SERVICIOS
INTRODUCCIÓN

📕 Descargar PDF

ÍNDICE

SERVICIOS

Un **servicio** es una **aplicación que se ejecuta en segundo plano**. Los servicios utilizan una interfaz de programación de aplicaciones (API) para manejar las tareas principales sin la interacción del usuario. Por ello, normalmente utilizan archivos de bitácora (logs) para registrar mensajes de error y de actividad del servicio.Los servicios brindan funciones básicas del sistema operativo, como servicio web, registro de eventos, servicio de archivos, ayuda y soporte, impresión, criptografía e informes de errores y actividad.

El administrador de servicios es un programa con el que se pueden realizar diferentes tareas sobre los servicios. En concreto, un **servicio** se puede **iniciar, detener y reiniciar**. También podemos **mostrar el estado** del servicio, **habilitarlo y deshabilitarlo**. Los servicios se pueden configurar para que se inicien cuando se inicie el equipo, antes de abrir una sesión de usuario o bien puede iniciarlos un usuario u otra aplicación si tiene los permisos adecuados.

MODELO CLIENTE-SERVIDOR

Normalmente, los servicios siguen el llamado **modelo cliente-servidor**. En este modelo de servicio tenemos dos componentes:

- **Servidor**
  Proceso que ofrece un servicio. El servidor siempre se está ejecutando, normalmente en segundo plano, esperando a recibir (escuchando) peticiones de los clientes. Una vez recibida la petición ejecuta la tarea asociada. El servidor se puede configurar para ejecutar las peticiones de los clientes de dos formas diferentes:
  - **Servidor dedicado**
    Cuando recibe una petición de un cliente la sirve y hasta que no acabe de dar el servicio no acepta ninguna petición. Es ineficiente.
  - **Servidor no dedicado**
    Cuando recibe una petición crea otro proceso (hijo) que realmente sirve la petición al cliente. De esta manera el proceso inicial (padre) puede recibir más peticiones de otros clientes. Es la forma habitual de trabajar.
- **Cliente**
  Proceso que solicita un servicio a un servidor. El cliente puede ser un proceso ejecutado bajo demanda o bien un proceso en ejecución en segundo plano. Un proceso cliente puede ejecutarse en una máquina remota o en la misma maquina en la que se ejecuta el proceso servidor.

En este modelo cliente-servidor, normalmente los procesos servidores se ejecutan en máquinas dedicadas para prestar el servicio. Por este motivo, la mayoría de las veces llamamos servidor a la maquina y no al proceso. En realidad, una maquina podría tener varios procesos de tipo servidor alojados en ella como por ejemplo un servidor web, un servidor ftp y un servidor ssh.

AMBITO DE EJECUCIÓN

Los **servicios** se pueden clasificar **según el ambito de ejecución** del servidor como:

- **Servicios locales**
  El servidor ofrece sus servicios solo a los clientes que se ejecutan en la misma máquina.
- **Servicios de red**
  El servidor ofrece sus servicios a cualquier cliente. Se pueden clasificar en servicios de:
  - **Intranet**
    El servidor ofrece sus servicios solo a los equipos de su misma red. Un ejemplo de este tipo sería el servicio de asignación automática de direcciones (DHCP), o un servicio de resolución de nombres de tipo caché (DNS caché).
  - **Internet**
    El servidor ofrece sus servicios a cualquier equipo de cualquier red a la que este interconectado. Un ejemplo de este tipo sería el servicio de páginas web (HTTP), o un servicio global de resolución de nombres (DNS), o un servicio de correo electrónico (IMAP, SMTP, POP3).

Algunos servicios se pueden configurar indistintamente como de tipo Intranet o Internet en función de las necesidades de la empresa.

CONEXIÓN TCP-IP

Como hemos indicado servidor y cliente son en realidad procesos en ejecución y por tanto se encuentran en la capa de aplicación. Para realizar la comunicación entre el cliente y el servidor se utiliza a nivel de aplicación el concepto de **conexión**. Una conexión esta definida por **2 sockets**: **socket origen** y **socket destino**. Un **socket** esta definido por la pareja **IP:PUERTO**.

En la pila TCP-IP las **IPs** se definen en **nivel de red** (protocolo IP), mientras que los **puertos** se definene en el nivel de **transporte** (protocolo TCP o UDP). **Cada proceso en ejecución en una red debe tener asociado un puerto** de comunicaciones a nivel de transporte. Estos **puertos** se pueden **clasificar** en:

Nombre	Puertos	Descripción
Sistema	[0...1023]	Denominados también WKP. Son puertos asignados por la IANA para identificar los principales servicios ofrecidos en Internet y en los sistemas operativos de Red
Usuario	[1024...49151]	Denominados también registrados. Son puertos asignados por la IANA para identificar servicios específicos de determinadas tecnologías o empresas.
Dinámicos	[49152...65535]	Denominados también efímeros. No estan asignados por la IANA y se utilizan para identificar servicios privados o procesos con propósito temporal. Estos puertos son utilizados principalmente por los procesos clientes para identificarse con el servidor.

A los puertos definidos por la IANA (sistema y usuario) se les denomina puertos **standard**. Puedes consultarlos en la siguiente url: **IANA standard**

conexión

Un navegador web (cliente) alojado en un equipo con dirección IP 192.168.10.20 solicita una pagina web a través del protocolo HHTP (puerto 80) al servicio de paginas web (servidor) alojado en un equipo con dirección IP 192.168.10.250. La **conexión** que define la comunicación estaría definida por la cuadrupla:

socket origen	socket destino
192.168.10.20:50001	192.168.10.250:80

ESTADOS DE UNA CONEXIÓN

Las conexiones TCP tienen un ciclo de vida de 3 fases:

- Establecimiento de la conexión
- Transmisión de datos
- Finalización de la conexión

Las conexiones UPD tienen un ciclo de vida de 1 fase:

- Transmisión de datos.

Durante el ciclo de vida de una conexión los puertos que forman parte de ella se pueden encontrar en diferentes estados:

Fase	Estado	Descripción
Establecer	Listen	Se ha abierto el puerto y está a la espera de conexiones. Es el estado típico de un proceso servidor
	SYN-Sent	Se ha enviado una petición de conexión
	SYN-Received	Se ha recibido una petición de conexión
Transmitir	Established	Se ha establecido una conexión. Se pueden enviar y recibir datos
Finalizar	Fin-Wait-1	Se ha solicitado la finalización de la conexión
	Close-Wait	Se ha recibido una solicitud de finalización
	Closing	Se esta realizando la finalización
	Fin-Wait-2	Se informa de que se está a la espera de finalización
	Last-Ack	Se encuentra en espera del ACK de finalización
	Time-Wait	Se espera a la recepción del Last-Ack por el destino
	Closed	Se ha cerrado la conexión

NETSTAT WINDOWS

En los sistemas windows podemos comprobar el estado de las conexiones mediante el comando `netstat`

netstat

```
  netstat -ano
  Proto  Dirección local          Dirección remota     Estado          PID
  TCP    0.0.0.0:135              0.0.0.0:0              LISTENING       1456
  TCP    0.0.0.0:445              0.0.0.0:0              LISTENING       4
  TCP    127.0.0.1:5939           0.0.0.0:0              LISTENING       5012
  TCP    192.168.1.138:139        0.0.0.0:0              LISTENING       4
  TCP    192.168.1.138:49712      23.196.96.159:80       CLOSE_WAIT      8728
  TCP    192.168.1.138:57012      192.168.1.128:8009     ESTABLISHED     13260
  TCP    [::]:135                 [::]:0                 LISTENING       1456
  TCP    [::]:445                 [::]:0                 LISTENING       4
  TCP    [2a0c::5035:277f]:10584  [2a00::c07::bc]:5228   ESTABLISHED     13260
  TCP    [2a0c::5035:277f]:49693  [2603::400]:443        ESTABLISHED     4848
  UDP    0.0.0.0:60196          *:*                                    2952
  UDP    0.0.0.0:62285          *:*                                    3248
  UDP    127.0.0.1:52111        127.0.0.1:52111                        4708
  UDP    127.0.0.1:52112        127.0.0.1:52113                        4564
  UDP    127.0.0.1:52113        127.0.0.1:52112                        4564
  UDP    [::]:123               *:*                                    8424
...
```

Se han omitido resultados en la salida de netstat y se han utilizado solo algunos a modo de ejemplo. Se puede comprobar como netstat distingue entre conexiones TCP y UDP. Las conexiones UDP solo tienen una fase por lo que el puerto carece de estado. La salida incluye direcciones IPv4 e IPv6. El campo PID nos indica el identificador del proceso que ha ejecutado la conexión.

Cuando en el campo dirección local la IP es any-address (0.0.0.0 en IPv4 ó [::] en IPv6) nos indica que el puerto esta siendo utilizado para cualquier dirección IP que tenga configurado el host.

Cuando en dirección remota la IP es any-address nos está indicando que no hay conexión activa y se puede comprobar que el estado es LISTENING para TCP. En UDP esta situación se refleja con dirección remota *:*.

obtener programa que ejecuta la conexión con tasklist

Podemos obtener el programa que ha ejecutado la conexión con el comando `tasklist`

```
  sintaxis
  tasklist /FI "PID eq [nºPID]"
  tasklist /FI "PID eq 1456"
  Nombre de imagen               PID Nombre de sesión Núm. de ses Uso de memoria
  ========================= ======== ================ =========== ==============
  svchost.exe                   1456 Services                   0      15.676 KB
```

obtener programa que ejecuta la conexión con netstat

Podemos obtener el programa que ha ejecutado la conexión añadiendo la opción -b a netstat

```
netstat -anob
Proto  Dirección local        Dirección remota       Estado          PID
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       1456
RpcSs
[svchost.exe]
....
```

netstat filtrar la salida

Podemos filtar las conexiones añadiendo un filtrado con `findstr`

Obtener solo conexiones IPv4

```
netstat -ano | findstr /V "["
```

Obtener solo conexiones IPv6

```
netstat -ano | findstr "["
```

Obtener solo conexiones TCP

```
netstat -ano | findstr "TCP"
```

Obtener solo conexiones IPv4 que sean UDP

```
netstat -ano | findstr /V "[" | findstr "UDP"
```

NETSTAT LINUX

En los sistemas linux podemos comprobar el estado de las conexiones mediante el comando `netstat` perteneciente al paquete `net-tools`. Las opciones del comando son ligeramente diferentes a las de windows.

netstat

```
netstat -an
Proto  Recv-Q    Send-Q  LocalAddress            ForeingAddress    State
tcp         0         0  10.0.2.15:53            0.0.0.0:*         LISTEN
tcp         0         0  0.0.0.0:22              0.0.0.0:*         LISTEN
tcp6        0         0  fe80::a00:27ff:fe64::53  :::*             LISTEN
udp         0         0  10.0.2.15:53            0.0.0.0:*
udp         0         0  10.0.2.15:32959         100.100.1.1:53    ESTABLISHED
...
```

Se han omitido resultados en la salida de netstat y se han utilizado solo algunos a modo de ejemplo.

obtener programa que ejecuta la conexión con netstat

Podemos obtener el programa que ha ejecutado la conexión añadiendo la opción -p a netstat

```
netstat -anp
Proto  Recv-Q    Send-Q  LocalAddress            ForeingAddress    State   PID/Program name
tcp         0         0  10.0.2.15:53            0.0.0.0:*         LISTEN  412/named
....
```

En los sistemas linux podemos también comprobar el estado de las conexiones mediante el comando `ss` perteneciente al paquete `iproute2`.

ss

Obtener conexiones IPv4

```
ss -an -4
Proto  State     Recv-Q    Send-Q    Local Address:Port      Peer Address:Port
udp    UNCONN    0         0         10.0.2.15:53            0.0.0.0:*
udp    ESTAB     0         0         10.0.2.15:39784         100.90.1.1:53
tcp    LISTEN    0         128       0.0.0.0:22              0.0.0.0:*
...
```

otras opciones ss

Obtener conexiones IPv6

```
ss -an -6
```

Obtener el proceso que realiza la conexión

```
ss -an -p
```