Example: Matrix Multiplication

Example (1)

```
DATA section
   _____ */
      .data
/* --- variable a --- */
      .type a, %object
a:
      .word 1
      .word 2
/* --- variable b --- */
      .type b, %object
b:
      .word 3
/* --- variable c --- */
      .type c, %object
C:
      .space 8, 0
```

$$a = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

$$b = \begin{bmatrix} 3 \end{bmatrix}$$

$$c = b \times a$$

$$c = \begin{bmatrix} ? & ? \end{bmatrix}$$

Example (1)

```
DATA section
/* ======== */
     data
/* --- variable a --- */
a:
     .word 1
     .word 2
/* --- variable b --- */
     .word 3
/* --- variable b --- */
     .space 8, 0
```

$$a = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

$$b = \begin{bmatrix} 3 \end{bmatrix}$$

$$c = b \times a$$

$$c = \begin{bmatrix} ? & ? \end{bmatrix}$$

```
/* ======= */
/* DATA section */
/* ======= */
     .data
/* --- variable a --- */
a:
     .word 1
     .word 2
/* --- variable b --- */
b:
     .word 3
/* --- variable b --- */
C:
      .space 8, 0
```

```
/* ======= */
/* TEXT section
/* ======== */
      .section .text
      .global main
      .type main, %function
.matrix:
      .word a
      .word b
     .word c
main:
     ldr r0, .matrix
     ldr r1, [r0], #4
```

Linker幫忙把位址填好

```
/*
                                     TEXT section
       DATA section
                                   .section .text
      .data
                                   .global main
                                   .type main, %function
      variable a --- */
                             .matrix:
                                   .word a
      .word 1
                                   word b
      .word 2
                                   .word c
                             main:
   -- variable b --- */
                                   ldr r0, .matrix
b:
                                   ldr r1, [r0], #4
      .word 3
    -- variable b --- */
C:
      .space 8,
```

```
/* ======= */
/* DATA section */
/* ======= */
      .data
/* --- variable a --- */
0 \times 100
      .word 1
      .word 2
/* --- variable b --- */
0x108
      .word 3
/* --- variable b --- */
0x10c
      .space 8, 0
```

```
/* ======= */
/* TEXT section
/* ======== */
      .section .text
      .qlobal main
      .type main, %function
.matrix:
      0 \times 100 /* .word */
      0x108 /* .word */
      0x10c /* .word */
main:
     ldr r0, .matrix
      ldr r1, [r0], #4
```

```
/*
       TEXT section
                                          Example (2)
/* ======== */
      .section .text
      .qlobal main
      .type main, %function
.matrix:
      .word a
      .word b
                            抓到a的address,放入r0
      .word c
main:
                                          抓到第一個a的值
      ldr r0, .matrix
      ldr r1, [r0], #4 /* r1 := mem32[r0]
                       /* r0 := r0 + 4
      ldr r2, [r0] —
                                       ▶抓到第二個a的值
      1dr r0, .matrix + 4
      1dr r3, [r0] /* r3 := mem32[r0] */
      ldr r4, .matrix + 8
      mul r5, r3, r1
      mul r6, r3, r2
      str r5, [r4], #4 /* mem32[r4] := r5 */
                        /* r4 := r4 + 4
      str r6, [r4]
       nop
```

```
/* ======= */
/*
                           */
        TEXT section
                                             Example (2)
       .section .text
       .qlobal main
       .type main, %function
.matrix:
       .word a
       .word b
       .word c
                                     抓到b的address,放入r0
main:
       ldr r0, .matrix
                       /* r1 := mem32[r0] */
       ldr r1, [r0], #4
                        /* r0 : \neq r0 + 4
       ldr r2, [r0]
      ldr r0, .matrix + 4
      ldr r3, [r0] /* r3 := mem32[r0] */
                                 → 抓到c的address,放入r4
      ldr r4, .matrix + 8 —
      mul r5, r3, r1
      mul r6, r3, r2
       str r5, [r4], #4 /* mem32[r4] := r5 */
                          /* r4 := r4 + 4
       str r6, [r4]
       nop
```

```
_____ */
       TEXT section
                                   Example (2)
      .section .text
      .qlobal main
      .type main, %function
.matrix:
      .word a
      .word b
      .word c
main:
      ldr r0, .matrix
      ldr r1, [r0], #4 /* r1 := mem32[r0] */
                       /* r0 := r0 + 4
      ldr r2, [r0]
      1dr r0, .matrix + 4
      ldr r3, [r0] /* r3 := mem32[r0] */
                                將結果存入第一個c的位置
      ldr r4, .matrix + 8
      mul r5, r3, r1
                                           字入第二個c的位置
      mul r6, r3, r2
      str r5, [r4], #4 /* mem32[r4] := r5 */
                        /* r4 := r4 + 4
      str r6, [r4].
       nop
```