

Descripción general de Proyecto “Virtual File System”

Periodo: (Semestre 1, Periodo 4, Año 2019)

Ing. Juan Carlos Zepeda

Se requiere desarrollar en C++ un sistema de archivos virtual, el cual pueda realizar operaciones básicas sobre directorios, además de gestionar el espacio en disco de archivos importados al medio.

Sistema de Archivos

El sistema de archivos proporcionará un mecanismo para el almacenamiento y acceso al contenido de los archivos en un disco virtual (archivo binario).

Se tomará como base la estructura extendida (i-nodo) del kernel de Linux ext2. (ver [Anexo 1](#))

Cada entrada del directorio, tendrá quince (15) apuntadores para la gestión del espacio en disco: Los primeros doce (12), serán apuntadores a bloques de datos de forma directa. Los siguientes tres hacen referencia a apuntadores de bloques indirectos: (ver [Anexo 2](#))

- a) Indirecto de un nivel
- b) Indirecto de dos niveles
- c) Indirecto de tres niveles

Como parte de la estructura de un archivo/directorio se tomará en cuenta los siguientes atributos:

1. Name: Nombre del archivo/directorio creado
2. Type: Tipo de objeto (Archivo o Directorio)
3. Date: Fecha de creación del archivo/directorio
4. Size: Tamaño total en bloques de datos, del archivo/directorio.
5. 12 apuntadores unsigned int a bloques de datos indirectos
6. 1 apuntador a bloque de índice de un nivel
7. 1 apuntador a bloque de índice de dos niveles
8. 1 apuntador a bloque de índice de tres niveles

¡IMPORTANTE! Debe proponer una estructura de directorio multinivel, que le permita crear archivos y directorios, dentro de otro directorio hasta agotar las entradas de directorio disponibles.

Operaciones sobre directorios/archivos

Las operaciones sobre archivos o directorios, se harán mediante un shell (línea de comandos) o interprete de ordenes donde se tendrá acceso solamente a los siguientes comandos:

1. **create disc <<disc_name>>**: Crea un nuevo disco virtual (archivo binario), con el formato de estructura de archivos y gestión de espacio en disco vacío. Señalar las entradas de directorio disponibles por disco. Crear los bloques de datos necesarios para satisfacer la cantidad de directorios a crear.
2. **mkdir <<directory>>**: Crea un directorio vacío en el directorio actual.
3. **cd <<directory>>**: Cambia de directorio, a uno existente dentro del subdirectorio. Para navegar al directorio anterior, podrá colocar “dos puntos” (. .) Ej. **cd . .**

4. **ls**: Lista todos los archivos y directorios contenidos en un directorio actual.
5. **rm «file | «directory »**: Elimina el archivo o directorio señalado. Si es directorio, elimina de manera recursiva todos los archivos y directorios contenidos.
6. **Import «filename path»**: Importa el archivo de Windows/Linux al disco virtual, en la carpeta actual.
7. **Export «filename path»**: Exporta el archivo de la carpeta actual del disco virtual, a la dirección del Sistema operativo.

Gestión del espacio en disco

Para todos aquellos archivos que se importarán al sistema de archivos virtual, se contará con los siguientes bloques de datos:

- a) Bloques de datos: Tamaño de bloque de **4K (4096 bytes)**.
- b) Bloques indirectos de un nivel: Cada bloque indirecto de un nivel, podrá direccionar hasta **16** bloques de datos.
- c) Bloques indirectos de dos niveles: Cada bloque indirecto de dos niveles, podrá direccionar hasta **32** bloques indirectos de un nivel.
- d) Bloques indirectos de tres niveles: Cada bloque indirecto de tres niveles, podrá direccionar hasta **64** bloques indirectos de dos niveles.

¡IMPORTANTE!: La gestión del espacio libre, se hará mediante mapa de bits o vector de bits para cada tipo de bloque de datos (directos, indirectos de uno, dos y tres niveles respectivamente), en donde cada bloque estará representado por 1 bit. Si el bloque está libre, el bit será igual a 0; si el bloque está asignado, el bit será 1. (Ver [Anexo 3](#))

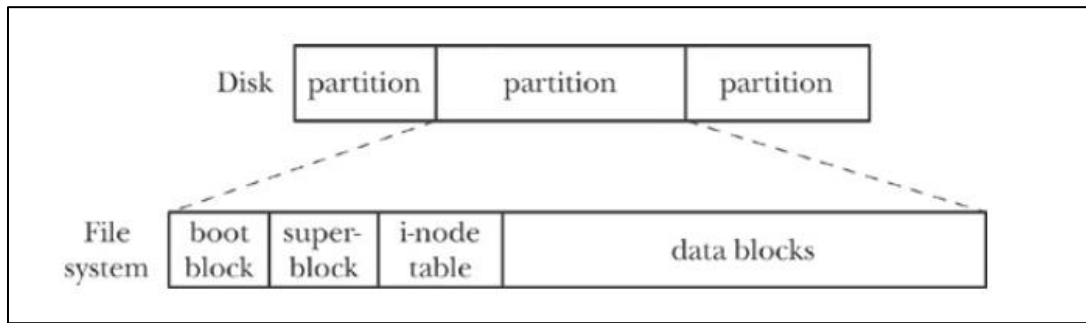
Cuando un archivo se elimine, los bloques de datos directos e indirectos se liberan y se debe actualizar los vectores de bits para indicar que los bloques de datos están disponibles para ser utilizados por otro archivo.

Hacer uso de operadores a nivel de bits, para indicar el bit correcto a encender para un número de bloque indicado.

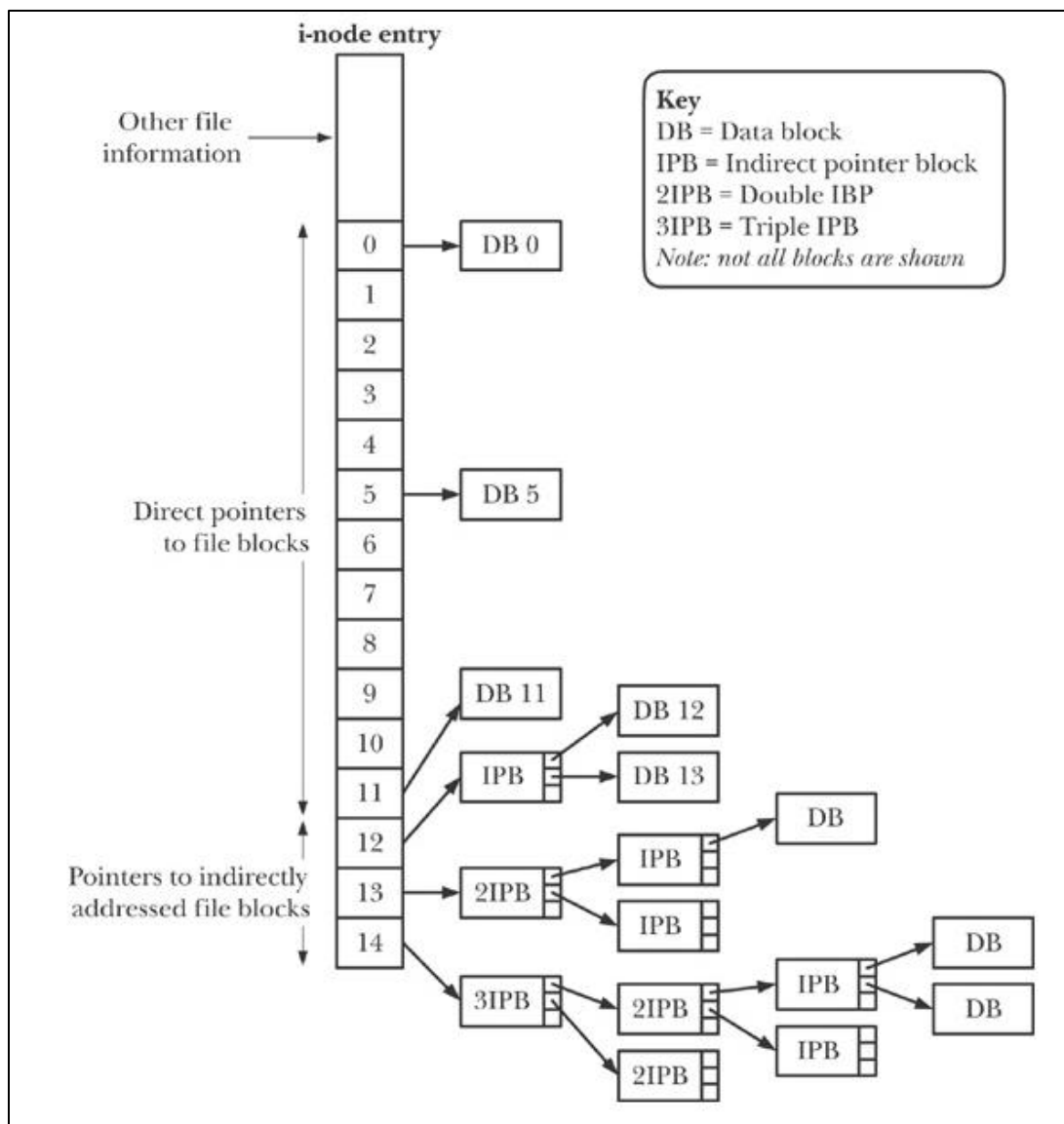
Condiciones de entrega

- Se debe entregar un proyecto de aplicación de consola C++.
- El proyecto será entregado a mas tardar el lunes de la Semana 10 en el portal de la clase, antes de la hora clase.
- El proyecto es individual.
- Los proyectos con alto grado de similitud en código tendrán cero sin derecho a reclamo.
- Todos los puntos del proyecto se ganarán en la defensa. El simple hecho de entregar un programa no tendrá ningún puntaje.
- Por cada día de retraso de la entrega el proyecto perderá el 33% de su valor total.

ANEXOS



Anexo 1 - Estructura de ext2



Anexo 2 - Estructura de un i-nodo

Free Space Management

- Bit Vector management



- One bit for each block
 - 0 = free; 1 = occupied
- Use bit manipulation commands to find free block

Anexo 3 - Gestión del espacio libre mediante vector de Bits

Rubrica de Evaluación

Característica		Excelente (100%)	Notable (75%)	Suficiente (50%)	Insuficiente (25%)	No Desarrollado (0%)
Creación de disco virtual (create disc «disc_name») <ul style="list-style-type: none"> - Estructura de superbloque - Estructura de entradas i-nodo - Estructuras de Bloques <ul style="list-style-type: none"> o Datos o Indirecto 1 Nivel o Indirecto 2 nivel o Indirecto 3 Nivel - Mapa de bits de bloques disponibles 	3	Permite crear en un archivo binario, las estructuras de datos solicitadas para hacer funcionar el sistema de directorio: <ul style="list-style-type: none"> - Estructura de superbloque - Estructuras de entradas i-nodo - Estructuras de bloques - Mapa de bits de bloques disponibles 	Permite crear en un archivo binario, las estructuras de datos básicas para hacer funcionar el sistema de directorio: <ul style="list-style-type: none"> - Estructura de superbloque - Estructuras de entradas i-nodo - Estructuras de bloques 	Permite crear en un archivo binario, las estructuras de datos esenciales para hacer funcionar el sistema de directorio: <ul style="list-style-type: none"> - Estructura de superbloque - Estructuras de entradas i-nodo 	Permite crear en un archivo binario, las estructuras de datos mínimas para hacer funcionar el sistema de directorio: <ul style="list-style-type: none"> - Estructuras de entradas i-nodo 	No permite crear en un archivo binario, las estructuras de datos necesarias para hacer funcionar el sistema de directorio. No se permite manejar estructuras en memoria RAM.
Operaciones sobre directorios <ul style="list-style-type: none"> - mkdir - cd - ls - rm 	4	Permite a partir de una interfaz apropiada de línea de comandos, las operaciones solicitadas para trabajar en directorios multinivel. <ul style="list-style-type: none"> - Crea directorio - Cambia directorio - Lista elementos en directorio - Elimina archivos/directorios 	Permite a partir de una interfaz apropiada de línea de comandos, las operaciones básicas para crear directorios multinivel. <ul style="list-style-type: none"> - Crea directorio - Cambia directorio - Elimina archivos/directorios 	Permite a partir de una interfaz apropiada de línea de comandos, las operaciones esenciales para crear directorios multinivel. <ul style="list-style-type: none"> - Crea directorio - Cambia directorio - Lista elementos en directorio 	Permite a partir de una interfaz apropiada de línea de comandos, las operaciones mínimas para crear directorios multinivel. <ul style="list-style-type: none"> - Crea directorio - Cambia directorio 	No permite mediante una interfaz de línea de comandos, operaciones para trabajar en directorios multinivel.
Operaciones sobre archivos <ul style="list-style-type: none"> - Import - export 	2	Permite importar y exportar mediante los comandos apropiados archivos desde y hasta el sistema de directorio virtual, haciendo uso de la gestión de espacio en disco.	Permite importar y exportar parcialmente mediante comandos apropiados archivos desde y hasta el sistema de directorio virtual, haciendo uso de la gestión de espacio en disco.	Permite solamente importar archivos mediante comandos apropiados hasta el sistema de directorio virtual, haciendo uso de la gestión de espacio en disco.	Permite importar parcialmente archivos mediante comandos apropiados hasta el sistema de directorio virtual, haciendo uso de la gestión de espacio en disco.	No permite importar y exportar archivos mediante comandos.

Característica		Excelente (100%)	Notable (75%)	Suficiente (50%)	Insuficiente (25%)	No Desarrollado (0%)
Gestión del espacio en disco - Asignación de bloques directos - Asignación de bloques indirectos de 1, 2 y 3 nivel - Liberación de bloques al eliminar archivos - Actualización de Mapa de bits para bloques liberados al eliminar archivos	6	Permite la asignación apropiada de bloques directos a los apuntadores de cada entrada de i-nodo, además de la asignación de bloques indirectos de 1, 2 y 3 nivel. Permite la liberación de bloques usados, al eliminar se actualiza mapa bits de bloques disponibles.	Permite la asignación apropiada de bloques directos a los apuntadores de cada entrada de i-nodo, además de la asignación de bloques indirectos de 1 y 2 nivel. Permite la liberación de bloques usados, al eliminar se actualiza mapa bits de bloques disponibles.	Permite la asignación apropiada de bloques directos a los apuntadores de cada entrada de i-nodo, además de la asignación de bloques indirectos de 1 nivel. Permite la liberación de bloques usados, al eliminar se actualiza mapa bits de bloques disponibles.	Permite la asignación apropiada de bloques directos a los apuntadores de cada entrada de i-nodo. Permite la liberación de bloques usados, al eliminar se actualiza mapa bits de bloques disponibles.	No permite la asignación apropiada de bloques directos e indirectos de 1, 2 y 3 nivel por cada entrada de i-nodo.
Total	15					

Subir proyecto desarrollado en C++ en enlace de entrega de Semana 5.

El enlace para subir el URL será “[Proyecto I – Virtual File System](#)” en la semana 10 de la clase en el portal.

La tarea se presentará como fecha máxima el día lunes de la Semana 10, antes de la hora de clase.