

Universidad Simón Bolívar  
Departamento de Computación y Tecnología de la Información  
CI-3825 Sistemas de Operación 1  
Enero - Marzo 2025



## INFORME PROYECTO 3

Integrantes:  
Franco Murillo 16-10782  
Jesús Cuéllar 15-10345

Abril, 2025

## Descripción del programa

El programa se dividió en 4 archivos:

- **file\_ops.h**: Contiene las definiciones de las funciones usadas para realizar la sincronización de los directorios.
- **file\_ops.c**: Contiene las funciones que se usarán para la sincronización de los directorios:
  - **cp\_file\_to\_dir(const char \*src\_file, const char \*dest\_dir)**: copia el archivo 'src\_file' al directorio 'dest\_dir' creando en éste un nuevo archivo con el mismo nombre que el origen.
  - **cp\_dir\_to\_dir(const char \*src\_dir, const char \*dest\_dir)**: copia recursivamente el directorio 'src\_dir' al directorio 'dest\_dir'. Se crea en dest\_dir un subdirectorio con el mismo nombre que src\_dir.
  - **rm\_dir(const char \*dir\_path)**: elimina recursivamente el contenido de un directorio. Se borran todos los archivos y subdirectorios contenidos en 'dir\_path'.
  - **same\_content\_file(const char \*file1, const char \*file2)**: compara el contenido de dos archivos. Si poseen el mismo tamaño y todos los bloques leídos son idénticos, retorna 1; de lo contrario, retorna 0.
  - Como la función que sincroniza los dos directorios fue tan larga, se decidió implementarla en un archivo separado

- **sync.c**: contiene la función que sincroniza los dos directorios, así como la función que muestra las estadísticas de transferencias entre los directorios:

- **sum\_dir\_files(const char \*dir\_path, unsigned long \*file\_count, unsigned long \*total\_bytes)**: recorre recursivamente un directorio y acumula en \*file\_count el número de archivos regulares y en \*total\_bytes la suma en bytes de sus tamaños.

dir\_path : ruta del directorio a recorrer.

file\_count : puntero donde se acumulará la cantidad de archivos.

total\_bytes: puntero donde se sumarán los bytes de cada archivo.

Retorna 0 en éxito o -1 en caso de error.

- **sync\_dirs(const char \*d1, const char \*d2)**: sincroniza el contenido de los directorios d1 y d2 según el siguiente esquema:

1) En nivel 1 se recorre cada entrada (archivo o directorio) de d1.

- Si un elemento existe en d1 y NO en d2, se le pregunta al usuario si desea copiarlo a d2 o borrarlo de d1

- Si el elemento existe en ambos:

\* Si se trata de archivos regulares, se los compara (por tamaño y contenido)

y en caso de diferencia se revisa la fecha de modificación para recomendar

la copia del archivo más reciente. Luego se pregunta cuál versión conservar.

\* Si son directorios, se invoca recursivamente sync\_dirs para sincronizarlos.

2) Se recorre d2 para identificar aquellos elementos que existen en d2 y NO en d1,

y se realiza una operación similar (preguntar al usuario si se desea copiar a d1 o borrar).

Durante las operaciones de copia, se actualizan variables globales que acumulan las estadísticas:

- files\_d1\_to\_d2 / kilobytes\_d1\_to\_d2: transferencias de d1 hacia d2.

- files\_d2\_to\_d1 / kilobytes\_d2\_to\_d1: transferencias de d2 hacia d1.

- **main.c:** recibe los argumentos (los dos directorios), verifica que los caminos proporcionados por el usuario sean efectivamente caminos a directorios válidos, y llama a las funciones que sincroniza los directorios, y que muestra las estadísticas de transferencia.