Compiler Final Project

姓名:魏駿帆

系級:資訊102

學號:F74981183

1. 檔案

F74981183.l

F74981183.y

makefile

TM.exe

1. 程式編譯
   1. 方法一  
      flex F74981183.l

yacc -d F74981183.y

cc -c lex.yy.c y.tab.c

cc -o output lex.yy.o y.tab.o -lfl

./output <test.txt

* 1. 方法二(利用makefile)

make <==產生obj.tm 並顯示quadruple

make obj <==將obj.tm餵給tiny machine

註:測試檔檔名需為test.txt

1. 完成功能(85%)
   1. 加法
   2. 減法
   3. 乘法
   4. 除法
   5. 先乘除後加減
   6. 分配常數給變數
   7. 輸入(input(x))
   8. 輸出(output(x))
   9. If
   10. While loop
   11. For loop
   12. 一維陣列
   13. Relational Operation
2. 做法&程式架構
   1. 程式架構
3. 利用lex & yacc，分析input檔案
4. 在parsing的過程中，若遇到符合的grammar，則在節點中做相對應的動作，並把指令加入到quadruple中
5. parse完整個檔案之後，顯示quadruple
6. 將quadruple轉換成instruction
7. 最後將instruction轉換成tiny machine 可以運作的四位元碼
8. 輸出四位元碼到obj.tm
9. 將obj.tm餵給tiny machine執行
   1. 加減乘除(先乘除後加減)
      1. Bottom up parsing當parse到factor時，一直向上回傳至有運算元的grammar
      2. Grammar會先判斷factor為數字或變數，若為數字則new一個新的變數，將數字存入變數後，再加入指令，若為變數則直接加入指令。
      3. 將結果存入變數後向上傳，若之後還有運算則可以取用。
      4. 因為grammar的關係，會先運算乘除再運算加減
   2. If
      1. 先將判斷式加入到quadruple，並啟動紀錄branch
      2. 當if中間的指令都加入到quadruple後，再回去修改判斷式要跳的指令數
   3. While loop
      1. 先將判斷式加入到quadruple，並啟動紀錄branch
      2. 當while中間的指令都加入到quadruple後，先加goto指令跳回判斷式，再回去修改判斷式要跳的指令數。
   4. For loop
      1. 先將判斷式加入到quadruple，並啟動紀錄branch
      2. 當while中間的指令都加入到quadruple後，先執行forloop要對變數執行的動作，再加入goto指令跳回判斷式，再回去修改判斷式要跳的指令數。
   5. 一維陣列
      1. 宣告變數時，若為陣列則會判斷其要宣告的大小，並分配給他
      2. 若運算中運用到陣列，則程式會先去讀其陣列的起始address，加上offset後即為其真正的記憶體位置。
   6. Relational Operation
      1. 已含在上列if、while、for裡面，當遇到相對應的relop，則會加入相對的指令到quadruple裡面。