

Arm Assembly Mid-term project

老師：朱守禮

組別：7

組員：

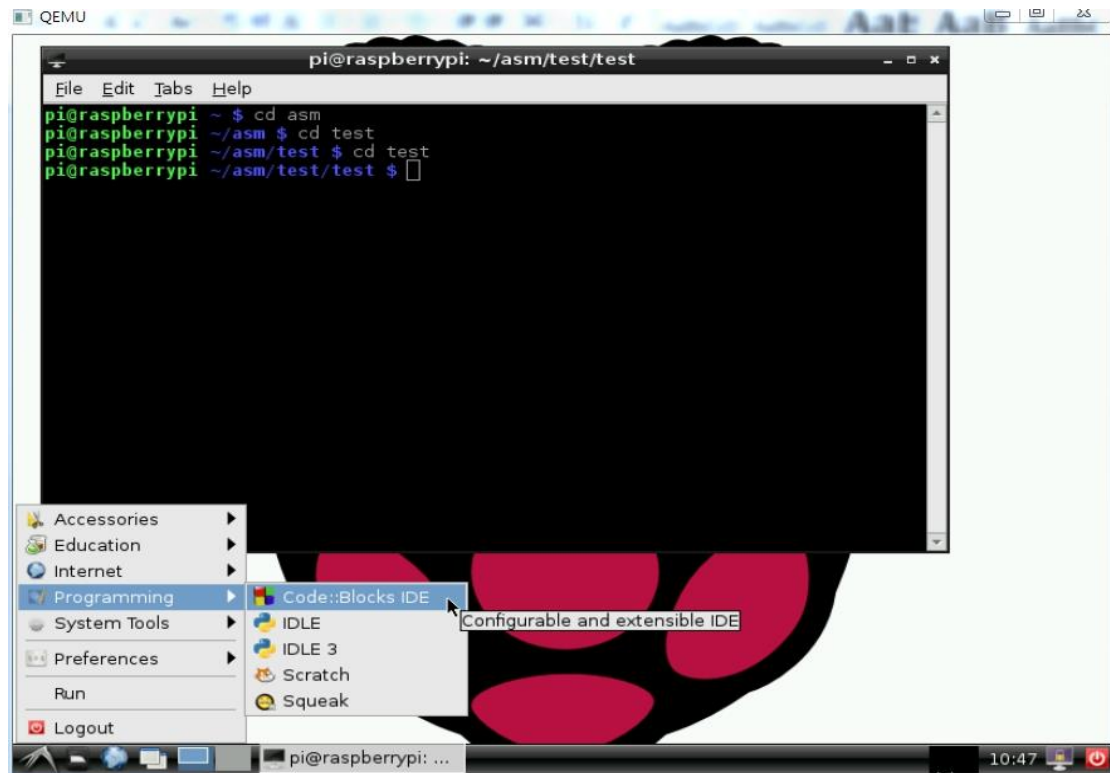
10520127 電資三 戴承浩

10520104 電資三 莊東翰

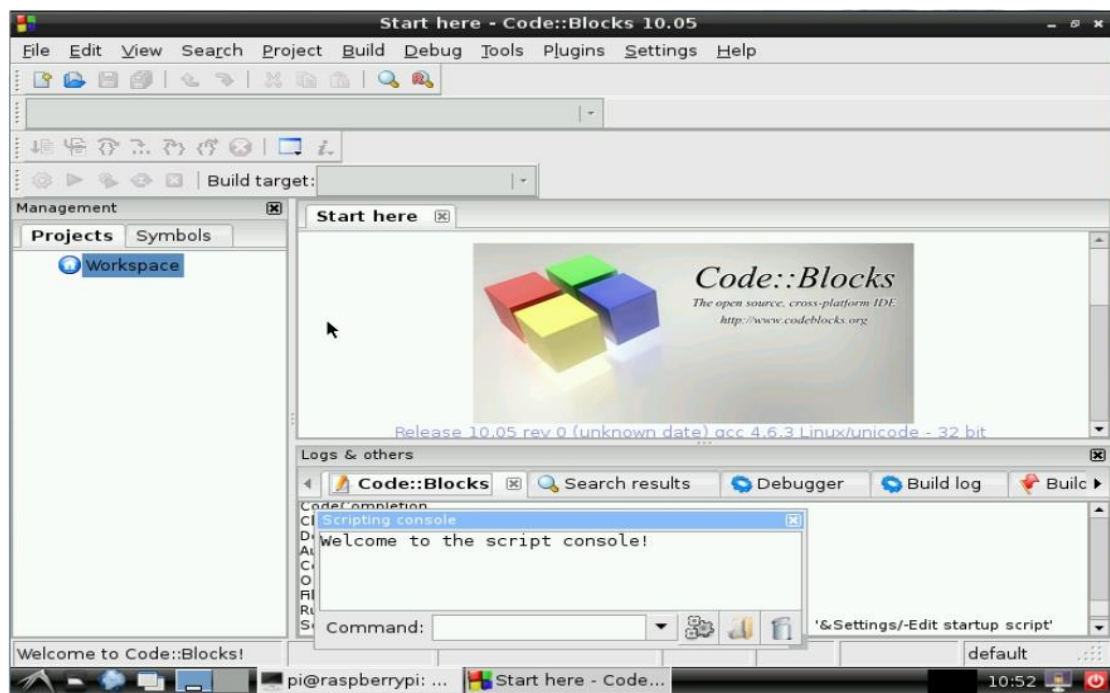
1. 背景

ARM Assembly Language

在 raspberrypi 模擬器下運行



Programing >> Code::Blocks



2. 方法

分三個檔 ID.s NAME.s main.s

先做介面需要的記憶體空間規劃 例：main ↓ NAME ↓ ID

```
.data
Str: .asciz "%s\n"
st0:  .asciz "Function1: Name\n"
st1:  .asciz "*****End Print*****\n"
st2:  .asciz "Function2: ID\n**** *Input ID*****\n"
st3:  .asciz "*****Print Team Member ID and ID Summation*****\n"
st4:  .asciz "\nMain Function:\n*****Print All*****\n"

All:  .asciz "%d\t%s\n"
pformat:.asciz "ID Summation = %d\n"  @ Format string for printf
      .align 2

.text

.globl main

main:
```

```
.data

msg1:  .asciz "Team07\n"
msg2:  .asciz "ZHUANG DONG HAN\n"
msg3:  .asciz "DAI CHENG HAO\n"
msg4:  .asciz "ZHUANG DONG HAN\n"

.text

.globl NAME

NAME:
```

```

.data
str1:      .asciz  "*****Print Team Member ID and ID Summation*****\n"
st5:      .asciz  "***  Command Error  ***\n"
prompt1:   .asciz  "Enter the first sid:\n" @ user prompt1
          .align  2

prompt2:   .asciz  "Enter the second sid:\n" @ user prompt2
          .align  2

prompt3:   .asciz  "Enter the third sid:\n" @ user prompt3
          .align  2

prompt4:   .asciz  "Please enter the command:\n" @ user prompt4
          .align  2

prompt5:   .asciz  "Command error.\n" @ user prompt5
          .align  2

letterp:   .asciz  "p"
          .align  2

printint:  .asciz  "%d\n" @ Format string for reading an int with scanf
          .align  2

sformat:   .asciz  "%d" @ Format string for reading an int with scanf
          .align  2

pformat:   .asciz  "The sum is: %d\n" @ Format string for printf
          .align  2

stringformat: .asciz "%s"
          .align  2

n:         .word   0 @ int n = 0
m:         .word   0 @ int m = 0
a:         .word   0 @ int a = 0
b:         .word   0 @ int b = 0 */sum*/
scanword:  .asciz  "%s"
.text

.globl ID

```

各別的函數內部運作

NAME 函數功能：

說明：先將字串存在記憶體內利用 MOV r7, #4 來輸出每一筆字串

```

pi@raspberrypi ~/asm/test/test $ ./main.out
Function1: Name
Team07
ZHUANG DONG HAN
DAI CHENG HAO
ZHUANG DONG HAN
*****End Print*****

```

程式設計：在 MOV r7, #4 這道指令後面要加上 SWI 0 來停止 print

且在程式的最後重新讓 r0-r3 存取 msg1-4

```

store:
    LDR r0, =msg1
    LDR r1, =msg2
    LDR r2, =msg3
    LDR r3, =msg4

```

ID 函數功能：

說明：先輸入三筆學號資料，並讀入一個 command 只有輸入 command 為 p

的時候才要輸出三筆學號與一個所有學號相加後的結果。

```

Function2: ID
**** *Input ID****
Enter the first sid:
10520104
Enter the second sid:
10520127
Enter the third sid:
10520104
Please enter the command:
p
****Print Team Member ID and ID Summation****
10520104
10520127
10520104
The sum is: 31560335
****End Print****

```

程式設計：

先利用 scanf 讀入並改變記憶體內的三個變數，再將變數相加完的結果 str 寫入

第四個變數。

然後根據讀入的 command 是否為 p 去決定是否輸出變數結果。

Main 函數功能：

說明：呼叫 ID、NAME 兩函數功能，並用暫存器存取資料

b1 NAME	b1 ID
MOV r4, r0	MOV r8, r0
MOV r5, r1	MOV r9, r1
MOV r6, r2	MOV r10, r2
MOV r7, r3	MOV r11, r3

完成兩函數功能後，利用 r4-r11 輸出資料合併過後的結果

```

Main Function:
*****Print All*****
Team07
10520104      ZHUANG DONG HAN
10520127      DAI CHENG HAO
10520104      ZHUANG DONG HAN
ID Summation = 31560335
*****End Print*****
pi@raspberrypi ~/asm/test/test $

```

設計重點：在任何時候確保 r0-r3 的資料有被其他站存器存取，因為要利用 r0-r3

做任何事方便程式運作(進 function 也會利用 r0-r3，bl printf 也需要用到

r0-r4)

3. 結果

根據是否為 'p' 會有兩種不同的結果

以下為若不是 'p' 的情況

```

pi@raspberrypi ~/asm/test/test $ ./main.out
Function1: Name
Team07
ZHUANG DONG HAN
DAI CHENG HAO
ZHUANG DONG HAN
*****End Print*****
Function2: ID
**** *Input ID*****
Enter the first sid:
10520104
Enter the second sid:
10520127
Enter the third sid:
10520104
Please enter the command:
k
*** Command Error ***
*****End Print*****

Main Function:
*****Print All*****
Team07
10520104      ZHUANG DONG HAN

10520127      DAI CHENG HAO

10520104      ZHUANG DONG HAN

ID Summation = 31560335
*****End Print*****

```

以下為正常讀入全輸出的情況

```

pi@raspberrypi ~/asm/test/test $ ./main.out
Function1: Name
Team07
ZHUANG DONG HAN
DAI CHENG HAO
ZHUANG DONG HAN
*****End Print*****
Function2: ID
**** *Input ID*****
Enter the first sid:
10520104
Enter the second sid:
10520127
Enter the third sid:
10520104
Please enter the command:
p
*****Print Team Member ID and ID Summation*****
10520104
10520127
10520104
The sum is: 31560335
*****End Print*****

Main Function:
*****Print All*****
Team07
10520104      ZHUANG DONG HAN

10520127      DAI CHENG HAO

10520104      ZHUANG DONG HAN

ID Summation = 31560335
*****End Print*****

```

還有考慮到若不是單純的一個字元' p' 所遇到的情況，輸出仍然正確

範例讀入為" pp"

```
pi@raspberrypi ~/asm/test/test $ ./main.out
Function1: Name
Team07
ZHUANG DONG HAN
DAI CHENG HAO
ZHUANG DONG HAN
*****End Print*****
Function2: ID
***** *Input ID*****
Enter the first sid:
10520104
Enter the second sid:
10520127
Enter the third sid:
10520104
Please enter the command:
pp
*** Command Error ***
*****End Print*****

Main Function:
*****Print All*****
Team07
10520104          ZHUANG DONG HAN

10520127          DAI CHENG HAO

10520104          ZHUANG DONG HAN

ID Summation = 31560335
*****End Print*****
```

4. 討論

暫存器不能隨使用會常常有亂掉的問題，若要使用過多的暫存器除 r0-r3 外的需要再程式內先 stmfd 內先存否則會遇到 segment fault。再寫條件式的時候要確定當下 CPSR 的狀況，還有 branch Instructions 要確定前後的位置避免無限迴圈。

5. 結論

在規劃程式撰寫的時候，要先規劃暫存器的使用與 function 之間的配合，還有

要記錄 CPSR 的狀況在程式 run 到不同地方的時候狀態是什麼樣的，並釐清 address 跟 value 的關係與使用。在讀入、輸出或判斷的時候要確定使用的是什麼樣的型別，多少個 bit。

6. 未來展望

《 心得 》

10520127 電資三 戴承浩

題目本身對現在得我們來說並不難，很能感受到實做這次作業的目的是為了先讓我們了解 ARM 與組合語言的結構，跟以往的 C 與 JAVA 有著實的不同，但是顯然光是理解語言的格式、結構、運作就夠讓我忙了，到現在還沒很能搞清楚暫存器是一個怎麼樣的東西與怎麼靈活的使用裡面所有的位元，什麼時候用 byte 什麼時候用 word。但確實對這樣的東西有著很大的認識，比起平常學習到的知識、實驗跟實做可以更有效的了解其中的構造。

10520104 電資三 莊東翰

經過這次的期中作業我學習到了 ARM 組合語言的文法用法，在有些指令中某些暫存器有特殊的功能，像是 printf 指令，如果要印出輸入的字串或數字，以 r0 紀錄格式定義的位置，以 r1 紀錄值。若只是要印出已定義的字串，直接用 r0 紀錄位置就可以印出。令我印象深刻的還有條件執行的分支指令，C 語言中 if 與 while 有專屬的指令，但在組合語言中都可以利用分支指令控制是否重複執行。經過這次的作業再次讓我體驗到用高階語言的方便性，想在組合語言中寫出

相同功能的程式就要下更多功夫，必須對電腦內部處理的方式更佳了解才行。

組員分工：

10520104 電資三 莊東翰

程式撰寫(主要 ID.s 和 NAME.s)與 Debug

資料查尋

討論

10520104 電資三 戴承浩

程式撰寫(主要 main.s) 與 Debug

資料查尋

討論

書面資料