# Funzionamento senza failure

Il flusso di eventi che caratterizzano il sistema secondo un funzionamento senza errori necessita di particolare attenzione verso le diverse fasi di scrittura dei file: un file modificato su un server replica deve essere inoltrato agli altri file-server della rete, oppure, uno stesso file modificato da due client distinti presso due o più file-server distinti deve, al termine delle operazioni, risultare identico in tutte le sue copie esistenti presso tali file-server. Verranno qui di seguito illustrati passo-passo entrambi i due casi.

Si considera il sistema composto da un DNS, tre server replica, due client connessi a due server replica differenti.

Nello specifico, il primo client, che verrà chiamato Client1, è connesso al server replica con ID 3, mentre il secondo client, analogamente Client2, è connesso al server replica con ID 2.

Come precedentemente specificato, la priorità nelle commit di scrittura, ordinate dai client ai server, segue uno schema prioritario basato sull’ID dei server: minore sarà l’ID, maggiore sarà la priorità.

# 1) Scrittura di un file presso un server e propagazione del suo aggiornamento

Come primo passo viene avviato il server DNS.

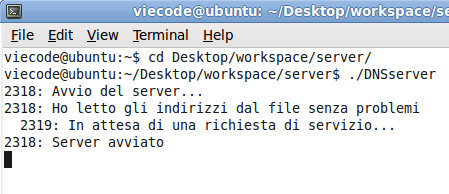


Figure : Avvio del server DNS

Successivamente vengono avviati i file-server replica.

Il primo server avviato, server replica ID 1, riceverà dal DNS la lista degli altri server presenti nella rete e proverà a contattarli. Nel caso specifico i server con ID 2 e ID 3 non sono ancora avviati. Fatto ciò il sistema si pone in attesa di richieste di agrawala per le operazioni sui file.

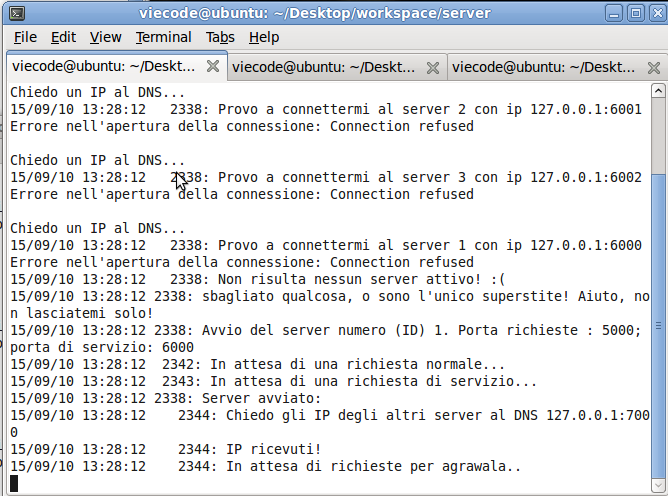


Figure : Avvio del server 1

Viene avviato il secondo server, server replica ID 2, che come il primo, contatta e riceve dal DNS la lista dei server registrati. Oltre a questo, trovando il server ID 1 attivo, richiederà la versione più aggiornata di un file testuale presente in entrambi, “marina.txt”. La transazione avrà un id univoco.

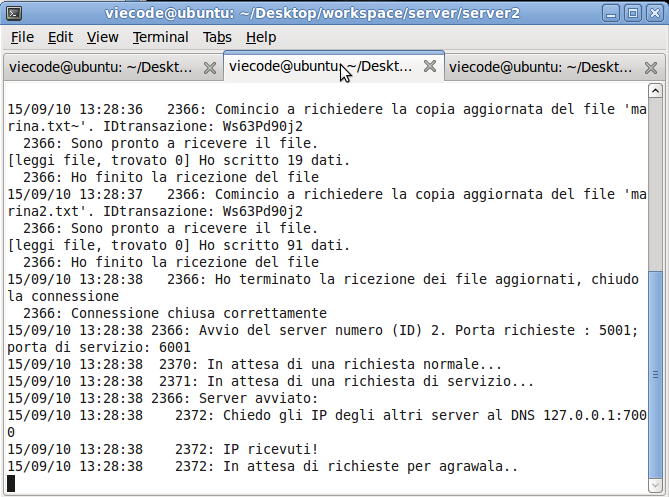


Figure : Avvio del server 2

L’ultimo ad essere avviato è il server ID 3, che troverà sia ID 1 che ID 2 operanti.

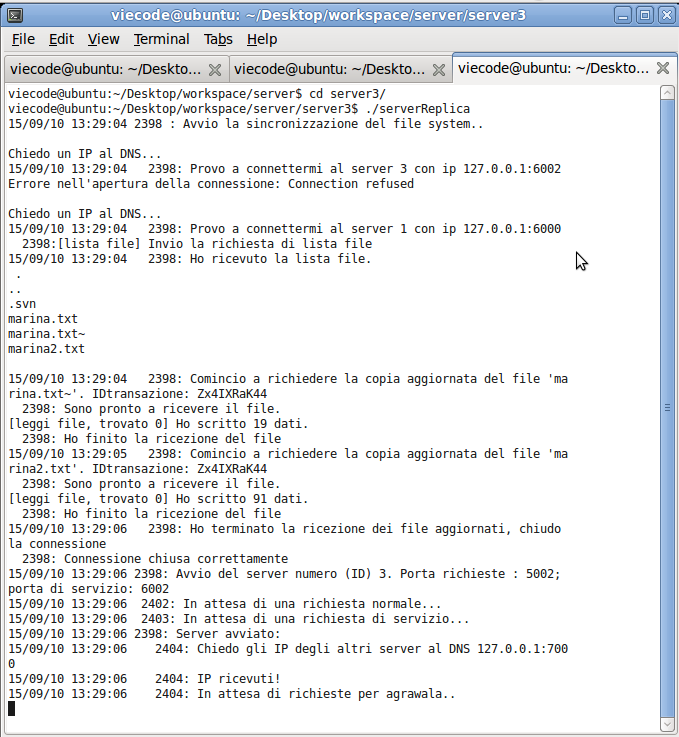


Figure : Avvio del server 3

Vengono lanciati i due client:

* Il client 1, che stabilisce una connessione con il server ID 3, assegnatogli dal DNS.

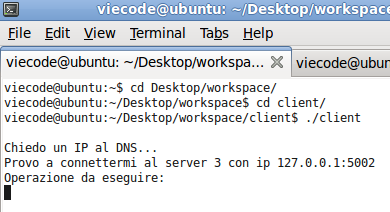


Figure : Avvio del primo client

* Il client2, che stabilisce una connessione con il server replica ID 1, assegnatoli dal DNS.

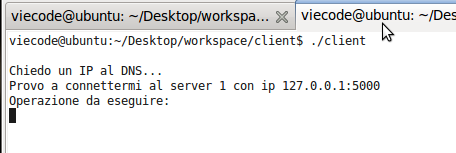


Figure : Avvio del client 2

Il client 1 richiede al server ID 3 di poter effettuare una scrittura su un file inizialmente vuoto, “marina.txt”, digitando il comando “scrivi file”. Ricevuto un messaggio di conferma dal server replica, immetterà nel terminale la stringa “stringa 1”, darà invio ed al messaggio successivo del server che ha preso in consegna la stringa, risponderà con “commit” per chiedere a questo di salvare le modifiche.

Successivamente, con il comando “uscita”, terminerà la connessione col server ID 3.

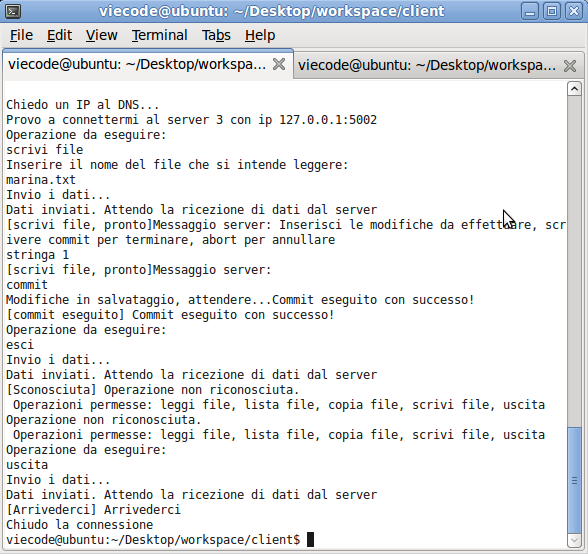


Figure : Scrittura di un file e uscita con il client 1

Nel server ID 3 è ora presente la versione aggiornata del file.

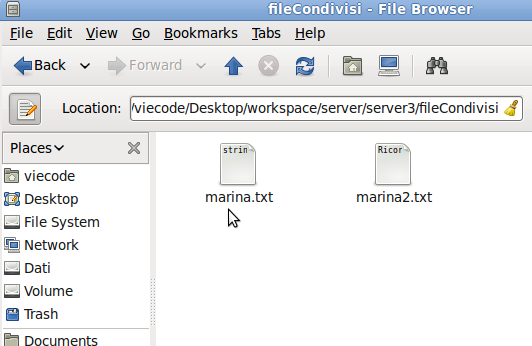


Figure : File marina.txt presente sul file system

Il file “marina.txt” è stato tenuto aperto, per comodità, da un editor testuale che infatti mostra che il file ha subito delle modifiche da parte di un’applicazione (i server replica in esecuzione sulla macchina) ed occorre farne il reload per poterne visualizzare il contenuto aggiornato.

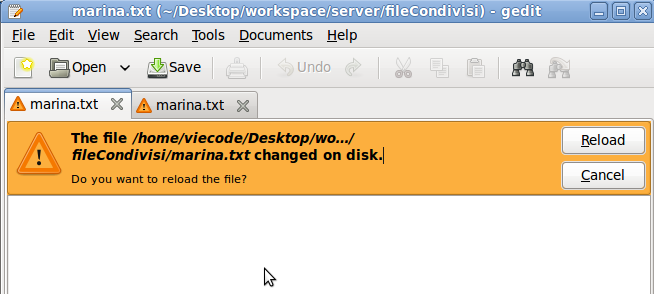


Figure : Gedit notifica la scrittura del file da parte di un altro processo

Le modifiche risuntano infatti esser state propagate dal server ID 3 agli altri due server replica:

* Il server ID 1

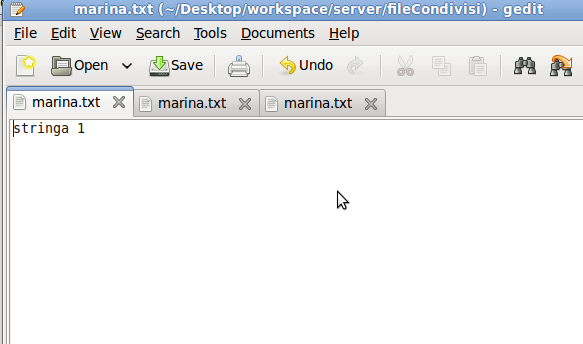


Figure : Contenuto del file presente nel file system del server 1

* Il server ID 2

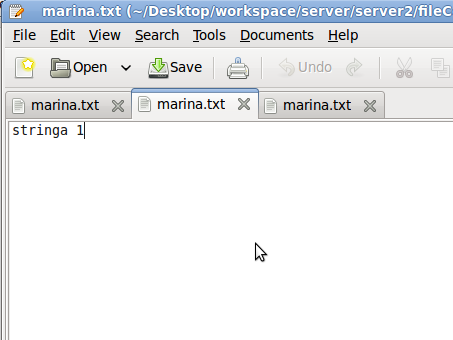


Figure : Contenuto del file presente nel file system numero 2

Dal punto di vista del server ID 3, sono visibili in shell le operazioni di ricezione della stringa testuale, ricezione della richiesta di salvataggio, avvio dell’algoritmo di agrawala per stabilire delle priorità (in questo caso nessun altro server era interessato ad effettuare il commit), ricezione delle reply dagli altri server e propagazione del file.

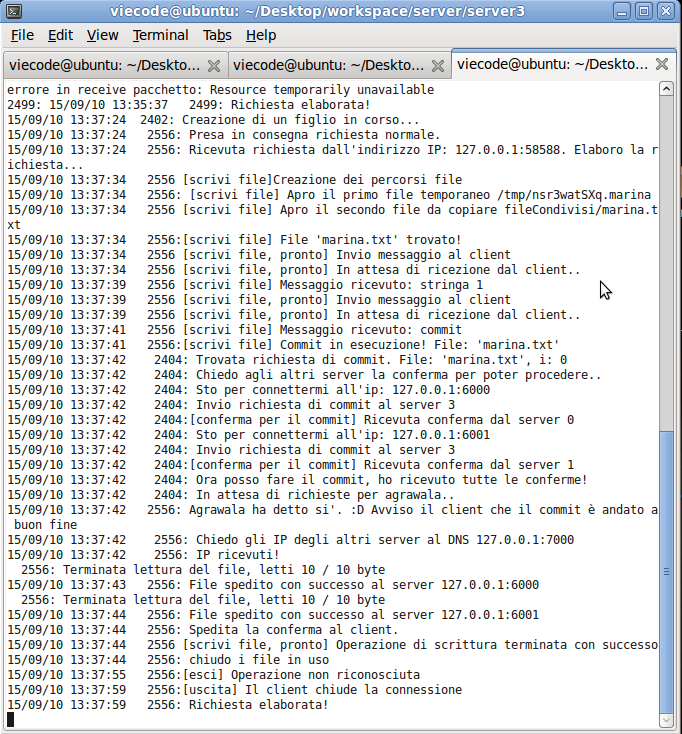


Figure : Operazioni svolte dal server 3 durante la scrittura del file

Dal punto di vista degli altri due server ID 1 e ID 2 è possibile notare la ricezione della richiesta, l’invio della reply ad ID 3 e l’aggiornamento del file ricevuto.

* Output del server ID 1

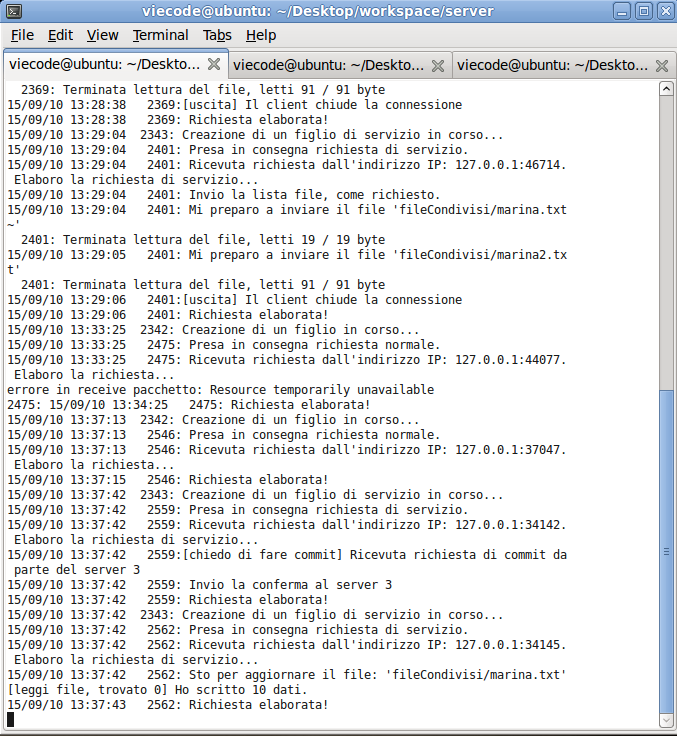


Figure : Operazioni svolte dal server 1 durante la scrittura

* Output del server ID 2, del tutto analogo al server precedente

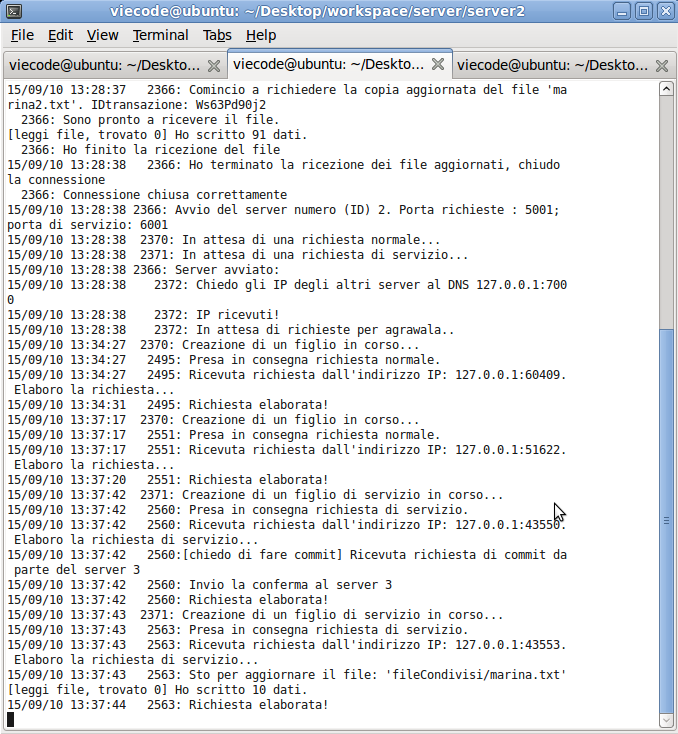


Figure : Operazioni svolte dal server 2 durante la scrittura

# 2) Scritture concorrenti

Verrà ora illustrata la gestione delle priorità tra server replica.

Due client, connessi a due server distinti, effettueranno delle scritture concorrenti su due copie dello stesso file.

Il file in questione è quello usato nel paragrafo precedente, “marina.txt”.

I server sono tutti avviati come nel caso precedente.

Il client 1 viene ravviato, e ottiene una connessione con il server ID 3.

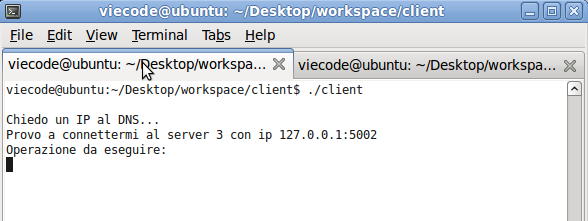


Figure : Avvio del client 1

Il client 2 viene anch’esso avviato e ottiene una connessione con il server replica ID 1, il quale avrà maggiore priorità nelle scritture rispetto al server ID 3, essendo il suo ID minore.

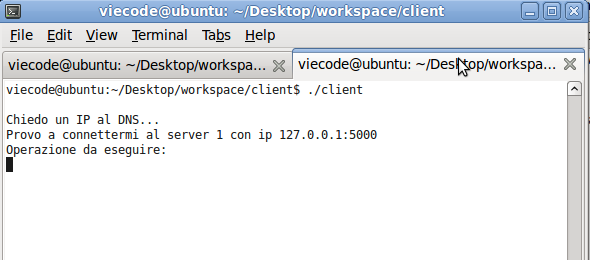


Figure : Avvio del client 2

Tornando al client 1, questo effettuerà presso il server ID 3 le operazioni di “scrivi file”, specifica del nome del file “marina.txt”, immissione di una stringa, “stringa 23”, ma senza effettuare la richiesta di commit.

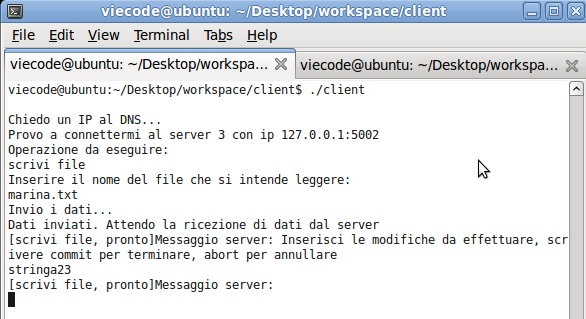


Figure : Richiesta scrittura file da parte del client 1

Il client 2, eseguirà contemporaneamente al client 1, le seguenti operazioni: “scrivi file”, specifica del nome del file “marina.txt”, immissione di una stringa “stringa 45”, immissione della richiesta (che verrà accettata) di “commit”.

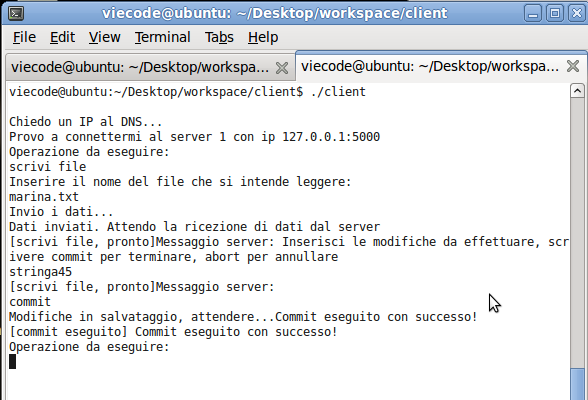


Figure : Scrittura del file e commit da parte del client 2

La richiesta di commit è stata accettata in quanto il server del client 2, server ID 1, ha l’ID minore rispetto agli altri server interessati (ID 3).

A questo punto il client 1 effettuerà la richiesta di commit (accettata perché quella del client 2 è già stata eseguita) e si disconnette.

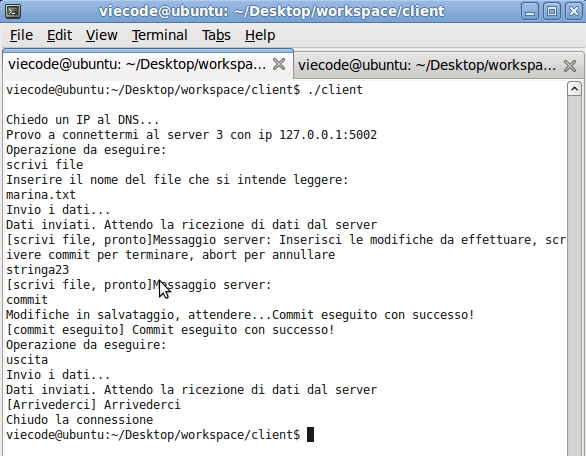


Figure : Commit da parte del client 1

Dal punto di vista dei server replica, è possibile vedere la richiesta presa in consegna dal server ID 1, e l’avvio di agrawala

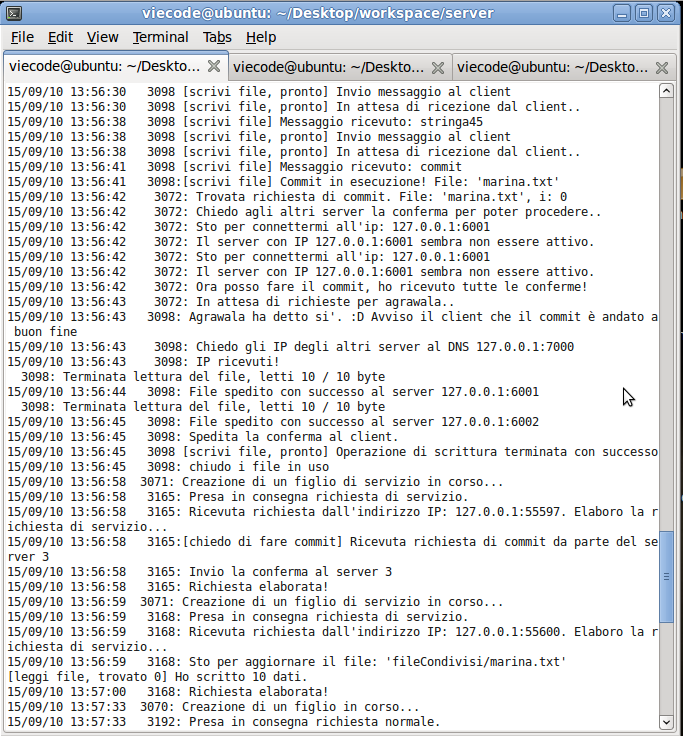


Figure : Il server 1 scrive il file effettuado anche Agrawala

L’output generato dal server ID 3 sarà il seguente:

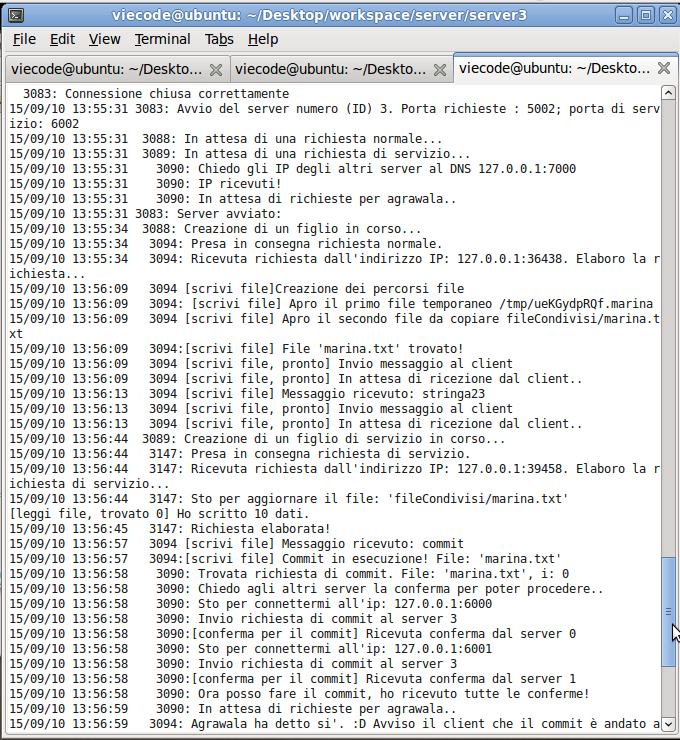


Figure : Scrittura del file da parte del server 3

L’editor di testo mostra le modifiche correttamente inserite, come nell’ordine delle commit, quindi prima la stringa “stringa 45” e dopo la stringa “stringa 23”.

* Nel server ID 1

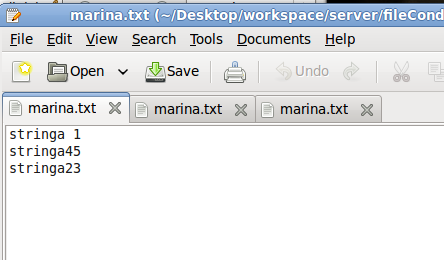


Figure : Server 1, Contenuto del file marina.txt

* Nel server ID 2

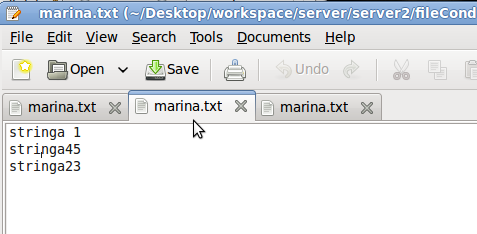


Figure : Server 2, Contenuto del file marina.txt

* Nel server ID 3

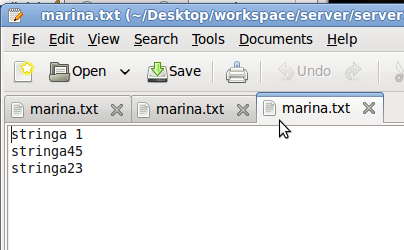


Figure : Server 3, Contenuto del file marina.txt