Compiladores 2022

**PRÁCTICA 7**

|  |  |
| --- | --- |
| Nombre de la Práctica | **Creación de un convertidor de números a base distinta** |
| Nombre y Apellido | Ernesto Flores Machuca |
| Matrícula | 2193041595 |

**Objetivo**

Implementar programa en Flex/Bison para conversión de números enteros de base 10 a base 2 y vice versa.

**Descripción**

FLEX (Fast Lexical Analyzer Generator) es una herramienta para generar analizadores léxicos. La actividad de esta práctica consiste en escribir el programa usando la sintaxis de flex para crear una aplicación que permita hacer la conversión entre un número con base 10 al equivalente con base 2 y al revés.

**Procedimiento**

1) Familiarizarse con los ejemplos dados en las referencias [1] y [2]

2) Agregar una producción que permita determinar si es un número con base 10 o 2 corroborando el primer símbolo de la cadena, por ejemplo, “d 127” o “b 111111”. Consultar el ejemplo 4 del tutorial de la practica anterior.

3) Implementar el código completo que permita detección y conversión del número. 4) Realizar pruebas con 2 números de bases diferentes, haciendo la captura de pantalla.

5) Implementar la operación suma de dos números enteros de cualquier base (2 o 10), regresando el valor resultante con base 10, por ejemplo, “b10+d10” regresaría el número 12.

**Entregables**

**Programa para el convertidor de Binario a decimal:**

%{

/\* Definition section \*/

#include<stdio.h>

#include<stdlib.h>

**void** yyerror(**char** \*s);

%}

%token ZERO ONE

/\* Rule Section \*/

%%

**N:** L {printf("**\n**%d", $$);}

**L:** L B {$$=$**1**\***2**+$**2**;}

| B {$$=$**1**;}

**B:**ZERO {$$=$**1**;}

|ONE {$$=$**1**;};

%%

//driver code

**int** main()

{

**while**(yyparse());

}

yyerror(**char** \*s)

{

fprintf(stdout, "**\n**%s", s);

}

%{

/\* Definition section \*/

#include<stdio.h>

#include<stdlib.h>

#include"y.tab.h"

**extern** **int** yylval;

%}

/\* Rule Section \*/

%%

**0** {yylval=**0**;**return** ZERO;}

**1** {yylval=**1**;**return** ONE;}

[ \t] {;}

\n **return** **0**;

. **return** yytext[**0**];

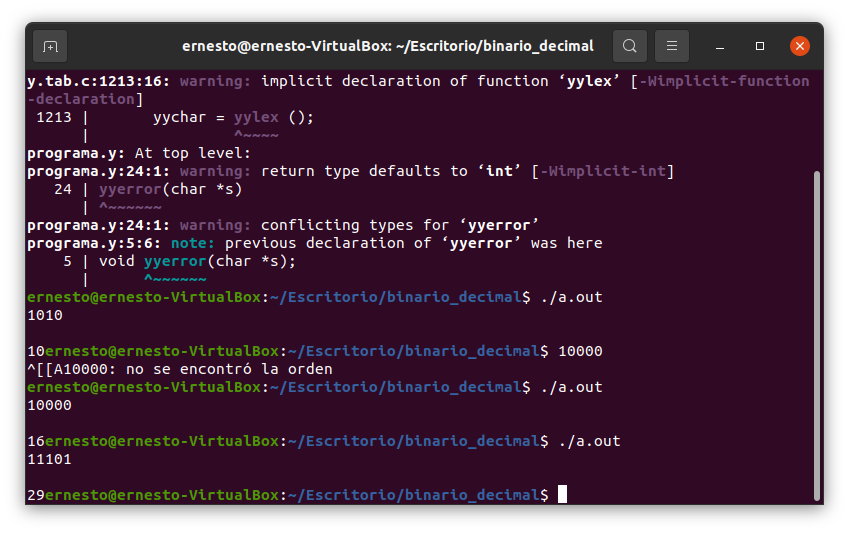
%%

**int** yywrap()

{

**return** **1**;

}



**Programa para el convertidor de decimal a binario:**

/\* Lex program for decimal to binary conversion \*/

%{

/\* Definition section \*/

#include<stdio.h>

**int** num, r, b=**0**, p=**1**;

%}

DIGIT [**0**-**9**]

/\* Rule Section \*/

%%

{DIGIT}+ { num=atoi(yytext);

**while** (num > **0**)

{

r= num % **2**;

b+= r\*p;

p\*= **10**;

num/= **2**;

}

printf("%d", b);

}

.|\n ECHO;

%%

// driver code

**int** main()

{

yylex();

**return** **0**;

}

Texto

Descripción generada automáticamente

**Conclusiones:**

Para concluir cabe mencionar que gracias a los programas Bison y Flex se ha podido comprobar un convertidor de Binario a decimal y viceversa, para esto se ha utilizado reglas gramaticales que identificaran la entrada y posteriormente poder traducir de un numero a otro.

**Referencias**

1. https://www.geeksforgeeks.org/yacc-program-for-binary-to-decimal-conversion/ 2. https://www.geeksforgeeks.org/lex-program-for-decimal-to-binary-conversion/

1