



Universidad Simón Bolívar
División de Ciencias Físicas y Matemáticas
Departamento de Computación y Tec. de Información
Redes de Computadoras I (CI-4835)
Sección 1
Prof. Kity Alvarez

Laboratorio nro. 3

Elaborado por:

Maria Gabriela Hernandez 15-10696

Mariagabriela Jaimes 14-10526

Sartenejas, Enero 2020

Actividad 1

a) \$ echo \$HOSTNAME

El comando echo en linux se usa para mostrar la línea de texto o cadena que se pasa como argumento. Este es un comando integrado que se usa principalmente en scripts de shell y archivos por lotes para enviar texto de estado a la pantalla o un archivo.

En este caso la línea de texto que se pasa es la variable \$HOSTNAME que guarda el nombre que se le da a la computadora que se conecta a la red para identificarla de manera única.

b) \$ cat /etc/network/interfaces

El comando Cat lee datos del archivo y da su contenido como salida. Ayuda a crear, ver y concatenar archivos.

En este caso muestra en el terminal el contenido del filename *interfaces* ubicado en la ruta */etc/network*. El fichero */etc/network/interfaces* se utiliza para definir los nombres lógicos de las interfaces de redes, a los que se les asociará una configuración determinada. El comando *ifup* es el encargado de relacionar una interfaz física con una interfaz lógica.

c) \$ cat /etc/resolv.conf

En este caso el comando cat lee los datos y muestra el contenido en el terminal del archivo *resolv.conf*. Este es un archivo dinámico *resolv.conf* para conectar clientes locales a

la resolución interna de código auxiliar DNS de *systemd-resolved*. Este archivo enumera todos los dominios de búsqueda configurados

En el fichero */etc/resolv.conf* es donde se ponen los servidores de nombres (DNS) que utilizará el equipo.

d) \$ cat /etc/protocols

Se lee y muestra el contenido del archivo *protocols*, que contiene información y lista los protocolos conocidos utilizados en Internet de DARPA. Para cada protocolo, una sola línea está presente con la siguiente información:

nombre_oficial_del_protocolo número_del_protocolo alias

e) \$ cat /etc/services

En este comando se muestra en terminal el contenido del fichero */etc/services*, que almacena y lista información sobre numerosos servicios que las aplicaciones cliente pueden usar en la computadora. Para cada servicio, se especifica el número de puerto conocido del servicio y señala si el servicio está disponible como un servicio TCP o UDP.

f) \$ ifconfig eth0 192.120.67.200 netmask 255.255.255.0 up

El comando "ifconfig" se utiliza para mostrar información de configuración de red actual, configurar una dirección IP, máscara de red o dirección de difusión en una interfaz de red, crear un alias para la interfaz de red, configurar la dirección de hardware y habilitar o deshabilitar las interfaces de red.

Lo que hace el comando ejecutado es asignar la Dirección IP *92.120.67.200* a la interfaz de red *eth0*, luego configura la network mask (máscara de red) *255.255.255.0* a la interfaz dada (*eth0*) y finalmente activa dicha interfaz de red con el flag *up*

g) \$ iwconfig eth1 essid "red local privada 1"

iwconfig es similar a *ifconfig*, pero está dedicado a las interfaces inalámbricas. Se utiliza para establecer los parámetros de la interfaz de red que son específicos de la operación inalámbrica (la frecuencia inalámbrica, por ejemplo). *Iwconfig* también se puede usar para mostrar esos parámetros y las estadísticas inalámbricas (extracted from */proc/net/wireless*).

En el comando ejecutado se establece el ESSID (o nombre de red; en algunos productos también se puede llamar ID de dominio) de *eth1* como "*red local privada 1*".

h) \$ netstat -a

El comando *netstat* muestra todas las conexiones activas de una computadora. Con él se pueden controlar las conexiones de red, así como las estadísticas de la interfaz, las tablas de enrutamiento, entre otros.

Utilizando este comando junto al flag *-a*, se muestran todos los sockets, incluyendo los que poseen "Listening" como estado.

i) \$ netstat -s

El comando *netstat* utilizando el flag *-s* muestra las estadísticas de red por protocolo (Como Simple Network Management Protocol), en donde se pueden visualizar *Ip*, *Icmp*, *Tcp*, *Udp*, *UdpLite*, entre otros.

j) \$ route

En sistemas operativos tipo Unix, el comando de ruta muestra o modifica la tabla de enrutamiento IP.

Cuando se ejecuta este comando en el terminal se muestra una tabla con los detalles de las entradas de la tabla de enrutamiento IP (Kernel IP routing table) . Esto nos muestra cómo está configurado el sistema actualmente

k) \$ route add default gw 192.168.110.3 eth0

Al ejecutar dicho comando se añade una nueva ruta y se enrutan paquetes por default a la red 192.168.110.3 via la interfaz de red "eth0"

l) \$ ping www.google.com

Con el comando ping, puede determinar si una IP de destino remota está activa o inactiva. También puede encontrar el retraso de ida y vuelta en la comunicación con el destino y verificar si hay una pérdida de paquetes. El comando ping es una de las utilidades más utilizadas para solucionar problemas, probar y diagnosticar problemas de conectividad de red.

En este caso, al ejecutar el comando se puede verificar la conectividad de red entre el host y el servidor *www.google.com* cada cierto tiempo.

m) \$ traceroute 202.176.87.35 , traceroute google.com

El comando traceroute imprime la ruta que toma un paquete para llegar al host. Este comando es útil cuando desea saber sobre la ruta y sobre todos los saltos que toma un paquete.

En el caso \$ *traceroute 202.176.87.35*, se imprime la ruta que toma un paquete para llegar a la dirección 202.176.87.35 . En el comando *traceroute google.com* se imprime la ruta que toma para llegar al host *google.com*

La primera columna corresponde al conteo de saltos. La segunda columna representa la dirección de ese salto y después de eso, se verán tres tiempos separados por espacios en milisegundos.

n) \$ arp -a , arp -n

El comando *arp* manipula o muestra la caché de vecinos de red IPv4 del núcleo. Puede agregar entradas a la tabla, eliminar una o mostrar el contenido actual.

En el caso del comando \$ *arp -a*, muestra todas las entradas de los hosts. Los nombres de host se muestran utilizando un formato de salida alternativo de estilo BSD (sin columnas fijas). Por otra parte, el comando \$ *arp -n* muestra las direcciones numéricas en lugar de intentar determinar nombres de host.

o) \$ nslookup usb.ve

nslookup (name server lookup) se utiliza para realizar búsquedas DNS. Se muestran detalles de DNS, como la dirección IP de una computadora en particular, los registros MX de un dominio.

Al ejecutar el comando se muestran detalles como el servidor , la dirección , y el nombre del host *usb.ve*.

Actividad 2

```
#!/bin/sh
ANS=""
while :
do
    echo Timeout en 3 segundos...
    sleep 3
    echo "¿Desea saltar el procedimiento? (responda con s o n): "
    read ANS
    ANS=`echo $ANS | tr "[A-Z]" "[a-z]"`

    if [ "$ANS" = "s" ]
    then
        echo "finalizando el procedimiento ..."
        exit 0
    fi
done
```

El script descrito con anterioridad tiene un tiempo de sleep que dura 3 segundos, que funciona como un timer. Al finalizar dicho tiempo, el programa pregunta al usuario si desea saltar el procedimiento y se debe responder con *s* o *n*. Si se contesta con *n* entonces se vuelve a esperar 3 segundos más y al final se repite dicha pregunta, hasta que se responda *s*. En caso de que se responda con *s*, el programa muestra un mensaje en donde indica que está finalizando el procedimiento, y finalmente termina la ejecución.

Actividad 3 / Actividad evaluada adicional

El shell script de dicha actividad es el archivo *actividad3.sh*

Importante: Para que el script pueda ejecutarse exitosamente y para que se puedan enviar los correos electrónicos a los administradores de los dominios es necesario tener instalada y configurada la librería externa *sendmail*

Para ejecutar el archivo hacer :

```
$ chmod +x actividad3.sh
$ ./actividad3.sh
```