

Relazione progetto

Nome: Matteo Marini

Progetto: File system

What

Progetto numero 5, file system in c; realizzato a partire dallo stub presente nella repository del corso. Il progetto finale è disponibile alla directory https://github.com/Heinzeen/file_system

How

Il progetto si compone fondamentalmente di 4 parti:

1. Bitmap
2. Disk driver
3. File system
4. shell

Bitmap:

La bitmap serve ad indicare quali blocchi del disco sono occupati e quali invece sono liberi; essa viene memorizzata all'inizio del file del disco (un file .dat) e viene modificata ogni volta che un blocco viene liberato o preso per essere utilizzato (quindi in ogni processo di create o di remove).

Disk driver:

Il disk driver serve a fornire al programma l'astrazione del disco a partire da un file. Tutte le funzioni che compongono questo modulo servono a fare da interfaccia tra il file system e il disco (file, in questo caso); in particolare il disk driver dovrà mappare nella memoria del processo pezzi del disco per leggerli e/o modificarli. A questo scopo il disk driver memorizza una lista di pezzi di memoria (da PAGE_SIZE byte, nel mio sistema sono 4096 B) e ogni volta che ha bisogno di fare operazioni sul disco controlla se l'indirizzo a cui deve accedere è tra quelli che ha già mappato o meno; nell'ultimo caso farà una nuova mmap e aggiungerà il nuovo blocco di memoria alla lista, per far sì che non ci sarà bisogno di rifarlo dopo se si dovesse di nuovo aver bisogno di accedere a quel blocco.

Le altre funzioni in questo modulo si occupano dell'avvio e della chiusura del disco (munmap comprese) e della gestione della bitmap (anch'essa mappata in memoria).

File system:

Il file system era l'obiettivo finale di questo progetto; gli altri moduli servono a fornire le astrazioni necessarie per realizzare il file system.

Il file system fa uso di strutture di tipo handler per lavorare con file e cartelle. Sono stati implementati i metodi necessari per creare, modificare, distruggere e leggere il contenuto di file e cartelle.

Shell:

Lo scopo della shell è fornire una modalità interattiva per navigare il file system senza dover studiare le funzioni all'interno dei vari moduli. Quando la shell parte inizializza il disco (senza alterarne i contenuti) e il file system, dopodiché viene data la possibilità di inserire dei comandi per navigare e modificare il file system. Inserendo il comando "help" è possibile vedere la lista dei comandi disponibili, che comprendono il necessario per creare, eliminare, modificare e visualizzare il contenuto di file e cartelle. C'è anche la possibilità di farsi stampare dei dettagli più "a basso livello" di file e cartelle, così facendo verranno mostrate anche informazioni relative ai blocchi che vengono occupati dai relativi file e cartelle, e varie altre informazioni visibili a livello di disco.

How to run

Per far partire il tutto va compilato il codice sorgente (make) e bisogna lanciare fs_shell (./fs_shell).

Il disco è già presente nella repository (in versione formattata) e verrà aperto quello, di default.

All'interno della shell è possibile digitare "help" per ricevere in output la lista dei comandi. La riporto di seguito:

cd, exit, format, info, ls, mkdir, read, rm, touch, write, writerand.