

FINGER COMMAND (DIY)

Mattia Pandolfi 2087310

Il codice `finger.c` replica il comando `"finger"`, che fornisce informazioni sugli utenti di un sistema in ambiente linux. Il programma accetta opzioni da riga di comando per specificare quali informazioni visualizzare e per quali utenti.

Il codice è scritto in linguaggio C.

Strutture dati e funzioni

nel codice sono definite due strutture dati:

1. `"struct options"` : Rappresenta le opzioni fornite dall'utente tramite la riga di comando. Le opzioni sono memorizzate come valori booleani che indicano se un'opzione è stata selezionata o meno.
2. `"struct UserInfo"` : Contiene le informazioni di un singolo utente, come il nome utente, la directory home, il shell, etc..

il programma implementa diverse funzioni per gestire le operazioni principali:

1. `"longFinger()"` e `"shortFinger()"` : Stampa le informazioni dettagliate o sintetiche di un utente rispettivamente, a seconda delle opzioni fornite.
2. `"get_logged_in_users()"` : Ottiene l'elenco degli utenti attualmente loggati nel sistema.
3. `"phoneFormat()"` : Formatta il numero di telefono nel formato descritto nella documentazione del comando `finger` originale.
4. `"getOptions()"` : Analizza gli argomenti dalla riga di comando e imposta le opzioni corrispondenti.
5. `"strCaseSense()"` : Confronta i nomi utenti presi dagli argomenti con il campo `gecos` in cerca di corrispondenze.
6. `"checkPlanningFile()"` : Recupera le informazioni relative ai file di pianificazione (`".plan"`, `".pgpkey"` e `".project"`) di un utente.
7. `"mailBox()"` : Recupera le informazioni relative alla casella di posta di un utente.
8. `"fingerDIY()"` : esegue e coordina le operazioni principali e stampa le informazioni degli utenti richiesti da riga di comando.

Opzioni supportate

il programma può essere eseguito da riga di comando con diverse opzioni:

- `"-l"` : stampa informazioni dettagliate sull'utente.
- `"-s"` : stampa informazioni sintetiche sull'utente.
- `"-m"` : Considerare solo il nome utente tralasciando il nome completo.
- `"-p"` : Non visualizza informazioni sui file di pianificazione (`".plan"`, `".pgpkey"` e `".project"`) di un utente.

è possibile specificare uno o più nomi utente dopo le opzioni per visualizzare informazioni su quegli utenti.

Logica del programma

Il programma inizierà gestendo le opzioni selezionate dall'utente. Verranno analizzati gli argomenti alla ricerca di elementi che iniziano con il carattere `' - '` per poi identificare a quale opzione si riferiscono e aggiornare di conseguenza la struttura che memorizza le opzioni attive.

Identificate le opzioni, il programma procederà a cercare le informazioni degli utenti nel seguente modo:

1. Il programma scorrerà tutto il file `/etc/passwd` e, per ogni utente, controllerà se c'è una corrispondenza con i nomi utenti presenti negli argomenti (anche con nome e cognome se l'opzione `-m` è disattivata).
2. Se viene trovata una corrispondenza, il programma reperirà le informazioni dell'utente, come nome utente, directory home, shell, ecc. Per ottenere queste informazioni, il programma utilizza la libreria `pwd.h`.
3. Successivamente, per raccogliere le informazioni sull'accesso al sistema (ad esempio tty, ultimo login, ecc.), verrà utilizzata la libreria `utmp.h` per scansionare il file `utmp`.

Queste informazioni verranno salvate in una struttura chiamata `UserInfo`.

In base alle opzioni selezionate, il programma deciderà se stampare la versione dettagliata (`-l`) o la versione ridotta (`-s`). Inoltre, il programma dispone di due funzioni per stampare lo stato della mailbox e i file di pianificazione (`.plan`, `.project`, ecc.).

La funzione `mailbox` richiede solo il nome dell'utente e verifica se esiste il file `/var/mail/username`. Se il file esiste, la funzione controlla se ci sono nuove email o quando è stata letta l'ultima volta; se il file non esiste, indicherà semplicemente che non ci sono email. Confronta il tempo di ultimo accesso con il tempo di ultima modifica. Se il tempo di ultimo accesso è precedente al tempo di ultima modifica, significa che ci sono nuove email non lette. Viene stampata la data di ricezione dell'ultima email e da quanto tempo queste email non sono state lette.

La funzione `checkPlanningFile` viene richiamata tre volte per stampare il contenuto dei tre file di pianificazione `.plan`, `.project` e `.pgpkey`, se presenti.

Da notare che il numero di telefono deve essere stampato in modo diverso e la formattazione varia in base alla lunghezza. La funzione `phoneFormat` prende una stringa che rappresenta il numero di telefono e alloca un buffer per la nuova stringa formattata e utilizza un puntatore per costruirla all'interno di questo buffer. I vari casi dello switch sono in cascata, il che significa che non c'è un'istruzione `break` alla fine dei casi.

Completate queste operazioni, il programma in C termina la sua esecuzione stampando tutte le informazioni previste dal comando `finger`.