

SimpleFS

Report del progetto

Studente: *Emilio Martino 1715650*

Il progetto consiste nella realizzazione di un minimale filesystem, strutturato seguendo una suddivisione in blocchi, della stessa dimensione, di un file binario. Il filesystem consente la creazione e l'accesso ai contenuti di file, che si dividono in **file di testo** e **file di cartella**.

Blocchi e file

I blocchi sono tutti collegati tra loro tramite *linked list*, sono, dunque, adiacenti al precedente e al successivo. Ogni blocco è formato da un **header** e da uno spazio contenente informazioni. Bisogna però distinguere diversi tipi di blocco:

- ◆ **File di testo**, sono formati da:
 - Un primo blocco (*FirstFileBlock*), contenente l'**header**, il **FileControlBlock** (*fcb*) e un array di caratteri contenente informazioni.
 - Blocchi successivi, formati da **header** e l'array di caratteri.
- ◆ **File di cartella**, formati da:
 - Un primo blocco (*FirstDirectoryBlock*), contenente anch'esso l'**header** e il **FileControlBlock**. Contiene anche il numero e i primi blocchi dei file/directory presenti.
 - Blocchi successivi, formati da **header** e un array di primi blocchi.

Operazioni

Si possono eseguire operazioni base fondamentali, come la creazione di file e cartelle, apertura, chiusura ed editing dei file con relativa lettura del contenuto, cambio cartella (*cd*), eliminazione di file e cartelle (compreso il loro contenuto).

Il codice comprende numerosi commenti che aiutano a comprendere gli algoritmi utilizzati, si è data molta priorità alla resa di un codice leggero e schematizzato, che ricorre a funzioni dello standard POSIX.

Esecuzione

All'avvio, viene creato o aperto (se già esistente) il file binario. Viene utilizzata una bitmap per segnare, tramite bit, se uno spazio è utilizzato o libero. E' stato realizzato un disk driver che permette di svolgere le operazioni basilari su blocchi. Il disk driver è responsabile della creazione e dell'inizializzazione del disco (*DiscDriver_init*). Tutte le funzioni su disco comprendono il calcolo automatico dell'offset, necessario per la buona riuscita delle operazioni.

L'intero filesystem viene inizializzato su una *root directory* ("/), corrispondente al blocco 0 del disco.

User-space

E' stata sviluppata una semplice shell per poter svolgere tutti i comandi del filesystem. L'interfaccia ricorda molto l'ambiente UNIX, ricorrendo a una nomenclatura molto simile a quella del terminale. La shell si divide principalmente in tre aree: **test**, **shell** e **editor**. Il test mostra all'utente l'esecuzione di una serie di operazioni, stampando su schermo commenti e risultati. La shell è il core del programma e lascia completa libertà d'uso all'utente. Nella shell c'è l'editor, che permette l'accesso e la modifica ai file di testo, con possibilità di spostare il puntatore.

Per la compilazione, basta eseguire il comando *make*, poi far partire il programma con *./shell*

A.A. 2018/2019