

提案：

從大一開始學程式，我就對 PL 相當有興趣。有一次跟學長的談話，發現 Compiler 實為語言的一個核心，便與同學決定要寫出 Compiler。所以其實在暑假我們就有在看計算理論跟編譯器相關書籍，並把 protocol 定好。剛剛好這學期組合語言，學到 intel assembly，不過又只有一個作業(雖然我好像寫了兩個)。好像並沒有很深的練習到，倒是 ARM 學了不少。因此，決定在此時開始動工寫 Compiler，剛剛好也練習 Intel Assembly。

NPC 的概觀：

Compiler 的架構，就我們所知有：Lexer, Parse, Symbol table 等 組成前端。Code generator 在後端。我們這次依照這樣的設計原則。

Lexer 像一台 FA，FA 畫完就做完了，不過 state 的轉換要注意，所以畫一台好的 FA，是重要的地方。在 0.0.1 版的 FA 有明顯 redundancy，就是一台不好的 FA。

Parser 用 CFG 的子集合形成文法，但是不好的文法可能讓時間複雜度很高。如果有 ambiguous 文法還直接爆掉，相當不好。文法的設計，對 Parser 相當重要。我們參考 C99 的文件，改自其文法。

Symbol table 存變數的位置跟型態。以至於可以在 C 生成組語的 Code 時知道變數位置，或是知道這個變數有沒有被宣告。

Code Generation 需要大量的組語，我們失敗了很多次，才想到拿 eax 跟 stack 頂端放上一次用過的東西的這種通訊協定。

心得：

Compiler 的設計是長而辛苦的。除了要瞭解一點計算理論，還要懂組合語言。這次做作業有花很多的時間在懂文法跟語言的架構。然後覺得 GCC 真不是人寫的東西！

參考書目：

Compilers.

Introduction to theory of computation.

C99 standard document

特別感謝：

莊永裕 老師
陳學毅 學長
侯昆邦 學長
溫在宇 同學
馮俊崧 同學
陳柏龍 同學

NPC 1.0.0 版，介紹：

NPC (not pretty-perfect C-compiler)是 2008 年寒假前，由兩個大二學生寫出來的。他們分別是台大資訊系的蔡明亨(B95902028)跟楊耀翔(B95902078)。預計此作品將於 2008 年寒假加入 Fancy Cat 公司的 freeware，在 Fancy Cat 的網站 (<http://fancycat.csie.org>)，給予下載。

NPC 1.0.0 版，使用說明：

NPC 是一個 C-compiler，它配合 Visual Studio 的開發環境，使用有名的 Irvine32.inc 作為組語包含檔案，因此，欲使用 NPC，要有 Visual Studio 以及 Irvine 這個 library。

以下介紹如何使用 NPC：

首先下載過後，會有一個 npcx.x.x.yyy(x 均為數字,yyy 是壓縮檔案副檔名)的壓縮檔案，解壓縮之，然後會看到一堆檔案在資料夾裡面。

通常來講 test 會是我們的工作空間，因此，進入 test。如果有看到 compiler.exe，那就可以直接用了；否則，必須編譯 source files。

編譯 source file 的方法：

在 npc 的目錄下鍵入：

```
gcc -o compiler codegen.c parser.c main.c debug.c token.c lex.c symbol_table.c sysinit.c
```

```
move compiler test
```

兩行指令，即可回到 test 目錄下，做接下來的準備工作。

使用 NPC：

首先，將 Visual studio 打開，如果發現自己沒有 Irvine 的 lib，就去問 google，它會告訴你在哪裡下載，或是此資料夾下有個 IrvineExamples 檔案，可直接雙擊使用。另外可以去以下網址：

<http://w.csie.org/~cyy/courses/assembly/07fall/lectures/handouts/MASM.pdf>

(cyy 課程的小金助教的投影片，看如何使用 VC，以及如果 include path)

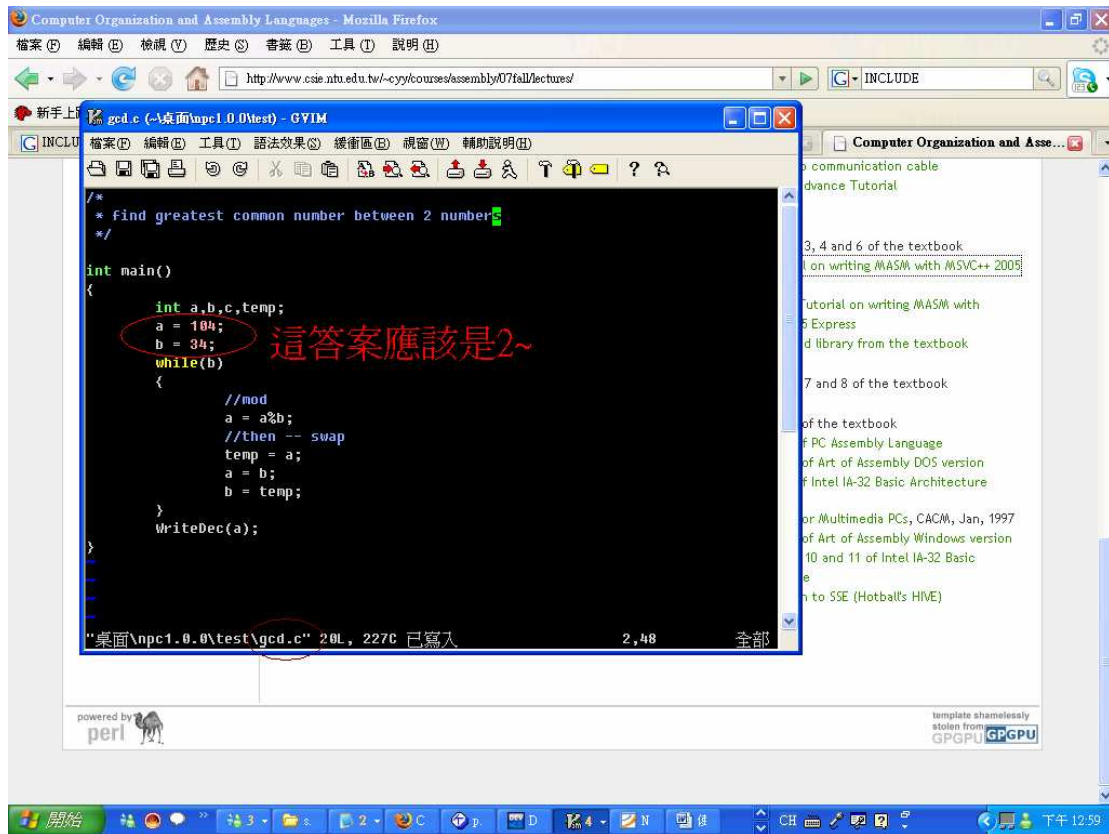
都好了以後，就可以放著專案待命了。

把想寫編譯程式放到 test 資料夾，鍵入：[compiler] [input file] [output file] 然後再去 VC 中，讀入檔案，Compile 之，即可。

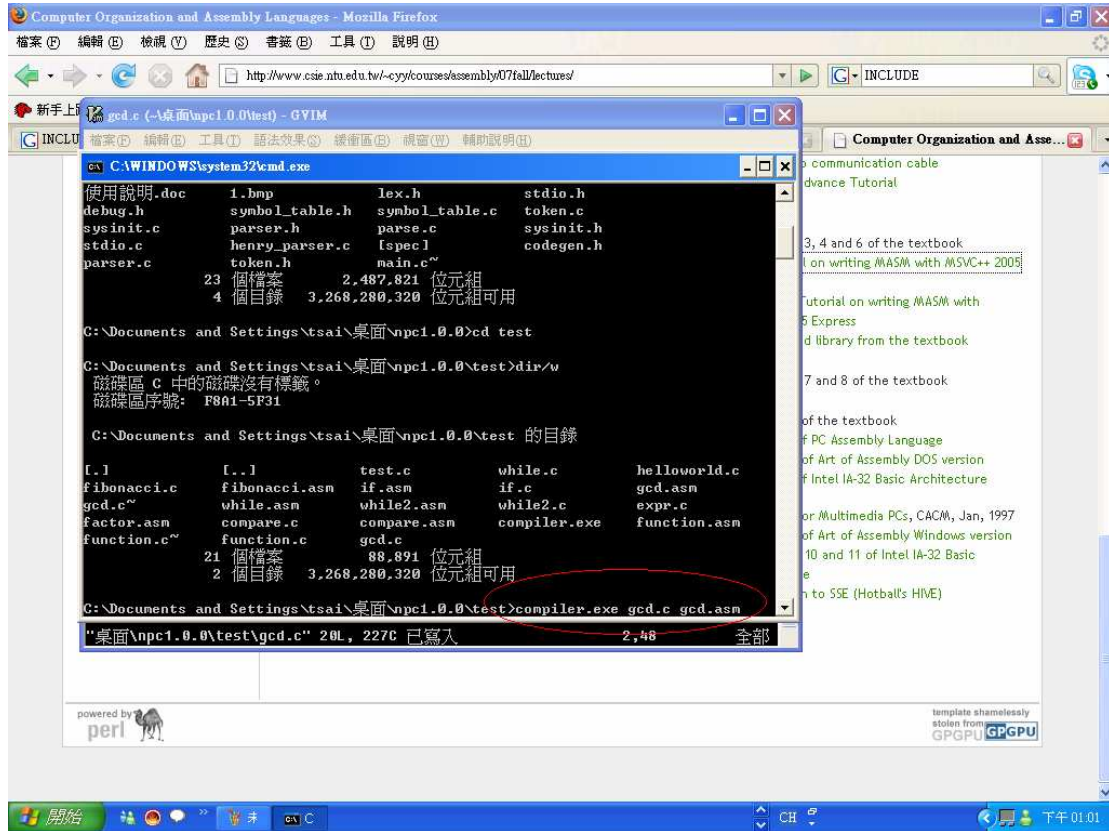
範例：

以下為一個簡單的範例：

1.要有個簡單的程式，我用求兩數字，用輾轉鄉除法，求出最大公因數的範例。

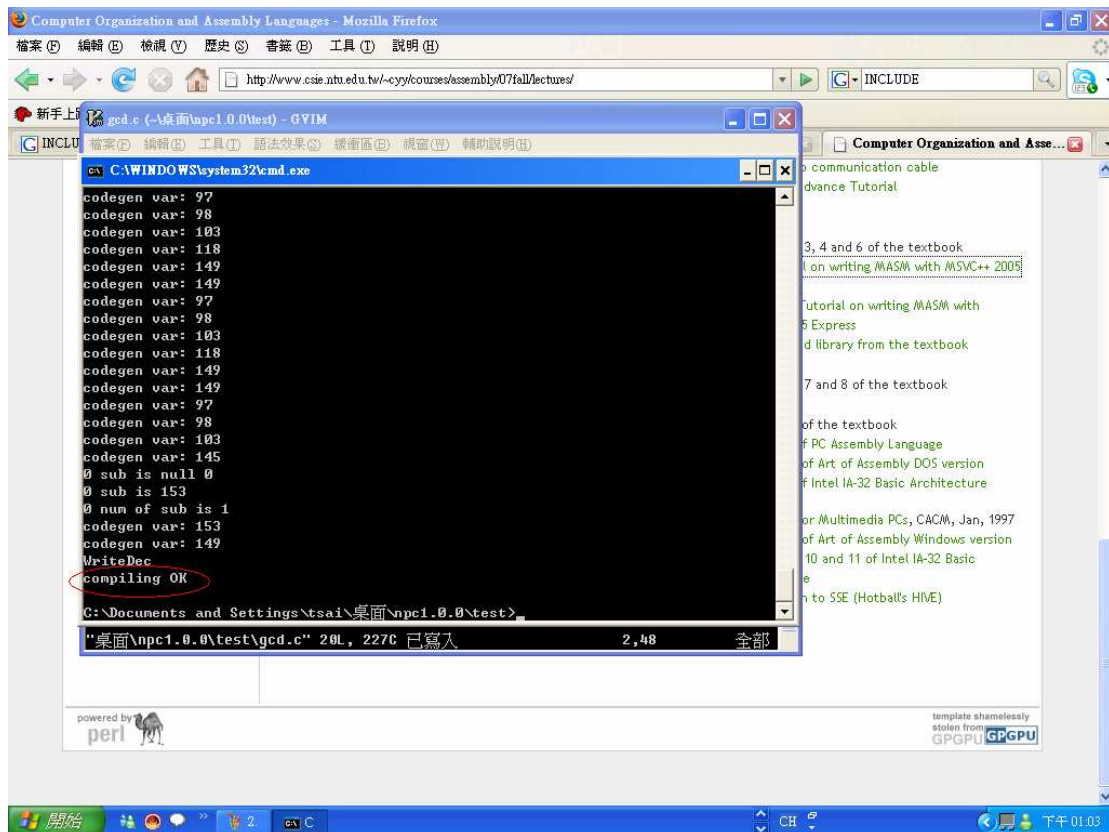


我存在 test 下面的 gcd.c，然後準備 compiler
2.Compile 它



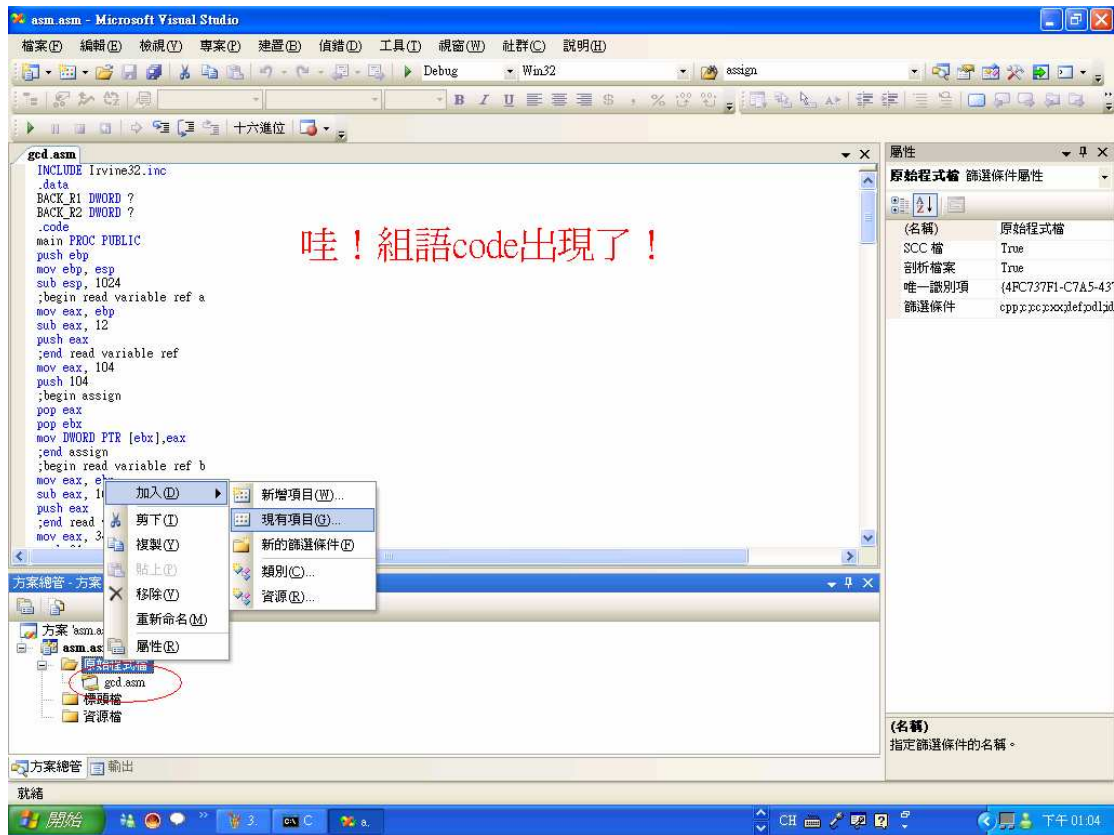
輸入 compiler.exe gcd.c gcd.asm

3.過了耶~~



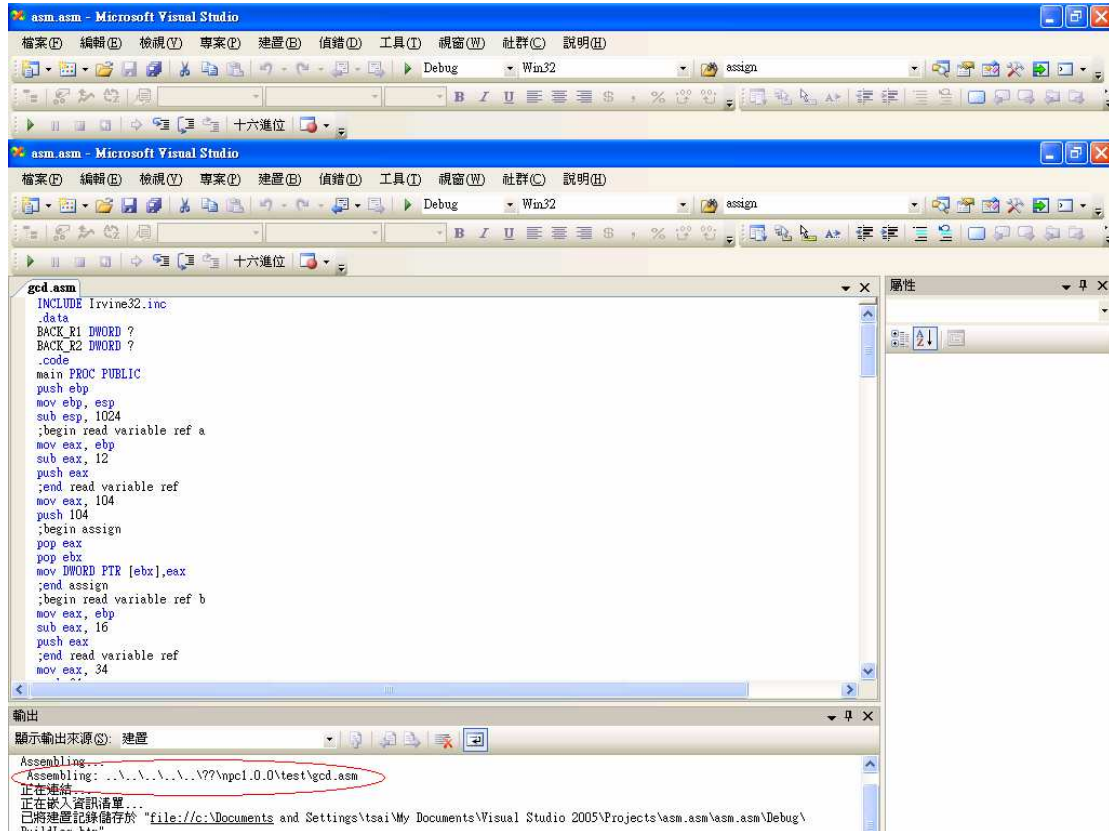
看到 compiling OK 的字眼，就是過了～～

4. 進去 VS 讀入檔案

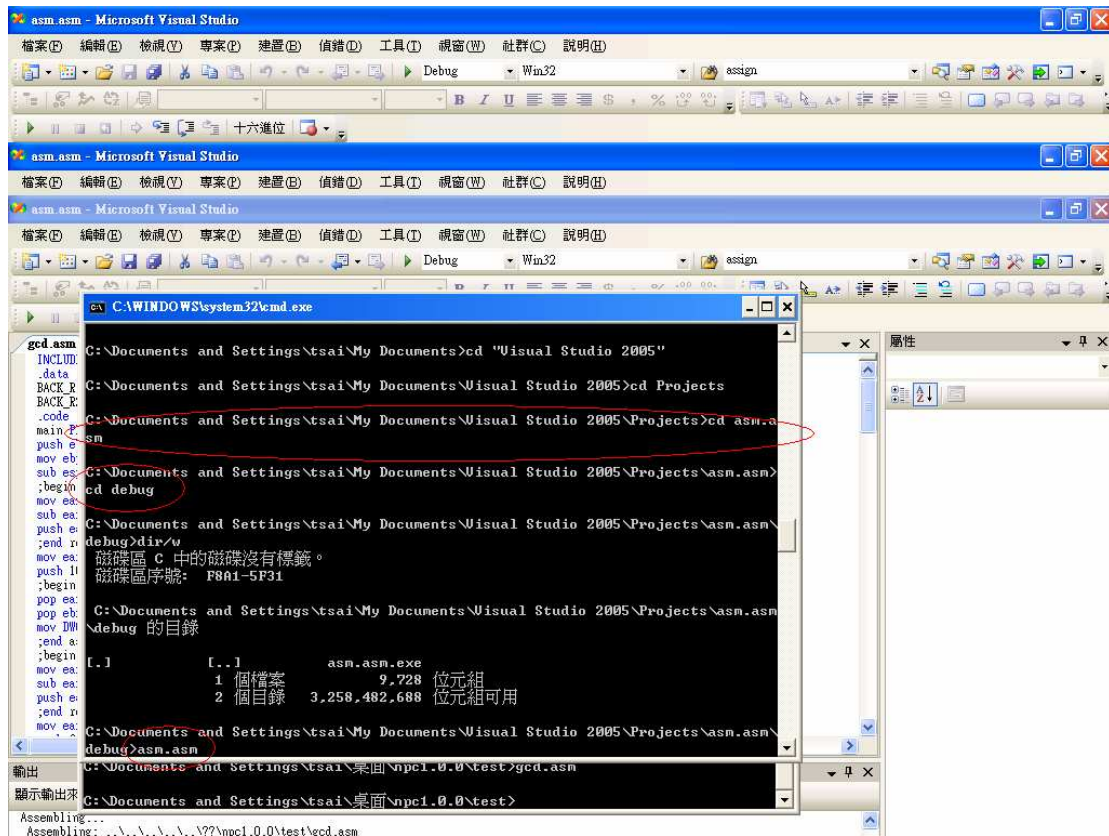


像這樣加入那個 gcd.asm

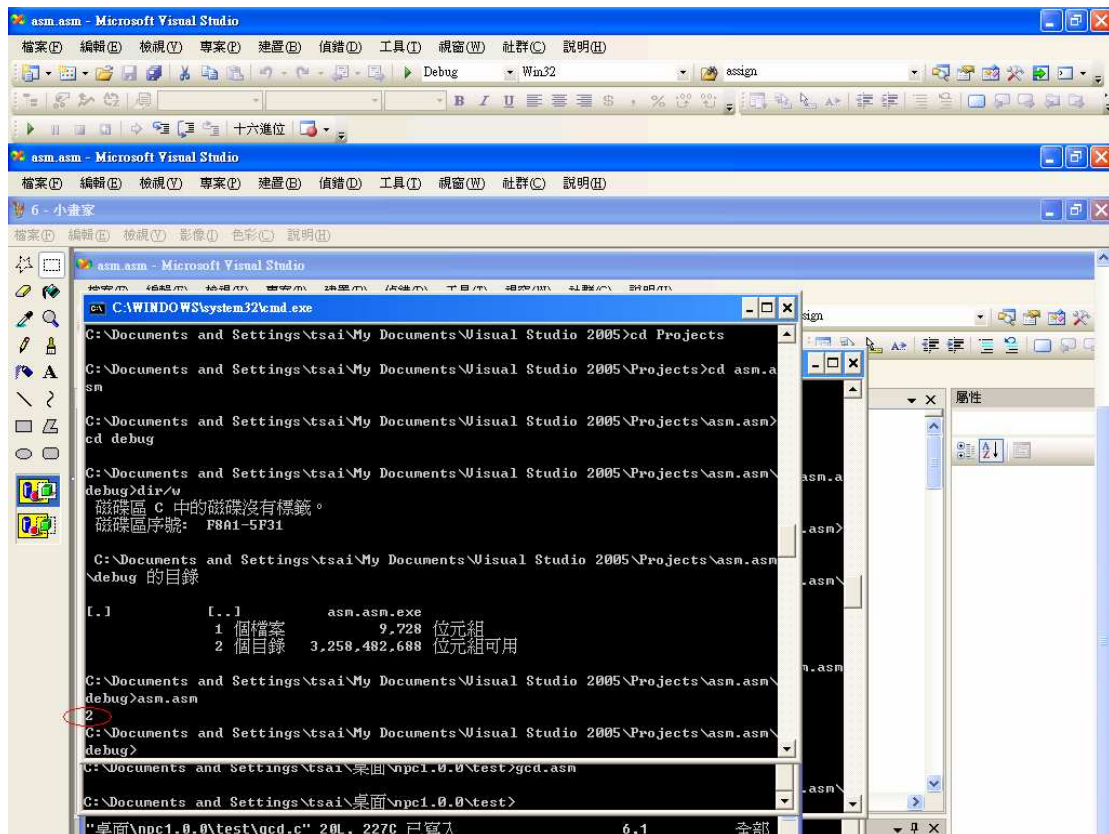
5 組譯組語



6 去專案的 debug 資料夾下,看到執行檔(我的專案名稱叫 asm.asm 故執行檔案也是此名稱)



7 執行



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd Projects
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd asm.a
sm
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd debug
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd debug
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd debug
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>dir/w
debug
磁碟區 C 中的磁碟沒有標籤。
磁碟區序號: F8A1-5F31
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>dir/w
debug
的目錄
[.]          [..]          asm.asm.exe
1 個檔案          9,728 位元組
2 個目錄          3,258,482,688 位元組可用
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd debug
debug>asm.asm
2
C:\Documents and Settings\tsai\My Documents\Visual Studio 2005>cd debug
C:\Documents and Settings\tsai\桌面\npc1.0.0\test>gcd.asm
C:\Documents and Settings\tsai\桌面\npc1.0.0\test>
"桌面\npc1.0.0\test\gcd.c" 20L, 227C 已寫入
6,1 全部
```

過了~~答案真的是 2 耶~~

NPC 1.0.0 版，規則：

詳細規範跟實做方法在 spec 資料夾。Test 裡面一些範例程式。然後因為我們知道大家一定很不喜歡看 spec。所以大概在這裡提一下語法。

型態：

僅支援 int,不支援在一開始就 initialize 唷！(no array)

Statement：

支援 for 跟 if，if 沒有 else。可以不用 block。

Expression：

支援不要太奇怪的運算，有做 expression precedence！(C99 standard 的 precedence，也就是跟我們用的一樣，不過不支援括號)

支援註解：

單行註解跟段落註解。

Function Call:

WriteDec：給定數字，寫數字

WriteChar：給定 ascii 碼，寫其對應之字元。