

Homework #7 (1)

- Write a function called **NumSort** to sort an integer array from the smallest to the biggest.
- Two arguments will be passed into your function by the **APCS rule**
 - **Array size**
 - **The address of the first element in array**
- **The result of NumSort**
 - A result array in which each element is sorted from the smallest to the biggest. (原來的integer array沒有被修改，只是讀取原integer array，排序好的結果存放於result array)
 - Register **r0** will have the address of the result array.

Homework #7 (2)

- Homework #6 :
 - hw6_test.s => 設定欲搜尋的資料, 呼叫ARM組合語言寫成的NumSort function。
 - numsort.s => ARM組合語言寫成的NumSort function。
 - 透過**stack**與自訂的規則來傳遞資料。
- Homework #7 :
 - call_numsort.c => 以**C語言撰寫**, 設定欲排序的資料, 呼叫**ARM組合語言寫成的NumSort function**。
 - numsort.s => ARM組合語言寫成的NumSort function。
 - 透過**APCS**規範來傳遞資料。

Homework #7 (3)

- **call_numsort.c**：以C語言撰寫，設定欲排序的資料，呼叫ARM組合語言寫成的**NumSort** function。

```
extern int* NumSort(int, int*);

int main(void)
{
    int* result;

    /* initial a integer array */

    /* call NumSort function */
    result = NumSort(array_size, array_address);

    /* print out integer array */

    return 0;
}
```

輸出部分，請使用**printf**印出所排序好的資料，每個**integer**由一個空格隔開

Homework #7 (4)

- `call_numsort.c`最後的輸出部分請由**`printf`**印出排序好的**`integer`**，每個**`integer`**由空格隔開。
- Ex: **4 8 10 11**
- 在執行insight debugger時，請先打開**GDB console**，就可以看到輸出的結果。

Homework #7 (5)

- 若在homework#6裡，存放搜尋結果的記憶體區域是在 hw6_test.s 裡宣告，則在homework#7裡，需要移到numsort.s的檔案宣告。(如下頁所示)

若sorting時有用到callee saved registers，需要事先存起來。(這部分請自己完成)

```
.section .text
.global sort
.type sort,%function

sort:
```

```
mov    ip, sp
stmfd  sp!, {fp, ip, lr, pc}
sub    fp, ip, #4
```

```
/* r0 <= the size of array */
/* r1 <= the address of array */
```

```
/* do sorting */
```

```
...
```

```
/* r0 <= the address of result array */
```

```
ldmea  fp, {fp, sp, pc}
```

```
.section .data
.type result,%object
.size result,400

result:
.space 400
```

假設result array最大可以存放100個integer。

How to Compile Your Program?

- `%arm-elf-gcc -g call_numsort.c numsort.s -o \call_numsort.exe`

Homework #7 (6)

- Program should be assembled and linked by gcc (ARM-ELF format)
 - 使用於作業一所編譯完成的cross compiler與cross binutils
- Program should be executed under **GDB ARM simulator**
- 程式中應有適當的說明（註解）
- You should turn in to **ECOURSE**
 - “**README**” file: 文字檔，描述你程式的內容、如何編譯程式、如何執行你的程式
 - Your ARM assembly procedure，檔名為：**numsort.s**
 - A C program which uses your sorting procedure to demo your sorting algorithm，檔名為：**call_numsort.c**
 - Any file needed in your work (ex: Makefile)
- **Deadline: December 13 (Sunday), 2015**