

Programación de comunicaciones en red (Programación de servicios y procesos)

Comunicación entre aplicaciones

La comunicación entre aplicaciones es la base del funcionamiento de todo sistema distribuido, recordemos que un sistema distribuido consistía en utilizar sistemas en los que los componentes hardware o software se encuentran en ordenadores unidos mediante la red y se comunican mediante el paso de mensajes. Es importante indicar que los sistemas distribuidos no están sincronizados.

En el proceso de comunicación intervienen los siguientes elementos:

- Mensaje
- Emisor
- Receptor
- Canal
- Paquete de información
- Protocolo de comunicación

Los protocolos de comunicación son (capas del modelo TCP/IP):

- Capa física
 - Capa de Internet
 - Capa de transporte
 - Capa de aplicación
-

Más información

- <https://www.jesusninoc.com/07/09/9-gestion-de-la-red-en-powershell/2/#Capa-fsica>
 - <https://www.jesusninoc.com/07/09/9-gestion-de-la-red-en-powershell/3/#Capa-de-Internet>
 - <https://www.jesusninoc.com/07/09/9-gestion-de-la-red-en-powershell/4/#Capa-de-transporte>
 - <https://www.jesusninoc.com/07/09/9-gestion-de-la-red-en-powershell/5/#Capa-de-aplicacin>
-

Roles cliente y servidor

Una arquitectura cliente-servidor consiste en una comunicación en la que un proceso cliente solicita un recurso a otro equipo denominado servidor.

La comunicación debe ser iniciada por uno de los procesos que se denomina programa «cliente». El segundo proceso espera a que otro inicie la comunicación, por este motivo se denomina programa «servidor».

Más información

- <https://www.jesusninoc.com/10/13/server-and-client-datagram-socket-udp/>
- <https://www.jesusninoc.com/10/11/server-and-client-sockets-tcp-2/>
- <https://www.jesusninoc.com/01/30/enviar-un-mensaje-udp-a-un-ordenador-desde-powershell/>
- <https://www.jesusninoc.com/10/07/crear-un-cliente-y-un-servidor-tcpip-con-powershell/>

Ejercicios

- <https://www.jesusninoc.com/01/30/realizar-una-comunicacion-entre-dos-ordenadores-por-udp-desde-powershell/>
-

Elementos de programación de aplicaciones en red. Librerías

Los elementos de programación dependen del lenguaje de programación utilizado además del sistema operativo para el que se programa.

Sockets

Un socket representa un canal de comunicación entre un emisor y un receptor (o entre un cliente un servidor).

Un socket es un proceso o hilo existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

Son los mecanismos que permiten la comunicación entre aplicaciones, proporcionan una abstracción de la pila de protocolos.

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es Transmission Control Protocol; una alternativa común a este es User Datagram Protocol.

La capa de transporte TCP/IP garantiza que los paquetes

lleguen en secuencia y sin errores, al intercambiar la confirmación de la recepción de los datos y retransmitir los paquetes perdidos. Los protocolos de capa de transporte de este nivel son el Protocolo de control de transmisión (TCP) y el Protocolo de datagramas de usuario (UDP). El protocolo TCP proporciona un servicio completo y fiable. UDP proporciona un servicio de datagrama poco fiable.

Creación de sockets

Para crear un socket es necesaria una dirección IP y un puerto.

La dirección IP es un número que identifica de forma única a cada máquina de la red y que sirve para comunicarse con ella. El puerto es un número que identifica a un socket dentro de un sistema operativo.

Enlazado y establecimiento de conexiones

Para que dos procesos puedan comunicarse entre sí es necesario que se cumplan ciertos requisitos:

- Que un proceso sea capaz de localizar al otro.
- Que ambos procesos sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad.

Para ello son necesarios los dos recursos que originan el concepto de socket:

- Un par de direcciones del protocolo de red (dirección IP, si se utiliza el protocolo TCP/IP), que identifican al ordenador de origen y la remota.
- Un par de números de puerto, que identifican a un programa dentro de cada ordenador.

Ejercicios

- [Analizar direcciones IP para detectar puertos abiertos \(versión 1\)](#)
 - [Analizar direcciones IP para detectar puertos abiertos \(versión 2\)](#)
-

Utilización de sockets para la transmisión y recepción de información

Un socket es un proceso o hilo existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

Ejercicios

- [Comprobar si un puerto TCP está abierto](#)
- [Server and client \(Sockets TCP\)](#)
- [Multiserver and clients \(Sockets TCP and threads\)](#)
- [Enviar una imagen entre un cliente y un servidor con Java \(Sockets TCP\)](#)
- [Enviar mensajes constantemente entre cliente y servidor con Java](#)
- [Server and client \(Datagram Socket UDP\)](#)
- [MulticastSocket \(Datagram Socket UDP\)](#)

Más información

- <https://www.jesusninoc.com/03/17/enviar-un-sonido-entre-un-cliente-y-un-servidor-con-powershell/>
 - <https://www.jesusninoc.com/03/17/crear-una-tarea-programada-en-la-que-un-servidor-escucha-los-mensajes-que-son-mandados-por-parte-de-los-cliente-con-powershell/>
 - <https://www.jesusninoc.com/03/17/crear-una-tarea-programada-en-la-que-un-servidor-escucha-los-mensajes-que-son-mandados-por-parte-de-los-cliente-con-powershell-la-comunicacion-entre-el-servidor-y-el-cliente-se-realiza-a-traves-de-in/>
-

Programación de aplicaciones cliente y servidor

La programación depende del lenguaje de programación y del sistema operativo para el que se programe.

Ejercicios

- [Conectar con una URL en Java](#)
 - [Conectar con una URL en Java cambiando el User-Agent](#)
 - [Realizar petición HTTP mediante el método GET](#)
 - [Realizar petición HTTP mediante el método POST](#)
 - [Realizar petición HTTP mediante el método POST utilizando librerías de Apache](#)
-

Utilización de hilos en la programación de aplicaciones en red

Los sockets además de ser utilizados por los procesos también se pueden gestionar por parte de hilos, lo que permite concurrencia (varios sockets abiertos y gestionados a la vez) y responder a múltiples peticiones.

Ejercicios

- [Multiserver and clients \(Sockets TCP and threads\)](#)
-