

SISTEMAS OPERATIVOS

Algunos ejercicios sobre el Modulo I de prácticas con comentarios.

1. Explique cómo podríamos conseguir que el usuario *usu1* al hacer un login se ejecute el programa **/bin/hostname** (sin entrar en ningún shell ni poder ejecutar nada más, solo ejecuta esa orden y sale)

Editamos el archivo `/etc/passwd` y ponemos como último campo el literal `/bin/hostname`

Utilizar el archivo `.bash_profile` no nos sirve para este propósito; tras ejecutarse su contenido se ejecuta el interprete `bash`, ¡que no queremos! ¡solo queremos que se ejecute el programa `/bin/hostname` y se vaya!

2. ¿Qué orden daríamos al shell para saber el tamaño de inodo en un sistema de archivos?

```
tune2fs -l  
o bien con df
```

3. Escriba la orden para lanzar la ejecución del archivo `/bin/programax` con la mayor prioridad (mayor preferencia o peso) posible desde nuestro perfil de usuario.

Para lanzarlo con el mayor peso posible (es decir menor valor numérico de prioridad) ejecutaríamos la orden `nice` con valor 0:

```
nice -0 /bin/programax
```

o lo que es lo mismo, podemos lanzar simplemente su ejecución desde el prompt:

```
/bin/programax
```

Aunque no lo pide la pregunta, para lanzar `$p` con el menor peso posible ejecutaríamos

```
nice -19 /bin/programax
```

4. Explique cómo podemos obtener en pantalla el pid, y nombre del proceso de aquellos procesos que no tienen asociado ningún terminal (explique qué órdenes del shell tendríamos que proporcionar, la secuencia de pasos a ejecutar.....)

```
ps -eo pid,tty,comm | grep "?"
```

Los procesos que no tienen asociado ningún terminal son aquellos que tienen en el campo `tty` el valor `"?"` (o bien `"?"`, dependiendo de la distribución en particular que se esté usando).

5. Deseamos llegar un registro sobre cuándo hace login el usuario *usu1*; explique cómo podríamos conseguirlo (es decir, conseguir que quede almacenado de alguna forma en algún sitio la fecha-hora de cada vez que ese usuario hace login)

En el archivo `.bash_profile` ubicado en el directorio inicial del usuario *usu1* incluimos una orden del estilo de la siguiente:

```
date >> /home/usu1/registroLogin
```

6. Deseamos que un determinado sistema de archivos sea utilizado de forma que el sistema operativo realice físicamente toda operación de escritura inmediatamente, no de forma postergada; explique cómo podemos conseguir esto.

mount -o sync

o bien en fstab especificando sync en el lugar correspondiente.

7. En la variable \$b está almacenada la ruta de un archivo que es un enlace **duro** al archivo regular \$a (variable que contiene su ruta).

a) ¿Estos dos archivos pueden tener diferentes tiempos de última modificación de su contenido? Justifique su respuesta

No. La fecha de última modificación del contenido de un archivo está en su inodo. Puesto que \$a y \$b aluden al mismo inodo, comparten la fecha de última modificación que necesariamente es la misma para ambos.

b) ¿Estos dos archivos pueden tener diferentes tiempos de última modificación de sus atributos? Justifique su respuesta

No. La fecha de última modificación de los atributos de un archivo está en su inodo. Puesto que \$a y \$b aluden al mismo inodo, comparten la fecha de última modificación que necesariamente es la misma para ambos.

* En la variable \$b está almacenada la ruta de un archivo que es un enlace **simbólico** al archivo regular \$a (variable que contiene su ruta).

a) ¿Estos dos archivos pueden tener diferentes tiempos de última modificación de su contenido? Justifique su respuesta

no. El contenido es el mismo para ambos.

b) ¿Estos dos archivos pueden tener diferentes tiempos de última modificación de sus atributos? Justifique su respuesta

si. Son elementos distintos del arbol de archivos, cada uno podría tener permisos distintos, etc... y tienen cada uno su propia fecha de última modificación de atributos.

8. Explique cómo conseguir que el usuario usu3 tenga como lista de búsqueda a hacer login el siguiente valor:

`.:bin`

Editamos el archivo /home/usu3/.bash_profile y añadimos

PATH=.:bin

12. Desde sistema UML (el kernel explicado en la sesión1 para trabajar como usuario root): Explique cómo montar el directorio **home** del sistema de archivos anfitrión.

Suponemos que somos el usuario de nombre alumnox, cuyo directorio inicial en turing es /fenix/alum/d3/alumnox.

Suponemos que ya hemos lanzado el kernel UML; primero creamos en el directorio raíz el directorio de nombre `mi-cuenta-en-turing` que servirá como punto de montaje;

```
$ mkdir /mi-cuenta-en-turing
```

después hemos de hacer lo siguiente:

```
$ mount none /mi-cuenta-en-turing -t hostfs -o /fenix/alum/d3/alumnox
```

13. Configura el sistema UML para que cada vez que arranque tengamos montado el directorio *home* del sistema de archivos anfitrión.

incluimos la orden vista en la pregunta anterior en el `.bash_profile`

14. Construya una orden para ver el pid del proceso padre, pid y prioridad de todos los procesos existentes; deberá ordenarse y llevarse a un archivo cuyo nombre sea “listaps” seguido del pid del proceso actual y seguido de la fecha-hora (con el formato que usted elija).

```
ps -eo ppid,pid,prio | sort > listaps$$`date +%Y-%j-%T`
```

* Explique cómo obtener el número de bloques libres que tiene un determinado sistema de archivos.

* Explique cómo saber qué sistema de archivos hay montados.

* Deseamos que un determinado sistema de archivos sea utilizado de forma que el sistema operativo realice físicamente toda operación de escritura inmediatamente, no de forma postergada; explique cómo podemos conseguir esto.

- **Desde sistema UML (el kernel explicado en la sesión 1 para trabajar como usuario root):** Explique cómo conseguir que cada vez que hagamos login tengamos montado el directorio *home* del sistema de archivos anfitrión.

-

* Indica cómo buscarías si un determinado grupo (cuyo nombre tenemos almacenado en la variable `$g`) se encuentran creado en el sistema.

* Trabajando en el intérprete de órdenes bash, hemos almacenado en la variable `$nombre` el nombre de un usuario; explica qué archivos se deben consultar para saber el nombre del grupo al que pertenece el usuario cuyo nombre está almacenado en `$nombre`; indica la secuencia de pasos y las órdenes que darías al shell.

* Explica cómo conseguir que el día 1 de cada mes a las 14 horas y 30 minutos se explore el directorio `/home` y se borren todos los directorios cuyo nombre no corresponda a un usuario definido en `/etc/passwd`

* Utilizando las órdenes `at` y `crontab` queremos que dentro de una hora se lance la ejecución con periodicidad de un minuto del archivo **`/fenix/depar/lisi/so/ver-entorno`**.

* Queremos construir un archivo `crontab` que debe realizar lo siguiente:

- a) añadir a la lista de búsqueda el directorio `/fenix/depar/lisi/so`
- b) copiar los archivos que cuelguen de `/tmp` que contengan en su nombre la palabra “dat” en `/tmp/salvadat`.
- c) se debe generar el archivo `/tmp/salva-dat-log` de modo que cada línea contenga el nombre de un archivo copiado y la fecha-hora en que se ha copiado.
- d) queremos que la shell que se utilice sea `/bin/bash` en lugar de `/bin/sh`

* Explique cómo podríamos conseguir que el usuario *usu1* al hacer un login se ejecute el programa **`/bin/hostname`** (sin entrar en ningún shell ni poder ejecutar nada más, solo ejecuta esa orden y sale)

* Deseamos llegar un registro sobre cuándo hace login el usuario *usu1*; explique cómo podríamos conseguirlo (es decir, conseguir que quede almacenado de alguna forma en algún sitio la fecha-hora de cada vez que ese usuario hace login)

* **Desde sistema UML (el kernel explicado en la sesión1 para trabajar como usuario root):** Explique cómo ejecutar la orden *ls* sobre el directorio *home* asociado al usuario dentro del sistema de archivos anfitrión

* **Desde sistema UML (el kernel explicado en la sesión1 para trabajar como usuario root):** Explique cómo montar el directorio *home* asociado al usuario dentro del sistema de archivos anfitrión.

* Tenemos ya creados dos usuarios *usu1* y *usu2* que, aunque son dos personas diferentes y queremos que cada una use su propio login pero que tengan el mismo directorio inicial. Explique cómo conseguir esto.

* Una vez realizado todo lo descrito en el apartado A de la sesión 2 (**Preparación previa a la partición de un dispositivo simulado mediante un archivo especial de dispositivo**) respecto a */dev/loop0*, explicar qué ordenes deben proporcionarse, como mínimo, hasta poder crear un directorio dentro de dicho dispositivo simulado. Muéstrese la secuencia de órdenes incluyendo la creación del directorio.

* Explique cómo conseguir que el script *ej13* ubicado en el directorio *SSSOO* que cuelga de su directorio inicial *home* sea ejecutado todos los días de noviembre a las 12 horas 00 minutos.