

Sistemas Operativos

Sesión 3. Automatización de tareas



Sistemas Operativos Practica 3

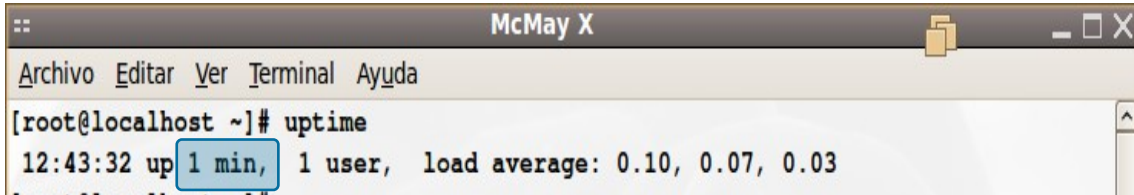
Actividad 3.1

[man time](#), [man uptime](#)

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:

- ¿cuánto tiempo lleva en marcha el sistema?

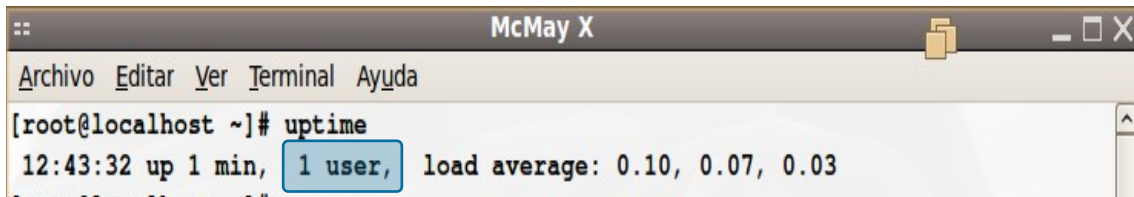
Según la imagen podemos observar que lleva 1 min.



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# uptime
12:43:32 up 1 min, 1 user, load average: 0.10, 0.07, 0.03
```

- ¿cuántos usuarios hay trabajando?

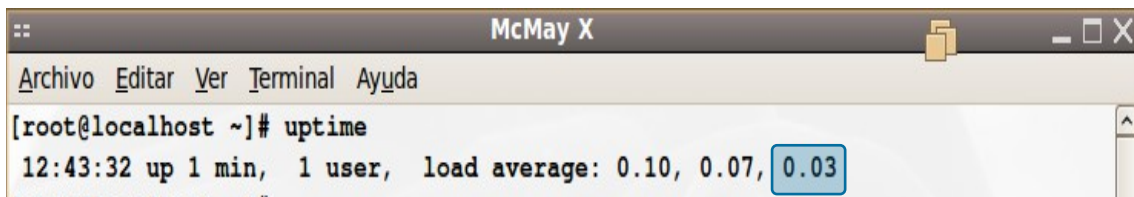
Según la imagen podemos observar que solo hay un usuario on.



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# uptime
12:43:32 up 1 min, 1 user, load average: 0.10, 0.07, 0.03
```

- ¿cuál es la carga media del sistema en los últimos 15 minutos?

Como podemos observar es de 0.03.



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# uptime
12:43:32 up 1 min, 1 user, load average: 0.10, 0.07, 0.03
```

Actividad 3.2

[man nice](#), [man renice](#).

- 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
```

Script:

```
#!/bin/bash
```

```
var=0;
```

```
for ((i=0; i<$1; i++))
```

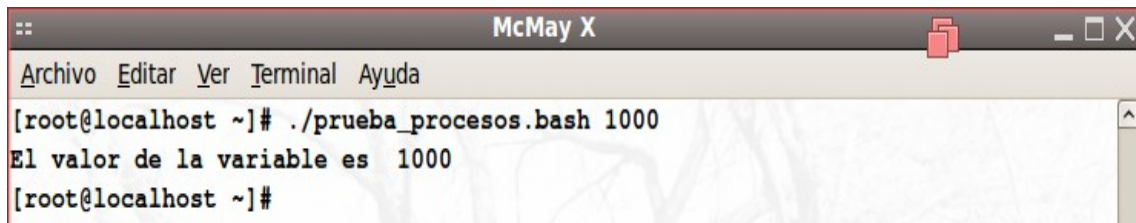
```
do
```

```
    let n=8*9
```

```
    ((var++))
```

```
done
```

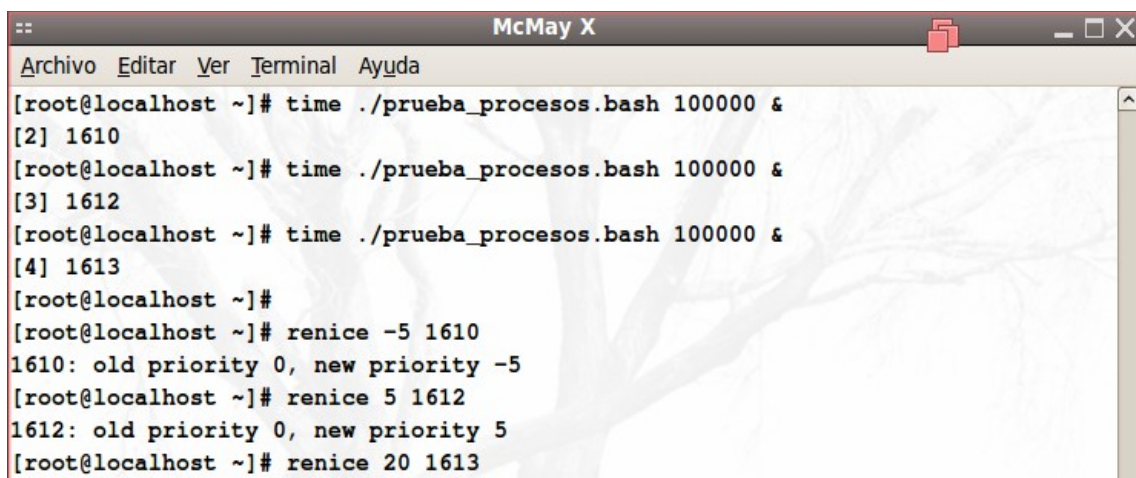
```
echo "el valor de la variable es " $var
```



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# ./prueba_procesos.bash 1000
El valor de la variable es 1000
[root@localhost ~]#
```

- 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?

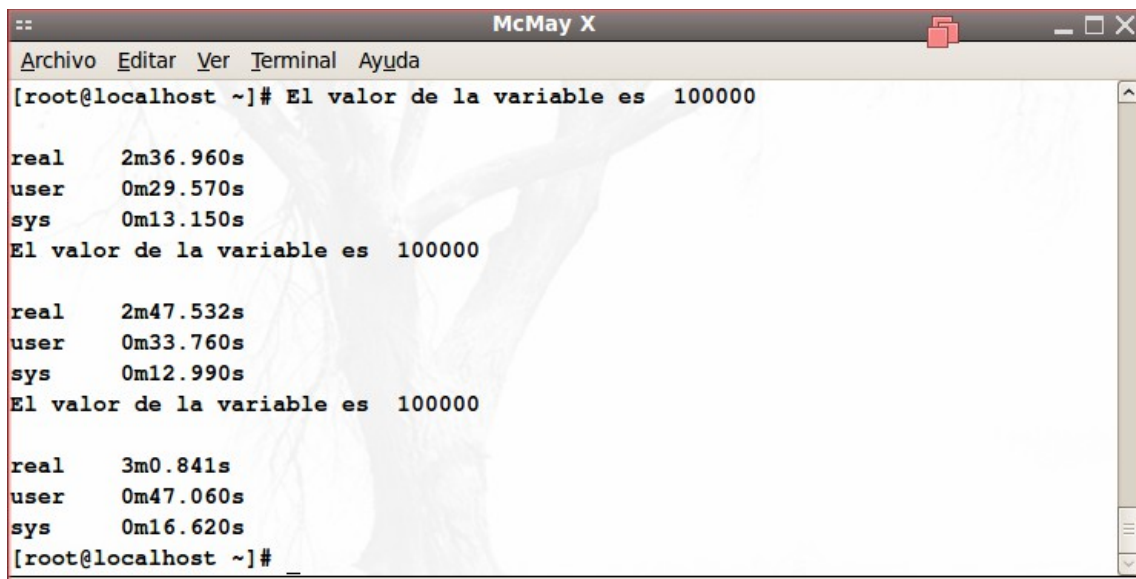
Para ejecutar un proceso en segundo plano añadimos al final de su inicialización un `&`, iniciamos varios procesos y anotamos su ID para poder reconocerlos posteriormente.



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# time ./prueba_procesos.bash 100000 &
[2] 1610
[root@localhost ~]# time ./prueba_procesos.bash 100000 &
[3] 1612
[root@localhost ~]# time ./prueba_procesos.bash 100000 &
[4] 1613
[root@localhost ~]#
[root@localhost ~]# renice -5 1610
1610: old priority 0, new priority -5
[root@localhost ~]# renice 5 1612
1612: old priority 0, new priority 5
[root@localhost ~]# renice 20 1613
```

Respuesta en el apartado C.

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



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# El valor de la variable es 100000

real    2m36.960s
user    0m29.570s
sys     0m13.150s
El valor de la variable es 100000

real    2m47.532s
user    0m33.760s
sys     0m12.990s
El valor de la variable es 100000

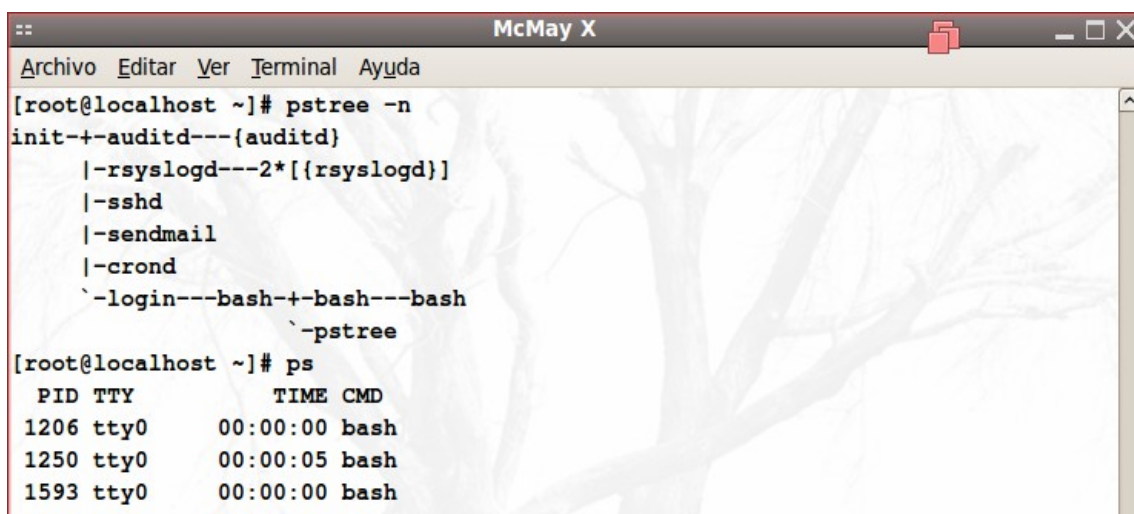
real    3m0.841s
user    0m47.060s
sys     0m16.620s
[root@localhost ~]#
```

Como vemos en los tiempos del sistema, en lo que afecta al sys es que cuanto mayor es el número de prioridad, más tiempo tarda en terminar y utiliza más tiempo de user como de sys.

Actividad 3.3

[man ps](#), [man pstree](#).

- 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.



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# pstree -n
init--+-auditd---{auditd}
      |-rsyslogd---2*[{rsyslogd}]
      |-sshd
      |-sendmail
      |-crond
      `--login---bash--+-bash---bash
                       `--pstree
[root@localhost ~]# ps
  PID TTY          TIME CMD
 1206 tty0        00:00:00 bash
 1250 tty0        00:00:05 bash
 1593 tty0        00:00:00 bash
```

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

```

[root@localhost ~]# ps -A
  PID TTY          TIME CMD
    1 ?           00:00:00 init
    2 ?           00:00:00 kthreadd
    3 ?           00:00:01 ksoftirqd/0
    4 ?           00:00:00 kworker/0:0

```

Que no hay ningún terminal asociado con el proceso.

Actividad 3.4

[man mpstat](#), [man top](#).

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

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

```

mcmay@mcmay-virtual:~$ mpstat 2 10
Linux 2.6.32-67-generic (mcmay-virtual)      23/10/14      _i686_  (4 CPU)

00:21:19   CPU   %usr   %nice    %sys %iowait  %irq   %soft  %steal  %guest   %idl
00:21:21   all    0,34    0,00    0,17    0,00    0,00    0,00    0,00    0,00   99,4
00:21:23   all    0,10    0,00    0,30    0,00    0,00    0,00    0,00    0,00   99,6
00:21:25   all    0,15    0,00    0,15    0,00    0,00    0,00    0,00    0,00   99,7
00:21:27   all    0,11    0,00    0,11    0,00    0,00    0,00    0,00    0,00   99,7
00:21:29   all    0,75    0,00    0,60    0,00    0,00    0,00    0,00    0,00   98,6
00:21:31   all    2,91    0,00    1,56    0,00    0,00    0,00    0,00    0,00   95,5
00:21:33   all    1,24    0,00    0,50    0,00    0,00    0,00    0,00    0,00   98,2
00:21:35   all    6,22    0,00    3,61    0,00    0,25    0,00    0,00    0,00   89,9
00:21:37   all   13,96    0,00   10,28    0,00    0,25    0,00    0,00    0,00   75,5
00:21:39   all    8,23    0,00    4,12    0,00    0,14    0,00    0,00    0,00   87,5
Media:     all    3,37    0,00    2,13    0,00    0,06    0,00    0,00    0,00   94,4
mcmay@mcmay-virtual:~$

```

La secuencia que utilizamos para visualizar las interrupciones hardware, software, etc, es `mpstat 2 10` para que cada exploración dure dos segundos y realice diez exploraciones.

Como podemos observar el tiempo medio de interrupciones hardware es de 0.06

- ¿y qué porcentaje en tratar interrupciones software?


```

mcmay@mcmay-virtual: ~ - McMay X
Archivo Editar Ver Terminal Ayuda
mcmay@mcmay-virtual:~$ mpstat 2 10
Linux 2.6.32-67-generic (mcmay-virtual)      23/10/14      _i686_ (4 CPU)

00:21:19  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:21:21  all     0,34    0,00    0,17    0,00    0,00    0,00    0,00    0,00   99,4
00:21:23  all     0,10    0,00    0,30    0,00    0,00    0,00    0,00    0,00   99,6
00:21:25  all     0,15    0,00    0,15    0,00    0,00    0,00    0,00    0,00   99,7
00:21:27  all     0,11    0,00    0,11    0,00    0,00    0,00    0,00    0,00   99,7
00:21:29  all     0,75    0,00    0,60    0,00    0,00    0,00    0,00    0,00   98,6
00:21:31  all     2,91    0,00    1,56    0,00    0,00    0,00    0,00    0,00   95,5
00:21:33  all     1,24    0,00    0,50    0,00    0,00    0,00    0,00    0,00   98,2
00:21:35  all     6,22    0,00    3,61    0,00    0,25    0,00    0,00    0,00   89,9
00:21:37  all    13,96    0,00   10,28    0,00    0,25    0,00    0,00    0,00   75,5
00:21:39  all     8,23    0,00    4,12    0,00    0,14    0,00    0,00    0,00   87,5
Media:    all     3,37    0,00    2,13    0,00    0,06    0,00    0,00    0,00   94,4
mcmav@mcmav-virtual:~$

```

Como podemos observar el tiempo medio de interrupciones software es de 0.00

– ¿cuánto espacio de swap está libre y cuánto ocupado?

```

McMay X
Archivo Editar Ver Terminal Ayuda
Tasks: 38 total, 1 running, 35 sleeping, 2 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1019408k total, 40496k used, 978912k free, 3676k buffers
Swap: 0k total, 0k used, 0k free, 18568k cached

```

En este caso ni tenemos libre ni ocupado ya que no hay espacio swap.

```

mcmay@mcmay-virtual: ~ - McMay X
Archivo Editar Ver Terminal Ayuda
top - 00:29:54 up 3:00, 3 users, load average: 0.46, 0.22, 0.13
Tasks: 221 total, 1 running, 209 sleeping, 11 stopped, 0 zombie
Cpu(s): 0.9%us, 0.7%sy, 0.0%ni, 98.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2061480k total, 804756k used, 1256724k free, 68892k buffers
Swap: 1764344k total, 0k used, 1764344k free, 455096k cached

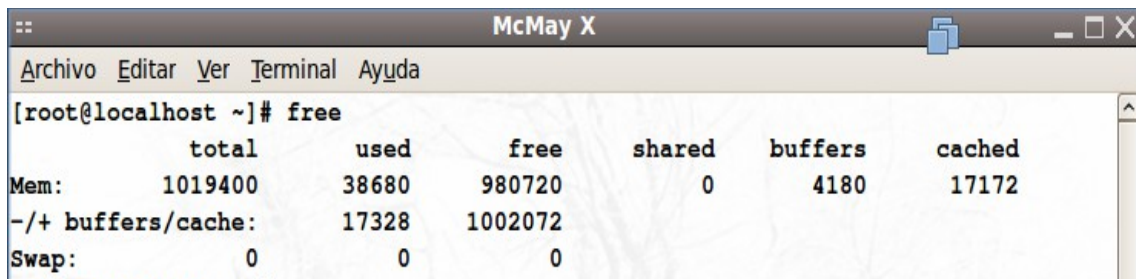
```

Pero en este caso en mi pc podemos observar como tengo 1764344k libres y usados 0.

Actividad 4. 1

[man free.](#) [man 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.



```

[root@localhost ~]# free
              total        used        free      shared    buffers     cached
Mem:          1019400       38680       980720           0         4180       17172
-/+ buffers/cache:        17328       1002072
Swap:           0           0           0

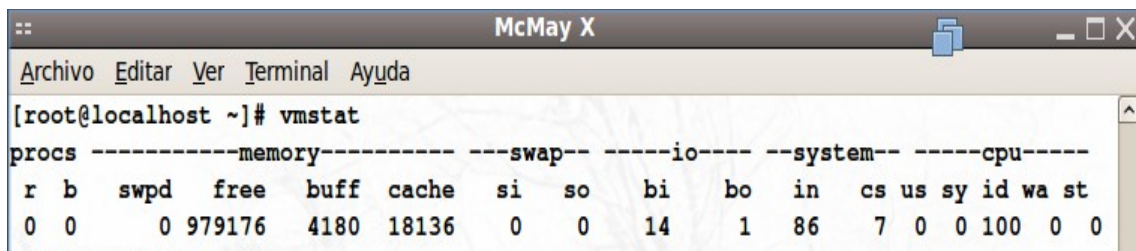
```

Cuando utilizamos la orden; `#watch free` , se refresca la información constantemente durante cada dos segundos, hasta que se finalice manualmente el proceso.

Actividad 4.2

[man ls](#), [man ln](#), [man df](#), [man du](#).

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
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 979176  4180  18136    0    0    14    1   86   7  0  0 100  0  0

```

Para guardarlo en un fichero;

```
#vmstat > vmstat.txt
```

Actividad 5.1

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?

```

McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# ls -lai
total 52
 311 dr-xr-x---  2 root root  4096 Oct 23 17:48 .
   2 dr-xr-xr-x 22 root root  4096 Oct 23 17:23 ..
  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
14239 -rw-r--r--  1 root root 12288 Oct 23 17:48 .vmstat.txt.swp
14238 -rwxrwxrwx  1 root root   113 Oct 22 17:11 prueba_procesos.bash
14175 -rw-r--r--  1 root root   245 Oct 23 17:47 vmstat.txt

```

```

root@mcmay-virtual: ~ - McMay X
Archivo Editar Ver Terminal Ayuda
root@mcmay-virtual:~# ls -ali
total 80
1572865 drwx----- 16 root root  4096 2014-04-16 04:35 .
   2 drwxr-xr-x 23 root root  4096 2014-10-13 17:20 ..
1573694 drwx-----  2 root root  4096 2014-04-16 04:34 .aptitude
1572867 -rw-r--r--  1 root root  3106 2010-04-23 11:45 .bashrc
1716715 drwx-----  4 root root  4096 2014-04-16 04:51 .config
1572889 drwx-----  3 root root  4096 2014-04-15 00:29 .dbus
1573696 drwxr-xr-x  2 root root  4096 2014-04-16 04:36 .debtags
1573687 drwxr-xr-x  2 root root  4096 2014-04-16 04:21 Desktop
1576419 drwx-----  3 root root  4096 2014-10-17 11:41 .gconf
1576420 drwx-----  2 root root  4096 2014-10-17 11:45 .gconfd
1572886 drwx-----  5 root root  4096 2014-04-16 04:23 .gnome2
1572887 drwx-----  2 root root  4096 2014-04-15 00:29 .gnome2_private
1573686 drwxr-xr-x  2 root root  4096 2014-04-16 04:21 .nautilus
1572868 -rw-r--r--  1 root root   140 2010-04-23 11:45 .profile
1572882 drwx-----  2 root root  4096 2014-04-14 23:49 .pulse
1572876 -rw-----  1 root root   256 2014-04-14 23:49 .pulse-cookie
1573691 -rw-----  1 root root  1400 2014-04-16 04:32 .recently-used.xbel
1572870 drwx-----  3 root root  4096 2014-10-17 12:45 .synaptic
1716710 drwx-----  3 root root  4096 2014-04-16 04:22 .thumbnails
1572866 drwxr-xr-x  2 root root  4096 2011-07-19 14:22 .wapi
root@mcmay-virtual:~#

```

Actividad 5.2 Ordenar un listado usando los campos de tiempos.

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.


```

[root@localhost ~]# ls -laiu
total 52
 311 dr-xr-x---  2 root root  4096 Apr  8 2011 .
   2 dr-xr-xr-x 22 root root  4096 Apr  8 2011 ..
  58 -rw-----  1 root root    53 Aug 31 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 Aug 31 2011 .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
14239 -rw-r--r--  1 root root 12288 Oct 23 17:48 .vmstat.txt.swp
14238 -rwxrwxrwx  1 root root   113 Oct 22 17:10 prueba_procesos.bash
14175 -rw-r--r--  1 root root   245 Oct 23 17:47 vmstat.txt
[root@localhost ~]#

```

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

```

[root@localhost ~]# ls -laic
total 52
 311 dr-xr-x---  2 root root  4096 Oct 23 17:48 .
   2 dr-xr-xr-x 22 root root  4096 Oct 23 17:23 ..
  58 -rw-----  1 root root    53 Sep 13 2011 .bash_history
3958 -rw-r--r--  1 root root    18 Nov  3 2010 .bash_logout
3959 -rw-r--r--  1 root root   176 Nov  3 2010 .bash_profile
3960 -rw-r--r--  1 root root   176 Nov  3 2010 .bashrc
3961 -rw-r--r--  1 root root   100 Nov  3 2010 .cshrc
3962 -rw-r--r--  1 root root   129 Nov  3 2010 .tcshrc
14239 -rw-r--r--  1 root root 12288 Oct 23 17:48 .vmstat.txt.swp
14238 -rwxrwxrwx  1 root root   113 Oct 22 17:11 prueba_procesos.bash
14175 -rw-r--r--  1 root root   245 Oct 23 17:47 vmstat.txt
[root@localhost ~]#

```

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

```
$> info coreutils 'ls invocation'
```

Como se puede apreciar en la imagen para el caso a utilizaremos el comando; `#ls -laiu` y en el caso b usaremos el comando; `#ls -laic`.

Actividad 5.3

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.

Con la orden

```
#du -c / , obtenemos el tamaño total de bloques, 377204.
```



```
48      /root
377204  /
377204  total
[root@localhost ~]#
```

Y con la orden

```
#du -ch /, obtenemos el tamaño total en megas, 369 Mb.
```



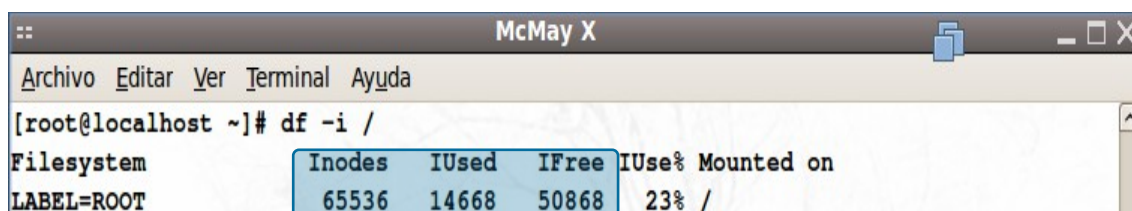
```
369M    /
369M    total
[root@localhost ~]#
```

Actividad 5.4

¿cuántos inodos se están usando en la partición raíz? ¿cuántos nuevos archivos se podrían crear en esta partición?

Introducimos el siguiente comando;

```
#df -i /, en uso hay 14668 y se podrían crear 65536
```



```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# df -i /
Filesystem          Inodes    IUsed    IFree IUse% Mounted on
LABEL=ROOT          65536    14668   50868   23% /
```

Introducimos el siguiente comando para obtenerlo en megas, en uso hay 15kb y se podrían crear 64kb.

Actividad 5.5

¿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.

Directorio /etc; introducimos la siguiente orden en el terminal,

```
#du -ch /etc, 21 mb.
```



```
21M total
[root@localhost ~]#
```

Directorio /var; introducimos la siguiente orden en el terminal,

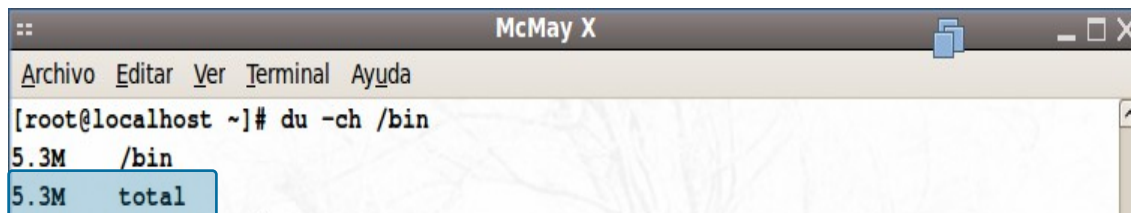
```
#du -ch /var, 14 mb.
```



```
14M total
[root@localhost ~]#
```

Directorio /bin; introducimos la siguiente orden en el terminal,

```
#du -ch /bin, 5.3 mb.
```



```
5.3M /bin
5.3M total
[root@localhost ~]#
```

Directorio /usr; introducimos la siguiente orden en el terminal,

```
#du -ch /usr, 297 mb.
```



```
297M /usr
297M total
[root@localhost ~]#
```

Directorio /lib; introducimos la siguiente orden en el terminal,

```
#du -ch /lib, 21 mb.
```



```
24M /lib
24M total
[root@localhost ~]#
```

Como se puede observar el más pesado es el de usr.

Actividad 5.6

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?

Por defecto tiene 1k de tamaño el bloque, introduciendo la siguiente orden, `# du -c --block-size=4K /etc`. Siendo un total de 5264.

```
4      /etc/yum.repos.d
5264   /etc
5264   total
[root@localhost ~]#
```

Actividad 5.7 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.

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

```
McMay X
Archivo Editar Ver Terminal Ayuda
[root@localhost ~]# ls -li
total 12
14241 drwxr-xr-x 2 root root 4096 Oct 23 19:14 D1
14248 -rw-r--r-- 2 root root  0 Oct 23 19:03 archivo.txt
14248 -rw-r--r-- 2 root root  0 Oct 23 19:03 hardLink
14249 -rw-r--r-- 2 root root  0 Oct 23 19:04 hardLink2
14238 -rwxrwxrwx 1 root root 113 Oct 22 17:11 prueba_procesos.bash
14251 lrwxrwxrwx 1 root root 11 Oct 23 19:17 softLink -> archivo.txt
14249 -rw-r--r-- 2 root root  0 Oct 23 19:04 target_hardLink2.txt
14175 -rw-r--r-- 1 root root 245 Oct 23 17:47 vmstat.txt
```

¿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`?

Actividad 5.8 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:

1. 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.

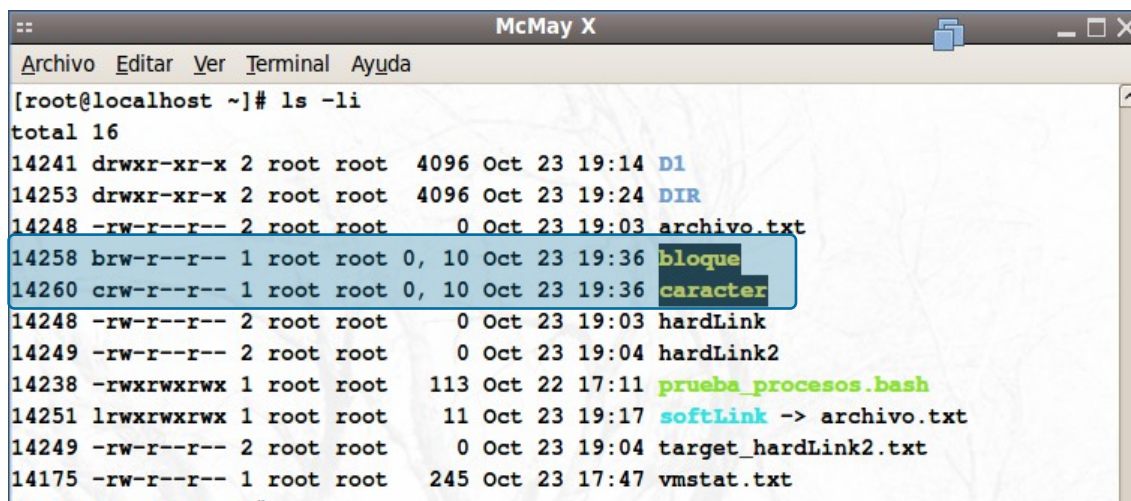
Actividad 5.9

[man mknod.](#)

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.

```
# mknod bloque b 0 10
```

```
# mknod caracter c 0 10
```



```

[root@localhost ~]# ls -li
total 16
14241 drwxr-xr-x 2 root root 4096 Oct 23 19:14 D1
14253 drwxr-xr-x 2 root root 4096 Oct 23 19:24 DIR
14248 -rw-r--r-- 2 root root 0 Oct 23 19:03 archivo.txt
14258 brw-r--r-- 1 root root 0, 10 Oct 23 19:36 bloque
14260 crw-r--r-- 1 root root 0, 10 Oct 23 19:36 caracter
14248 -rw-r--r-- 2 root root 0 Oct 23 19:03 hardLink
14249 -rw-r--r-- 2 root root 0 Oct 23 19:04 hardLink2
14238 -rwxrwxrwx 1 root root 113 Oct 22 17:11 prueba_procesos.bash
14251 lrwxrwxrwx 1 root root 11 Oct 23 19:17 softLink -> archivo.txt
14249 -rw-r--r-- 2 root root 0 Oct 23 19:04 target_hardLink2.txt
14175 -rw-r--r-- 1 root root 245 Oct 23 17:47 vmstat.txt

```