

**Universidad politécnica de quintana roo**

**Sistemas operativos  
Ingeniería en software  
7° cuatrimestre**

**12/10/2023**

**comandos ejercicio 2**

**Ake Prott David Roman  
202100077**



1.-



imac\_20\_invitado — less < man ping — 80x23

PING(8)

System Manager's Manual

PING(8)

#### NAME

**ping** — send ICMP ECHO\_REQUEST packets to network hosts

#### SYNOPSIS

```
ping [-AaCdfnoQqRrv] [-b boundif] [-c count] [-G sweepmaxsize]
      [-g sweepminsize] [-h sweepincrsize] [-i wait] [-k trafficclass]
      [-K netservicetype] [-l preload] [-M mask | time] [-m ttl]
      [-P policy] [-p pattern] [-S src_addr] [-s packetsize] [-t timeout]
      [-W waittime] [-z tos] [--apple-connect] [--apple-time] host
ping [-AaDdfLnoQqRrv] [-b boundif] [-c count] [-I iface] [-i wait]
      [-k trafficclass] [-K netservicetype] [-l preload] [-M mask | time]
      [-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s packetsize]
      [-T ttl] [-t timeout] [-W waittime] [-z tos] [--apple-connect]
      [--apple-time] mcast-group
```

#### DESCRIPTION

The **ping** utility uses the ICMP protocol's mandatory ECHO\_REQUEST datagram to elicit an ICMP ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a "struct timeval" and then an arbitrary number of "pad" bytes used to fill out the

:|

2.-



imac\_20\_invitado — zsh — 80x24

```
64 bytes from 127.0.0.1: icmp_seq=96 ttl=64 time=0.175 ms
64 bytes from 127.0.0.1: icmp_seq=97 ttl=64 time=0.107 ms
64 bytes from 127.0.0.1: icmp_seq=98 ttl=64 time=0.174 ms
64 bytes from 127.0.0.1: icmp_seq=99 ttl=64 time=0.183 ms
64 bytes from 127.0.0.1: icmp_seq=100 ttl=64 time=0.161 ms
64 bytes from 127.0.0.1: icmp_seq=101 ttl=64 time=0.148 ms
64 bytes from 127.0.0.1: icmp_seq=102 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_seq=103 ttl=64 time=0.132 ms
64 bytes from 127.0.0.1: icmp_seq=104 ttl=64 time=0.170 ms
64 bytes from 127.0.0.1: icmp_seq=105 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=106 ttl=64 time=0.170 ms
64 bytes from 127.0.0.1: icmp_seq=107 ttl=64 time=0.179 ms
64 bytes from 127.0.0.1: icmp_seq=108 ttl=64 time=0.170 ms
64 bytes from 127.0.0.1: icmp_seq=109 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=110 ttl=64 time=0.159 ms
64 bytes from 127.0.0.1: icmp_seq=111 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_seq=112 ttl=64 time=0.154 ms
64 bytes from 127.0.0.1: icmp_seq=113 ttl=64 time=0.180 ms
64 bytes from 127.0.0.1: icmp_seq=114 ttl=64 time=0.158 ms
^C
--- 127.0.0.1 ping statistics ---
115 packets transmitted, 115 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.067/0.155/0.198/0.021 ms
imac_20_invitado@iMac-20 ~ %
```

3.-

```
● ○ ● imac_20_invitado — -zsh — 80x24
64 bytes from 127.0.0.1: icmp_seq=109 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=110 ttl=64 time=0.159 ms
64 bytes from 127.0.0.1: icmp_seq=111 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_seq=112 ttl=64 time=0.154 ms
64 bytes from 127.0.0.1: icmp_seq=113 ttl=64 time=0.180 ms
64 bytes from 127.0.0.1: icmp_seq=114 ttl=64 time=0.158 ms
^C
--- 127.0.0.1 ping statistics ---
115 packets transmitted, 115 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.067/0.155/0.198/0.021 ms
imac_20_invitado@iMac-20 ~ % ping www.google.com
PING www.google.com (142.250.189.132): 56 data bytes
64 bytes from 142.250.189.132: icmp_seq=0 ttl=118 time=30.540 ms
64 bytes from 142.250.189.132: icmp_seq=1 ttl=118 time=28.028 ms
64 bytes from 142.250.189.132: icmp_seq=2 ttl=118 time=30.035 ms
64 bytes from 142.250.189.132: icmp_seq=3 ttl=118 time=29.595 ms
64 bytes from 142.250.189.132: icmp_seq=4 ttl=118 time=33.354 ms
64 bytes from 142.250.189.132: icmp_seq=5 ttl=118 time=30.057 ms
64 bytes from 142.250.189.132: icmp_seq=6 ttl=118 time=27.128 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 27.128/29.820/33.354/1.837 ms
imac_20_invitado@iMac-20 ~ %
```

4.-

```
● ○ ● imac_20_invitado — less < man nslookup — 80x24
NSLOOKUP(1) BIND9 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 is used to print just the name and requested information for a host or domain.

ARGUMENTS
    Interactive mode is entered in the following cases:
    1. when no arguments are given (the default name server will be used)
    2. when the first argument is a hyphen (-) and the second argument is the host name or Internet address of a name server.
:
```

5.-

```
● ○ ● imac_20_invitado — -zsh — 80x24
imac_20_invitado@iMac-20 ~ % ping www.google.com
PING www.google.com (142.250.189.132): 56 data bytes
64 bytes from 142.250.189.132: icmp_seq=0 ttl=118 time=30.540 ms
64 bytes from 142.250.189.132: icmp_seq=1 ttl=118 time=28.028 ms
64 bytes from 142.250.189.132: icmp_seq=2 ttl=118 time=30.035 ms
64 bytes from 142.250.189.132: icmp_seq=3 ttl=118 time=29.595 ms
64 bytes from 142.250.189.132: icmp_seq=4 ttl=118 time=33.354 ms
64 bytes from 142.250.189.132: icmp_seq=5 ttl=118 time=30.057 ms
64 bytes from 142.250.189.132: icmp_seq=6 ttl=118 time=27.128 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 27.128/29.820/33.354/1.837 ms
imac_20_invitado@iMac-20 ~ % man nslookup
Unknown locale, assuming C
imac_20_invitado@iMac-20 ~ % nslookup www.upqroo.edu.mx
Server:      8.8.8.8
Address:     8.8.8.8#53

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

imac_20_invitado@iMac-20 ~ %
```

6.-



imac\_20\_invitado --zsh -- 80x24

Non-authoritative answer:

Name: www.upqroo.edu.mx  
Address: 77.68.126.20

```
imac_20_invitado@iMac-20 ~ % ping 77.68.126.20
PING 77.68.126.20 (77.68.126.20): 56 data bytes
64 bytes from 77.68.126.20: icmp_seq=0 ttl=50 time=119.734 ms
64 bytes from 77.68.126.20: icmp_seq=1 ttl=50 time=127.867 ms
64 bytes from 77.68.126.20: icmp_seq=2 ttl=50 time=128.184 ms
64 bytes from 77.68.126.20: icmp_seq=3 ttl=50 time=126.992 ms
64 bytes from 77.68.126.20: icmp_seq=4 ttl=50 time=123.060 ms
64 bytes from 77.68.126.20: icmp_seq=5 ttl=50 time=126.970 ms
64 bytes from 77.68.126.20: icmp_seq=6 ttl=50 time=129.197 ms
64 bytes from 77.68.126.20: icmp_seq=7 ttl=50 time=120.238 ms
64 bytes from 77.68.126.20: icmp_seq=8 ttl=50 time=128.885 ms
64 bytes from 77.68.126.20: icmp_seq=9 ttl=50 time=128.438 ms
64 bytes from 77.68.126.20: icmp_seq=10 ttl=50 time=123.460 ms
64 bytes from 77.68.126.20: icmp_seq=11 ttl=50 time=130.150 ms
^C
--- 77.68.126.20 ping statistics ---
12 packets transmitted, 12 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 119.734/126.098/130.150/3.408 ms
imac_20_invitado@iMac-20 ~ %
```

7.-



imac\_20\_invitado less < man netstat -- 80x24

NETSTAT(1)

General Commands Manual

NETSTAT(1)

**NAME**

**netstat** — show network status

**SYNOPSIS**

```
netstat [-AaLnW] [-f address_family | -p protocol]
netstat [-gilns] [-v] [-f address_family] [-I interface]
netstat -i | -I interface [-w wait] [-c queue] [-abdqqrts]
netstat -s [-s] [-f address_family | -p protocol] [-w wait]
netstat -i | -I interface -s [-f address_family | -p protocol]
netstat -m [-m]
netstat -r [-Aaln] [-f address_family]
netstat -rs [-s]
netstat -B [-I interface]
```

**DESCRIPTION**

The **netstat** command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol. The second form presents the contents of one of the other network data structures according to the option selected. Using the third form, with a

8.-

9.-

10.-

```
imac_20_invitado@iMac-20 ~ % netstat -p tcp
Active Internet connections
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4       0      0 172.16.128.20.50636    17.57.144.25.5223   ESTABLISHED
tcp4       0      0 172.16.128.20.50632    mia07s56-in-f3.1.80  ESTABLISHED
tcp4       0      0 172.16.128.20.50631    71.239.117.34.bc.443 ESTABLISHED
tcp4       0      0 172.16.128.20.50630    a23-201-195-135..80  TIME_WAIT
tcp4       0      0 172.16.128.20.50628    a23-201-195-135..80  TIME_WAIT
tcp4       0      0 172.16.128.20.50603    ec2-3-219-193-15.443 ESTABLISHED
tcp4       0      0 172.16.128.20.50601    211.253.186.35.b.443 ESTABLISHED
tcp4       0      0 172.16.128.20.50586    69.173.151.96.443   ESTABLISHED
tcp4       0      0 172.16.128.20.50582    sjc06-nessy-floa.443 ESTABLISHED
tcp4       0      0 172.16.128.20.50550    209.54.182.161.443  TIME_WAIT
tcp4       0      0 172.16.128.20.50548    69.173.151.100.443 TIME_WAIT
tcp4       0      0 172.16.128.20.50543    104.19.159.19.443   ESTABLISHED
tcp4       0      0 172.16.128.20.50533    ec2-54-176-8-58..443 ESTABLISHED
tcp4       0      0 172.16.128.20.50532    104.18.22.145.443   ESTABLISHED
tcp4       0      0 172.16.128.20.50529    104.18.14.101.80    ESTABLISHED
tcp4       0      0 172.16.128.20.50528    96.46.186.182.443   ESTABLISHED
tcp4       0      0 172.16.128.20.50522    147.75.195.77.443   ESTABLISHED
tcp4       0      0 172.16.128.20.50521    746217339.dal.cd.443 ESTABLISHED
tcp4       0      0 172.16.128.20.50518    104.26.8.169.443    TIME_WAIT
tcp4       0      0 172.16.128.20.50510    104.18.20.226.80    TIME_WAIT
tcp4       0      0 172.16.128.20.50509    104.18.20.226.80    TIME_WAIT
```

11.-

12.-

```
[imac_20_invitado@iMac-20 ~ % ps
  PID TTY          TIME CMD
 1373 ttys000      0:00.07 -zsh
imac_20_invitado@iMac-20 ~ %
```

13.-

```
imac_20_invitado@iMac-20 ~ % top
imac_20_invitado@iMac-20 ~ % killall App Store
No matching processes belonging to you were found
imac_20_invitado@iMac-20 ~ % kill 1548
imac_20_invitado@iMac-20 ~ %
```

14.-

```
[imac_20_invitado@iMac-20 ~ % traceroute www.google.com
traceroute to www.google.com (142.250.189.132), 64 hops max, 52 byte packets
 1  172.16.128.1 (172.16.128.1)  14.398 ms  4.400 ms  3.744 ms
 2  192.168.109.1 (192.168.109.1)  6.037 ms  4.633 ms  4.121 ms
 3  fixed-187-188-58-130.totalplay.net (187.188.58.130)  7.096 ms  9.685 ms  7.3
00 ms
 4  10.180.58.1 (10.180.58.1)  7.516 ms  8.055 ms  6.965 ms
 5  72.14.242.148 (72.14.242.148)  20.200 ms  23.302 ms  22.314 ms
 6  * * *
 7  142.251.253.6 (142.251.253.6)  30.118 ms
 209.85.244.152 (209.85.244.152)  21.802 ms
 108.170.232.200 (108.170.232.200)  22.141 ms
 8  108.170.249.30 (108.170.249.30)  20.736 ms
 142.251.68.235 (142.251.68.235)  21.482 ms  20.489 ms
 9  142.250.212.203 (142.250.212.203)  22.562 ms
 mia09s26-in-f4.1e100.net (142.250.189.132)  21.796 ms
 142.250.212.251 (142.250.212.251)  23.022 ms
.
```

## 15.-

```
--> [imac_20_invitado@iMac-20 ~ % arp -a
? (169.254.91.153) at 1c:bf:c0:e1:92:7 on en1 [ethernet]
? (169.254.130.60) at ee:d2:ad:60:e4:e5 on en1 [ethernet]
? (169.254.194.96) at 7e:f3:92:4a:2b:9c on en1 [ethernet]
? (172.16.128.1) at e0:23:ff:b4:2e:9a on en1 ifscope [ethernet]
? (172.16.128.2) at 22:b1:c2:ea:c7:f7 on en1 ifscope [ethernet]
? (172.16.128.4) at 3c:a6:f6:b1:4:fe on en1 ifscope [ethernet]
? (172.16.128.6) at 94:e7:b:c2:96:ec on en1 ifscope [ethernet]
? (172.16.128.7) at 3c:a6:f6:a9:17:bd on en1 ifscope [ethernet]
? (172.16.128.9) at 3c:91:80:4c:c3:97 on en1 ifscope [ethernet]
? (172.16.128.10) at 3c:a6:f6:a0:4:69 on en1 ifscope [ethernet]
? (172.16.128.11) at 3c:a6:f6:a0:86:ed on en1 ifscope [ethernet]
? (172.16.128.12) at fc:e2:6c:1d:c0:2f on en1 ifscope [ethernet]
? (172.16.128.13) at 3c:a6:f6:a5:3:59 on en1 ifscope [ethernet]
? (172.16.128.14) at e6:2e:c1:d7:1c:3e on en1 ifscope [ethernet]
? (172.16.128.15) at c2:bd:84:c9:52:dc on en1 ifscope [ethernet]
? (172.16.128.16) at 3c:a6:f6:ab:dc:b6 on en1 ifscope [ethernet]
? (172.16.128.17) at 24:46:c8:82:35:ee on en1 ifscope [ethernet]
? (172.16.128.22) at 3c:a6:f6:a4:d:9 on en1 ifscope [ethernet]
? (172.16.128.23) at 3c:a6:f6:a6:55:57 on en1 ifscope [ethernet]
? (172.16.128.24) at 7e:f3:92:4a:2b:9c on en1 ifscope [ethernet]
? (172.16.128.25) at 3c:a6:f6:a5:a1:2e on en1 ifscope [ethernet]
? (172.16.128.26) at 3c:a6:f6:b0:c3:96 on en1 ifscope [ethernet]
```

## B)Preguntas

### **1.-¿Para que sirve el comando ping?**

sirve para verificar la conectividad de red entre dos dispositivos. En particular, se utiliza para enviar paquetes de datos a una dirección IP o un nombre de dominio y recibir respuestas para determinar si un dispositivo remoto está disponible y cuánto tiempo toma para que los datos viajen de ida y vuelta (latencia). Es una herramienta útil para diagnosticar problemas de conectividad en una red, identificar retrasos y comprobar la disponibilidad de hosts.

### **2.-¿Para que sirve el comando nslookup?**

se utiliza para consultar y resolver nombres de dominio (DNS, Domain Name System). Puedes usarlo para obtener información sobre direcciones IP correspondientes a nombres de dominio y viceversa. Es útil para verificar la configuración de DNS, solucionar problemas de resolución de nombres y obtener información sobre los registros de dominio.

### **3.-¿Para que sirve el comando netstat?**

muestra información detallada sobre las conexiones de red y estadísticas de red en un sistema. Puede mostrar puertos abiertos, conexiones activas, tablas de enrutamiento y más. Es útil para monitorear la actividad de red en un sistema, identificar problemas de congestión o puertos abiertos no deseados y diagnosticar conexiones de red.

### **4.-¿Para que sirve el comando tasklist?**

muestra una lista de procesos en ejecución en un sistema Windows. Proporciona información sobre los nombres de los procesos, los identificadores de proceso (PID), el uso de CPU, la memoria y más. Es útil para supervisar y diagnosticar el uso de recursos del sistema y para identificar procesos que pueden estar causando problemas.

### **5.-¿Para que sirve el comando taskkill?**

permite finalizar procesos en un sistema Windows de forma forzada. Puedes usarlo para detener procesos que no responden o que están causando problemas en el sistema. Debes especificar el PID o el nombre del proceso que deseas finalizar.

### **6.-¿Para que sirve el comando tracert?**

se utiliza para rastrear la ruta que toma un paquete de datos desde tu computadora hasta un destino específico, mostrando los saltos intermedios (routers) en el camino. Es útil para diagnosticar problemas de enrutamiento y para identificar dónde se producen demoras o pérdida de paquetes en una ruta de red.

### **7.-¿Cómo ayudan los primeros 3 comandos para detectar problemas en la red?**

- ping se utiliza para verificar la conectividad básica y detectar problemas de disponibilidad.
- nslookup ayuda a solucionar problemas de resolución de nombres de dominio.
- netstat permite monitorear la actividad de red y diagnosticar conexiones y puertos problemáticos.

En conjunto, estos comandos proporcionan información esencial para identificar y resolver problemas de red, como latencia, problemas de resolución de nombres, congestión y problemas de enrutamiento.

## C)Comandos

Investigar los siguientes comandos y anotar ejemplos prácticos:

### **atmadm**

Este comando se utiliza para administrar conexiones de red de modo asíncrono (ATM). En la mayoría de los casos, su uso es técnico y específico para la administración de redes ATM, por lo que no se usa comúnmente en situaciones cotidianas.

### **atmadm -c consulta**

### **bitsadmin**

Permite administrar tareas de transferencia de archivos en segundo plano (Background Intelligent Transfer Service). Un ejemplo práctico sería usarlo para descargar archivos grandes en segundo plano.

```
bitsadmin /transfer mi_descarga /download /priority normal http://ejemplo.com/archivo.zip  
C:\carpeta\archivo.zip
```

### **cmstp**

Utilizado para instalar o desinstalar perfiles de conexión de red. Esto es útil principalmente en entornos corporativos para implementar configuraciones de red específicas.

```
cmstp /s archivo_de_configuracion.inf
```

### **ftp**

El comando FTP se utiliza para transferir archivos entre sistemas a través del Protocolo de Transferencia de Archivos. Puedes conectarte a un servidor FTP y transferir archivos

```
ftp ejemplo.com  
get archivo_remoto.txt
```

### **getmac**

Muestra la dirección MAC (Media Access Control) de una interfaz de red. Puedes utilizarlo para obtener la dirección MAC de tu tarjeta de red, lo que es útil para la resolución de problemas de red.

`getmac`

**hostname**

Muestra el nombre del host o computadora local. Es útil para averiguar el nombre de tu propia computadora.

`hostname`

**nbstat**

Proporciona información sobre la resolución de nombres de NetBIOS en una red. Se usa para diagnosticar problemas de resolución de nombres NetBIOS.

`nbstat -A 192.168.1.1`

**net**

Este comando se utiliza para administrar varias configuraciones y recursos de red. Por ejemplo, puedes usar `net user` para administrar cuentas de usuario y `net share` para administrar recursos compartidos.

`net user nombre_usuario contraseña /add`

**net use**

Permite conectar o desconectar recursos compartidos de red en tu computadora. Por ejemplo, puedes mapear una unidad de red

`net user nombre_usuario contraseña /add`

**netsh**

Es una herramienta de configuración de red versátil que permite modificar la configuración de red, firewall, VPN y más. Por ejemplo, puedes usar `netsh` para configurar un servidor proxy.

`netsh interface ipv4 show interfaces`

**pathping**

Combina la funcionalidad de ping y tracert. Proporciona información sobre la ruta y la latencia en una red.

`pathping www.google.com`

### **rcp**

se utiliza para copiar archivos desde y hacia sistemas remotos en una red.

`rcp archivo.txt usuario@servidor:/ruta/destino/`

### **reexec**

permite ejecutar comandos en un sistema remoto. Se utiliza para iniciar programas o scripts en un sistema remoto si tienes permisos para hacerlo. Por razones de seguridad, su uso se ha vuelto menos común debido a posibles vulnerabilidades.

`reexec servidor comando`

### **route**

se utiliza para ver y manipular la tabla de enrutamiento en sistemas Windows. Puedes utilizarlo para agregar, eliminar o modificar rutas de red. Por ejemplo, para agregar una ruta predeterminada a través de una puerta de enlace específica

`route add 0.0.0.0 mask 0.0.0.0 192.168.1.1`

### **rpcping**

Este comando se utiliza para realizar pruebas de ping a servicios RPC (Remote Procedure Call). Es útil para verificar la conectividad y la disponibilidad de servicios RPC en sistemas remotos.

`rpcping -s servidor`

### **rsh**

Similar a reexec, el comando rsh (Remote Shell) permite ejecutar comandos en sistemas remotos, pero su uso también ha disminuido debido a preocupaciones de seguridad. Puede ejecutar comandos en un sistema remoto si tienes permisos para hacerlo.

`rsh servidor comando`

### **tcmsetup**

Este comando se utiliza para configurar la autenticación de Trusted Platform Module (TPM) en sistemas Windows. Es una herramienta técnica utilizada para configurar la seguridad de hardware en sistemas compatibles con TPM.

`tcmsetup -v -f -b 123456`

**telnet**

El comando telnet se utiliza para conectarse a otros dispositivos o servidores a través de una sesión de terminal. Permite acceder a sistemas remotos para administrarlos o realizar pruebas

telnet servidor

**tftp**

El Protocolo de Transferencia de Archivos Trivial (TFTP) se utiliza para transferir archivos de manera sencilla desde y hacia sistemas remotos. Es una forma simple de copiar archivos en sistemas en red.

```
tftp -i servidor GET archivo.txt
```