

Laborator 6 - DHT

Balog David Alexandru

Link GITHUB : <https://github.com/BalogDavid/DHT>

1. La Adăugarea unui Nod

Când un nod nou este adăugat în rețeaua Chord:

1. Se identifică succesorul său, adică nodul cu ID-ul imediat mai mare decât ID-ul nodului nou.
2. Succesorul și predecesorul nodului nou sunt actualizați pentru a-l integra în inel.
3. Predecesorul succesorului anterior este actualizat pentru a indica noul nod.

2. La Inserarea unei Chei

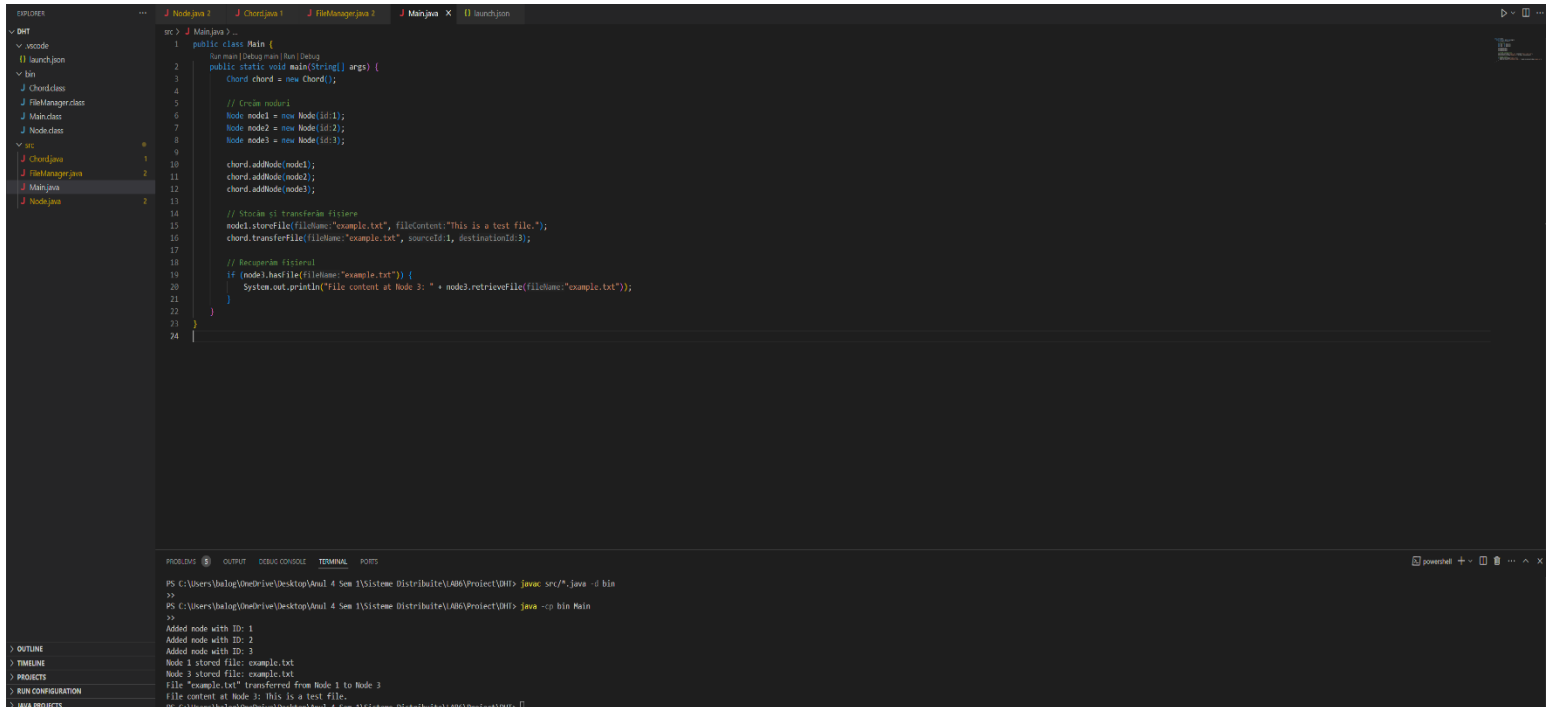
Când o cheie (de exemplu, hash-ul unui fișier) trebuie stocată:

1. Se caută nodul care are un ID mai mare sau egal cu hash-ul cheii.
2. Cheia este stocată pe acel nod.
 - Dacă cheia este între ID-ul unui nod și succesorul său, acel nod este responsabil de cheie.
 - Dacă nu, căutarea continuă către succesor.

3. Cum Funcționează Căutarea Succesorului?

- **Adăugare Nod:** Verificăm ID-urile nodurilor existente în ordine circulară până găsim succesorul.
- **Inserare Cheie:** Verificăm dacă cheia aparține intervalului gestionat de un nod. Dacă nu, căutarea este trimisă către următorul nod din rețea.

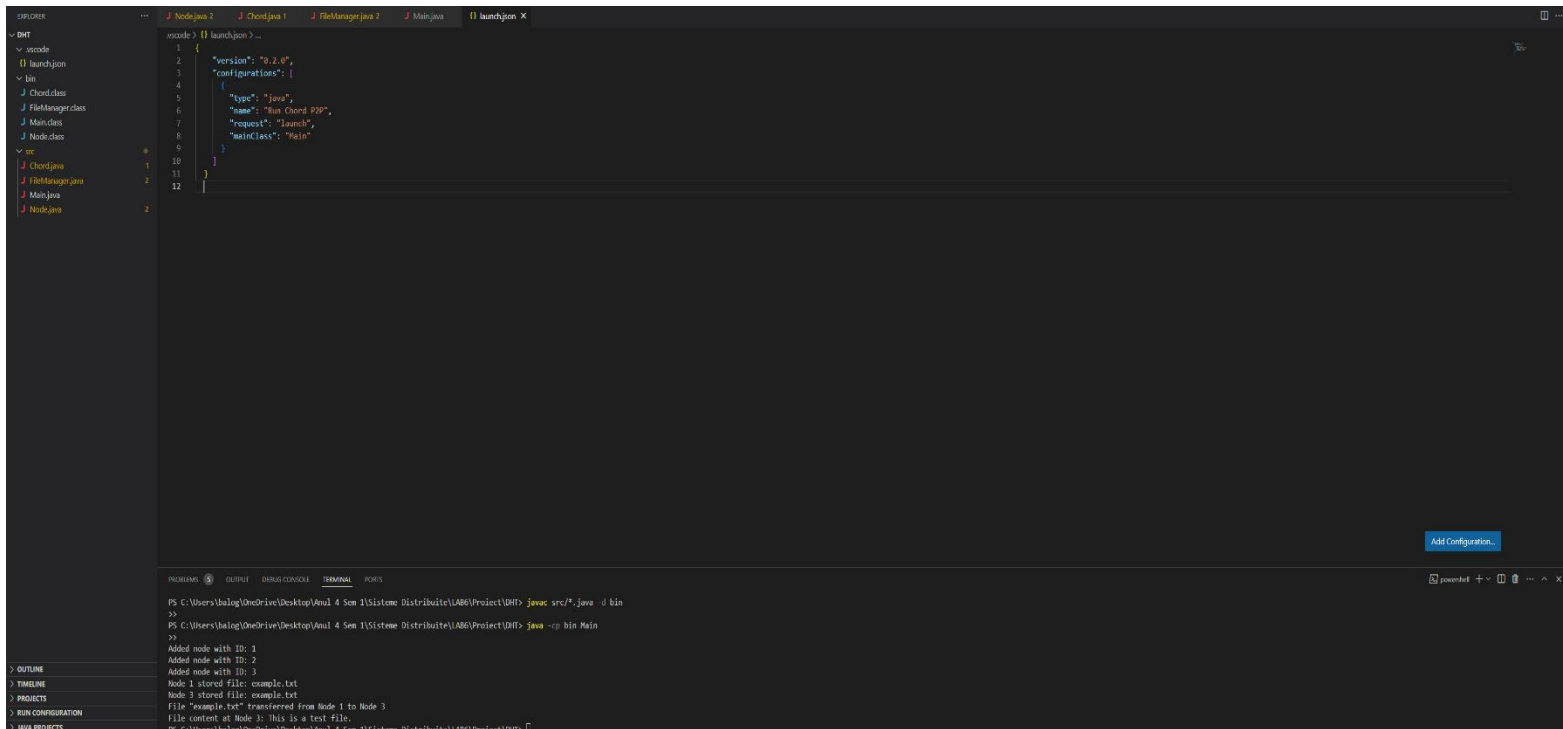
Screenshot :



```
src > J Main.java
1 public class Main {
2     Run main (Debug main) Run (Debug)
3     public static void main(String[] args) {
4         Chord chord = new Chord();
5
6         // Create nodes
7         Node node1 = new Node(id:1);
8         Node node2 = new Node(id:2);
9         Node node3 = new Node(id:3);
10
11         chord.addNode(node1);
12         chord.addNode(node2);
13         chord.addNode(node3);
14
15         // Store a file in node 1
16         node1.storeFile(filename:"example.txt", fileContent:"This is a test file.");
17         chord.transferFile(filename:"example.txt", sourceId:1, destinationId:3);
18
19         // Retrieve file from node 3
20         if (node3.hasFile(filename:"example.txt")) {
21             System.out.println("File content at Node 3: " + node3.retrieveFile(filename:"example.txt"));
22         }
23     }
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1> javac src/*.java -d bin
>>
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1> java -cp bin Main
>>
Added node with ID: 1
Added node with ID: 2
Added node with ID: 3
Node 1 stored file: example.txt
Node 3 stored file: example.txt
File "example.txt" transferred from Node 1 to Node 3
File content at Node 3: This is a test file.
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1>
```



```
src > J launch.json
1 {
2     "version": "0.2.0",
3     "configurations": [
4         {
5             "type": "java",
6             "name": "Run Chord P2P",
7             "request": "launch",
8             "mainClass": "Main"
9         }
10     ]
11 }
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1> javac src/*.java -d bin
>>
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1> java -cp bin Main
>>
Added node with ID: 1
Added node with ID: 2
Added node with ID: 3
Node 1 stored file: example.txt
Node 3 stored file: example.txt
File "example.txt" transferred from Node 1 to Node 3
File content at Node 3: This is a test file.
PS C:\Users\hahag\OneDrive\Desktop\Amul 4 Sem 1\System Distribute\LAB0\Project\DH1>
```