

LENGUAJES DE PROGRAMACIÓN ESOTÉRICOS

TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

DESCRIPCIÓN BREVE

Un trabajo de investigación sobre el funcionamiento de los lenguajes de programación esotéricos, e implementación práctica del lenguaje COW.

Diego Lopez Reduello
2ºC Ingeniería del Software

ÍNDICE

Introducción.....	2
Definición.....	2
Ejemplos destacables.....	2
INTERCAL	2
Piet.....	4
Lenguaje COW: Lógica y funcionamiento	5
Lenguaje COW: Instalación y preparación.....	6
Implementación de programas en COW	7
Hello world.cow	7
Fibonacci.cow	9
99.cow	11
Miprograma.cow	12
Repositorio GitHub del proyecto	19
Repositorio original de COW.....	19
Bibliografía.....	19

Introducción

En este trabajo de investigación para la asignatura Teoría de Autómatas y Lenguajes Formales, explicaré qué es un lenguaje de programación esotérico, mostraré algunos ejemplos de lenguajes de programación esotéricos destacables tanto por su complejidad como por su originalidad y, por último, una implementación práctica de un lenguaje esotérico a mi elección. En este caso, he escogido el lenguaje esotérico COW.

Definición

Un lenguaje de programación esotérico es un lenguaje de programación minimalista que tiene como principal función demostrar que se pueden crear lenguajes pocos intuitivos, o incluso a veces absurdos, que son Turing-computables, es decir, lenguajes que pueden computar todas las funciones que podría computar una Máquina de Turing.

Su utilidad en casos prácticos o en proyectos de gran escala es bastante dudosa, debido a su complejidad de código y otras características poco comunes en lenguajes de programación. Son programas que suponen un reto para el que lo crea, y también un reto para los que intenten programar cualquier cosa con ellos.

Actualmente existe una comunidad en internet bastante activa que se dedica a crear todo tipo de lenguajes de programación esotéricos y escribir programas en ellos por diversión, además de que también debaten sus propiedades computacionales.

Ejemplos destacables

INTERCAL

INTERCAL es uno de los lenguajes de programación esotéricos más conocidos, extendidos y antiguos de la historia. Su diseño se basa en la siguiente mecánica: debemos pedir por favor la ejecución de ciertas sentencias. En vez de usar la instrucción mítica de los lenguajes de programación de bajo nivel *“GO TO”* para saltar a una instrucción concreta, usaremos una instrucción llamada *“COME FROM”* en la instrucción concreta a la que queremos llegar. También tiene otras peculiaridades, como evitar que se ejecuten ciertas sentencias con *“PLEASE ABSTAIN FROM”*,

entre otras peculiaridades. Este lenguaje de programación se creó con la intención de crear algo totalmente distinto. A continuación, tenemos un programa escrito en INTERCAL que lee dos enteros de 32 bits, tratándolos como enteros con signo en formato de complemento a dos, y escribiendo su valor absoluto.

```
DO (5) NEXT
(5) DO FORGET #1
PLEASE WRITE IN :1
DO .1 <- 'V":1~'#32768$#0"$#1'~#3
DO (1) NEXT
DO :1 <- "'V":1~'#65535$#0"$#65535'
~'#0$#65535"'$"'V":1~'#0$#65535"'
$#65535'~'#0$#65535"'
DO :2 <- #1
PLEASE DO (4) NEXT
(4) DO FORGET #1
DO .1 <- "'V":1~'#65535$#0"$":2~'#65535
$#0"'~'#0$#65535"'$"'V":1~'#0
$#65535"'$":2~'#65535$#0"'~'#0$#65535"'
DO (1) NEXT
DO :2 <- ":2~'#0$#65535"'
$"'":2~'#65535$#0"$#0'~'#32767$#1"'
DO (4) NEXT
(2) DO RESUME .1
(1) PLEASE DO (2) NEXT
PLEASE FORGET #1
DO READ OUT :1
PLEASE DO .1 <- 'V":1~:1'~#1"$#1'~#3
DO (3) NEXT
PLEASE DO (5) NEXT
(3) DO (2) NEXT
PLEASE GIVE UP
```

Piet

Piet fue el lenguaje de programación que personalmente hizo que me interesara por el tema de los lenguajes esotéricos. Se trata de un lenguaje de programación basado en stacks (una estructura de datos usada frecuentemente en las memorias de los lenguajes de programación esotéricos) cuyos programas parecen pinturas de arte abstracto. El programa que imprime en consola Hello World es el siguiente:



Piet usa 20 colores en sus programas, de los cuales 18 tienen dos propiedades: *hue* y *lightness* (se podría traducir al español como matiz y dureza). En la siguiente tabla podemos ver las diferentes variaciones de colores:

Light red (#FFC0C0)	Light yellow (#FFFFC0)	Light green (#C0FFC0)	Light cyan (#C0FFFF)	Light blue (#C0C0FF)	Light magenta (#FFC0FF)
Red (#FF0000)	Yellow (#FFFF00)	Green (#00FF00)	Cyan (#00FFFF)	Blue (#0000FF)	Magenta (#FF00FF)
Dark red (#C00000)	Dark yellow (#C0C000)	Dark green (#00C000)	Dark cyan (#00C0C0)	Dark blue (#0000C0)	Dark magenta (#C000C0)

Las instrucciones se escriben estableciendo la diferencia de propiedades entre el un color y otro, siguiendo la siguiente tabla de comandos:

Hue change	Light	ness	change
	No change	1 darker	2 darker
No change	N/A	Push	Pop
1 step	Add	Subtract	Multiply
2 steps	Divide	Modulo	Not
3 steps	Greater	Pointer	Switch
4 steps	Duplicate	Roll	Input num
5 steps	Input char	Output num	Output char

Por ejemplo, de *Light red* a *Yellow* hay un paso de *Hue change* y un paso de *lightness* (se cuenta en la tabla cuántos pasos de diferencia hay) y eso significaría que tenemos que ejecutar la instrucción *Substract*.

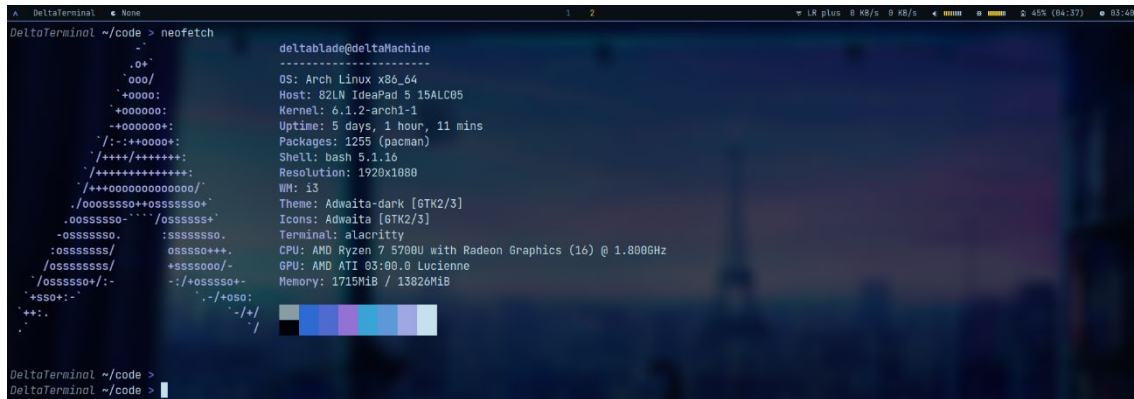
Lenguaje COW: Lógica y funcionamiento

COW fue creado a partir de Brainfuck, uno de los lenguajes esotéricos más conocidos. Funciona con un vector de memoria, un puntero que señala un bloque de ese vector, un flujo de entrada y un flujo de salida. Las posibles instrucciones son las siguientes:

ID	Instrucción	Descripción
0	moo	Esta instrucción está conectada con la instrucción MOO. Sirve para crear bucles. Cuando se encuentra durante una ejecución, busca en el programa (hacia atrás) un comando MOO y empieza a ejecutar las instrucciones desde ahí.
1	mOo	Mueve el puntero de memoria al bloque anterior.
2	moO	Mueve el puntero de memoria al bloque siguiente.
3	mOO	Lee el valor del bloque de memoria actual y ejecuta la instrucción asociada a ese valor. No funciona para el 3, ya que provocaría un bucle infinito, ni para números estrictamente mayores a 11.
4	Moo	Si el valor del bloque de memoria actual es 0, lee del flujo de entrada un carácter ASCII y almacena su valor en el bloque. En otro caso, imprime en la salida el carácter ASCII que corresponde al valor del bloque de memoria.
5	MOo	Decrementa en uno el valor del bloque de memoria actual.
6	MoO	Incrementa en uno el valor del bloque de memoria actual.
7	MOO	Si el valor del bloque de memoria actual es 0, salta al siguiente comando moo y sigue ejecutando desde la instrucción posterior a moo. En otro caso, continua con el siguiente comando.
8	OOO	Establece a 0 el valor del bloque de memoria actual.
9	MMM	Si no hay ningún valor almacenado en el registro, copia el valor del bloque de memoria actual. Si hay un valor almacenado, se pega el valor en el bloque de memoria actual y se resetea el registro.
10	OOM	Imprime en la salida el valor del bloque de memoria como un Integer.
11	oom	Lee de la entrada un número entero y lo escribe en el bloque de memoria actual.

Lenguaje COW: Instalación y preparación

Primero, inicié el sistema operativo Arch Linux en mi máquina local para trabajar de forma más cómoda con COW.



Después, en la carpeta correspondiente, cloné el repositorio original de cow con el comando “git clone <https://github.com/BigZaphod/COW.git>” y moví la carpeta source y algunos ejemplos de código a mi carpeta cowInvestigation. Creé el repositorio (al que se puede acceder con este enlace: <https://github.com/Deinigu/cowInvestigation>) y añadí el contenido de la carpeta.

En la carpeta source, hay dos archivos que nos interesan:

- `cow.k`: Es simplemente un archivo que contiene las traducciones de las instrucciones de COW a C++.
- `cow.cpp`: Es el programa escrito en C++ que, al compilarse, se puede usar para interpretar archivos escritos en el lenguaje COW.

Compilé cow.cpp con el comando “gcc cow.cpp -o compiladorcow” y moví el archivo de salida a la carpeta cow, donde están también los diferentes ejemplos.

A continuación, explicaré algunos ejemplos de programas escritos en cow y escribiré alguno propio.

Implementación de programas en COW

Helloworld.cow


Este programa, encontrado en la página de esolangs que refiere al lenguaje COW, imprime “Hello,world!” en pantalla, y está ligeramente tabulado por mí para entenderlo mejor. El código funciona de la siguiente manera:

```
; Incrementa hasta el número 72, que corresponde al carácter 'H' en la tabla
; ASCII
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO
; Imprime por pantalla el caracter
Moo
; Se repite el mismo proceso para todos los caracteres,
; incrementando, restando o inicializando a cero el bloque de memoria
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO
Moo
MoO MoO
MoO MoO MoO MoO MoO
Moo
Moo
MoO MoO MoO
Moo
OOO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
```


MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
 MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
 MoO MoO MoO MoO MoO MoO MoO MoO MoO
 Moo
 MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
 MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
 MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
 MOo MOo MOo MOo MOo
 Moo
 MOo MOo MOo MOo MOo MOo MOo MOo
 Moo
 MoO MoO MoO
 Moo
 MOo MOo MOo MOo MOo MOo
 Moo
 MOo MOo MOo MOo MOo MOo MOo MOo
 Moo
 OOO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
 MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
 MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
 Moo

Al ejecutarlo, imprime lo siguiente:

```

DeltaTerminal ~/code/cowInvestigation/cow > ./cow.out
Hello,  world!
  
```

Fibonacci.cow

Este programa viene en la carpeta de ejemplos del repositorio original. Imprime en pantalla en bucle los números de la sucesión de Fibonacci. El código (comentado con su modo de funcionar) es el siguiente:

```
; Inicialización de las variables
MoO
moO
MoO
mOo
; Loop
MOO
; Imprime el primer número
OOM
; Copia en el registro el primer número y lo lleva a un bloque
; al que llamaremos temp
MMM
moO
moO
MMM
mOo
mOo
; Pone el segundo número en la primera posición
moO
MMM
mOo
MMM
; Volvemos al bloque temp
moO
moO
; Usamos el bloque temp para sumarlo al primer número y lo guardamos
; en el segundo y volvemos a comenzar el bucle
MOO
MOo
mOo
MoO
moO
moo

mOo
mOo
moo
```

Al ejecutar el programa, imprime lo siguiente:

```
DeltaTerminal ~/code/cowInvestigation/cow > ./cow.out
```

```
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
28657
46368
75025
121393
196418
317811
514229
832040
1346269
2178309
3524578
5702887
9227465
14930352
24157817
39088169
63245986
102334155
```

165580141
267914296
433494437
701408733
1134903170
1836311903
-1323752223
^C

El número negativo no es problema del programa, sino del tipo de datos signed int, que es de 4 bytes y tiene un rango de valores desde -2,147,483,648 hasta 2,147,483,647. Como el siguiente número, que sería 2,971,215,073, no se encuentra en este rango, devuelve un número negativo.

99.cow

En el repositorio original también hay un programa adicional llamado 99.cow, que imprime la canción tradicional americana 99 Bottles of Beer:

```
DeltaTerminal ~/code/cowInvestigation/cow > ./cow.out
```

99

bottles of beer on the wall,

99

bottles of beer,

take one down, pass it around,

98

bottles of beer on the wall.

(Así consecutivamente...)

2

bottles of beer on the wall,

2

bottles of beer,

take one down, pass it around,

1

bottle of beer on the wall.

1

bottle of beer on the wall,

1

bottle of beer,

take it down, pass it around,

0

bottles of beer on the wall.

El programa es demasiado extenso como para adjuntarlo en el documento, pero puede consultarse tanto en mi repositorio como en la carpeta de ejemplos del repositorio original.

Miprograma.cow

Por último, me he animado a crear un programa sencillo. El código es el siguiente:

```
; Sumar hasta 80
```

```
OOO
```

```
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
```

```
; Imprimir carácter
```

```
Moo
```

```
; Sumar hasta 114
```

```
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO
```

```
; Imprimir carácter
```

```
Moo
```

```
; Restar hasta 111
```

```
MOo MOo MOo
```

```
; Imprimir carácter
```

```
Moo
```

```
; Restar hasta 102
```

```
MOo MOo MOo MOo MOo MOo MOo MOo MOo
```

```
; Imprimir carácter
```

```
Moo
```

; Restar hasta 101
MOo

; Imprimir carácter
Moo

; Sumar hasta 44
OOO

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO

; Imprimir carácter
Moo

; Restar hasta 32
MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo

; Imprimir carácter
Moo

; Sumar hasta 97

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 112

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 114
MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 117
MoO MoO MoO

; Imprimir carácter
Moo

; Restar hasta 101
MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo MOo MOo MOo MOo

; Imprimir carácter
Moo

; Restar hasta 98
MOo MOo MOo

; Imprimir carácter
Moo

; Restar hasta 97
MOo

; Imprimir carácter
Moo

; Sumar hasta 109
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO

; Imprimir carácter
Moo

; Restar hasta 101
MOo MOo MOo MOo MOo MOo MOo MOo

; Imprimir carácter
Moo

; Sumar hasta 32

OOO

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO

; Imprimir carácter

Moo

; Sumar hasta 84

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO

; Imprimir carácter

Moo

; Restar hasta 65

MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo MOo MOo MOo MOo MOo MOo MOo

; Imprimir carácter

Moo

; Sumar hasta 76

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO

; Imprimir carácter

Moo

; Restar hasta 70

MOo MOo MOo MOo MOo MOo

; Imprimir carácter

Moo

; Restar hasta 44

MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo MOo MOo MOo MOo

; Imprimir carácter
Moo

; Restar hasta 32
MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo

; Imprimir carácter
Moo

; Sumar hasta 112
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO

; Imprimir carácter
Moo

; Restar hasta 111
MOo

; Imprimir carácter
Moo

; Sumar hasta 114
MoO MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 32
OOO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 102
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO

; Imprimir carácter
Moo

; Restar hasta 97
MOo MOo MOo MOo MOo

; Imprimir carácter
Moo

; Sumar hasta 118
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO

; Imprimir carácter
Moo

; Restar hasta 111
MOo MOo MOo MOo MOo MOo MOo

; Imprimir carácter
Moo

; Sumar hasta 114
MoO MoO MoO

; Imprimir carácter
Moo

; Sumar hasta 32

OOO

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO

; Imprimir carácter

Moo

; Sumar hasta 58

MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO MoO MoO MoO MoO
MoO MoO MoO MoO MoO MoO

; Imprimir carácter

Moo

; Restar hasta 41

MOo MOo MOo MOo MOo MOo MOo MOo MOo MOo
MOo MOo MOo MOo MOo MOo MOo

; Imprimir carácter

Moo

Al ejecutarlo, el programa imprime lo siguiente en consola:

DeltaTerminal ~/code/cowInvestigation/cow > ./cow.out

Profe, apruebame TALF, por favor :)

Repositorio GitHub del proyecto

<https://github.com/Deinigu/cowInvestigation>

Repositorio original de COW

<https://github.com/BigZaphod/COW>

Bibliografía

- <https://esolangs.org/wiki/>
- <http://www.formauri.es/personal/pgimeno/compurec/LenguajesEsotericos.php>
- https://es.wikipedia.org/wiki/Lenguaje_de_programación_esotérico
- <https://bigzaphod.github.io/COW/>
- <https://frank-buss.de/cow.html>
- <https://es.wikipedia.org/wiki/Brainfuck>
- <https://r-knott.surrey.ac.uk/Fibonacci/fibtable.html>
- <https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/>
- <https://www.learncpp.com/cpp-tutorial/signed-integers/>