

這是我的所有檔案(還不包含pdf)且這些檔案都是還未被
lex/yacc/gcc編譯過的

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls  
compiler.h  makefile  parser.y  scanner.l  testData  
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

compiler.h: header file

parser.y: parser

scanner.l: scanner

testData: testData directory(testData1~testData10)

makefile: 我寫好了makefile編譯檔, 老師可以make一按鍵全部編譯完成為一個可執行檔叫做compiler

compiler final project

一鍵編譯成可執行檔案compiler

1.如上頁所示, 此為未編譯:

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler.h  makefile  parser.y  scanner.l  testData
```

2.先確定有 make utilities, 在命令行執行這段指令: `sudo apt install make`

3.可以一鍵編譯了, 如下圖, 在compiler_final_project這個目錄下執行這段指令: `make`

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler.h  makefile  parser.y  scanner.l  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ make
yacc -d -o parser.c parser.y
parser.y: warning: 2 reduce/reduce conflicts [-Wconflicts-rr]
gcc -Wall -c -o parser.o parser.c
parser.c: In function 'yyparse':
parser.c:2105:7: warning: implicit declaration of function 'yyerror'; did you mean 'yyerrok'? [-Wimplicit-function-declaration]
 2105 |         yyerror (YY_("syntax error"));
      |         ^~~~~~
      |         |
      |         yyerrok
lex -o scanner.c scanner.l
gcc -Wall -c -o scanner.o scanner.c
scanner.c:1373:16: warning: 'input' defined but not used [-Wunused-function]
 1373 |     static int input (void)
      |
scanner.c:1330:17: warning: 'yyunput' defined but not used [-Wunused-function]
 1330 |     static void yyunput (int c, char * yy_bp )
      |
gcc -o compiler parser.o scanner.o -ly -lfl
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler  compiler.h  makefile  parser.c  parser.h  parser.o  parser.y  scanner.c  scanner.l  scanner.o  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

即產生了scanner.c和parser.c和最重要的我們編出來的可執行檔 `compiler`

編譯完成之後可以執行了

在compiler_final_project這個目錄下

執行: `./compiler < testData/testDataX`

即顯示結果, X為1,2,3,...,10 因為有10筆testData

學生發誓以上指令並無任何對系統產生危害的指令, 僅僅是安裝make和執行make

1.sudo apt install make

2.make

這是老師ppt的規格

%%the beginning of an test data for Micro/Ex
Program testP

Begin
declare I as integer;
declare A,B,C,D, LLL[100] as float;

FOR (I:=1 TO 100)
 A:=-LLL[I]+B*D-C;
ENDFOR

IF (A>=10000.0) THEN
 print(1);
ELSE
 print(2,1.4);
ENDIF

End

定義/說明:

- 1.所有關鍵詞(非變數)的大小寫會依照前一頁的圖, 例如Begin, declare, as, ENDIF, Program, FOR, print還有註解%%等等。
- 2.註解僅存在於最一開始, Program XXX前面, 註解可為多行或是無註解。
- 3.四則運算滿足結合律及運算子優先權大小: 負號 > 乘除 > 加減 且左結合。
- 4.變數必須宣告才能使用, 否則產生error message: Undefined variable。
- 5.在if判斷式中兩個operands比較時, 兩個operands data type必須一致(皆為int或float)否則產生error message: Incompatible type of variable。
- 6.print中的參數個數可以是多個(變動參數個數)。
- 7.給定input program, compiler成功產生對應的組合語言並輸出至螢幕上的充要條件為input program語法無任何錯誤, 且將於compiler完一併從預設stdout的(螢幕上)輸出。

testData說明：每個testData想要測的東西

testData1->僅有declare且無註解

testData2/3/4->都是declare + for 但是expression不一樣

testData5->declare + for + if/else

testData6/7->多行註解

testData8->print函式有多個參數

testData9->if判斷式中兩個operand不同資料型別

testData10->變數未宣告就使用

下列數頁展示各個testData執行測試結果以確定無誤(擔心自己寫的scanner/parser在老師的機器上有可能會沒辦法被lex/yacc compile成功, 因為和我的ubuntu上lex/yacc版本不同)

testData1

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler  compiler.h  makefile  parser.c  parser.h  parser.o  parser.y  scanner.c  scanner.l  scanner.o  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData1
 1 Micro/Ex
 2
 3 Program moumou123
 4
 5
 6 Begin
 7
 8 declare mu[222],yu[44],xy ,yy[100],vv as float;
 9
10 declare qwe, r44, y333[11],uv22[123],pp as integer;
11
12 End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData1
START moumou123
Declare mu,Float_array,222
Declare yu,Float_array,44
Declare xy,Float
Declare yy,Float_array,100
Declare vv,Float
Declare qwe,Integer
Declare r44,Integer
Declare y333,Integer_array,11
Declare uv22,Integer_array,123
Declare pp,Integer
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ █
```

僅有declare

testData2

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler  compiler.h  makefile  parser.c  parser.h  parser.o  parser.y  scanner.c  scanner.l  scanner.o  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat testData/testData2 -n
 1 Micro/Ex
 2
 3 Program mowmou123
 4 Begin
 5   declare I as integer;
 6   declare A,B,C,D, LLL[100] as float;
 7
 8   FOR (I:=1 TO 100)
 9     A:=-LLL[I]+B*D-C;
10   ENDFOR
11
12 End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData2
      START mowmou123
      Declare I,Integer
      Declare A,Float
      Declare B,Float
      Declare C,Float
      Declare D,Float
      Declare LLL,Float_array,100

lb&1:  I_STORE 1,I
      F_UMINUS LLL[I],T&1
      F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_SUB T&3,C,T&4
      F_STORE T&4,A
      INC I
      I_CMP I,100
      JL lb&1

      Declare T&1,Float
      Declare T&2,Float
      Declare T&3,Float
      Declare T&4,Float
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

僅有declare+for

testData3

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat testData/testData3 -n
 1 Micro/Ex
 2
 3 Program moumou123
 4 Begin
 5 declare I as integer;
 6 declare A,B,C,D, LLL[100] as float;
 7
 8 FOR (I:=1 TO 100)
 9     A:=-LLL[I]+B-C;
10 ENDFOR
11
12 End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData3
START moumou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

lb&1:  I_STORE 1,I
      F_UMINUS LLL[I],T&1
      F_ADD T&1,B,T&2
      F_SUB T&2,C,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb&1

      Declare T&1,Float
      Declare T&2,Float
      Declare T&3,Float
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

僅有declare+for

testData4

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData4
 1 Micro/Ex
 2 Program moumou123
 3 Begin
 4   declare I as integer;
 5   declare A,B,C,D, LLL[100] as float;
 6
 7   FOR (I:=1 TO 100)
 8     A:=-LLL[I]+B*D;
 9   ENDFOR
10
11 End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData4
START moumou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

I_STORE 1,I
lb&1: F_UMINUS LLL[I],T&1
      F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb&1

      Declare T&1,Float
      Declare T&2,Float
      Declare T&3,Float
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

僅有declare+for

testData5

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData5
1  Micro/Ex
2  Program mowmou123
3  Begin
4      declare I as integer;
5      declare A,B,C,D, LLL[100] as float;
6
7      FOR (I:=1 TO 100)
8          A:=-LLL[I]+B*D;
9      ENDFOR
10
11     IF (A>=100000.0) THEN
12         print(A+3.14);
13     ELSE
14         print(2,1.4);
15     ENDIF
16
17 End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData5
START mowmou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

I_STORE 1,I
lb&1: F_UMINUS LLL[I],T&1
      F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb&1

      F_CMP A,100000.0
      JL lb&2
      F_ADD A,3.14,T&4
      CALL print,T&4
      J lb&3
lb&2: CALL print,2,1.4
lb&3: HALT mowmou123

Declare T&1,Float
Declare T&2,Float
Declare T&3,Float
Declare T&4,Float
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

declare+for+if/else

testData6

```
nownow@nownow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
nownow@nownow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData6
 1  %%the beginning of an test data for
 2  %% gcc gdb g++ gnu
 3  %%arm a57 a73 a53 x86-64 IA32 IA64 trustzone nvlink
 4  %%rbtree treap skiplist fabonacci heap kmp
 5  %% asd      ksaml   objdump dsd
 6
 7
 8  Micro/Ex
 9
10  Program mounou123
11  Begin
12  declare I as integer;
13  declare A,B,C,D, LLL[100] as float;
14
15  FOR (I:=1 TO 100)
16      A:=-LLL[I]*B*D;
17  ENDFOR
18
19  IF (A>=100000.0) THEN
20      print(A*3.14);
21  ELSE
22      print(2,1.4);
23  ENDIF
24
25  End
nownow@nownow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData6
START mounou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

I_STORE 1,I
lb81: F_MINUS LLL[I],T&1
      F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb81

      F_CMP A,100000.0
      JL lb82
      F_ADD A,3.14,T&4
      CALL print,T&4
      J lb83
lb82: CALL print,2,1.4
lb83: HALT mounou123

Declare T&1,Float
Declare T&2,Float
Declare T&3,Float
Declare T&4,Float
nownow@nownow-Cyborg-15-A12UDX:~/compiler_final_project$
```

testData7

```
ounow@ounow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
ounow@ounow-Cyborg-15-A12UDX:~/compiler_final_project$ cat testData/testData7 -n
1 %!kdwddq
2 %% fw goo grgg
3
4
5 Micro/Ex
6
7 Program mounou123
8 Begin
9 declare I as integer;
10 declare A,B,C,D, LLL[100] as float;
11
12 FOR (I:=1 TO 100)
13     A:=-LLL[I]+B*D;
14 ENDFOR
15
16 IF (A=-100000.0) THEN
17     print(A+3.14);
18 ELSE
19     print(2,1.4);
20 ENDIF
21
22 End
ounow@ounow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData7
START mounou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

I STORE 1,I
F_LMINUS LLL[I],T&1
b&1: F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb&1

      F_CMP A,100000.0
      JL lb&2
      F_ADD A,3.14,T&4
      CALL print,T&4
      J lb&3
b&2: CALL print,2,1.4
b&3: HALT mounou123

Declare T&1,Float
Declare T&2,Float
Declare T&3,Float
Declare T&4,Float
ounow@ounow-Cyborg-15-A12UDX:~/compiler_final_project$
```

多行註解

testData8

```
nounow@nounow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler compiler.h makefile parser.c parser.h parser.o parser.y scanner.c scanner.l scanner.o testData
nounow@nounow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData8
 1  %%the beginning of an test data for
 2  %% gcc gdb g++ gnu
 3  %%arm a57 a73 a53 x86-64 IA32 IA64 trustzone nvlink
 4  %%rbtree treap skiplist fabonacci heap knp
 5  %% asd ksaml objdump dsd
 6
 7
 8  Micro/Ex
 9
10  Program nounou123
11  Begin
12  declare I as integer;
13  declare A,B,C,D, LLL[100] as float;
14
15  FOR (I:=1 TO 100)
16  A:=-LLL[I]*8*0;
17  ENDFOR
18
19  IF (A<=100000.0) THEN
20  print(A*3.14);
21  ELSE
22  print(2,1.4,3, 4.66, 999.99);
23  ENDF
24
25  End
nounow@nounow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData8
START nounou123
Declare I,Integer
Declare A,Float
Declare B,Float
Declare C,Float
Declare D,Float
Declare LLL,Float_array,100

lb&1:  I_STORE I,I
      F_UMINUS LLL[I],T&1
      F_MUL B,D,T&2
      F_ADD T&1,T&2,T&3
      F_STORE T&3,A
      INC I
      I_CMP I,100
      JL lb&1

      F_CMP A,100000.0
      JL lb&2
      F_ADD A,3.14,T&4
      CALL print,T&4
      J lb&3
lb&2:  CALL print,2,1.4,3,4.66,999.99
lb&3:  HALT nounou123

      Declare T&1,Float
      Declare T&2,Float
      Declare T&3,Float
      Declare T&4,Float
```

print函式有多個參數

testData9

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler  compiler.h  makefile  parser.c  parser.h  parser.o  parser.y  scanner.c  scanner.l  scanner.o  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat testData/testData9 -n
 1 %%the beginning of an test data for
 2 %% gcc gdb g++ gnu
 3 %%arm a57 a73 a53 x86-64 IA32 IA64 trustzone nvlink
 4 %%rbtree treap skiplist fabonacci heap kmp
 5 %% asd      ksaml  objdump dsd
 6
 7
 8 Micro/Ex
 9
10 Program mowmou123
11 Begin
12   declare I, X as integer;
13   declare A,B,C,D, LLL[100] as float;
14
15   FOR (I:=1 TO 100)
16       A:=-LLL[I]+B*D;
17   ENDFOR
18
19   IF (I>=100000.0) THEN
20       print(A+3.14);
21   ELSE
22       print(2,1.4,3, 4.66, 999.99);
23   ENDIF
24
25   End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData9
Incompatible type of variable I : Integer, expect Float data type.
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

if判斷式中兩個operand不同資料型別

testData10

```
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ls
compiler  compiler.h  makefile  parser.c  parser.h  parser.o  parser.y  scanner.c  scanner.l  scanner.o  testData
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ cat -n testData/testData10
 1  %%the beginning of an test data for
 2  %% gcc gdb g++ gnu
 3  %%arm a57 a73 a53 x86-64 IA32 IA64 trustzone nvlk
 4  %%rbtree treap skiplist fabonacci heap kmp
 5  %% asd      ksaml  objdump dsd
 6
 7
 8  Micro/Ex
 9
10  Program mowmou123
11  Begin
12    declare I as integer;
13    declare A,B,C,D, LLL[100] as float;
14
15    FOR (I:=1 TO 100)
16      A:=-LLL[I]+B*D;
17    ENDFOR
18
19  IF (X>=100000.0) THEN
20    print(A+3.14);
21  ELSE
22    print(2,1.4,3, 4.66, 999.99);
23  ENDIF
24
25  End
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$ ./compiler < testData/testData10
Undefined variable:X
mowmow@mowmow-Cyborg-15-A12UDX:~/compiler_final_project$
```

變數未宣告就使用

What you have learned and experienced during the implementation of Micro/Ex compiler.

這個compiler的project過程中有幾個重點，我在過程中maintain了一個global symbol table根據我定義好的symbol structure(裡面記載許多屬性)，我把每個變數和常數都視為一個symbol(不包含關鍵字)，這樣子我只要在過程中不斷回傳這些symbol structure pointer我就可以隨時取用這些symbol的所有資訊。另外我在過程中的輸出都會先使用sprintf輸出到一個string array，唯有成功compile完成，在最後我只要將這個string array的所有string輸出就好，就是把所有組合語言輸出到stdout了。還有我在lex跟yacc之間會定義很多變數，在其中一邊我只要用extern把這些變數傳遞進來，等於說lex和yacc之間可以共用與傳遞許多變數，來達成我想要的目的。還有一點是我將declare/for/ifelse都邏輯上視為不同的block或可以說是不同的state，當我進入不同block/state後，我就可以根據我所在的block/state來做該做的事，這樣滿方便我coding的。

What you have learned and experienced during the implementation of Micro/Ex compiler.

最後一點是我在撰寫這次project之前，我先花了一些時間，將老師之前的chXXX.y和chXXX.l的那些範例將他們分別compile成對應的.c檔，然後去trace那些.c檔案，這樣更有助於我了解lex和yacc的一些關係以及他們之間是怎麼運作的。

Copyright Claim: Do you make the implementation yourself?

學生 姜任懋 台灣大學資訊網路與多媒體研究所 R12944042 本人聲明並無參考/抄襲任何紙本/網路上的資源(github/chatgpt)的行為, 本次project全由學生自己根據授課老師內容親自完成的, 本人於project release隔天花費三天的時間完成此次project程式碼的撰寫並於6/3~6/4完成report的撰寫, 若有任何違規行為, 願意接受校規處置。

Any thing you would like to let G.H.Hwang know.

謝謝老師這學期的教導，因為自己本身是在做系統軟體和計算機架構方面相關的研究，為了想更深入且透徹了解程式的運行所以一直很想來修編譯器，但台大資工已經有一段時間都沒有老師在教編譯器了，有學長去電機系修的心得是並沒有太大收穫，所以就想來修看看老師的課，我的教授也有和我說老師是門教這門課也教很教很久了，可以學到很多。實際上修老師的課我覺得收穫很多，我覺得我是很少有課可以認真聽完每週三節課的，因為老師可以把一些很難的概念講解的很好懂，或是舉一些讓人一聽就懂的例子，因為編譯器本來就是一門很難的課程，我之前自己讀沒幾頁讀不懂就放棄了，謝謝老師讓我這學期收穫這麼多，也讓我有能力之後接觸更多更深入編譯器相關的知識，總之我覺得這門課幫我打好滿多編譯器相關的基礎的，謝謝老師。