

Redes de Computadores

Tarea 1: Implementación de cliente http simple en C++14.

Escuela de Ingeniería Civil Informática
Universidad de Valparaíso

1 Descripción del trabajo a realizar

1.1 Captura de tráfico HTTP

A través del programa Wireshark, su grupo debe capturar el tráfico entre su computador y un servidor web, para analizar cuáles son las acciones y comandos HTTP que se utilizan en dicha conexión.

Para lograr el objetivo mencionado, debe explicar, en no más de una hoja, que es HTTP (RFC2616) y HTML (<https://www.w3.org/html/>) y cómo están relacionados. Para esto, debe leer y estudiar los respectivos estándares de estos protocolos.

Cada página analizada, debe ser presentada a través de un esquema, sin detalles, sólo con los datos de interés: nombre de la página, de las imágenes y objetos que pudiese tener.

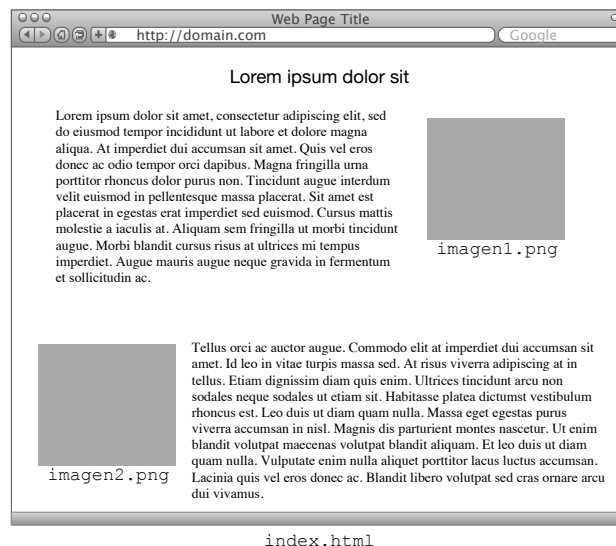


Figura 1

Con Wireshark, describir claramente al handshake inicial hacia el servidor y las tramas de los comandos HTTP relacionados con la solicitud y respuestas de objetos de la respectiva página. Además, debe mencionar todas las transacciones HTTP que pueda tener la página y no ser completamente visibles al usuario.

Debe analizar dos sitios web, sólo con su “página de entrada”. Por ejemplo, <http://web.observador.cl>, <http://www.munivalpo.cl>. Usted debe buscar otras páginas. La única condición es que sean páginas no seguras (http://), para que pueda leer el contenido de las tramas en Wireshark.

1.2 Implementación de cliente HTTP basado en texto

En base al conocimiento adquirido en el punto anterior, deberá implementar un cliente HTTP basado en texto (**tcclienthttp**), el que permitirá guardar el código HTML de una página web en un archivo especificado.

Ejemplo de uso con parámetros:

```
./tthttp -s www.uv.cl -p 80 -o archivo.txt
```

Ejemplo de uso sin parámetros:

```
./tthttp
```

Uso: `./http -s www_server -p puerto_tcp -o archivo`

Descripción:

```
-s IP o nombre del servidor web
-p Puerto TCP de conexión del servidor
-o Archivo de salida
```

Para generar un programa apropiado, puede basarse en los códigos disponibles en el aula virtual “Cliente/Servidor de Eco”

1.3 Condiciones generales de implementación

El software debe implementar en C++14, en Sistema Operativo Linux y debe compilar sin warnings en los computadores de los laboratorios. Puede utilizar todas las facilidades de este lenguaje para resolver el problema como la *Standard Library Template*[†] y *boost c++ libraries*[‡]. Bajo ningún punto de vista, debe utilizar librerías que ya implementen el problema.

Deberá crear un repositorio en github de su proyecto. La corrección se realizará clonando dicho repositorio. Por ejemplo, si el repositorio del *alumno Zacarías Flores del Campo* es “zfc” y creó un proyecto denominado “*fracaso-proy*”, cuyo ejecutable es “*fracaso*”, la secuencia de comando que se realizará será:

```
git clone https://github.com/zfc/fracaso-proy.git
cd fracaso
make clean
make
./fracaso
```

En el mismo repositorio, bajo el directorio “documentos”, debe subir un informe PDF del trabajo realizado. La documentación de la arquitectura de su software, debe tener, por lo menos, la Vista Lógica y la Vista de Procesos del modelo 4+1. En un archivo README.md, debe especificar los integrantes del grupo y sus respectivos correos. Las copias serán penalizadas con nota 1.0.

Adicionalmente, deberá llenar un formulario en el aula virtual donde especificará la URL del repositorio git utilizado.

[†] <http://www.cplusplus.com/reference/stl>

[‡] <https://www.boost.org>