

嵌入式系統實驗, Fall 2016

Lab10 20161203

指導教授：王勝德 教授 B03901023 電機三 許秉鈞

實驗目的

認識assembly並學會使用gdb逐步操作register，撰寫.s檔並用gdb執行、學習設立breakpoint、學習一行一行瞭解組語的資料存儲形式。

請實驗完成後，12/9上課前，以個人為單位，寫實驗報告，內容包括gdb 使用，ARM Assembly programming, GNU ARM In-line Assembly, memcpy32 手寫版與compiler 產生版比較等心得或討論。

問題討論

How is ARM/EABI different to ARM/OABI?

兩種ABI在如下方面有區別：

調用規則（包括參數如何傳遞及如何獲得返回值）

系統調用的數目以及應用程序應該如何做系統調用，目標文件的二進制格式，程序庫等structure中的填充（padding/packing）和對齊。

注意：對於同一套系統，比如uboot，kernel，rootfs等等，編譯器最好是用同一版本的，要是oabi要不就然是eabi，否則不同版本的不同的東西混用，很容易出問題。

心得比較

我發現手寫板自己刻出來的好像會比compiler產生的慢一些、然後記憶體卻似乎比較節省，可能是因為compiler會優化所以變快了、但compiler的會加一些additional 存儲空間，所以就記憶體用變多了（也可能是誤差、其實結果差不多）。但基本上看起來幾乎一樣。

實驗紀錄

```
(gdb) b use_gdb.s:1
Breakpoint 1 at 0x10058: file use_gdb.s, line 1.
(gdb) r
Starting program: /home/pi/tmp/use_gdb
```

```
Breakpoint 1, _start () at use_gdb.s:7
```

```
7      cmp r1, #4
```

```
(gdb) info registers
```

| | | | |
|------|------------|----|--------------------|
| r0 | 0x0 | 0 | |
| r1 | 0x5 | 5 | |
| r2 | 0x0 | 0 | |
| r3 | 0x0 | 0 | |
| r4 | 0x0 | 0 | |
| r5 | 0x0 | 0 | |
| r6 | 0x0 | 0 | |
| r7 | 0x0 | 0 | |
| r8 | 0x0 | 0 | |
| r9 | 0x0 | 0 | |
| r10 | 0x0 | 0 | |
| r11 | 0x0 | 0 | |
| r12 | 0x0 | 0 | |
| sp | 0x7efff690 | | 0x7efff690 |
| lr | 0x0 | 0 | |
| pc | 0x10058 | | 0x10058 <_start+4> |
| cpsr | 0x10 | 16 | |

```
(gdb) n
```

```
8      sub r1, r1, #1
```

```
(gdb) info registers
```

| | | |
|----|-----|---|
| r0 | 0x0 | 0 |
| r1 | 0x5 | 5 |
| r2 | 0x0 | 0 |
| r3 | 0x0 | 0 |
| r4 | 0x0 | 0 |

```

r5          0x0  0
r6          0x0  0
r7          0x0  0
r8          0x0  0
r9          0x0  0
r10         0x0  0
r11         0x0  0
r12         0x0  0
sp          0x7efff690      0x7efff690
lr          0x0  0
pc          0x1005c      0x1005c <_start+8>
cpsr        0x20000010      536870928
(gdb) n
9      cmp r1, #4
(gdb) info registers
r0          0x0  0
r1          0x4  4
r2          0x0  0
r3          0x0  0
r4          0x0  0
r5          0x0  0
r6          0x0  0
r7          0x0  0
r8          0x0  0
r9          0x0  0
r10         0x0  0
r11         0x0  0
r12         0x0  0
sp          0x7efff690      0x7efff690
lr          0x0  0
pc          0x10060      0x10060 <_start+12>
cpsr        0x20000010      536870928
(gdb) n
10     sub r1, r1, #1
(gdb) info registers
r0          0x0  0
r1          0x4  4
r2          0x0  0
r3          0x0  0
r4          0x0  0
r5          0x0  0
r6          0x0  0
r7          0x0  0
r8          0x0  0
r9          0x0  0
r10         0x0  0
r11         0x0  0
r12         0x0  0
sp          0x7efff690      0x7efff690
lr          0x0  0
pc          0x10064      0x10064 <_start+16>
cpsr        0x60000010      1610612752

```

```

(gdb) n
11    cmp r1, #4
(gdb) info registers
r0          0x0    0
r1          0x3    3
r2          0x0    0
r3          0x0    0
r4          0x0    0
r5          0x0    0
r6          0x0    0
r7          0x0    0
r8          0x0    0
r9          0x0    0
r10         0x0    0
r11         0x0    0
r12         0x0    0
sp          0x7efff690    0x7efff690
lr          0x0    0
pc          0x10068    0x10068 <_start+20>
cpsr       0x60000010    1610612752
(gdb) n
13    mov r0, r1
(gdb) info registers
r0          0x0    0
r1          0x3    3
r2          0x0    0
r3          0x0    0
r4          0x0    0
r5          0x0    0
r6          0x0    0
r7          0x0    0
r8          0x0    0
r9          0x0    0
r10         0x0    0
r11         0x0    0
r12         0x0    0
sp          0x7efff690    0x7efff690
lr          0x0    0
pc          0x1006c    0x1006c <_start+24>
cpsr       0x80000010    -2147483632
(gdb) n
14    mov r7, #1
(gdb) info registers
r0          0x3    3
r1          0x3    3
r2          0x0    0
r3          0x0    0
r4          0x0    0
r5          0x0    0
r6          0x0    0
r7          0x0    0
r8          0x0    0

```

```

r9          0x0  0
r10         0x0  0
r11         0x0  0
r12         0x0  0
sp          0x7efff690      0x7efff690
lr          0x0  0
pc          0x10070      0x10070 <_start+28>
cpsr       0x80000010      -2147483632
(gdb) n
15      svc #0
(gdb) info registers
r0          0x3  3
r1          0x3  3
r2          0x0  0
r3          0x0  0
r4          0x0  0
r5          0x0  0
r6          0x0  0
r7          0x1  1
r8          0x0  0
r9          0x0  0
r10         0x0  0
r11         0x0  0
r12         0x0  0
sp          0x7efff690      0x7efff690
lr          0x0  0
pc          0x10074      0x10074 <_start+32>
cpsr       0x80000010      -2147483632
(gdb) n
[Inferior 1 (process 1231) exited with code 03]

```

實驗步驟

利用-g or -gstabs 來試著assemble，並用gdb來run。

```
(target) as -gstabs use_gdb.s
```

```
(target) ld -gstabs use_gdb.s -o use_gdb
```

```
(target) gdb use_gdb
```