

Homework #5 (1)

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

Homework #5 (2)

- Ex: an integer array=[1,10,6,3,20,40,9]
 - Result: **1, 3, 6, 9, 10, 20, 40**
- Ex: an integer array=[12,4,2,45,23,8,50,67]
 - Result: **2, 4, 8, 12, 23, 45, 50, 67**

參數傳遞 (APCS compliance)

- Array size (r0)
- Array address (r1)

hw5_test.s

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

main:

```
MOV ip, sp  
STMFD sp!, {fp, ip, lr, pc}  
SUB fp, ip, #4
```

```
...  
bl NumSort  
...
```

```
LDMEA fp, {fp, sp, pc}
```

numsort.s

NumSort

傳回值 (APCS compliance)

- Result array's address (r0)

hw5_test.s

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

main:

```
MOV ip, sp
STMFD sp!, {fp, ip, lr, pc}
SUB fp, ip, #4
```

```
/* prepare input array */
```

```
...
```

```
/* put array size into r0 */
/* put array address into r1 */
```

```
bl NumSort
/* --- end of your function --- */
```

```
LDMEA fp, {fp, sp, pc}
```

```
.end
```

- array size => r0
- array address => r1

Call **printf()** to print the result array.

Homework #5 (3)

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

numsort.s

NumSort:

/ function start */*

STMFD sp!, {r0-r9, fp, ip, lr}

/ --- begin your function --- */*

/ Get array size from r0 */*

/ Get array address from r1 */*

參數傳遞

/ DO NumSort */*

Write your function

/ put result array's address into r0 */*

/ --- end of your function --- */*

/ function exit */*

LDMFD sp!, {r0-r9, fp, ip, pc}

.end

傳回值

Homework #5 (4)

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

numsort.s

NumSort:

```
/* function start */
```

```
STMFD sp!, {r0-r9, fp, ip, lr}
```

```
/* --- begin your function --- */
```

```
/* Get array size from r0 */
```

```
/* Get array address from r1 */
```

```
/* DO NumSort */
```

```
/* put result array's address into r0 */
```

```
/* --- end of your function --- */
```

```
/* function exit */
```

```
LDMFD sp!, {r0-r9, fp, ip, pc}
```

```
.end
```

- ~~Array size最大為100個 elements~~
- 呼叫**malloc()**，取得**result array**的記憶體空間與位址

How to Compile Your Program?

- `$arm-none-eabi-gcc -g hw5_test.s numsort.s -o hw5.exe`

Output

- Ex: an integer array=[1,10,6,3,20,40,9]

輸出畫面範例

Input array: 1, 10, 6, 3, 20, 40, 9

Result array: **1, 3, 6, 9, 10, 20, 40**

Homework #5 (5)

- Program should be assembled and linked by gcc
 - 使用於作業一所安裝完成的cross toolchain.
- 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 **ECOURSE2**
 - “**README.txt**” 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**
 - An ARM assembly program which uses your NumSort procedure，檔名為：**hw5_test.s**
 - Makefile / any file needed in your work
 - 請將欲繳交的檔案壓縮成 <**hw5_學號.tar.bz2**>，上傳壓縮檔
- **Deadline: December 10 (Friday), 2021**