

Laboratorio de Administración y Gestión de Sistemas y Redes
Prueba escrita sobre la teoría. 17 de junio de 2014
Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

Instrucciones:

- Encontrarás en el *home* del puesto del laboratorio el fichero el fichero `~/teoria.tulogin.txt`. Cámbiale el nombre, reemplazando *tulogin* por tu verdadero login. Por ejemplo, si eres *jperez*, debes hacer

```
mv teoria.tulogin.txt teoria.jperez.txt
```

Revisa que esté bien hecho. Si te equivocas en este paso tan sencillo, suspenderás.
- Dentro del fichero `teoria.jperez.txt`, escribe tu login, nombre y apellidos y contesta al examen.

Ejercicio 1 (2.5 puntos)

La transparencia 57 del tema 2, que habla de los enlaces *duros*, dice lo siguiente: *Dado un fichero, se sabe cuántos nombres tiene. Para saber cuáles son sus nombres, habría que buscarlos.*

Explica esta afirmación

Respuesta

Dado un fichero llamado A, se le puede poner otro nombre, por ejemplo B. Decimos que B es un *enlace duro* de A. El nombre B no apunta al nombre A, sino al mismo fichero apuntado por A.

El sistema de ficheros almacena (en el inodo) el número de nombres que tiene cada fichero. Lo podemos ver con la orden `ls -l`, es el número que aparece tras los permisos.

Cada nombre apunta al fichero, pero el fichero no apunta a sus nombres. Esto está representado en la figura: hay una flecha desde cada nombre al fichero, pero la flecha va en un solo sentido. Por tanto, a partir del fichero, no es posible llegar a los nombres. El fichero *sabe* que recibe dos flechas, pero no puede saber *de donde vienen*.

Para conocer todos los nombres del fichero A, habría que recorrer todo el árbol, buscando otros nombres que apunten al mismo fichero. Por ejemplo usando la orden `ls -i A`, que nos indica el número de inodo del nombre A. Tendríamos que buscar por todo el árbol otro nombre con el mismo número de inodo, esto es, otro nombre que apunte al mismo fichero.

Ejercicio 2 (2.5 puntos)

¿Para qué sirve el *journal* en un sistema de ficheros?

Respuesta

Para garantizar que las operaciones sean atómicas, esto es, que se hagan por completo o que no se hagan en absoluto, pero que nunca queden incompletas.

Un sistema de ficheros es una estructura bastante compleja. Si hay algún problema en el sistema en mitad de una operación, por ejemplo un apagón, el sistema podría quedar en un estado inconsistente, con una parte de las estructuras actualizadas y otra no. Ejemplo: mover un fichero consiste en copiarlo en la nueva ubicación y borrar el original. Si la operación queda a medias, podríamos tener dos ficheros o ninguno.

Un sistema de ficheros con *journal* lo que hace es almacenar lo que va a hacer, y luego hacerlo realmente. Si algo falla, una herramienta de recuperación puede comparar lo que dice el *journal* con lo que realmente hay en el sistema de ficheros, y si hay alguna inconsistencia, repararla.

Ejercicio 3 (2.5 puntos)

Tenemos un sistema Unix. Queremos instalar cierto software y disponemos de una versión `.tgz` y otra `.deb`.

1. ¿Cuál deberíamos instalar? Si la respuesta es *depende*, ¿de qué depende?
2. ¿Qué pasaría instaláramos *el otro*?

Respuesta

El formato `.tgz` es general para Unix, el `.deb` es específico de Debian y derivados, por tanto:

1. Depende. Si tenemos un Linux Debian o algún otro sistema basado en Debian, como Ubuntu, deberíamos instalar el `.deb`. Pero si se trata de otra variante de Linux, o bien otro Unix que no sea Linux, entonces normalmente deberíamos instalar el `.tgz`, que es más general.
2. Si intentamos instalar un `.deb` en un sistema que no está basado en estos paquetes, simplemente no podremos, no funcionará nada porque no estarán disponibles los gestores de paquetes.

Pero normalmente sí será posible instalar un `.tgz` en un Debian. No es recomendable porque lo estamos haciendo *de espaldas* al sistema de paquetes, tendremos que resolver a mano la dependencias y las incompatibilidades que pueda haber. Pero normalmente será posible.

Ejercicio 4 (2.5 puntos)

Un compañero que no ha cursado la asignatura te dice lo siguiente:

No se si entiendo bien el protocolo SNMP. Veo que iso.org.dod.internet.mgmt.mib-2.system.sysUpTime es una representación textual que se corresponde con la representación numérica .1.3.6.1.2.1.1.3. Esto es muy parecido a los nombres de dominio y las

direcciones IP, esto es, la representación en modo texto es una dirección web donde yo puedo consultar el valor correspondiente, en este caso el tiempo que lleva una máquina encendida. Y los números serían la dirección IP que me ofrece ese servicio. Pero la secuencia de números no tiene pinta de dirección IPv4. Parece una dirección IPv6, así que supongo que este es un ejemplo de SNMP funcionando sobre IPv6

¿Qué le respondes?

Respuesta

Es verdad que la idea se parece un poco. Pero esa representación numérica no tiene nada que ver con una dirección IP, ni v4 ni v6. Son dos representaciones del mismo objeto, pero no son direcciones de hosts.