



HashFS

Un filesystem basato sull'hashing.

Damian Tosoni, Danilo Salvati

Sicurezza dei sistemi informatici e delle reti

Indice

Lo strumento fondamentale: FUSE.....	pag.2
Che cos'è.....	pag.2
Come funziona.....	pag.2
Esempi di filesystems realizzati con FUSE.....	pag.3
Le API.....	pag.1
 Valutazione delle prestazioni dei filesystems.....	pag.X
Che cos'è.....	pag.1
Che cos'è.....	pag.1
 I principali problemi da risolvere.....	pag.Y
 L'architettura.....	pag.Y
Che cos'è.....	pag.1
Che cos'è.....	pag.1
 Implementazione del filesystem.....	pag.Z
Che cos'è.....	pag.1
Che cos'è.....	pag.1
Che cos'è.....	pag.1
Che cos'è.....	pag.1

Legenda:

NERO Lavoro svolto in comune

BLU Lavoro svolto da Damian Tosoni

VERDE Lavoro svolto da Danilo Salvati

Capitolo 1

Lo strumento fondamentale: FUSE

Che cos'è

FUSE è un acronimo che sta a significare **F**ilesystem in **u**space.

È un progetto open source il cui scopo principale è quello di realizzare un *modulo per il kernel Linux* che permetta agli utenti non privilegiati di un sistema di *creare un proprio file system senza la necessità di scrivere codice a livello kernel*. In pratica, quindi, si interpone tra l'utente ed il kernel fungendo da ponte e permettendo all'utente di effettuare richieste al kernel usando però un linguaggio di programmazione più user-friendly, che può essere scelto tra tutti quelli supportati (oltre ai classici C e C++, troviamo Java, Python, Perl, Ruby, OCaml, ...). Per fare ciò, il codice del filesystem è eseguito in User Space.

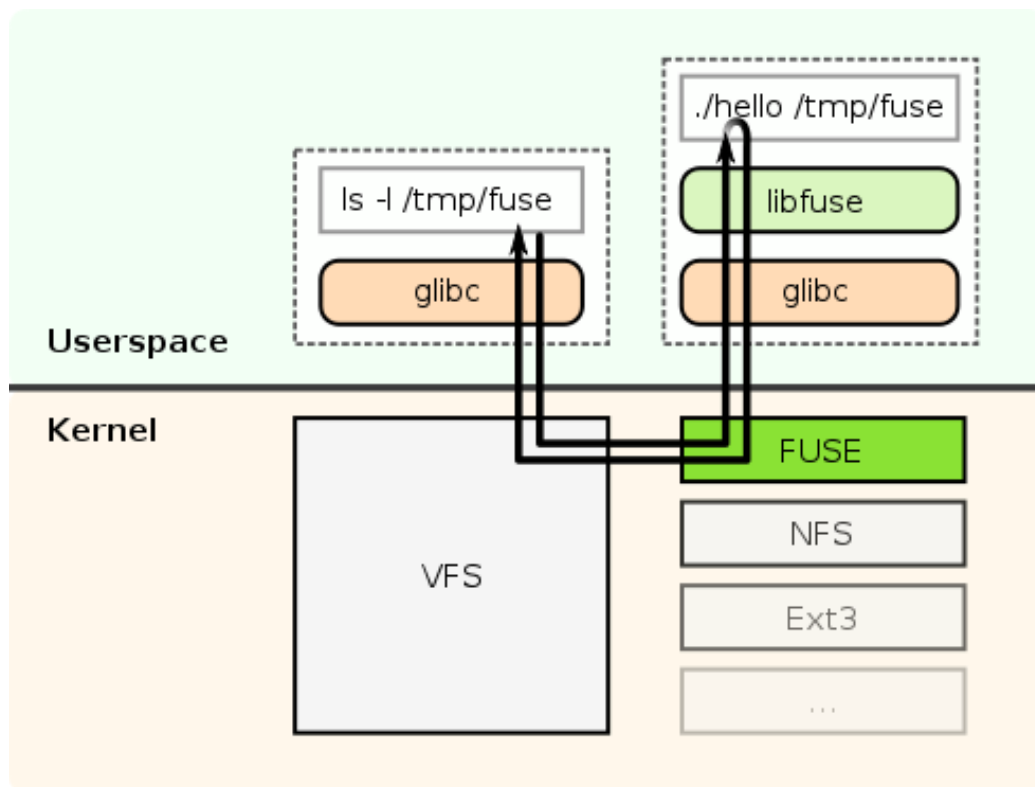
FUSE è particolarmente utile per scrivere *filesystem virtuali*. A differenza dei filesystem tradizionali che si preoccupano principalmente di organizzare e memorizzare i dati su disco, i filesystem virtuali non memorizzano realmente i dati per conto proprio ma come un tramite fra l'utente ed il filesystem reale sottostante.

FUSE è divenuto ufficialmente parte del codice del kernel Linux a partire dalla release 2.6.14, è in grado di supportare la semantica dei più comuni filesystem ed è disponibile per Linux, FreeBSD, OpenSolaris e Mac OS X.

Come funziona

In dettaglio, i principi di funzionamento sono i seguenti:

- a livello utente, la componente che si occupa di gestire il filesystem FUSE si comporta come un demone che riceve richieste dal kernel attraverso uno speciale device;
- a livello kernel, il modulo si occupa di trasformare le richieste per il filesystem FUSE in richieste al demone.



Un filesystem FUSE è un sistema composto da tre elementi:

1. un modulo del kernel
2. una libreria a livello utente per gestire la comunicazione col modulo del kernel
3. una implementazione del filesystem di interesse

Il progetto FUSE fornisce il modulo del kernel e la libreria/interfaccia per la comunicazione col kernel. L'implementazione della struttura filesystem deve essere fornita dallo sviluppatore.

Esempi di filesystems realizzati con FUSE

Esistono decine di filesystems che sono realizzati utilizzando FUSE, più o meno famosi. Tra i più noti ricordiamo:

- **SSHFS:** permette di montare in locale una directory posizionata su un server remoto in cui gira SSH, con il vantaggio di avere una connessione cifrata tramite ssh non intercettabile.

- **EncFS:** un filesystem crittografico che coinvolge due cartelle: la directory di origine, ed il mountpoint. Ogni file nel mountpoint ha un file specifico nella directory di origine che corrisponde ad esso; il file nel mountpoint fornisce la visione non criptata di quello nella directory di origine. I nomi dei file sono criptati nella directory di origine.
- **WikipediaFS:** permette di leggere e modificare gli articoli di Wikipedia (o qualsiasi sito basato su Mediawiki) come se fossero veri file.
- **GmailFS:** permette di montare e usare lo spazio di posta elettronica di GMail come un disco rigido fisico locale.
- **CryptoFS:** un altro filesystem criptato con un principio di funzionamento simile ad EncFS.