TAREA #987 REALIZAR EL SIGUIENTE LABORATORIO DURANTE LA CLASE DE HOY, EN UN PDF DOCUMENTAR CON SCREENSHOTS LOS RESULTADOS DE SUS COMANDOS Y LAS RESPUESTAS A SUS PREGUNTAS.

Practica de laboratorio
Comandos en MSDOS

A) Anotar los comandos necesarios para ejecutar las siguientes instrucciones desde la consola de Ms-DOS
1.- Obtener la ayuda del comando ping
2.- Enviar un ping a 127.0.0.1 aplicando cualquier parametro
3.- Verificar la conectividad del equipo utilizando el comando ping, anotar conclusiones
4.- Obtener la ayuda del comando nslookup
5.- Resolver la direccion ip de https://upqroo.edu.mx/ usando nslookup
6.- Hacer ping a la ip obtenida en el paso anterior, anotar conclusiones
7.- Obtener la ayuda del comando netstat
8.- Mostrar todas las conexiones y puertos de escucha
9.- Ejecutar netstat sin resolver nombres de dominio o puertos
10.- Mostrar las conexiones TCP
11.- Mostrar las conexiones UDP
12.- Utilizar el comando tasklist
13.- Utilizar el comando taskkill
14.- Utilizar el comando tracert
15.- Utilizar el comando ARP

B) Contesta con tus propias palabras las siguientes preguntas:
1.- ¿Para que sirve el comando ping?
2.- ¿Para que sirve el comando nslookup?
3.- ¿Para que sirve el comando netstat?
4.- ¿Para que sirve el comando tasklist?
5.- ¿Para que sirve el comando taskill?
6.- ¿Para que sirve el comando tracert?
7.- ¿Como ayudan los primeros tres comandos para detectar problemas en la red?

C) Investigar los siguientes comandos y anotar ejemplos practicos:
atmadm, bitsadmin, cmstp, ftp, getmac, hostname, nbtstat, net, net use, netsh, pathping, rcp, rexec, route, rpcping, rsh, tcmsetup, telnet, tftp

Comandos en MSDOS

A) Anotar los comandos necesarios para ejecutar las siguientes instrucciones desde la consola de Ms-Das

1.- Obtener la ayuda del com ando ping

```
dani@Debian:~$  ping -h

Usage
  ping [options] <destination>

Options:
  <destination>      dns name or ip address
  -a                 use audible ping
  -A                 use adaptive ping
  -B                 sticky source address
  -c <count>         stop after <count> replies
  -C                 call connect() syscall on socket creation
  -D                 print timestamps
  -d                 use SO_DEBUG socket option
  -e <identifier>    define identifier for ping session, default is random for
                     SOCK_RAW and kernel defined for SOCK_DGRAM
                     Imply using SOCK_RAW (for IPv4 only for identifier 0)
```

2.- Enviar un ping a 127.0.0.1 aplicando cualquier parámetro.

```
dani@Debian:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.075 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.081 ms
```

3. - Verificar la conectividad del equipo utilizando el comando ping, anotar conclusiones

```
[1]+  Stopped                 ping 127.0.0.1
dani@Debian:~$ ping www.google.com
PING www.google.com (142.251.34.4) 56(84) bytes of data.
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=1 ttl=114 time=70.8 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=2 ttl=114 time=70.2 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=3 ttl=114 time=70.3 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=4 ttl=114 time=69.9 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=5 ttl=114 time=70.5 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=6 ttl=114 time=175 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=7 ttl=114 time=69.9 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=8 ttl=114 time=70.7 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=9 ttl=114 time=71.1 ms
64 bytes from qro01s27-in-f4.1e100.net (142.251.34.4): icmp_seq=10 ttl=114 time=70.4 ms
```

Esto muestra que la conexión es correcta.

4.- Obtener la ayuda del com ando nslookup

```
NSLOOKUP(1)                          BIND 9                          NSLOOKUP(1)

NAME
       nslookup - query Internet name servers interactively

SYNOPSIS
       nslookup [-option] [name | -] [server]

DESCRIPTION
       nslookup  is  a  program to query Internet domain name servers.  nslookup has
       two modes: interactive and non-interactive. Interactive mode allows the  user
       to  query  name servers for information about various hosts and domains or to
       print a list of hosts in a domain.  Non-interactive mode prints just the name
       and requested information for a host or domain.

ARGUMENTS
       Interactive mode is entered in the following cases:

       a. when no arguments are given (the default name server is used);

       b. when  the  first  argument  is a hyphen (-) and the second argument is the
          host name or Internet address of a name server.

 Manual page nslookup(1) line 1 (press h for help or q to quit)
```

5. - Resolver la direccion ip de https://upqroo.edu.mx/ usando nslookup

```
dani@Debian:~$ nslookup upqroo.edu.mx
Server:         192.168.100.1
Address:        192.168.100.1#53

Non-authoritative answer:
Name:   upqroo.edu.mx
Address: 77.68.126.20
```

6.- Hacer ping a la ip obtenida en el paso anterior, anotar conclusiones

```
dani@Debian:~$ ping 77.68.126.20
PING 77.68.126.20 (77.68.126.20) 56(84) bytes of data.
64 bytes from 77.68.126.20: icmp_seq=1 ttl=49 time=128 ms
64 bytes from 77.68.126.20: icmp_seq=2 ttl=49 time=125 ms
64 bytes from 77.68.126.20: icmp_seq=3 ttl=49 time=143 ms
64 bytes from 77.68.126.20: icmp_seq=4 ttl=49 time=123 ms
64 bytes from 77.68.126.20: icmp_seq=5 ttl=49 time=139 ms
64 bytes from 77.68.126.20: icmp_seq=6 ttl=49 time=124 ms
```

Esto muestra que la conexión es correcta.

7.- Obtener la ayuda del com ando netstat

```
dani@Debian:~$ ss -h
Usage: ss [ OPTIONS ]
       ss [ OPTIONS ] [ FILTER ]
   -h, --help          this message
   -V, --version       output version information
   -n, --numeric       don't resolve service names
   -r, --resolve       resolve host names
   -a, --all           display all sockets
   -l, --listening     display listening sockets
   -o, --options       show timer information
   -e, --extended      show detailed socket information
   -m, --memory        show socket memory usage
   -p, --processes     show process using socket
   -T, --threads       show thread using socket
   -i, --info          show internal TCP information
       --tipcinfo      show internal tipc socket information
   -s, --summary       show socket usage summary
       --tos           show tos and priority information
       --cgroup        show cgroup information
   -b, --bpf           show bpf filter socket information
   -E, --events        continually display sockets as they are destroyed
   -Z, --context       display task SELinux security contexts
   -z, --contexts      display task and socket SELinux security contexts
```

8. - Mostrar todas las conexiones y puertos de escucha

```
dani@Debian:~$ lsof -i -n
dani@Debian:~$
```

9. Ejecutar netscat sin resolver nombres de dominio o puertos

```
dani@Debian:~$ ss -n
Netid State  Recv-Q Send-Q                          Local Address:Port    Peer Address:
Port                                    Process
u_str ESTAB  0      0                                       * 17784              *
17787
u_str ESTAB  0      0                 /run/systemd/journal/stdout 18743              *
18742
u_str ESTAB  0      0                 /run/dbus/system_bus_socket 17613              *
17612
u_str ESTAB  0      0                                       * 19203              *
19204
u_str ESTAB  0      0                                       * 14627              *
14628
u_str ESTAB  0      0                                       * 18230              *
18231
u_str ESTAB  0      0                 /run/systemd/journal/stdout 18019              *
18018
u_str ESTAB  0      0                 /run/dbus/system_bus_socket 17787              *
17784
u_str ESTAB  0      0                                       * 19468              *
19469
```

10- Muestra las conexiones TCP

```
dani@Debian:~$ ss -tn
State    Recv-Q    Send-Q       Local Address:Port       Peer Address:Port   Process
dani@Debian:~$
```

11. - Mostrar las conexiones UDP

```
dani@Debian:~$ ss -un
Recv-Q    Send-Q            Local Address:Port         Peer Address:Port    Process
0         0          10.0.2.15%enp0s3:68              10.0.2.2:67
```

12.- Utilizar el comando tasklist

```
dani@Debian:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.6 102456 12524 ?        Ss   20:57   0:01 /sbin/init
root           2  0.0  0.0      0     0 ?        S    20:57   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   20:57   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   20:57   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   20:57   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   20:57   0:00 [netns]
root          10  0.0  0.0      0     0 ?        I<   20:57   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    20:57   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    20:57   0:00 [rcu_tasks_rude_kthr
root          13  0.0  0.0      0     0 ?        I    20:57   0:00 [rcu_tasks_trace_kth
root          14  0.0  0.0      0     0 ?        S    20:57   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    20:57   0:00 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    20:57   0:00 [migration/0]
root          18  0.0  0.0      0     0 ?        S    20:57   0:00 [cpuhp/0]
root          20  0.0  0.0      0     0 ?        S    20:57   0:00 [kdevtmpfs]
root          21  0.0  0.0      0     0 ?        I<   20:57   0:00 [inet_frag_wq]
root          22  0.0  0.0      0     0 ?        S    20:57   0:00 [kauditd]
root          23  0.0  0.0      0     0 ?        S    20:57   0:00 [khungtaskd]
root          25  0.0  0.0      0     0 ?        S    20:57   0:00 [oom_reaper]
root          26  0.0  0.0      0     0 ?        I<   20:57   0:00 [writeback]
root          27  0.0  0.0      0     0 ?        I    20:57   0:00 [kworker/u2:2-events
```

13.- Utilizar el comando taskkill

```
root@Debian:~# kill 2200
root@Debian:~#
```

14.- Utilizar el comando tracert

```
root@Debian:~# traceroute www.google.com
traceroute to www.google.com (142.251.34.4), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.545 ms  0.433 ms  0.347 ms
```

15. - Utilizar el comando ARP

```
dani@Debian:~$ arp
bash: arp: command not found
```

B) Contesta con tus propias palabras las siguientes preguntas:

1.- ¿Para que sirve el comando ping?

se utiliza para comprobar la conectividad entre dos dispositivos a través de una red, generalmente a través de Internet. Envía paquetes de datos a una dirección IP o un nombre de dominio y espera las respuestas. La respuesta indica si el dispositivo de destino está accesible y cuánto tiempo tarda en responder.

2.- ¿Para que sirve el comando nslookup?

se utiliza para consultar servidores de nombres (DNS) para obtener información sobre nombres de dominio y direcciones IP. Puedes usarlo para resolver nombres de dominio, encontrar la dirección IP asociada a un dominio o buscar información sobre registros DNS.

3.- ¿Para que sirve el comando netstat?

se utiliza para mostrar información detallada sobre las conexiones de red, las tablas de enrutamiento, las estadísticas de la interfaz y los puertos abiertos en un sistema.

4.- ¿Para que sirve el comando tasklist?

se utiliza en sistemas Windows para mostrar una lista de los procesos en ejecución en la computadora. Proporciona información sobre el nombre del proceso, el ID del proceso y otros detalles.

5.- ¿Para que sirve el comando taskill?

se utiliza en sistemas Windows para finalizar o cerrar procesos o aplicaciones en ejecución. Puedes utilizarlo para detener procesos problemáticos o no deseados de manera forzada.

6.- ¿Para que sirve el comando tracert?

se utiliza para rastrear la ruta que siguen los paquetes de datos desde tu computadora hasta un destino específico en la red. Muestra una lista de los saltos intermedios que los paquetes hacen a medida que viajan a través de la red.

7.- ¿Como ayudan los primeros tres comandos para detectar problemas en la red?

"Ping" ayuda a verificar si un host remoto está accesible y mide la latencia de la conexión. Si no se recibe respuesta o los tiempos de respuesta son altos, esto puede indicar problemas de conectividad.

"Nslookup" permite verificar la resolución de nombres de dominio, lo que ayuda a identificar problemas de DNS, como errores en la traducción de nombres a direcciones IP.

"Netstat" proporciona información sobre las conexiones y los puertos en uso, lo que puede ayudar a detectar problemas de congestión, escuchas no deseadas o conexiones inusuales en el sistema.

C) Investigar los siguientes comandos y anotar ejemplos practicos:

atmad, bitsadmin, cmstp, fip, getmac, hestname. abtstat, net, net use, netsh, pathping, rep. rexec, route, tesping, Ish, tomsetup, telnet, tip.

atm (Asynchronous Transfer Mode): El comando "atm" es utilizado para configurar y administrar interfaces ATM en sistemas operativos UNIX. Puede configurar parámetros de calidad de servicio y monitorear el estado de las interfaces ATM.

bitsadmin (Background Intelligent Transfer Service): Como mencioné antes, se utiliza en sistemas Windows para crear y gestionar tareas de transferencia de archivos en segundo plano.

cmstp: Este comando se utiliza en sistemas Windows para instalar o desinstalar perfiles de conexión de red. Puede ser útil en la configuración de VPN y conexiones de red.

ftp (File Transfer Protocol): El comando "ftp" se utiliza para transferir archivos entre un cliente y un servidor FTP. Ejemplo práctico: Conectar a un servidor FTP y transferir archivos.

getmac: Muestra la dirección MAC de una interfaz de red en sistemas Windows. Ejemplo práctico: Obtener la dirección MAC de la tarjeta de red Ethernet.

hostname: Muestra el nombre del host de la computadora en la línea de comandos. Ejemplo práctico: Mostrar el nombre de host actual.

nbstat (NetBIOS Statistics): Este comando muestra estadísticas y cierta información relacionada con NetBIOS, un protocolo de comunicación de red en sistemas Windows. Ejemplo práctico: Mostrar estadísticas de NetBIOS.

net: El comando "net" en sistemas Windows se utiliza para administrar recursos compartidos, usuarios y grupos. Ejemplo práctico: Agregar un usuario al grupo.

net use: Como mencioné anteriormente, se utiliza para conectar o desconectar recursos compartidos de red en sistemas Windows.

netsh (Network Shell): Permite configurar y gestionar la configuración de red en sistemas Windows, incluyendo firewall, interfaces de red, etc. Ejemplo práctico: Cambiar la configuración de firewall con "netsh".

pathping: Combinación de "traceroute" y "ping", muestra información sobre la ruta de red y el tiempo de ping a lo largo de esa ruta. Ejemplo práctico: Realizar un diagnóstico de red más detallado.

rexec: Permite ejecutar comandos en sistemas remotos en un entorno UNIX. Ejemplo práctico: Ejecutar un comando en un sistema remoto usando "rexec".

route: Muestra y manipula la tabla de enrutamiento de un sistema. Puede ser utilizado para agregar rutas personalizadas. Ejemplo práctico: Agregar una ruta estática a una red específica.

telnet: Permite establecer una conexión con otro dispositivo a través del protocolo Telnet. Ejemplo práctico: Conectar a un servidor Telnet remoto.

tftp (Trivial File Transfer Protocol): El comando "tftp" se utiliza para transferir archivos de manera simple a través del protocolo TFTP. Ejemplo práctico: Transferir archivos a través de TFTP.