

# Informe de instalación de Elinks

Leandro Spagnolo, Juan Bertoni, Santiago Calligari, Julián Cabrera

4 de agosto de 2020

## Resumen

En este informe se detalla el proceso seguido para la instalación del navegador de Internet para consola “Elinks”. En este proceso, básicamente instalamos todos los programas y todas las herramientas necesarias para aprovechar al máximo Elinks, siempre bajo las restricciones que impone no contar con una interfaz gráfica; y luego descargamos el repositorio con el código fuente de Elinks, y lo compilamos e instalamos.

## Índice

<b>1. Instalación de los programas básicos</b>	<b>1</b>
<b>2. Instalación de las dependencias</b>	<b>2</b>
2.1. Instalación de Lua . . . . .	3
2.2. Instalación de zlib . . . . .	4
2.3. Instalación de bzip2 . . . . .	4
2.4. Instalación de LZMA Utils . . . . .	4
2.5. Instalación de Open SSL . . . . .	4
2.6. Instalación de pkg-config . . . . .	4
2.7. Instalación de expat . . . . .	5
2.8. Instalación de TRE . . . . .	5
2.9. Instalación de libsmclient . . . . .	5
2.10. Instalación de libidn . . . . .	5
2.11. Instalación de SpiderMonkey . . . . .	6
<b>3. Instalación de Elinks</b>	<b>6</b>

## 1. Instalación de los programas básicos

Para poder obtener código fuente de repositorios desde internet, utilizamos GIT. Este fue instalado ingresando el comando `$ sudo apt-get install git`.

También ingresamos `$ sudo apt-get install build-essential`, que es importante para construir programas antes de compilarlos.

Por otro lado, también notamos que no todos los vínculos necesarios para realizar descargas con `wget` funcionaban. Este problema fue solucionado actualizando `wget`, con el comando `$ sudo apt-get install wget`.

Además, para asegurarnos de trabajar con un repositorio siempre actualizado, ingresamos `$ sudo apt-get update`. De esta forma, el repositorio del cual obteníamos los programas instalados ya se encontraba actualizado

También ingresamos `$ sudo apt-get upgrade`, para tener actualizado el software.

## 2. Instalación de las dependencias

Antes de proceder a la obtención del código fuente desde Internet, nos detuvimos para instalar librerías y programas necesarios para instalar Elinks y poder utilizar muchas funciones que si no no estarían disponibles. Este es el resumen de todas estas dependencias de Elinks, que también se encuentra disponible en <http://elinks.cz/documentation/installation.html>:

- Lua o Guile: permite la escritura de scripting plugins.
- zlib 1.2.0.2 o más: Para manejar gzip y descomprimir archivos comprimidos, tanto locales como enviados desde el servidor
- bzip2: Ídem, para documentos comprimidos con bzip2.
- Utilidades LZMA: Ídem, para archivos comprimidos con LZMA.
- OpenSSL o GNU TLS: Para llevar a cabo navegación HTTP segura.
- pkg-config: Necesario para localizar ciertas librerías (al menos GNU TLS y TRE).
- GPM: Mouse de propósito general (General Purpose Mouse, GPM) para soporte del mouse.
- expat: Paquete de herramientas del parseador XML necesarios para soporte de XBEL.
- TRE: Para realizar búsquedas regexp.
- libsmclient: Librería requerida para soporte de smb://protocol.
- rxvt-unicode: Para emulador terminal que soporta 88 colores.
- xterm con 256 colores: necesario para utilizar el modo 256 colores. Las versiones más actualizadas de PuTTY también cuentan con esta funcionalidad.
- libidn: Para nombres de dominio internacionalizados.
- SpiderMonkey: Motor JavaScript de Mozilla para tener soporte de JavaScript/ECMAScript en ELinks.

Debe recordarse que estamos trabajando en una computadora sin interfaz gráfica. De este modo, GPM, rxvt-unicode y xterm se vuelven inútiles. En efecto, al no poder utilizar el mouse, GPM no tiene utilidad. Por otro lado, a pesar de que el framebuffer de Linux pueda soportar 256 (o más) colores, el driver de la consola no. De hecho, las aplicaciones de consola siempre estarán limitadas a 16 colores. Esto hace inútil instalar rxvt-unicode para obtener 88 colores, o xterm para obtener 256.

A continuación, explicaremos los pasos seguidos para la instalación de cada dependencia:

## 2.1. Instalación de Lua

Lua se encuentra disponible en Internet mediante el enlace <https://www.lua.org/ftp/lua-5.0.3.tar.gz>, que automáticamente inicia su descarga en un archivo comprimido. Para comenzar la instalación, creamos un directorio específico ingresando `$ mkdir install_lua` e ingresamos a él con `$ cd install_lua`.

Inicialmente, para iniciar la descarga ingresamos:

```
$ wget https://www.lua.org/ftp/lua-5.0.3.tar.gz
```

Ingresando el comando `ls`, vimos que ahora en este directorio se encontraba un archivo llamado `lua-5.0.3.tar.gz`.

Para descomprimir el archivo, utilizamos `tar` de la siguiente manera:

```
$ tar -zxf lua-5.0.3.tar.gz
```

Esto creó el archivo `lua-5.0.3` (lo cual verificamos con `ls`); el archivo descomprimido.

Este archivo no era un archivo común; en realidad era un directorio, al cual ingresamos con el comando `$ cd lua-5.0.3`.

De la lectura detallada del manual de instalación, accesible mediante el enlace <http://elinks.or.cz/documentation/manual.html>, que da abarca a grandes rasgos del proceso de instalación, puede inferirse que es recomendable descomentar la línea de `POpen` en `./configure` (quitarle el `#`). De este modo, ingresamos `nano config` para poder ajustar este detalle. Oprimiendo `Ctrl+w`, ingresamos `popen`, y buscamos la palabra. La primera aparición no era la buscada, pues se encontraba en una línea no comentada (que no iniciaba con `#`). Luego de repetir la búsqueda y leer un poco, notamos que en realidad había tres líneas juntas comentadas que contenían el nombre de la variable `POpen`...

Resolvimos la duda respecto a cuál descomentar de la siguiente manera:

- La última no era, puesto que arriba decía que probablemente funcionara para algunos sistemas de Windows (en cualquier caso, no corresponde a Linux)
- Para definir entre la primera y la segunda, notamos que tomando la segunda, no teníamos la posibilidad de soportar pipes, mientras que en la primera esto no sucedía
- La primera opción era la que debía elegirse en sistemas no POSIX (como Linux) que tuvieran `popen` y `pclose` (verificamos la existencia de los mismos en Linux ingresando `man popen`, que explicaba las funciones `popen` y `pclose`, lo cual indicaba que existían)

Entonces, elegimos la primera opción y descomentamos la línea:

```
#POpen= -DUSE_POpen=1
```

Que ahora quedó así:

```
POpen= -DUSE_POpen=1
```

Luego, cerramos `nano` y comenzamos con el procedimiento clásico para instalación de software:

- Primero, ingresamos el comando `$ ./configure` para construir el programa
- Luego, lo compilamos con `$ make`

- Después, chequeamos con el comando `$ make check`
- Finalmente, utilizamos `$ sudo make install`

## 2.2. Instalación de zlib

Navegando en Internet, encontramos que la librería correspondiente es `zlib1g`. Entonces, ingresamos:

```
$ sudo apt-get install zlib1g
```

## 2.3. Instalación de bzip2

En este caso, la librería es `bzip2`:

```
$ sudo apt-get install bzip2
```

## 2.4. Instalación de LZMA Utils

Nuevamente, al igual que con Lua, descargamos desde Internet el archivo comprimido y luego lo descomprimos. Regresamos al directorio del usuario con `$ cd`, creamos un directorio específico con `$ mkdir install_lzma`, y nos movimos hacia él con `$ cd install_lzma`. Tipeamos:

```
$ wget https://tukaani.org/lzma/lzma-4.32.7.tar.gz
```

Este comando en realidad había fallado, pero fue allí cuando descubrimos que era necesario actualizar `wget` como se había detallado más arriba. Una vez hecho esto, reingresamos el comando y funcionó:

```
$ wget https://tukaani.org/lzma/lzma-4.32.7.tar.gz
```

Luego, con el comando `ls`, notamos que el archivo comprimido se llamaba `lzma-4.32.7.tar.gz`. Lo descomprimos ingresando:

```
$ tar -xzf lzma-4.32.7.tar.gz
```

Entonces, cambiamos al directorio, ahora descomprimido, mediante el comando `$ cd lzma-4.32.7`. Repetimos el mismo procedimiento que con Lua:

- ingresamos el comando `$ ./configure`, para construir;
- ingresamos `$ make`, para compilar;
- tipeamos `$ make check`, para verificar;
- finalizamos con `$ sudo make install`, para instalar.

## 2.5. Instalación de Open SSL

Luego de realizar múltiples búsquedas en Internet, descubrimos que la librería `libssl-dev` es la que corresponde a OpenSSL. Para instalarla, ingresamos el comando:

```
$ sudo apt-get install libssl-dev
```

## 2.6. Instalación de pkg-config

A continuación, para la instalación de `pkg-config`, también procedemos con una mecánica similar

```
$ sudo apt-get install pkg-config
```

## 2.7. Instalación de expat

Hacemos lo mismo, instalando expat con este comando:

```
$ sudo apt-get install expat
```

## 2.8. Instalación de TRE

La instalación de TRE es un poco más compleja. De hecho, el procedimiento es similar a la instalación de Elinks en sí, solo que no hay que instalar tantas dependencias complejas.

Para empezar, lo primero que hicimos fue clonar el repositorio con GIT, que habíamos instalado previamente. Una vez que regresamos al directorio del usuario con `cd` y creamos y nos fuimos al directorio `install_TRE` con `mkdir TRE` y `cd TRE`, usamos el comando:

```
$ git https://github.com/laurikari/tre/
```

Ingresando al enlace provisto mediante Internet, no solo está disponible el repositorio, sino también una explicación relacionada a la compilación. Básicamente, se nos informó de una lista de herramientas necesarias para la compilación de TRE (de hecho muchas son de gran utilidad para compilar software complejo en general). Estas herramientas eran `autoconf`, `automake`, `gettext`, `libtool` y `zip` (opcional). Para instalarlas, ingresamos los comandos:

- `$ sudo apt-get install autoconf`
- `$ sudo apt-get install automake`
- `$ sudo apt-get install gettext`
- `$ sudo apt-get install libtool`
- `$ sudo apt-get install zip`

Después, siguiendo la guía de la página de Internet, tipeamos:

```
$ ./utils/autogen.sh
```

Esto generó el archivo `./configure`. Ahora ya podíamos repetir el mismo procedimiento que en las dos oportunidades anteriores en que compilamos un programa, ingresando la sucesión de comandos:

- `$ ./configure`
- `$ make`
- `$ make check`
- `$ sudo make install`

## 2.9. Instalación de libsmbclient

Esta librería puede instalarse sencillamente con el comando:

```
$ sudo apt-get install libsmbclient
```

## 2.10. Instalación de libidn

En este caso, la librería buscada es `libidn11-dev`. Entonces, basta con tipear:

```
$ libidn11-dev
```

## 2.11. Instalación de SpiderMonkey

Tras investigar en Internet, descubrimos que la librería que contiene SpiderMonkey es libmozjs-dev, con lo cual era suficiente tipear:

```
$ sudo apt-get install libmozjs-dev
```

Recordando que no se instalarán GPM, rxvt-unicode o xterm para 256 colores, esto finaliza la instalación de las dependencias.

## 3. Instalación de Elinks

Llegamos a la parte final del proceso de instalación:

- Primero, regresamos al directorio del usuario con `cd`, ingresamos `$ mkdir install_Elinks`, y nos vamos a ese directorio con el comando `$ cd install_Elinks`;
- luego, ingresamos `$ git clone http://elinks.cz/elinks.git`;
- esto crea el directorio `elinks` que contiene los archivos con el código fuente, al cual ingresamos mediante el comando `$ cd elinks`;
- alistamos la rama principal, con el comando `$ git branch elinks-0.11 origin/elinks-0.11`;
- tipeamos `$ git branch -v`;
- una vez hecho esto, ya estamos en condiciones de ingresar `$ auto-gen.sh` para generar el archivo `configure`;
- ingresamos el comando `$ ./configure` para construir, y esperamos;
- tipeamos `$ make` para compilar, y esperamos;
- chequeamos con `$ make check`;
- e instalamos con `$ sudo make install`.

Luego de esto, `elinks` ya queda instalado, y se puede ejecutar ingresando el comando `elinks`.