**以太坊实践**

1. **以太坊开发环境搭建**
   1. Mac OS X系统安装go环境

官网链接：<https://golang.org/dl/>

官网下载最新的稳定版本：go1.11.2.darwin-amd64.pkg，直接双击完成安装，安装目录在/usr/local/go/下。

配置环境变量：

* vim ~/.bash\_profile

export PATH=$PATH:/usr/local/go/bin

查看是否配置成功

* go version

go version go1.11.2 darwin/amd64

1.2安装Node.js、Npm

官网链接：<https://nodejs.org/en/download/>

查看是否安装成功

* node -version或node –v

NPM是随同NodeJS一起安装的包管理工具。

* npm -version或npm -v
  1. 以太坊Ethereum安装
* apt install ethereum

查看以太坊是否安装成功

* geth version
  1. solc编辑器安装
* npm install -g solc

查看solc支持的功能

* solcjs --help

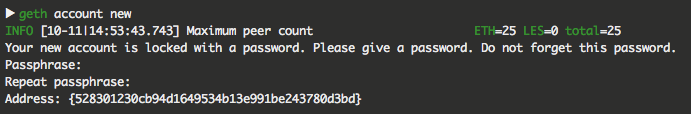
注意：solcjs命令与Solidity编译器提供的solc包并不兼容，因此不能通过eth.compile.solidity() RPC的方式与以太坊客户端结合使用。

**二、以太坊集成开发环境**

2.1创建以太坊账户

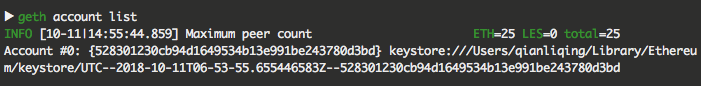
* geth account new

输入两遍密码后，生成账户地址：528301230cb94d1649534b13e991be243780d3bd



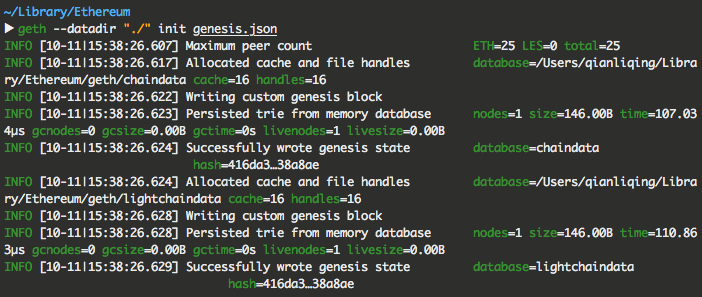
查看账户

* geth account list



2.2初始化创世块文件

* geth –datadir "./" init genesis.json

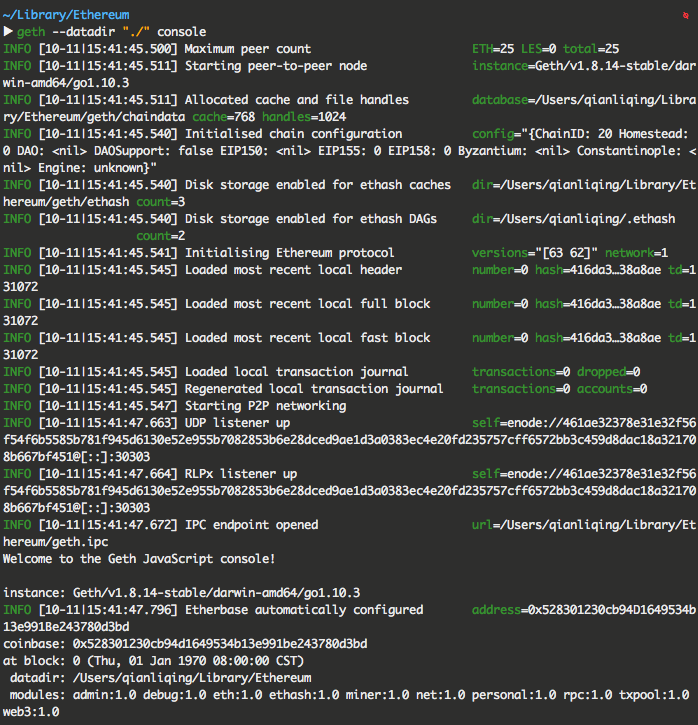


注意："./"表示以太坊的路径。

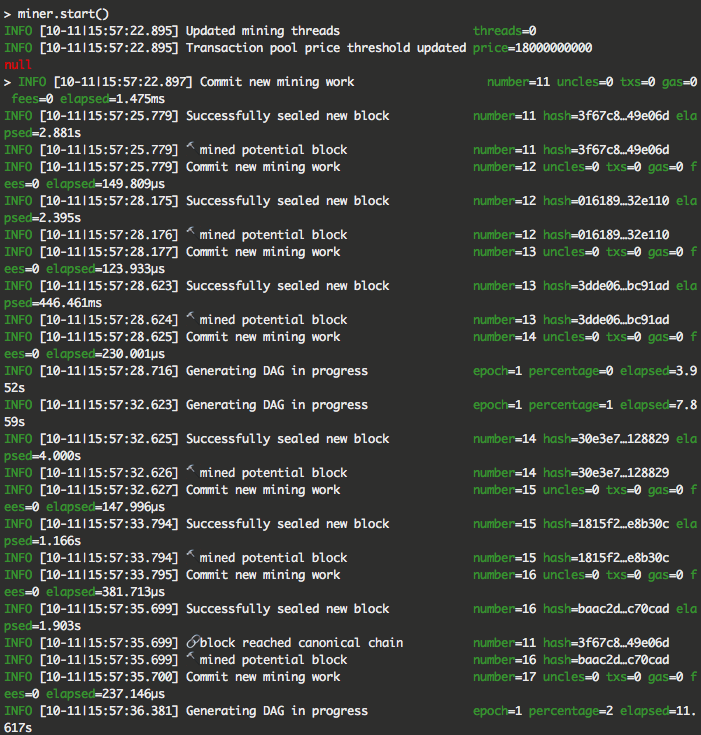
运行以太坊私有链，首先要定义自己的创世区块，创世区块信息写在一个 JSON 格式的配置文件中，即genesis.json。其中，chainID指定了独立的区块链网络ID。网络ID在连接到其他节点的时候会用到，以太坊公网的网络ID是1，为了不与公有链网络冲突，运行私有链节点的时候要指定自己的网络ID。不同ID网络的节点无法相互连接。配置文件还对当前挖矿难度difficulty、区块Gas消耗限制gasLimit等参数进行了设置。

2.3启动以太坊

* geth --datadir "./" console

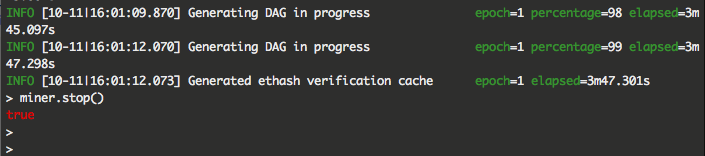


启动挖矿 miner.start()



其中start可以设置参数，表示挖矿使用的线程数。第一次启动挖矿会先生成挖矿所需的DAG文件，等进度达到100%后，就会开始挖矿。

停止挖矿miner.stop()

挖矿所得的奖励会进入矿工的账户，这个账户叫做coinbase，默认情况下coinbase是本地账户中的第一个账户，可以通过miner.setEtherbase()将其他账户设置成coinbase。另外注意第一次停止挖矿需要一段时间，因为节点需要为工作证明算法生成1GB数据集。

**三、以太坊编程接口**

3.1通过JSON-RPC编译合约

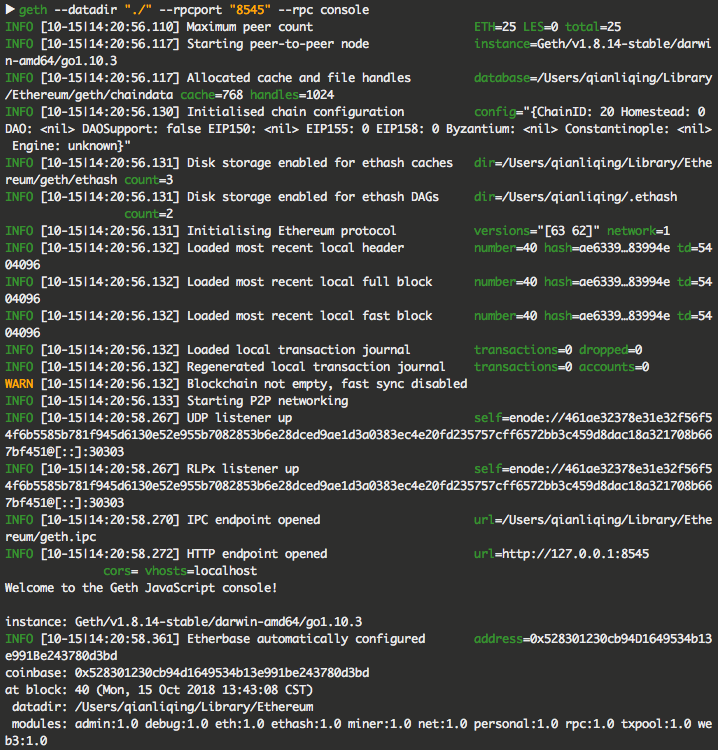
* curl --data '{"jsonrpc":"2.0","method": "eth\_compileSolidity", "params": ["contract Multiply7 {event Print(uint);function multiply(uint input) returns (uint) {Print(input \* 7);return input \* 7;}}"], "id": 5}' localhost:8545



提示port 8545: Connection refused

开启RPC端口

* geth --datadir "./" --rpcport "8545" --rpc console



编译合约

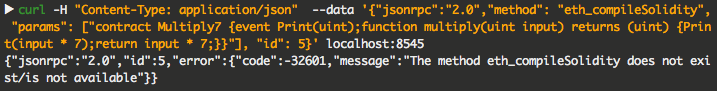
* curl --data '{"jsonrpc":"2.0","method": "eth\_compileSolidity", "params": ["contract Multiply7 {event Print(uint);function multiply(uint input) returns (uint) {Print(input \* 7);return input \* 7;}}"], "id": 5}' localhost:8545



提示only application/json is supported

指定Content-Type

* curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method": "eth\_compileSolidity", "params": ["contract Multiply7 {event Print(uint);function multiply(uint input) returns (uint) {Print(input \* 7);return input \* 7;}}"], "id": 5}' localhost:8545

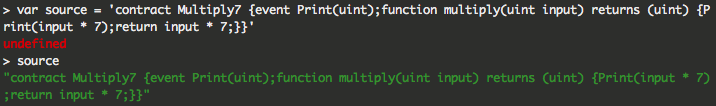


提示eth\_compileSolidity is not available，新版本中已不能通过JSON RPC部署合约，官方推荐用Remix进行合约的部署任务。

3.2 通过JavaScript API编译合约

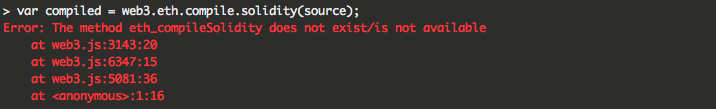
* var source = 'contract Multiply7 {event Print(uint);function multiply(uint input)

returns (uint) {Print(input \* 7);return input \* 7;}}'



编译合约

* var compiled = web3.eth.compile.solidity(source)

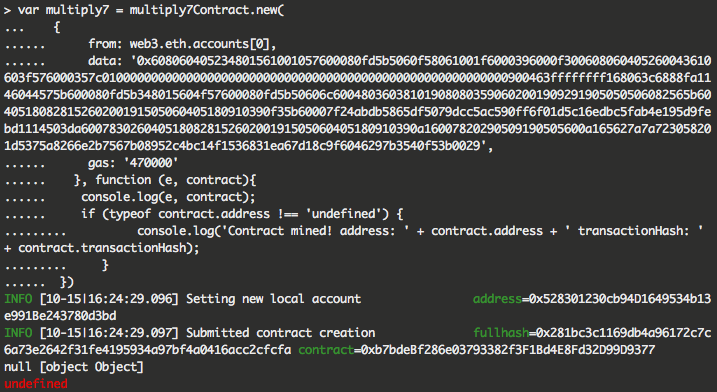


同样提示eth\_compileSolidity is not available。

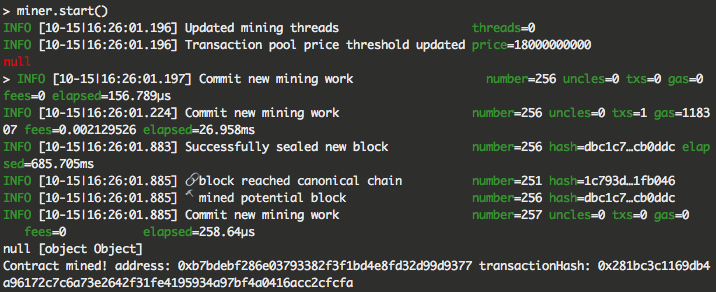
3.3通过在线Remix web3编译合约



部署合约



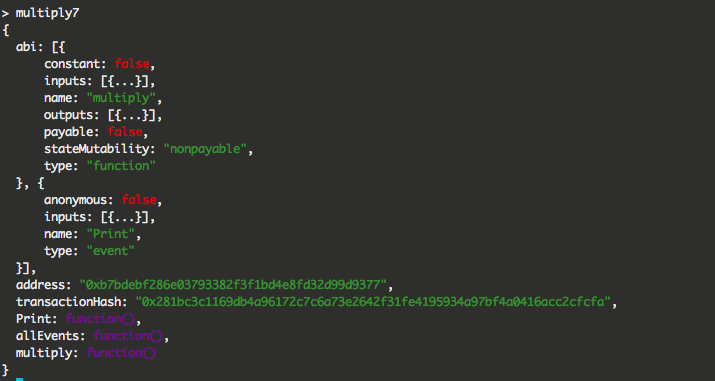
挖矿产生合约地址



合约地址：0x281bc3c1169db4a96172c7c6a73e2642f31fe4195934a97bf4a0416acc2cfcfa

查看multiply7合约

* multiply7



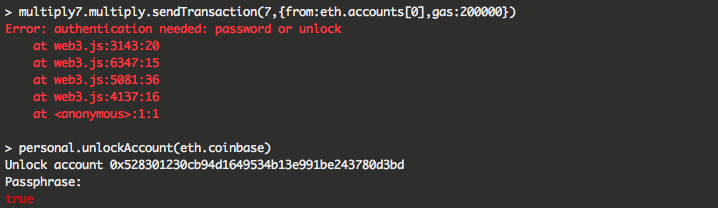
调用call()测试合约函数结果

* multiply7.multiply.call(uint);



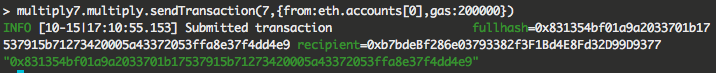
发起交易之前解锁账户

* personal.unlockAccount(eth.coinbase);



发起交易

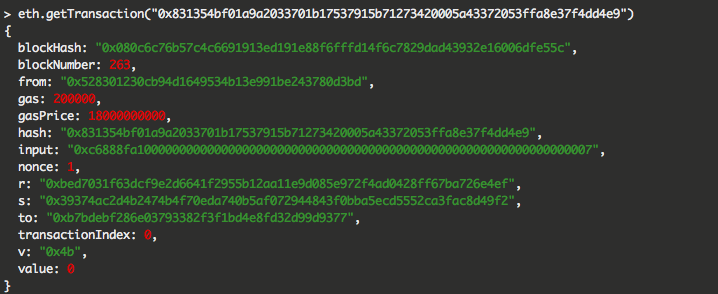
* multiply7.multiply.sendTransaction(7,{from:eth.accounts[0],gas:200000})



获取交易hash：0x831354bf01a9a2033701b17537915b71273420005a43372053ffa8e37f4dd4e9

查看交易

* eth.getTransaction("0x831354bf01a9a2033701b17537915b71273420005a43372053ffa8e37f4dd4e9")

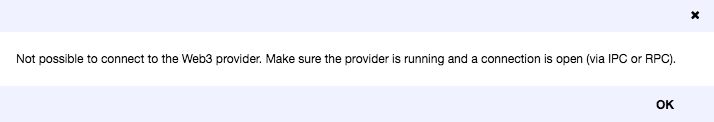


3.4通过在线Remix编译部署合约

本地私有链监听的RPC端口

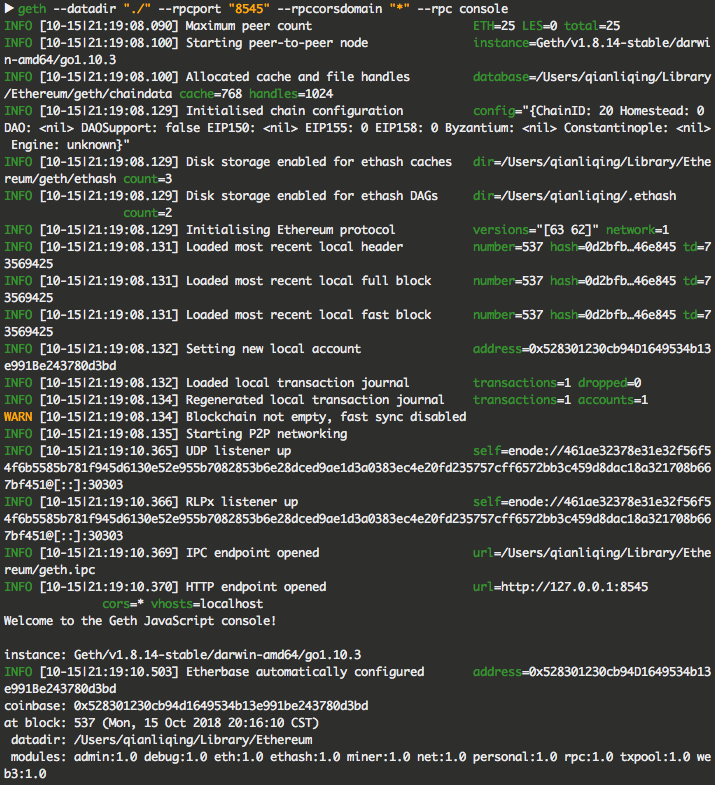


OK提示 is open (via IPC or RPC)

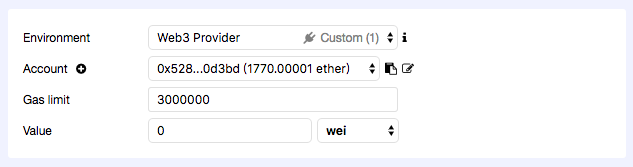


以web方式连接私有链，必须开启–rpccorsdomain参数，设置IP请求白名单 \* 为所有。

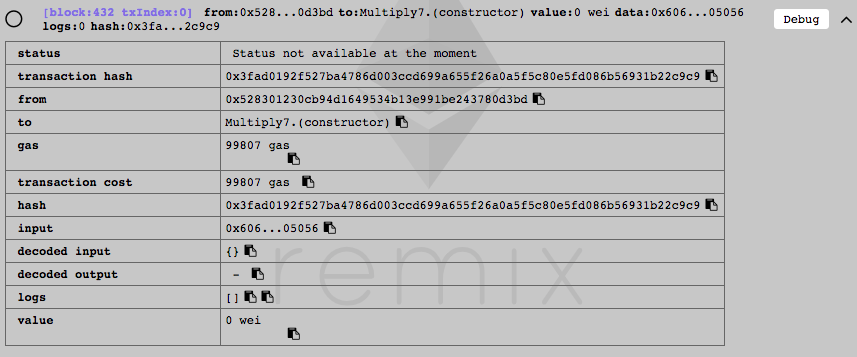
* geth --datadir "./" --rpcport "8545" --rpccorsdomain "\*" --rpc console



成功后读取Account列表



部署信息

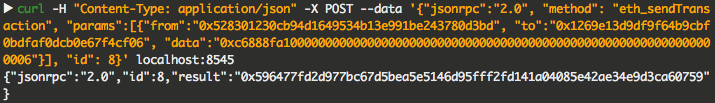


控制台查看合约部署的合约地址

../Desktop/address.png

发起交易

* curl -H "Content-Type: application/json" -X POST --data '{"jsonrpc":"2.0", "method": "eth\_sendTransaction", "params":[{"from":"0x528301230cb94d1649534b13e991be243780d3bd", "to":"0x1269e13d9df9f64b9cbf0bdfaf0dcb0e67f4cf06", "data":"0xc6888fa10000000000000000000000000000000000000000000000000000000000000006"}], "id": 8}' localhost:8545



提交交易

../Desktop/3.png

查看交易

* eth.getTransaction("0x596477fd2d977bc67d5bea5e5146d95fff2fd141a04085e42ae34e9d3ca60759")

