

Documentación del  
sistema de computación Set  
Set Computing System  
ScS

*David Díaz García*<sup>1</sup>

7 de agosto de 2014

<sup>1</sup>✉ [david.diaz.uni@gmail.com](mailto:david.diaz.uni@gmail.com)



## **Resumen**

Vacío



# Índice general

<b>Lista de figuras</b>	<b>V</b>
<b>Lista de cuadros</b>	<b>VII</b>
<b>Lista de códigos</b>	<b>IX</b>
<b>1. El lenguaje Set</b>	<b>1</b>
1.1. Introducción al lenguaje Set . . . . .	1
1.2. Un programa en Set . . . . .	1
1.3. Elementos léxicos . . . . .	2
1.3.1. Comentarios . . . . .	3
1.4. Elementos sintácticos . . . . .	3
1.4.1. Declaración de funciones . . . . .	3
<b>2. El compilador Set</b>	<b>5</b>
2.1. El makefile . . . . .	5
<b>Apéndice</b>	<b>7</b>
<b>A. Instalación de herramientas</b>	<b>9</b>
A.1. Linux . . . . .	9
A.2. OS X . . . . .	11
A.3. Windows . . . . .	11
<b>B. Instalación del compilador Set</b>	<b>13</b>
B.1. Linux . . . . .	13
B.2. OS X . . . . .	13
B.3. Windows . . . . .	13
<b>Bibliografía</b>	<b>15</b>



# Lista de figuras

A.1. Apertura de un terminal en Ubuntu . . . . .	10
A.2. Instalación de Flex en Ubuntu mediante el shell . . . . .	10





# Lista de cuadros



# Lista de códigos

1.1. Mensaje «Hola mundo» (.mset) . . . . .	2
1.2. Comentario monolínea (.mset) . . . . .	3
1.3. Comentario multilínea (.mset) . . . . .	3



# Capítulo 1

## El lenguaje Set

### 1.1. Introducción al lenguaje Set

Cuando estudiamos un nuevo lenguaje natural como el inglés, el alemán o el chino, empezamos aprendiendo el alfabeto de símbolos usados en el lenguaje, a continuación formamos palabras con esos símbolos. Lo siguiente es aprender la manera correcta en la cual se deben colocar las palabras para formar frases y el significado de dichas combinaciones de palabras. La misma idea se aplica al lenguaje de programación de alto nivel Set<sup>1</sup>. La característica principal de este lenguaje es que permite representar los algoritmos de forma muy parecida a como se hace en los lenguajes de especificación de algoritmos (LEA); es decir, mediante uso de operandos, términos y expresiones que son propias del lenguaje matemático. Para ello se permiten algunos símbolos del juego de *caracteres unicode* con los que representar de forma explícita

- tipos de datos como conjuntos numéricos matemáticos ( $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$ ),
- declaraciones de variables con el operador de pertenencia de conjuntos  $\in$ ,
- asignaciones de valores a variables con el símbolo  $\leftarrow$  o
- declaraciones de funciones con los símbolos  $\times$  (para separar los parámetros) o  $\rightarrow$ , entre otros.

Este lenguaje carece de los conceptos de referencia, puntero, reserva y liberación de memoria, y dado se permite el uso de notación matemática, es el lenguaje de alto nivel más puro que existe.

### 1.2. Un programa en Set

Un programa fuente en lenguaje Set está compuesto por

---

<sup>1</sup>Un lenguaje de programación de «alto nivel» se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad

ejecutora de la máquina de las máquinas. Para ello se utilizan términos o palabras propias del lenguaje natural o del lenguaje Matemático.

- un fichero denominado *principal* que puede tener cualquier nombre con extensión *.mset*<sup>2</sup>, y por
- cero o varios *módulos* que pueden tener cualquier nombre y cuya extensión es
  - *.hset* para los módulos de cabecera<sup>3</sup> y
  - *.iset* para los módulos de implementación<sup>4</sup>.

Para compilar un programa fuente en lenguaje Set debe ejecutar en su terminal el comando *cset*, seguido de la ruta donde se encuentra el programa fuente principal (el que tiene la extensión *.mset*) seguido de su nombre, con o sin extensión. Para poder ejecutar el comando de compilación *cset* debe haber instalado previamente el compilador Set en su sistema (consulte el apéndice B).

Por ejemplo, abra su editor de texto plano preferido y escriba el siguiente programa fuente:

---

Código 1.1. Mensaje «Hola mundo» (*.mset*)

---

```
1 message("Hola mundo")
```

Guarde el programa fuente con el nombre que prefiera y añádale la extensión *.mset* justo a continuación del nombre. Suponiendo que se ha guardado en el escritorio como *hello\_world.set*, la compilación en diferentes sistemas se hace de la siguiente forma:

- 
- 

Una vez compilado obtendrá un programa ejecutable que puede ejecutar de la siguiente forma:

- En linux y en OS X:
- En Windows:

Como se puede observar, la ejecución muestra en la terminal de su sistema el mensaje «Hola mundo».

Las funciones solo se pueden declarar e implementar en los módulos, de hecho todo el código que aparezca en un módulo debe estar dentro de una función. Con esto se fuerza al programador, desde el primer momento,

### 1.3. Elementos léxicos

En las siguientes subsecciones se discuten los elementos léxicos del lenguaje Set: comentarios, identificadores, palabras reservadas, números, caracteres y cadenas de caracteres.

---

<sup>2</sup>La *m* que aparece en la extensión *.mset* es la primera letra de la palabra inglesa «main» que significa «principal».

<sup>3</sup>La *h* que aparece en la extensión *.hset* es la primera letra de la palabra inglesa «header»

que significa «cabecera».

<sup>4</sup>La *i* que aparece en la extensión *.iset* es la primera letra de la palabra inglesa «implementation» que significa «implementar».

### 1.3.1. Comentarios

Los comentarios son descripciones o aclaraciones del código en lenguaje natural que aparecen escritas junto a este y que ayudan a entender su estructura y su lógica (su significado). En el lenguaje Set se permiten los siguientes dos tipos de comentarios:

- Comentarios monolínea. Se añaden escribiendo un punto y coma (;) seguido por el texto del comentario como se muestra en el siguiente código:

---

#### Código 1.2. Comentario monolínea (.mset)

---

```
1 ; Un comentario explicativo de una sola línea.  
2 ; Otro comentario de una sola línea.
```

- Comentarios multilínea. Comienzan con los caracteres de apertura de comentario punto y coma guion (;-), sin ningún espacio entre ambos caracteres, y se extienden hasta los caracteres de cierre guion punto y coma (-;), de nuevo, sin ningún espacio entre ambos. Los caracteres de apertura y cierre pueden estar en diferentes líneas o sobre la misma línea como se muestra en el siguiente código:

---

#### Código 1.3. Comentario multilínea (.mset)

---

```
1 ;- Este comentario se extiende por más de  
2 de un línea, es decir, entre cada línea de  
3 de un comentario multilínea puede haber más  
4  
5 de un salto de...  
6  
7  
8 ...línea. -;
```

## 1.4. Elementos sintácticos

### 1.4.1. Declaración de funciones

Como ya se ha dicho en secciones anteriores





## Capítulo 2

# El compilador Set

### 2.1. El makefile

Para la recompilación y generación del archivo ejecutable Set  
sss



# Apéndice



## Apéndice A

# Instalación de herramientas

Este apéndice está dedicado al proceso de instalación de las herramientas que permiten la generación del compilador Set en las plataformas Ubuntu Linux, OS X y Windows. Las herramientas necesarias son cuatro:

- el generador de analizadores léxicos, *Flex*;
- el generador de analizadores sintácticos, *Bison*;
- la herramienta para el control de la recompilación, *Make* y
- el compilador del lenguaje C++, *GNU C++* (g++)<sup>1</sup>.

### A.1. Linux

En los sistemas operativos basados en Linux la instalación se suele realizar desde el shell (terminal)<sup>2</sup> de la siguiente manera:

1. Asegúrese de que está conectado a Internet.
2. Abra una ventana de terminal pulsando la combinación de teclas Ctrl+Alt+T o pase al modo consola virtual pulsando la combinación Ctrl+Alt+F1<sup>3</sup>. Otra forma de abrir un terminal en Ubuntu es haciendo clic con el botón izquierdo del ratón sobre el botón Inicio (figura A.1a) y donde aparece Buscar (figura A.1b) escriba terminal (figura A.1c), por último haga clic con el botón izquierdo sobre el icono del terminal.
3. Para instalar Flex escriba el siguiente comando en la terminal y pulse la tecla enter:

```
sudo apt-get install flex
```

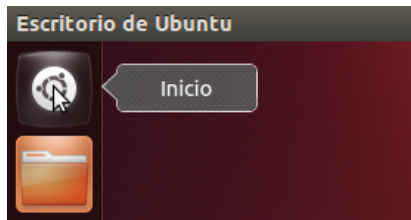
Siempre que, durante la instalación de una herramienta, se le pregunte ¿Desea continuar [S/n]? escriba una S mayúscula y pulse enter (figura A.2).

<sup>1</sup>g++ es el alias tradicional de «GNU C++» un conjunto gratuito de compiladores de C++. Forma parte del GCC, «GNU Compiler Collection» (del inglés, colección de compiladores GNU).

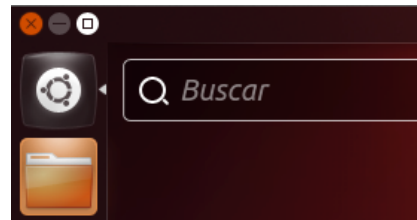
<sup>2</sup>Al shell también se le conoce como intérprete de órdenes o línea de comandos.

<sup>3</sup>En el modo de consola virtual el terminal

se abre a pantalla completa, es decir, desaparece la GUI (interfaz gráfica de usuario) y es necesario escribir el nombre de usuario y la contraseña antes de insertar los comandos para la instalación de las herramientas.



(a) Botón de inicio de Ubuntu



(b) Buscador de Ubuntu



(c) Icono del terminal de Ubuntu

Figura A.1. Apertura de un terminal en Ubuntu

```

uservmu@ubuntu: ~
uservmu@ubuntu:~$ sudo apt-get install flex
[sudo] password for uservmu:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libfl-dev m4
Paquetes sugeridos:
  bison build-essential
Se instalarán los siguientes paquetes NUEVOS:
  flex libfl-dev m4
0 actualizados, 3 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 431 kB de archivos.
Se utilizarán 964 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://es.archive.ubuntu.com/ubuntu/ precise/main m4 i386 1.4.16-2ubuntu1
[195 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu/ precise/main libfl-dev i386 2.5.35-10
ubuntu3 [19,2 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu/ precise/main flex i386 2.5.35-10bunt
u3 [216 kB]
Descargados 431 kB en 2seg. (208 kB/s)
Seleccionando el paquete m4 previamente no seleccionado.
(Leyendo la base de datos ... 211940 ficheros o directorios instalados actualmen
te.)
Desempaquetando m4 (de .../m4_1.4.16-2ubuntu1_i386.deb) ...
Seleccionando el paquete libfl-dev previamente no seleccionado.
Desempaquetando libfl-dev (de .../libfl-dev_2.5.35-10ubuntu3_i386.deb) ...
Seleccionando el paquete flex previamente no seleccionado.
Desempaquetando flex (de .../flex_2.5.35-10ubuntu3_i386.deb) ...
Procesando disparadores para install-info ...
Procesando disparadores para man-db ...
Configurando m4 (1.4.16-2ubuntu1) ...
Configurando libfl-dev (2.5.35-10ubuntu3) ...
Configurando flex (2.5.35-10ubuntu3) ...
uservmu@ubuntu:~$

```

Figura A.2. Instalación de Flex en Ubuntu mediante el shell

4. Para instalar Bison escriba el siguiente comando y pulse enter:

```
sudo apt-get install bison
```

5. Para instalar g++ escriba el siguiente comando y pulse enter:

```
sudo apt-get install g++
```

6. La herramienta Make viene instalada por defecto en la mayoría de los sistemas basados en Linux. Si por casualidad su sistema no le proveyera dicha herramienta, puede utilizar el siguiente comando para instalarla:

```
sudo apt-get install make
```

Otro modo de instalación es usando el centro de software de ubuntu.

## A.2. OS X

En los sistemas OS X puede obtener todas las herramientas necesarias instalando el IDE (entorno de desarrollo integrado) Xcode

El IDE Xcode ocupa un espacio considerable (más de 5 GiB), puede obtener una instalación más liviana de las herramientas necesarias utilizando MacPorts <http://www.macports.org/ports.php?by=name&substr=gcc>

## A.3. Windows





## **Apéndice B**

# **Instalación del compilador Set**

**B.1. Linux**

**B.2. OS X**

**B.3. Windows**



# Bibliografía

- [1] Ravi Sethi Alfred V. Aho Monica S. Lam y Jeffrey D. Ullman. *Compiladores. Principios, técnicas y herramientas*. En: 2.<sup>a</sup> ed. Pearson (Addison Wesley), 2008.
- [2] Charles Donnelly y Richard Stallma. *Bison. The Yacc-compatible Parser Generator*. 2013.
- [3] Tony Mason John R. Levine y Doug Brown. *Lex & Yacc*. En: 2.<sup>a</sup> ed. O'Reilly, 1992.
- [4] John R. Levine. *flex & bison. Unix Text Processing Tools*. En: O'Reilly, 2009.
- [5] Kenneth C. Loudon. *Construcción de compiladores. Principios y práctica*. En: Thomson, 2004.
- [6] Sergio Gálvez Rojas y Miguel Ángel Mora Mata. *Java a tope. Traductores y compiladores con Lex/Yacc, JFlex/Cup y JavaCC*. En: 2.<sup>a</sup> ed. Universidad de Málaga, 2005.
- [7] Will Estes Vern Paxson y John Millaway. *Lexical Analysis with Flex*. 2012.