Synopsis

my_blockchain

Description

Blockchain is a command that allows for the creation and management of a blockchain. When the program starts (it loads a backup if there is one) then a prompt appears.
This prompt allows to execute commands. When the commands are successful they display "**ok**" and if not, "**nok:** ***info***" or *info* is an error message - see below:

add node ***nid*** add a *nid* identifier to the blockchain node.

rm node ***nid...*** remove nodes from the blockchain with a *nid* identifier. If **nid** is '*', then all nodes are impacted.

add block ***bid nid...*** add a *bid* identifier block to nodes identified by *nid*. If **nid** is '*', then all nodes are impacted.

rm block ***bid...*** remove the *bid* identified blocks from all nodes where these blocks are present.

ls list all nodes by their identifiers. The option **-l** attaches the blocks bid's associated with each node.

sync synchronize all of the nodes with each other. Upon issuing this command, all of the nodes are composed of the same blocks.

quit save and leave the blockchain.

The blockchain prompt must display (see example below):

a **[** character

a first letter that indicates the state of synchronization of the chain:

"**s**" if the blockchain is synchronized

"**-**" if the blockchain is not synchronized.

n number of nodes in the chain.

the "**]>** " string (with a space)

Error messages

1: no more resources available on the computer
2: this node already exists
3: this block already exists
4: node doesn't exist
5: block doesn't exist
6: command not found

Technical Information

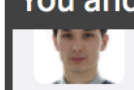
```
$>my_blockhain
[s0]> add node 12
[s1]> add block 21 *
[s1]> add node 13
[-2]> sync
[s2]> ls -l
12: 21,
13: 21,
[s2]> quit
```

1. you must create a Makefile, and the output is the command itself
2. NID is an integer, BID is a string
3. You can use:

   - malloc(3)
   - free(3)
   - printf(3)
   - write(2)

You and Lucas Robert

Jun 22



Hi Lucas!

Can you give me few hints for my_blockchain to get started. I understood the logic behind it and did some reading. 14:09

Do we need to come with hashing algorithm, what about proof of work? 14:09

I guess the idea for this exercise is to understand the linked lists, right? Should we use Linked lists as our data structure for data blocks? 14:12

Lucas Robert 15:10

You don't actually have to implement a real blockchain, so don't worry about hashing and proof of work. As you said, the idea is to manipulate some linked lists, see how you can store data on a file, and create a basic command line interface

Nurdos Omarkulov (EDITED) 19:56

Could You please elaborate more on what you said or suggest any good resources how to implementing blockchain in C

Lucas Robert (EDITED) 20:04

> Could You please elaborate more on what you said or suggest any good resources how to implementing blockchain in C

Yep, of course. What we basically did with my teammate was to implement two linked list: one for node, and another one for blocks.
Our data structure looked like this

```
[Node 1] ------------------->    [Node 2] ---------------------------> [Node 3]
   |                                |                                    |
[Block A]-> [Block B]              [Block A]-> [Block B]                [Block A]-> [Block B]
```

You don't have to implement hash, or proof of work. The idea is to understand how linked list work. You just have to add/remove node to the list, or add/remove blocks to the other one, according to the command passed

JUN 22 YESTERDAY