

Proyecto
Black & White Filesystem

Noviembre, 2025



Objetivo

Realizar una re-implementación de algunas de las funciones de un filesystem en el espacio de usuario del Sistema Operativo GNU/Linux.

Datos Generales

- **Fecha de Entrega:**
Ver en Plataforma.
- **Fecha de Revisión:**
Virtual.
- **Lenguaje:**
C o Rust para GNU/Linux
- **Recurso Humano:**
Grupos de 2 o 3
- **Valor de la asignación:** 30 %

Profesor

Kevin Moraga
kmoragas@ic-itcr.ac.cr
Escuela de Computación

Introducción

BWFS es un sistema de archivos que reside en el espacio de usuario. Este sistema de archivos tiene como objetivo utilizar los imágenes blanco y negro para almacenar archivos. Para BWFS el espacio físico donde se almacena su información y toda su estructura se encuentra en pixels de color blanco y negro, definidos en la creación del FS.

Requerimientos Funcionales

BWFS

Se realizará la re-implementación de las siguientes funciones utilizando la biblioteca FUSE:

- getattr
- create
- open
- read
- write
- rename
- mkdir
- readdir
- opendir
- rmdir
- statfs
- fsync
- access
- unlink
- flush
- lseek

mkfs.bwfs

Este binario consiste en la creación de un nuevo sistema de archivos tipo BWFS. Teniendo en cuenta lo siguiente:

- Cada bloque de BWFS podrá tener un máximo de 1000 px por 1000 px.
- El sistema de archivos deberá de utilizar i-nodos como estructura de indexación de bloques.
- El sistema de archivos deberá ser distribuido.
- Deberá poder obtener la configuración desde un ini.
- La sintaxis será:

```
mkfs.bwfs -c config.ini
```

mount.bwfs

Se encarga de montar el BWFS en algún punto de montaje perteneciente al FS del Sistema Operativo.

- Para el usuario es posible especificar cual archivo corresponde al inicio del FS o bien, el FS se encargará de encontrar un fingerprint.
- La sintaxis será:

```
mount.bwfs -c config.ini mountpoint/
```

Requerimientos Técnicos Adicionales

- Es necesario utilizar la biblioteca FUSE para la implementación.
- El Sistema de Archivos debe ser persistente en disco.
- Será posible crear archivos de cualquier tipo y cualquier tamaño (siempre y cuando sea menor al tamaño definido en la creación).
- Debe poderse comunicar por TCP/IP

Aspectos Administrativos

Entregables

- Código fuente del programa que cumpla los requerimientos funcionales y técnicos.
- Código fuente de mkfs.bwfs
- Código fuente de mount.bwfs
- Código fuente de BWFS.
- Binario del programa, compilado para una arquitectura x86.
- Fuente de la documentación en Markdown o en Latex y luego a PDF.
- PDF con la documentación.
- Un sistema de archivos impreso.
- Video con la demo del funcionamiento.

Documentación

Las siguientes son las instrucciones para la documentación. NO LA IMPRIMA. Además la documentación se debe de realizar utilizando MD con Latex.

1. **Introducción:** Presentar el problema. Puede “reciclar” partes del enunciado de la tarea programada.
2. **Ambiente de desarrollo:** Indicar las herramientas usadas para implementar la tarea.
3. **Estructuras de datos usadas y funciones:** Se debe describir las principales funciones y estructuras utilizadas en la elaboración de esta asignación.
4. **Instrucciones para ejecutar el programa:** Presentar las consultas concretas usadas para correr el programa para el problema planteado en el enunciado de la tarea y para los casos planteados al final de esta documentación.
5. **Actividades realizadas por estudiante:** Este es un resumen de las bitácoras de cada estudiante (estilo timesheet) en términos del tiempo invertido para una actividad específica que impactó directamente el desarrollo del trabajo, de manera breve (no más de una línea) se describe lo que se realizó, la cantidad de horas invertidas y la fecha en la que se realizó. Se deben sumar las horas invertidas por cada estudiante, sean conscientes a la hora de realizar esto el profesor determinará si los reportes están acordes al producto entregado.
6. **Autoevaluación:** Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales. Adicionalmente debe de incluir el reporte de commits de git. Por otro lado, también debe incluir una calificación con la rúbrica de la sección “Evaluación” y “Autoevaluación” con cada ítem evaluado de 0 a 10.
7. **Lecciones Aprendidas** del proyecto: Orientados a un estudiante que curse el presente curso en un futuro.
8. **Bibliografía** utilizada en la elaboración de la presente asignación.
9. Es necesario documentar el código fuente.

Evaluación

- BWFS:
 - mkfs.bwfs: 14 %
 - mount.bwfs: 15 %
 - Funciones de la biblioteca: 26 %.
 - getattr

- create
- open
- read
- write
- rename
- mkdir
- readdir (opcional)
- opendir (opcional)
- rmdir (opcional)
- statfs
- fsync
- access
- unlink
- flush
- lseek

- Documentación: 20 %
- Persistencia en Disco: 25 %

Aspectos Adicionales

Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:

1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
2. Si el código no compila se obtendrá una nota de 0, por lo cuál se recomienda realizar la defensa con un código funcional.
3. El código debe ser desarrollado en el lenguaje especificado en los Datos Generales, en caso contrario se obtendrá una nota de 0.