

Sistemas Operativos

Modulo 1: . Administración de Linux

Sesión 1: Herramientas de administración básicas

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
#Autor: Marlene Vásquez
#=====laboratorio=====
#Creamos un directorio en /tmp donde almacenar
#mkdir /tmp/UML
#Copiamos los archivos del servidor al directorio temporal
#cp /fenix/depar/lsi/UML/*.gz /tmp/UML
#Extraemos los archivos mediante la herramienta gunzip
#gunzip /tmp/UML/*.gz
#Cambiamos al directorio /tmp/UML
#cd /tmp/UML
#cambiamos los permisos
#chmod 700 kernel32-3.0.4
#Ejecutamos
#./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m
#=====Personal=====
echo "creando la carpeta tmp/UML"
mkdir /tmp/UML
echo "realizando la copia"
cp *.gz /tmp/UML
echo "realizando la extraccion de archivos"
gunzip /tmp/UML/*.gz
echo "Cambio de directorio /tmp/UML"
cd /tmp/UML
echo "ejecutando"
chmod 700 kernel32-3.0.4
./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m

#root sin contraseña
```

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.

/etc/default/useradd	/etc/login.defs.
cat /etc/default/useradd # useradd defaults file GROUP=100 HOME=/home INACTIVE=-1 EXPIRE=	cat /etc/login.defs # *REQUIRED* # Directory where mailboxes reside, _or_ name of file, relative to the # home directory. If you _do_ define both, MAIL_DIR takes precedence.

SHELL=/bin/bash SKEL=/etc/skel CREATE_MAIL_SPOOL=yes	<pre> # QMAIL_DIR is for Qmail # #QMAIL_DIR Maildir MAIL_DIR /var/spool/mail #MAIL_FILE .mail # Password aging controls: # # PASS_MAX_DAYSMaximum number of days a password may # be used. # PASS_MIN_DAYSMinimum number of days allowed between # password changes. # PASS_MIN_LEN Minimum acceptable password length. # PASS_WARN_AGENumber of days warning given before a # password expires. # PASS_MAX_DAYS99999 PASS_MIN_DAYS0 PASS_MIN_LEN 5 PASS_WARN_AGE7 # # Min/max values for automatic uid selection in useradd # UID_MIN 500 UID_MAX 60000 # # Min/max values for automatic gid selection in groupadd # GID_MIN 500 GID_MAX 60000 # # If defined, this command is run when removing a user. # It should remove any at/cron/print jobs etc. owned by # the user to be removed (passed as the first argument). # #USERDEL_CMD /usr/sbin/userdel_local # # If useradd should create home directories for users by # default # On RH systems, we do. This option is overridden with the # -m flag on # useradd command line. # CREATE_HOME yes # The permission mask is initialized to this value. If not # specified, # the permission mask will be initialized to 022. UMASK 077 # This enables userdel to remove user groups if no members # exist. # USERGROUPS_ENAB yes # Use SHA512 to encrypt password. ENCRYPT_METHOD SHA512 </pre>
--	--

/etc/passwd	/etc/group
cat /etc/passwd root::0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin [...] user"/var/empty/saslauth:/sbin/nologin sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin juan:x:500:500::/home/juan:/bin/bash mar:x:501:501::/home/mar:/bin/bash	cat /etc/group root:x:0:root bin:x:1:root,bin,daemon daemon:x:2:root,bin,daemon sys:x:3:root,bin,adm adm:x:4:root,adm,daemon tty:x:5: [...] sshd:x:74: mailnull:x:47: smmsp:x:51: juan:x:500: mar:x:501:

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.

```
(1) useradd -d /home/prueba -m -s /bin/sh -g 501 aux
(2) useradd -d /home/prueba2 -m -s /bin/bash aux2
```

```
vi new_user
mar:x:501:501::/home/mar:/bin/bash
aux4:x:507:507::/home/prueba4:/bin/bash
newusers new_user
```

- (2) Elimina alguno de ellos y comprueba que “rastros” ha dejado la cuenta recién eliminada en el sistema.

```
userdel mar
- Ha dejado su carpeta home personal -> /home/mar
```

- (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.

```
su juan
ls /home/juan/

ls -a /home/juan/
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc
```

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
root::0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
[...]
smtsp:x:51:51::/var/spool/mqueue:/sbin/nologin
juan:x:500:500::/home/juan:/bin/bash
```

El formato de línea es :

```
juan: Nombre de la cuenta (Login)
x: Clave de acceso encriptada (password)
500: UID de esta cuenta
500: GID del grupo principal al que pertenece la cuenta
/home/juan: Directorio de trabajo del usuario
/bin/bash: Interpretador de comandos (shell) de usuario
```

```
ls -lai /etc/passwd
14313 -rw-r--r-- 1 root root 919 Oct  3 13:55 /etc/passwd
```

```
ls -lF /etc/passwd
-rw-r--r-- 1 root root 919 Oct  3 13:55 /etc/passwd
```

- El propietario es root y tiene permisos de lectura y escritura para el propio

```
usuario(root), lectura para el grupo al que pertenece y lectura para el resto de usuarios.
```

5. Archivo /etc/shadow

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

```
cat /etc/shadow
```

- No se puede desde otro usuario dado que ese archivo tiene los siguientes permisos:

```
ls -lF /etc/shadow
```

```
-rw-r----- 1 root shadow 1238 2012-10-11 11:18 /etc/shadow
```

```
marlen@mar:~$ ls -lF /etc/shadow
-rw-r----- 1 root shadow 1276 oct  3 13:52 /etc/shadow
marlen@mar:~$ _
```

-root tiene permisos de lectura y escritura

-el grupo al que pertenece root tiene permiso solo de lectura.

-los demás no pueden acceder al archivo ya que no tienen permisos.

en la mv el archivo /etc/shadow no tiene permisos.

```
aux2:x:501:500:1:/home/prueba2:/bin/sh
aux4:x:504:504:1:/home/prueba4:/bin/bash
juan:x:505:505:1:/home/juan:/bin/bash
[root@localhost ~]# su juan
[juan@localhost root]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[juan@localhost root]$ ls -lF /etc/shadow
----- 1 root root 795 Oct 10 08:08 /etc/shadow
[juan@localhost root]$ _
```

6. Creación de grupos

1. Crea un par de grupos y asignáelos a algunos de los usuarios de tu sistema.

```
groupadd grupo1
```

```
gpasswd -a juan grupo1
```

```
[root@localhost ~]# groupadd grupo1
[root@localhost ~]# gpasswd -a juan grupo1
Adding user juan to group grupo1
```

```
groupadd grupo2
```

```
gpasswd -a aux4 grupo2
```

```
[root@localhost ~]# groupadd grupo2
[root@localhost ~]# gpasswd -a aux4 grupo2
Adding user aux4 to group grupo2
[root@localhost ~]# _
```

2. ¿Qué información devuelve la orden id si estás conectado como root?

```
id
```

```
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

```
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@localhost ~]# _
```

7. Archivos 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

```
>
```

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?

```
El directorio donde podríamos guardar información y que no se borre entre arranques del sistema es el directorio /var/tmp
```

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?

```
- El archivo /etc/fstab se encuentra comúnmente en sistemas Unix como parte de la configuración del sistema. Lo más destacado de este fichero es la lista de discos y particiones disponibles. En ella se indican cómo montar cada dispositivo y qué configuración utilizar.
```

```
- El archivo /etc/mtab es un archivo de información del sistema, comúnmente en Unix. Este archivo lista todos los sistemas de ficheros montados actualmente junto con sus opciones de inicialización. mtab tiene mucho en común con fstab, pero la diferencia principal es que este último enumera todos los sistemas de archivos disponibles.
```

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.

```
# /etc/fstab
#
LABEL=ROOT / auto noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
tmp /tmp tmpfs rw,mode=1777,fscontext=system_u:object_r:tmp_t:s0
0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

```
<file system> <mount point> <type> <options> <dump> <pass>
```

```
- El primer campo nos indica el archivo de sistema de dispositivo especial remoto a ser montado, que en este caso es LABEL=ROOT.
```

```
- El segundo campo nos va a indicar el punto de montaje para el archivo de sistema.
```

```
- El tercer campo describe el tipo de archivo del sistema.
```

```
- El cuarto nos indica la cantidad de opciones asociados al archivo del sistema.
```

```
- El quinto campo es usado por los archivos de sistema por el comando de volcado.
```

```
- El sexto campo es usado por el fsck para determinar el orden en que los archivos del sistema son analizados como correctos a la hora de reiniciar.
```

11. Archivos de Información para los Sas

Compara la información que contienen los cuatro archivos de texto que se han presentado en este apartado (/etc/fstab, /etc/mtab, /proc/filesystems y /proc/mounts). Describe en un párrafo para qué te sirve la información que registra cada archivo.

```
- El archivo /etc/fstab se encuentra comúnmente en sistemas Unix como parte de la configuración del sistema. Lo más destacado de este fichero es la lista de discos y particiones disponibles. En ella se indican cómo montar cada dispositivo y qué configuración utilizar.

- El archivo /etc/mtab es un archivo de información del sistema, comúnmente en Unix. Este archivo lista todos los sistemas de ficheros montados actualmente junto con sus opciones de inicialización. mtab tiene mucho en común con fstab, pero la diferencia principal es que este último enumera todos los sistemas de archivos disponibles.

El archivo /proc/filesystems muestra una lista de los tipos del sistema de archivos soportados actualmente por el kernel.

El archivo /proc/mounts proporciona una lista de todos los montajes en uso por el sistema. La salida de datos que encontramos aquí se parece a /etc/mtab , excepto que /proc/mounts está más actualizada
```

Sesión 2: Herramientas de administración del SA

1. Partición de un dispositivo: “USB pen drive” o “memory stick”

A) Preparación previa a la partición de un dispositivo simulado mediante un archivo especial de dispositivo.

Utilizar un dispositivo simulado mediante un archivo /dev/loop?.

(a) Crea los archivos /dev/loop0 y /dev/loop1, si no se encuentran en el sistema, utilizando las siguientes órdenes:

```
mknod /dev/loop0 b 7 0
mknod /dev/loop1 b 7 1
```

(b) Crea un archivo de 20 MB y otro de 30 MB en tu sistema de archivos con las siguientes órdenes:

```
dd if=/dev/zero of=/root/archivo_SA20 bs=2k count=10000
dd if=/dev/zero of=/root/archivo_SA30 bs=3k count=10000
```

(c) Ahora vamos a asociar un archivo de dispositivo loop a cada uno de los archivos que acabas de crear. De esta forma el “disco virtual” que representa el archivo pasará a estar asociado al archivo de dispositivo /dev/loop0 y /dev/loop1. Para ello ejecuta las siguientes órdenes:

```
losetup /dev/loop0 /root/archivo_SA20
losetup /dev/loop1 /root/archivo_SA30
```

```
[root@localhost ~]# mknod /dev/loop0 b 7 0
[root@localhost ~]# mknod /dev/loop1 b 7 1
[root@localhost ~]# dd if=/dev/zero of=/root/archivo_SA20 bs=2k count=10000
10000+0 records in
10000+0 records out
20480000 bytes (20 MB) copied, 0.540702 s, 37.9 MB/s
[root@localhost ~]# dd if=/dev/zero of=/root/archivo_SA30 bs=3k count=10000
10000+0 records in
10000+0 records out
30720000 bytes (31 MB) copied, 0.6858 s, 44.8 MB/s
[root@localhost ~]# losetup /dev/loop0 /root/archivo_SA20
[root@localhost ~]# losetup /dev/loop1 /root/archivo_SA30
[root@localhost ~]# _
```

- (d) Puedes comprobar la configuración de tus “discos virtuales” mediante la siguiente orden que producirá como salida el siguiente resultado:

```
fdisk -l /dev/loop0 /dev/loop1
```

```
[root@localhost ~]# fdisk -l /dev/loop0 /dev/loop1

Disk /dev/loop0: 20 MB, 20480000 bytes
255 heads, 63 sectors/track, 2 cylinders, total 40000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/loop0 doesn't contain a valid partition table

Disk /dev/loop1: 30 MB, 30720000 bytes
255 heads, 63 sectors/track, 3 cylinders, total 60000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/loop1 doesn't contain a valid partition table
[root@localhost ~]#
```

Ahora podemos proceder a crear la tabla de particiones mediante **fdisk**,

```
fdisk /dev/loop0

dentro del menu: (n)

- creamos una partición primaria. (P)
  dejamos los valores por defecto
- añadimos una partición extendida o lógica (e)
  dejamos los valores por defecto.

fdisk /dev/loop1

dentro del menu: (n)

- creamos una partición primaria. (P)
  dejamos los valores por defecto
- añadimos una partición extendida o lógica (e)
  dejamos los valores por defecto.

#comprobamos
    fdisk -l /dev/loop0 /dev/loop1
```

2. Creación de sistemas de archivos

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

```
mke2fs -L LABEL_ext3 -t ext3 /dev/loop0
```

```
[root@localhost ~]# mke2fs -L LABEL_ext3 -t ext3 /dev/loop0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=LABEL_ext3
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
5016 inodes, 20000 blocks
1000 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=20709376
3 block groups
8192 blocks per group, 8192 fragments per group
1672 inodes per group
Superblock backups stored on blocks:
    8193

Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

```
mke2fs -L LABEL_ext4 -t ext4 /dev/loop1
```

```
[root@localhost ~]# mke2fs -L LABEL_ext4 -t ext4 /dev/loop1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=LABEL_ext4
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
7520 inodes, 30000 blocks
1500 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=30932992
4 block groups
8192 blocks per group, 8192 fragments per group
1880 inodes per group
Superblock backups stored on blocks:
    8193, 24577

Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 29 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

para ver la configuración

```
tune2fs -l /dev/loop1
```

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?

```
Tune2fs -C  
tune2fs -c
```

- (b) ¿Cómo podrías conseguir reservar para uso exclusivo de un usuario username un número de bloques del sistema de archivos?

```
Tune2fs -r <numero de bloques> <SA> <username>  
tune2fs -r 5 /dev/loop0 -u root
```

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:

- a) El SA etiquetado como LABEL_ext3 debe estar montado en el directorio /mnt/SA_ext3 y en modo de solo lectura.

```
- creación del directorio donde vamos a montar nuestro SA.  
mkdir /mnt/SA_ext3  
mount /dev/loop0 -r /mnt/SA_ext3
```

```
[root@localhost mnt]# mount /dev/loop0 /mnt/SA_ext3  
[75249.910000] EXT3-fs: barriers not enabled  
[75249.910000] kjournald starting. Commit interval 5 seconds  
[75249.910000] EXT3-fs (loop0): using internal journal  
[75249.910000] EXT3-fs (loop0): mounted filesystem with writeback data mode
```

- b) 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.

```
- creación del directorio donde vamos a montar nuestro SA.  
mkdir /mnt/LABEL_ext4  
mount /dev/loop1 -o dirsync /mnt/LABEL_ext4
```

```
[root@localhost mnt]# mkdir /mnt/LABEL_ext4  
[root@localhost mnt]# mount /dev/loop1 /mnt/LABEL_ext4  
[75499.810000] EXT4-fs (loop1): mounted filesystem with ordered data mode. Opts: (null)  
[root@localhost mnt]# mount /dev/loop1 /mnt/LABEL_ext4  
mount: /dev/loop1 already mounted or /mnt/LABEL_ext4 busy  
mount: according to mtab, /dev/loop1 is already mounted on /mnt/LABEL_ext4  
[root@localhost mnt]#
```

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 4.

```
vi /etc/fstab  
  
/dev/loop0 /mnt/SA_ext3 ext3 auto,ro 0 0  
/dev/loop1 /mnt/LABEL_ext4 ext4 auto,dirsync 0 0
```

```
# /etc/fstab
#
LABEL=ROOT                                /                auto          noatime        1 1
tmpfs                                       /dev/shm         tmpfs         defaults       0 0
tmp                                         /tmp             tmpfs         rw,mode=1777,fscontext=system_u:object_r:tmp_t:s0 0 0
devpts                                     /dev/pts         devpts        gids=3,mode=620 0 0
sysfs                                      /sys             sysfs         defaults       0 0
proc                                       /proc            proc          defaults       0 0
/dev/loop0                                /mnt/SA_ext3     ext3          auto,ro        0 0
/dev/loop1                                /mnt/LABEL_ext4 ext4          auto,dirsync   0 0
```

6. Repositorios de paquetes

Accede a los sitios web especializados que ofrecen software para sistemas operativos Linux y enumera las principales características de cada uno de ellos en base, por ejemplo, a si contiene software abierto y/o propietario, tipos de aplicaciones disponibles, tamaño del sitio en cuanto a la cantidad de software que mantiene, y otras características que considere interesantes.

7. Trabajo con el gestor de paquetes YUM

```
mkdir /mnt/paquetes
mount none /mnt/paquetes -t hostfs -o /home/marlen/A_SO/paquetes
ls /mnt/paquetes/
```

```
[root@localhost mnt]# mkdir /mnt/paquetes
[root@localhost mnt]# mount none /mnt/paquetes -t hostfs -o /home/marlen/A_SO/paquetes
[root@localhost mnt]# ls /mnt/paquetes/
DRBit-0.5.17-25.fc11.i586.rpm      glib-1.2.10-33.fc12.i686.rpm    quota-3.17-13.fc14.i686.rpm
DRBit-0.5.17-30.fc14.i686.rpm    kpartx-0.4.9-14.fc13.i686.rpm  sysstat-9.0.6-3.fc13.i686.rpm
device-mapper-1.02.54-3.fc14.i686.rpm  libnl-1.1-12.fc14.i686.rpm    tcp_wrappers-7.6-59.fc14.i686.rpm
device-mapper-libs-1.02.54-3.fc14.i686.rpm  parted-2.3-4.fc14.i686.rpm
[root@localhost mnt]#
```

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.

- (a) 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.

```
rpm -qi vim-minimal.i686
```

```
[root@localhost paquetes]# rpm -qli vim-minimal.i686
Name       : vim-minimal                Relocations: (not relocatable)
Version    : 7.3.056                   Vendor: Fedora Project
Release    : 1.fc14                     Build Date: Tue Nov 16 12:05:43 2010
Install Date: Wed Nov 24 15:33:29 2010   Build Host: x86-04.phx2.fedoraproject.org
Group      : Applications/Editors        Source RPM: vim-7.3.056-1.fc14.src.rpm
Size       : 723436                      License: Vim
Signature  : RSA/SHA256, Tue Nov 16 12:31:00 2010, Key ID 421caddb97a1071f
Packager   : Fedora Project
URL        : http://www.vim.org/
Summary    : A minimal version of the VIM editor
Description:
VIM (VIsual editor iMproved) is an updated and improved version of the
vi editor. Vi was the first real screen-based editor for UNIX, and is
still very popular. VIM improves on vi by adding new features:
multiple windows, multi-level undo, block highlighting and more. The
vim-minimal package includes a minimal version of VIM, which is
installed into /bin/vi for use when only the root partition is
present. NOTE: The online help is only available when the vim-common
package is installed.
/bin/ex
/bin/rvi
/bin/rview
/bin/vi
/bin/view
/etc/virc
```

- (b) Idem que el anterior pero mostrando únicamente los archivos de configuración que contiene el paquete.

```
rpm -qic vim-minimal.i686
[root@localhost paquetes]# rpm -qic vim-minimal.i686
Name       : vim-minimal                Relocations: (not relocatable)
Version    : 7.3.056                   Vendor: Fedora Project
Release    : 1.fc14                     Build Date: Tue Nov 16 12:05:43 2010
Install Date: Wed Nov 24 15:33:29 2010   Build Host: x86-04.phx2.fedoraproject.org
Group      : Applications/Editors        Source RPM: vim-7.3.056-1.fc14.src.rpm
Size       : 723436                      License: Vim
Signature  : RSA/SHA256, Tue Nov 16 12:31:00 2010, Key ID 421caddb97a1071f
Packager   : Fedora Project
URL        : http://www.vim.org/
Summary    : A minimal version of the VIM editor
Description:
VIM (VIsual editor iMproved) is an updated and improved version of the
vi editor. Vi was the first real screen-based editor for UNIX, and is
still very popular. VIM improves on vi by adding new features:
multiple windows, multi-level undo, block highlighting and more. The
vim-minimal package includes a minimal version of VIM, which is
installed into /bin/vi for use when only the root partition is
present. NOTE: The online help is only available when the vim-common
package is installed.
/etc/virc
```

- (c) 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.

```
rpm -q --whatrequires <nombre del paquete>

[root@localhost paquetes]# rpm -q --whatrequires udev.i686
no package requires udev.i686
[root@localhost paquetes]# rpm -q --whatrequires /mnt/paquetes/quota-3.17-13.fc14.i686.rpm
no package requires /mnt/paquetes/quota-3.17-13.fc14.i686.rpm
[root@localhost paquetes]# _
```

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

```
# para instalar este paquete es necesario instalar otros dos paquetes.
rpm -ivh /mnt/paquetes/libnl-1.1-12.fc14.i686.rpm
rpm -ivh /mnt/paquetes/tcp_wrappers-7.6-59.fc14.i686.rpm
rpm -ivvh /mnt/paquetes/quota-3.17-13.fc14.i686.rpm
```

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

```
rpm -ivh /mnt/paquetes/sysstat-9.0.6-3.fc13.i686.rpm
```

```
[root@localhost paquetes]# rpm -ivh /mnt/paquetes/sysstat-9.0.6-3.fc13.i686.rpm
warning: /mnt/paquetes/sysstat-9.0.6-3.fc13.i686.rpm: Header V3 RSA/SHA256 Signature, key ID e8e40fde: NOKEY
Preparing... ##### [100%]
1:sysstat ##### [100%]
[root@localhost paquetes]# rpm -evv sysstat
D: loading keyring from pubkeys in /var/lib/rpm/pubkeys/*.key
D: couldn't find any keys in /var/lib/rpm/pubkeys/*.key
D: loading keyring from rpmdb
D: opening db environment /var/lib/rpm/cdb:mpool:joinenv
D: opening db index /var/lib/rpm/Packages rdonly mode=0x0
D: locked db index /var/lib/rpm/Packages
D: opening db index /var/lib/rpm/Name rdonly mode=0x0
D: read h# 181 Header sanity check: OK
D: added key gpg-pubkey-97a1071f-4c49d6fe to keyring
D: Using legacy gpg-pubkey(s) from rpmdb
D: read h# 423 Header V3 RSA/SHA256 Signature, key ID e8e40fde: NOKEY
D: read h# 423 Header V3 RSA/SHA256 Signature, key ID e8e40fde: NOKEY
D: ===== --- sysstat-9.0.6-3.fc13 i686/linux 0x1
D: opening db index /var/lib/rpm/Requirename rdonly mode=0x0
D: ===== recording tsort relations
D: ===== tsorting packages (order, #predecessors, #successors, tree, depth)
D: 0 0 0 0 1 -sysstat-9.0.6-3.fc13.i686
D: erasing packages
```

```
rpm -evv sysstat
```

9. Sistema de cuotas para el sistema de archivos tipo ext3

- a) 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.

```
vi /etc/fstab
/dev/loop0 /mnt/SA_ext3 ext3 ro,usrquota 0 0

#debemos dar permisos de escritura
/dev/loop0 /mnt/SA_ext3 ext3 rw,usrquota 0 0
```

- b) 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>
```

```
mount -o remount /mnt/SA_ext3
```

- c) 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>
```

```
quotacheck -nm /mnt/SA_ext3
```

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

```
quotaon -a
# /etc/fstab
#
LABEL=ROOT / auto noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
tmp /tmp tmpfs rw,mode=1777,fsconten
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/loop0 /mnt/SA_ext3 ext3 rw,usrquota 0 0
/dev/loop1 /mnt/LABEL_ext4 ext4 dirsnc 0 0
~
~
~
~
~
"/etc/fstab" 11L, 456C written
[root@localhost ~]# mount -o remount /mnt/SA_ext3
[54209.510000] EXT3-fs (loop0): using internal journal
[root@localhost ~]# quotacheck -nm /mnt/SA_ext3/
quotacheck: Error checking device name: LABEL=ROOT
quotacheck: Cannot get device name for LABEL=ROOT
[root@localhost ~]# quotaon -a
quotaon: Error checking device name: LABEL=ROOT
quotaon: Cannot get device name for LABEL=ROOT
[root@localhost ~]# cd /mnt/SA_ext3/
[root@localhost SA_ext3]# ls -lai
total 23
  2 drwxr-xr-x  3 root root  1024 Oct 11 04:19 .
309 drwxr-xr-x  5 root root  4096 Oct 10 13:44 ..
 12 -rw-----  1 root root   6144 Oct 11 04:19 aquota.user
 11 drwx-----  2 root root 12288 Oct 10 13:42 lost+found
```

- e) 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

```
edquota -u juan
Disk quotas for user juan (uid 508):
Filesystem      blocks      soft      hard    inodes      soft      hard
/dev/loop0       56          0          0         14          0          0
~
```

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

edquota -t

```
edquota -t

Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period    Inode grace period
/dev/loop0       7days                  7days
~
```

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.

```
edquota -u juan
```

```
Disk quotas for user juan (uid 508):
Filesystem      blocks      soft      hard    inodes      soft      hard
/dev/loop0       56           0         0        14           0         0
```

```
edquota -u aux4
```

```
Disk quotas for user aux4 (uid 507):
Filesystem      blocks      soft      hard    inodes      soft      hard
/dev/loop0        0           0         0         0           0         0
```

```
Disk quotas for user aux4 (uid 507):
Filesystem      blocks      soft      hard    inodes      soft      hard
/dev/loop0       50           0         0        15           0         0
```

Sesión 3: Monitorización del sistema

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:

```
[root@localhost ~]# uptime
14:52:44 up 1:37, 1 user, load average: 0.00, 0.01, 0.03
[root@localhost ~]# w
14:52:47 up 1:37, 1 user, load average: 0.00, 0.01, 0.03
USER      TTY      FROM          LOGIN@      IDLE        JCPU       PCPU       WHAT
root      tty0      -              13:15       0.00s       1.10s      0.00s     w
[root@localhost ~]# _
```

uptime

```
14:52:44 up 1:37, 1 user, load average: 0.00, 0.01, 0.03
```

w

```
14:52:47 up 1:37, 1 user, load average: 0.00, 0.01, 0.03
USER      TTY      FROM          LOGIN@      IDLE        JCPU       PCPU       WHAT
root      tty0      -              13:15       0.00s       1.10s      0.00s     w
```

(a) ¿Cuánto tiempo lleva en marcha el sistema?

1:37min

(b) ¿Cuántos usuarios hay trabajando?

Hay 1 usuario

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

0,03s

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:

```
#!/bin/bash
# actividad 3.2
# echo "parametro : $*"
x=0
while [[ $x -lt $1 ]]; do
    let aux=823*746
    let x=x+1
done
echo "el valor de la variable es: $x"

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?

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1502	root	15	-5	3120	1108	976	R	76.3	0.1	0:18.71	prueba_proceso.
1503	root	20	0	3120	1108	976	R	22.7	0.1	0:12.47	prueba_proceso.
1501	root	34	14	3120	1112	976	R	1.0	0.1	0:06.70	prueba_proceso.

```
renice 14 1501 // aumenta la prioridad
renice -5 1502 // disminuye la prioridad
```

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

```
[root@localhost ~]# time ./prueba_proceso.sh 100000
el valor de la variable es: 100000

real    0m26.483s
user    0m21.240s
sys     0m5.230s
```

```
[root@localhost ~]# time ./prueba_proceso.sh 100000 &
[1] 1514
[root@localhost ~]# renice 14 1514
1514: old priority 0, new priority 14
[root@localhost ~]# el valor de la variable es: 100000

real    0m26.294s
user    0m21.180s
sys     0m5.000s
```



```
[root@localhost ~]# time ./prueba_proceso.sh 100000 &
[1] 1511
[root@localhost ~]# renice -5 1511
1511: old priority 0, new priority -5
[root@localhost ~]# el valor de la variable es: 100000

real    0m21.611s
user    0m17.510s
sys     0m4.080s
```

se observa que los procesos con NI negativos son mas prioritario (terminan antes)

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.

```
[root@localhost ~]# pstree
init--+-auditd---{auditd}
      |-crond
      |   |-login---bash---su---bash---su---bash---su---bash---su---bash---
pstree
      |   |-rsyslogd---2*[{rsyslogd}]
      |   |-sendmail
      |   `--sshd
[root@localhost ~]# ps
  PID TTY          TIME CMD
 1180 tty0      00:00:00 bash
 1221 tty0      00:00:00 su
 1240 tty0      00:00:00 su
 1242 tty0      00:00:00 bash
 1254 tty0      00:00:00 su
 1273 tty0      00:00:00 su
 1274 tty0      00:00:00 bash
 1518 tty0      00:00:00 ps

[root@localhost ~]# ps -Al
F S      UID        PID    PPID    C  PRI   NI  ADDR  SZ  WCHAN  TTY          TIME CMD
 4 S          0          1        0  0  80    0    721 queue_ ?          00:00:00 init
 1 S          0          2        0  0  80    0    kthrea ?          00:00:00 kthreadd
 1 S          0          3        2  0  80    0    run_ks ?          00:00:02 ksoftirqd/0
 1 S          0          4        2  0  80    0    worker ?          00:00:00 kworker/0:0
 1 S          0          5        2  0  80    0    call_u ?          00:00:00 kworker/u:0
 1 S          0          6        2  0  58   -    0 proces ?          00:00:00 rcu_kthread
 5 S          0          7        2  0 -40   -    0 sched_ ?          00:00:00 watchdog/0
[.....]
 5 S          0    1105          1  0  80    0   7367 hrtime ?          00:00:00 rsyslogd
 5 S          0    1130          1  0  80    0   2302 set_si ?          00:00:00 sshd
 1 S         51    1156          1  0  80    0   2984 pause ?          00:00:00 sendmail
 1 S          0    1167          1  0  80    0   947 hrtime ?          00:00:00 crond
 4 S          0    1179          1  0  80    0   926 wait  ?          00:00:00 login
 4 S          0    1180    1179  0  80    0   820 wait  tty0        00:00:00 bash
 4 S          0    1221    1180  0  80    0   813 wait  tty0        00:00:00 su
 4 S         508    1222    1221  0  80    0   822 wait  tty0        00:00:00 bash
 4 S          0    1240    1222  0  80    0   813 wait  tty0        00:00:00 su
 4 S          0    1242    1240  0  80    0   822 wait  tty0        00:00:00 bash
 4 S          0    1254    1242  0  80    0   813 wait  tty0        00:00:00 su
 4 S         500    1256    1254  0  80    0   822 wait  tty0        00:00:00 bash
 4 S          0    1273    1256  0  80    0   813 wait  tty0        00:00:00 su
 4 S          0    1274    1273  0  80    0   823 wait  tty0        00:00:00 bash
 1 S          0    1304          2  0  60 -20   -    0 loop_t ?          00:00:00 loop0
 1 S          0    1306          2  0  60 -20   -    0 loop_t ?          00:00:00 loop1
 1 S          0    1309          2  0  80    0    bdi_wr ?          00:00:00 flush-7:0
 1 S          0    1311          2  0  80    0    bdi_wr ?          00:00:00 flush-7:1
 1 S          0    1318          2  0  80    0    kjourn ?          00:00:00 kjournald
```


1	S	0	1322	2	0	80	0	-	0	kjourn ?	00:00:00	jbd2/loop1-8
1	S	0	1323	2	0	60	-20	-	0	rescue ?	00:00:00	ext4-dio-unwrit
4	R	0	1520	1274	0	80	0	-	656	- tty0	00:00:00	ps

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

Significa que no tiene asociada una terminal.

4. Estadísticas de recursos del sistema

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

```
[root@localhost ~]# mpstat
Linux 3.0.4 (localhost)      10/10/16      _i686_ (1 CPU)

16:07:47   CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
16:07:47   all     2.55    0.94    0.86    0.12     0.00    0.00    0.00    0.00   95.53

[root@localhost ~]# mpstat 2 5
Linux 3.0.4 (localhost)      10/10/16      _i686_ (1 CPU)

16:07:52   CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
16:07:54   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
16:07:56   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
16:07:58   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
16:08:00   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
16:08:02   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
Average:   all     0.00    0.00    0.00    0.00     0.00    0.00    0.00    0.00  100.00
```

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

Tal y como indica la columna **%irq**, correspondiente al porcentaje de CPU dedicado a interrupciones hardware, se ha dedicado un **0,00%**.

- b) ¿Y qué porcentaje en tratar interrupciones software?

Tal y como indica la columna **%soft**, correspondiente al porcentaje de CPU dedicado a interrupciones software, se ha dedicado un **0,00%**.

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

```
top - 16:12:01 up 2:56, 1 user, load average: 0.00, 0.01, 0.09
Tasks: 52 total, 1 running, 51 sleeping, 0 stopped, 0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1019628k total, 128084k used, 891544k free, 7212k buffers
Swap:      0k total,    0k used,    0k free, 98776k cached
```

Como podemos ver con la orden **top**, el total de memoria SWAP es de 0k, de los cuales en uso hay 0k y libres 0k .

5. Utilización de las órdenes free y watch

Explora las opciones de las que consta la orden free prestando especial atención a las diferentes unidades de medida según las que puede informar acerca de memoria. Además, compare los resultados con los obtenidos usando la orden watch.

free

```
[root@localhost ~]# free
              total        used        free      shared    buffers     cached
Mem:          1019628      128084      891544           0        7228      98788
-/+ buffers/cache:      22068      997560
Swap:           0           0           0
```

watch free

```
Mon Oct 10 16:24:46 2016
              total        used        free      shared    buffers     cached
Mem:          1019628      128348      891280           0        7228      98800
-/+ buffers/cache:      22320      997308
Swap:           0           0           0
```

free -s 4 //watch -n 4 free

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.

```
[root@localhost ~]# vmstat 2 20 >>vmstat.txt
[root@localhost ~]# cat vmstat.txt
procs -----memory----- ---swap-- ----io---- --system-- ----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs us sy id wa st
1  0       0 891420  7244  98828    0    0     5     7   91  11  3  1 96  0  0
0  0       0 891420  7244  98828    0    0     0     0   93   5  0  0 100  0  0
0  0       0 891420  7244  98828    0    0     0     0   92   1  0  0 100  0  0
0  0       0 891420  7252  98828    0    0     0    14   95   8  0  0 100  0  0
0  0       0 891420  7252  98828    0    0     0     0   93   1  0  0 100  0  0
0  0       0 891420  7252  98828    0    0     0     0   92   3  0  0 100  0  0
0  0       0 891420  7260  98828    0    0     0     6   92   6  0  0 100  0  0
0  0       0 891420  7260  98828    0    0     0     0   92   2  0  0 100  0  0
0  0       0 891420  7260  98828    0    0     0     0   92   2  0  0 100  0  0
0  0       0 891420  7268  98828    0    0     0     6   94   8  0  0 100  0  0
0  0       0 891420  7268  98828    0    0     0     0   93   2  0  0 100  0  0
0  0       0 891420  7268  98828    0    0     0     0   93   2  0  0 100  0  0
0  0       0 891420  7276  98828    0    0     0     6   94   6  0  0 100  0  0
0  0       0 891420  7276  98828    0    0     0     0   93   4  0  0 100  0  0
0  0       0 891420  7276  98828    0    0     0     0   92   1  0  0 100  0  0
0  0       0 891420  7284  98828    0    0     0     8   94   9  0  0 100  0  0
0  0       0 891420  7284  98828    0    0     0     0   93   1  0  0 100  0  0
0  0       0 891420  7284  98828    0    0     0     0   91   2  0  0 100  0  0
0  0       0 891420  7292  98828    0    0     0    12   95   9  0  0 100  0  0
0  0       0 891420  7292  98828    0    0     0     0   93   2  0  0 100  0  0
[root@localhost ~]# vmstat
procs -----memory----- ---swap-- ----io---- --system-- ----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs us sy id wa st
0  0       0 891420  7300  98828    0    0     5     7   91  11  3  1 96  0  0
[root@localhost ~]# _
```

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?

```
[root@localhost ~]# ls -la
total 50048
dr-xr-x---  2 root root    4096 Oct 10 16:33 .
dr-xr-xr-x 22 root root    4096 Oct 10 13:15 ..
-rw-----  1 root root     53 Sep 13  2011 .bash_history
-rw-r--r--  1 root root     18 Mar 30  2009 .bash_logout
-rw-r--r--  1 root root    176 Mar 30  2009 .bash_profile
-rw-r--r--  1 root root    176 Sep 22  2004 .bashrc
-rw-r--r--  1 root root    100 Sep 22  2004 .cshrc
-rw-r--r--  1 root root    129 Dec  3  2004 .tcshrc
-rw-r--r--  1 root root 20480000 Oct 10 13:42 archivo_SA20
-rw-r--r--  1 root root 30720000 Oct 10 13:43 archivo_SA30
-rw-r--r--  1 root root     75 Oct 10 13:22 new_user
-rwx-----  1 root root    161 Oct 10 15:05 prueba_proceso.sh
-rw-r--r--  1 root root   1821 Oct 10 16:33 vmstast.txt
-rw-r--r--  1 root root    472 Oct 10 13:58 ?
```

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.

```
ls -ltu <directorio>
```

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

```
ls -ltc <directorio>
```

```
[root@localhost ~]# echo $HOME
/root
[root@localhost ~]# ls -ltu $HOME
total 50016
-rw-r--r--  1 root root    1821 Oct 10 16:33 vmstast.txt
-rwx-----  1 root root    161 Oct 10 15:05 prueba_proceso.sh
-rw-r--r--  1 root root    472 Oct 10 13:58 ?
-rw-r--r--  1 root root 30720000 Oct 10 13:39 archivo_SA30
-rw-r--r--  1 root root 20480000 Oct 10 13:39 archivo_SA20
-rw-r--r--  1 root root     75 Oct 10 13:22 new_user
[root@localhost ~]# ls -ltc $HOME
total 50016
-rw-r--r--  1 root root    1821 Oct 10 16:33 vmstast.txt
-rwx-----  1 root root    161 Oct 10 15:05 prueba_proceso.sh
-rw-r--r--  1 root root    472 Oct 10 13:58 ?
-rw-r--r--  1 root root 30720000 Oct 10 13:43 archivo_SA30
-rw-r--r--  1 root root 20480000 Oct 10 13:42 archivo_SA20
-rw-r--r--  1 root root     75 Oct 10 13:22 new_user
```

Para más información sobre la orden ls consultar el manual Texinfo. Utiliza la orden:

info coreutils 'ls invocation'

9. Metadatos del sistema de archivos: df y du

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

- (a) 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.

```
# du /usr
136  /usr/lib/libuser
[.....]
4    /usr/local/etc
4    /usr/local/libexec
4    /usr/local/games
128  /usr/local
305048  /usr

# du -h /usr
136K  /usr/lib/libuser
12K   /usr/lib/coreutils
4.0K  /usr/lib/nss/saved
[.....]
4.0K  /usr/local/etc
4.0K  /usr/local/libexec
4.0K  /usr/local/games
128K  /usr/local
298M  /usr
```

- (b) ¿Cuántos inodos se están usando en la partición raíz? ¿Cuántos nuevos archivos se podrían crear en esta partición?

```
df -i /
```

```
[root@localhost ~]# df -i /
Filesystem          Inodes    IUsed    IFree IUse% Mounted on
LABEL=ROOT          65536    14811   50725   23% /
```

- los inodos que está usando la partición raíz es **65536**.
- Se podrían crear tantos archivos como inodos libres haya, en este caso **50725**.

- (c) ¿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.

```
[root@localhost ~]# du -sh /etc/
21M   /etc/
[root@localhost ~]# du -sh /var/
16M   /var/
[root@localhost ~]# du -sh /bin/
5.3M  /bin/
[root@localhost ~]# du -sh /usr/
298M  /usr/
[root@localhost ~]# du -sh /lib/
25M   /lib/
```

1. du -sh /etc
- El tamaño del directorio /etc es de 21M.
2. du -sh /var
- El tamaño del directorio /var es de 16M.
3. du -sh /bin
- El tamaño del directorio /bin es de 5.3M.
4. du -sh /usr
- El tamaño del directorio /usr es de 298M.

5. `du -sh /lib`
- El tamaño del directorio `/lib` es de 25M.

Se observa que el directorio `/usr` es el que más ocupa. Esto es debido a que los programas están instalados ahí.
`/bin` es el que menos ocupa ya que solo contiene ejecutables.

- (d) 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?

```
du -B 4 /etc
```

```
4096    /etc/yum.repos.d
5398528 /etc/
[root@localhost ~]#
```

- El directorio `/etc` tiene 5398528 bloques de tamaño 4 KB.
- El tamaño de bloque por omisión utilizado en el SA es de 1024 bytes (1K)

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`?

```
touch archivo.txt
touch target_hardLink2.txt
ln archivo.txt hardLink
ln target_hardLink2.txt hardLink2
ln -s archivo.txt softLink
```

```
[root@localhost D3]# ls -lai
total 28
14716 drwxr-xr-x 3 root root 4096 Oct 12 17:52 .
  311 dr-xr-x--- 7 root root 4096 Oct 11 11:23 ..
14721 drwxr-xr-x 2 root root 4096 Oct 12 17:52 D1
14717 -rw-r--r-- 2 root root    5 Oct 12 17:50 archivo.txt
14717 -rw-r--r-- 2 root root    5 Oct 12 17:50 hardLink
14718 -rw-r--r-- 2 root root    3 Oct 12 17:50 hardLink2
14719 lrwxrwxrwx 1 root root   11 Oct 11 11:24 softLink -> archivo.txt
14718 -rw-r--r-- 2 root root    3 Oct 12 17:50 target_hardLink2.txt
[root@localhost D3]#
```

lo del contador es porque solo se contabilizan los enlaces duros y no los simbólicos.

Extra:

enlaces con directorios

```
ln -d -s enlace_sim/ e_simbolico
```

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.

```
[root@localhost ~]# ls -laill
total 50072
 311 dr-xr-x---  3 root root      4096 Oct 10 18:38 .
   2 dr-xr-xr-x 22 root root      4096 Oct 10 13:15 ..
   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
14699 drwxr-xr-x  2 root root      4096 Oct 10 18:38 D1
14696 -rw-r--r--  2 root root         79 Oct 10 18:34 archivo.txt
14238 -rw-r--r--  1 root root    20480000 Oct 10 13:42 archivo_SA20
14239 -rw-r--r--  1 root root    30720000 Oct 10 13:43 archivo_SA30
14697 -rw-r--r--  2 root root        100 Oct 10 18:34 hardLin2
14696 -rw-r--r--  2 root root         79 Oct 10 18:34 hardLink
13061 -rw-r--r--  1 root root         75 Oct 10 13:22 new_user
14629 -rwx-----  1 root root        161 Oct 10 15:05 prueba_proceso.sh
14696 -rw-r--r--  2 root root         79 Oct 10 18:34 sofLink
14697 -rw-r--r--  2 root root        100 Oct 10 18:34 target_hardLink2.txt
14695 -rw-r--r--  1 root root       1821 Oct 10 16:33 vmstast.txt
14627 -rw-r--r--  1 root root        472 Oct 10 13:58 ?
```

- 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.

```
[root@localhost ~]# ls -laid
311 dr-xr-x--- 3 root root 4096 Oct 10 18:38 .
```

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.

```
dispositivo de caracteres c
mknod /dev/loop2 c 9 1

dispositivo de bloques b
mknod /dev/loop3 b 9 1
```



```
[root@localhost ~]# mknod /dev/loop2 c 9 1
[root@localhost ~]# mknod /dev/loop3 b 9 1
[root@localhost ~]# ls -li /dev/
total 12
 28 crw----- 1 root root 5, 1 Oct 10 13:15 console
 23 crw-rw-rw- 1 root root 1, 7 Nov 3 2010 full
14118 -rw-r--r-- 1 root root 54 Sep 13 2011 kmsg
 87 srw-rw-rw- 1 root root 0 Oct 10 13:15 log
 4543 brw-r--r-- 1 root root 7, 0 Oct 10 13:42 loop0
12616 brw-r--r-- 1 root root 7, 1 Oct 10 13:42 loop1
14700 crw-r--r-- 1 root root 9, 1 Oct 10 18:52 loop2
14701 brw-r--r-- 1 root root 9, 1 Oct 10 18:52 loop3
 21 drwxr-xr-x 2 root root 4096 Nov 3 2010 mapper
 22 crw-rw-rw- 1 root root 1, 3 Nov 3 2010 null
 29 crw-rw-rw- 1 root root 5, 2 Nov 3 2010 ptmx
 1 drwxr-xr-x 2 root root 0 Oct 10 13:15 pts
 25 crw-rw-rw- 1 root root 1, 8 Nov 3 2010 random
 20 drwxr-xr-x 2 root root 4096 Nov 3 2010 shm
 32 lrwxrwxrwx 1 root root 15 Nov 3 2010 stderr -> /proc/self/fd/2
 30 lrwxrwxrwx 1 root root 15 Nov 3 2010 stdin -> /proc/self/fd/0
 31 lrwxrwxrwx 1 root root 15 Nov 3 2010 stdout -> /proc/self/fd/1
 27 crw-rw-rw- 1 root root 5, 0 Nov 3 2010 tty
 67 crw--w---- 1 root tty 4, 0 Oct 10 18:53 tty0
 26 cr--r--r-- 1 root root 1, 9 Nov 3 2010 urandom
 24 crw-rw-rw- 1 root root 1, 5 Nov 3 2010 zero
```

```
mknod disp_bloques b 9 5
mknod disp_caracteres c 9 5
ls -lai
```

```
[root@localhost d2]# mknod disp_bloques b 9 5
[root@localhost d2]# mknod disp_caracteres c 9 5
[root@localhost d2]# ls -lai
total 8
14628 drwxr-xr-x 2 root root 4096 Oct 11 04:57 .
 311 dr-xr-x--- 4 root root 4096 Oct 11 04:52 ..
14702 brw-r--r-- 1 root root 9, 5 Oct 11 04:57 disp_bloques
14705 crw-r--r-- 1 root root 9, 5 Oct 11 04:57 disp_caracteres
[root@localhost d2]# _
```

Sesión 4: Automatización de tareas

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.

```
#arrancar atd
service atd restart
ps aux|grep atd
ps -p 2804 -f
```

```
[root@localhost ~]# service atd restart
Stopping atd: [ OK ]
[ OK ] atd: [ OK ]
[root@localhost ~]# ps aux |grep atd
root      2804  0.0  0.0  2872  432 ?        Ss   02:05   0:00 /usr/sbin/atd
root      2807  0.0  0.0  2340  564 tty0     R+   02:05   0:00 grep --color=auto atd
[root@localhost ~]# ps -p 2804 -f
UID      PID  PPID  C  STIME TTY          TIME CMD
root      2804    1  0  02:05 ?        00:00:00 /usr/sbin/atd
[root@localhost ~]# _
```

padre → 1(init)

```
usuario→ root
terminal→ ?
```

```
ps aux|grep cron
ps -p 1167 -F
```

```
[root@localhost bloques]# ps aux|grep cron
root      1167    0.0  0.1   3820   1204 ?        Ss   Oct10    0:01  cron
root      2662    0.0  0.0   2340    560 tty0    R+   13:29    0:00  grep --color=auto cron
[root@localhost bloques]# ps -p 1167 -F
UID        PID  PPID  C   SZ   RSS  PSR  STIME TTY          TIME CMD
root       1167    1    0   955  1204    0  Oct10 ?           00:00:01 cron
[root@localhost bloques]# _
```

```
padre → 1(init)
usuario→ root
terminal→ ?
```

2. Ejecución postergada de órdenes con at (I)

Crea un archivo genera-apunte que escriba la lista de hijos del directorio home en un archivo de nombre listahome-`date +%Y-%j-%T-\$\$`, es decir, la yuxtaposición del literal “listahome” y el año, día dentro del año, la hora actual y pid (consulte la ayuda de date). Lanza la ejecución del archivo genera-apunte un minuto más tarde de la hora actual. ¿En qué directorio se crea el archivo de salida?

```
[root@localhost script]# ls
genera_apunte.sh
[root@localhost script]# date
Mon Oct 17 02:34:03 EDT 2016
[root@localhost script]# at -f genera_apunte.sh now+1min
job 3 at Mon Oct 17 02:35:00 2016
[root@localhost script]# date
Mon Oct 17 02:35:19 EDT 2016
[root@localhost script]# ls
genera_apunte.sh listahome-2016-291-02:35:00-2850
[root@localhost script]# _
```

#consultar la hora date

!/bin/bash

ls ~/ > listahome-`date +%Y-%j-%T-\$\$`

at -f genera_apunte.sh now+1min

¿En qué directorio se crea el archivo de salida?

Se crea en el directorio donde se ejecuta.

3. Ejecución postergada de órdenes con at (II)

Lanza varias órdenes at utilizando distintas formas de especificar el tiempo como las siguientes:

(será de utilidad la opción -v):

a) a medianoche de hoy

```
at -f genera_apunte.sh midnight
job 4 at Tue Oct 18 00:00:00 2016
```

b) un minuto después de la medianoche de hoy

```
at -f genera_apunte.sh midnight+1min
job 5 at Tue Oct 18 00:01:00 2016
```

c) a las 17 horas y 30 minutos de mañana

```
at -f genera_apunte.sh 17:30 tomorrow
job 6 at Tue Oct 18 17:30:00 2016
```

d) a la misma hora en que estemos ahora pero del día 25 de diciembre del presente año


```
at -f genera_apunte.sh Dec 25 2016
```

```
job 8 at Sun Dec 25 02:52:00 2016
```

e) a las 00:00 del 1 de enero del presente año

```
at -f genera_apunte.sh midnight Jan 01 2016
```

```
at: refusing to create job destined in the past
```

```
[root@localhost script]# date
Mon Oct 17 02:35:19 EDT 2016
[root@localhost script]# ls
genera_apunte.sh  listahome-2016-291-02:35:00-2850
[root@localhost script]# at -f genera_apunte.sh midnight
job 4 at Tue Oct 18 00:00:00 2016
[root@localhost script]# at -f genera_apunte.sh midnight+1min
job 5 at Tue Oct 18 00:01:00 2016
[root@localhost script]# at -f genera_apunte.sh 17:30 tomorrow
job 6 at Tue Oct 18 17:30:00 2016
[root@localhost script]# date
Mon Oct 17 02:49:52 EDT 2016
[root@localhost script]# at -f genera_apunte.sh Dec 25
job 7 at Sun Dec 25 02:51:00 2016
[root@localhost script]# at -f genera_apunte.sh Dec 25 2016
job 8 at Sun Dec 25 02:52:00 2016
[root@localhost script]# at -f genera_apunte.sh midnight Jan 01 2016
at: refusing to create job destined in the past
[root@localhost script]# at -f genera_apunte.sh midnight Jan 01 2015
at: refusing to create job destined in the past
[root@localhost script]#
```

Utiliza las órdenes atq y atrm para familiarizarte con su funcionamiento (consulta la ayuda de estas órdenes).

```
[root@localhost ~]# at -l
5      Tue Oct 18 00:01:00 2016 a root
4      Tue Oct 18 00:00:00 2016 a root
7      Sun Dec 25 02:51:00 2016 a root
6      Tue Oct 18 17:30:00 2016 a root
8      Sun Dec 25 02:52:00 2016 a root
[root@localhost ~]# at -d 8
[root@localhost ~]# at -l
5      Tue Oct 18 00:01:00 2016 a root
4      Tue Oct 18 00:00:00 2016 a root
7      Sun Dec 25 02:51:00 2016 a root
6      Tue Oct 18 17:30:00 2016 a root
```

at

-l lista los trabajos pendientes. **#atq**

-d elimina trabajo **#atrm**

4. Cuestiones sobre at

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

1. ¿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?

El directorio de trabajo se mantiene desde el momento de la invocación (según dice el manual de at), así que hereda el que tenía el proceso que invocó a at.

2. ¿qué máscara de creación de archivos umask tiene? ¿es la heredada del padre o la que se usa por omisión?

La máscara es 0022, se mantiene desde el momento de la invocación, la hereda.

3. ¿hereda las variables locales del proceso padre?

El entorno también se mantiene desde el momento de la invocación, excepto las variables **TERM**, **DISPLAY**, **SHELLOPTS**, **_**, **PPID**, **BASH_VERSINFO**, **EUID**, **UID**, **GROUPS**.

Experimenta con la orden at lanzando las órdenes adecuadas para encontrar las respuestas. (Puede encontrar información en la ayuda de at)

```
atc -c 5
[root@localhost ~]# at -c 5
#!/bin/sh
# atrun uid=0 gid=0
# mail root 0
umask 22
HOSTNAME=localhost; export HOSTNAME
HISTSIZE=1000; export HISTSIZE
USER=aux; export USER
```

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:

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

```
at -f ./actividad_4_5.sh now+1minute

UID          PID  PPID  C  STIME TTY          TIME CMD
[.....]
root         2804    1    0  02:05 ?           00:00:00 /usr/sbin/atd
root         2941   2804    0  03:23 ?           00:00:00 /usr/sbin/atd
root         2942   2941    0  03:23 ?           00:00:00 sh
root         2944   2942   99  03:23 ?           00:52:47 /bin/bash
root         3045   2804    0  04:16 ?           00:00:00 /usr/sbin/atd
root         3046   3045   18  04:16 ?           00:00:00 sh
root         3048   3046    0  04:16 ?           00:00:00 /bin/bash
root         3050   3048    0  04:16 ?           00:00:00 ps -ef
Mi pid = 3048

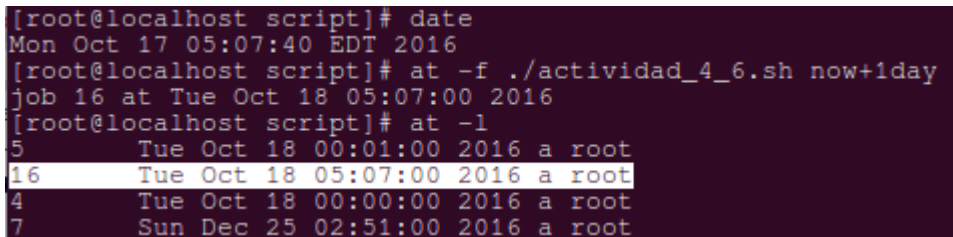
proceso actual PID 3048
Es hijo del proceso con PID 3046
```

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 provoque que se ejecute dentro de un día a partir de este momento.

```
#!/bin/bash
#Nombre: actividad_4.6.sh
find ~ -mtime 1 > modificados:`date +%Y-%j-%T`
-----
at -f ./actividad_4_6.sh now+1day
at -l
```



```
[root@localhost script]# date
Mon Oct 17 05:07:40 EDT 2016
[root@localhost script]# at -f ./actividad_4_6.sh now+1day
job 16 at Tue Oct 18 05:07:00 2016
[root@localhost script]# at -l
5          Tue Oct 18 00:01:00 2016 a root
16         Tue Oct 18 05:07:00 2016 a root
4          Tue Oct 18 00:00:00 2016 a root
7          Sun Dec 25 02:51:00 2016 a root
```

falta el volcado de documento generado

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.

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,

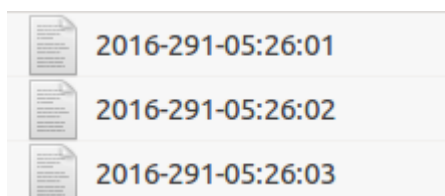
```
# lanzo el script de la actividad 5 en distintas colas
# para ello lanzo el siguiente script
-----
#!/bin/bash
#Nombre: actividad_4_8.sh
at -q c -f ./actividad_4_5.sh now+1 minute
at -q d -f ./actividad_4_5.sh now+1 minute
at -q e -f ./actividad_4_5.sh now+1 minute
-----
#compruebo las tareas que tiene at
at -l
# compruebo que tareas están en cada cola.
atq -q c
```

```

[root@localhost script]# at -l
5          Tue Oct 18 00:01:00 2016 a root
16         Tue Oct 18 05:07:00 2016 a root
4          Tue Oct 18 00:00:00 2016 a root
7          Sun Dec 25 02:51:00 2016 a root
[root@localhost script]# ./actividad_4_8.sh
job 20 at Mon Oct 17 05:26:00 2016
job 21 at Mon Oct 17 05:26:00 2016
job 22 at Mon Oct 17 05:26:00 2016
[root@localhost script]# at -l
20         Mon Oct 17 05:26:00 2016 c root
5          Tue Oct 18 00:01:00 2016 a root
16         Tue Oct 18 05:07:00 2016 a root
4          Tue Oct 18 00:00:00 2016 a root
22         Mon Oct 17 05:26:00 2016 e root
7          Sun Dec 25 02:51:00 2016 a root
21         Mon Oct 17 05:26:00 2016 d root
[root@localhost script]# atq -q c
20         Mon Oct 17 05:26:00 2016 c root
[root@localhost script]# atq -q d
21         Mon Oct 17 05:26:00 2016 d root
[root@localhost script]# atq -q e
22         Mon Oct 17 05:26:00 2016 e root

```

La creación de los archivos ejecutados por el script actividad_4_5.sh se realizan en el orden de prioridad.



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=/root/SO:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /root/SO/actividad_4_5.sh
-----

crontab actividad_4_9
-----
root      1167      1 0 02:44 ?          00:00:00 crond
root      1179      1 0 02:44 ?          00:00:00 login -- root
root      1180    1179 0 02:44 tty0        00:00:00 -bash
root      1219      2 0 02:46 ?          00:00:00 [loop0]
root      1221      2 0 02:46 ?          00:00:00 [loop1]
root      1257      1 0 02:47 ?          00:00:00 /usr/sbin/atd
root      1382      1 0 03:01 ?          00:00:00 /usr/sbin/anacron -s
root      1641    1167 0 03:37 ?          00:00:00 CROND
root      1642    1641 0 03:37 ?          00:00:00 /bin/sh -c /root/SO/actividad_4_5.sh
root      1644    1642 0 03:37 ?          00:00:00 ps -ef
Mi pid = 1642

```

el fichero se genera en la raíz que es donde se lanza el cron

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)

```
# script que crea los ficheros para realizar la actividad
-----
# !/bin/bash
cd /tmp/varios
find ~ -mtime 1 > core_`date +%Y-%j-%T`
-----
# script para borrar archivos que empiecen por core
-----
#!/bin/bash
rm -vf /tmp/varios/core* >> /tmp/listacores
-----
# añadimos una línea en el fichero actividad_4_10d para que el script
anterior se ejecute cada minuto
-----
SHELL=/bin/sh
PATH=/root/SO:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:
/root/bin

#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * ~/SO/actividad_4_10a.sh #crear ficheros
* * * * * ~/SO/actividad_4_10c.sh # borrar core*
-----
#lanzamos el cron
crontab actividad_4_10d
```

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**.

```
#script para controlar listacores reducelista.sh
-----
# !/bin/bash
#Nombre: reducelista.sh

head -10 /tmp/listacores >/tmp/aux
head -10 /tmp/aux > /tmp/listacores
rm -f /tmp/aux
-----
#añadimos una línea en el fichero actividad_4_10d para que se ejecute el
script reducelista.sh cada minuto.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /root/SO/actividad_4_10a.sh #crea ficheros
* * * * * /root/SO/actividad_4_10c.sh #borra ficheros
* * * * * /root/SO/reducelista.sh #reduce fichero listacores
-----
#lanzamos crontab

crontab actividad_4_10d
```

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 +%Y-%j-%T`+$PATH >>~/SO/listabusqueda
find ~/SO/*.sh >>~/SO/listabusqueda
```

```
2016-292-16:43:18+/sbin:/bin:/usr/sbin:/usr/bin
/root/SO/actividad_3_2.sh
/root/SO/actividad_4_10a.sh
/root/SO/actividad_4_10c.sh
/root/SO/actividad_4_12.sh
/root/SO/actividad_4_13a.sh
/root/SO/actividad_4_15a.sh
/root/SO/actividad_4_5.sh
/root/SO/actividad_4_6.sh
/root/SO/actividad_4_8.sh
/root/SO/genera_apunte.sh
/root/SO/prueba_proceso.sh
/root/SO/reducelista.sh
/root/SO/sincronizar.sh
2016-292-16:45:01+/root/SO:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
/root/SO/actividad_3_2.sh
/root/SO/actividad_4_10a.sh
```

Como podemos ver, la variable PATH es diferente según si lo ejecutamos directamente o con crontab, dado que la hemos redefinido dentro del fichero en formato crontab.

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 linea 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/sh
#Nombre: actividad_4_13a.sh (crea: crontab-reducelista)

echo "SHELL=/bin/sh" > ~/SO/crontab-reducelista
echo "PATH=$HOME/SO:$PATH">>~/SO/crontab-reducelista
echo "* * * * * reducelista.sh">>~/SO/crontab-reducelista
echo '-----'
echo '      Lanzando CRON'
echo '-----'
crontab ~/SO/crontab-reducelista
```

```

echo '===== crontab -l ====='
crontab -l
echo '-----'
echo 'contenido de crontab-reducelista'
echo '-----'
head ~/SO/crontab-reducelista

```

```

[root@localhost SO]# ./actividad_4_13a.sh
-----
Lanzando CRON
-----
===== crontab -l =====
SHELL=/bin/sh
PATH=/root/SO:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
* * * * * reducelista.sh
-----
contenido de crontab-reducelista
-----
SHELL=/bin/sh
PATH=/root/SO:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
* * * * * reducelista.sh

```

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?

```
cat /fenix/depar/lsi/so/ver-entorno
```

```

marlenevasquez@ei142134:~$ cat /fenix/depar/lsi/so/ver-entorno
nombre="salida"-`date +%Y-%j-%T-$$`
echo LOGNAME=$LOGNAME >> $nombre
echo HOME=$HOME>> $nombre
echo y ahora la orden set >> $nombre
set >> $nombre

```

```
cat /fenix/depar/lsi/so/crontabver
```

```

marlenevasquez@ei142134:~$ cat /fenix/depar/lsi/so/crontabver
* * * * * /fenix/depar/lsi/so/ver-entorno

```

```
ls -l
```

```

marlenevasquez@ei142134:~$ ls -l
total 22
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:46 Descargas
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Documentos
drwxr-xr-x 3 marlenevasquez alumno 2048 oct 18 15:50 Escritorio
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Imágenes
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Música
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Plantillas
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Público
-rw-r--r-- 1 marlenevasquez alumno 553 oct 18 16:27 salida-2016-292-16:27:01-4742
-rw-r--r-- 1 marlenevasquez alumno 553 oct 18 16:28 salida-2016-292-16:28:01-4750
-rw-r--r-- 1 marlenevasquez alumno 553 oct 18 16:29 salida-2016-292-16:29:01-4758
drwxr-xr-x 2 marlenevasquez alumno 2048 oct 18 15:36 Vídeos

```



```
ls -li <directorio del propietario de crontab>
```

```
marlenevasquez@ei142134:~$ ls -li /fenix/depar/lsi/so/
total 89652
165783844 -rwxr-xr-x 1 12909 lsi      42 oct 27  2011 crontabver
165996198 drwxr-xr-x 2 12909 lsi     4096 ene 19  2015 ma
165790522 drwxr-xr-x 2 12909 lsi     4096 oct 27  2011 paquetes
165781506 drwxr-xr-x 8 12909 lsi     4096 nov  5  2007 soi
165782852 drwxr-xr-x 9 12909 lsi     4096 abr  1  2008 soi
165783843 -rwxr-xr-x 1 12909 lsi      147 oct 27  2011 ver-entorno
165783845 -rw-r--r-- 1 12909 lsi 91681072 oct  4  2011 VirtualBox-4.1.4-74291-Win.exe
```

```
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
operador:x:300:1000:operador,,,:/home/operador:/bin/bash
marlenevasquez:x:6835:1500:MARLENE ELIZABETH VASQUEZ CALERO:/home/marlenevasquez:/bin/bash
marlenevasquez@ei142134:~$ cat salida-2016-292-16\:29\:01-4758
LOGNAME=marlenevasquez
HOME=/home/marlenevasquez
y ahora la orden set
HOME='/home/marlenevasquez'
IFS=''
```

conclusion:

LOGNAME y HOME lo toman de la linea del usuario que lanza el archivo crontab.

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

```
#script
-----
# !/bin/sh

find ~ -mtime 1 | cpio -pmduv /tmp/salvados$HOME
-----
#fichero crontab
-----
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
* * * * * /home/marlenevasquez/SO/actividad_4_15a.sh

#0 0 * * * ~/SO/actividad_4_15a.sh
-----

crontab actividad_4_15b
```


16. Gestión del servicio crond como usuario root (UML)

Prueba las siguientes operaciones sobre el demonio crond:

1. Como usuario root, deshabilita/habilita a un determinado usuario para que pueda utilizar el servicio cron; comprueba que efectivamente funciona.

→ deshabilitando

```
[root@localhost SO]# echo marlen >>/etc/cron.deny
[root@localhost SO]# su marlen
[marlen@localhost SO]$ ls -lai
total 84
14708 drwxr-xr-x 2 root root 4096 Oct 18 15:53 .
 311 dr-xr-x--- 6 root root 4096 Oct 18 04:36 ..
14744 -rw-r--r-- 1 root root 161 Oct 18 03:30 actividad_3_2.sh
14978 -rwx----- 1 root root 109 Oct 18 09:06 actividad_4_10a.sh
14776 -rw-r--r-- 1 root root 190 Oct 18 04:38 actividad_4_10b
14874 -rwx----- 1 root root 58 Oct 18 05:32 actividad_4_10c.sh
14891 -rw-r--r-- 1 root root 288 Oct 18 08:34 actividad_4_10d
14950 -rwx----- 1 root root 98 Oct 18 08:22 actividad_4_12.sh
15166 -rwx----- 1 root root 523 Oct 18 15:53 actividad_4_13a.sh
14952 -rwx----- 1 root root 62 Oct 18 09:23 actividad_4_15a.sh
15132 -rw-r--r-- 1 root root 118 Oct 18 15:27 actividad_4_15b
14747 -rwx----- 1 root root 104 Oct 18 03:30 actividad_4_5.sh
14748 -rw-r--r-- 1 root root 85 Oct 18 03:30 actividad_4_6.sh
14749 -rwx----- 1 root root 167 Oct 18 03:30 actividad_4_8.sh
14729 -rw-r--r-- 1 root root 188 Oct 18 03:47 actividad_4_9
14724 -rw-r--r-- 1 root root 125 Oct 18 15:53 crontab-reducelista
14752 -rw-r--r-- 1 root root 52 Oct 18 03:30 genera_apunte.sh
14954 -rw-r--r-- 1 root root 1092 Oct 18 08:40 listabusqueda
14753 -rwx----- 1 root root 168 Oct 18 03:30 prueba_proceso.sh
14865 -rwx----- 1 root root 99 Oct 18 07:58 reducelista.sh
14953 -rwx----- 1 root root 40 Oct 18 08:26 sincronizar.sh
[marlen@localhost SO]$ crontab actividad_4_10d
You (marlen) are not allowed to use this program (crontab)
See crontab(1) for more information
[marlen@localhost SO]$ _
```

→ habilitando

echo marlen >> /etc/cron.allow

2. Iniciar y terminar el servicio cron. Prueba las siguientes órdenes para iniciar y terminar este servicio:

Iniciar el servicio cron: **/sbin/service crond start**

Terminar el servicio cron: **/sbin/service crond stop**