

MANUAL DE USUARIO

PROYECTO 1

Nombre:	Victor Alfonso López Morales
Carnet:	2011-13915
Curso:	Manejo e implementación de archivos
Sección:	A

INTRODUCCIÓN

Proyecto 1 es una aplicación de consola que permite gestionar la creación de discos, particiones y formateo con el sistema de archivos LWH v 2.0. También permite generar reportes para visualizar el estado de sus particiones y sistema de archivos.

REQUISITOS

- Tener un sistema con una distribución de Linux basada en debian.
- Tener un visor de documentos PDF.
- Se utilizo QT Creator para la codificación junto a las herramientas Flex y Bison.

Comandos

La aplicación es totalmente en consola, a excepción de los reportes en Graphviz. La aplicación consta únicamente de una consola en la cual el usuario podrá ingresar comando que le servirán para la realización de las diferentes acciones posibles. Algunas normas que debe cumplir se presentan a continuación, no distinguirá entre **mayúsculas** y **minúsculas**. Hay parámetros obligatorios y opcionales. Solo se puede colocar un comando por línea.

Si se utiliza un parámetro no permitido, mostrara un mensaje de error. Se utilizarán espacios en blanco para separar cada parámetro. Los parámetros pueden venir en cualquier orden.

Si un comando necesita más de una línea se debe utilizar `\n` al final de la línea para indicar que continúa en la siguiente línea. También se puede ejecutar archivos de scripts con estos comandos a través del comando `exec`, que se describe más adelante. Se podrá comentar cada comando Los comentarios de estos scripts empezarán con `#`, dichos comandos son únicamente de línea. Estos comandos se explicarán en detalle a continuación.

MKDISK

Parametros

- **Size:** 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.
- **Path:** Este parámetro será la ruta en el que se creará el archivo que representará el disco duro.
NOTA: La dirección de la carpeta únicamente llevará comillas dobles (") cuando algún name dentro de la dirección contenga espacios, de lo contrario podría venir sin comillas.
- **Name:** Este parámetro será el name del disco con extensión dsk. Si no contiene la extensión dsk debe mostrar un mensaje de error.
- **Unit:** Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:
 - **k** que indicará que se utilizarán Kilobytes (1024 bytes)
 - **m** en el que se utilizarán Megabytes (1024 * 1024 bytes)

Ejemplo:

```
#Crea un disco de 3000 Kb en la carpeta home
Mkdisk &size->32 &path->/home/user/ \n
&name->Disco1.dsk %uniT&>k
#Se crearán carpetas si no existen
mkdisk &SiZe->8 &pAth->"/home/mis discos/DISCO Prueba/" \n
&namE->Disco_3.dsk
#Crearé un disco de 16 Mb ya que no hay parámetro unit
Mkdisk &size->16 &path->"/home/mis discos/" &NaMe->Disco4.dsk
```

RMDISK

Parametros

- **Path:** 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:

```
#Eliminar Disco_4.dsk
rmDisk &path&>"/home/mis discos/Disco_4.dsk"
```

FDISK

Parámetros

- **Size:** Este parámetro recibirá un número que indicará el tamaño de la partición a crear. Debe ser positivo y mayor a cero, si no mostrará un mensaje de error.
- **Unit:** Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:
 - **B:** que indicará que se utilizarán bytes
 - **K:** que indicará que se utilizarán Kilobytes (1024 bytes)
 - **M:** en el que se utilizarán Megabytes (1024 * 1024 bytes)
- **Path:** 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.
NOTA: La dirección de la carpeta únicamente llevará comillas dobles (") cuando algún name dentro de la dirección contenga espacios, de lo contrario podría venir sin comillas.
- **Type:** 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:** en este caso se creará una partición primaria.
 - **E:** en este caso se creará una partición extendida.
 - **L:** Con este valor se creará una partición lógica.
- **Fit:** Indicará el ajuste que utilizará la partición para asignar espacio. 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
- **Delete:** 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.
 - **Fast:** Esta opción marca como vacío el espacio en la tabla de particiones.
 - **Full:** Esta opción además marcar como vacío el espacio en la tabla de particiones, rellena el espacio con el carácter \0.
- **Name:** Indicará el name de la partición. El name 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.
- **Add:** Este parámetro se utilizará para agregar o quitar espacio de la partición. Puede ser positivo o negativo. Tomará el parámetro %UNIT para las unidades a agregar o eliminar. 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).

Ejemplo:

```
#Crear una Particion primaria llamada Particion1 de 72kb
#Con el peor ajuste y con asignacion Indexada en el Disco1.dsk
fdisk &size->72 &path->/home/Disco1.dsk &name->Particion1
#Crea una partición extendida dentro de Disco2 de 56 kb
#Tiene el peor ajuste y asignación Enlazada
fdisk %Type->E &path->/home/Disco2.dsk %Unit->K \^
&name->Particion2 &size->56 %Allocation->E
#Crea una partición lógica con el mejor ajuste, llamada Particion3,y
#de 1 Mb en el Disco3 y asignación contigua
fdisk &size->1 %tipo->L %unit->M %fit->BF \^
```

```
%Allocation->C &path->/mis discos/Disco3.dsk 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 %tipo->E &path->/home/Disco2.dsk &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.dsk"
#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.dsk"
\^
&name->"Particion 4"
```

MOUNT

Parametros

- **Path:** Este parámetro será la ruta en la que se encuentra el disco que se montará en el sistema. Este archivo ya debe existir.
- **Name:** Indica el name de la partición a cargar. Si no existe debe mostrar error.

Ejemplo:

```
#Monta las particiones de Disco1.dsk
mount &path->/home/Disco1.dsk &name->Part1 #id->vda1
mount &path->/home/Disco2.dsk &name->Part1 #id->vdb1
mount &path->/home/Disco3.dsk &name->Part2
#id->vdc1mount &path->/home/Disco1.dsk &name->Part2
#id->
mount &path->/home/Disco1.dsk &name->Part2 #id->vda2
```

UNMOUNT

Parametros

- **Idn:** Especifica una lista de id de las particiones que se Desmontará, Si no existe algun ID mostrara error.

Ejemplo:

```
#Desmonta la partición con id vda1 (En Disco1.dsk)
umount &id1->vda1
#Si no existe, se debe mostrar error
umount &id1->vdx1
#Desmonta una lista de particiones.
umount &id1->vda1 &id2->vdb2 &id3->vdc
```

MKFS

Parametros

- **Id:** Indicará el id que se generó con el comando mount de la fase anterior. Si no existe

mostrará error. Se utilizará para saber la partición y el disco que se utilizará para hacer el sistema de archivos.

- **Type:** Indicará que tipo de formato se realizará. Ya que es opcional, se tomará como un formato completo si no se especifica esta opción. Podrá tener los siguientes valores:
 - **Fast:** en este caso se realizará un formato rápido.
 - **Full:** en este caso se realizará un formato completo.

Ejemplo:

```
#Realiza un formateo rápido con LWH de la "Particion 1" del  
#Disco1.dsk
```

```
mkfs &id->vda1 &type->fast
```

```
#Realiza un formateo completo con LWH de Particion2 en  
Disco2.dsk
```

```
mkfs &id->vdb1
```

```
#Agrega 500 Kb a Particion3 en  
Disco3.dsk
```

```
mkfs &add->500 &id->vdc1
```

```
#Agrega 1 Mb a Particion3 en  
Disco3.dsk
```

```
Mkfs &add->1 &id->vdc1 &unit->m
```

LOGIN

Parametros

- **Usr:** Especifica el nombre del usuario que iniciará sesión. Si no se encuentra en la primera partición mostrará un mensaje indicando que el usuario no existe. En este caso si distinguirá mayúsculas de minúsculas.
- **Pwd:** Indicará la contraseña del usuario, si no coincide debe mostrar un mensaje de autenticación fallida. Distinguirá entre mayúsculas y minúsculas.
- **Id:** Indica el id de la partición a la cual se desea obtener acces Únicamente se puede obtener acceso a las particiones que se encuentren montadas en ese momento. De lo contrario la aplicación debe mostrar un mensaje de error indicando que la partición no se encuentra montada.

Ejemplo:

```
#Se loguea en el sistema como usuario root  
login &usr->root &pwd->201020576 &id->vda1
```

```
login &usr->"mi usuario" &pwd->"mi pwd" /  
&id->vda2
```

LOGOUT

Parametros

No cuenta cuenta con parametros.

Ejemplo:

```
Logout
```

MKGROUP

Parametros

- **Id:** Especifica el id de la partición en la que se creará el grupo. Si no existe, debe mostrar un error.
- **Name:** Indicará el nombre que tendrá el grupo

Ejemplo:

```
#Crea el grupo usuarios en la partición  
vda1  
mkgrp &id->vda1 &nombre-  
>"usuarios"  
  
#Debe mostrar mensaje de error ya que el grupo ya  
existe  
mkgrp &id->vda1 &nombre->"usuarios"
```

RMGROUP

Parametros

- **Id:** Especifica el id de la partición en la que se creará el grupo. Si no existe, debe mostrar un error.
- **Name:** Indicará el nombre que tendrá el grupo

Ejemplo:

```
#Elimina el grupo de usuarios en la partición vda1  
rmgrp &id->vda1 &nombre->"usuarios"  
  
#Debe mostrar mensaje de error ya que el grupo no existe porque ya fue eliminado  
rmgrp &id->vda1 &nombre->"usuarios"
```

MKUSR

Parametros

- **Id:** Indicará la contraseña del usuario, si no coincide debe mostrar un mensaje de autenticación fallida. Distinguirá entre mayúsculas y minúsculas.
- **Usr:** Indicará el nombre del usuario a crear, si ya existe, deberá mostrar un error indicando que ya existe el usuario. Máximo: 10 caracteres.
- **Pwd:** Indicará la contraseña del usuario. Máximo: 10 caracteres.
- **Grp:** Indicará el grupo al que pertenece el usuario. Debe de existir en la partición en la que se está creando el usuario, si no debe mostrar un mensaje de error. Máximo: 10 caracteres.

Ejemplo:

```
#Crea el grupo usuarios en la partición vda1
```



```
Mkusr &id->vda1 &nombre->"user1" -grp->usuarios -pwd->usuario
```

RMUSR

Parametros

- **Id:** Especifica el id de la partición en la que se eliminará el usuario. Si no existe, debe mostrar un error.
- **Usr:** Indicará el nombre del usuario a eliminar, si no existe, deberá mostrar un error indicando que el usuario no existe.

Ejemplo:

```
#Elimina el usuario user1 de la partición  
rmusr &id->vda1 &usr->user
```

CHMOD

Parametros

- **Id:** Especifica el id de la partición en la que se encuentra el archivo o carpeta a la que se le cambiarán los permisos.
- **Path:** Este parámetro será la ruta en la que se encuentra el archivo O carpeta a la que se le cambiarán los permisos.
- **UGO:** Indica los permisos que tendrán los usuarios. Serán tres números, uno para el Usuario, el siguiente para el Grupo al que pertenece el usuario y el último para Otros usuarios fuera del grupo. Cada número tendrá los valores desde el 0 al 7.
- **R:** Indica que el cambio será recursivo en el caso de carpetas. El cambio afectará a todos los archivos y carpetas en la que la ruta contenga la carpeta especificada por el parámetro &path y que sean propiedad del usuario actual

Ejemplo:

```
#Cambia los permisos de la carpeta home recursivamente  
#Todos los archivos o carpetas que tengan /home cambiarán  
#Por ejemplo si existiera /home/user/docs/a.txt  
#Cambiaría los permisos de las tres carpetas y del  
archivo  
chmod &id->vda1 &path->"/home" +R &ugo->764  
#Cambia los permisos de la carpeta home  
#Se debe comprobar que la carpeta home pertenezca al usuario #actual, si no deberá mostrar  
un mensaje de error.  
chmod &id->vda1 &path->"/home" &ugo->777
```

MKFILE

Parametros

- **Id:** Especifica el id de la partición en la que se creará el archivo. Si no existe debe mostrar error.
- **Path:** Este parámetro será la ruta del archivo que se creará. Si lleva espacios en blanco deberá encerrarse entre comillas. Si ya existe debe sobrescribir el archivo. Si no existen las carpetas padres, debe mostrar error, a menos que se utilice el parámetro -p, que se explica posteriormente.
- **P:** Si se utiliza este parámetro y las carpetas especificadas por el parámetro -path no existen, entonces deben crearse las carpetas padres.
- **Size:** Este parámetro indicará el tamaño en bytes del archivo, el contenido será el abecedario cuantas veces sea necesario.

- **Cont:** Indicará la Cadena que sera Escrita dentro del Archivo.

Ejemplo:

```
#Crea el archivo a.txt
#Si no existen las carpetas home user o docs se crean
#El tamaño del archivo es de 15 bytes
#El contenido sería: 012345678901234
mkFile %SIZE->15 &id->vdb1 &PatH->"/home/user/docs/a.txt" %p

#Crea "archivo 1.txt" la carpeta "mis documentos" ya debe existir
#el tamaño es de 0 bytes
mkfile &id->vda1 &path->"/home/mis documentos/archivo 1.txt"

#Crea el archivo b.txt
#El contenido del archivo será el mismo que el archivo b.txt
#que se encuentra en el disco duro de la
computadora. mkfile &id->vda1 &path-
>"/home/user/docs/b.txt" %p \^
%cont->"/home/Documents/b.txt"
```

CAT

Parametros

- **Id:** Especifica el id de la partición en la que se leerá el archivo. Si no existe debe mostrar error.
- **Filen:** Permitirá Admitir como argumentos una lista de n ficheros que hay que enlazar. Estos se encadenarán en el mismo orden en el cual fueron especificados. Si no existe el archivo o no tiene permiso de lectura, debe mostrarse un mensaje de error.

Ejemplo:

```
#Lee el archivo a.txt En la terminal debería mostrar el contenido, en #este ejemplo
01234567890123
Cat &file1->"/home/user/docs/a.txt" &id->vdb1

#enlazara los archivos a.txt (datos archivo a) b.txt (01234567890123) #c.txt (0123) y debería
mostrar el contenido siguiente, cada archivo #va separado por salto de línea
# datos archivo a
# 01234567890123
# 0123
Cat &file1->"/home/a.txt" &id->vdb1 &file2->"/home/b.txt" \ &file3->"/home/c.txt"
```

RM

Parametros

- **Id:** Especifica el id de la partición en la que se eliminará el archivo
- **Path:** Este parámetro será la ruta del archivo o carpeta que se eliminará. Deberá encerrarse entre comillas.
- **RF:** Permite hacer borrados recursivos. Con esta opción se pueden borrar sin importar los permisos siempre y cuando sea el usuario root.

Ejemplo:

```
#Elimina el archivo a.txt, b.txt muestra error si no
tiene permiso
rm &PatH->"/home/user/docs/a.txt" &Id->vdb1
rm &PatH->"/home/user/docs/b.txt" &Id->vdb1
#Error por permisos
#Elimina la carpeta user y todo su
contenido (docs, a.txt)
#Si el usuario no tuviera permiso de
escritura sobre b.txt
#No debería eliminar las carpetas
padre docs ni user, solo a.txt
rm &PatH->"/home/user" &Id->vdb1
#Elimina la carpeta user y todo su contenido (docs, a.txt) sin importar los permisos(logueado
como usuario root)
Rm &Id->vdb1 &path->"/home/user" +RF
```

EDIT

Parametros

- **Id:** Especifica el id de la partición en la que se modificará el archivo. Si no existe debe mostrar error.
- **Path:** Este parámetro será la ruta del archivo que se modificará. Deberá encerrarse entre comillas. Si no existe, debe mostrar un Mensaje de error.
- **Size:** Este parámetro indicará el nuevo tamaño en bytes del a archivo, el contenido serán el abecedario cuantas veces sea necesario. Si es negativo debe mos trar error.
- **Cont:** Indicará una cadena la cual sera el nuevo contenido del Archivo.

Ejemplo:

```
#Modifica el archivo a.txt
#El tamaño del archivo es de 22 bytes
#El contenido sería: abcdefghijklmnopqrstu
Edit %SIZE->22 &id->vdb1 &PatH->"/home/user/docs/a.txt"
#Modifica el archivo b.txt
#El contenido del archivo será el mismo que el archivo c.txt
#que se encuentra en el disco duro de la computadora.
edit &id->vda1 &path->"/home/user/docs/b.txt" %cont ->"/home/Documents/c.txt"
#Modifica nuevamente el archivo b.txt
edit &id->vda1 &path->"/home/user/docs/b.txt" &cont ->"/home/Documents/d.txt"
```

REN

Parametros

- **Id:** Especifica el id de la partición en la que se renombrará el archivo o carpeta. Si no existe debe mostrar error.
- **Path:** Este parámetro será la ruta del archivo o carpeta al que se le cambiará el nombre. Deberá encerrarse entre comillas. Si no existe el archivo o carpeta o no tiene permisos de escritura sobre la carpeta o archivo, debe mostrarse un mensaje de error.
- **Name:** Especificará el nuevo nombre del archivo, debe verificar que no exista un archivo

con el mismo nombre, de ser así debe mostrar un mensaje de error. Deberá encerrarse entre comillas.

Ejemplo:

```
#Cambia el nombre del archivo a.txt a b1.txt
ren &PatH->"/home/user/docs/a.txt" &Id->vdb1 &nombre->"b1.txt"
#Debera mostrar error ya que el archivo b1.txt ya existe
ren &PatH->"/home/user/docs/c.txt" &Id->vdb1 &nombre->"b1.txt"
```

MKDIR

Parametros

- **Id:** Especifica el id de la partición en la que se creará la carpeta. Si No existe debe mostrar error.
- **Path:** Este parámetro será la ruta de la carpeta. Si no existen las carpetas padres, debe mostrar error, a menos que se utilice el parámetro +p, que se explica posteriormente.
- **P:** Si se utiliza este parámetro y las carpetas padres en el Parámetro path no existen, entonces deben crearse.

Ejemplo:

```
#Crea la carpeta usac
#Si no existen las carpetas home user o docs se crean
Mkdir %P &id->vda1 &path->"/home/user/docs/usac"

#Crea la carpeta "archivos 2016"
#La carpeta padre ya debe existir
Mkdir &ID->vda1 &path->"/home/mis documentos/archivos 2016"
```

CP

Parametros

- **Id:** Especifica el id de la partición en la que está el archivo o Carpeta que se quiere copiar. Si no existe, debe mostrar un error.
- **Path:** Este parámetro será la ruta del archivo o carpeta que se desea copiar.
- **Destiny:** Este parámetro será la ruta de carpeta a la que se copiara el archivo o carpeta. Debe tener permiso de escritura sobre la carpeta. Debe mostrar un mensaje de error si no tiene permiso de escritura o si la carpeta no existe.

Ejemplo:

```
#!/
#|_home      #664
# |_user      #664
# | |_documents #664
# | |_a.txt    #664
# | |_b.txt    #224
# |_images    #664
#Copia documents a images
cp &id->vda2 &Path->"/home/user/documents" &iddestiny->vda2 &dest->"/home/images"
```

MV

Parametros

- **Id:** Especifica el id de la partición en la que está el archivo o Carpeta que se quiere mover. Si no existe, debe mostrar un error.
- **Path:** Este parámetro será la ruta del archivo o carpeta que se desea mover.
- **Destiny:** Este parámetro será la ruta de carpeta a la que se moverá el archivo o carpeta. Debe tener permiso de escritura sobre la carpeta. Debe mostrar un mensaje de error si no tiene permiso de escritura o si la carpeta no existe.

Ejemplo:

```
#/  
#|_home      #664  
# |_user      #664  
# | |_documents #664  
# | |_a.txt    #664  
# | |_b.txt    #224  
# |_images    #664  
  
#Mueve documents a images  
mv &id->vda1 &iddestiny->vda1 &Path->"/home/user/documents" \^  
&destiny->"/home/images"
```

FIND

Parametros

- **Id:** Especifica el id de la partición en la que se buscará el archivo O carpeta. Si no existe, debe mostrar un error.
- **Path:** Este parámetro será la ruta de la carpeta en el que se inicia la búsqueda, deberá buscar en todo su contenido. Debe tener permisos de lectura en los archivos que buscará.
- **Name:** Indica el nombre del archivo o carpeta que se está buscando. Debe aceptar los caracteres especiales definidos anteriormente

Ejemplo:

```
#Arbol actual  
#/  
#|_home      #664  
# |_user      #664  
# | |_a.txt    #664  
# | |_b.txt    #420  
# |_images    #664  
#Busca los archivos que tengan una letra como nombre  
#y cualquier extensión  
find &id->vda1 &Path->"/home" &nombre->"?.*"
```

CHOWN

Parametros

- **Id:** Especifica el id de la partición en la que se encuentra el archivo o carpeta a la que se le cambiará el propietario.

- **Path:** Este parámetro será la ruta en la que se encuentra el archivo o carpeta a la que se le cambiará el propietario. Si no existe la ruta deberá mostrar mensaje de error.
- **R:** Indica que el cambio será recursivo en el caso de carpetas. El cambio afectará a todos los archivos y carpetas en la que la ruta contenga la carpeta especificada por el parámetro &path.
- **Usr:** Nombre del nuevo propietario del archivo o carpeta. Si no Existe o está eliminado debe mostrar error.

Ejemplo:

```
#Cambia el propietario de la carpeta home recursivamente
chown &id->vda1 &path->"/home" +R &usr->user2

#Cambia los permisos de la carpeta home
chown &id->vda1 &path->"/home" &usr->user1
```

CHGRP

Parametros

- **Usr:** Especifica el nombre del usuario al que se le cambiará de grupo. Si no existe debe mostrar un error.
- **Grp:** Contendrá el nombre del nuevo grupo al que pertenece El usuario. Si no existe o está eliminado debe mostrar un error.

Ejemplo:

```
#Cambia el grupo del user2
chgrp &usr->user2 &grp->grupo1

#Cambia el grupo del user1
chgrp &usr->user1 &grp->grupo2
```

LOSS

Parametros

- **Id:** Especifica el id de la partición en la que se desea simular la perdida.

Ejemplo:

```
#Simulando la perdida del sistema de archivos de la partición1
Loss &id->vda1
```

RECOVERY

Parametros

- **Id:** Especifica el id de la partición en la que se desea recuperar la partición que simulo la perdida la perdida.

Ejemplo:

```
#Recuperando el sistema de archivos EXT3 de la partición 1
Recovery &id->vda1
```

REP

Parametros

- **Name:** Nombre del reporte a generar. Tendra los siguientes valores:
 - MBR

- DISK
 - INODE
 - SB
 - BM_ARBDIR
 - BM_DETDIR
 - BM_INODE
 - BM_BLOCK
 - BITACORA
 - DIRECTORIO
 - TREE_FILE
 - TREE_DIRECTORIO
 - TREE_COMPLETE
 - LS
- **Path:** Indicará una carpeta y el nombre que tendrá el reporte. Si no existe la carpeta, deberá crearla. Si lleva espacios se encerrará entre comillas.
 - **Id:** Indica el id de la partición a que se utilizará.
 - **Ruta:** Funcionará para el reporte file y ls. Será el nombre de directorio o archivo del que se mostrará el reporte. Si no existe muestra error.

Ejemplo:

```
rep &id->vda2 &Path="/home/user/reports/reporte 2.pdf" &nombre=bm_arbdir -ruta="/home/mis documentos"
rep &id->vda1 &Path="/home/user/reports/reporte 3.jpg" &nombre=archive_directorio
```

EXEC

Parámetros

- **Path:** Especifica el nombre del script que se va a ejecutar.

Ejemplo:

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

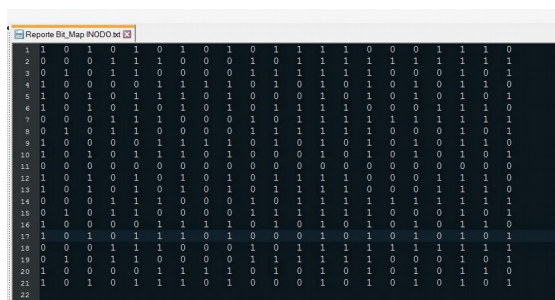
PAUSE

Este comando será solo la palabra “pause” no tiene atributos al ingresar este comando se pondrá en pausa solicitando que apasche cualquier tecla para continuar. Este comando NO detiene la ejecución de un archivo solo queda a la espera de presionar una tecla para continuar su ejecución.

Reportes

REPORTES DE BITMAP

Los reportes se generarán en un archivo de texto, el formato para este reporte es criterio del estudiante, el formato SUGERIDO: "0,0,1,0,1,0" o "0|0|1|0|1|0" Con un máximo de 20 bit de información por línea a generada.



REPORTE DE BITACORA

Se debe escribir en un archivo de texto toda la información que esta almacenada en la bitácora, el formato SUGERIDO:

Tipo Operación: "El tipo de operación a realizarse" Tipo:

"Archivo/Directorio"

Nombre: "Nombre archivo o directorio"

Contenido: "Si hay datos contenidos" Fecha

"Fecha de transacción"

Tipo Operación: "El tipo de operación a realizarse" Tipo:

"Archivo/Directorio"

Nombre: "Nombre archivo o directorio"

Contenido: "Si hay datos contenidos"

Fecha"Fecha de transacción"

Tipo Operación: "El tipo de operación a realizarse" Tipo:

"Archivo/Directorio"

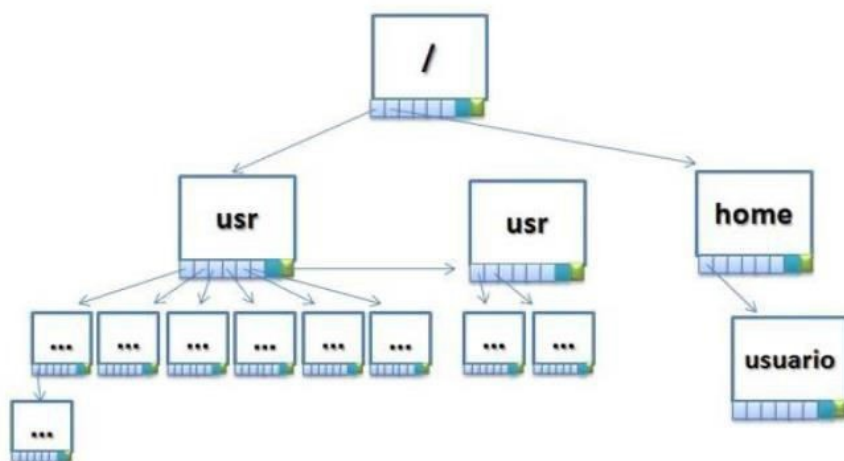
Nombre: "Nombre archivo o directorio"

Contenido: "Si hay datos contenidos" Fecha

"Fecha de transacción"

REPORTES DE DIRECTORIO

Para verificar si la estructura lógica del sistema es la correcta, se debe de realizar una gráfica en Graphviz mostrando todos los directorios y subdirectorios mostrando su orden jerárquico similar a la siguiente:

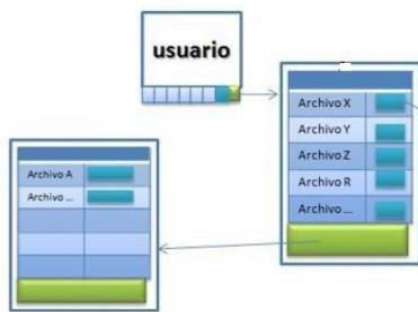


REPORTES DE TREE FILE

Se debe de realizar una gráfica en Graphviz mostrando la ubicación de un archivo (carpeta/s) y su contenido (carácter/es). Se debe de elegir el archivo deseado por medio de un menú desde consola y la gráfica debe contener una estructura similar a la siguiente:[arreglo](Solo incluir

Se debe mostrar un árbol completo con todas las carpetas que se encuentran en el sistema de

Se deben mostrar en graphviz los archivos que contiene un directorio. El directorio



REPORTES DE DISK

Este reporte mostrará la estructura de las particiones (mostrar nombre de la partición y el tipo) y el mbr del disco.

Disco1.dsk

MBR	Libre	Extendida					Primaria	Libre
		EBR	Lógica	Libre	EBR	Lógica		

REPORTES DE LS

Este reporte mostrará la información de los archivos y carpetas con permisos, fecha de modificación, tipo, fecha de creación, reporte realizado con Consola

-rw-rw-r--	user1	Mi grupo	40661	24/02/2016	09:53	Manual.txt
-rw-r--rwx	user2	Mi grupo	123	24/02/2016	09:53	archivo1.txt
drw-rw-r--	root	root	0	24/02/2016	09:53	home

REPORTES DE SB

Muestra toda la información del superbloque en una tabla.

NOMBRE	VALOR
sb_nombre_hd	Disk1
sb_arbol_virtual_count	78
sb_detalle_directorio_count	128
sb_inodos_count	64
sb_bloques_count	50
sb_arbol_virtual_free	180
sb_detalle_directorio_free	128
sb_inodos_free	256
sb_bloques_free	128
sb_date_creacion	01/12/2016 15:38
sb_date_ultimo_montaje	30/12/2016 15:38
sb_montajes_count	52
sb_ap_bitmap_arbol_directorio	100
sb_ap_arbol_directorio	56
sb_ap_bitmap_detalle_directorio	98
sb_ap_detalle_directorio	5631
sb_ap_bitmap_tabla_inodo	200
sb_ap_tabla_inodo	600
sb_ap_bitmap_bloques	10
sb_ap_bloques	100
sb_ap_log	200
sb_size_struct_arbol_directorio	128
sb_size_struct_detalle_directorio	64
sb_size_struct_inodo	50
sb_size_struct_bloque	180
sb_first_free_bit_arbol_directorio	128
sb_first_free_bit_detalle_directorio	328
sb_first_free_bit_tabla_inodo	630
sb_first_free_bit_bloques	20000
sb_magic_num	201020576