

Redes Informáticas II (IS18)

Evaluación de Prácticas.

Comunicación entre procesos mediante Sockets

SOFTWARE A UTILIZAR:

1. Sistema operativo Linux.
2. Compilador gcc de lenguaje C o intérprete de Python.

INTRODUCCIÓN TEÓRICA

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

NOTA

Los recursos u objetos que actúan como entrada o salida de un comando HTTP están clasificados por su descripción MIME. De esta forma, el protocolo puede intercambiar cualquier tipo de dato, sin preocuparse de su contenido. La transferencia se realiza en modo binario, byte a byte, y la identificación MIME permitirá que el receptor trate adecuadamente los datos.

Las principales características del protocolo HTTP son:

- Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor: GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto (por ejemplo, la fecha de modificación de un documento HTML).
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.
- No mantiene estado. Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.

- Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

NOTA

HTTP se diseñó específicamente para el World Wide Web: es un protocolo rápido y sencillo que permite la transferencia de múltiples tipos de información de forma eficiente y rápida. Se puede comparar, por ejemplo, con FTP, que es también un protocolo de transferencia de ficheros, pero tiene un conjunto muy amplio de comandos, y no se integra demasiado bien en las transferencias multimedia.

Existen dos versiones del protocolo, la 1.0 y la 1.1. La descrita en la brevemente en este punto es la versión 1.0. Para conocer más sobre este protocolo, por ejemplo la estructura de los mensajes, utilícense las referencias [1], [2] y [3]. También es interesante capturar algunas tramas de una comunicación HTTP con programas de análisis de tráfico como Ethereal.

DESARROLLO DE LA PRÁCTICA.

Se propone el desarrollo de un servidor de archivos que se implemente utilizando sockets.. El alumno deberá desarrollar la aplicación de modo que los formatos de los mensajes se basen en el protocolo HTTP.

El alumno deberá implementar la aplicación utilizando como lenguajes de implementación python o c. Todo aquello que no esté especificado se deja a la elección del alumno, que tendrá que justificar sus criterios para esas elecciones.

Especificaciones:

- Aplicación: Servidor concurrente (utilizando hilos o procesos) de archivos.
- Puerto: 18080
- Cliente: No es necesario desarrollar ninguno. Para comprobar el funcionamiento se utilizarán navegadores estándar o la aplicación telnet.
- Mensajes: Utilizar HTTP v1.0. Si el alumno desea implementar alguna característica no soportada por la versión 1.0 puede utilizar la versión 1.1. Como mínimo deberá implementarse el comando GET y los códigos de estado del servidor 200 (Ok), 400 (Bad Request, error de sintaxis en la petición), 404 (Not Found, la url solicitada no existe) y 501 (Not implemented, el servidor no tiene implementado ese comando).
- Estructura de Carpetas: En la carpeta en la que esté el ejecutable de la aplicación deberá haber otra carpeta con nombre prefijado, por ejemplo "datos", en la que estarán los archivos disponibles para ser enviados.
- El servidor deberá responder con una página html con la lista de archivos en la carpeta de "datos" y con enlaces a estos siempre que el comando GET pida el path raíz "/". Cuando se solicite un archivo concreto deberá responderse con el mensaje HTTP adecuado (que incluya al archivo o el error que ha aparecido según sea el caso)

EVALUACIÓN.

La evaluación será el día **19 de Septiembre a las 16:00h en el laboratorio de prácticas TD1108**. Los alumnos deberán llevar, para entregar, un disquete con el código de la aplicación así como un esquema de la implementación. La evaluación consistirá en una entrevista personal en la que el alumno deberá mostrar el funcionamiento de su aplicación así como responder a las cuestiones que sobre el funcionamiento y sobre sockets se le planteen.

REFERENCIAS.

- [1] <http://cdec.unican.es/libro/HTTP.htm>
Descripción básica del protocolo http en castellano.
- [2] [http:// http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/x49.html](http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/x49.html)
Artículo sobre el protocolo http en castellano. Introduce las diferencias entre v1.0 y v1.1
- [3] <http://www.w3.org/Protocols/>
RFC y otros sobre el protocolo http.
- [4] <http://ditec.um.es/laso/docs/tut-tcpip/>
Tutorial y descripción técnica de TCP/IP
- [5] <http://ditec.um.es/laso/docs/>
Otros documentos de IPC (interprocess communication)
- [6] <http://www.arrakis.es/~dmrq/beej/home.htm>
Guía Beej de Sockets en español.
(y la original en: <http://www.ecst.csuchico.edu/~beej/guide/net/>)
- [7] <http://www.developerweb.net/forum/viewforum.php?f=53>
UNIX Sockets FAQs
- [8] <http://www.scit.wlv.ac.uk/~jphb/comms/sockets.html>
Sockets Programming. Resumen de las funciones.
- [9] <http://lowtek.com/sockets/>
Spencer's Socket Site: Network Programming with Sockets. Reunión de direcciones útiles
- [10] http://www3.uji.es/~ochera/curso_2003_2004/e52/
Material variado, con ejemplos de comunicaciones unicast.