

### Actividad 1.1. Repaso de scripts de bash

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

```
#!/bin/bash
cp /fenix/depar/lsi/UML/*.gz /tmp
cd /tmp
gunzip *.gz
./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m
```

### Actividad 1.2. Valores por omisión para nuevas cuentas

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

```
$ cat /etc/default/useradd
$ cat /etc/login.defs
$ useradd prueba
$ cat /etc/passwd
$ cat /etc/group
```

### Actividad 1.3. Creación de usuarios

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

```
1)
$ useradd -d /home/diferente -m -s /bin/sh -g 1002 otro
$ useradd -d /home/pato -m -s /bin/bash pato
2)
$ userdel pato
- Ha dejado su carpeta home personal -> /home/pato
3)
$ su pato
$ ls /home/pato
```

Solo tiene un archivo -> examples.desktop. Tenemos que ver los archivos ocultos:

```
$ ls -a /home/pato
. .. .bash_logout
```

-

.bashrc  
examples.desktop  
.profile

#### **Actividad 1.4. Archivo /etc/passwd**

Vemos que tiene los archivos de configuración .bash\_logout, .bashrc y .profile.

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:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
pato:x:1003:1003:/home/pato:/bin/bash
superjes:x:1000:1000:SuperJes:/home/superjes:/bin/bash
```

El formato de línea es :

superjes: Nombre de la cuenta (Login)  
x: Clave de acceso encriptada (password)  
1000: UID de esta cuenta  
1000: GID del grupo principal al que pertenece la cuenta  
SuperJes: Nombre del usuario  
/home/superjes: Directorio de trabajo del usuario  
/bin/bash: Interprete de comando (shell) de usuario

```
$ ls -lF /etc/ | grep passwd
-rw-r--r--
1 root root
1795 2012-10-10 20:31 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.

#### **Actividad 1.5. Archivo /etc/shadow**

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

```
$ 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
```

-Tiene solamente permisos de lectura y escritura para root, y permiso de lectura para el grupo al que pertenece root. Para los demás no tiene ninguno, así que no se puede hacer nada.

### Actividad 1.6. Creación de grupos

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

```
$ groupadd grupo1  
$ gpasswd -a pato grupo1
```

Añadiendo al usuario pato al grupo grupo1

```
$ groupadd grupo2  
$ gpasswd -a otro grupo2
```

Añadiendo al usuario otro al grupo grupo2

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)
```

### Actividad 1.7. Archivo del kernel de Linux

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

```
$ find / -name Fedora  
#-> /temp/Fedora14-x86-root_fs
```

### Actividad 1.8. Organización del SA

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

En /var/tmp

### Actividad 1.9. Información de los SAs

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

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 actualmente los sistemas de ficheros montados 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.

### **Actividad 1.10 Información de los SAs**

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

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.

### **Actividad 1.11. 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 actualmente los sistemas de ficheros montados 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.

### **Actividad 2.1. Partición de un dispositivo: “USB pen drive” o “memory stick”**

En esta actividad utilizaremos un dispositivo USB de tipo “pen drive” para establecer particiones. Vamos a crear una tabla de particiones en las que se van a definir dos particiones primarias que se configuraran para albergar dos sistemas de archivos tipo linux, de manera que la primera albergará un SA `ext3` y la segunda un `ext4`. Ambas serán particiones tipo Linux `0x83`. El tamaño de las particiones queda a vuestra libre elección pero por lo menos deberían tener 512 MB.

El proceso de creación de las particiones es interactivo, pulsando `p` se listan todas las particiones creadas, con `n` creamos una nueva y a continuación nos da a elegir si ha de ser primaria o extendida, y el primer y último cilindro de la misma. Para salir pulsamos `w`. Para que tenga 512 MB tenemos que poner `+512M` en el último cilindro. Hemos de tener en cuenta si la partición que creamos es la segunda de empezar al final de la primera.

## Actividad 2.2. Creación de sistemas de archivos

El objetivo es simplemente formatear lógicamente las particiones creadas con anterioridad de forma consistente con el tipo de SA que se estableció que iba a ser alojado. En la primera partición crearemos un SA de tipo ext3 y en la segunda un ext4.

La orden que permite establecer un SA de los reconocidos dentro del sistema Linux sobre una partición de disco es mke2fs (consulta el manual en línea para familiarizarte con sus opciones). El resultado de la ejecución de esta orden es el formateo lógico de la partición escogida utilizando el SA que se ha seleccionado.

Utiliza el manual en línea para conocer cómo ejecutar la orden de creación de SA. mke2fs es la orden genérica para creación de sistemas de archivos. Como requisito es necesario que establezcas dos etiquetas de volumen para los SAs: LABEL\_ext3 para la primera partición y LABEL\_ext4 para la segunda. Debería aparecer un listado en pantalla similar al siguiente.

```
# mkfs.ext3 -L 'LABEL_ext3' /dev/loop0
# mkfs.ext4 -L 'LABEL_ext4' /dev/loop1
```

## Actividad 2.3. Personalización de los metadatos del SA

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

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

a) Con el modificador -C y un poniéndole un número más grande de veces que el sistema de archivos es montado que max-mount-counts .

b) \$ tune2fs /dev/loop0 -r [numero] -u username

## Actividad 2.4. Montaje de sistemas de archivos

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

- 3. El SA etiquetado como LABEL\_ext3 debe estar montado en el directorio /mnt/SA\_ext3 y en modo de solo lectura.
- 4. El SA etiquetado como LABEL\_ext4 debe estar montado en el directorio /mnt/LABEL\_ext4 y debe tener sincronizadas sus operaciones de E/S de modificación de directorios.

3.

```
$ mount /dev/loop0 -r /mnt/SA_ext3
```

## Actividad 2.5. Automontaje de Sistemas de Archivos

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

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

La opción auto se podría obviar, ya que por defecto se monta al arrancar el sistema todo lo que haya en el fichero /etc/fstab.

## Actividad 2.6. 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.

## Actividad 2.7. Trabajo con el gestor de paquetes YUM

Encuentra los archivos de configuración de YUM y explora las distintas órdenes disponibles en YUM ejecutándolas. En concreto, lista todos los paquetes instalados y disponibles, elimina el paquete instalado que te indique el profesor de prácticas, y a continuación vuelve a instalar el mismo paquete haciendo uso de los paquetes que se encuentran disponibles en /fenix/depar/lsi/so/paquetes. Para obtener acceso a este directorio del sistema de archivos anfitrión ejecute la siguiente orden de montaje una vez lanzado el sistema operativo User Mode Linux (UML):

Los archivos de configuración del “yum” se encuentran en el directorio /etc/yum/yum.conf

```
# yum install “paquete” → Instala la última versión del paquete indicado. Pide por confirmación.  
# yum -y install “paquete” → Instala la última versión del paquete sin pedir confirmación.  
# yum -y install “paquete1 paquete2” → Instala muchos paquetes a la vez. La opción -y es para que no pida confirmación.  
# yum -y update → Actualiza todos los paquetes del sistema.  
# yum check-update → Muestra una lista de paquetes que necesitan ser actualizados sin instalarlos.  
# yum info paquete → Descripción completa del paquete indicado.  
# yum list → Lista todos los paquetes disponibles para instalación, actualización o actualizados.  
# yum list installed → Lista todos los paquetes instalados en el sistema.  
# yum remove “paquete” → Eliminar un paquete instalado.  
# yum install “paquete” → Instalar un paquete.
```

## Actividad 2.8. Trabajo con el gestor de paquetes rpm

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

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

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

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

4. Instala el paquete quota que encontrarás en el directorio de software de la asignatura

(directorio que ya has montado en la Actividad 2.7).

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

Para mostrar cierta meta información acerca de uno o más paquetes ya instalados debemos ejecutar el siguiente comando:

```
# rpm -qi <nombre-paquete-instalado>
```

A continuación vamos a realizar una serie de operaciones: instalar, desinstalar y actualizar sobre un paquete específico que indicará el profesor (en este caso uno cualquiera):

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 -qli <nombre-paquete-instalado>
```

Idem que el anterior pero mostrando únicamente los archivos de configuración que contiene el paquete.

```
# rpm -q -c -i <nombre-paquete-instalado> // o "-c"?
```

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

```
# rpm -q --whatrequires <nombre-paquete-instalado>
```

El profesor de prácticas te indicará que instales un paquete que se encuentra accesible a través del directorio punto de montaje, tal como se ha indicado en la actividad anterior. Consiga que RPM muestre en pantalla la máxima información posible acerca del proceso de instalación de dicho paquete

```
# rpm -ivh </fenix/depar/lisi/so/paquetes/"nombre del paquete"> //ya lleva el -v en -ivh para ver la información
```

Desinstala el paquete del punto anterior mostrando en pantalla también la máxima información posible acerca del propio proceso de eliminación del paquete.

```
# rpm -e --nodeps -v <nombre del paquete>
```

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

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

1. Editar el archivo /etc/fstab y activar el sistema de cuotas de usuario para el SA tipo ext3.

Busca cómo se especifica esta opción en el manual en línea. Una ayuda para la búsqueda es que a realices sobre la orden mount y recuerdes que las opciones de montaje vienen especificadas

en los apartados: FILESYSTEM INDEPENDENT MOUNT OPTIONS y FILESYSTEM SPECIFIC MOUNT OPTIONS.

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

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

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

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

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

```
#> quotaon -a
```

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

```
#> edquota username
```

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

```
#> edquota -t
```

En la siguiente tabla se muestran algunas órdenes útiles para el administrador a la hora de trabajar con el sistema de control de cuotas.

1. Editar el archivo /etc/fstab y activar el sistema de cuotas de usuario para el SA tipo ext3. Busca como 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 bienen especificadas en los apartados: FILESYSTEM INDEPENDENT MOUNT OPTIONS y FILESYSTEM SPECIFIC MOUNT OPTIONS.

```
/dev/loop0 /mnt/SA_ext3 ext3 auto,ro,usrquota 0 0
```

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

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

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

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

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

```
# quotaon -a
```

5. Ahora solo falta activar la cuota para cada usuario del sistema mediante la siguiente orden. En este caso, implementa un script que automatice la tarea. Puede ser buena idea utilizar el archivo /etc/passwd.



```
#!/bin/bash
for [ linea in `cat /etc/passwd` ]
do
usuario = `cut -d : -f1`
setquota -u $usuario
done
```

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

```
# setquota -a -t 7 7
```

### **Actividad 2.10. Establecer límites sobre recursos de un SA**

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

```
# setquota -u pato -b nombre 7 10 7 10
```

### **Actividad 3.1. Consulta de estadísticas del sistema**

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

- ¿Cuánto tiempo lleva en marcha el sistema?
- ¿Cuántos usuarios hay trabajando?
- ¿Cuál es la carga media del sistema en los últimos 15 minutos?

```
# uptime
o
# w
```

### **Actividad 3.2. Prioridad de los procesos**

a) Crea un script o guión shell que realice un ciclo de un número variable de iteraciones en el que se hagan dos cosas: una operación aritmética y el incremento de una variable. Cuando terminen las iteraciones escribirá en pantalla un mensaje indicando el valor actual de la variable. Este guión debe tener un argumento que es el número de iteraciones que va a realizar. Por ejemplo, si el script se llama prueba\_procesos, ejecutaríamos:

```
# prueba_procesos 1000
el valor de la variable es 1000
```

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

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

```
#!/bin/bash
x=0
for i in `seq 1 $1`
do
sleep 1
x=`expr $x + 1`
done
echo "El valor de la variable es " $x
```

```
# ./prueba.sh 100 &  
# ./prueba.sh 150 &
```

```
# top  
- La prioridad inicial es de 20 para ambos.
```

```
# renice -5 4305  
4305: prioridad antigua 0, nueva prioridad 5
```

```
# renice 25 4311  
4311: prioridad antigua 0, nueva prioridad 19
```

```
# top
```

### **Actividad 3.3. Jerarquía e información de procesos**

- a) La orden pstree muestra el árbol de procesos que hay en ejecución. Comprueba que la jerarquía mostrada es correcta haciendo uso de la orden ps y de los valores "PID" y "PPID" de cada proceso.
- b) Ejecuta la orden ps con la opción -A, ¿qué significa que un proceso tenga un carácter "?" en la columna etiquetada como TTY?

b) - Significa que no tiene asociado ningún terminal en concreto.

### **Actividad 3.4. Estadísticas de recursos del sistema**

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

- a) ¿Qué porcentaje de tiempo de CPU se ha usado para atender interrupciones hardware?
- b) ¿Y qué porcentaje en tratar interrupciones software?
- c) ¿Cuánto espacio de swap está libre y cuánto ocupado?

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

b)  
# mpstat  
- Tal y como indica la columna %soft, correspondiente al porcentaje de CPU dedicado a interrupciones software, se ha dedicado un 0,04%.

c)  
# top  
- Como podemos ver con la orden top, el total de memoria SWAP es de 1046524k, de los cuales en uso hay 0k, así que toda está libre: 1046524k.

### **Actividad 3.5. 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.

## OPTIONS

- b Display the amount of memory in bytes.
- c count Display the result count times. Requires the -s option.
- g Display the amount of memory in gigabytes.
- k Display the amount of memory in kilobytes. This is the default.
- l Show detailed low and high memory statistics.
- m Display the amount of memory in megabytes.
- o Display the output in old format, the only difference being this option will disable the display of the "buffer adjusted" line.
- s Continuously display the result delay seconds apart. You may actually specify any floating point number for delay, usleep(3) is used for microsecond resolution delay times.
- t Display a line showing the column totals.
- V Display version information.

-Por lo tanto, si queremos ver cada x segundos el resultado de la orden free (en este ejemplo cada 4 segundos) debemos poner:

```
# free -s 4
Ó
# watch -n 4 free
```

### Actividad 3.6. Utilización de vmstat

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

Para ello he creado el siguiente script, donde creo el fichero de texto, lo leo y voy cogiendo cada dato de la columna correspondiente, para luego mostrarlo en formato texto para que resulte más práctico, el archivo se llama script\_4-2.sh:

```
#!/bin/bash
vmstat >> monitorizacion.txt
num_procs=`cat "monitorizacion.txt"|tail -1|tr -s " " "|cut -d " " -f2`
num_procs_swap=`cat "monitorizacion.txt"|tail -1|tr -s " " "|cut -d " " -f17`
incremento_swap=`cat "monitorizacion.txt"|tail -1|tr -s " " "|cut -d " " -f9`
mem_libre=`cat "monitorizacion.txt"|tail -1|tr -s " " "|cut -d " " -f5`
echo "Hay " $num_procs " procesos en cola de ejecución"
echo "Hay " $num_procs_swap " procesos ejecutándose en el área de intercambio"
if [ $incremento_swap -eq 0 ]
then
echo "No hay incremento de procesos pasándose a la memoria de intercambio"
else
echo "Se ha incrementado el numero de procesos que van a la memoria de intercambio en " $incremento_swap
fi
echo "La memoria libre es de " $mem_libre "KB"
-
```

Para realizar una revisión periódica( en este ejemplo que se actualice cada 4 segundos) , podemos ejecutar en un terminal el siguiente comando:

```
# watch -n 4 ./script_4-2.sh
```

Lo que nos da la siguiente salida:

Every 4,0s: ./script\_4-2.sh

Sun Oct 21 12:41:39 2012

Hay 0 procesos en cola de ejecución

Hay 0 procesos ejecutándose en el área de intercambio

No hay incremento de procesos pasándose a la memoria de intercambio

La memoria libre es de 2298828 KB

### **Actividad 3.7. Consulta de metadatos de archivo**

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

```
# ls -a /root
```

### **Actividad 3.8. Listados de metadatos de archivos: ls**

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

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

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

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

(a) Que contenga el campo “access time” de los archivos del directorio especificado y que esté ordenado por dicho campo.

```
# ls -ltu $HOME
```

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

```
# ls -ltc $HOME
```

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

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

1. Comprueba cuántos bloques de datos está usando la partición raíz del sistema UML del laboratorio. Ahora obtén la misma información pero expresada en “human readable format”: Megabytes o Gigabytes. Para ello consulta en detalle el manual en línea.
2. ¿Cuántos inodos se están usando en la partición raíz? ¿Cuántos nuevos archivos se podrían crear en esta partición?
3. ¿Cuál es el tamaño del directorio /etc? ¿Y el del directorio /var? Compara estos tamaños con los de los directorios /bin, /usr y /lib. Anota brevemente tus conclusiones.
4. Obtén el número de bloques de tamaño 4 KB que utiliza la rama de la estructura jerárquica de directorios que comienza en el directorio /etc. En otras palabras, los bloques de tamaño 4 KB del subárbol cuya raíz es /etc. ¿Cuál es el tamaño de bloque, por omisión, utilizado en el SA?

1.

```
# du /usr
```

```
# du -h /usr
```

2.

```
# df -i /
```

3.

```
# du -h /etc
```

4.

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

```
# tune2fs -l /dev/sda1 | grep -i "Block size"
```

### **Actividad 3.10. Creación de enlaces con la orden ln**

Construye los mismos enlaces, duros y simbólicos, que muestra la salida por pantalla anterior. Para ello crea los archivos `archivo.txt` y `target_hardLink2.txt` y, utilizando el manual en línea para `ln`, construye los enlaces `softLink`, `hardLink` y `hardLink2`. Anota las órdenes que has utilizado.

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

```
# ln -s archivo.txt ./softLink
```

```
# ln archivo.txt hardlink
```

```
# ln target_hardLink2.txt hardlink2
```

```
# mkdir D1/
```

```
# ls -lai
```

¿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`?

-

Porque los enlaces blandos no cuentan en el contador de enlaces, tan sólo los duros, que son realmente referencias reales al metadato, los blandos tan sólo al nombre del archivo.

### **Actividad 3.11. Trabajo con enlaces**

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

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

```
# ls -laiL
```

```
# ls -laid
```

### Actividad 3.12. Creación de archivos especiales

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

```
# mknod disp_bloques b 10 5
# mknod disp_caracteres c 10 5
# ls -li
```

### Actividad 4.1. Consulta de información sobre procesos demonio

A partir de la información proporcionada por la orden ps encuentre los datos asociados a los demonios atd 8 y crond, en concreto: quién es su padre, qué terminal tienen asociado y cuál es su usuario.

```
# ps -aux|grep atd
# ps -p 816 -f

# ps -aux|grep cron
# ps -p 817 -f
```

### Actividad 4.2. 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?

```
#!/bin/bash
ls ~ > listahome-`date +%Y-%j-%T-$$`

# at -f ./genera-apunte.sh 11:03
```

### Actividad 4.3. 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
  - b) un minuto después de la medianoche de hoy
  - c) a las 17 horas y 30 minutos de mañana
  - d) a la misma hora en que estemos ahora pero del día 25 de diciembre del presente año
  - e) a las 00:00 del 1 de enero del presente año
- Utiliza las órdenes atq y atrm para familiarizarte con su funcionamiento (consulta la ayuda de estas órdenes).

```
a) a medianoche de hoy
# at -f ./genera-apunte.sh midnight
warning: commands will be executed using /bin/sh
```

job 5 at Tue Oct 30 00:00:00 2012

b) un minuto después de la medianoche de hoy

```
# at -f ./genera-apunte.sh midnight+1 minute
```

warning: commands will be executed using /bin/sh

job 7 at Tue Oct 30 00:01:00 2012

c) a las 17 horas y 30 minutos de mañana

```
# at -f ./genera-apunte.sh 17:30 tomorrow
```

warning: commands will be executed using /bin/sh

job 8 at Tue Oct 30 17:30:00 2012

d) a la misma hora en que estemos ahora pero del día 25 de diciembre de 2013

```
# at -f ./genera-apunte.sh Dec 25 2013
```

warning: commands will be executed using /bin/sh

job 11 at Wed Dec 25 11:24:00 2013

e) a las 00:00 del 1 de enero de 2012

```
at -f ./genera-apunte.sh midnight Jan 01 2013
```

warning: commands will be executed using /bin/sh

job 11 at Wed Dec 25 11:24:00 2013

#### **Actividad 4.4. Cuestiones sobre at**

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

20. ¿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?

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

22. ¿hereda las variables locales del proceso padre?

Experimenta con la orden at lanzando las órdenes adecuadas para encontrar las respuestas.

(Puede encontrar información en la ayuda de 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, como dicho anteriormente, 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 BASH\_VERSION, DISPLAY, EUID, GROUPS, SHELL\_OPTS, TERM, UID, y \_

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

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

```
nombrearchivo=`date +%Y-%j-%T`
```

```
ps -ef > $nombrearchivo
```

```
echo Mi pid = $$ >> $nombrearchivo
```

```
#!/bin/bash
#Nombre: script_4.4.sh
nombrearchivo=`date +%Y-%j-%T`
ps -ef > $nombrearchivo
echo Mi pid = $$ >> $nombrearchivo
Ejecuto el programa:
# at -f ./script_4.4.sh now+1 minute
warning: commands will be executed using /bin/sh
job 17 at Mon Oct 29 11:57:00 2012
Vemos el resultado:
# cat 2012-303-11\57:00
daemon
14192
816 0 11:56 ?
superjes 14194 14192 0 11:56 ? 00:00:00 atd
superjes 14196 14194 0 11:56 ? 00:00:00 sh
Mi pid = 14194 00:00:00 ps -ef
```

#### **Actividad 4.6. Script para orden at**

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

```
#!/bin/bash
#Nombre: script_4.5.sh
find ~ -mtime 1 > modificados:`date +%Y-%j-%T`
Ejecutamos con at:
# at -f ./script_4.5.sh now+1 day
```

#### **Actividad 4.7. Trabajo con la orden batch**

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

#### **Actividad 4.8. Utilización de las colas de trabajos de at**

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



```
#!/bin/bash
#Nombre: script_4.7.sh
at -q c -f ./script_4.4.sh now+1 minute
at -q d -f ./script_4.4.sh now+1 minute
at -q e -f ./script_4.4.sh now+1 minute
```

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

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

```
#Nombre: cron_5.1
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/script_4.4.sh
Ejecutamos con cron:
$ crontab cron_5.1
```

#### **Actividad 4.10. Ejecución de scripts con crontab (I)**

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

```
#!/bin/bash
#Nombre: script_5.2.sh
rm -v /tmp/varios/core* >> /tmp/listacores
```

-  
Creamos el archivo que determina las opciones de crontab:

```
#Nombre: cron_5.2
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/script_5.2.sh
```

-  
Ejecutamos con crontab:

```
$ crontab cron_5.2
```

Pruebe 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).

```
$ crontab -l
```

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/script_5.2.sh
```

#### Actividad 4.11. Ejecución de scripts con crontab (II)

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

```
#!/bin/bash
#Nombre: reducelista
tail -n 10 /tmp/listacores > /tmp/temporal
rm /tmp/listacores
mv /tmp/temporal /tmp/listacores
-
Creamos el archivo al que llamará crontab:
#Nombre: cron_5.3
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/reducelista
-
Ejecutamos la orden crontab:
$ crontab cron_5.3
```

#### Actividad 4.12. Ejecución de scripts con crontab (III)

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

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

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

```
#Nombre: script_5.4.sh
#!/bin/bash
fecha=`date +%Y-%j-%T`
echo $fecha - $PATH >> /home/superjes/SO/Practica_1/listabusqueda
-
Creamos el archivo que ejecutará crontab posteriormente(cada minuto):
#Nombre: cron_5.4
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/script_5.4.sh
-
Probamos la ejecución directamente:
$ cat listabusqueda
2012-303-19:57:16 -
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

Probamos la ejecución con crontab:

```
$ crontab cron_5.4
```

```
$ cat listabusqueda
```

```
2012-303-20:01:01 - /usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

-

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.

#### **Actividad 4.13. Variables de entorno en archivos crontab**

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

... como primera línea la asignación a la variable PATH de la lista de búsqueda actual y además el directorio \$HOME/SO

. después la indicación a cron de la ejecución con periodicidad de 1 minuto del script reducelista

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

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

```
#Nombre: script_5.5.sh
```

```
echo "SHELL=/bin/sh" > crontab-reducelista
```

```
echo "PATH=`pwd`/:$HOME/SO:$PATH" >> crontab-reducelista
```

```
echo "* * * * * reducelista" >> crontab-reducelista
```

-

El archivo crontab-reducelista se crea pues con este contenido:

```
SHELL=/bin/sh
```

```
PATH=/home/superjes/SO/Practica_1/:/home/superjes/SO:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

```
* * * * * reducelista
```

-

Lanzamos con crontab:

```
$ crontab crontab-reducelista
```

#### **Actividad 4.14. Archivos crontab de diferentes usuarios**

Vamos a lanzar un archivo crontab cuyo propietario es otro usuario. Visualiza el contenido del archivo /fenix/depar/lisi/so/ver-entorno y /fenix/depar/lisi/so/crontabver. Comprueba con ls -l que el propietario es el usuario lisi. Sin copiarlos, úsalos para lanzar la ejecución cada minuto del script /fenix/depar/lisi/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?

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

```
#Nombre: script_5.6.sh
```

```
echo $LOGNAME > /tmp/fichero
```

```
echo $HOME >> /tmp/fichero
```

-

Creo el fichero crontab:

```
#Nombre: cron_5.6
```

```
SHELL=/bin/sh
```

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

```
* * * * * /home/superjes/SO/Practica_1/script_5.6.sh
```

-

Con otro usuario( en este caso uno llamado pato), lanzo con la orden crontab el fichero anterior:

```
$ crontab /home/superjes/SO/Practica_1/cron_5.6
```

-

Compruebo el fichero creado:

```
$ cat /tmp/fichero
```

```
pato
```

```
/home/pato
```

-

Como vemos, los valores se toman del usuario que lanza el comando crontab.

#### **Actividad 4.15. Ejecución de scripts con crontab (IV)**

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

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

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

```
#Nombre: script_5.7.sh
```

```
find ~ -mtime 1|cpio -pmduv /tmp/salvado$HOME
```

-

Creo el archivo crontab:

```
#Nombre: cron_5.7
```

```
SHELL=/bin/sh
```

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

```
0 0 * * * /home/superjes/SO/Practica_1/script_5.7.sh
```

-

Ejecutamos con crontab:

```
$ crontab cron_5.7
```

#### **Actividad 4.16. Gestión del servicio crond como usuario root**

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.
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

1)

Como root añado (o creo si no existe) al archivo /etc/cron.deny una nueva línea con el nombre del usuario (en este caso el usuario pato) al que quiero prohibir el uso de crontab:

```
# echo pato >> /etc/cron.deny
```

-

Nos metemos como el usuario que acabamos de indicar y probamos a ejecutar crontab:

```
# su pato
```

Contraseña:

```
pato@superjes-VirtualBox:/home/superjes/SO/Practica_1$ crontab cron_5.6
```

```
You (pato) are not allowed to use this program (crontab)
```

```
See crontab(1) for more information
```

-

Como podemos ver, efectivamente no tenemos permisos para ejecutar crontab.

2)

```
# service cron stop
```

```
cron stop/waiting
```

```
# service cron start
```

```
cron start/running, process 21796
```

## Examen

1º[1/11 12:02 a.m.] Salva :3: Explique como conseguir un usuario almacenado en dolar u que tenga como lista de búsqueda cada vez que se ejecute el valor /usr/bin:/bin

2º[1/11 12:03 a.m.] Salva :3: Buscar un archivo de enlace duro en la variable dolar a de una ruta

3º[1/11 12:04 a.m.] Salva :3: Generar el archivo /tmp/nombres conteniendo la lista de los Nombres de usuario

[1/11 12:04 a.m.] Salva :3: Solo nombres

```
cat /etc/passwd | cut -d: -f 1 > /tmp/nombres
```

Explicación:

El comando cut es muy versatil. Entre otras cosas, podemos dividir un fichero de texto que esté estructurado en líneas, y que contenga un separador de campos, como puede ser un espacio, o en este caso, ":"

cut -d: (-d indica el separador de campo :) -f1 (-f indica el número de campo que queremos utilizar, en este caso el 1)

Como la pregunta no dice si queremos usuarios personas o no, pues ahí salen todos.

Si queremos usuarios personas, tendríamos que ver el valor de UID (tercer campo de passwd) más bajo, y hacer una búsqueda con awk, por ejemplo:

```
awk -F: '{if ($3>=500){print $1}}' /etc/passwd > /tmp/nombres
```

A mi me salen los usuarios personas y también el usuario del sistema "nobody" cuyo UID es 65531.

Entonces podríamos poner como condición `(( $3 >= 500 ) && ( $3 <= 9999 ))`, o sea:

```
awk -F: '{if ($3 >= 500) && ($3 <= 9999)}{print $1}}' /etc/passwd > /tmp/nombres
```

`&&` es el `.and.` lógico de `awk`.

`Awk` es todo un lenguaje. Cuando no tengas examen, échale un vistazo, que da muchísimo de sí.

Otra opción diferente es listar el directorio `/home`, y esperar que no haya nada más que usuarios. Yo, por ejemplo, tengo ahí otras cosas.

4º[1/11 12:05 a.m.] Salva :3: 1 de cada mes a las 23 horas y 59 minutos lanzar el script `tmp/scrip` 1

[1/11 12:06 a.m.] Salva :3: Con el shell `/bin /bash`

5º[1/11 12:08 a.m.] Salva :3: Otra pregunta, te da el PS 218 y que te salga el usuario la prioridad el PID del padre

```
ps -P -ly 5023 | awk -F" " '{print $2 " " $7 " " $4}'
```

`$2` es el UID numérico, para obtener el nombre del usuario hay que hacer el comando un poco más larguillo.

[1/11 12:09 a.m.] Salva :3: Lanzar una ejecución `fénix/...../ver entorno` con máxima prioridad

[1/11 12:10 a.m.] Salva :3: Mostrar en la salida el nombre de los archivos montados de tipo `ext 3`

```
mount -l | grep ext3
```

`mount -l` lista los sistemas montados.

[1/11 12:11 a.m.] Salva :3: Obtener la fecha de creación de un dispositivo `dev/loop 0`

`ls -li` fichero

```
debugfs -R 'stat <INODE_NUMBER>' /dev/sda1
```

[1/11 12:12 a.m.] Salva :3: Explica el número de procesos que NO se están ejecutando