TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

LAB1:

Simplified Blockchain Implementation and Verification

GV hướng dẫn:

Nguyễn Đình Thúc

Ngô Đình Hy

Nhóm sinh viên thực hiện: 07

20127066 - Nguyễn Nhật Quân

20127192 - Trần Anh Huy

20127299 – Trần Hoàng Minh Quang

20127338 – Trương Gia Thịnh

Thành phố Hồ Chí Minh, ngày 26 tháng 11 năm 2023

MỤC LỤC

1. TỔNG QUAN

1.1. THÔNG TIN NHÓM

MSSV	Họ tên	Email	Vai trò
20127066	Nguyễn Nhật Quân	20127066@student.hcmus.edu.vn	Thành viên
20127192	Trần Anh Huy	20127192@student.hcmus.edu.vn	Nhóm trưởng
20127299	Trần Hoàng Minh Quang	20127299@student.hcmus.edu.vn	Thành viên
20127338	Trương Gia Thịnh	20127338@student.hcmus.edu.vn	Thành viên

1.2. THÔNG TIN BÀI TẬP NHÓM

Tên bài tập	Lab 1: Simplified Blockchain Implementation and Verification
Công cụ	GitHub, Google Meet, Google Doc, Visual Code
Ngôn ngữ lập trình	Go lang
Product Owner	Nguyễn Đình Thúc, Ngô Đình Hy

2 BẢNG ĐÁNH GIÁ VÀ PHÂN CÔNG

2.1. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH: 99%

Đồ án hoàn thành đủ các yêu cầu đề bài.

Phần	Các công việc đã hoàn thành
1	The chain of blocks: Implement a chain of blocks as an ordered, back-linked list data structure.
2	Efficient transactions and blocks verification: Implement an efficient way to verify membership of certain transactions in a block using Merkle Trees.
3	Command line client and benchmarks
4	Report

2.2. BẢNG % ĐÓNG GÓP

Các thành viên đều tham dự các buổi họp trực tuyến, có tinh thần tích cực và đóng góp ý kiến cho đồ án.

MSSV	Tên	% đóng góp
20127066	Nguyễn Nhật Quân	100%
20127192	Trần Anh Huy	100%
20127299	Trần Hoàng Minh Quang	100%
20127338	Trương Gia Thịnh	100%

3. MÔ HÌNH LÝ THUYẾT

3.1. BLOCKCHAIN

3.1.1. Block

Khối Trong blockchain là các khối lưu trữ thông tin có giá trị. Ngoài ra, khối còn chứa các thông tin kỹ thuật, như phiên bản của nó, dấu thời gian hiện tại và hàm băm của khối trước đó.

Mỗi khối được liên kết với khối trước đó bằng hàm băm. Các cách tính toán băm là một tính năng rất quan trọng của blockchain và cũng chính tính năng này giúp blockchain bảo đảm an toàn. Vấn đề là việc tính toán hàm băm là một thao tác khó về mặt tính toán và phải mất một thời gian ngay cả trên các máy tính nhanh. Đây là một thiết kế kiến trúc có chủ ý của các hệ thống blockchain, khiến việc thêm các khối mới trở nên khó khăn, do đó ngăn chặn việc sửa đổi chúng sau khi chúng được thêm vào

Chúng ta cũng muốn tất cả các giao dịch trong một khối được xác định duy nhất bằng một hàm băm duy nhất. do đó, ta sẽ nhận giá trị băm của mỗi giao dịch, ghép nối chúng và nhận được giá trị băm của kết hợp. hàm băm của

3.1.2. Blockchain

Về bản chất, blockchain là một cơ sở dữ liệu có cấu trúc nhất định: nó là một danh sách được liên kết ngược, có thứ tự. Điều đó có nghĩa là các khối được lưu trữ theo thứ tự chèn và mỗi khối được liên kết với khối trước đó. Cấu trúc này cho phép nhanh chóng lấy khối mới nhất trong chuỗi và lấy khối bằng hàm băm của nó.
Vì mọi khối cần phải được liên kết với khối trước đó, nên để thêm khối mới, chúng ta cần một khối hiện có, nhưng ngay từ đầu không có khối nào trong chuỗi khối. Vì vậy, trong bất kỳ chuỗi khối nào cũng phải có ít nhất một khối và khối đó là khối đầu tiên trong chuỗi và được gọi là khối gốc.

3.2. PROOF-OF-WORK

Proof of Work (PoW) là một phương pháp để xác minh và xác nhận các giao dịch trong các mạng blockchain, đặc biệt là trong việc tạo và xác minh các khối mới trong mạng Bitcoin và một số loại khác của tiền điện tử.

Trong hệ thống PoW, các thợ đào (miners) phải thực hiện công việc tính toán cực kỳ phức tạp để tìm ra một giá trị hash (băm) thỏa mãn một số điều kiện cụ thể. Quá trình tìm ra giá trị hash này đòi hỏi sử dụng năng lượng tính toán lớn và thời gian để tạo ra một "bằng chứng" rằng họ đã tiêu tốn một lượng tài nguyên để thực hiện công việc này.

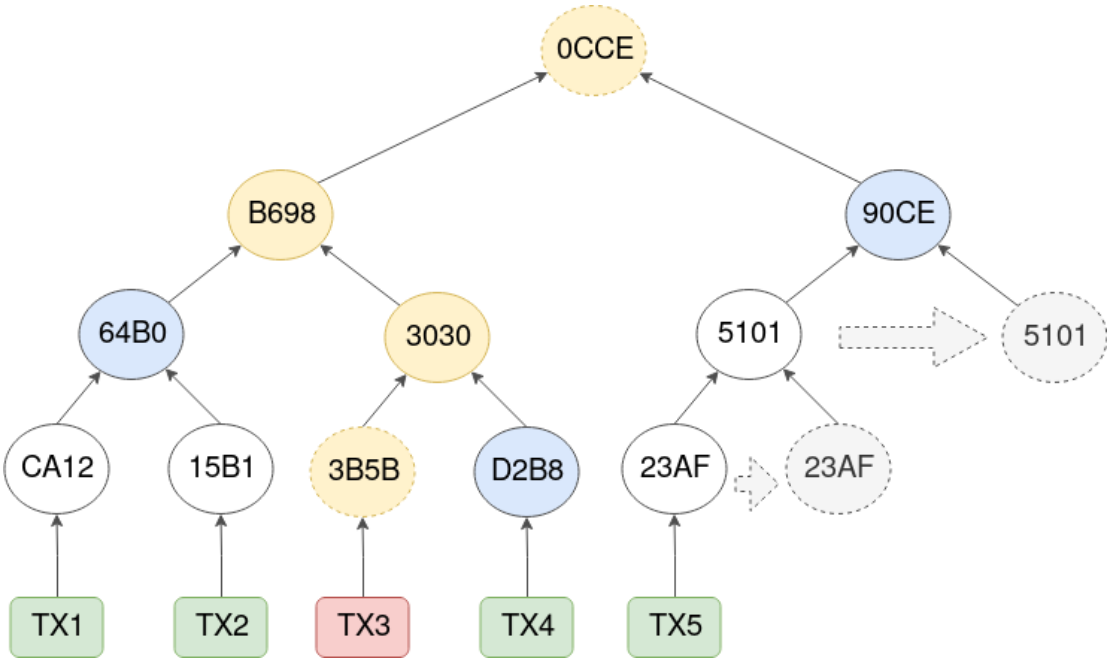
Khi một thợ đào tìm ra một giá trị hash hợp lệ, họ có thể tạo một khối mới chứa các giao dịch cùng với giá trị hash này. Công việc tính toán của PoW được xem như là một "chứng minh" cho việc thợ đào đã dành một lượng công sức và tài nguyên để tạo ra khối mới, và điều này giúp mạng blockchain chấp nhận khối đó như là một phần của chuỗi.

3.3. MERKEL TREE

Cấu trúc cây này cung cấp một cơ chế an toàn và hiệu quả để tóm tắt và xác thực tính toàn vẹn của các tập dữ liệu lớn. Trong blockchain, chúng đặc biệt quan trọng trong việc tổng hợp và xác minh các giao dịch, đảm bảo độ tin cậy và tính mạnh mẽ của tính toàn vẹn dữ liệu của blockchain
Cây Merkle được Bitcoin sử dụng để lấy hàm băm giao dịch, sau đó được lưu trong tiêu đề khối và được xét theo hệ thống proof-of-work. Lợi ích của cây Merkle là một nút có thể xác minh tư cách thành viên của một giao dịch nhất định mà không cần tải xuống toàn bộ khối, chỉ sử dụng hàm băm giao dịch, hàm băm gốc của cây merkle và một tập hợp các hàm băm trung gian cần thiết để xây dựng lại đường dẫn Merkle cho giao dịch đó , được gọi là merkle proof. Đường dẫn Merkle là tập hợp các giá trị băm từ giao dịch tại nút lá đến nút gốc Merkle. merkle proof là một cách chứng minh rằng một giao dịch nhất định là một phần của cây merkle mà không yêu cầu hiển

thị bất kỳ giao dịch nào khác, chỉ cần hiển thị các giá trị băm. Mỗi hàm băm trong bằng chứng là anh em của hàm băm trong đường dẫn ở cùng cấp độ trong cây.

Cây Merkle được xây dựng cho mỗi khối và bắt đầu bằng các lá (phần dưới của cây), trong đó một lá là hàm băm giao dịch. Trong Cây Merkle nhị phân tiêu chuẩn, mỗi nút bên trong có hai nút con và tất cả các lá có cùng độ sâu, nhưng không phải mọi khối đều chứa số lượng giao dịch chẵn. Trong trường hợp có số lượng giao dịch lẻ, hàm băm của giao dịch cuối cùng sẽ được sao chép trong Cây để tạo thành Cây Merkle nhị phân đầy đủ, trong đó mỗi nút có 0 hoặc 2 con.



Hình minh họa Merkle tree 1

3.4. DATA STRUCTURE

Cấu trúc của Cây Merkle có thể được hiểu ở ba thành phần chính:

- Nút Lá: Mỗi nút lá trong Cây Merkle biểu thị một khối dữ liệu riêng lẻ, có thể là một giao dịch đơn lẻ, một tài liệu hoặc bất kỳ phần dữ liệu nào. Các nút này là tòa nhà cơ bản các khối của cây và được hình thành bằng cách băm dữ liệu mà chúng biểu thị bằng hàm băm mật mã, như SHA-256, đảm bảo tính toàn vẹn của dữ liệu được bảo mật bằng mật mã.
- Nút không có lá: Các nút không có lá hoặc nút trung gian được hình thành bằng cách băm quốc gia nối của các nút con của chúng. Quá trình này được áp dụng đệ quy, bắt đầu từ các nút lá và di chuyển lên trên, từ đó liên kết các nút bằng mật mã. Hiệu ứng chuỗi này đảm bảo rằng bất kỳ thay đổi nào trong một phần dữ liệu sẽ làm thay đổi hàm băm của nút cha, khiến những thay đổi trái phép có thể bị phát hiện.
- Nút gốc: Phần trên cùng của cây, là kết quả của quá trình băm từng bước này. Hàm băm được lưu trữ trong nút này, được gọi là Merkle Root, là một hàm băm duy nhất thể hiện hiệu quả toàn bộ tập dữ liệu bên dưới nó. Tính toàn vẹn của Merkle Root cực kỳ quan trọng vì nó phản ánh tính toàn vẹn của toàn bộ cây.

Cấu trúc dữ liệu của blockchain được thiết kế như sau:

```
Prev Hash:
Transaction in Block: [Genesis]
Hash: e7b62903d89a6f974da61122de52a3361b184f6e98d625ce8b55c7d9c7e92090

Prev Hash: e7b62903d89a6f974da61122de52a3361b184f6e98d625ce8b55c7d9c7e92090
Transaction in Block: [test1]
Hash: d73f234039af9f3c67fbe45196e2f4f31eab2e07ebf6f6c838d69dfe6feee6b2
```

- Các block trong chuỗi
- Giá trị hash của khối phía trước
- Tên Giao dịch
- Giá trị hash hiện tại

4. THỰC HÀNH

4.1. BLOCKCHAIN THE CHAIN OF BLOCKS: IMPLEMENT A CHAIN OF BLOCKS AS AN ORDERED, BACK-LINKED LIST DATA STRUCTURE.

Trong thực nghiệm này, chúng ta sẽ không triển khai khối như được mô tả trong các chuỗi khối được triển khai hiện tại hoặc thông số kỹ thuật của Bitcoin, thay vào đó, chúng tôi sẽ sử dụng phiên bản đơn giản hóa của khối. Cấu trúc sẽ gồm một số yếu tố sau:

- Timestamp là dấu thời gian hiện tại (khi khối được tạo),
- Transactions là thông tin có giá trị thực tế có trong khối,
- PrevBlockHash lưu trữ hàm băm của khối trước đó
- Hash là hàm băm của khối.

Hiện tại, giao dịch của chúng ta chỉ là một chuỗi byte hai chiều chứa dữ liệu sẽ được lưu trữ. Bạn có thể đọc thêm về cách thực hiện giao dịch tại đây. Mỗi khối được liên kết với khối trước đó bằng hàm băm. Trong thực nghiệm này, bạn sẽ chỉ lấy các trường khối (tức là các header), nối chúng và tính toán hàm băm SHA-256 trên tổ hợp được nối.

Để tính tổng kiểm tra SHA-256 của dữ liệu, bạn có thể sử dụng hàm Sum256 từ thư viện go crypto. Hàm nhận tham số đầu vào là một chuỗi byte, vì vậy ta sẽ cần chuyển đổi từng trường của tiêu đề thành byte rồi nối nó.

Blockchain sẽ được xây dựng bằng một số hàm sau:

- **CreateBlock**: tạo một block mới
- **Genesis**: khởi tạo khối gốc Genesis
- **AddBlock**: lấy khối băm trước đó và thêm khối mới liên kết nó với khối trước đó.
- **InitBlockChain**: khởi chạy 1 blockchain từ block gốc Genesis
- **GetBlock**: Hàm này sẽ trả về một khối dựa trên hàm băm của nó.

4.2. EFFICIENT TRANSACTIONS AND BLOCKS VERIFICATION: IMPLEMENT AN EFFICIENT WAY TO VERIFY MEMBERSHIP OF CERTAIN TRANSACTIONS IN A BLOCK USING MERKLE TREES

Trong thực nghiệm, để đơn giản hơn phần lý thuyết, ta sẽ thực hiện thử nghiệm các giao dịch trên môi trường nội bộ để hiểu và minh họa được cách thức hoạt động cơ bản của giao dịch. Một số hàm hỗ trợ cho giao dịch:

- **SetTransactions**: thiết lập giao dịch trong blockchain đã tạo
- **GetTransaction**: lấy thông tin và dữ liệu của các giao dịch khác
- **AddTransaction**: thêm giao dịch vào blockchain

Trong thực nghiệm, để đơn giản hơn phần lý thuyết, ta sẽ bỏ qua đường dẫn Merkle. Một số hàm hỗ trợ cho Merkle

Tree:

- **NewMerkleTree**: tạo cây Merkle mới
- **NewMerkleNode**: Tạo nút mới trên cây Merkle
- **Sum256**: tính tổng kiểm tra SHA-256 của dữ liệu từ thư viện crypto/sha256

4.3. COMMAND LINE CLIENT AND BENCHMARKS

Giao diện người dùng được thiết kế đơn giản, phù hợp với người dùng thông qua console hoặc terminal.
Khi chạy chương trình bằng lệnh **go run "/path/to/main.go",** chương trình sẽ hiển thị menu các chức năng chương trình như sau:

```
PS C:\Users\HP\Downloads\Golang> go run "C:\Users\HP\Downloads\Golang\main.go"
Blockchain Console:
-------------------

Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice:
```

Các thông báo và yêu cầu từ chương trình trong từng bước cũng được hiển thị tùy theo lựa chọn mà người dùng nhập vào.

5. CÁC CHỨC NĂNG

5.1. THÊM GIAO DỊCH MỚI

Khi nhập lựa chọn 1 ở menu chính, chương trình sẽ yêu cầu nhập tên giao dịch muốn tạo, sau đó nếu được sẽ hiển thị thông báo thành công.

```
Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 1
Enter transaction data: test1
Transaction added successfully.
```

5.2. THÊM KHỐI MỚI

Khi nhập lựa chọn 2 ở menu chính, chương trình sẽ mặc định tạo 1 khối gốc cho blockchain và một khối mới theo yêu cầu, sau đó nếu được sẽ hiển thị thông báo thành công

```
Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 2
New block added successfully.
```

5.3. HIỂN THỊ BLOCKCHAIN

Khi nhập lựa chọn 3 ở menu chính, chương trình hiển thị blockchain gồm block gốc () và các block hiện có. Sau đó sẽ hiển thị blockchain ra console.

```
Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 3

Prev Hash:
Transaction in Block: [Genesis]
Hash: 3124cf4be042c23fe384f0400a2f427602849d57c0ab76bfd503dc5b7aefadf0

Prev Hash: 3124cf4be042c23fe384f0400a2f427602849d57c0ab76bfd503dc5b7aefadf0
Transaction in Block: [test1,2]
Hash: 12bf8a371d9e75ff0f5238556a40e05b2c5424214d9610041c9587f93a3418c6
```

5.4. HIỂN THỊ GIAO DỊCH

Khi nhập lựa chọn 4 ở menu chính, chương trình hiển thị các giao dịch đã có trước đó. Sau đó sẽ hiển thị ra console.

```
Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 1
Enter transaction data: tx1
Transaction added successfully.

Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 4
Transactions:
[tx1]
```

```
Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 1
Enter transaction data: tx2
Transaction added successfully.

Select an operation:
1. Add Transaction
2. Add New Block
3. Display Blockchain
4. Display Transactions
5. Exit
Enter your choice: 4
Transactions:
[tx1,tx2]
```

6 NGUỒN THAM KHẢO

SST	Link tham khảo
1	https://viblo.asia/p/xay-dung-blockchain-tu-dau-voi-go-phan-1-GyZJZN9EJjm
2	https://en.wikipedia.org/wiki/Merkle_tree
3	https://en.bitcoin.it/wiki/Transaction
4	https://www.ardanlabs.com/blog/2022/03/blockchain-02-transaction-distribution-synchronization.html
5	https://dev.to/douglasmakey/my-introduction-to-the-blockchain-and-merkle-tree-6ge
6	https://xlinux.nist.gov/dads//HTML/fullBinaryTree.html
7	https://bitcoin.org/bitcoin.pdf

HẾT