# **Homework 2**

1. Total points: **100**
2. Deadline: **Nov. 16 11:59pm**
3. 請將本份 google doc 文件下載成 word 格式 並依照指示將操作與答案填入紅色的部分
4. 將下載的檔案命名為 “學號\_姓名\_hw2.doc(x)”
5. 至作業繳交區上傳

## **01. Install Geth**

You will find the instructions for installing geth [here](https://geth.ethereum.org/docs/install-and-build/installing-geth).

## **02. Check the version of geth**

|  |
| --- |
| $ geth version Geth Version: 1.8.16-unstable Git Commit: 62e94895da80838ec694b4ba19eeac80c9dc88c1 Architecture: amd64 Protocol Versions: [63 62] Network Id: 1 Go Version: go1.11 Operating System: darwin GOPATH=/Users/changwu/golang GOROOT=/usr/local/opt/go/libexec |

**[Paste your geth version here.] (5%)**

**Geth**

**Version: 1.9.24-stable**

**Git Commit: cc05b050df5f88e80bb26aaf6d2f339c49c2d702**

**Git Commit Date: 20201112**

**Architecture: amd64**

**Protocol Versions: [65 64 63]**

**Go Version: go1.15.5**

**Operating System: windows**

**GOPATH=**

**GOROOT=go**

## **03. Run a local devnet**

|  |
| --- |
| $ geth --dev --dev.period 5 --rpc --rpcapi "eth,web3,personal" |

What do the above command and its options mean?

**[Please answer the question here.] (10%)**

1. --dev: [**dev 模式，也叫回歸測試模式，主要用來給開發人員提供一個方便的開發測試環境。在dev模式下，可以輕鬆的獲得以太幣，方便發起交易，交易也會被快速的打包，節省時間方便驗證。**]
2. --dev.period 5: [**開發者模式下挖礦週期 (0 = 有pending狀態交易時進行挖礦) (預設: 0)**]
3. --rpc: [**啟動HTTP-RPC服務，RPC預設的port是使用8545，可以用 –rpcport 更改**]
4. --rpcapi "eth,web3,personal": [**指定需要調用的HTTP-RPC API接口，默認有三：eth,net,web3**]

## **04. Geth client log**

**[Please answer the question here.] (15%)**

1. What is address? **For using a** [**faucet**](https://goerli-faucet.slock.it/) **to send ETH to one of your new account addresses to use for this guide.**
2. What is enode id in Ethereum? **The hexadecimal node ID is encoded in the username portion of the URL, and the username portion is a 512-bit public key that is used to verify communication came from a particular node on the network.**
3. What is geth.ipc used for?  **In order to connect geth via IPC.**
4. What is Ethereum HTTP or Websocket endpoint? **To make RPC requests over WebSockets or HTTP.** **Default WebSocket port is 8546, and the default HTTP port is 8545. Use curl in "JSON-RPC-http-endpoint:port", and "JSON-RPC-ws-endpoint:port" for wscat.**
5. What is Etherbase? **It’s an account for checking balances and mining profits.**

## **05. Attach to local geth console**

Before you connect to geth console, you need to open two command line windows. One is for running a geth client with the following command. Please keep it running.

|  |
| --- |
| $ geth --dev --dev.period 5 --rpc --rpcapi "eth,web3,personal" |

Another is to connect to geth console by IPC endpoint. First, you need to find the path of **geth.ipc** from the log printed by previous command window.

Tip: You can find something like

“IPC endpoint opened url=/var/folders/1t/jlxzt23s26b98wcc7dqp8t5m0000gn/T/geth.ipc“

|  |
| --- |
| $ geth attach /var/folders/x8/gs3w34\_n4wx6cmrp35b\_q4r40000gn/T/geth.ipc Welcome to the Geth JavaScript console!  instance: Geth/v1.8.16-unstable-62e94895/darwin-amd64/go1.11 coinbase: 0x049d00a5021081d9da04cbb52665dd0f33e9bd9b at block: 2 (Wed, 02 Oct 2019 11:43:25 CST)  datadir:  modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 shh:1.0 txpool:1.0 web3:1.0 > eth.coinbase "0x049d00a5021081d9da04cbb52665dd0f33e9bd9b" |

**[Please paste your operation here.] (5%)**

**instance: Geth/v1.9.24-stable-cc05b050/windows-amd64/go1.15.5**

**coinbase: 0x0f281ad6a1e675bac9f3beb19bdf0672953dffb9**

**at block: 3 (Sun Nov 15 2020 03:30:01 GMT+0800 (CST))**

**datadir:**

**modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0**

**> eth.coinbase**

**"0x0f281ad6a1e675bac9f3beb19bdf0672953dffb9"**

### **Questions**

#### **1. Please get the info of block number 10.**

|  |
| --- |
| > eth.getBlock(10) {  difficulty: 2,  extraData: "0xd783010810846765746886676f312e31318664617277696e00000000000000005e3ce39d3629e171a792f9480a0185cf34b49b1905892a48f77753093efe72762cbc56bbf83be61ef5dbc83e3823a67cf24501ea05e58e5774d1ebbf5491f63100",  gasLimit: 6344797,  gasUsed: 0,  hash: "0x704d97e9a1bfebcf1353d1daea46db32dd817a3d4ca68d45a1dfcb6baa5814c1",  logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",  miner: "0x0000000000000000000000000000000000000000",  mixHash: "0x0000000000000000000000000000000000000000000000000000000000000000",  nonce: "0x0000000000000000",  number: 10,  parentHash: "0xdde6787c8465b3a1abbcc48c6a24cbc0c46eacfa594d5577fa71aadb37c6b396",  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",  size: 606,  stateRoot: "0x4931658a78fa51c77c7f0ab095389937da3340adf6434c62eb7de64362a9aebb",  timestamp: 1569987845,  totalDifficulty: 21,  transactions: [],  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",  uncles: [] } |

**[Paste your result here.] (5%)**

**{**

**difficulty: 2,**

**extraData: "0xda83010918846765746888676f312e31352e358777696e646f7773000000000062e43baed26290c92a6be160c4d3766b1f8c01580365c1914a0381c7374a97ce5fbfdf2d99620d7d62c717d2b494ae59d27aa3973166c04f99bbf035f9cb0d3d01",**

**gasLimit: 11388203,**

**gasUsed: 0,**

**hash: "0xe5f412745e96ed4412973caaaba8691364d74ddf47a7f61c203493a2a26018e8",**

**logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",**

**miner: "0x0000000000000000000000000000000000000000",**

**mixHash: "0x0000000000000000000000000000000000000000000000000000000000000000",**

**nonce: "0x0000000000000000",**

**number: 10,**

**parentHash: "0x96c2f97d1198954ea7bcf84ab1a7373fa214caafd88b1cf0f390fb29c7630860",**

**receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",**

**sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",**

**size: 606,**

**stateRoot: "0xf74b9009f0c69c935be323246f9c28ba01ea7b85f0217f97834f6530d13f745d",**

**timestamp: 1605382236,**

**totalDifficulty: 21,**

**transactions: [],**

**transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",**

**uncles: []**

**}**

#### **2. Please get the latest block info.**

**[Paste your command and result here.] (5%)**

**> eth.getBlock("latest")**

**{**

**difficulty: 2,**

**extraData: "0xda83010918846765746888676f312e31352e358777696e646f77730000000000da04bfab3f2eb52e9d21f251d827b7b4524d7081d4b670b7aa0b26b9526cfb1e57603d7144af93362ea8bdf8ee8259ca50ec92a4eb4440ddd28b62a72c35d58100",**

**gasLimit: 10919728,**

**gasUsed: 0,**

**hash: "0xde2ac1372d0dcea93555230d477b49b0dd5d7c9dee32de9dca524751d8217a79",**

**logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",**

**miner: "0x0000000000000000000000000000000000000000",**

**mixHash: "0x0000000000000000000000000000000000000000000000000000000000000000",**

**nonce: "0x0000000000000000",**

**number: 53,**

**parentHash: "0x6556dcd62034f8803ec35f811a0f80e6c6f81045949685faf6168894d41aaca6",**

**receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",**

**sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",**

**size: 606,**

**stateRoot: "0xf74b9009f0c69c935be323246f9c28ba01ea7b85f0217f97834f6530d13f745d",**

**timestamp: 1605382451,**

**totalDifficulty: 107,**

**transactions: [],**

**transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",**

**uncles: []**

**}**

#### **3. Create a new account with password "nccu"**

**[Paste your command and result here.] (5%)**

**> personal.newAccount("nccu")**

**"0x6a5568a86fe2da0cfc3b7147a14e41338c845dd5"**

#### **4. Please check the balance of eth.coinbase.**

|  |
| --- |
| > **web3.fromWei(**eth.getBalance(eth.coinbase), 'ether') |

**[Paste your command and result here.] (5%)**

**> web3.fromWei(eth.getBalance(eth.coinbase), 'ether')**

**1.15792089237316195423570985008687907853269984665640564039457584007913129639927e+59**

#### **5. Please check the balance of your new account.**

|  |
| --- |
| > **web3.fromWei(**eth.getBalance(**"YOUR\_NEW\_ACCOUNT\_ADDRESS"), 'ether')** |

**[Paste your command and result here.] (5%)**

**> web3.fromWei(eth.getBalance("0x6a5568a86fe2da0cfc3b7147a14e41338c845dd5"), 'ether')**

**0**

#### **6. Transfer 100 ETH from eth.coinbase to your new account.**

Tip: use eth.sendTransaction

**[Paste your command and result here.] (10%)**

**> eth.sendTransaction({from: eth.coinbase, to: "0x6a5568a86fe2da0cfc3b7147a14e41338c845dd5", value: "15000000000000000"})**

**"0xdf0749ea660a71aeb76b261a7c44e5c6c96786119751727a91f9ba4f03180fe4"**

#### **7. Check the transaction status.**

Tip: use eth.getTransactionReceipt

**[Paste your command and result here.] (10%)**

**> eth.getTransactionReceipt("0xdf0749ea660a71aeb76b261a7c44e5c6c96786119751727a91f9ba4f03180fe4")**

**{**

**blockHash: "0xad4b5191d6a5211aaa902b01e1d8b0f63dc9d266baf1ec6abf8648a3b01d4c5b",**

**blockNumber: 243,**

**contractAddress: null,**

**cumulativeGasUsed: 21000,**

**from: "0x0f281ad6a1e675bac9f3beb19bdf0672953dffb9",**

**gasUsed: 21000,**

**logs: [],**

**logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",**

**status: "0x1",**

**to: "0x6a5568a86fe2da0cfc3b7147a14e41338c845dd5",**

**transactionHash: "0xdf0749ea660a71aeb76b261a7c44e5c6c96786119751727a91f9ba4f03180fe4",**

**transactionIndex: 0**

**}**

## **Other questions**

1. What is mainnet and testnet? (10%) **A Testnet is an Ethereum blockchain that uses identical technology and software as the “Mainnet” Ethereum blockchain. However, whereas the Mainnet network is used for “actual” transactions with “value”, Testnets are used for testing smart contracts and decentralized applications (“DApps”).**
2. How many testnets in Ethereum? (10%) **There are three testnets currently in use. Ropsten, Kovan, Rinkeby.**

## **References**

1. <https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options>
2. <https://github.com/ethereum/go-ethereum/wiki/Managing-your-accounts>
3. <https://github.com/ethereum/wiki/wiki/JSON-RPC>