

Relazione MyFinger

1. Memorizzare Le Informazioni

La prima cosa che ho fatto andando ad implementare Finger, è stata quella di recuperare tutte le informazioni sugli utenti di cui avevo bisogno. Per prima cosa, mi sono andato a **creare una Struct** chiamata **User**, contenente una serie di campi di diverso tipo, che successivamente avrebbero contenuto tutte le informazioni che il mio Finger doveva stampare. Tra queste informazioni, troviamo **il nome utente**, il **nome reale**, la **directory principale dell'utente** e la **shell dell'utente**. Inoltre, troviamo lo **stato dell'utente**, fondamentale per distinguere se un Utente è loggato nel momento in cui chiamo il comando, oppure no. In realtà ci sono altri 3 campi fondamentali in questa Struct. Infatti, ogni istanza della Struct User, rappresenta un utente nel mio sistema Linux. Ma per realizzare Finger, ho bisogno di tutti gli utenti del sistema. Ho deciso quindi di inserire come campo nella Struct stessa, un **puntatore ad una Struct dello stesso tipo**, in modo tale da ottenere una **lista linkata di User**. Inoltre, Linux ci dice che uno stesso utente, può essere loggato da diversi terminali del sistema, anche contemporaneamente. Per risolvere questa cosa, infatti, ho deciso di inserire all'interno di ogni User, **una sotto-lista linkata di una nuova struct, chiamata Log**, che mi permettesse di tenere traccia di tutti i **terminali in cui è loggato quell'utente e le relative informazioni**. Inoltre, nell'Utente, ho un intero che mi serve per segnalare se un utente è stato stampato o Meno.

2. Ottenere Le Informazioni

Una volta aver creato la Struttura User, devo ottenere le informazioni da memorizzare e successivamente stampare. Sul sistema Linux, gli utenti sono memorizzati nel file di sistema **Passwd**. Per aprire questo file, ho utilizzato la libreria c **Pwd.h**, che tramite una serie di funzioni e ad una Struct dedicata, mi apre il file Passwd e man mano, **itera su tutte le entry** (Utenti) di quel file. Man mano che itero, memorizzo le informazioni sugli utenti nella mia Struct. Ottenuti tutti gli utenti, devo recuperare le informazioni su quelli che sono attualmente loggati. Queste informazioni sono contenute all'interno del file **Utmp**, anch'esso tipico del Sistema Operativo Linux. Per aprirlo ed esplorarlo, ho utilizzato una libreria c chiamata **Utmp.h**. La procedura per ottenere le informazioni su questo file è la stessa di passwd. Infatti, attraverso sempre delle funzioni ed una struct, mi itera su tutte le entry del file e mi restituisce tutti i dati all'interno della struct. Mi basta semplicemente **confrontare lo username** che ho nelle entry dell'utmp con la mia lista di utenti. La particolarità però, sta nel fatto che Utmp mi restituisce le informazioni, non tanto sugli utenti, ma sul **dove sono loggati**. Quindi se un Utente è loggato da due diversi terminali, sarà presente in più entry di questo file. Quindi mi basta iterare sulla mia lista di utenti e

man mano, aggiungere alla sotto-lista linkata di quell'utente, le posizioni in cui è loggato. Tra queste informazioni abbiamo il **tempo in Idle** (calcolato poi successivamente tramite una funzione ausiliaria) e il fatto se una **shell è bloccata o meno** (realizzato con una funzione che va a controllare i permessi sul file della shell)

3. Fare la Ricerca di Un Utente specifico e stampare

La **ricerca** tipica di Finger l'ho implementata in questa maniera: dopo aver ottenuto tutti gli argomenti del mio programma, vado ad **ordinarli in ordine alfabetico**. In questa maniera, riesco a mettere da una parte, gli argomenti con il **-** (ovvero le **opzioni** di Finger) e dall'altra, gli argomenti senza **-** (ovvero i **nomi degli utenti da ricercare**). Una volta fatto ciò, metto tutti quei nomi in una **lista**, e tramite una funzione, li ricerco all'interno della mia lista di utenti, e stampo tutte le informazioni che li riguardano. Le **funzioni che fanno questa cosa sono 2**, una per la stampa **-s** e l'altra per la stampa **-l**. La ricerca cambia a seconda se passo un determinato parametro come argomento (**-m**). Infatti, la sua presenza va a modificare il comportamento della mia ricerca (**case sensitive** oppure **case insensitive**). Per quanto riguarda le opzioni di stampa invece (senza la ricerca di un utente), semplicemente ho **due funzioni (una per il -s ed una per il -l)** che presa la lista di utenti, stampa le informazioni degli utenti loggati.