

s

Instituto Politécnico Nacional

Escuela Superior de Computo

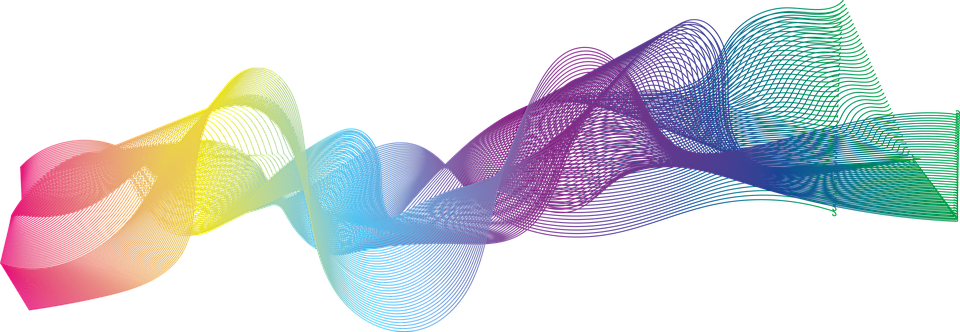
Unidad de aprendizaje: Compiladores 3CV8

Profesor: Saucedo Delgado Rafael Norman

**Proyecto: Compilador básico de lenguaje de programación C - Python**

***Alumno:*** ***García González Aarón Antonio***

Abril, 2020



Índice

[Objetivo 3](#_Toc37423255)

[Introducción 3](#_Toc37423256)

[Compilador 3](#_Toc37423257)

[Flex 3](#_Toc37423258)

[Yacc 3](#_Toc37423259)

[Desarrollo 4](#_Toc37423260)

[Categorías léxicas 4](#_Toc37423261)

[[1]. Identificadores 4](#_Toc37423262)

[[2]. Comentarios 4](#_Toc37423263)

[[3]. Palabras reservadas 5](#_Toc37423264)

[[4]. Operadores 5](#_Toc37423265)

[[5]. Constantes 5](#_Toc37423266)

[Gramática 6](#_Toc37423267)

[Resultados 7](#_Toc37423268)

[Conclusiones 7](#_Toc37423269)

[Anexo (Código) 7](#_Toc37423270)

[Referencias 7](#_Toc37423271)

# Objetivo

El desarrollo del proyecto busca reflejar los conocimientos estudiados a lo largo del curso por medio de la implementación de un compilador básico de lenguaje de programación C a lenguaje de programación Python en las versiones 3 en adelante.

# Introducción

## Compilador

Un compilador es aquel traductor que tiene como entrada una sentencia en lenguaje formal y como salida tiene un archivo ejecutable, es decir, realiza una traducción de un código de alto nivel a código máquina (también se entiende por compilador aquel programa que proporciona un archivo objeto en lugar del ejecutable final).

## Flex

Flex ( fast lexical analyzer generator ) es una alternativa de software libre y de código abierto a lex . Es un programa de computadora que genera analizadores léxicos (también conocidos como "escáneres" o "lexers").

Se usa con frecuencia como la implementación de lex junto con el generador de analizador Berkeley Yacc en sistemas operativos derivados de BSD (como ambos lexy yaccson parte de POSIX ), o junto con GNU bison (una versión de yacc) en puertos BSD y en distribuciones de Linux. A diferencia de Bison, flex no es parte del Proyecto GNU y no se publica bajo la Licencia Pública General de GNU, aunque la Fundación para el Software Libre produjo y publicó un manual para Flex.

## Yacc

Yacc es un programa para generar analizadores sintácticos. Las siglas del nombre significan Yet Another Compiler-Compiler, es decir, "Otro generador de compiladores más". Genera un analizador sintáctico (la parte de un compilador que comprueba que la estructura del código fuente se ajusta a la especificación sintáctica del lenguaje) basado en una gramática analítica escrita en una notación similar a la BNF. Yacc genera el código para el analizador sintáctico en el Lenguaje de programación C.

Puesto que el analizador sintáctico generado por Yacc requiere un analizador léxico, se utiliza a menudo juntamente con un generador de analizador léxico, en la mayoría de los casos lex o Flex, alternativa del software libre. El estándar de IEEE POSIX P1003.2 define la funcionalidad y los requisitos a Lex y Yacc.

# Desarrollo

## Categorías léxicas

### Identificadores

int check\_type()

{

/\*

\* pseudo code --- this is what it should check

\*

\* if (yytext == type\_name)

\* return(TYPE\_NAME);

\*

\* return(IDENTIFIER);

\*/

/\*

\* it actually will only return IDENTIFIER

\*/

return(IDENTIFIER);

}

### Comentarios

comment()

{

char c, c1;

loop:

while ((c = input()) != '\*' && c != 0)

putchar(c);

if ((c1 = input()) != '/' && c != 0)

{

unput(c1);

goto loop;

}

if (c != 0)

putchar(c1);

}

### Palabras reservadas

"auto"

"break"

"case"

"char"

"const"

"continue"

"default"

"do"

"double"

"else"

"enum"

"extern"

"float"

"for"

"goto"

"if"

"int"

"long"

"register"

"return"

"short"

"signed"

"sizeof"

"static"

"struct"

"switch"

"typedef"

"union"

"unsigned"

"void"

"volatile"

"while"

### Operadores

"..."

">>="

"<<="

"+="

"-="

"\*="

"/="

"%="

"&="

"^="

"|="

">>"

"<<"

"++"

"--"

"->"

"&&"

"||"

"<="

">="

"=="

"!="

";"

("{"|"<%")

("}"|"%>")

","

":"

"="

"("

")"

("["|"<:")

("]"|":>")

"."

"&"

"!"

"~"

"-"

"+"

"\*"

"/"

"%"

"<"

">"

"^"

"|"

"?"

### Constantes

0[xX]{H}+{IS}?

0{D}+{IS}?

{D}+{IS}?

L?'(\\.|[^\\'])+'

{D}+{E}{FS}?

{D}\*"."{D}+({E})?{FS}?

{D}+"."{D}\*({E})?{FS}?

## Gramática

# Resultados

# Conclusiones

# Anexo (Código)

# Referencias