Michele Nicoletti
Matricola n° 1886646

Progetto Sistemi Operativi

• Organizzazione

Il progetto si sviluppa in due programmi:

1. Server
2. Main

Per poter verificare il suo corretto funzionamento, dobbiamo compilare i due programmi tramite il MakeFile e, successivamente, avviare su due terminali distinti l'eseguibile "server" e l'eseguibile "main", nell'ordine citato.

• Struttura

Il progetto si basa sulla struttura **struct fat**, nella quale sono inserite tutte le variabili che dovranno memorizzare le informazioni dei singoli file:

- inode: numero univoco che identifica il file,
- name: nome del file,
- path: percorso del file a partire dalla cartella base del progetto *file_system/*,
- type: tipo specifico, può essere *F* per i file o *D* per le cartelle,
- inode_padre: identifica l'inode della cartella in cui è contenuto il file,
- creator: nome dell'utente che ha creato il file.

Il buffer, messo in condivisione tra le due app, è un *array* di *MAX_INODE* elementi di tipo **struct fat (***MAX_INODE* è una variabile impostata a 128**)**. Il server apporta modifiche all'array attraverso inserimenti ed eliminazioni, mentre il main legge gli elementi di tale buffer.

I programmi comunicano tra di loro attraverso due *fifo*, alle quali accedono uno in lettura e l'altro in scrittura.

• Server

Il server rimane in ascolto dei comandi che l'utente invia tramite main. All'avvio, il server ha il compito di: preparare le risorse necessarie a entrambi i software per poter comunicare (come i semafori e il buffer in condivisione), creare il file *FAT.txt* (il quale registrerà tutte le informazioni relative agli elementi creati dagli utenti) e la directory *file_system* (cartella madre del progetto, nella quale vengono create le cartelle e le directory). Se quest'ultimi esistono, il file *FAT.txt* verrà utilizzato per popolare l'array.

Avviate le risorse, il server procede con la condivisione del buffer attraverso una *shared_memory*. La gestione della memoria condivisa è delegata alla funzione *sharing_father*, la quale prima crea una zona di memoria e, successivamente vi carica il buffer. L'accesso alla *shared_memory* è gestita dai semafori per evitare fenomeni di concorrenza.

Non appena i dati sono messi in condivisione, il server entra nel ciclo di attesa dei comandi inviati dal main. Quando arriva un comando, esso viene analizzato per determinare la sua tipologia (creazione, eliminazione o chiusura) ed eseguito. Dopodiché il server torna in attesa di altri comandi. Nel caso in cui venga ricevuto l'ordine di chiusura, il server esce dal ciclo e predispone la sua fase finale che prevede la chiusura di tutte le risorse prima del terminare.

Michele Nicoletti
Matricola n° 1886646

- Main

È il software con il quale l'utente interagisce, inserendo per primo il nome e successivamente i comandi che devono essere eseguiti dai programmi. I comandi eseguibili sono:

- **cf**, creazione del file
- **ef**, eliminazione del file
- **wf**, scrittura in un file
- **rf**, lettura del contenuto di un file
- **sf**, seeking del file
- **cd**, creazione di una directory
- **ed**, eliminazione di una directory
- **md**, cambio di directory
- **ld**, stampa degli elementi presenti in una directory
- **help**, comando per eseguire stampe informative sui comandi eseguibili
- **exit**, comando di chiusura del programma *main*
- **close**, comando di chiusura dei programmi *main* e *server*

È fondamentale che le richieste siano costituite da un unico elemento, qualsiasi inserimento multiplo viene ignorato. Il nome del file o della directory di interesse viene chiesto dopo aver mandato il comando. Solo nella scrittura nei file è possibile utilizzare più parole. Premendo invio si conclude l'inserimento.

La gestione degli input dell'utente avviene in un ciclo *while*: inserito il comando, esso viene analizzato, eseguito e, in caso, mandato al server (se rientra tra i comandi di gestione di tale programma). Letture e scritture nei file, cambio di directory o stampa degli elementi della directory sono i comandi affidati al main.

Per i file appena creati, è fondamentale effettuare la scrittura dalla posizione 0. Prima di procedere con la lettura o la scrittura di un file, il programma effettua il "seeking" in base al valore in input che l'utente inserisce. È consigliato non andare oltre i limiti del file perché potrebbero verificarsi scritture errate o potrebbero essere negate le letture.