計算機組織 Lab1: RISC-V Programming Report

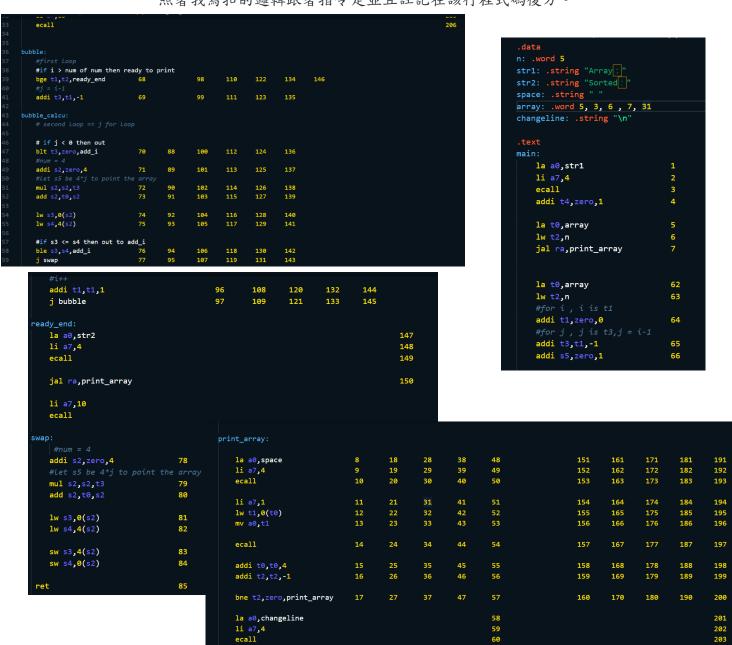
109550025 謝翔丞

1. Bubble_sort

1-1. How many instructions are actually executed?

這題因為助教很慷慨地給予我們縮減 array,所以我將長度變為 5,取原先的前五個值也就是:5,3,6,7,31。也因此從原本預估一千多 道指令縮減為 206 道。

計算指令的方法非常簡單,我根據寫的組合語言,一行一行的假裝自己是電腦跟著走,然後仿照助教給予的 count 檔案將該步數寫在對應的 code 後方(如下圖所示),從讀入 array 接著輪流印出,然後進入複雜的 sorting 再存入 array 最後印出排序好的結果,一步一步照著我寫扣的邏輯跟著指令走並且註記在該行程式碼後方。



61

204

ret

1-2. What is the maximum number of variable be pushed into the stack at the same time when yourcode execute? 這小題慷慨的助教說可以不用寫。

2. Gcd

2-1. How many instructions are actually executed?

這題經過我的計算我使用了 73 道指令。

計算指令的方法也非常簡單,我根據寫的組合語言,一行一行的 又一次假裝自己是電腦跟著走,然後仿照助教給予的 count 檔案將該 步數寫在對應的 code 後方(如下圖所示),從分別讀入兩個數字接著 輪流印出,然後進入比較程序再分別更改其值,最後印出結果,一步 一步照著我寫扣的邏輯跟著指令走並且註記在該行程式碼後方。

```
num1: .word 4
num2: .word 8
str1: .string "GCD value of "
str2: .string " and "
str3: .string " is "
                a0, num1
                a1, num2
     jal
                ra, gcd
                a0, num1
               a1, num2
                                                           48
                                                           49
     jal
               ra, print
     1i
                a7,10
     ecall
         la
li
                       a0, str1
          ecall
         m∨
1i
          ecall
                       a0, str2
                                                              60
61
          ecall
          m∨
li
         ecall
                       a0, str3
          ecall
         m∨
li
```

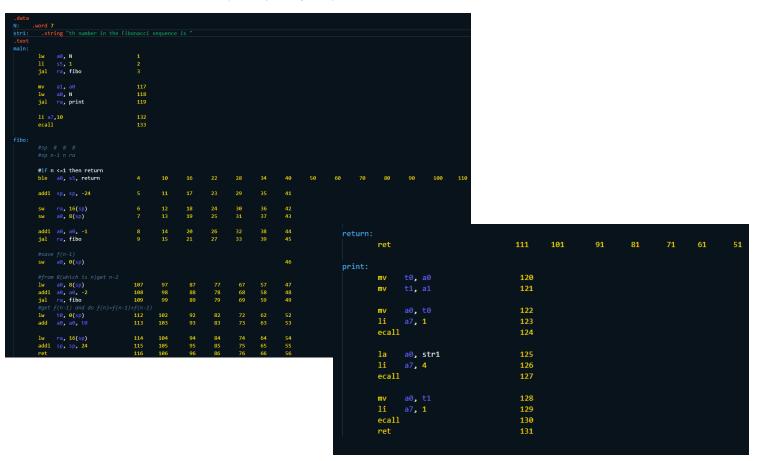
```
addi
            sp, sp, -24
                                                14 24
            ra, 16(sp)
            a1, 8(sp)
                                                16
                                                    26
    sw
            a0, 0(sp)
                                                17
                                                18 28
    addi
    # if i still \geq 0 , them keep doing calcu
                                                19
                                                   29
            t0, zero, ngcd
    addi
                sp, sp, 24
                                                    30
ngcd:
                                            10 20
            a0,a1
                                            12 22
           ra, gcd
                                            13 23
    jal
    mν
            a0, 0(sp)
    1w
                                            40 33
            a1, 8(sp)
           ra, 16(sp)
    addi
               sp, sp, 24
                                            43 36
```

2-2. What is the maximum number of variable be pushed into the stack at the same time when yourcode execute? 這題經過我的判斷和計算,需要用到3層 * 3個 (a0, a1, ra)總共9個 variable在stack裡面。

3. Fibonacci

3-1. How many instructions are actually executed? 這題經過我縝密的計算,我使用了 133 個指令。

我根據寫的組合語言,一行一行的再一次假裝自己是電腦跟著走,然後仿照助教給予的 count 檔案將該步數寫在對應的 code 後方(如下圖所示),從 n=7 開始進入 fibo 函示,然後隨著 code、開 sp空間、減少 n 等等,以及算到後面開始 return 回來,這些都沒有一絲遺漏,最後印出結果,一步一步照著我寫扣的邏輯跟著指令走並且註記在該行程式碼後方。



3-2. What is the maximum number of variable be pushed into the stack at the same time when yourcode execute? 這題經過我的判斷和計算,需要用到 7層 * 3個(a0, ra) 總共21個 variable在stack裡面,但是當n=1時只需要存一個argument 所以推測應該是21-1=20。

Experience:

這次的lab其實對我個人而言是滿大的挑戰,畢竟之前沒有接觸過組合語言,要從無到生出一份可以執行的code本身就是一件不容易的事情,而這次又有兩份的遞迴要寫更是難上加難,因為遞迴本身就不是學程式語言的基礎入門課程。

先從我寫的第一份code講起,我在讀完範例code之後第一個先從fibonacci下手。一開始我沒注意到. c檔是用遞迴的方式,因此我很直覺地用兩個暫存器直接痛快地爆加到7,然後解出來答案也對,所以就爽快地進入下一題。沒想到過幾天發現要照. c檔的方式寫遞迴,導致我原本寫好的count版本與report都要大改,真是有驚無險。

第二份則是gcd,有了第一份的經驗之後就比較容易上手,這次個人比較深刻的印象是我一開始一直找不到取餘數的指令,所以自己多寫了一個用來把兩個數扣到底的函式。雖然有效的找出餘數,但是因為我沒有使用sp,導致我ret回去的位址都十分詭異,因此雖然有印出正確答案,但是他會不斷印出來、無法停止,後來是利用stack pointer然後存取ra才成功救回我的code。並且找到rem來取代落落長又沒效率的自創函式,減少不少的指令數。

最後寫出來的是bubble_sort,我花了很多天在處理這道題目,主要是在思考怎麼存一整串array進去,後來才想到放成一整個word然後根據第幾個把他*4去找出對應的位置。

另外比較好笑的是,我在2.2.4版本上面跑bubble_sort結果跑了快一分種還沒結束我就以為我寫錯讓他死掉了,所以一直中斷並尋找我認為可能出現錯誤的地方。最後是我丟著讓他跑、先去做別的事,才發現他有結束的一天。

總結,這次除了寫code以外,最辛苦的地方是計算instruction數量,因為我的作法是當作自己是電腦、一行一行去走,有時候快算完才發現在很初期的地方有遺漏,整個就要重來十分崩潰,也幸好bubble_sort那題可以縮短array長度,德政!否則我大概處理算錯指令數量 的時間會比debug久。

這次lab下來,感覺對組語有基本的認識,也很感謝助教幾乎隨時都在線上幫我們解決問題,疑惑都能在短時間內得到很完整的答覆,辛苦了!

109550025 謝翔丞