

Nombre:

Fecha: /12/2011

Grupo: 1 🗆 2 🗆 3 🗆 4 🗆

PRÁCTICA 21

CONFIGURACIÓN DE RED EN LINUX. MANDATO AWK. SCRIPTS BÁSICOS

En esta práctica veremos algunas cuestiones básicas sobre configuración de redes en nuestra distribución Linux desde la shell. Veremos algunos de los mandatos que permiten conocer la configuración de red en Linux, saber si la misma está funcionando, comprobar ciertas estadísticas sobre la red o comprobar los servidores DNS.

En la segunda parte de la misma recuperaremos algunos de los mandatos para tratamiento de cadenas que ya hemos utilizado en prácticas anteriores (grep, sed, sort...) e introduciremos algunos nuevos (awk) para tratar de conseguir herramientas más potentes que nos permitan trabajar con ficheros de texto.

Antes de arrancar la máquina virtual, vamos a modificar su configuración. Entra en el menú "Configuración" de VirtualBox y en el mismo, en la pestaña "Adaptador 2", deshabilita el adaptador de red. El mismo genera un adaptador de red adicional para nuestra máquina virtual que no funcionará una vez en marcha la misma, así que podemos prescindir de él.

1. Tras realizar el cambio en la configuración, arranca la máquina virtual de Linux. Inicia una terminal. Ejecuta el mandato "man ifconfig". Encuentra la utilidad de ifconfig. Ejecuta el mandato "ifconfig". Con ayuda de <u>http://es.wikipedia.org/wiki/Loopback</u> y de <u>http://es.wikipedia.org/wiki/Ethernet</u> identifica y apunta en tu informe de prácticas las distintas interfaces de red que tiene la máquina.

Para cada una de ellas, y con ayuda de "man ifconfig" o de la explicación en <u>http://es.wikipedia.org/wiki/Ifconfig</u> identifica (y apunta en tu informe), si tienen, la dirección de la tarjeta MAC (o dirección hardware), la dirección IP (inet y inet6), la dirección de difusión y la dirección de máscara.

Comprueba en los siguientes campos de la interfaz si la misma se encuentra en funcionamiento (activa, "up" sin funcionar, "down"). Comprueba el valor de MTU y recuerda su significado y su valor de las prácticas que hicimos en Windows con Wireshark.

2. El mandato ifconfig nos permite gestionar las principales funciones de los interfaces de red de forma sencilla. Veamos cómo podemos "detener" una interfaz de red. Ejecuta el mandato:

\$ifconfig eth0 down

¿Qué ha sucedido? Ejecuta de nuevo el mandato por medio de "sudo".

Comprueba el mensaje que aparece en la pantalla (o trata de abrir el navegador y abrir una página web). Acabamos de detener la interfaz de red, y por tanto nuestra máquina ha perdido la conexión a Internet.

Para "levantar" de nuevo la interfaz, ejecuta el mandato:

\$sudo ifconfig eth0 up

Asegúrate de que has recuperado la interfaz de red (puedes ejecutar "ifconfig eth0" y ver su estado, o simplemente comprobar que vuelves a tener conexión a Internet).

Además de para levantar o "tumbar" interfaces de red, ifconfig permite modificar máscaras de red, direcciones ip's y algunos de los otros parámetros de interfaces.

3. A partir de la dirección de la máscara de red y de la IP que tiene tu máquina, ¿qué direcciones IP estarán disponibles en tu red local? Ten en cuenta que la última corresponde a la dirección de difusión (broadcast), y la primera a la dirección de red.

4. Ejecuta el mandato:

\$nslookup belenus.unirioja.es

Apunta la dirección de tu servidor DNS y la de la página web <u>belenus.unirioja.es</u>. Ejecuta el siguiente mandato y apunta la dirección IP de la página:

\$nslookup www.w3c.org

5. Visualiza el fichero /etc/resolv.conf. Apunta las direcciones de los servidores DNS allí disponibles.

6. Comprueba el contenido del fichero /etc/host.conf. Este fichero es consultado por el sistema cada vez que tiene que resolver una IP.

La línea "order" indica en qué orden se resuelven los nombres; "hosts" indica el fichero "/etc/hosts", "bind" consulta el nombre en los servidores DNS (si un nombre de dominio se encuentra disponible en /etc/hosts el sistema se "ahorrará" la invocación al servidor DNS).

La línea "multi on" indica que un determinado host puede tener más de una IP en el fichero "/etc/hosts".

Puedes encontrar información detallada de otras opciones que ofrece este fichero en <u>http://www.tldp.org/LDP/nag/node82.html</u>.

7. Comprueba en <u>http://es.wikipedia.org/wiki/Archivo hosts</u> la utilidad del fichero "/etc/hosts". Apunta en tu informe de prácticas las técnicas que utiliza para bloquear páginas web no deseadas.

Comprueba los permisos y propietario del fichero "/etc/hosts" en tu sistema (por medio de "ls -l"). Vamos a introducir ahora las siguientes líneas en el fichero /etc/hosts por medio de un editor de textos de línea (como por ejemplo nano); añádelas antes de las líneas relativas al protocolo ipv6 (si

no dispones de los permisos necesarios realiza las acciones necesarias para poder editarlo).

193.146.250.30 casitodo.com 128.30.52.45 elqueponelasreglas

Guarda el fichero, abre un navegador y teclea en el mismo las direcciones:

casitodo.com elqueponelasreglas

¿Qué páginas se han abierto? ¿Por qué?

8. Recupera el fichero /etc/hosts y limpia las líneas que has añadido a mano. Añade la siguiente línea en su lugar:

255.255.255.0 <u>www.google.com</u>

¿A qué dirección de la red corresponde 255.255.255.0? Comprueba ahora el resultado de las siguientes operaciones:

\$nslookup <u>www.google.com</u>

¿Qué direcciones IP aparecen?

\$ping <u>www.google.com</u>

¿Qué dirección IP aparece asignada a www.google.com?

Por medio de la anterior acción no hemos conseguido "engañar" a los servicios que consultan DNS en nuestro servidor DNS (como nslookup), pero sí a todos los servicios (como ping o el navegador) que comprueban las direcciones IP en el fichero "/etc/hosts" y posteriormente en los servidores DNS.

Abre una pestaña del navegador y trata de acceder a <u>www.google.com</u>; ¿cuál es el resultado?

Elimina la línea que hemos añadido en "/etc/hosts" y guarda el fichero.

9. El mandato ifconfig dispone de facilidades que permiten modificar la IP de una interfaz de red (por ejemplo, de eth0), su máscara de red o la puerta de enlace de forma tan sencilla como:

ifconfig eth0 "nueva ip" ifconfig eth0 netmask "nueva máscara" ifconfig eth0 broadcast "nueva difusión"

También se puede acceder a información sobre las distintas interfaces de red de que dispone el ordenador en el fichero "/etc/network/interfaces". En nuestro caso, los interfaces de red son configurados de forma automática, pero en caso de que tuviésemos que definirlas de forma estática puedes

encontrar en <u>http://www.cyberciti.biz/faq/setting-up-an-network-</u> <u>interfaces-file/</u> algunos ejemplos útiles.

En el resto de la práctica seguiremos trabajando con ficheros de texto y utilizando algunos de los mandatos que vimos en prácticas anteriores.

10. El mandato awk contiene un escaneador de patrones y un lenguaje de procesado de textos. Tiene capacidades propias de un lenguaje de programación (<u>http://es.wikipedia.org/wiki/AWK</u>). Sus capacidades, junto con las de sed y las de grep, permiten completar tareas realmente elaboradas sobre ficheros de texto. Nosotros nos centraremos en sus capacidades para escanear ficheros y procesarlos, generando salidas a partir de los datos escaneados.

Para comenzar, empezaremos por una orden sencilla:

\$awk '{print "Hola mundo"}'

Ejecútalo. El prompt espera a que introduzcas una cadena de caracteres. ¿Por qué? ¿Qué sucede cada vez que introduces una cadena de caracteres y pulsas el salto de línea?

El programa awk (al igual que grep o sed) necesita por defecto de un fichero sobre el cual va a procesar la orden (u órdenes) encerrada entre comillas simples ({print "Hola mundo"}).

En el caso anterior, lo que hemos hecho es ejecutar el mandato {print "Hola mundo"} para cada una de las líneas del fichero de entrada (el propio shell).

11. Veamos ahora un segundo ejemplo de su uso. El mandato awk nos permite escanear ficheros de texto (o la propia shell) y dividirlos por líneas, pero además escanea cada línea separándola por cualquier separador que nosotros le indiquemos. Por defecto, el separador que considera es el espacio en blanco. Para hacer referencia al campo número "n" de los campos en que una línea ha sido separada, hacemos uso de "\$n".

Ejecuta el mandato:

\$who

Queremos quedarnos, de las líneas procesadas, sólo con el primer campo, el nombre de los usuarios "logados" en la máquina, y con el cuarto, su hora de "login". Escribe el siguiente mandato:

\$ who | awk '{print "El usuario " \$1 " se ha logado a la hora " \$4}'

Comprueba el resultado de ejecutar el mandato.

12. No siempre estaremos interesados en usar el espacio en blanco como separador de campos dentro de ficheros. Por ejemplo, por medio de "less /etc/passwd" puedes comprobar el contenido del fichero de usuarios de tu máquina, y que en el mismo el carácter que actúa como separador es ":".

Puedes modificar el carácter usado como separador en un fichero en awk por medio de la opción -F ":".

Ejecuta el siguiente mandato:

\$less /etc/passwd | awk -F ":" '{print "El usuario " \$1 " tiene como directorio origen " \$6}'

Comprueba el resultado y observa cómo ahora hemos usado como separador entre campos el carácter ":".

13. A partir del mandato anterior, y por medio del uso de grep y expresiones regulares, muestra por pantalla el anterior mensaje pero sólo para aquellos usuarios cuya primera letra de login sea "r", "a" o "c" (quizá tengas que modificar el orden de las tuberías).

14. Ejecuta un mandato que te permita hacer la siguiente función: listar los contenidos de un directorio, filtrar sólo los directorios, y para cada directorio mostrar el mensaje:

"El directorio " nombre " con permisos " lista permisos " y con " número enlaces fuertes " enlaces fuertes pertenece a : " propietario

Prueba a ejecutarlo en la carpeta "/etc".

15. Ejecuta un mandato que te permita hacer la siguiente función: listar los contenidos de un directorio, filtrar sólo los archivos, y para cada archivo mostrar el mensaje:

"El archivo " nombre " con permisos " lista permisos " y con " número enlaces fuertes " enlaces fuertes pertenece a : " propietario

Prueba a ejecutarlo sobre la carpeta "/etc".

En la primera práctica sobre scripts vimos cómo crear scripts sencillos, definir variables, estructuras condicionales (if ... else ...) o utilizar parámetros. Vamos a repasar alguna de las ideas vistas.

16. Por medio de nano crea el siguiente script:

#!/bin/bash
#En la primera línea colocamos el intérprete usado

echo "Algunas de las variables usadas en el script:" echo "La variable \$# es el número total de parámetros" echo "La variable \$0 es el nombre de este script" echo "La variable \$1 es el primer parámetro pasado" echo "La variable \$2 es el segundo parámetro pasado" echo "La variable \$\$ nos permite conocer el PID del proceso asociado"

TEXT_VAR="una variable de texto" NUM_VAR=43 echo "También podemos mostrar variables de texto o numéricas" echo "como \$TEXT_VAR o \$NUM_VAR"

echo "Y por supuesto también podemos mostrar variables de entorno" echo "como \$HOME, \$USER o \$PWD"

Guárdalo en un fichero de nombre "variables.sh". Modifica sus permisos para que tú puedas leerlo, escribirlo y ejecutarlo pero los otros miembros de tu grupo y el resto de usuarios sólo puedan leerlo y ejecutarlo (estos permisos son muy comunes para scripts, 755).

Ejecútalo (./nombre_script) pasándole una lista de dos parámetros cualesquiera y comprueba el resultado. Copia el script obtenido a tu informe de prácticas.

(Sobre el script anterior puedes comprobar que, en la programación de shell, algunos detalles a priori poco relevantes, como la presencia de espacios en blanco entre el nombre de una variable y el "=", pueden hacer que el script no se ejecute).

17. Vamos a crear también un script condicional. Para ello debemos conocer primero la estructura básica de las cláusulas condicionales:

if [condicion]#los espacios antes y después de la condición son relevantes then

else

fi

Para conocer la lista de condiciones de que puedes hacer uso, puedes consultar la tabla que encontrarás en <u>http://linuxcommand.org/wss0090.php</u>.

Con la anterior información, crea un script en un fichero condicional.sh que compruebe si el parámetro que recibe es un fichero que ya existe en esa carpeta. Si es así, mostrará un mensaje en pantalla con el nombre del fichero. Si no es así, creará el fichero por medio de touch y le asignará permisos de lectura, escritura y ejecución sólo para el propietario.

Comprueba que el script funciona tanto si existe el fichero previamente como si no.

Copia el anterior script a tu guión de prácticas.

18. Recupera la historia de la práctica en un fichero de nombre "mandatos_practica_21" y súbela junto al guión de la práctica a tu página de inicio en belenus.