



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Práctica 5: Aplicación WGet

Estudiantes:

Morales Hernández Carlos Jesús

Ramirez Hidalgo Marco Antonio

Grupo: 3CM17

Materia: Aplicaciones para comunicaciones en red

Docente: Axel Ernesto Moreno Cervantes

Carrera: Ingeniería en Sistemas Computacionales

Martes, 13 de junio del 2022

Índice

1. Introducción	1
1.1. WGet	1
1.2. URL	1
2. Desarrollo experimental	2
2.1. Wget.java	2
2.1.1. Función procesoURL	2
2.1.2. Función writeData	2
2.1.3. Función isDir	3
2.1.4. Función main	4
3. Conclusiones	4
Bibliografía	5

Práctica 5: Aplicación WGet

1. Introducción

1.1. WGet

GNU Wget es una herramienta libre que permite la descarga de contenidos desde servidores web de una forma simple. Su nombre deriva de World Wide Web (w), y de «obtener» (en inglés get), esto quiere decir: obtener desde la WWW. Fue escrito originalmente por Hrvoje Nikšić y por ser un proyecto de software libre tiene una gran cantidad de colaboradores directos e indirectos. Actualmente admite descargas mediante los protocolos HTTP, HTTPS y FTP. Entre las características más destacadas que ofrece Wget está la posibilidad de fácil descarga de mirrors (repositorios) complejos de forma recursiva, conversión de enlaces para la visualización de contenidos HTML localmente, soporte para proxies, etc. Su primera versión se lanzó en 1996, coincidiendo con el boom de popularidad de la web. Es un programa utilizado a través de línea de comandos, principalmente en sistemas tipo UNIX, especialmente en GNU/Linux. Escrito en el lenguaje de programación C, Wget puede ser fácilmente instalado en sistemas derivados de UNIX, y ha sido portado a muchas interfaces gráficas de usuario (GUI) y aplicaciones gráficas de descarga como Gwget2 para GNOME, wGetGUI3 y VisualWget4 para Microsoft Windows, Wget 1.10.2r25 para Mac OS X.

1.2. URL

URL significa Uniform Resource Locator y es la dirección única y específica que se asigna a cada uno de los recursos disponibles de la World Wide Web para que puedan ser localizados por el navegador y visitados por los usuarios.

Una URL no es más que una dirección que es dada a un recurso único en la Web. En teoría, cada URL valida apunta a un único recurso. Dichos recursos pueden ser páginas HTML, documentos CSS, imágenes, etc. En la práctica, hay algunas excepciones, siendo la más común una URL apuntando a un recurso que ya no existe o que ha sido movido. Como el recurso representado por la URL y la URL en si son manejadas por el servidor Web, depende del dueño del servidor web manejar ese recurso y su URL asociada adecuadamente.

Las direcciones URL tienen las presentes características:

- **Exactitud.** Las URL deben escribirse con precisión para que arrojen el resultado esperado. Cualquier error en la cadena impedirá recobrar la información deseada.
- **Secuencia.** Las URL son una secuencia de detalles ordenados, que deben introducirse en secuencia precisa, como una dirección postal.
- **Unicidad.** Cada URL es única para un recurso o página de la Red, no existen dos iguales ni dos diferentes que conduzcan al mismo exacto elemento (aunque puede haber muchas versiones de un mismo elemento, supongamos, una fotografía, o puede haber muchas páginas usando una misma dirección URL para recuperarla).

2. Desarrollo experimental

Para lograr desarrollar esta práctica se utilizó un archivo que fue nombrado como Wget.java. En seguida se describe la clase con sus respectivos métodos que hicieron posible la implementación de la aplicación WGet.

2.1. Wget.java

En esta clase se engloba todo lo correspondiente a la práctica, la cual extiende un hilo, esto para que se pueda en todo momento extraer los datos provenientes del URL. A continuación se presentan los métodos que se utilizaron dentro de la clase

2.1.1. Función procesoURL

Código

```
public void procesoUrl()
{
    if(url.getFile().indexOf(str: ".") == -1){
        isDir();
    }
    else{
        writeData(new File(path + getName(url.getFile())));
    }
}
```

Descripción

Esta función se utiliza principalmente para procesar la URL y ver mandar a llamar a dos métodos, el cual básicamente nos dice si nuestro URL tiene algún directorio dentro de la carpeta a descargar o simplemente trae archivos para llamar al método writeData.

2.1.2. Función writeData

Código

```

public void writeData(File file)
{
    try
    {
        System.out.println("Descargando el archivo proveniente de: " + file.getName());
        DataOutputStream dos = new DataOutputStream(new FileOutputStream(file));
        DataInputStream dis = new DataInputStream(url.openStream());
        byte[] bytes = new byte[65535];
        int read = dis.read(bytes);
        while(read != -1)
        {
            dos.write(bytes, off: 0, read);
            read = dis.read(bytes);
        }
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
}

```

Descripción

Esta función se encarga de descargar cada uno de los archivos que se encuentren dentro de la URL, aquí básicamente se utiliza un objeto DOS y DIS para poder escribir y leer cada uno de los bytes provenientes del archivo.

2.1.3. Función isDir**Código**

```

private void isDir()
{
    String pare = "PARENTDIR";
    File file = new File(path + getName(url.getFile()));
    if(file.mkdir())
    {
        path = file.getAbsolutePath() + "\\";
        try
        {
            DataInputStream dis = new DataInputStream(url.openStream());
            byte[] bytes = new byte[65535];
            int read = dis.read(bytes);
            String str = "";
            while(read != -1)
            {
                str += new String(bytes, offset: 0, read);
                read = dis.read();
            }

            int index = str.indexOf(pare);
            if(index == -1)
            {
                System.out.println(getName("Error " + url.getFile()) + "No encontrado");
                System.out.println(x: "Obteniendo bytes.");
                pool.execute(new Wget(url.toString(), path, pool, opcion: 2));
                return;
            }
            index = str.indexOf(str: "href", index);
            pool.execute(new Wget(url.toString(), path, pool, opcion: 2));
        }
    }
}

```

Descripción

Esta función hace lo mismo que la anterior, pero simplemente es cuando el archivo que tiene es otra dirección, así que extrae los bytes provenientes de la siguiente URL. Cabe recalcar que esta práctica permite solamente hasta dos procesos recursivos, ósea que simplemente se pueden descargar archivos que se encuentren en la carpeta raíz y una mas

2.1.4. Función main**Código**

```
Run | Debug
public static void main(String[] args)
{
    if(args.length > 0){
        Wget wg = new Wget(args[0], path: "", opcion: 1);
    }
    else{
        Scanner scanner = new Scanner(System.in);
        System.out.println(x: "Ingresa la URL:");
        String url = scanner.nextLine();
        Wget wg = new Wget(url, path: "", opcion: 1);
    }
}
```

Descripción

La función main se pueden recibir el url mediante la compilación del archivo WGet.java o introduciéndola dentro de la terminal, se crea un objeto Wget y simplemente se llama al constructor y empieza la aplicación Wget.

3. Conclusiones

En esta práctica, pudimos entender un poco como funciona de manera profunda los programas que incluyen sistemas de descargas dada una ruta. La implementación de un hilo para las descargas y la conexión mediante URL Connection, fueron algunos tópicos que fueron utilizados para lograr realizar una conexión con el servidor.

Fue complicado en un principio lograr descargar los archivos que estaban enlistados en la página de los servidores, porque era necesario realizar un manejo de strings de forma profunda para obtener si lo que ofrecía era un archivo o una carpeta, o en su defecto algún otro link de escape. De esta manera, de manera similar a la práctica del Drive, pudimos mapear todos los recursos que ofrecía el servidor de manera recursiva y poder ir descargando y creando las carpetas con mkdir() el árbol que se iba generando

Bibliografía

[1] Instituto Tecnológico de Roque. *Protocolos de la capa de aplicación*. [En línea]. Disponible en: <http://itroque.edu.mx/cisco/cisco1/course/module10/10.1.2.5/10.1.2.5.html>

[2] Microsoft. (2022, marzo 12). *Windows Sockets: Sockets de datagramas*. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/cpp/mfc/windows-sockets-datagram-sockets?view=msvc-170>