

DJILLALI LIABES UNIVERSITY OF SIDI BEL ABBES
FACULTY OF EXACT SCIENCES
DEPARTMENT OF COMPUTER SCIENCES



Module : Compilation 2
1ST YEAR OF MASTER'S DEGREE IN
NETWORKS, INFORMATION SYSTEMS & SECURITY (RSSI)
2021 / 2022

Analyse Syntaxique avec Bison
TP-03

Student:
HADJAZI M.Hisham
Group: 01 / RSSI

Module Instructor:
Dr. S.AISSAOUI
TP Instructor:
Dr . L.Niar

A paper submitted in fulfilment of the requirements for the
Compilation 2 TP-03

December 20, 2021

Contents

1	Solutions of Fiche TP-03	1
1.1	Exercise 1	2
1.1.1	1. Enregistrer le code dans un fichier nommé cal.y	2
	cal.y file	2
	cal.y file	3
1.1.2	2. Compiler le fichier enregistré avec Bison à l'aide des commandes suivantes :	3
	Execution of flex and bison commands	3
	Generated Files	4
1.1.3	3. Analyser les expressions suivantes:	4
	8*-6	4
	5+2	5
	6-2/2	5
	bonus (3*4)+(8/2)	6

Chapter 1

Solutions of Fiche TP-03

Notes regarding this solution :

This solution and the executions of the code in it was done in the following machine :

- *PC* : Lenovo IdeaPad S210 8GB
- *OS* : Linux Mint 20.2 Cinnamon Kernel v.5.4.0-88
- *Bison* : v3.5.1
- *Flex* : v2.6.4
- *gcc* : v9.3.0

1.1 Exercise 1

Soit le code suivant d'un analyseur syntaxique des expressions arithmétiques en Bison qui génère la suite de dérivation pour les expressions syntaxiquement correctes

1.1.1 1. Enregistrer le code dans un fichier nommé cal.y

cal.y file

```
1  %{
2  #include <stdio.h>
3  int yylex(void);
4  int yyerror(char* s);
5  %}
6
7  %token Tnb
8  %start S
9
10 %%
11 S : E {printf("1\n");}
12   | S E {printf("2\n");}
13   ;
14
15 E : T {printf("3\n");}
16   | E '+' T {printf("4\n");}
17   | E '-' T {printf("5\n");}
18   ;
19
20 T : F {printf("6\n");}
21   | T '*' F {printf("7\n");}
22   | T '/' F {printf("8\n");}
23   ;
24
25 F : Tnb {printf("9\n");}
26   | '(' E ')' {printf("10\n");}
27   ;
28 %%
29
30 int yyerror(char *s) {
31     fprintf(stderr, " Syntaxe incorrecte : %s\n", s);
32     return 0;
33 }
34 int yywrap() {
35     return 1;
36 }
37 int main(void)
38 {
39     yyparse();
40 }
```

cal.y file

```

41 %{
42 #include "cal.tab.h"
43 int yylex(void);
44 int yyerror(char* s);
45 %}
46
47 %%
48 [0-9]+ {yylval = atoi(yytext); return Tnb;}
49 [-+*/] {return yytext[0];}
50 [ \n\t]+ {;}
51 . yyerror("Invalid input character");
52 %%

```

1.1.2 2. Compiler le fichier enregistré avec Bison à l'aide des commandes suivantes :

1. flex cal.lex (cal.lex est l'analyseur lexical en Flex)
2. bison cal.y
3. gcc cal.tab.c -lfl -o calcul

Execution of flex and bison commands

1. flex cal.l
2. bison -d cal.y
3. gcc lex.yy.c cal.tab.c -o calc
4. ./calc

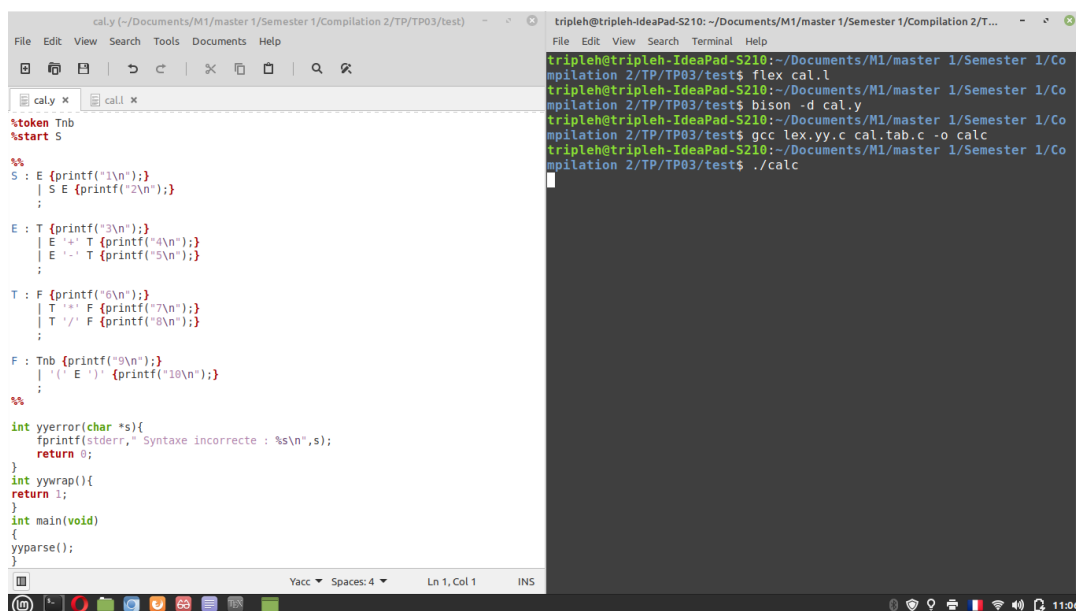


FIGURE 1.1: execution of commands

Generated Files

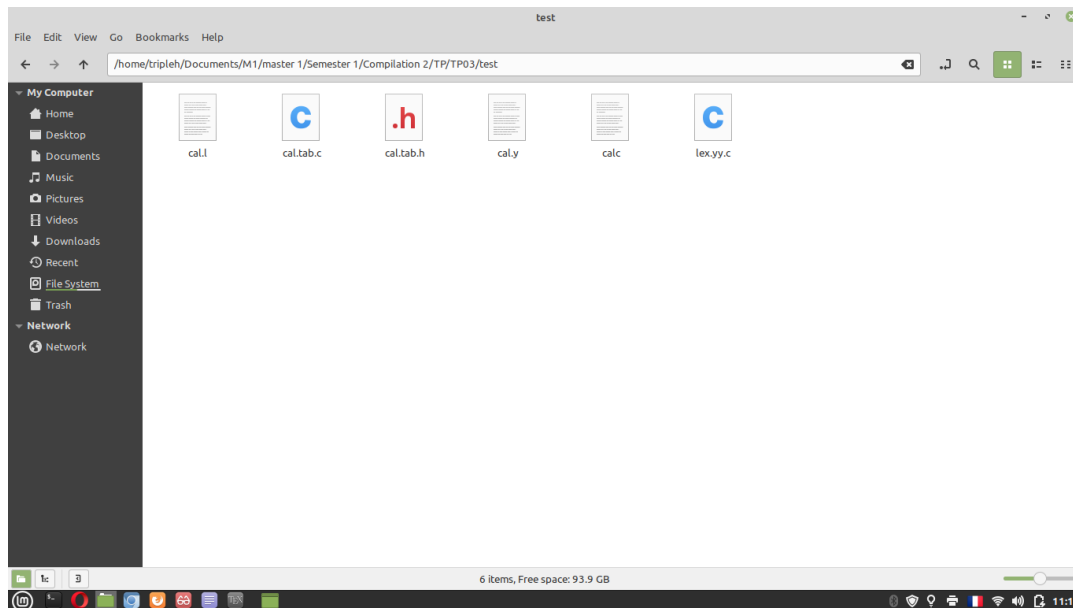


FIGURE 1.2: Generated Files

1.1.3 3. Analyser les expressions suivantes:

8*-6

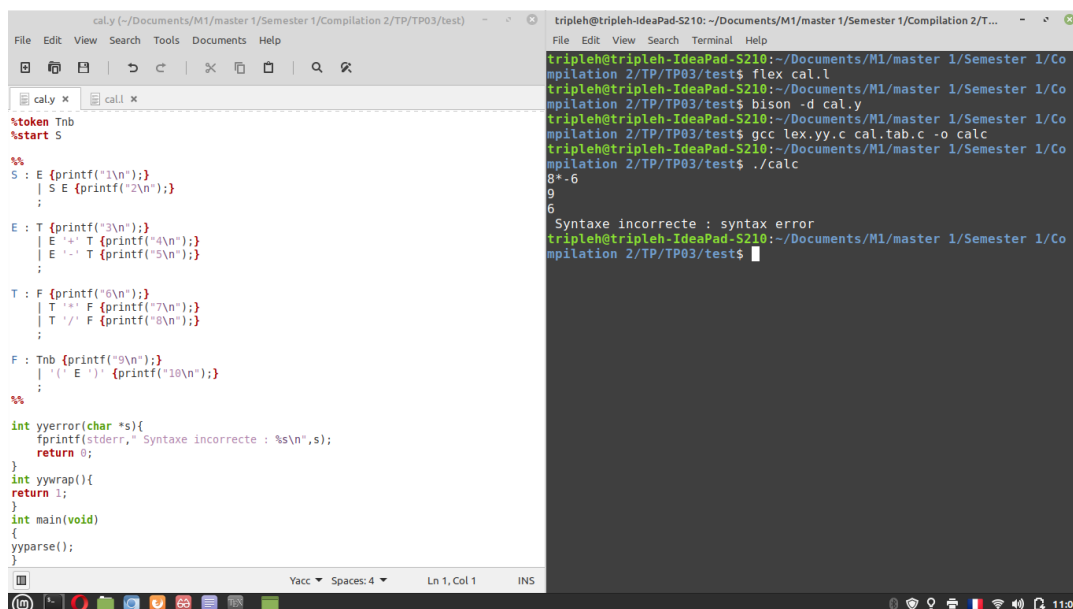


FIGURE 1.3: 8*-6

5+2

The screenshot shows a terminal window with the following content:

```

caly x
cali x

%token Tnb
%start S

%%
S : E {printf("\n\n");
   | S E {printf("2\n\n");
   ;

E : T {printf("3\n\n");
   | E '+' T {printf("4\n\n");
   | E '-' T {printf("5\n\n");
   ;

T : F {printf("6\n\n");
   | T '*' F {printf("7\n\n");
   | T '/' F {printf("8\n\n");
   ;

F : Tnb {printf("9\n\n");
   | '(' E ')' {printf("10\n\n");
   ;

%%

int yyerror(char *s){
    fprintf(stderr, " Syntax incorrecte : %s\n",s);
    return 0;
}

int yywrap(){
    return 1;
}

int main(void)
{
    yyparse();
}

```

The terminal output shows the result of the calculation for the expression '5+2':

```

tripleh@tripleh-IdeaPad-S210:~/Documents/M1/master 1/Semester 1/Compilation 2/TP/TP03/test$ ./calc
5+2
9
6
3
9
6

```

FIGURE 1.4: 5+2

6-2/2

The screenshot shows a terminal window with the following content:

```

caly x
cali x

%token Tnb
%start S

%%
S : E {printf("\n\n");
   | S E {printf("2\n\n");
   ;

E : T {printf("3\n\n");
   | E '+' T {printf("4\n\n");
   | E '-' T {printf("5\n\n");
   ;

T : F {printf("6\n\n");
   | T '*' F {printf("7\n\n");
   | T '/' F {printf("8\n\n");
   ;

F : Tnb {printf("9\n\n");
   | '(' E ')' {printf("10\n\n");
   ;

%%

int yyerror(char *s){
    fprintf(stderr, " Syntax incorrecte : %s\n",s);
    return 0;
}

int yywrap(){
    return 1;
}

int main(void)
{
    yyparse();
}

```

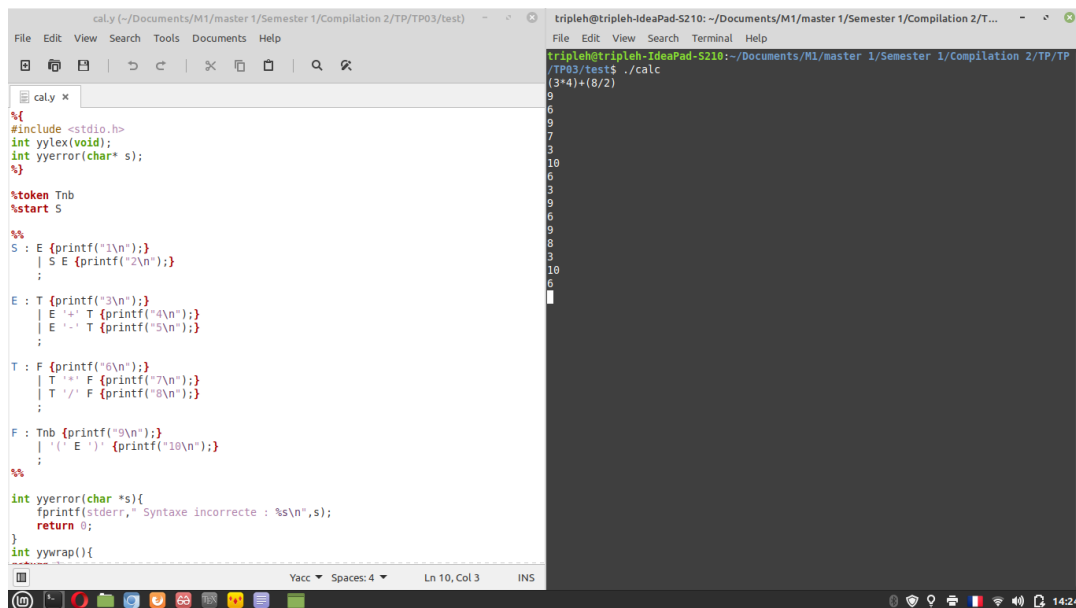
The terminal output shows the result of the calculation for the expression '6-2/2':

```

tripleh@tripleh-IdeaPad-S210:~/Documents/M1/master 1/Semester 1/Compilation 2/TP/TP03/test$ ./calc
6-2/2
6
6
3
9
6
8

```

FIGURE 1.5: 6-2/2

bonus $(3*4)+(8/2)$ 

```
caly x
%{
#include <stdio.h>
int yylex(void);
int yyerror(char* s);
}%

%token Tnb
%start S

%%
S : E {printf("1\n");}
  | S E {printf("2\n");}
  ;

E : T {printf("3\n");}
  | E '+' T {printf("4\n");}
  | E '-' T {printf("5\n");}
  ;

T : F {printf("6\n");}
  | T '*' F {printf("7\n");}
  | T '/' F {printf("8\n");}
  ;

F : Tnb {printf("9\n");}
  | '(' E ')' {printf("10\n");}
  ;

%%

int yyerror(char *s){
    fprintf(stderr, "Syntaxe incorrecte : %s\n", s);
    return 0;
}

int yywrap(){
    return 1;
}
```

```
tripleh@tripleh-ideaPad-S210: ~/Documents/M1/master 1/Semester 1/Compilation 2/TP/TP...
tripleh@tripleh-ideaPad-S210: ~/Documents/M1/master 1/Semester 1/Compilation 2/TP/TP...
/TP03/test$ ./calc
(3*4)+(8/2)
9
6
6
9
7
3
10
6
3
9
6
9
8
3
10
6
1
```

FIGURE 1.6: $(3*4)+(8/2)$