

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas
Manejo e Implementación de Archivos
Escuela de vacaciones Junio 2020
Catedráticos: Ing. Oscar Paz
Tutor académico: Diego Osorio



Proyecto 1 - Fase 1

Introducción:

El curso de Manejo e Implementación de Archivos busca que el estudiante aprenda los conceptos sobre la administración de archivos, tanto en hardware como software, sistemas de archivos, particiones, entre otros conceptos, para después introducirse en las bases de datos. El proyecto busca que el estudiante aplique estos conceptos y pueda aprenderlos implementando su funcionalidad.

Objetivos:

- Aprender a administrar archivos y escribir estructuras en C++
- Comprender el sistema de archivos EXT2 y EXT3
- Aplicar el formateo rápido y completo en una partición
- Crear una aplicación de comandos
- Aplicar la teoría de ajustes
- Aplicar la teoría de particiones
- Utilizar GraphViz para mostrar reportes

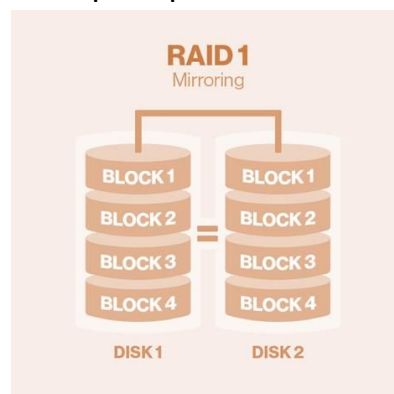
Aplicación de Comandos:

La aplicación será totalmente en consola, a excepción de los reportes en Graphviz. Esta no tendrá menús, sino que se utilizarán comandos. No distinguirá entre mayúsculas y minúsculas. Hay parámetros obligatorios y opcionales. Solo se puede colocar un comando por línea.

Si se utiliza un parámetro que no está especificado en este documento, debe mostrar un mensaje de error. Se utilizarán espacios en blanco para separar cada parámetro. Si se necesita que algún valor lleve espacios en blanco se encerrará entre comillas " ". **Los parámetros pueden venir en cualquier orden.**

RAID:

Para la primera fase se requiere que el estudiante aplique los conocimientos sobre raid para lo cual se requiere que el alumno desarrolle RAID por software para la creación de discos duros y todo lo que conlleva su administración de manejo. El tipo asignado a desarrollar es un RAID 1 en el cual el estudiante deberá desarrollar lo necesario para poder realizar la administración de discos.



El proceso a seguir para la administración de los RAID será el siguiente:

- Creación del disco: Por cada disco creado dentro del sistema se deberá crear uno idéntico (Con el mismo tamaño) el cual funcionará como espejo.
- Operaciones sobre disco: Por cada operación que se realice en el disco que el usuario cree en la consola de comandos, se deberá replicar esta operación sobre el disco creado de manera automática, esto para realizar la simulación de un RAID 1, creado por software.

ADMINISTRACIÓN DE DISCOS:

Estos comandos permitirán crear archivos que simularán discos duros en los que se podrá formatear más adelante con el sistema de archivos ext2 o ext3. Estos comandos estarán disponibles desde que se inicia el programa. Estos comandos son:

1.MKDISK

Este comando creará un archivo binario que simulará un disco duro, estos archivos binarios tendrán la extensión disk y su contenido al inicio será 0. Deberá ocupar físicamente el tamaño indicado por los parámetros, (no importa que el sistema operativo no muestre el tamaño exacto). Recibirá el nombre del archivo que simulará el disco duro y tendrá los siguientes parámetros:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-size	Obligatorio	Este parámetro recibirá un número que indicará el tamaño del disco a crear. Debe ser positivo y mayor que cero, si no se mostrará un error.
-fit	Opcional	Indicará el ajuste que utilizará el disco para crear las particiones dentro del disco Podrá tener los siguientes valores: BF: Indicará el mejor ajuste (Best Fit) FF: Utilizará el primer ajuste (First Fit) WF: Utilizará el peor ajuste (Worst Fit) Ya que es opcional, se tomará el

		<p>primer ajuste si no está especificado en el comando. Si se utiliza otro valor que no sea alguno de los anteriores mostrará un mensaje de error.</p>
-unit	Opcional	<p>Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:</p> <p>k que indicará que se utilizarán Kilobytes (1024 bytes)</p> <p>m en el que se utilizarán Megabytes (1024 * 1024 bytes)</p> <p>Este parámetro es opcional, si no se encuentra se creará un disco con tamaño en Megabytes. Si se utiliza otro valor debe mostrarse un mensaje de error.</p>
-path	Obligatorio	<p>Este parámetro será la ruta en el que se creará el archivo que representará el disco duro. Si las carpetas de la ruta no existen deberán crearse.</p>

Ejemplos:

```
#Crea un disco de 3000 Kb en la carpeta home
Mkdisk -Size=3000 -unit=K -path=/home/user/Disco1.disk
```

```
#No es necesario utilizar comillas para la ruta en este caso ya
que la ruta no tiene ningún espacio en blanco
Mkdisk -path=/home/user/Disco2.disk -Unit=K -size=3000
```

```
#Se ponen comillas por la carpeta "mis discos" ya que tiene espacios en blanco, se crea si no está no existe
mkdisk -size=5 -unit=M -path="/home/mis discos/Disco3.disk"
```

```
#Crearé un disco de 10 Mb ya que no hay parámetro unit
Mkdisk -size=10 -path="/home/mis discos/Disco4.disk"
```

2. RMDISK

Este parámetro elimina un archivo que representa a un disco duro mostrando un mensaje de confirmación para eliminar. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-path	Obligatorio	Este parámetro será la ruta en el que se eliminará el archivo que representará el disco duro. Si el archivo no existe, debe mostrar un mensaje de error.

Ejemplo:

```
#Elimina Disco4.disk
rmDisk -path="/home/mis discos/Disco4.disk"
```

3. FDISK

Este comando administra las particiones en el archivo que representa al disco duro. Deberá mostrar un error si no se pudo realizar la operación solicitada sobre la partición, especificando por qué razón no pudo crearse (Por espacio, por restricciones de particiones, etc.).

No se considerará el caso de que se pongan parámetros incompatibles, por ejemplo, en un mismo comando fdisk llamar a delete y add. La estructura de cada disco se explicará más adelante. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-size	Obligatorio al crear	Este parámetro recibirá un número que indicará el tamaño de la partición a crear. Debe ser positivo y mayor a cero, de lo contrario se mostrará un mensaje de error.
-unit	Opcional	<p>Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:</p> <p>B: indicará que se utilizarán bytes.</p> <p>K: indicará que se utilizarán Kilobytes(1024 bytes)</p> <p>M: indicará que se utilizarán Megabytes(1024 * 1024 bytes).</p> <p>Este parámetro es opcional, si no se encuentra se creará una partición en Kilobytes. Si se utiliza un valor diferente mostrará un mensaje de error.</p>
-path	Obligatorio	Este parámetro será la ruta en la que se encuentra el disco en el que se creará la partición. Este archivo ya debe existir, si no se mostrará un error.
-type	Opcional	<p>Indicará que tipo de partición se creará. Ya que es opcional, se tomará como primaria en caso de que no se indique. Podrá tener los siguientes valores:</p> <p>P: en este caso se creará una partición primaria.</p> <p>E: en este caso se creará una partición extendida.</p> <p>L: Con este valor se creará una partición lógica. Las particiones lógicas sólo pueden estar dentro de la extendida sin sobrepasar su tamaño. Deberá tener en cuenta las restricciones de teoría de particiones:</p> <p>La suma de primarias y extendidas debe ser como máximo 4. Solo puede haber una partición extendida por disco.</p> <p>No se puede crear una partición lógica si no hay una extendida.</p> <p>Si se utiliza otro valor diferente a los anteriores deberá mostrar un mensaje de error.</p>

-fit	Opcional	<p>Indicará el ajuste que utilizará la partición para asignar espacio en la segunda fase del proyecto. Podrá tener los siguientes valores:</p> <p>BF: Indicará el mejor ajuste (Best Fit) FF: Utilizará el primer ajuste (First Fit) WF: Utilizará el peor ajuste (Worst Fit)</p> <p>Ya que es opcional, se tomará el peor ajuste si no está especificado en el comando. Si se utiliza otro valor que no sea alguno de los anteriores mostrará un mensaje de error.</p>
-delete	Opcional	<p>Este parámetro indica que se eliminará una partición. Este parámetro se utiliza junto con -name y -path. Se deberá mostrar un mensaje que permita confirmar la eliminación de dicha partición.</p> <p>Si la partición no existe deberá mostrar error. Si se elimina la partición extendida, deben eliminarse las particiones lógicas que tenga adentro.</p> <p>Recibirá los siguientes valores:</p> <p>Fast: Esta opción marca como vacío el espacio en la tabla de particiones.</p> <p>Full: Esta opción además marcar como vacío el espacio en la tabla de particiones, rellena el espacio con el carácter \0. Si se utiliza otro valor diferente, mostrará un mensaje de error.</p>
-name	Obligatorio	<p>Indicará el nombre de la partición. El nombre no debe repetirse dentro de las particiones de cada disco. Si se va a eliminar, la partición ya debe existir, si no existe debe mostrar un mensaje de error.</p>

-add	Opcional	<p>Este parámetro se utilizará para agregar o quitar espacio de la partición. Puede ser positivo o negativo. Tomará el parámetro -units para las unidades a agregar o eliminar.</p> <p>En el caso de agregar espacio, deberá comprobar que exista espacio libre después de la partición. En el caso de quitar espacio se debe comprobar que quede espacio en la partición (no espacio negativo).</p>
------	----------	---

Ejemplos:

```
#Crea una partición primaria llamada Particion1 de 300 kb
#con el peor ajuste en el disco Disco1.disk
fdisk -Size=300 -path=/home/Disco1.disk -name=Particion1
#Crea una partición extendida dentro de Disco2 de 300 kb
#Tiene el peor ajuste
fdisk -type=E -path=/home/Disco2.disk -Unit=K -name=Particion2
-size=300
#Crea una partición lógica con el mejor ajuste, llamada Paricion3,
#de 1 Mb en el Disco3
fdisk -size=1 -type=L -unit=M -fit=BF -path="/mis
discos/Disco3.disk" -name="Particion3"
#Intenta crear una partición extendida dentro de Disco2 de 200 kb
#Debería mostrar error ya que ya existe una partición extendida
#dentro de Disco2
fdisk -type=E -path=/home/Disco2.disk -name=Part3 -Unit=K
-size=200
#Elimina de forma rápida una partición llamada Particion1
fdisk -delete=fast -name="Particion1" -path=/home/Disco1.dsk
#Elimina de forma completa una partición llamada Particion1
fdisk -name=Particion1 -delete=full -path=/home/Disco1.disk
#Quitan 500 Kb de Particion4 en Disco4.dsk
#Ignora los demás parametros (size, delete)
#Se toma como válido el primero que aparezca, en este caso add
fdisk -add=-500 -size=10 -unit=K
-path="/home/misdiscos/Disco4.disk" -name="Particion4"
#Agrega 1 Mb a la partición Particion4 del Disco4.dsk
#Se debe validar que haya espacio libre después de la partición
fdisk -add=1 -unit=M -path="/home/mis discos/Disco4.disk"
```



```
-name="Particion 4"
```

4.MOUNT

Este comando montará una partición del disco en el sistema. Cada partición se identificará por un id que tendrá la siguiente estructura: VD + LETRA + NÚMERO. Por ejemplo: vda1, vda2, vdb1, vdc1... La letra será la misma para particiones en el mismo disco y el número diferente para particiones en el mismo disco. (NOTA: Este Comando Debe Realizar el montaje en memoria ram no debe escribir esto en el disco) Los parámetros admitidos por este comando son:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Este parámetro será la ruta en la que se encuentra el disco que se montará en el sistema. Este archivo ya debe existir.
-name	Obligatorio	Indica el nombre de la partición a cargar. Si no existe debe mostrar error.

Ejemplo:

```
#Monta las particiones de
Disco1.dsk

mount -path=/home/Disco1.disk -name=Part1 #id=vda1
mount -path=/home/Disco2.disk -name=Part1 #id=vdb1
mount -path=/home/Disco3.disk -name=Part2 #id=vdc1
mount -path=/home/Disco1.disk -name=Part2
#id=vda2
```

5. UNMOUNT

Desmonta una partición del sistema. Se utilizará el id que se le asignó a la partición al momento de cargarla. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición que se desmontará. Si no existe el id deberá mostrar un error.

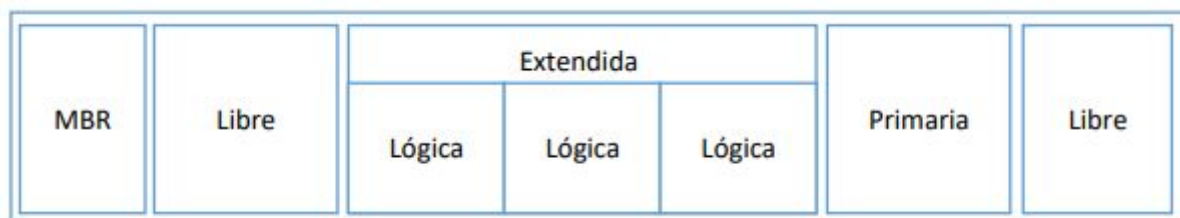
Ejemplos:

```
#Desmonta la partición con id vda1 (En Disco1.dsk)
umount -id=vda1
#Si no existe, se debe mostrar error
umount -id=vdx1
```

6. Discos

Los discos serán archivos binarios que tendrán información del MBR, y espacio con particiones o bien, espacio sin utilizar.

La siguiente figura es un ejemplo de los bloques en un disco con particiones en el que ya se ha eliminado una partición:



7. Estructuras

7.1 Master Boot Record

Cuando se crea una partición debe utilizarse el primer ajuste para crearla. En la figura anterior debería utilizarse el primer bloque libre para crear una nueva partición. El MBR tendrá los siguientes campos:

Nombre	Tipo	Descripción
mbr_tamano	int	Tamaño total del disco en bytes
mbr_fecha_creacion	time	Fecha y hora de creación del disco

mbr_disk_signature	int	Número random, que identifica de forma única a cada disco
disk_fit	char	Tipo de ajuste de la partición. Tendrá los valores B (Best), F (First) o W (worst)
mbr_partition_1	partition	Estructura con información de la partición 1
mbr_partition_2	partition	Estructura con información de la partición 2
mbr_partition_3	partition	Estructura con información de la partición 3
mbr_partition_4	partition	Estructura con información de la partición 4

7.2 Partition

Nombre	Tipo	Descripción
part_status	char	Indica si la partición está activa o no
part_type	char	Indica el tipo de partición, primaria o extendida. Tendrá los valores P o E
part_fit	char	Tipo de ajuste de la partición. Tendrá los valores B (Best), F (First) o W (worst)
part_start	int	Indica en que byte del disco inicia la partición
part_size	int	Contiene el tamaño total de la partición en bytes
part_name	char[16]	Nombre de la partición

7.3 Extended Boot Record

Las particiones extendidas tendrán una estructura diferente. Se utilizará una estructura llamada EBR (Extended Boot Record) en forma de lista enlazada, que será como la siguiente:

Nombre	Tipo	Descripción
part_status	char	Indica si la partición está activa o no
part_fit	char	Tipo de ajuste de la partición. Tendrá los valores B (Best), F (First) o W (worst)
part_start	int	Indica en qué byte del disco inicia la partición
part_size	int	Contiene el tamaño total de la partición en bytes.
part_next	int	Byte en el que está el próximo EBR. -1 si no hay siguiente
part_name	char[16]	Nombre de la partición

La estructura lógica de la partición extendida será como la siguiente:



El EBR inicial siempre debe existir, aunque se elimine la primera partición. Para crear el archivo del disco se recomienda utilizar un char[1024] como buffer para crear el archivo, si se utiliza un char[1] normalmente se tarda demasiado al momento de crear el archivo.

8. REPORTE

Se deberán generar los reportes con el comando rep. Se generarán en graphviz. Se puede utilizar html dentro de los reportes si el estudiante lo considera necesario. Deberá mostrarlos de forma similar a los ejemplos mostrados.

IMPORTANTE: Esta parte es **obligatoria** para tener derecho a la calificación de los aspectos que muestre el reporte. Si falta alguno de los reportes no se calificará.

8.1 REP

Recibirá el nombre del reporte que se desea y lo generará con graphviz en una carpeta existente.

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-name	Obligatorio	Nombre del reporte a generar. Tendrá los siguientes valores: -mbr -disk Si recibe otro valor que no sea alguno de los anteriores, debe mostrar un error.
-path	Obligatorio	Indica una carpeta y el nombre que tendrá el reporte. Si no existe la carpeta, deberá crearla.
-id	Obligatorio	Indica el id de la partición que se utilizará. Si el reporte es sobre la información del disco, se utilizará el disco al que pertenece la partición. Si no existe debe mostrar un error.

Ejemplo:

```
rep -id=vda1 -Path=/home/user/reports/reportel.jpg -name=mbr
```

```
rep -id=vda2 -Path=/home/user/reports/reportel2.pdf -name=disk
```

8.2 REPORTE MBR

Mostrará tablas con toda la información del MBR, así como de los EBR que se pudieron haber creado.

Ejemplo:

MBR Disco1.dsk

Nombre	Valor
mbr_tamaño	10485760
mbr_fecha_creacion	30/11/2015 13:45
mbr_disk_signature	16684811
Disk_fit	W
part_status_1	1
part_type_1	P
part_fit_1	B
part_start_1	300
part_size_1	1048576
part_name_1	Particion 1
part_status_2	1
part_type_2	E
part_fit_2	W
part_start_2	1048876
part_size_2	1048576
part_name_2	Particion 2

EBR_1

Nombre	Valor
part_status_1	1
part_fit_1	F
part_start_1	1048876
part_size_1	524438
part_next_1	1573314
part_name_1	Logica 1

EBR_2

Nombre	Valor
part_status_1	1
part_fit_1	B
part_start_1	1573514
part_size_1	524438
part_next_1	-1
part_name_1	Logica 2

8.2 REPORTE DISK

Este reporte mostrará la estructura de las particiones, el mbr del disco y el porcentaje que cada partición o espacio libre tiene dentro del disco (La sumatoria de los porcentajes debe de ser 100%).

Disco1.dsk

MBR	Libre 25% del disco	Extendida					Primaria 20% del disco	Libre 15% del disco
		EBR	Lógica 10% del Disco	Libre 10% del Disco	EBR	Lógica 10% del Disco		

9. SCRIPT

Son archivos con los comandos definidos en este documento. También puede haber comentarios y líneas en blanco. Tendrán la extensión sh y se utilizarán para que los ejecute el comando exec.

9.1. EXEC

El programa podrá ejecutar scripts con el comando exec. Debe mostrar el contenido de la línea que está leyendo y su resultado. También debe mostrar los comentarios del script.

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Especifica el nombre del script que se va a ejecutar.

Ejemplo:

```
#ejecuta el script  
exec -path=/home/Desktop/calificacion.sh
```

10. Manuales

Para comprender el proyecto, se deberá documentar el proyecto con dos manuales y documentar cada método del código con una pequeña descripción sobre lo que realiza y cómo lo realiza.

10.1 Manual de usuario

Deberá explicar para qué sirven los comandos del sistema y mostrar screenshots de cómo funciona cada comando. También deberá mostrar cómo generar reportes y que significa el contenido que tienen los reportes.

10.2 Manual técnico

Deberán crearse diagramas de flujo para los comandos de la aplicación, mostrando el algoritmo utilizado para ejecutar la operación y explicando dicho algoritmo.

Entrega

El proyecto se entregará el día **Miércoles 09 de junio del 2020 hasta las 23:59** se habilitará una opción en el classroom para que puedan subir la primera fase del proyecto. La impuntualidad anulara el trabajo entregado. Se debe informar si alguien no puede asistir a la calificación.

Se deberán entregar con el nombre `Carnet.tar.gz` o `.zip`. Dentro del comprimido pondrán todo el código fuente que utilizaron, sus manuales y el ejecutable del cual se calificara.

Se deberá entregar un ejecutable funcional que se utilizará para la calificación.

1. Calificación: Fechas por definir.
2. Para tener derecho a calificación se deberá contar con **requisitos mínimos** los cuales son:
 - Creación de Particiones Primarias y Extendidas
 - Aplicar Ajustes
 - Manuales
 - Reportes
4. El proyecto debe realizarse de forma individual, copias tendrán una nota de 0 y serán reportadas a la escuela.
5. Las dudas se responderán en UEDI y aplicarán a todos los estudiantes.
6. El lenguaje por utilizar es C++ en el IDE QT. No se permite el uso de otro lenguaje.
7. Solo se calificará sobre una instalación física de una distribución GNU/Linux.
8. NO se permite la modificación de código durante la calificación. El estudiante no tendrá acceso al código fuente durante la calificación.
9. El archivo binario que representa a los discos no debe crecer de lo contrario tendrá una penalización del 20%.
10. No se permite la utilización de estructuras en memoria (listas, arboles, etc.) para el manejo de los archivos o carpetas.

11. No se permite agregar o quitar atributos a los structs que se utilizarán en el proyecto.
12. Para Interpretar el Código el Uso de la Herramienta Bison es obligatorio.