Final Project 心得報告

B11201033 宋承軒

Project 6 : Hack Assembler

當初在想 Final Project 的主題時有考慮用 Jack 語言寫遊戲。但因為是一個人一組，而且我對 Jack 語言並不熟悉，所以在嘗試了幾天後就放棄了。之後後我選擇以課本裡的 Project 6 來作為 Final Project 的主題，因爲是用高階語言來撰寫，所以速度上快了不少。

本次我使用 Python 來完成 Hack Assembler。因為剛開始還沒有什概念，所以在 coding 方面卡住一段時間。但在複習過上課內容，以及上網找一些資料後，就稍為順利一些。

以下是我的一些整理：

我們要將 Hack 的組合語言轉換為機械語言需要經過以下步驟：

1. 打開要轉換的 asm 檔案
2. 對檔案內容做清裡，包括空格或註解符號
3. 辨別每一行的內容為何（是 A 指令， C 指令或是 label）
4. 把每一行轉變成對應的機械語言
5. 把結果存成 hack 檔

我們做第二點（對檔案內容做清裡）的原因，是因為我們撰寫的程式碼可能如左圖，其中包含了註解以及許多空格。而我們必須將它整理成右圖的形式，才能更方便的轉換成機械語言。在這個步驟裡我們可以使用許多 Python 內建的函數，比如 strip ， split 等等。

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/06/max/Max.asm

// Computes R2 = max(R0, R1)  (R0,R1,R2 refer to RAM[0],RAM[1],RAM[2])

   @R0
   D=M              // D = first number
   @R1
   D=D-M            // D = first number - second number
   @OUTPUT_FIRST
   D;JGT            // if D>0 (first is greater) goto output_first
   @R1
   D=M              // D = second number
   @OUTPUT_D
   0;JMP            // goto output_d
(OUTPUT_FIRST)
   @R0
   D=M              // D = first number
(OUTPUT_D)
   @R2
   M=D              // M[2] = D (greatest number)
(INFINITE_LOOP)
   @INFINITE_LOOP
   0;JMP            // infinite loop
```

```
@R0
D=M
@R1
D=D-M
@OUTPUT_FIRST
D;JGT
R1
D=M
@OUTPUT_D
0;JMP
(OUTPUT_FIRST)
@R0
D=M
(OUTPUT_D)
@R2
M=D
(INFINITE_LOOP)
@INFINITE_LOOP
0;JMP
```

我們做第三點（辨別每一行的內容為何）的方法，就是一般處理文字的方法。我們可以使用 if else 做一些判斷，比如若是第一個字是“@”的話那就表示這是一條 A 指令，若第一個字是“(”的話，那這是 label，依此類推。

當我們對該行做一個分類後，我們就可以將它們翻譯成機械語言了。藉由第三點，我們會有三種不同的情況（A 指令， C 指令或是 label）。

A 指令：
當我們遇到 A 指令時，我們要先分辨“@”後面接的是什麼，若是數字那我們可以直接將數字轉換為二進制地址並添加到指令中。若是文字的話，那我們則需要判斷 symbol table 是否有該文字，假如沒有那我們還需要創建該文字對應的 address ，然後才能獲取該符號對應的 address 並添加到指令中。

C 指令：
C 指令的部分比較繁瑣，我們需要將指令做適當的分解，在得到 computation, destination, jump 的資訊後，再將它們做轉換。

Label ：
這部分我們只需要將指令轉換為標籤及 address，並將這些信息存儲在 symbol table 中。這樣在後續的組譯過程中，當遇到標籤需要轉換為 address 時，就可以從符號表中查找相應的信息。

經過以上步驟後，我們就可以把 Hack 的組合語言轉換成機械語言了。最後我們只需要將結果從成 hack 檔案即可。

總結來說，我們需要設計四個 Class (Assembler , Parser , Analysis , Symbol Table)，分別處理各項事情。這樣就可以完成 Hack Assembler。