000000000000081525060009080519060200190610005c929190610062565b5
0610107565b828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f106
100a357805160ff191683800117855561100d1565b828001600101855582156100d1579182015b828111156100d0578251825
591602001919060010190610100b5565b5b5090506100de91906100e2565b5090565b61010491905b808211156101005760008
160009055506001016100e8565b5090565b90565b6102d78061011660000396000f300608060405260004361061004c5760003
57c01000000000000000000000000000000000000000000000000000000000000000900463ffffffff1680634ed3885e14610051578
0636d4ce63c146100ba575b600080fd5b34801561005d57600080fd5b506100b860048036038101908080359060200190820
180359060200190808060601f016020809104026020016040519081016040528093929190818152602001838380828437782019
150505050505091929192905050506101 4a565b005b3480156100c657600080fd5b506100cf610164565b604051808060200
182810382528381815181526020019150805190602001908083836000 5b8381101561010f578082015181840152602081019
0506100f4565b505050509050908101906601f16801561013c578082038051600183602003610100 0a031916815260200191 5
05b509250505060405180910390f35b806000908051906020019061016092919061020656 5b5050565b60606000805460018
160011615610100020316600290048060 1f016020809104026020016040519081016040528092919081815260200182805 46
0018160011615610100020316600290048015 6101fc5780601f106101d157610100808354040283529160200191610 1fc565
b820191906000526020600020905b815481529060010190602001808311610 1df57829003601f168201915b5050505050905
090565b828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061024
757805160ff191683800117855561027556 5b828001600101855582156102755791820 15b828111156102745782518255916
020019190600101906102595565b5b5090506102829190610286565b5090565b6102a891905b808211156102a457600081600
090555060010161028c565b5090565b905600a165627a7a72305820201856c90bd31c7b8e3a14e9362273298cb470d4c46fa
6dfe63453dec66ac1d00029',
    nonce='0x340fc29623329a32e04225e208ce3fa96683651aa145232954450c06bf03866',
```

6. 第六步，调用合约方法

输入以下命令调用HelloWorld合约中的get方法（注意：这里后面跟的地址为部署合约部署合约步骤中返回的合约地址，每个人的都不一样）：

```
call HelloWorld 0xcd0bc011c6c0fd5b2717a9ad337717fd1c887c6f get
```

返回结果：

```
[group:1]> call HelloWorld 0xcd0bc011c6c0fd5b2717a9ad337717fd1c887c6f get
-------------------------------------------------------------------------------------------
---
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
---
Return value size:1
Return types: (STRING)
Return values:(Hello, World!)
-------------------------------------------------------------------------------------------
---
```

7. 第七步，再次查看总区块数量

输入以下命令查看总区块数量，可以看到总区块数量变为 1，因为我们刚刚部署合约产生了交易，这一操作使得总区块数量加1：

```
getBlockNumber
```

```
[group:1]> getBlockNumber
1
```

8. 第八步，退出控制台

输入以下命令退出控制台：

```
quit
```

```
[group:1]> quit
root@08d1fc1a26c4:~/fisco/console#
```

拓展任务1

请同学们重新进入控制台，并使用help命令来查看其控制台的所有命令及其用法：

```
help
```

```
[group:1]> help
* help([-h, -help, --h, --H, --help, -H, h])  Provide help information
* addObserver                                 Add an observer node
* addSealer                                   Add a sealer node
* call                                        Call a contract by a function and parameters
* callByCNS                                   Call a contract by a function and parameters b
y CNS
* create                                      Create table by sql
* delete                                      Remove records by sql
* deploy                                      Deploy a contract on blockchain
* deployByCNS                                 Deploy a contract on blockchain by CNS
* desc                                        Description table information
* quit([quit, q, exit])                       Quit console
* freezeAccount                               Freeze the account
* freezeContract                              Freeze the contract
* generateGroup                               Generate a group for the specified node
* generateGroupFromFile                       Generate group according to the specified file
* getAccountStatus                            GetAccountStatus of the account
* getAvailableConnections                     Get the connection information of the nodes co
nnected with the sdk
* getBatchReceiptsByBlockHashAndRange         Get batched transaction receipts according to
block hash and the transaction range
* getBatchReceiptsByBlockNumberAndRange       Get batched transaction receipts according to
block number and the transaction range
* getBlockByHash                              Query information about a block by hash
* getBlockByNumber                            Query information about a block by number
* getBlockHashByNumber                        Query block hash by block number
* getBlockHeaderByHash                        Query information about a block header by hash
* getBlockHeaderByNumber                      Query information about a block header by bloc
k number
* getBlockNumber                              Query the number of most recent block
* getCode                                     Query code at a given address
* getConsensusStatus                          Query consensus status
* getContractStatus                           Get the status of the contract
* getCryptoType                               Get the current crypto type
* getCurrentAccount                           Get the current account info
* getDeployLog                                Query the log of deployed contracts
* getGroupConnections                         Get the node information of the group connecte
```

然后自行体验各命令的作用。

拓展任务2

请同学们退出控制台并使用以下命令进入solidity目录：

```
cd contracts/solidity/
```

```
root@1e4e0e50e32b:~/fisco/console# cd contracts/solidity/
root@1e4e0e50e32b:~/fisco/console/contracts/solidity#
```

使用以下命令创建一个名为SimpleStorage.sol的文件：

```
touch SimpleStorage.sol
```

```
root@1e4e0e50e32b:~/fisco/console/contracts/solidity# touch SimpleStorage.sol
root@1e4e0e50e32b:~/fisco/console/contracts/solidity#
```

输入以下命令，可以发现当前目录下存在一个名为SimpleStorage.sol的文件，这个文件就是我们刚刚创建的：

```
ls
```

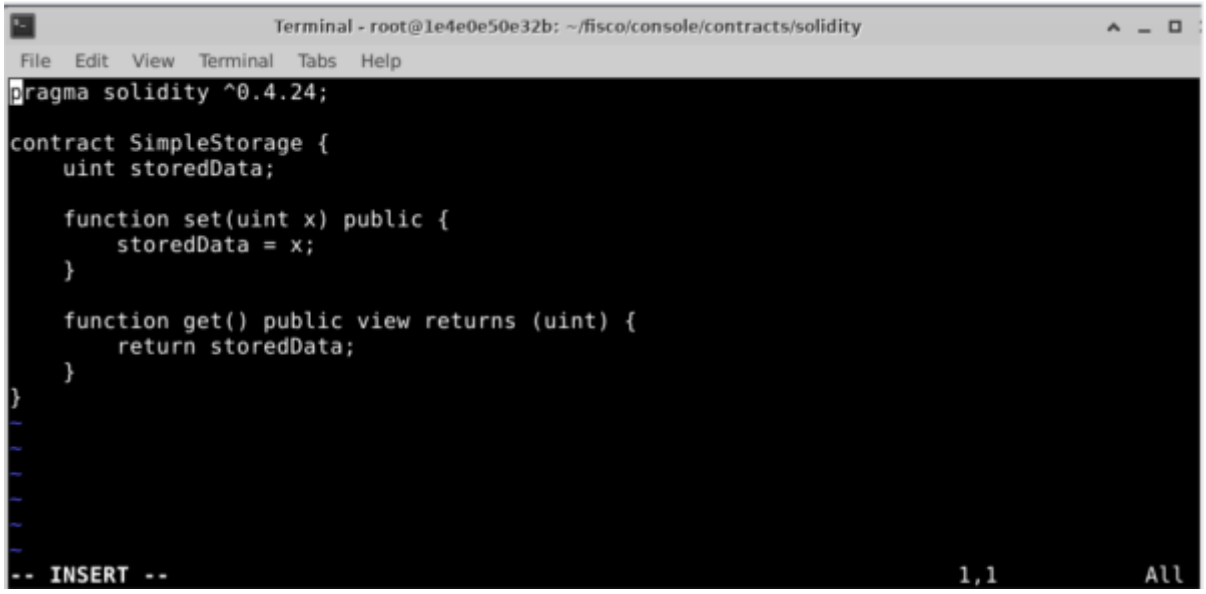
输入以下命令进入SimpleStorage.sol文件的编辑页面：

```
vi SimpleStorage.sol
```

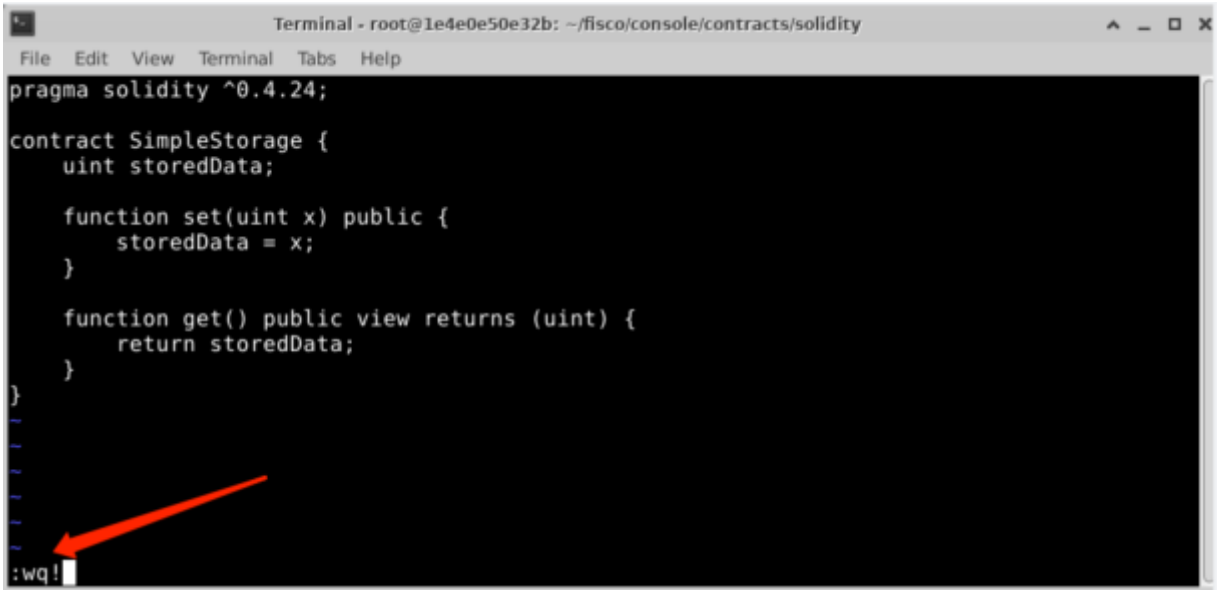
在vim编辑器界面先按下“i”键，把以下代码复制到粘贴板，再将代码粘贴到SimpleStorage.sol文件当中（此处使用右键选择粘贴）：

```
pragma solidity ^0.4.24;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

粘贴完成如下：



最后按 ” esc“键，再输入“:wq!”按下回车键保存并退出文件编辑：



现在请同学们重新执行第三节里面的所有步骤，部署我们新创建的 SimpleStorage.sol 合约，根据交易哈希获取交易详情，调用合约中的 get 方法等操作。

以上就是我们的实验内容。