

Project Report

CSE 3212: Compiler Design Laboratory

Topic: Designing a Simple Compiler using flex and bison.

Submission Date: 15/06/2021

Submitted By:

Md. Asif Ud Daula

Roll: 1707031

Department of Computer Science and Engineering

Khulna University of Engineering & Technology, Khulna

Introduction:

In this project I have created my own compiler named Mycom compiler using Flex and Bison. Flex works as a scanner to match pattern and Bison generates rule and takes action according to matched items.

A **compiler** is a special program that processes statements written in a particular programming language and turns them into machine language that a computer's processor uses.

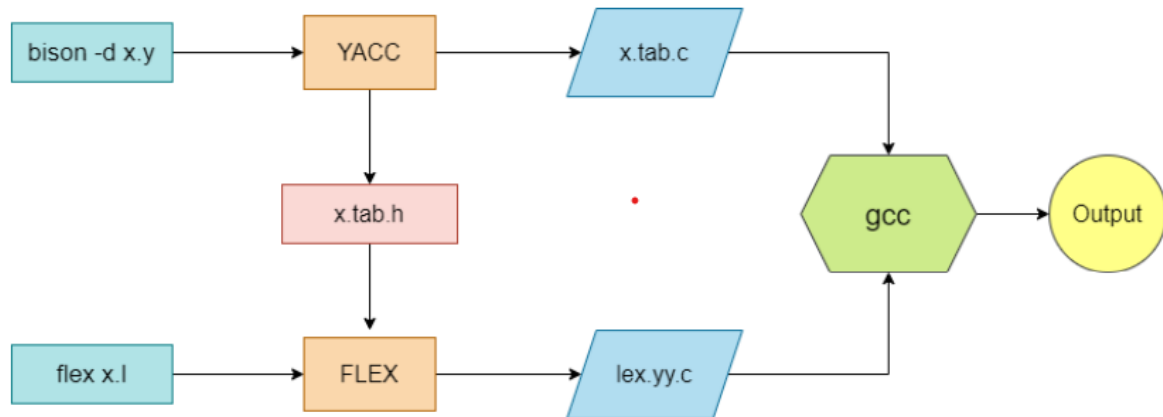
Lex is a computer program that generates lexical analyzers. Lex is commonly used with the yacc parser generator.

Bison, is a parser generator that is part of the GNU Project. Bison reads a specification of a context-free language and generates a parser that reads sequences of tokens and decides whether the sequence conforms to the syntax specified by the grammar.

Objectives:

- To know about the compiler
- To implement Context Free Grammar in the compiler.
- To know the top-down parser and the bottom-up parser.
- To know about Flex and Bison and how to use them to create a compiler.
- To create a new language and it's semantic and syntactic rules.

Work Flow:



Command:

- 1.bison -d Mycom.y
- 2.flex Mycom.l
- 3.gcc lex.yy.c Mycom.tab.c -o test
- 4.test

Shows output..

Manual Table:

In C language	In Mycom input	Lfile token
int main	dig prime	IntMain
()	[]	LP3 RP3
int	dig	Int
float	dec	Float

In C language	In Mycom input	Lfile token
char	byte	Char
;	/	sm
++	+	aadd
--	-	ssub
+	#	add
-	--	sub
/		divi
*	&	mult
=	<-	assign
<=	=<	Lequal
>=	=>	Gequal
>	>>	Gthan
<	<<	Lthan
if	is	If
else	no	Else
//	!!	n/a
/* */	!+ +!	n/a
while	go for	Loop
printf	out	Print

Syntax:

Declaration::::

Int I; ---->dig I/
float d;----->dec d/
char c;-----> byte c/

assign::::

a=12; ----> a<-12/
a=c; ----> a<-c/

addition::::

c=12+23;----->c<-12#23/
c=a+b; -----> c<-a#b/

subtraction::::

c=12-23;----->c<-12--23/
c=a-b; -----> c<-a--b/

other operators:::

a>23----->a>>23
b<23----->b<<23
a++;----->a+*/
b--;----->b-*/

Printf:::::

1. out "this is a print"/
2. out variable/

if/else:::::

```

is var>>12 [
.....statement/
]
No [
.....statement/
]

```

Loop:::::

```

int i<-0/
go for i<<10[
    out i/
]

```

Comments:::::

!! -->single line comment.

!+ +! -----> multiple line comment.

Built in Function:::::

oddeve var/-----> prints whether the var(variable) is odd or even;

Sample input:

```
1  dig prime[
2
3      dig a,r/
4      dec c/
5      byte d/
6
7      a<-12/
8      c<-13.5/
9      !!d<-'e'/
10
11     out a /
12     a+*/
13     out a /
14     a-*/
15     out a /
16
17     is a<<23 [
18         out "valid number"/]
19     no [
20         out "not valid"/]
21
22     !+is a<<15 [
23         out "a is a small number"/] +!
24
25     dig j<-1/
26     gofor j<<10 [
27         out j/
28     ]
29
30
31     dig i<-9/
32
33     oddeve i/
34
35     gofor i>>3 [
36         out j/
37     ]
38
39
40     out "This compiler is best for young gen"/
41
42 ]
```

Output:

```
This is a Compiler made by Asif
Its integer
Valid declaration
Its float
Valid declaration
Its character
Valid declaration
a = 12 (assigned)
c = 2 (assigned)
This is a single comment
Printin Variable value: a = 12
Printin Variable value: a = 13
Printin Variable value: a = 12
Printing if output: valid number
condition is True
This is a multi comment
Its integer
j = 1 (declared and assigned)
Valid declaration
loop
Printing Variable value3: 1
Printing Variable value3: 2
Printing Variable value3: 3
Printing Variable value3: 4
Printing Variable value3: 5
Printing Variable value3: 6
Printing Variable value3: 7
Printing Variable value3: 8
Printing Variable value3: 9
Its integer
i = 9 (declared and assigned)
Valid declaration
odd numberloop
Printing Variable value2: 1
Printing Variable value2: 1
Printing Variable value2: 1
Printing Variable value2: 1
Printing Variable value2: 1
Printing output: This compiler is best for young gen
compiled successfully
```

References:

- 1.<https://www.wikipedia.org/>
- 2.Book:Flex and Bison by John Levine