

Il progetto consiste in un file system distribuito di tipo client-server che utilizza la tecnologia Java RMI.

1. SPECIFICA DEI REQUISITI SOFTWARE

1.1 OBIETTIVI

L'obiettivo è fornire una serie di funzionalità per gestire un file system distribuito attraverso una interfaccia utente.

Gli utenti (client) posso richiedere l'esecuzione di operazioni su file oppure su folder. Le prime vengono eseguite da un File Service e le seconde da un Folder Service.

Le funzionalità su file che sono fornite sono:

1.1.1 Creazione di un nuovo file. L'utente deve indicare l'intero path del file che vuole creare, indicando anche l'estensione (ad esempio /documenti/contratto.txt) e il contenuto del file. Se nella cartella esiste già un file con lo stesso nome, verrà restituito un errore.

1.1.2 Eliminazione di un file. L'utente deve indicare l'intero path del file che vuole eliminare, indicando anche l'estensione.
Se il file che si vuole eliminare non esiste, viene restituito un errore.

1.1.3 Lettura di un file. L'utente deve indicare l'intero path del file che vuole leggere, indicando anche l'estensione.
Se il file non esiste, viene restituito un errore.

1.1.4 Append su un file. L'utente deve indicare l'intero path del file che vuole modificare, indicando anche l'estensione, e il testo che vuole aggiungere all'interno del file.
Il file indicato dall'utente viene modificato aggiungendo in fondo il contenuto indicato.
Se il file non esiste, viene restituito un errore.

1.1.5 Spostare un file. L'utente deve indicare l'intero path del file che vuole spostare e il nuovo path, indicando anche l'estensione.
Se il file che si vuole spostare non esiste, viene restituito un errore.
Se nella nuova cartella il file che si vuole spostare esiste già, viene restituito un errore.

1.1.6 Rinomina del file. L'utente deve indicare l'intero path del file che vuole rinominare, e il nuovo nome del file (ad esempio nuovo_nome.txt).
Se il file che si vuole rinominare non esiste, viene restituito un errore.
Se il nuovo nome del file esiste già in quella cartella, viene restituito un errore.

Le operazioni sui folder sono:

1.1.7 Elenco di cartelle e file contenuti in una cartella. L'utente deve indicare il path del folder che vuole ispezionare mettendo un '/' alla fine (ad esempio /documenti/). Se il folder non esiste, viene restituito un errore.

1.1.8 Eliminazione di un folder. L'utente deve indicare il path del folder che vuole eliminare mettendo un '/' alla fine (ad esempio /documenti/).

Questi servizi sono forniti tramite un'interfaccia a linea di comando.

Quando viene fatto partire il Client, viene mostrato l'elenco delle operazioni disponibili e i comandi da effettuare per eseguirle

operazioni disponibili tramite cli:

read <path> per leggere un file

create <path> <text> per creare un nuovo file

delete <path> per eliminare un file

move <oldPath> <newPath> per spostare un file

append <path> <text> per aggiungere del testo al file

rename <path> <name> per rinominare un file

children <path> per visualizzare il contenuto di una cartella

delete-folder <path> per eliminare una cartella

exit per uscire

1.2 REQUISITI FUNZIONALI

1.2.1 Creazione di un nuovo file.

Introduzione

Consente all'utente di creare un nuovo file in una cartella.

Input

L'intero path del file che si vuole creare, indicando anche l'estensione, e il contenuto del file. Ad esempio /documenti/contratto.txt testo.

Elaborazione

Prima si verifica se il file nella cartella indicata esiste già. Nel caso verrà restituito un errore.

Viene aggiunta la entry del nuovo file alla tabella dei metadati (un semplice file .txt usato come registro dove vengono annotate delle informazioni per la gestione dei file).

Il nuovo file viene creato e replicato su 3 database diversi.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

1.2.2 Eliminazione di un file.

Introduzione

Consente all'utente di eliminare un file.

Input

L'intero path del file che si vuole eliminare, indicando anche l'estensione.

Elaborazione

Prima si fanno i controlli:

- Si verifica se il file che si vuole eliminare esiste. Se non esiste, verrà restituito un errore.
- Si controlla che su quel file non ci sia una modifica o lettura in corso. In tal caso si chiede di ritentare in seguito.

Si aggiorna la tabella dei metadati eliminando la entry del file.

Si elimina il file da tutti i database.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

1.2.3 Lettura di un file.

Introduzione

Consente all'utente di leggere il contenuto di un file.

Input

L'intero path del file che si vuole leggere, indicando anche l'estensione.

Elaborazione

Prima si fanno i controlli:

- Si verifica se il file che si vuole leggere esiste. Se non esiste, verrà restituito un errore.
- Si controlla che su quel file non ci sia una modifica in corso. In tal caso si restituisce un messaggio di errore.

Si aggiorna la entry del file nella tabella dei metadati per indicare che c'è un lettore.

Si va a prendere in uno dei database il contenuto del file.

Si aggiorna la entry del file nella tabella dei metadati per decrementare il numero di lettori.

Output

Il contenuto del file che si vuole leggere oppure un messaggio di errore.

1.2.4 Append su un file.

Introduzione

Consente all'utente di modificare un file aggiungendo del contenuto.

Input

File che si vuole modificare e contenuto che si vuole aggiungere.

Elaborazione

Si va a leggere il contenuto del file. Si utilizza l'operazione di lettura precedentemente presentata.

Si elimina il file. Si utilizza l'operazione di eliminazione precedentemente presentata.

Si crea un nuovo file con lo stesso nome e con il vecchio contenuto a cui viene aggiunto il nuovo. Si utilizza l'operazione di creazione precedentemente presentata.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

1.2.5 Spostare un file.

Introduzione

Consente all'utente di spostare un file da una cartella all'altra.

Input

L'intero path del file che vuole spostare e il nuovo path, indicando anche l'estensione.

Elaborazione

Prima si fanno i controlli:

- Si verifica se il file che si vuole spostare esiste. Se non esiste, verrà restituito un messaggio di errore.
- Si controlla che su quel file non ci sia una modifica o una lettura in corso. In tal caso si chiede di riprovare in seguito.
- Si controlla che nella cartella di destinazione non esiste già un file con lo stesso nome di quello che vogliamo spostare.

Nella tabella dei metadati si aggiunge la entry con il nuovo path del file e si elimina le entry con il vecchio path.

Nei database si cambiano i nomi dei file.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

1.2.6 Rinomina del file.

Introduzione

Consente all'utente di modificare il nome di un file.

Input

L'intero path del file che vuole leggere, indicando anche l'estensione, e il nuovo nome del file (ad esempio nuovo_nome.txt).

Elaborazione

Al nuovo nome del file si antepone il percorso del file di partenza e poi si utilizza l'operazione di spostamento del file presentata precedentemente.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

1.2.7 Elenco di cartelle e file contenuti in una cartella.

Introduzione

Consente all'utente di visualizzare l'elenco dei file e cartelle contenute in una cartella.

Input

Il path del folder che vuole ispezionare mettendo un '/' alla fine.
Ad esempio /documenti/.

Elaborazione

Consultando la metadata table, si cercano i file o cartelle che sono in quel folder.

Se la ricerca non restituisce nulla, viene restituito un messaggio di errore.

Output

Messaggio che riporta il contenuto della cartella oppure un messaggio di errore.

1.2.8 Eliminazione di un folder.

Introduzione

Consente all'utente di eliminare tutti i file e cartelle contenute in una cartella.

Input

Il path del folder che vuole eliminare mettendo un '/' alla fine.
Ad esempio /documenti/.

Elaborazione

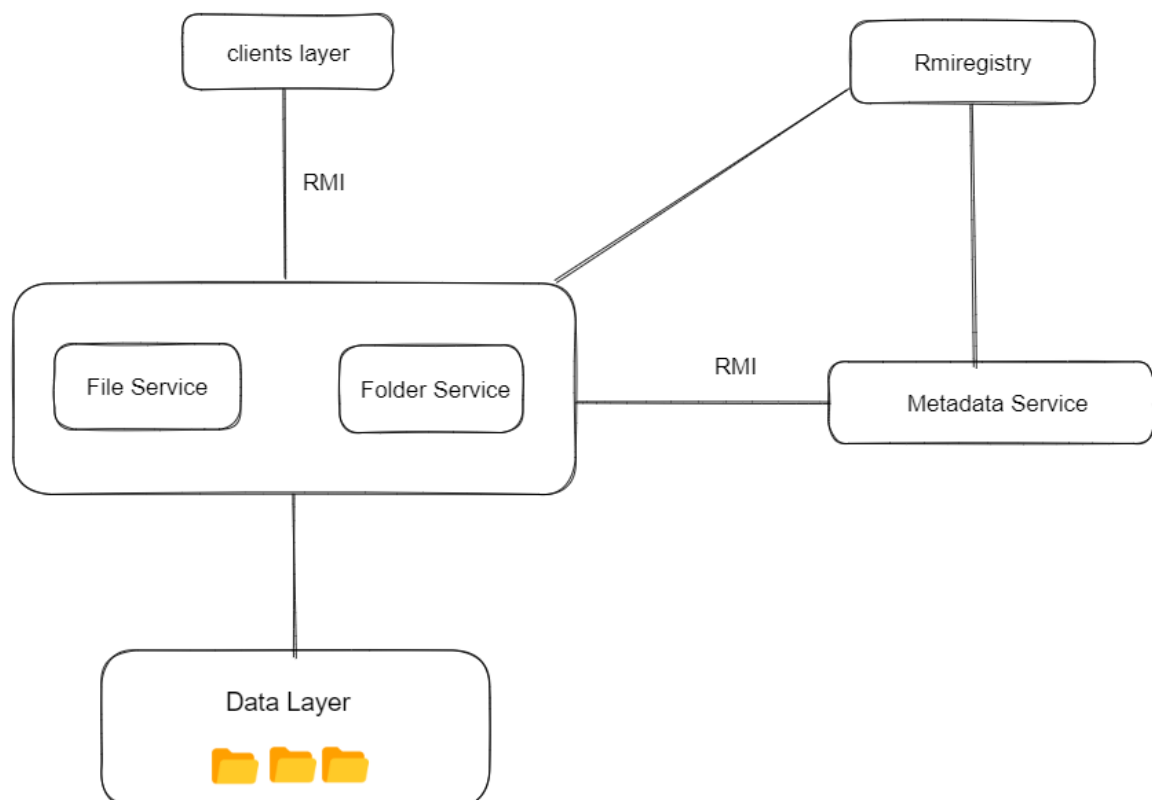
Consultando la metadata table, si cercano i file che sono in quel folder e vengono eliminati sia dalla metadata table sia dai database.

Se la ricerca non restituisce nulla, viene restituito un messaggio di errore.

Output

Messaggio che riporta l'operazione avvenuta oppure un messaggio di errore.

2. ARCHITETTURA



L'architettura è così composta:

- Più client che usando RMI chiamano i metodi remoti forniti da File Service e Folder Service.
- File Service che fornisce metodi per compiere operazioni sui file. Utilizzando RMI chiama i metodi remoti di Metadata Service.
- Folder Service fornisce metodi per effettuare operazioni sulle cartelle. Utilizzando RMI chiama i metodi remoti di Metadata Service.
- Metadata Service che fornisce dei servizi usati da File Service e Folder Service. Il Metadata Service gestisce i lock e tiene memorizzato l'elenco di file presenti nelle cartelle DB. I service chiederanno a lui se possono fare certe operazioni. Per ogni file il Metadata Service tiene memorizzata una entry della sua map. L'oggetto map è presente nella classe Metadata Service e a ogni modifica l'oggetto map viene serializzato e scritto sul file MetadataTable.txt.
- Rmiregistry: name server che mantiene l'associazione tra nomi e oggetti. Gli oggetti remoti (File Service, Folder Service, Metadata Service) si registrano a lui. I client con l'operazione di lookup potranno avere il riferimento di questi oggetti e poi chiamare i metodi remoti. I Client hanno i riferimenti a oggetti delle classi Folder Service e File Service. I service invece hanno il riferimento all'oggetto di classe Metadata Service.
- Data layer che è composto da cartelle del file system dove sono memorizzati i file. Ogni cartella del file system rappresenta un diverso DB. Avremo quindi 3 cartelle che contengono gli stessi file replicati. Nel nome del file è contenuto il suo path assoluto.

3. UML CLASS DIAGRAM

