

PRÁCTICAS DE SISTEMAS OPERATIVOS

Para realizar las prácticas, necesitaremos ser usuario root, para ello usaremos Fedora.

Actividad 1.1. Repaso de scripts de bash

Crea un script de bash que automatice todos los pasos vistos en este punto y que guardarás preferiblemente en tu directorio home. Al entrar de nuevo en el sistema sólo tendrás que ejecutar el script para empezar a trabajar en modo root.

```
#!/bin/bash
cp /home/albduranlopez/Descargas/*.gz /tmp
cd /tmp
gunzip *.gz
chmod 766 kernel32-3.0.4
./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024mcd
```

Una vez tengamos acceso root, podemos:

Crear usuario: useradd Alberto

Se creará una entrada en cada uno de los siguientes directorios:

/etc/passwd : Almacena información de las cuentas de usuarios.

/etc/shadow : Guarda los password encriptados.

/etc/group : Definición de los grupos y usuarios miembros.

usermod	Modifica una cuenta de usuario ya existente
userdel	Elimina una cuenta de usuario (no se borra del /home)
newusers	Crea cuentas de usuarios con información de un archivo de texto (con el formato de /etc/passwd)

USERADD:

-b	Directorio base
-c	Descripción del login
-e	El día que la cuenta estará inactiva Año-mes-día
-f	El número de días que después de que la pass expira hasta que la cuenta esté inactiva permanentemente
-U	Crear un grupo con el mismo nombre que el usuario y meterlo
-g	Asignarle un grupo ya creado grupo (groupadd)
-m	Crea el directorio home si no existe

Cambiar contraseña a un usuario: passwd user

1)Un bash shell restringido (bash -r) permite limitar algunas acciones del usuario que los usa, por ejemplo, no permite cambiar de directorio, ni utilizar nombres de archivos que contienen una barra /, ni cambiar las variables PATH y SHELL, ni redirigir las entradas/salidas, ni tampoco definir funciones. Se pide crear un usuario, denominado *invitado*, que debe usar un shell restringido, tiene un UID con valor 590 y pertenecerá a un grupo denominado *limitados* con GID 1010.

```
groupadd limitados --gid 1010
adduser invitado -shell="/bin/bash -r" --uid 590 -gid limitados
```

GESTIÓN DE GRUPOS

groupadd grupo	Crea un nuevo grupo
groupmod grupo	Modifica un grupo existente
groupdel grupo	Elimina un grupo
newgrp grupo	Cambia de grupo activo(lanza un shell con ese grupo)
gpasswd grupo	Asigna una contraseña a un grupo
gpasswd -a user grupo	Añade un usuario a un grupo
groups user	Informa de los grupos a los que pertenece un usuario
id user	Lista el identificador del usuario y los grupos a los que pertenece
grpck	Comprueba la consistencia del archivo de grupos

Saber directorio home del usuario: echo \$HOME

Saber si un usuario dispone de una cuenta en el sistema:

```
#!/bin/bash

cat /etc/passwd | cut -d ":" -f 1 | grep $1 &> /dev/null
var="$?"

if [ "$var" == "0" ];
then
    echo "El usuario $1 tiene cuenta en el sistema"
else
    echo "El usuario $1 NO tiene cuenta en el sistema"
fi
```

Actividad 1.2. Valores por omisión para nuevas cuentas

Visualiza el contenido de los dos archivos anteriores y comprueba cuáles son las opciones por defecto que tendría un usuario que se creara en nuestro sistema. A continuación, crea una cuenta de usuario y visualiza el contenido de los archivos /etc/passwd y /etc/group, y el directorio /home para comprobar que los nuevos datos se han rellenado conforme a la especificación tomada de /etc/default/useradd y /etc/login.defs.

```
#!/bin/bash

cat /etc/default/useradd
cat /etc/login.defs

echo "Introduce nombre de usuario a crear"
read var
sudo adduser $var

echo "El contenido de los directorios son: "
cat /etc/passwd
cat /etc/group
cat /home
```

Actividad 1.3. Creación de usuarios

- 1. Utiliza el manual en línea para leer la sintaxis completa de la utilidad para creación de cuentas y crea dos o tres usuarios en tu sistema cambiando alguno de los valores por defecto.**
- 2. Elimina alguno de ellos y comprueba que “rastro” ha dejado la cuenta recién eliminada en el sistema.**
- 3. Entra (orden su) en el sistema como uno de estos usuarios que has creado y mira qué archivos tiene en su directorio home. La orden sudo permite cambiar el modo de trabajo a modo root específicamente para ejecutar una orden con privilegios de supervisor y tras su ejecución continuar con los privilegios del usuario que abrió la sesión.**

```
# Antes de nada creamos el grupo "Alumno" y "Profesor"
# La opción -g sirve para indicar el grupo al que queremos que pertenezca
# La opción -d sirve para ubicar la carpeta
# La opción -m sirve para crear la carpeta home si no existe
# La opción -s sirve para indicar el intérprete de comandos

groupadd Alumno
groupadd Profesor

useradd -g Alumno -d /home/alberto -m -s /bin/bash alberto
useradd -g Profesor -d /home/jose -m -s /bin/sh jose

# borramos el usuario alberto
userdel alberto

# deja rastro en el directorio home "/home/alberto"

# Al entrar en el sistema como uno de estos usuarios encontramos la carpeta /home/"nombre-usuario" cuyo contenido es "examples.desktop"
# Para ver los archivos ocultos ejecutamos el siguiente comando:

ls -a /home/alberto
# Muestra . . . .bash_logout .bashrc examples.dektop .profile
```

Actividad 1.4. Archivo /etc/passwd

Visualiza el archivo /etc/passwd e indica cual es el formato de cada línea de dicho archivo. Para ello también puedes consultar el man o info de Linux. ¿Quién es el propietario de este archivo y cuáles son sus permisos?

```
cat /etc/passwd
Muestra la lista de usuarios junto con la descripción de sus características:
"albduranlopez:x:1000:1000:Alberto,,,:/home/albduranlopez:/bin/sh"
-El usuario es albduranlopez
-x indica que las passwords están encriptadas en el archivo /etc/shadow que solo tiene permisos root
-El nombre y Apellido están en Alberto,,
-/home/albduranlopez indica la carpeta home del usuario
-/bin/sh indica el intérprete de comandos
```

Actividad 1.5. Archivo /etc/shadow

Visualiza el archivo /etc/shadow desde un usuario distinto al root ¿Te da algún problema? ¿Sabes por qué? Intenta averiguarlo.

No tengo acceso al directorio /etc/shadow ya que es ahí donde se encuentran todas las contraseñas encriptadas y únicamente se puede acceder a ella con permisos root

Actividad 1.6. Creación de grupos

- 1. Crea un par de grupos y asignáelos a algunos de los usuarios de tu sistema.**
- 2. ¿Qué información devuelve la orden id si estás conectado como root?**

```
#Antes de nada creamos el grupo Profesor y grupo Alumno
groupadd Profesor
groupadd Alumno

#Creamos el usuario jose y alberto
useradd jose
useradd alberto

#Asignamos jose al grupo Profesor y alberto al grupo Alumno
gpasswd -a jose Profesor
gpasswd -a alberto Alumno

#Introducimos la orden id como root
#Muestra: uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

Actividad 1.7. Archivo del kernel de Linux

- **Utilizando la orden (find) que ya conoces para la búsqueda de archivos en el sistema de archivos, anota el nombre absoluto del archivo del kernel de Linux que se ha cargado en el sistema operativo que estás usando en el laboratorio de prácticas para acceso modo root.**

```
find / -name Fedora

#muestra:
/usr/share/icons/Fedora
```

Actividad 1.8. Organización del SA

Un programa que se ejecuta en modo root, ¿dónde podría guardar la información temporal de forma que ésta se mantuviese entre arranques del sistema?

#Se deberían guardar en el directorio /var/tmp ya que se espera que cambie durante el funcionamiento normal del sistema pero a la vez se tratan de archivos temporales

Actividad 1.9. Información de los SAs

Los archivos /etc/fstab y /etc/mtab muestran información sobre los sistemas de archivos que se encuentran montados en el sistema. ¿Cuál es la diferencia entre la información que muestra cada uno de ellos?

```
/etc/mtab muestra archivos del sistema
/etc/fstab se encarga de las particiones del sistema
```

Actividad 1.10 Información de los SAs

Edita el archivo /etc/fstab del sistema de archivos que estás utilizando en modo root y anota y describe la información que tiene registrada. Si no conoces alguna opción puedes consultar el manual en línea: man fstab.

```
#Siendo usuario root ejecutamos:
cat /etc/fstab
```

#Nos ayudamos de la orden "man fstab"

#El primer campo describe el dispositivo especial de bloque o sistema de ficheros remoto a ser montado "LABEL=ROOT"

#El segundo campo describe el punto de montaje para el archivo de sistema

#El tercer campo describe el tipo de archivo del sistema

#El cuarto campo describe las opciones de montaje asociadas con el archivo del sistema

#En el quinto campo se determina que archivos del sistema necesitan ser volcados

#En el sexto campo se determina el orden en el cual se van a chequear los archivos de sistema cuando este arranca

SESIÓN 2

Ejercicio inventado1:

Crear una partición en /dev/loop0 con 30MB con un SA ext3:

```
mknod /dev/loop0 b 7 0
dd if=/dev/zero of=/root/archivo_SA30 bs=3k count=10000
losetup /dev/loop0 /root/archivo_SA30
fdisk /dev/loop0 //n w p
fdisk -l /dev/loop0

#Para establecer un SA sobre una partición ya realizada, usaremos mke2fs:
mke2fs -t ext3 -L LABEL_ext3 /dev/loop0
#-t especifica el tipo del SA
#-L Establece la etiqueta
```

```
file -s /dev/loop0
#te dice de que tipo de partición es
```

Ejercicio inventado2:

Crear una partición /dev/loop1 con 20MB con un SA ext2:

```
mknod /dev/loop1 b 7 1
dd if=/dev/zero of=/root/archivo_SA20 bs=2k count=10000
losetup /dev/loop1 /root/archivo_SA20
fdisk /dev/loop1
fdisk -l /dev/loop1

#Para establecer un SA sobre una partición ya realizada, usaremos mke2fs:
mke2fs -t ext2 -L LABEL_ext2 /dev/loop1
#-t especifica el tipo del SA
#-L Establece la etiqueta
```

```
file -s /dev/loop1
```

tune2fs

-l /dev/loop0	Muestra información del SA
-c max_mount-counts /dev/loop0	Establece nº max de montajes
-L label /dev/loop0	Establece una etiqueta al SA
-C mount-count	Establece el nº de veces que se ha montado el sistema. Si es mayor que -c , e2fsck se iniciará en el prox inicio
-f	Fuerza a completar incluso si hay errores
-g	Establece el grupo que se usa para los bloques reservados
-j	Añade ext3 journal al sistema
-m reserved-block-percentage	Establece el porcentaje de bloques reservados
-r reserved-block-count	Establece el nº de bloques reservados
-u user	Establece el usuario que puede usar los bloques
-U	Establece el UUID
-T time-last-checked	Establece el tiempo en el que se checkeó el sistema usando e2fsck
-M last-mounted-directory	Establece el último directorio montado

Para mostrar la información de un sistema de archivos SA ya creado:

```
tune2fs -l /dev/loop0
```

A partir de este comando, podemos sacar cualquier campo, ejemplo:

Esto nos mostraría el nombre de la etiqueta del archivo montado:

```
tune2fs -l /dev/loop0 | grep "Filesystem volume name" | cut -d: -f2
```

Para saber cual es el sistema de archivos SA de una partición, ejecutaríamos:

```
file -s /dev/loop0 | cut -d " " -f5
```

Mount – Una vez creada la partición, con el SA, la etiqueta..etc, si queremos usar dicha partición para trabajar con ella, debemos montarla mediante la orden 'mount'

```
mount /dev/loop0 -opcion /mnt/SA_ext?
```

```
mkdir /mnt/SA_ext
```

-r	Con permiso de lectura
-o rw	Read/write
-L label	Monta la partición con la etiqueta label
-U uuid	Monta la partición con el uuid
-t system	Indica el tipo de sistema ex2, ext3, ext4..
-o async	E/S se sincroniza con los archivos del sistema
-o atime	El tiempo de acceso a los inodos se controla por el nucleo
-o noatime	No se actualizan los accesos a los inodos
-o defaults	Usa: rw, suid, dev, exec,auto, nouser, async
-o dev	Interpret character or block special devices on the filesystem.
-o diratime	Actualiza el directorio inodo
-o dircsync	Se actualizan los directorios con los archivos del sistema
-o exec	Permite ejecución de ficheros
-o group	Permite a un usuario montar si alguno de sus grupos coincide con el del disposit
-o relatime	Actualiza el acceso inodo para modificar la hora
-o suid	Permite establecer identificador de usuario o de grupo - bits
-o sync	Se sincroniza E/S con los archivos del sistema
user	Usuario que puede montar
umask=777	Establece todos los permisos
uid=500, gid=100	Propietario y grupo de los archivos
usrquota,grpquota	Cuotas de usuario y de grupo
gid=users	Establece grupo de usuario

```
mount /dev/loop0 -o async,atime,exec /mnt/ejemplo
```

-Modifique los elementos que sean necesarios para cada vez que arrancamos UML se realice el montaje de nuestro directorio home en el anfitrión en la máquina virtual en lugar de tener que hacerlo a mano, es decir, ejecutar manualmente mount none /mnt -t hostfs.

Hay que editar el archivo /etc/fstab con el editor Vi

```
Vim /etc/fstab
```

```
none / hostfs defaults 1 1
```

GESTOR DE PAQUETES: YUM

yum list	Lista los paquetes disponibles para instalar
yum list installed	Lista los paquetes instalados
yum list updates	Muestra todos los paquetes con actualizaciones disponibles
yum install <paquete>	Instala el paquete <nombre>
yum update	Actualiza todos los paquetes instalados
yum remove <paquete>	Borra <paquete>, así como los paquetes que dependen de él
yum list recent	Lista paquetes añadidos recientemente
yum clean packages	Limpia caché de paquetes
yum clean headers	Elimina todos los archivos cabecera
yum clean plugins	Elimina caché de plugins
yum clean rpmdb	Elimina el caché local de rpmdb
yum clean all	Elimina todo lo de arriba

Como somos usuario root, no tenemos acceso a los paquetes de nuestro ordenador. Para poder instalar un paquete, ejecutamos(en Fedora no se pueden descomprimir archivos por lo que hay que descomprimirlos antes en nuestro pc):

mount none /mnt/paquetes -t hostfs -o /home/Descargas/paquetes

GESTOR DE PAQUETES RPM

rpm -i <paquete>	Instala el paquete
rpm -e <paquete>	Borra el paquete instalado
rpm -U <paquete>	Borra y actualiza el paquete
rpm -qi <paquete>	Muestra información del paquete instalado
rpm -V <paquete>	Verifica la correcta instalación
Rpm -qc <paquete>	Muestra archivos de configuración
-v	Detalla la información
Rpm -a	Consulta todos los paquetes instalados
Rpm -g group	Consulta los paquetes con el group
-c	Lista archivos de configuracion
-d	Lista archivos de documentación
-i	Información(nombre, versión, descripción)
-l	Lista los archivos del paquete
-R	Lista las dependencias del paquete
-L	Lista archivos con licencia
rpm -qa grep nombre (sort)	Busca paquetes por su nombre y los ordena

Por ejemplo, si queremos ver cual es la versión de un paquete, ejecutamos:

rpm -qi <nombre-paquete> | grep Version | cut -d: -f2
rpm -qi <nombre-paquete> | grep Version | tr -s " " " " | cut -c11-15

Actividad 2.2. Creación de sistemas de archivos

El objetivo es simplemente formatear lógicamente las particiones creadas con anterioridad de forma consistente con el tipo de SA que se estableció que iba a ser alojado. En la primera partición crearemos un SA de tipo ext3 y en la segunda un ext4. La orden que permite establecer un SA de los reconocidos dentro del sistema Linux sobre una partición de disco es **mke2fs** (consulta el manual en línea para familiarizarte con sus opciones). El resultado de la ejecución de esta orden es el formateo lógico de la partición escogida utilizando el SA que se ha seleccionado. Utiliza el manual en línea para conocer cómo ejecutar la orden de creación de SA. **mke2fs** es la orden genérica para creación de sistemas de archivos. Como requisito es necesario que establezcas dos etiquetas de volumen para los SAs: **LABEL_ext3** para la primera partición y **LABEL_ext4** para la segunda.

```
#Para establecer un SA sobre una partición ya realizada, usaremos mke2fs:
mke2fs -t ext3 -L LABEL_ext3 /dev/loop0
mke2fs -t ext4 -L LABEL_ext4 /dev/loop1
#-t especifica el tipo del SA
#-L Establece la etiqueta
```

Actividad 2.3. Personalización de los metadatos del SA

Consultando el manual en línea para la orden **tune2fs** responde a las siguientes preguntas:

- (a) ¿Cómo podrías conseguir que en el siguiente arranque del sistema se ejecutara automáticamente **e2fsck** sin que se haya alcanzado el máximo número de montajes?
(b) ¿Cómo podrías conseguir reservar para uso exclusivo de un usuario **username** un número de bloques del sistema de archivos?

```
#La opción -c ajusta el número máximo de montajes y la opción -C
#establece el número de veces que el sistema se ha montado
#Hay que ponerle cualquier número más grande a la opción -C
```

```
tune2fs -c 1 -C 2 /dev/loop0
```

```
#La opción -u sirve para establecer el usuario que puede usar los bloques reservados
#la opción -r sirve para ajustar el número de bloques reservados
#En este ejemplo, el usuario va a ser <usuario> y el número de bloques son 50
```

```
tune2fs -u <usuario> -r 50 /dev/loop0
```

Actividad 2.4. Montaje de sistemas de archivos

Utiliza el manual en línea para descubrir la forma de montar nuestros SAs de manera que cumplas los siguientes requisitos:

3. El SA etiquetado como **LABEL_ext3** debe estar montado en el directorio **/mnt/SA_ext3** y en modo de solo lectura.
4. El SA etiquetado como **LABEL_ext4** debe estar montado en el directorio **/mnt/LABEL_ext4** y debe tener sincronizadas sus operaciones de E/S de modificación de directorios.

```
#Debemos crear la carpeta /mnt/SA_ext3 ya que si no está creada el
#siguiente comando da error:
mkdir /mnt/SA_ext3
mount /dev/loop0 -r /mnt/SA_ext3
```

```
mkdir /mnt/LABEL_ext4
mount /dev/loop1 -o dirsync /mnt/LABEL_ext4
```

```
#Usamos el siguiente comando para ver que se han creado correctamente:
mount -l
```

Actividad 2.5. Automontaje de Sistemas de Archivos

Escribe las dos líneas necesarias en el archivo /etc/fstab para que se monten automáticamente nuestros dos SA en el arranque del sistema con los mismos requisitos que se han pedido en la Actividad 2.4.

Como somos usuarios root, tendríamos que editar el fichero con el editor Vi

```
echo "/dev/loop0 /mnt/SA_ext3 ext3 ro 0 0" >> /etc/fstab
echo "/dev/loop1 /mnt/LABEL_ext4 ext4 dirsyc 0 0" >> /etc/fstab
```

Actividad 2.7. Trabajo con el gestor de paquetes YUM

Encuentra los archivos de configuración de YUM y explora las distintas órdenes disponibles en YUM ejecutándolas. En concreto, lista todos los paquetes instalados y disponibles, elimina el paquete instalado que te indique el profesor de prácticas, y a continuación vuelve a instalar el mismo paquete haciendo uso de los paquetes que se encuentran disponibles en /fenix/depar/lsi/so/paquetes. Para obtener acceso a este directorio del sistema de archivos anfitrión ejecute la siguiente orden de montaje una vez lanzado el sistema operativo User Mode Linux (UML):
mount none /<directorio-punto-montaje> /fenix/depar/lsi/so/paquetes

#Para buscar el directorio de YUM ejecutamos:
whereis yum

#Ahora bien, busquemos el archivo de configuración el
#cual se encuentra en /etc/yum/yum.conf

#Utilizades YUM

yum list	#Lista los paquetes disponibles
yum list installed	#Lista los paquetes instalados
yum remove <nombre-paquete>	#Elimina el paquete
yum install <nombre-paquete>	#Instala el paquete

#Montamos el paquete para tener acceso a él
mkdir /mnt/new
mount none /mnt/new -t hostfs -o /home/albduranlopez/Escritorio/SO/Paquetes

#Desde la shell, sin ser root descomprimos los paquetes
unzip paquetes.zip
#Ahora, siendo usuario root(Fedora), instalamos los paquetes usando:
yum install <nombre-paquete>

Actividad 2.8. Trabajo con el gestor de paquetes rpm

En primer lugar deseamos mostrar cierta metainformación acerca de uno o más paquetes ya instalados. Para ello debes utilizar la orden rpm con las opciones adecuadas. Utiliza el manual en línea si no sabes ya las opciones que debes utilizar.

1. Muestra la información general (nombre, versión, arquitectura, grupo, descripción, etc.) y lista los archivos que contiene un paquete ya instalado haciendo uso de la orden rpm y un único conjunto de opciones.

Guía Práctica de Sistemas Operativos-402. Idem que el anterior pero mostrando únicamente los archivos de configuración que contiene el paquete.

3. Escribe una orden que muestre los paquetes requeridos por un paquete determinado que se encuentre instalado en el sistema. Escribe la orden que devuelva el mismo resultado pero para un paquete no instalado en el sistema.

4. Instala el paquete quota que encontrarás en el directorio de software de la asignatura (directorio que ya has montado en la Actividad 2.7).

5. Instala y desinstala el paquete sysstat mostrando en pantalla también la máxima información posible acerca del propio proceso de eliminación del paquete.

```
#Mostrar información general de un paquete instalado
rpm -qi rpm-4.8.1-5.fc14.i686 #nombre del paquete
```

```
#Muestra archivos de configuración de un paquete
rpm -qc rpm-4.8.1-5.fc14.i686
```

```
#Mostrar información de un paquete instalado
rpm -ql rpm-4.8.1-5.fc14.i686
```

```
#Mostrar información de un paquete no instalado
rpm -q --whatrequires quota-3.17-13.fc14.i686.rpm
```

```
#Instalamos el paquete "quota...rpm" con la orden:
rpm -i quota-3.17-13.fc14.i686.rpm
```

```
#Para eliminar un paquete previamente instalado, usamos la orden:
rpm -e <nombre-paquete>
rpm -e sysstat-9.0.6-3.fc13.i686.rpm
```

```
#Como en este caso nos piden que a la hora de desinstalar el paquete sea
#de la forma más detallada posible, ejecutaremos lo siguiente:
rpm -ev sysstat-9.0.6-3.fc13.i686.rpm
```

```
#La opción -v detalla la información
#En este caso, da el error de que el paquete no está instalado:
error: package sysstat-9.0.6-3.fc13.i686.rpm is not installed
```

Actividad 2.9. Sistema de cuotas para el sistema de archivos tipo ext3

En esta actividad se van a presentar los pasos que necesitas llevar a cabo para establecer el sistema de cuotas de disco en Linux. El objetivo será activar el sistema de cuotas sobre el sistema de archivos tipo ext3 que has creado con anterioridad.

-1. Editar el archivo `/etc/fstab` y activar el sistema de cuotas de usuario para el SA tipo ext3. Busca cómo se especifica esta opción en el manual en línea. Una ayuda para la búsqueda es que la realices sobre la orden `mount` y recuerdes que las opciones de montaje vienen especificadas en los apartados: **FILESYSTEM INDEPENDENT MOUNT OPTIONS** y **FILESYSTEM SPECIFIC MOUNT OPTIONS**.

-2. Montar de nuevo el SA en el espacio de nombres para que se active la opción previamente establecida. Usa la siguiente orden:

```
#> mount -o remount <directorio_punto_de_montaje>
```

-3. Crear el archivo que permite llevar el control de cuotas de usuario para el SA. El nombre de este archivo es `aquota.user`. Para ello utiliza la siguiente orden:

```
#> quotacheck -nm <directorio_punto_de_montaje>
```

-4. Ahora procedemos a activar el sistema de control de cuotas de usuario. Para ello ejecuta la orden:

```
#> quotaon -a
```

-5. Ahora solo falta editar la cuota para cada usuario del sistema mediante la siguiente orden. En este caso, establece los parámetros para cada usuario existente. Puede ser buena idea utilizar el archivo `/etc/passwd` para localizar los nombres.

```
#> edquota username
```

-6. Para finalizar estableceremos el periodo de gracia para el límite soft.

```
#> edquota -t
```

```
#Editamos el directorio /etc/fstab añadiendo la información que sigue:  
vi /etc/fstab
```

```
#La explicación de cada campo está explicada en un ejercicio anterior  
#Añadimos las opciones usrquota/grpquota dependiendo si las cuotas  
#son por usuario o grupo  
/dev/loop0 /mnt/SA_ext3 ext3 noatime,usrquota,grpquota 0 0
```

```
#Ejecutamos esta orden para montar de nuevo el SA en el espacio  
#de nombres para que se active la opción anterior  
mount -o remount /mnt/SA_ext3
```

```
#Crear el archivo que permite llevar el control de cuotas de usuariopara el SA.  
#El nombre deeste archivo es aquota.user. Para ello se usa la siguiente orden:  
#quotacheck -nm /mnt/SA_ext3
```

```
#Ahora procedemos a activar el sistema de control de cuotas de usuario.  
#Para ello ejecuta la orden:  
quotaon -a
```

```
#Ahora solo falta editar la cuota para los del sistema mediante la siguiente orden  
edquota <usuario>
```

```
#Para finalizar estableceremos el periodo de gracia para el límite soft.  
edquota -t
```

Actividad 2.10. Establecer límites sobre recursos de un SA

Establece los límites de bloques e i-nodos para un par de usuarios del sistema UML sobre el que trabajas en el laboratorio.

#Usaremos la orden man setquota, donde encontramos las opciones:

#-u user (para seleccionar el usuario)

#-b "Read information to set from stdin (input format is 'name' 'block-softlimit' 'block-hardlimit' 'inode-softlimit' 'inode-hardlimit' "

#En este caso asignamos 250M de soft limit, 300MB de hardlimit y sin limite de inodos para el usuario

setquota -u <usuario> 250000 300000 0 0

#Con la orden edquota se haría:

edquota <usuario>

#Se tendría que editar el fichero correspondiente con edquota(el comando edquota abre el editor correspondiente)

SESIÓN 3

uptime: Hora actual, el tiempo que lleva en marcha el sistema, el nº de usuarios conectados y la carga media del sistema en los últimos 1, 5 y 15 minutos

w: Lo mismo que uptime pero también lo que están haciendo los usuarios

time: Muestra el tiempo total que el programa ha estado ejecutándose (real), el tiempo que se ha ejecutado en modo usuario (user) y el tiempo que se ha ejecutado en modo supervisor (sys)

time ps

nice: Establecer prioridades en un rango de [-20,19]. Un valor negativo == mayor prioridad
Sólo lo puede hacer el root

Ejecutar un script con la mayor prioridad posible:

```
# nice -n-20 ./prueba.sh
```

Con la mínima prioridad posible:

```
# nice -n19 ./prueba.sh
```

renice: Altera el valor de prioridad de uno o más procesos en ejecución.

renice -15 -p 1617

renice 5 -u root

Ejecutar en segundo plano: ./prueba.sh &

PS

ps: Muestra información sobre procesos en ejecución.

USER : usuario que lanzó el programa

PID : identificador del proceso

PPID : identificador del proceso padre

%CPU : porcentaje entre el tiempo usado realmente y el que lleva en ejecución

%MEM : fracción de memoria consumida (es una estimación)

VSZ : tamaño virtual del proceso (código+datos+pila) en KB

RSS : memoria real usada en KB

TTY : terminal asociado con el proceso

STAT : estado del proceso que puede ser una de las letras mostrada en la siguiente tabla:

R: en ejecución o listo (Running o Ready)

S: durmiendo (Sleeping)

T: parado (sTopped)

Z: proceso Zombie

D: durmiendo ininterrumpible (normalmente E/S)

N: prioridad baja (> 0)

<: prioridad alta (< 0)

s: líder de sesión

l: tiene multi-thread

+: proceso foreground

L: páginas bloqueadas en memoria

I	Formato largo
---	---------------

u	Muestra usuario y hora de inicio
e	Selecciona todo proceso de sistema
f	Muestra información completa
x	Procesos no controlados por ninguna terminal
r	Solo procesos que se están ejecutando

Ordenar procesos

ps aux --sort uid	Desde el id más bajo hasta el más alto
ps aux --sort -uid	Desde el id más bajo hasta el más alto
ps aux --sort pmem	Por el uso de memoria
ps aux --sort pid	Por el pid

TOP Ejecutamos top, y una vez dentro de su interfaz:

r	Cambia la prioridad
k	Matar procesos
n	Cambia el nº de procesos que se muestran
i	Descarta procesos inactivos y zombies

MPSTAT Instalado por defecto en Fedora

CPU	Número del procesador
%user	Porcentaje de uso de la CPU con tareas a nivel de usuario
%nice	Porcentaje de uso de la CPU con tareas a nivel de usuario con prioridad "nice" >0
%sys	Porcentaje de uso de la CPU para tareas del sistema (no incluye el tratamiento de interrupciones) (modo núcleo)
%iowait	Porcentaje de tiempo que la CPU estaba "desocupada" mientras que el sistema tenía pendientes peticiones de E/S
%irq	Porcentaje de tiempo que la CPU gasta con interrupciones hardware
%soft	Porcentaje de tiempo que la CPU gasta con interrupciones software (la mayoría son llamadas al sistema)
%idle	Porcentaje de tiempo que la CPU estaba "desocupada" y el sistema no tiene peticiones de disco pendientes
%intr/s	Número de interrupciones por segundo recibidas por el procesador

mpstat [intervalo] [número]

Intervalo: Cada cuántos segundos debe mostrar los datos.

Número: Cuántos muestreos se solicitan

FREE Para ver espacio de swap libre y ocupado y memoria ram // Espacio libre

Free -b	Muestra memoria en bytes
-k	Memoria en kilobytes(por defecto)
-m	Megabytes
-g	En gigabytes
-l	Muestra memoria detallada
-s x	Muestra el resultado cada x segundos

Esta última equivale a: watch -n x free

VMSTAT Sirve para supervisar el sistema mostrando información de memoria pero también acerca de procesos, E/S y CPU.

Ejecutar 20 iteraciones mostrando información cada 2 segundos:

vmstat 2 20

Procs

r: El número de procesos esperando su tiempo de ejecución.

b: El número de procesos en espera ininterrumpible.

w: El número de procesos en espacio de trasiego pero sin embargo ejecutables. Este campo se calcula, pero Linux nunca hace trasiego desesperado.

Memory

swpd: la cantidad de memoria virtual empleada (kB).

free: la cantidad de memoria inactiva (kB).

buff: la cantidad de memoria empleada como búferes (kB).

Swap

si: Cantidad de memoria traída del espacio de trasiego desde disco (kB/s).

so: Cantidad de memoria trasegada al disco (kB/s).

IO

bi: Bloques enviados a un dispositivo de bloques (bloques/s).

bo: Bloques recibidos desde un dispositivo de bloques (bloques/s).

System

in: El número de interrupciones por segundo, incluyendo al reloj.

cs: El número de cambios de contexto por segundo.

CPU

Éstos son porcentajes de tiempo total de CPU.

us: tiempo de usuario

sy: tiempo de sistema

id: tiempo de inactividad

LS

-l	En formato largo
-n	En formato largo con el numeric user &groups ID
-la	Igual que ls -l pero muestra archivos ocultos (formato largo)
-R	Muestra recursivamente
-li	En formato largo añadiendo el campo del número de inodo
-1 -uno	Muestra en 1 sola columna

-u	Clasifica según el último acceso
-f	Ordena según estén almacenados en el disco
-h (human-readable)	Ls -lh Muestra el tamaño
-X	Alfabéticamente por la extensión
-c	Fecha de última modificación de la información del estado del archivo
-d	Muestra información del archivo, no la del enlace. En caso de directorios no muestra el contenido sino los metadatos del directorio
-L	Lista la información del fichero correspondiente a un enlace simbólico en lugar de la del mismo enlace
-k	En kilobytes
-t	Por la fecha de modificación

DF La orden df permite visualizar, para cada SA montado, información sobre su capacidad de almacenamiento total, el espacio usado para almacenamiento y el espacio libre restante, y el punto de montaje en la jerarquía de directorios para cada SA.

Utilizando la orden df -i podemos visualizar la información sobre los inodos de cada SA montado.

-Número de bloques de tamaño 4kb del directorio /dir:

df --block-size=4 /dir

DU

-Para poder ver el espacio en disco que gasta un directorio de la jerarquía de directorios, y todo el subárbol de la jerarquía que comienza en él, se utiliza la orden du.

-**du** contabiliza el número de bloques de disco asignados estén o no completamente ocupados.

-m	megabytes
du -b	Muestra el tamaño en bytes
du -c	Muestra un total
du -h (human readable)	Añade una letra indicativa del tamaño. M megabytes
-k	kilobytes

-¿Bloques de datos que está usando la partición raíz?

Du /

-¿i-nodos que se están usando, inodos de una partición?

df -i

-¿Cual es el tamaño de un directorio?

du -h /dir

LN Para crear enlaces duros o enlaces simbólicos sobre un archivo creado previamente se utiliza la orden ln. Como argumentos básico debemos proporcionarle el nombre del archivo a enlazar (link target) y el nuevo nombre de archivo (link name).

-s	Enlace simbólico o blando
-d	Permite al superusuario hacer enlaces duros a directorios
-f	Borra los ficheros de destino que ya existieran
-i	Pregunta antes de borrar ficheros de destino existentes
-v	Muestra el nombre de cada fichero antes de enlazarlo

```
ln -s archivo.txt softlink
ln archivo.txt hardlink
```

Si al hacer ls -l tiene carácter l, es enlace simbólico

Al hacer ls -i, comprobamos que tienen un inodo distinto (hacer comando stat también sirve)

En enlace duro(solo entre ficheros), comparte inodo al hacer ls -li, tiene carácter 2, al hacer un cambio en uno de ellos también se cambia el resto.

-Crear en el directorio /root un enlace duro y un enlace simbólico a cualesquiera de los programa que aparecen en /bin.

```
ln /bin/sh /root/enlace_shell
```

```
ln -s /bin/bash /root/enlace_bash
```

MKNOD Podemos crear archivos especiales de dispositivo, tanto de bloques (buffered) como de caracteres (unbuffered) utilizando la orden mknod. Esta orden permite especificar el nombre del archivo y los números principal (major) y secundario (minor). Estos números permiten identificar a los dispositivos en el kernel, concretamente en la Tabla de Dispositivos.
mknod nombre {b,c} mayor menor

Dispositivo de bloques:

```
mknod bloques b 100 50
```

Dispositivo de caracteres:

```
mknod caracteres c 100 50
```

```
ls -lai
```

Actividad 3.1. Consulta de estadísticas del sistema

Responde a las siguientes cuestiones y especifica, para cada una, la opción que has utilizado (para ello utiliza man y consulta las opciones de las órdenes anteriormente vistas:

a) ¿Cuánto tiempo lleva en marcha el sistema?

b) ¿Cuántos usuarios hay trabajando?

c) ¿Cuál es la carga media del sistema en los últimos 15 minutos?

#Ejecutamos la orden uptime la cual muestra:

```
[root@localhost ~]# uptime
```

```
04:31:59 up 41 min, 1 user, load average: 0.00, 0.01, 0.02
```

a)El sistema lleva en marcha 41 minutos

b)Hay un 1 usuario

c)La carga media del sistema ha sido de 0.02

Actividad 3.2. Prioridad de los procesos

a) Crea un script o guión shell que realice un ciclo de un número variable de iteraciones en el que se hagan dos cosas: una operación aritmética y el incremento de una variable. Cuando terminen las iteraciones escribirá en pantalla un mensaje indicando el valor actual de la variable.

Este guión debe tener un argumento que es el número de iteraciones que va a realizar. Por ejemplo, si el script se llama prueba_procesos, ejecutaríamos:

prueba_procesos 1000

el valor de la variable es 1000

b) Ejecuta el guión anterior varias veces en background (segundo plano) y comprueba su prioridad inicial. Cambia la prioridad de dos de ellos, a uno se la aumentas y a otro se la disminuyes, ¿cómo se comporta el sistema para estos procesos?

c) Obtén los tiempos de finalización de cada uno de los guiones del apartado anterior.

```
#!/bin/bash

if [ $# -ne 1 ];
then
    echo "Número de parámetros incorrecto"
else
    CONTADOR=0
    while [ $CONTADOR -lt $1 ]; do
        let CONTADOR=CONTADOR+1
        sleep 0.01
    done

    echo "El valor de la variable es $CONTADOR"
fi
```

b) #Para ejecutar el guión como root, montamos el archivo para tenerlo disponible en Fedora:
mount none /mnt -t hostfs -o /home/albduranlopez/Escritorio

#Ejecutamos el guión en segundo plano:

```
[root@localhost mnt]# ./ejercicio2.sh 900 &
```

```
[4] 9906
```

```
[root@localhost mnt]# ./ejercicio2.sh 900 &
```

```
[5] 15393
```

#Ejecutamos la orden top para comprobar la prioridad del proceso, en mi caso ambos tienen #prioridad 0. Por tanto, ejecutamos:

```
[root@localhost mnt]# renice 15 -p 9906
```

```
9906: old priority 0, new priority 15
```

```
[root@localhost mnt]# renice -15 -p 15393
```

```
15393: old priority 0, new priority -15
```

#A un proceso se le asigna una prioridad de 15 y a otro de -15. Ejecutamos de nuevo la orden top y como podemos observar, al proceso con prioridad -15 se le asigna mayor cpu. Para ello podemos ejecutar 'top' o la orden 'ps aux'

```
USER  PID %CPU %MEM  VSZ   RSS TTY      STAT START   TIME COMMAND
root   15393 0.1  0.1  3120 1132 tty0    SN   12:52   0:01 /bin/bash ./eje
root    9906 0.0  0.1  3120 1132 tty0    SN   12:52   0:00 /bin/bash ./eje
```

c)#Ejecutamos de nuevo el guión y le asignamos una prioridad mayor

```
[root@localhost mnt]# time ./ejercicio2.sh 900 &
```

```
[6] 32312
```

```
[root@localhost mnt]# renice -15 -p 32312
```

```
32312: old priority 0, new priority -15
```

```
[root@localhost mnt]# El valor de la variable es 900
```

```
real    0m35.172s
user    0m0.020s
sys     0m0.010s
```

```
[root@localhost mnt]# time ./ejercicio2.sh 900 &
[6] 8100
```

```
[root@localhost mnt]# El valor de la variable es 900
```

```
real    0m36.138s
user    0m0.010s
sys     0m0.000s
```

#Como podemos observar, si le asignamos una prioridad mayor, el proceso termina 1 segundo antes

Actividad 3.3. Jerarquía e información de procesos

a) La orden pstree muestra el árbol de procesos que hay en ejecución. Comprueba que la jerarquía mostrada es correcta haciendo uso de la orden ps y de los valores "PID" y "PPID" de cada proceso.

b) Ejecuta la orden ps con la opción -A, ¿qué significa que un proceso tenga un carácter "?" en la columna etiquetada como TTY?

a) Ejecutar pstree y ps -Alf y comprobar que coinciden los PID y PPID de los procesos padres e hijos

b) Significa que no hay ninguna terminal asociada con el proceso.

Actividad 3.4. Estadísticas de recursos del sistema

Responde a las siguientes cuestiones y especifica, para cada una, la orden que has utilizado:

a) ¿Qué porcentaje de tiempo de CPU se ha usado para atender interrupciones hardware?

b) ¿Y qué porcentaje en tratar interrupciones software?

c) ¿Cuánto espacio de swap está libre y cuánto ocupado?

#Ejecutamos el comando mpstat donde %irq indica el porcentaje de tiempo que la CPU gasta con interrupciones hardware, en nuestro caso 0.00. La columna %soft muestra el porcentaje de tiempo que la CPU gasta con interrupciones software, 0.00

```
[root@localhost paquetes]# mpstat
```

```
Linux 3.0.4 (localhost)      10/15/16      _i686_ (1 CPU)
13:42:45  CPU   %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %idle
13:42:45  all    0.16   0.22   0.02   0.67   0.00   0.00   0.00   0.00   98.94
```

#Para el apartado c), ejecutamos la orden top que muestra:

```
top - 13:50:09 up 1:10, 1 user, load average: 0.00, 0.01, 0.02
Tasks: 40 total, 2 running, 38 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 90.4%id, 9.6%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1020564k total, 57192k used, 963372k free, 4796k buffers
Swap: 0k total, 0k used, 0k free, 34604k cached
```

#En mi caso tengo 0k de espacio total y 0k en uso (Estos resultados son desde Fedora, si ejecuto la orden top como usuario normal, si tengo espacio de swap)

Actividad 3.6. Utilización de vmstat

Intente reproducir el escenario justo descrito anteriormente supervisando la actividad del sistema mediante la ejecución periódica de vmstat tal cual se ha descrito, y proporcione como muestra la salida almacenada en un archivo de texto.

```
#!/bin/bash
vmstat >> sistema.txt

procesos=`cat "sistema.txt"|tail -1|tr -s " " "|cut -d " " -f2`
procesos_swap=`cat "sistema.txt"|tail -1|tr -s " " "|cut -d " " -f17`
incremento_swap=`cat "sistema.txt"|tail -1|tr -s " " "|cut -d " " -f9`
memoria_libre=`cat "sistema.txt"|tail -1|tr -s " " "|cut -d " " -f5`

echo "Hay un total de " $procesos " procesos en cola"
echo "Hay " $procesos_swap " procesos ejecutándose en el área de intercambio"
echo "EL incremento de la memoria es de: " $incremento_swap
echo "Hay un total de " $memoria_libre "memoria libre en Kb"
```

Actividad 3.7. Consulta de metadatos de archivo

Anota al menos dos nombres de archivo de dispositivo de bloques y dos nombres de dispositivo de caracteres de tu sistema UML. Anota los nombres de los archivos ocultos de tu directorio de inicio como usuario root que tienen relación con el intérprete de órdenes que tienes asignado por defecto. Ahora efectúa la misma tarea pero en una consola de terminal del sistema Ubuntu que arrancas inicialmente en el laboratorio de prácticas. ¿Qué diferencias encuentras entre los nombres de los archivos?

```
ls -a /root
```

Actividad 3.8. Listados de metadatos de archivos: ls

Conocemos la sintaxis de la orden para obtener un listado en formato largo (“long listing format”). Manteniendo la opción de listado largo añade las opciones que sean necesarias para obtener un listado largo con las siguientes especificaciones:

- **Que contenga el campo “access time” de los archivos del directorio especificado y que esté ordenado por dicho campo.**
- **Que contenga el campo “ctime” de los archivos del directorio especificado y que esté ordenado por dicho campo.**

Para más información sobre la orden ls consultar el manual Texinfo. Utiliza la orden: \$> info coreutils 'ls invocation'

```
#Que contenga el campo “access time” de los archivos del directorio especificado y que esté ordenado por dicho campo:
```

```
ls -ltu
```

```
#Que contenga el campo “ctime” de los archivos del directorio especificado y que esté ordenado por dicho campo
```

```
ls -ltc
```

Actividad 3.9. Metadatos del sistema de archivos: df y du

Resuelve las siguientes cuestiones relacionadas con la consulta de metadatos del sistema de archivos:

- 1. Comprueba cuántos bloques de datos está usando la partición raíz del sistema UML del laboratorio. Ahora obtén la misma información pero expresada en “human readable format”: Megabytes o Gigabytes. Para ello consulta en detalle el manual en línea.**
- 2. ¿Cuántos inodos se están usando en la partición raíz? ¿Cuántos nuevos archivos se podrían crear en esta partición?**

3. ¿Cuál es el tamaño del directorio /etc? ¿Y el del directorio /var? Compara estos tamaños con los de los directorios /bin, /usr y /lib. Anota brevemente tus conclusiones.

4. Obtén el número de bloques de tamaño 4 KB que utiliza la rama de la estructura jerárquica de directorios que comienza en el directorio /etc. En otras palabras, los bloques de tamaño 4 KB del subárbol cuya raíz es /etc. ¿Cuál es el tamaño de bloque, por omisión, utilizado en el SA?

a) #Bloques de datos que está usando la partición raíz de mi sistema:

```
du /
```

#Misma información anterior pero expresada en “human readable format”. Ejecutamos el comando 'man du' en el cual nos muestra la información de du:

```
du -h /
```

b) #Mostrar los inodos libres que se están usando

```
[root@localhost ~]# df -i /
```

```
Filesystem      Inodes   IUsed   IFree IUse% Mounted on
```

```
LABEL=ROOT      65536  14666  50870  23%    /
```

#hay un total de 50870 inodos libres por lo que se podrían crear 50870 archivos

c) ¿Cuál es el tamaño del directorio /etc?

```
[root@localhost ~]# du -h /etc
```

```
....
```

```
21M    /etc
```

#El tamaño de /etc es de 21 MB

¿Y el del directorio /var?

```
[root@localhost ~]# du -h /var
```

```
...
```

```
14M    /var
```

#El tamaño de /var es de 14MB

#Compara estos tamaños con los de los directorios /bin, /usr y /lib

```
5.3M   /bin
```

```
24M    /lib
```

```
297M   /usr
```

#El directorio que más ocupa es /usr ya que contiene la información más importante, en el directorio /lib están las librerías por lo que tiene un tamaño considerable y en el directorio /bin es el que menos ocupa.

d) df --block-size=4 /etc

#El tamaño de bloque por defecto es de 1k

Actividad 3.10. Creación de enlaces con la orden ln

Construye los mismos enlaces, duros y simbólicos, que muestra la salida por pantalla anterior.

Para ello crea los archivos archivo.txt y target_hardLink2.txt y, utilizando el manual en línea para ln, construye los enlaces softLink, hardLink y hardLink2. Anota las órdenes que has utilizado.

¿Por qué el contador de enlaces del archivo archivo.txt vale 2 si sobre el existen un enlace duro hardLink y un enlace simbólico softLink?

#Creamos los archivos que nos piden:

```
touch archivo.txt target_hardLink2.txt
```

#Nos ayudamos del manual en línea: man ln

```
ln -s archivo.txt softlink
ln archivo.txt hardlink
ln target_hardLink2.txt hardlink2
mkdir D1

[root@localhost ~]# ls -lai
total 36
 311 dr-xr-x---  3 root root 4096 Oct 15 15:33 .
   2 dr-xr-xr-x 22 root root 4096 Oct 15 14:37 ..
  58 -rw-----  1 root root   53 Sep 13  2011 .bash_history
3958 -rw-r--r--  1 root root  18 Mar 30  2009 .bash_logout
3959 -rw-r--r--  1 root root 176 Mar 30  2009 .bash_profile
3960 -rw-r--r--  1 root root 176 Sep 22  2004 .bashrc
3961 -rw-r--r--  1 root root 100 Sep 22  2004 .cshrc
3962 -rw-r--r--  1 root root 129 Dec  3  2004 .tcshrc
14241 drwxr-xr-x  2 root root 4096 Oct 15 15:33 D1
14238 -rw-r--r--  2 root root   0 Oct 15 15:31 archivo.txt
14238 -rw-r--r--  2 root root   0 Oct 15 15:31 hardlink
14239 -rw-r--r--  2 root root   0 Oct 15 15:31 hardlink2
14175 lrwxrwxrwx  1 root root  11 Oct 15 15:31 softlink -> archivo.txt
14239 -rw-r--r--  2 root root   0 Oct 15 15:31 target_hardLink2.txt
[root@localhost ~]# cd D1
[root@localhost D1]# ls -lai
total 8
14241 drwxr-xr-x  2 root root 4096 Oct 15 15:33 .
  311 dr-xr-x---  3 root root 4096 Oct 15 15:33 ..
```

#Como podemos observar, se muestran las mismas ordenes que nos han pedido

b) En el contador de enlaces del archivo.txt se muestra un 2 ya que solo se cuentan los enlaces duros.

Actividad 3.11. Trabajo con enlaces

Proporciona las opciones necesarias de la orden ls para obtener la información de metadatos de los archivos de un directorio concreto en los dos casos siguientes:

- En el caso de que haya archivos de tipo enlace simbólico, la orden debe mostrar la información del archivo al que enlaza cada enlace simbólico y no la del propio archivo de tipo enlace simbólico.
- En el caso de enlaces simbólicos debe mostrar la información del enlace en sí, no del archivo al cual enlaza. En el caso de directorios no debe mostrar su contenido sino los metadatos del directorio.

- a) ls -L
- b) ls -d

Actividad 3.12. Creación de archivos especiales

Consulta el manual en línea para la orden mknod y crea un dispositivo de bloques y otro de caracteres. Anota las órdenes que has utilizado y la salida que proporciona un ls -li de los dos archivos de dispositivo recién creados. Puedes utilizar las salidas por pantalla mostradas en esta sección del guión para ver el aspecto que debe presentar la información de un archivo de dispositivo.

```
#Nos ayudamos del manual en linea: man mknod
man mknod
mknod [opciones] nombre {bc} mayor menor
```

En nuestro caso, el dispositivo de bloques lo llamaremos “bloques” y el de caracteres lo llamamos “caracteres”

```
[root@localhost D1]# mknod bloques b 50 30
[root@localhost D1]# mknod caracteres c 50 30
[root@localhost D1]# ls -li
total 0
14248 brw-r--r-- 1 root root 50, 30 Oct 15 15:48 bloques
14249 crw-r--r-- 1 root root 50, 30 Oct 15 15:48 caracteres
```

SESIÓN 4 – No usaremos root

AT

at: ordenar la ejecución de órdenes a una determinada hora

atq: consultar la lista de órdenes

atrm: eliminar órdenes

batch: ordenar la ejecución de órdenes que se ejecutarán cuando la carga del sistema sea baja

```
at [-q queue] [-f <script>] [-mldbv] TIME
```

Ejercicio 4.1

**Cree un archivo genera-apunte que escriba la lista de hijos del directorio home en un archivo de nombre listahome-`date +%Y-%j-%T-\$`
año, día dentro del año, la hora actual y pid (consulte la ayuda de date).**

```
#script
genera-apunte=listahome-`date +%Y-%j-%T-$`
```

```
#terminal
at -f ./script now + 1 minutes
```

-Acepta horas con formato HH:MM (Si esa hora ya ha pasado, se asume que es del día siguiente.)

-También se puede especificar midnight, noon, o teatime (4pm) y se puede especificar la hora con los sufijos AM o PM para ejecutar por la mañana o por la tarde.

-También se puede indicar el día de ejecución, proporcionando la fecha con el formato nombre-del-mes día y opcionalmente año, o bien indicándolo con el formato MMDDYY o MM/DD/YY o DD.MM.YY.

-La especificación de una fecha debe seguir a la de la hora del día. También se pueden dar horas como now + count unidades-de-tiempo, donde unidades-de-tiempo puede ser minutes, hours, days, o weeks

-Today, tomorrow

-at 4pm + 3 days

-at 10am Jul 31

-at 1am tomorrow

month_name : JAN | FEB | MAR | APR | MAY | JUN
| JUL | AUG | SEP | OCT | NOV | DEC

cat /usr/share/doc/at/timespec

Actividad 4.3

a) a medianoche de hoy = at midnight

b) un minuto después de la medianoche de hoy = at midnight + 1 minutes

c) a las 17 horas y 30 minutos de mañana= at tomorrow 17:30pm

d) a la misma hora en que estemos ahora pero del día 25 de diciembre del presente año = at Dec 25 (no se puede poner una hora pasada por lo que interpreta el año siguiente)

e) a las 00:00 del 1 de enero del presente año= at midnight Jan 1

BATCH

La orden batch es equivalente a at excepto que no especificamos la hora de ejecución, sino que el trabajo especificado se ejecutará cuando la carga de trabajos del sistema esté bajo cierto valor umbral que se especifica a la hora de lanzar el demonio atd.

Su cola por defecto es la b

CRON

Muchas de las tareas de administración se tienen que llevar a cabo de forma periódica, a continuación veremos cómo el sistema operativo nos brinda facilidades para ello.

Uno de los demonios básicos es cron, responsable de ejecutar órdenes con una periodicidad determinada. La especificación de las tareas que se desea que ejecute se hace construyendo un archivo (llamado archivo "crontab") que deberá tener un determinado formato, (llamado formato "crontab"). Será con la orden crontab con la que indicamos el archivo con formato "crontab" que deseamos comunicar al demonio cron.

minuto	hora	día-del-mes	mes	día-de-la-semana	orden
--------	------	-------------	-----	------------------	-------

-A las 20 horas y 1 minuto de lunes a viernes:

1 20 * * 1-5

-Cada media hora el viernes y cada media hora el día 13 del mes:

0,30 * 13 * 5

-Cada minuto:

* * * * *

-Todos los días a las 0 horas 0 minutos

0 0 * * *

Instead of the first five fields, one of eight special strings may appear:

string	meaning
-----	-----
@reboot	Run once, at startup.
@yearly	Run once a year, "0 0 1 1 *".
@annually	(same as @yearly)
@monthly	Run once a month, "0 0 1 * *".

@weekly Run once a week, "0 0 * * 0".
@daily Run once a day, "0 0 * * *".
@midnight (same as @daily)
@hourly Run once an hour, "0 * * * *".

```
# run five minutes after midnight, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun echo "run at 5 after 4 every sunday"
# Run on every second Saturday of the month
0 4 8-14 * * test $(date +%u) -eq 6 && echo "2nd Saturday"
```

SHELL=/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

man 5 crontab

Instala el paquete `banner-1.2.2-1.i586.rpm` que contiene la orden `banner`, que permite poner engran tamaño un mensaje. Utilizar esta orden para mostrar el mensaje “UGR: sistemas operativos”. Modificar los elementos necesarios para que este mensaje se muestre el primer lunes de cada mes a las 12:00 horas.

Rpm -i banner-1.2.2-1.i586.rpm

banner “Ugr:sistemas operativos”

vim /etc/crontab

0 12 1-7 * 1 root banner “UGR: sistemas operativos”

Actividad 4.1. Consulta de información sobre procesos demonio

A partir de la información proporcionada por la orden `ps` encuentre los datos asociados a los demonios `atd` y `crond`, en concreto: quién es su padre, qué terminal tienen asociado y cuál es su usuario.

Ejecutamos la orden:

ps -alx | grep atd

ps -alx | grep cron

#La 4ª columna indica el proceso padre y TTY indica el terminal asociado (no tienen ya que son demonios) UUUUUuuuu

Actividad 4.4. Cuestiones sobre `at`

El proceso nuevo que se lanza al cumplirse el tiempo que se especificó en la orden `at`...

a) ¿qué directorio de trabajo tiene inicialmente? ¿hereda el que tenía el proceso que invocó a `at` o bien es el `home`, directorio inicial por omisión?

b) ¿qué máscara de creación de archivos `umask` tiene? ¿es la heredada del padre o

la que se usa por omisión?

c) ¿hereda las variables locales del proceso padre?

a) Se hereda el que tiene el proceso padre ya que el directorio de salida de las órdenes es el mismo

b) Ejecutamos umask cuyo valor es 0002. También la hereda

c) El directorio de trabajo, el entorno (excepto para las variables TERM, DISPLAY .. y el umask son tomados del momento de invocación, y no el de ejecución.

Actividad 4.5. Relación padre-hijo con órdenes ejecutadas mediante at

El proceso nuevo que se lanza al cumplirse el tiempo que se especificó en la orden at.... ¿de quién es hijo? Investiga lanzando la ejecución retardada de un script que muestre la información completa sobre los procesos existentes y el pid del proceso actual; el script podría contener lo que sigue:

```
nombrearchivo=`date +%Y-%j-%T`  
ps -ef > $nombrearchivo  
echo Mi pid = $$ >> $nombrearchivo
```

Ejecutamos el script anterior y como vemos, el PID se hereda del proceso padre al hijo

Actividad 4.6. Script para orden at

Construye un script que utilice la orden find para generar en la salida estándar los archivos modificados en las últimas 24 horas (partiendo del directorio home y recorriéndolo en profundidad), la salida deberá escribirse el archivo de nombre "modificados_<año><día><hora>" (dentro del directorio home). Con la orden at provoques que se ejecute dentro de un día a partir de este momento.

```
#!/bin/bash  
  
nombre=modificados_`date +"%Y-%d-%r"`  
  
find /home -atime 1 > $nombre
```

Actividad 4.7. Trabajo con la orden batch

Lanza los procesos que sean necesarios para conseguir que exista una gran carga de trabajo para el sistema de modo que los trabajos lanzados con la orden batch no se estén ejecutando (puede simplemente construir un script que esté en un ciclo infinito y lanzarla varias veces en segundo plano). Utiliza las órdenes oportunas para manejar este conjunto de procesos (la orden jobs para ver los trabajos lanzados, kill para finalizar un trabajo, ...y tal vez también las órdenes fg, bg para pasar de segundo a primer plano y viceversa, <Ctrl-Z> para suspender el proceso en primer plano actual, etc). Experimenta para comprobar cómo al ir disminuyendo la carga de trabajos habrá un momento en que se ejecuten los trabajos lanzados a la cola batch.

```
#!/bin/bash  
  
valor=5  
while [ $valor -ne 0 ]  
do  
    ((i=i+1))
```

```
echo $i
done

#Lo lanzamos en segundo plano con: ./script &
```

Actividad 4.8. Utilización de las colas de trabajos de at

Construye tres script que deberás lanzar a las colas c, d y e especificando una hora concreta que esté unos pocos minutos más adelante (no muchos para ser operativos). Idea qué actuación deben tener dichos script de forma que se ponga de manifiesto que de esas colas la más prioritaria es la c y la menos es la e. Visualiza en algún momento los trabajos asignados a las distintas colas.

```
#Ejecutamos un script cualquiera en diferentes colas
at -q c now + 2 minutes
at -q d now + 2 minutes
at -q e now + 2 minutes
```

Para ver la prioridad, ejecutamos:

```
atq ó:
```

```
atq -q c
atq -q d
atq -q e
```

Actividad 4.9. Relación padre-hijo con órdenes ejecutadas mediante crontab

Al igual que se investigó en la Actividad 4.5 sobre quién es el proceso padre del nuestro, lanza el script construido en dicha actividad con una periodicidad de un minuto y analiza los resultados.

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
***** /Escritorio/SO/Ejercicios prácticas/sesión 4/ejercicio5
```

Actividad 4.10. Ejecución de scripts con crontab (I)

Construye un script que sea lanzado con una periodicidad de un minuto y que borre los nombres de los archivos que cuelguen del directorio /tmp/varios y que comiencen por “core” (cree ese directorio y algunos archivos para poder realizar esta actividad). Utiliza la opción -v de la orden rm para generar como salida una frase de confirmación de los archivos borrados; queremos que el conjunto de estas salidas se añadan al archivo /tmp/listacores. Prueba la orden crontab -l para ver la lista actual de trabajos (consulte la ayuda para ver las restantes posibilidades de esta orden para gestionar la lista actual de trabajos).

```
#!/bin/bash
***** rmdir -v /tmp/varios/core* >> /tmp/listacores
```

Actividad 4.11. Ejecución de scripts con crontab (II)

Para asegurar que el contenido del archivo /tmp/listacores no crezca demasiado, queremos que periódicamente se deje dicho archivo solo con sus 10 primeras líneas (puede ser de utilidad la orden head). Construye un script llamado reducelista (dentro del directorio ~/SO) que realice la función anterior y lance su ejecución con periodicidad de un minuto.

```
#En el script
#nombre=reducelista
head /tmp/listacores > /tmp/temporal
rm /tmp/listacores
mv /tmp/temporal /tmp/listacores

#En el archivo crontab
#!/bin/bash
***** ~/SO/reducelista
```

Actividad 4.12. Ejecución de scripts con crontab (III)

Construye un sencillo script que escriba en el archivo ~/SO/listabusqueda una nueva línea con la fecha y hora actual y después el valor de la lista de búsqueda, por ejemplo:

```
...
2011-297-12:39:10 - /usr/local/bin:/usr/local/bin:/usr/bin...
```

Ejecuta este script desde el lenguaje de órdenes y también lánzalo como trabajo crontab y compara los resultados, ¿se tiene en ambos casos la misma lista de búsqueda?

```
#!/bin/bash

***** echo $(date) echo "Lista búsqueda:" $PATH >> ~/SO/listabusqueda
```

Actividad 4.13. Variables de entorno en archivos crontab

Practicamos ahora lo que acabamos de explicar situándonos en lo que hemos realizado en la Actividad 4.11. Construye un script que generará un archivo crontab llamado crontab- reducelista que deberá contener...

... como primera línea la asignación a la variable PATH de la lista de búsqueda actual y además el directorio \$HOME/SO
... después la indicación a cron de la ejecución con periodicidad de 1 minuto del script reducelista

Una vez construido crontab-reducelista lánzalo con la orden crontab. Comprueba que con esta nueva lista de búsqueda podremos hacer alusión a reducelista especificando únicamente su nombre independientemente del directorio de trabajo en que nos situemos (no como ocurría en la Actividad 4.11 en que el directorio \$HOME/SO no estaba en la lista de búsqueda).

```
#!/bin/bash
echo "SHELL=/bin/sh" > crontab-reducelista
echo "PATH=`pwd`"/: $HOME/SO:"$PATH >> crontab-reducelista
echo "***** reducelista >> crontab-reducelista
```

#Ya está creado el script como pedían, ahora lo ejecutamos:
crontab crontab-reducelista

Actividad 4.14. Archivos crontab de diferentes usuarios

Vamos a lanzar un archivo crontab cuyo propietario es otro usuario. Visualiza el contenido del archivo /fenix/depar/lsi/so/ver-entorno y /fenix/depar/lsi/so/crontabver. Comprueba con ls -l que el propietario es el usuario lsi. Sin copiarlos, úsalos para lanzar la ejecución cada minuto del script /fenix/depar/lsi/so/ver-entorno. Analiza el archivo de salida: ¿de qué línea del archivo /etc/passwd se toman LOGNAME y HOME, de la línea del propietario del archivo crontab o de la línea del usuario que lanza el archivo crontab?

Actividad 4.15. Ejecución de scripts con crontab (IV)

El objetivo es ejecutar todos los días a las 0 horas 0 minutos una copia de los archivos que cuelguen de \$HOME que se hayan modificado en las últimas 24 horas. Vamos a programar este salvado incremental utilizando la orden find que usábamos en la Actividad 4.6; ahora queremos que se copien los archivos encontrados por find utilizando la orden cpio:

<orden find de la Actividad 4.6> | cpio -pmduv /tmp/salvado\$HOME