



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.....	3
Brainfuck.....	4
Lenguaje COW: Lógica y funcionamiento	4
Implementación de programas en COW	4
Repositorio GitHub del proyecto	5
Bibliografía.....	5

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' "
```

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 *Subtract*.

Brainfuck

Lenguaje COW: Lógica y funcionamiento

Implementación de programas en COW

Repositorio GitHub del proyecto

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

Bibliografía

- <https://esolangs.org/wiki/>
- <http://www.formauri.es/personal/pgimeno/compurec/LenguajesEsotericos.php>
- https://es.wikipedia.org/wiki/Lenguaje_de_programación_esotérico