

Distributed File System

In acest laborator am completat TODO-urile (metoda `handleRead`, tratarea replicilor căzute, implementarea catalogului de fișiere). Am testat functionarea sistemului in felul urmator:

1. Am rulat `MasterServer` si 3 `ReplicaServer`.

```
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator5\DFS_Rmi\src> java MasterServer
RMI registry started on port 1099.
MasterServer is running...
Master: registered replica 1 at localhost
Master: registered replica 2 at localhost
Master: registered replica 3 at localhost
```

2. Am rulat un `Client` si am ales optiunea pentru scrierea unui mesaj.

```
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator5\DFS_Rmi\src> java Client
Conectat la MasterServer.
=== MENIU (Client5) ===
1. Write
2. Read
3. List
4. Quit
Alege optiunea: 1

Introdu mesajul de salvat: salut
Hash-ul asociat mesajului este: 1t0
Write reusit prin Master pentru hash 1t0
```

3. Mesajul a fost salvat cu success pe `Replica 1` si `Replica 2`.

```
Master: WRITE hash=1t0 -> message="salut"
Master: selected targets = [ReplicaLoc{id='1', host='localhost'}, ReplicaLoc{id='2', host='localhost'}]
Master: WRITE reusit pe replica 1
Master: WRITE reusit pe replica 2
Master: WRITE finalizat cu succes pe 2 replica(e)
```

- `Replica 1`

```
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator5\DFS_Rmi\src> java ReplicaServer 1
ReplicaServer 1 running...
Replica 1: stored 1t0 -> salut
```

- Replica 2

```
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator5\DFS_Rmi\src> java ReplicaServer 2
ReplicaServer 2 running...
Replica 2: stored 1t0 -> salut
```

4. Folosind Client-ul am listat continutul stocat pe replici (mesajul apare pe ambele replici).

```
=== MENU (Client5) ===
1. Write
2. Read
3. List
4. Quit
Alege optiunea: 3

Continutul replicilor (hash -> mesaj):
- Replica id=1, host=localhost
  1t0 -> salut
- Replica id=2, host=localhost
  1t0 -> salut
- Replica id=3, host=localhost
  (nu exista mesaje stocate)
```

5. Am citit mesajul folosind hash-ul care a fost asociat la scriere.

```
=== MENU (Client5) ===
1. Write
2. Read
3. List
4. Quit
Alege optiunea: 2

Introdu hash-ul de citit: 1t0
Cautam mesajul asociat hash-ului: 1t0
Mesajul gasit: salut
```

- MasterServer a citit catalogul de fisiere, a identificat replicile pe care se afla mesajul si a ales una dintre ele (Replica 1) pentru a citi mesajul.

```
Master: READ hash=1t0
Master: folosesc catalogul de fisiere pentru hash=1t0
Master: replici candidate din catalog: [ReplicaLoc{id='1', host='localhost'}, ReplicaLoc{id='2', host='localhost'}]
Master: citit cu succes de pe replica 1
```

6. Pentru a simula toleranta la defecte am oprit Replica 1 (cea aleasa initial de MasterServer).

7. Am folosit din nou Client-ul pentru a citi mesajul (a fost citit cu success).

```
=== MENU (Client5) ===
1. Write
2. Read
3. List
4. Quit
Alege optiunea: 2

Introdu hash-ul de citit: 1t0
Cautam mesajul asociat hash-ului: 1t0
Mesajul gasit: salut
```

8. MasterServer a identificat faptul ca Replica1 este inactive si a ales urmatoarea replica (Replica 2) pentru a citi mesajul.

```
Master: READ hash=1t0
Master: folosesc catalogul de fisiere pentru hash=1t0
Master: replici candidate din catalog: [ReplicaLoc{id='1', host='localhost'}, ReplicaLoc{id='2', host='localhost'}]
-> Replica este indisponibila
-> Incerc urmatoarea replica...
Master: citit cu succes de pe replica 2
```

9. In final am listat inca o data continutul de pe replici (Replica 1 a fost identificata ca indisponibila).

```
=== MENU (Client5) ===
1. Write
2. Read
3. List
4. Quit
Alege optiunea: 3

Continutul replicilor (hash -> mesaj):
- Replica id=1, host=localhost
  Replica 1 indisponibila:
- Replica id=2, host=localhost
  1t0 -> salut
- Replica id=3, host=localhost
  (nu exista mesaje stocate)
```

Deficiențele sistemului și propuneri de îmbunătățire

Am identificat cateva deficiente pe care acest sistem le are.

1. **Datele sunt stocate in memorie** pe replici, deci daca MasterServer sau toate replicile pe care este stocat un anumit mesaj cad, atunci informatia se pierde si nu mai poate fi recuperata.
 - **Solutie:** Salvarea pe disc(fisiere sau baza de date) a catalogului de fisiere in MasterServer si a mesajelor din replici si reincarcarea lor automata la repornire.
2. **MasterServer reprezinta un Single Point of Failure**, daca el cade atunci sistemul nu mai functioneaza (chiar daca replicile functioneaza).
 - **Solutie:** Replicarea Master-ului si folosirea algoritmului Bully pentru a alege un lider. Doar liderul poate sa scrie, iar ceilalti vor avea o copie a catalogului.

Un alt lucru care ar imbunatati sistemul este implementarea unui Heartbeat intre MasterServer si replici pentru a identifica imediat cand replicile cad.