DA51 Lab session 1

Creating a Blockchain

J. Gaber, gaber@utbm.fr

Target set of this session:
After completion of this session, you will successfully know:

- how to create a basic blockchain.
- how to obtain information about a blockchain using the command line.
- how to test blockchain operability by creating a blockchain transaction.

In what follows, we are going to use an open source blockchain. MultiChain is a platform designed for creating and deploying private blockchains. It's built on the Bitcoin core and has been extensively modified to support the creation of private and permissioned blockchain networks. MultiChain provides multiple asset financial transactions, based on the Bitcoin platform, and the source codes is available on GitHub.

You have 4 options to complete this lab exercise:
- On a UTBM machine, where you have VirtualBox that allows you to create multiple virtual machines (i.e., nodes) and set up a network among these virtual machines.
- Under Windows using the command line (Cmd or PowerShell)
- Under WSL (Windows Subsystem for Linux)
- By combining Windows and WSL, especially for simulating two nodes on your local machine.

Creating a Blockchain

1. Go to the MultiChain web site https://www.multichain.com/ to download and install the software for your system, on either Linux, Windows, or Mac.

2. You have got a zip file with five files, four executables and one README. The README contains the information that's useful for running your OS version you just downloaded. Browse this file to see its contents about resource links.

3. To install it, create a new folder, call it Multichain.

4. Copy the five files over from the zip file and pastes them in the Multichain folder, and that's it, it's installed.

5. Open a terminal or Windows command prompt, change to the multichain directory.

6. Your program to create a blockchain is multchain-util.exe program. Create a blockchain and call it myChain. Usage: multichain-util create myChain.

7. Examine the message you get. It’s telling that the blockchain has been created, and there is some information there about where the parameters are located. Take a look at params.dat file at its location ./multichain (you can modify it when you get accustomed to MultiChain…)

8. Then we have to generate the blockchain and start the daemon, using the name of your blockchain, myChain and -daemon as the switch. Usage: multichaind myChain -daemon

9. Examine the feedback and answer the following questions:
    a. Did this command start a node?
    b. did it also mine the very first block?
    c. how we this block is called?
    d. did you get a node address? Describe its format?

10. you have to remember the command that indicates how other nodes can connect to this node (copy it to paste it later)

Actually, we need two system instances with MultiChain installed (I will explain during the session how to get around this issue on a single system with windows 10 and WSL, for Linux see here after).

If you have two systems, you can hop over to system 2, and enter the command indicated to let other nodes to connect to the node of your system 1.

On a single system, to work around the problem, we 'll create the second node as follows, using a different datadir, port and rpcport.

11. Create a new folder, and call it .multichainother

12. Then, use the following command:

```
./multichaind -datadir=~/.multichainother -port=7730 -rcport=7729 myChain@ip:port
```

13. What have you got here as a command feedback?

14. You will see a message about needing connect permissions, so assign these via the regular node (or your system 1). Copy this line and paste it there
    ```
    multichain-cli myChain grant Wallet_ID connect,send,receive
    ```

15. What do you have as feedback?

16. Now, as we've granted permissions from the originating system for the other system go back to system 2 and type the following command:
    ```
    multichaind -daemon
    ```
    or in the case of one system the command:
    ```
    multichaind -datadir=.multichainother -port=7730 -rpcport=7729 myChain -daemon
    ```

17. So, at this point your blockchain is up and running and ready to process transactions. Check that's what you've got so far, using ps/tasklist command line (Linux,/windows)?

Obtaining Blockchain Information

We have so far, a blockchain called myChain, running on two nodes. Both nodes are running the multichain daemon. Describe the feedback of each the following command:

18. multichain-cli myChain getinfo

19. multichain-cli myChain help

20. multichain-cli myChain getblockchainparams

21. multichain-cli myChain listpermissions

22. ./multichain-cli myChain getaddresses

23. ./multichaind -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain - daemon

24. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain getinfo

25. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain listpermissions

26. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain getaddresses

27. ./multichain-cli myChain getnewaddress

28. ./multichain-cli myChain getaddresses

29. ./multichain-cli myChain getpeerinfo

30. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain getpeerinfo

31. Add a third node to the blockchain myChain and provide general information on the blockchain?

Creating a blockchain transaction

We have a blockchain called myChain and three systems with nodes running the Multichain Daemon. We want to create our first blockchain transaction.

32. start by obtaining the address of systems that have permissions to create assets:
    ./multichain-cli myChain listpermissions issue

33. Describe what you get?

Let's pretend that we're dealing with cryptocurrency called myCryptoMoney, and we're going to subdivide the currency into 100 units.

34. Create a new asset, give 500 units with the following command:

```
./multichain-cli myChain issue 1BwYFhoRDmu335ns22rHHtmF1rvTHXS2ArofJs myCryptoMoney 500 0.01
```

35. Describe this command line and its feedback?

36. Check that the transaction has been created and make sure it has indeed been added to the wallet :
    a. ./multichain-cli myChain listassets
    b. ./multichain-cli myChain gettotalbalances

37. Check the system 2 :
    a. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain listassets
    b. ./multichain-cli -datadir=~/.multichainother -port=7730 -rpcport=7729 myChain gettotalbalances

38. Do you get the same result as system 1? why?

39. Give the command to send 50 units from system 1 to system'2 walllet?

Hint : ./multichain-cli myChain sendasset wallet_ID myCryptoMoney 50

40. Give the command to check out the system's wallet on system 1?

41. Give the command to check out the system's wallet on system 2?

42. As each node has a ledger, provide and check the commands and their outputs to create transactions between the third node and each of the other nodes to send 30 units of myCryptoMoney to system 1 and to receive 130 units from system 2?