

Instalación de herramientas para compilación de md2tex

Luis E. Fajardo

08 de diciembre del 2020

Objetivos

- ✓ Instalar CMake.
- ✓ Instalar Flex.
- ✓ Instalar GNU Bison.
- ✓ Instalar TREECC.
- ✓ Preparar el entorno de desarrollo de md2tex.

Requisitos

- ✓ Contar con una distribución de Linux basada en Debian.
- ✓ Contar con una versión de **GNU GCC** (GNU Compiler Collection) versión 8.3 o superior.
- ✓ Tener instalada la herramienta de control de versiones **GIT**.
- ✓ Tener instalado **pdflatex** (sirve para compilar el código generado por md2tex).

Índice

1. Introducción	4
2. Acerca de md2tex	4
3. CMake	5
3.1. Instalación	5
4. Flex	6
4.1. Instalación	6
5. Bison	7
5.1. Instalación	7
6. TREECC	8
6.1. Instalación	8
7. Ambiente de desarrollo md2tex	9
Referencias	10

1. Introducción

En este documento se le brindará una serie de pasos para la instalación de las diferentes herramientas necesarias para la compilación de **md2tex** también se le brindará una breve explicación acerca de cada herramienta a instalar.

Al final de esta guía usted será capaz de compilar el proyecto **md2tex** en su ordenador; usted podrá visualizar, analizar y comprender cada etapa de ejecución para este proyecto, también podrá modificar el código escrito y de esta manera adaptarlo a sus necesidades.

2. Acerca de md2tex

Esta herramienta se elaboró pensando en las necesidades de usuarios que quieren aprender Latex... haciendo y aprendiendo. Los documentos son escritos en su mayoría con el lenguaje de marcado Markdown aunque tiene algunas variaciones para cumplir con algunos parámetros que Latex requiere. **md2tex** generará código Latex el cual puede ser visto y comprendido por el usuario.

3. CMake

Es una herramienta para definir y administrar compilaciones de código, principalmente para C++. Es una herramienta multiplataforma; la idea es tener una definición única de cómo se construye un proyecto, lo que se traduce en definiciones de compilación específicas para cualquier plataforma compatible.

CMake es un paso intermedio, que genera entradas de compilación para diferentes plataformas. En Linux, CMake genera Makefiles mientras que en Windows, puede generar proyectos de Visual Studio por ejemplo. El comportamiento de la compilación se define en los archivos **CMakeLists.txt** el cual define lo que debe hacer el sistema de construcción en un directorio (Contributors, s.f.).

En nuestro caso CMake será utilizado para compilar el proyecto **md2tex** y generar un archivo ejecutable. El código a compilar en su mayoría está escrito en C++.

3.1. Instalación

Como primer paso; ir a la página de descargas de CMake: <https://cmake.org/download/> luego, escoger una versión, descargar, descomprimir y desde la terminal de su sistema ubicarse en la carpeta descomprimida.

Como segundo paso; ejecute los siguientes comandos:

Instalación de CMake

```
$ ./bootstrap
$ make
$ make install
```

Recuerde que para estas acciones deberá tener acceso a los privilegios de superusuario en su sistema (ejecutar los comandos con **sudo** cuando sea requerido).

Para verificar que CMake se ha instalado correctamente, desde la terminal ejecute el comando: **cmake -version**. Deberá aparecerle una descripción similar a la siguiente:

Output

```
cmake version 3.18.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

En caso de no tener **make** instalado, puede instalarlo ejecutando **apt install make**.

4. Flex

FLEX (Fast LEXical analyzer generator) es una herramienta para generar escaneres. En lugar de escribir un escaner desde cero, solo se necesita identificar la sintaxis de un lenguaje determinado, escribir una serie de reglas utilizando expresiones regulares, por ejemplo: `DIGIT[0-9]`; y Flex construirá un escaner para esa expresión (Gao, s.f.).

4.1. Instalación

Para instalar Flex lo haremos desde sourceforge.net o bien puede hacerlo desde ??, descargue y descomprima el archivo (descargue el archivo de extensión .gz).

Ejecute los siguientes comandos:

Instalación de Flex

```
$ ./configure
$ make
$ make install
```

Asegurese de tener privilegios de superusuario para ejecutar el último comando. Para verificar que Flex ha sido instalado en su sistema, escriba el siguiente comando: **flex -V**

Aparecerá una información similar a la siguiente:

Output

```
flex 2.6.4
```

5. Bison

Es un generador de analizadores sintácticos de propósito general que convierte una gramática de libre contexto (*context-free*) en un analizador LR determinista o LR generalizado (GLR) que emplea tablas de análisis LALR(1). Como funcionalidad experimental Bison también puede generar tablas de análisis **IELR(1)** o canónicas **LR(1)**. (Foundation, 2014)

Bison nos servirá para construir las diferentes estructuras del lenguaje de marcado Markdown y las estructuras que servirán para el envío de parámetros requeridos por Latex. Aquí se manejan algunos errores de sintaxis aunque se buscará reducirlos al mínimo.

5.1. Instalación

Para instalar Bison, ir a la página de descarga <http://ftp.gnu.org/gnu/bison/> y buscar **bison-3.7.tar.xz**; luego descargar, descomprimir, ubicarse en la carpeta descomprimida y abrir la terminal.

En la terminal, ejecute los siguientes comandos:

Instalación de bison

```
$ ./configure
$ make
$ make install
```

Finalmente para que Bison sea visible por todo el sistema ejecute el siguiente comando:

```
$ cp /usr/local/bin/bison /usr/bin/
```

Asegurese de tener privilegios de superusuario para ejecutar los comandos anteriores. Verifique si bison esta instalado en su sistema ejecutando el comando: **bison -V**

Aparecerá una salida similar a la siguiente:

Output

```
bison (GNU bison) 3.0.4
Escrito por Robert Corbett y Richard Stallman.

Copyright (C) 2015 Free Software Foundation, Inc.
Esto es software libre; mire el fuente para las condiciones de copia.
No hay NINGUNA garantía; ni siquiera para COMERCIALIZACIÓN o ADAPTACIÓN
PARA UN USO PARTICULAR
```

6. TREECC

El programa TREECC está diseñado para ayudar en el desarrollo de compiladores y otras herramientas “*language-based*”. Gestiona la generación de código para manejar árboles de sintaxis abstracta y operaciones sobre árboles. (Weatherley, 2002)

Para este proyecto TREECC servirá para la generación de código Latex. TREECC nos ayuda creando una representación en memoria del AST (Abstract Syntax Tree) y también se encarga de visitar cada nodo del árbol y ejecutar las operaciones que contiene cada nodo.

6.1. Instalación

Para esta instalación utilizaremos el programa GIT, aunque también puede descargar el código en el siguiente enlace: <https://github.com/ideras/treecc>.

Desde la terminal ubíquese en la carpeta descargada y ejecute lo siguiente:

Instalación de treecc

```
$ ./auto_gen.sh
$ ./configure
$ make all
$ make check
$ make install
```

Es probable que necesite privilegios de superusuario para ejecutar los comandos anteriores. En este punto TREECC deberá estar instalado en su sistema. Verifique la instalación ejecutando el comando **treecc -v**, el cual genera una salida similar a la siguiente:

Output

```
TREECC 0.3.11 - Tree Compiler-Compiler
Copyright (c) 2001, 2002 Southern Storm Software, Pty Ltd.

TREECC comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under the terms of the
GNU General Public License. See the file COPYING for further details.

Use the '--help' option to get help on the command-line options.
```


7. Ambiente de desarrollo md2tex

En este punto ya deberá tener instaladas las herramientas necesarias para hacer modificaciones o desarrollar nuevas funcionalidades para md2tex. El objetivo en este apartado es generar un archivo ejecutable utilizando un **CMakeLists.txt**, en este archivo están escritas todas las definiciones necesarias para construir el proyecto.

Dentro de la carpeta de md2tex abra la terminal y escriba lo siguiente:

Compilación de md2tex

```
$ mkdir build && cd build
$ cmake ../
$ make
```

Si no hubieron errores durante la compilación, el ejecutable de **md2tex** está en la carpeta **build**. Para generar código Latex vaya a la carpeta tests y elija un archivo.

Escriba lo siguiente en la terminal:

```
$ ./md2tex ../tests/yourfile.md
```

Dentro de la carpeta md2tex podrá ver que se generó una carpeta llamada **latex/tex**, revise y compile el código generado. Para facilitar el proceso de compilación también puede hacer uso de CMake en este punto. Existe una definición de reglas de CMake para **pdflatex**, puede descargar el archivo: [UseLATEX.cmake](#) construya un **CMakeLists.txt** y ejecute los siguientes comandos:

Compilación con pdflatex

```
$ mkdir build && cd build
$ cmake ../
$ make
```

El archivo PDF deberá encontrarse dentro de la carpeta **build** el cual puede abrir desde cualquier visor PDF.

Referencias

Contributors, S. O. D. (s.f.). *Empezando con CMake*. Recurso online: <https://riptutorial.com/es/cmake>. (Contenido consultado: Observaciones)

Foundation, F. S. (2014, 8). *GNU Bison*. Recurso online: <https://www.gnu.org/software/bison/#tabAboutGNU>. (Contenido consultado: Introduction to Bison)

Gao, L. (s.f.). FLEX Tutorial. Recurso online: <http://alumni.cs.ucr.edu/~lgao/teaching/flex.html>. (Contenido consultado: How to use FLEX?)

Weatherley, R. (2002). *Tree Compiler-Compiler*. <https://github.com/rweather/treecc>. (Contenido consultado: Tree Compiler-Compiler)