José Sebastian Pirir Romero - 202300335

Manual de Usuario — GoDisk (Proyecto 2 MIA)

Este documento explica cómo usar la aplicación (CLI + UI web) para crear discos virtuales (.mia), particiones, formatear, gestionar usuarios/grupos, archivos y recuperación. Incluye pasos, ejemplos y resolución de problemas comunes.

Requisitos

- Sistema: Linux
- Go >= 1.18
- Node.js y npm (para la interfaz web)
- Permisos para crear archivos en carpetas usadas por las pruebas (/home/…)

Iniciar la aplicación

1. Compilar backend:
    - Desde la raíz del repo:
        - go build ./...
2. Ejecutar backend en modo servidor:
    - ./
    - Por defecto escucha en http://localhost:3001
3. Iniciar frontend:
    - cd frontend
    - npm install
    - npm start
4. Alternativa: usar el CLI interactivo:
    - go run backend/main.go
    - Escribe comandos (ej.: mkdisk ...) en la consola.

Interfaz web (rápido)

- Header: entrada rápida de comandos.
- CommandArea: escribir scripts o comandos individuales.
- OutputArea: muestra la salida.
- Las páginas de Login / Disks / Partitions / FileBrowser usan los mismos comandos por detrás.

Captura sugerida:

- Tomar screenshot del navegador con la UI abierta (anotar: Header, CommandArea, OutputArea).

José Sebastian Pirir Romero - 202300335

Flujo de uso básico (ejemplo mínimo)

1. Crear disco:
    - mkdisk -size=26 -unit=M -fit=FF -path=/home/usuario/DiscoA.mia
2. Crear particiones:
    - fdisk -type=P -unit=M -name=PartA1 -size=8 -path=/home/usuario/DiscoA.mia -fit=BF
3. Montar partición:
    - mount -path=/home/usuario/DiscoA.mia -name=PartA1
4. Formatear (mkfs):
    - mkfs -type=full -fs=3fs -id=
5. Login (para operar dentro del FS):
    - login -user=root -pass=123 -id=
6. Crear grupos / usuarios:
    - mkgrp -name=usuarios
    - mkusr -user=user1 -pass=pass353 -grp=usuarios
7. Crear carpetas y archivos:
    - mkdir -p -path=/home/user1/documentos/proyectos
    - mkfile -r -path=/home/user1/readme.txt -size=100
8. Leer archivo:
    - cat -file=/home/user1/readme.txt

Ejemplo de script:

```
mkdisk -size=10 -unit=M -path=/home/jose/Disco1.mia
fdisk -type=P -unit=M -name=Part1 -size=8 -path=/home/jose/Disco1.mia
mount -path=/home/jose/Disco1.mia -name=Part1
mkfs -type=full -fs=3fs -id=<ID returned by mount>
login -user=root -pass=123 -id=<ID>
mkgrp -name=usuarios
mkusr -user=user1 -pass=pass123 -grp=usuarios
mkdir -p -path=/home/user1/docs
mkfile -path=/home/user1/docs/readme.txt -cont=/home/jose/fuente.txt
cat -file=/home/user1/docs/readme.txt
```

Comando `copy`

Sintaxis:

- copy -path=/ruta/origen -destino=/ruta/destino

Notas:

- Si `-destino` apunta a carpeta existente, el archivo/carpeta se copia dentro manteniendo nombre base.
- Si `-destino` no existe, se interpreta como ruta de archivo (se busca el padre).
- Se requiere permiso de lectura en origen y de escritura en carpeta destino.

- En carpetas, la copia es recursiva.

Recovery (recuperación)

Comando:

- recovery -id=

Función:

- Reconstruye bitmaps y contadores del superbloque escaneando la tabla de inodos y los bloques referenciados por éstos.
- No aplica por defecto el replay de journaling (opcional, puede implementarse después).

Precaución: probar en discos de prueba antes de usar en datos reales.

Solución de problemas comunes

1. users.txt truncado o textos cortados
    - Causa habitual: cambiaste el tamaño de B_name (NAME_MAX) u otras structs que afectan el layout en disco sin reformatear.
    - Solución:
        - Asegúrate que `structs.FileBlock` es al menos tan grande como `FolderBlock` y que mkfs usa binary.Size para calcular S_block_size.
        - Recompila (go build ./...) y vuelve a formatear la partición con `mkfs` (el formato en disco cambió).
        - Evita usar discos con layout antiguo tras cambiar structs.
2. "No tienes permiso" al crear/escribir
    - Verifica la sesión: ejecutar `login` con usuario adecuado.
    - Revisa permisos del inodo padre (propietario, grupo, permisos UGO).
3. "Ruta no encontrada" al copiar o crear archivo
    - Si no usaste `-r` o `-p`, los directorios padre deben existir.
    - Para crear rutas recursivas, usar flags `-r` o `-p` según el comando.
4. Error al montar varias veces la misma partición
    - Usa `mounted` para listar montajes y evita duplicados.
5. Cambio de estructuras en structs
    - Cada cambio requiere recompilar y volver a formatear las particiones de prueba.
    - Comprobar con:
        - grep -R --line-number "[12]byte" .
        - go build ./...

José Sebastian Pirir Romero - 202300335

Capturas y ejemplos (sugerencias)

- Incluir estas capturas en el manual:
    - Pantalla principal del UI con un script ejecutado y salida.
    - Resultado de `cat -file=/users.txt` mostrando usuarios y grupos.
    - Ejecución de `copy` y comprobación del archivo destino.
    - Salida del comando `recovery`.

Cómo generar capturas:

- En Linux usa `gnome-screenshot` o `scrot`.
- Guardar en `docs/screenshots/` y referenciarlas en el manual con ruta relativa.

Ejemplo Markdown para incluir imagen:

```
![UI principal](docs/screenshots/ui_main.png)
```

FAQ (rápido)

- P: ¿Por qué cambió el tamaño del archivo en disco después de modificar structs?
    - R: Porque el layout on-disk depende de las structs; hay que re-formatear.
- P: ¿Cómo depuro lecturas/escrituras corruptas?
    - R: Usar `recovery` y revisar logs; verificar que `fs.Read...` y `fs.Write...` usan `sb.S_block_size`.
- P: ¿Puedo recuperar operaciones no finalizadas?
    - R: Sí si implementas replay del journaling: leer JournalEntry y aplicar operaciones pendientes tras reconstruir bitmaps.

Apéndice — Lista de comandos principales

- mkdisk, rmdisk
- fdisk
- mount, unmount, mounted
- mkfs
- login, logout
- mkgrp, rmgrp, mkusr, rmusr, chgrp
- mkdir, rm, copy, move, rename
- mkfile, cat, edit
- chmod, chown
- find
- rep (reportes)
- journaling, recovery