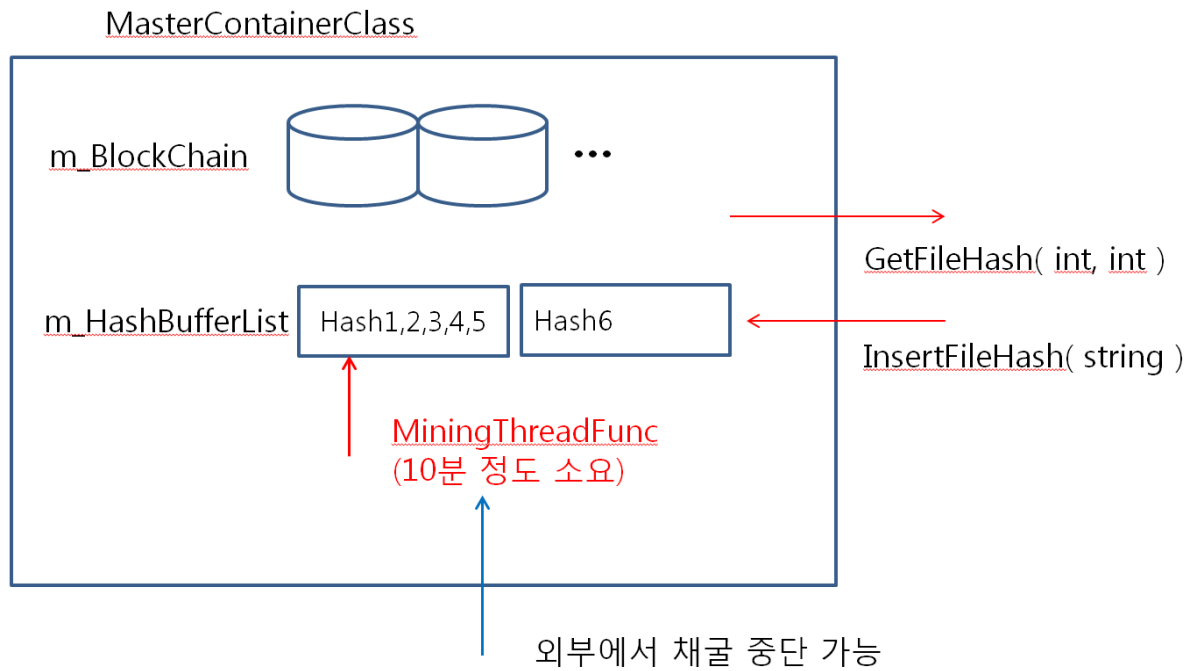


12주차 - 채굴기능 구현완료, 파일 해쉬 자료구조 완성

이번주 진행

- MasterContainerClass 완성



MasterContainerClass 사용법.

```

30
31 MasterContainerClass MasterContainer;
32
33 int main()
34 {
35     //제네시스 블록 출력
36     //cout << MasterContainer.GetFileHash(0, 0) << endl;
37
38     //채굴기 소유자 설정.
39     MasterContainer.SetOwnerUser("User1");
40
41     //채굴시키기 위해 파일 해쉬 삽입, 5개가 찼때마다 자동으로 채굴이 시작된다.
42     for (int i = 1; i < 7; i++)
43         MasterContainer.InsertFileHash(to_string(i));
44
45     //채굴 중지 (이로 인해 블럭하나를 채굴하다가 취소된다.)
46     //MasterContainer.m_MiningCancelFlag = true;
47
48     //채굴이 완료될때 까지 3초 대기
49     std::this_thread::sleep_for(3s);
50
51     cout << MasterContainer.GetFileHash(1, 0) << endl;
52     cout << MasterContainer.GetFileHash(1, 1) << endl;
53     cout << MasterContainer.GetFileHash(1, 2) << endl;
54     cout << MasterContainer.GetFileHash(1, 3) << endl;
55     cout << MasterContainer.GetFileHash(1, 4) << endl;
56     cout << MasterContainer.GetFileHash(2, 0) << endl;
57
58     //1000초 대기
59     std::this_thread::sleep_for(1000s);
60
61     //범위 에러가 나는 샘플
62     //cout << MasterContainer.GetFileHash(20, 4) << endl;
63
64     return 0;
65 }
--

```

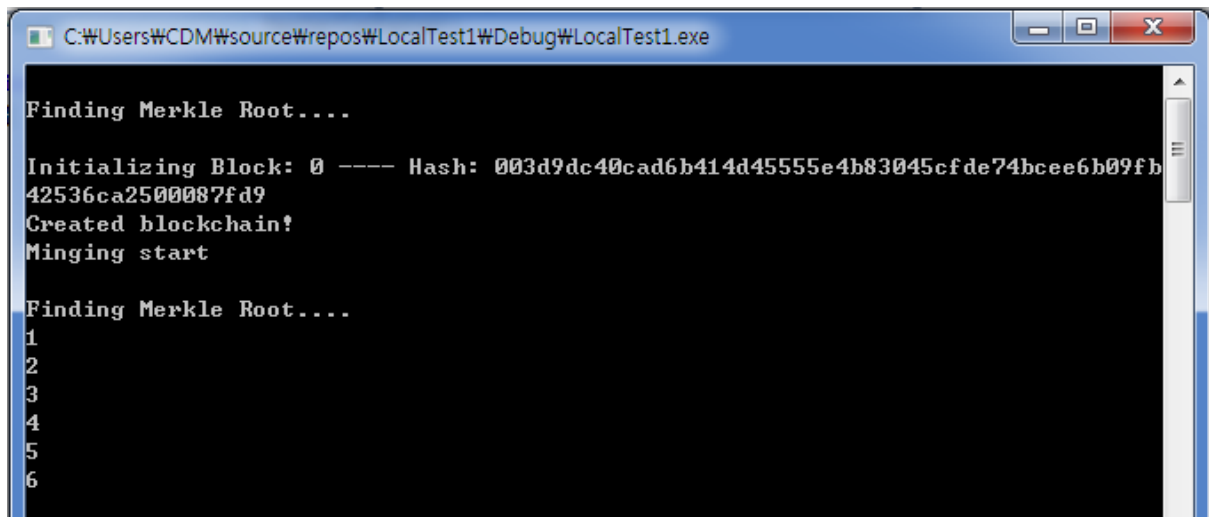
36 : (0,0) 은 "Genesis Block!" 이라는 문자열이 들어있는 블록체인의 블록 1번이다.

39 : SetOwnerUser는 채굴에 성공했을 때 보상을 받을 사용자를 지정한다.

43 : 1,2,3,4,5,6 총 6개의 샘플을 삽입한다. 이 중에서 5번 을 넣었을 때 내부적으로 채굴 스레드가 생성되고 채굴과정이 돌아간다. 채굴이 완료되기 까지는 10분정도의 시간이 소요된다.

46 : 외부에서 진행중인 채굴을 중지시키는 코드이다.

51 ~56 : 위에서 넣었던 1,2,3,4,5,6 을 꺼내온다.



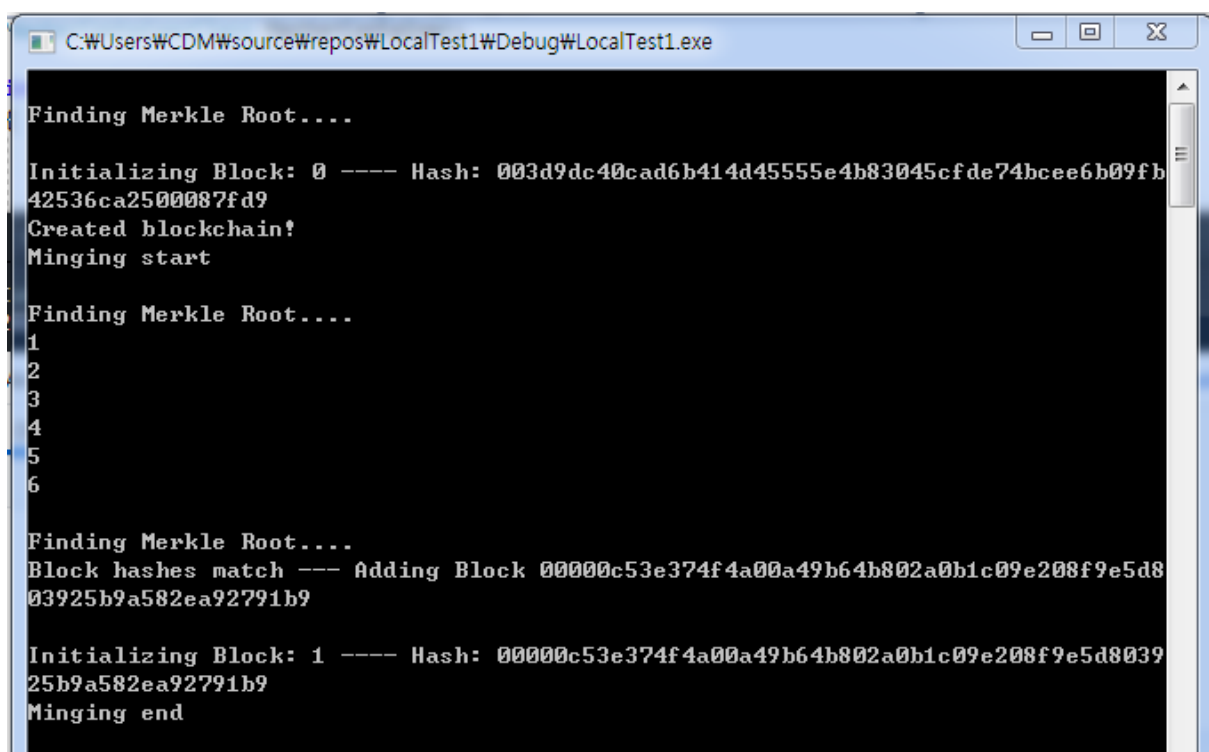
```
C:\Users\CDM\source\repos\LocalTest1\Debug\LocalTest1.exe

Finding Merkle Root....

Initializing Block: 0 ---- Hash: 003d9dc40cad6b414d45555e4b83045cfde74bcee6b09fb42536ca2500087fd9
Created blockchain!
Mining start

Finding Merkle Root....
1
2
3
4
5
6
```

채굴중인 모습



```
C:\Users\CDM\source\repos\LocalTest1\Debug\LocalTest1.exe

Finding Merkle Root....

Initializing Block: 0 ---- Hash: 003d9dc40cad6b414d45555e4b83045cfde74bcee6b09fb42536ca2500087fd9
Created blockchain!
Mining start

Finding Merkle Root....
1
2
3
4
5
6

Finding Merkle Root....
Block hashes match --- Adding Block 00000c53e374f4a00a49b64b802a0b1c09e208f9e5d803925b9a582ea92791b9

Initializing Block: 1 ---- Hash: 00000c53e374f4a00a49b64b802a0b1c09e208f9e5d803925b9a582ea92791b9
Mining end
```

채굴이 완료된 모습

다음주 계획 - 프로토콜 프로젝트에 포팅하고 테스트하기

깃허브 :

[https://github.com/CSID-DGU/2018-2-OSSP-](https://github.com/CSID-DGU/2018-2-OSSP-FileEncryptionBlockChain/tree/master/BlockChain%26Buffer/%5B2%5DMasterContainer%ED%94%84%EB%A1%9C%ED%86%A0%ED%83%80%EC%9E%85)

[FileEncryptionBlockChain/tree/master/BlockChain%26Buffer/%5B2%5DMasterContainer%ED%94%84%EB%A1%9C%ED%86%A0%ED%83%80%EC%9E%85](https://github.com/CSID-DGU/2018-2-OSSP-FileEncryptionBlockChain/tree/master/BlockChain%26Buffer/%5B2%5DMasterContainer%ED%94%84%EB%A1%9C%ED%86%A0%ED%83%80%EC%9E%85)