

Compilador twsc

Manual de usuario

Fecha: 04/Abril/2019

- Monterrubio López Charlie Brian
- Valdez Mondragón Erik
- Ortiz Martínez Brenda
- Barrón Pérez Marian Andrea


```
int main() {  
    return 2;  
}
```



111011011011000101000000011100001100011
100001101100011111011011011000101000000
10110001111011011011000101000000011100
01100011110110110110001010000000111000
00011110110110110001010000000111000011
11011000111101101101100010100000001110
00011011000111101101101100010100000001
01100011110110110110001010000000111000
11101101101100010100000001110001100011
00011110110110110001010000000111000011

Índice

1.	Objeto del documento	3
2.	Participantes	3
3.	Objetivos.....	3
4.	Manual de usuario	3
4.1.	Prerrequisitos de instalación	4
4.2.	Instalación	4
4.3.	Configuración	5
4.4.	Ejecución	5

	<div> <div> Compilador - twscc Manual de usuario </div> <div> Objeto del documento </div> </div>
---	--

1. Objeto del documento

El presente documento pretende mostrar al usuario el funcionamiento del compilador-twscc

2. Participantes

Nombre	Rol
Monterrubio López Charlie Brian	Integrador del sistema
Valdez Mondragón Erik	Pruebas
Ortiz Martínez Brenda	Arquitecto del sistema
Barrón Pérez Marian Andrea	Administrador del proyecto

3. Objetivos

Mostrar de manera clara y concisa el funcionamiento del compilador twscc, la instalación, configuración y ejecución del programa desde la línea de comandos.

4. Manual de usuario

• Prerrequisitos de instalación

1. Contar con algún Sistema Operativo UNIX
2. Tener instalado git
3. Tener una cuenta github
4. Tener instalado Elixir

• Instalación

El código del compilador twscc se encuentra almacenado en el repositorio remoto GitHub por lo que para poder acceder a el desde la máquina necesitamos utilizar la herramienta git. Para esto desde la terminal nos situamos en la ruta donde queremos almacenar los archivos, llamamos a git y le decimos que se traiga el contenido del repositorio, tecleamos lo siguiente:

`git clone https://github.com/hiphoox/compilers2019_2.git`

```
charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador
$ git clone https://github.com/hiphoox/compilers2019_2.git
Cloning into 'compilers2019_2'...
remote: Enumerating objects: 470, done.
remote: Counting objects: 100% (470/470), done.
remote: Compressing objects: 100% (254/254), done.
remote: Total 985 (delta 230), reused 427 (delta 208), pack-reused 515
Receiving objects: 100% (985/985), 66.68 MiB | 1.67 MiB/s, done.
Resolving deltas: 100% (435/435), done.
```

Al teclear ls podemos ver que creo un directorio llamado compiler2019_2

```
charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador
$ ls
compilers2019_2/
```

Ahora con cd podemos entrar a la carpeta para revisar lo que contiene.

Ahora para acceder al contenido del compilador twscc necesitamos cambiar de rama, ingresamos el siguiente comando:

git checkout twscc

```
charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador/compilers2019_2 (master)
$ git checkout twscc
Checking out files: 100% (40/40), done.
Switched to a new branch 'twscc'
Branch 'twscc' set up to track remote branch 'twscc' from 'origin'.

charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador/compilers2019_2 (twscc)
```

Des esta manera ya nos encontraremos en la rama twscc donde se encuentra todo el contenido respecto al proyecto. Encontraremos las siguientes carpetas.

- Members: Dentro de esta carpeta se encontrará el código desarrollado por los miembros del proyecto.
- ProjectCompiler-twscc: Dentro de esta carpeta se encontrará el código de la primera versión del compilador twscc, es aquí donde se generará el ejecutable del programa
- ProjectDocumentation: En esta carpeta está toda la documentación que es necesaria para describir el compilador.

Una vez obtengamos estas carpetas podemos movernos a la carpeta ProjectCompiler-twscc

```
charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador/compilers2019_2 (twscc)
$ cd ProjectCompiler-twscc

charl@CHARLIE-BML-PC MINGW64 ~/Desktop/Compilador/compilers2019_2/ProjectCompiler-twscc (twscc)
$ |
```

• Configuración

Una vez adentro de la carpeta con los archivos vamos a generar el ejecutable con la ayuda de la herramienta mix, para esto tecleamos el siguiente comando:

mix escript.build

```
brenda@brenda-HP-Pavilion-14-Notebook-PC:~/compilador_proyecto_deno1/compilers2019_2
Generated escript twscc_compilador_c with MIX_ENV=dev
```

• Ejecución

Se generará un archivo .out el cual mandaremos a ejecutar de la siguiente manera:

./twscc_compilador_c [parámetros de entrada]

```
brenda@brenda-HP-Pavilion-14-Notebook-PC:~/compilador_proyecto_deno1/compilers2019_2
./codigo_prueba_return.s: Mensajes del ensamblador:
./codigo_prueba_return.s:1: Error: el carácter a continuación del nombre no es '#'
"codigo_prueba_return.s"
```

- **Pruebas**

Para poder correr las pruebas automatizadas del compilador solo basta con poner desde la consola lo siguiente:

mix test

```
mix test test/twscc_compilador_c|
bash-3.2$ mix test test/twscc_compilador_c.exs|
Compiling 1 file (.ex)
.....

Finished in 0.06 seconds
8 tests, 0 failures

Randomized with seed 782447
bash-3.2$
```

Comenzarán a correr y ejecutarse todas las pruebas previamente programadas. Al final nos mandará un mensaje de 0 fallos si es que corrió adecuadamente todas las pruebas.