

Manual Técnico - Simulador de Sistema de Archivos EXT2

Descripción de la Arquitectura del Sistema

El sistema está compuesto por dos grandes módulos: el frontend y el backend.

Frontend: Implementado con HTML, Bootstrap y JavaScript. Permite al usuario cargar un archivo .mia, visualizarlo y ejecutarlo mediante un botón. Los resultados se muestran en un textarea de salida que simula una consola.

Backend: Desarrollado en Go. Expone un endpoint '/execute' que recibe scripts, los analiza e interpreta usando el módulo Analyzer. Cada comando es identificado y ejecutado de acuerdo a su tipo, modificando estructuras como el MBR, Superbloques, inodos, etc. Toda la lógica del sistema de archivos EXT2 simulado se ejecuta aquí.

La comunicación entre frontend y backend se realiza vía HTTP POST, donde el script es enviado en formato JSON.

Explicación de las Estructuras de Datos

El sistema utiliza varias estructuras para simular EXT2:

- MBR (Master Boot Record): Contiene información general del disco y sus particiones.
- EBR (Extended Boot Record): Maneja las particiones lógicas dentro de una extendida.
- Superbloque: Información vital del sistema de archivos, como conteo de inodos/bloques, mapas de bits, etc.
- Inodos: Cada archivo o carpeta está representado por un inodo, que contiene metadatos y punteros a bloques.
- Bloques: Almacenan el contenido real de archivos o referencias a carpetas.

Estas estructuras son serializadas y deserializadas desde archivos .mia, simulando un disco real.

Descripción de los Comandos Implementados

Comandos disponibles actualmente:

Manual Técnico - Simulador de Sistema de Archivos EXT2

- MKDISK: Crea un nuevo disco con tamaño y ajuste especificado.
- RMDISK: Elimina un disco existente.
- FDISK: Crea particiones primarias, extendidas o lógicas.
- MOUNT: Monta una partición y le asigna un ID para su uso.
- MKFS: Formatea una partición montada y crea la estructura EXT2.
- LOGIN / LOGOUT: Inicia y cierra sesión como usuario del sistema.
- MKGRP / RMGRP: Crea o elimina grupos de usuarios en users.txt.
- MKUSR / RMUSR: Crea o elimina usuarios del sistema.
- MKFILE: Crea un archivo dentro del sistema, con contenido o tamaño específico.

Cada comando modifica estructuras del disco y puede ser probado desde el frontend. El parser reconoce parámetros con o sin comillas, y puede manejar múltiples líneas en un script.