

9.

9.1

- (.text)
- (.rodata) const static
- (.data)
- (.bss) 0
- (.comment)

- bss bin/elf
-

Git NoosProgramProject/(9_ /001_segment)

9.1.1 1

main.c

```
05 char g_charA = 'A';            //    .data
06 const char g_charB = 'B';      //    .rodata
07 const char g_charC;    //    .bss
08 int g_intA = 0;    //    .bss
09 int g_intB;    //    .bss
```

9.1.2 2

9-1.2

imx6ull.lds

```

SECTIONS {
    . = 0x80100000;

    . = ALIGN( 4);
    .text      :
    {
        *(.text)
    }

    . = ALIGN( 4);
    .rodata : { *(.rodata) }

    . = ALIGN( 4);
    .data : { *(.data) }

    . = ALIGN( 4);
    __bss_start = .;
    .bss : { *(.bss) *(.COMMON) }
    __bss_end = .;
}

```

9.1.3 3 Makefile imx6ull.lds

```

#
$(LD) -T imx6ull.lds -g start.o uart.o main.o my_printf.o -o relocate.elf -lgcc -
L/home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-
gnueabi/lib/gcc/arm-linux-gnueabi/6.2.1

```

9.1.4 4 4-1.4-1.4.4

- 0x80100000
 - Disassembly of section ...
 - bss bss elf/bin
- relocate.dis

relocate.elf: file format elf32-littlearm

Disassembly of section .text: //

80100000 <_start>:

80100000: e59fd028 ldr sp, [pc, #40] 80100030 <clean+0x14>

80100004: eb000001 bl 80100010 <clean_bss>

80100008: fb000070 blx 801001d2 <main>

.....()

Disassembly of section .rodata: //

8010086c <g_charB>:

8010086c: 00000042 andeq r0, r2, asr #32

.....()

Disassembly of section .data: //

8010098c <g_charA>:

8010098c: 00000041 andeq r0, r1, asr #32

80100990 <hex_tab>:

80100990: 33323130 teqcc r2, #48, 2

80100994: 37363534 [] <UNDEFINED> instruction: 0x37363534

80100998: 62613938 sbvs r3, r1, #56, 18 0xe0000

8010099c: 66656463 strbtvs r6, [r5], -r3, ror #8

Disassembly of section .bss: //bss .bin

801009a0 <__bss_start>:

801009a0: 00000000 andeq r0, r0, r0

801009a4 <IOMUXC_SW_MUX_CTL_PAD_UART1_RX_DATA>:

801009a4: 00000000 andeq r0, r0, r0

801009a8 <g_intA>:

```

801009a8: 00000000 andeq r0, r0, r0

801009ac <g_intB>:
801009ac: 00000000 andeq r0, r0, r0

801009b0 <g_charC>:
. .

.....( )

Disassembly of section .comment: /comment .bin

.....( )

```

9.2

```
** Git NoosProgramProject/(9_ /02_clean_bss)**
```

```
Makefile *-T filename.lds*
```

```

# imx6ull.lds
$(LD) -T imx6ull.lds -g start.o uart.o main.o my_printf.o -o relocate.elf -lgcc -
L/home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-
gnueabi/lib/gcc/arm-linux-gnueabi/6.2.1

```

```

# .text 0x80100000
# .data 0x80102000
$(LD) -Ttext 0x80100000 -Tdata 0x80102000 -g start.o uart.o main.o my_printf.o -o relocate.elf
-lgcc -L/home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-
gnueabi/lib/gcc/arm-linux-gnueabi/6.2.1

```

9.2.1

http://ftp.gnu.org/old-gnu/Manuals/ld-2.9.1/html_mono/ld.html

```
SECTIONS {
...
secname start BLOCK(align) (NOLOAD) : AT ( ldadr )
    { contents } >region :phdr =fill
...
}
```

- secname
- start runtime addr relocation addr
- AT (ldadr) ldadr load addr AT ldadr
-
- { contents } { } content
- BLOCK(align) (NOLOAD) >region :phdr =fill

1.1.2 imx6ull.lds

9.2.2

imx6ull.lds

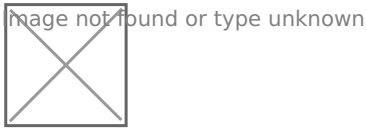
```
01 SECTIONS {
02     . = 0x80100000; [ ]//      0x80100000
03
04     . = ALIGN(4); [ ]//      4
05     .text      : [ ]//      .text
06     { [ ]//.text
07         *(.text) [ ]// *
08     }
09
10     . = ALIGN(4); [ ]//      4
11     .rodata : { *(.rodata) } [ ]//.rodata .text ,
12
13     . = ALIGN(4);
14     .data : { *(.data) } [ ]//.data .rodata
15
```

```

16     . = ALIGN( 4);
17     __bss_start = .; [REDACTED] __bss_start
18     .bss : { *(.bss) *(.COMMON) } [REDACTED] .bss .data ,          bss
19     __bss_end = .; [REDACTED] __bss_end
20 }

```

.bin



(.text) (.data)

bss

bss

```

SECTIONS {
    . = 0x80100000; [0000]// 0x80100000 .text

    . = ALIGN( 4);
    .text :
    {
        *(.text)
    }

    . = ALIGN( 4);
    .rodata : { *(.rodata) } [00]// rodata 0x8010xxxx

    . = ALIGN( 4);
    .data 0x80800000 : { *(.data) } [00]// data 0x80200000, rodata

    *****

```

1. flash

Flash

2. JTAG

9.2.3 bss

bin bss 0 bin

 bss CPU bss bss 0 bss

 bss 002_clean_bss

9.2.3.1 1

 bss bss

start.S

```
01
02 .text
03 .global _start
04
05 _start:
06
07     /*     */
08     ldr sp, =0x80200000
09
10     /*  bss  */
11     bl clean_bss
12
13     /*     */
14     bl main
15
16 halt:
17     b halt
18
19 clean_bss:
20     ldr r1, =__bss_start//      __bss_start    r1
21     ldr r2, =__bss_end//      __bss_end    r2
22     mov r3, #0
23 clean:
24     strb r3, [r1]//
25     add r1, r1, #1// r1    +1
26     cmp r1, r2//      clean    bss
27     bne clean
28
29     mov pc, lr
```

9.2.3.2 2

bss

main.c

```
37 int main (void)
38 {
39     Uart_Init();    //  uart
40
41     printf("g_intA = 0x%08x\n\r", g_intA);  //  g_intA
42     printf("g_intB = 0x%08x\n\r", g_intB);  //  g_intB
43
44     return 0;
45 }
46
```

9.2.3.3 3 4-1.4

9.2.3.4 4 3-1.4

bss g_intA, g_intB 0 bss

```
g_intA = 0x00000000
g_intB = 0x00000000
```

9.3

9.3.1

S3C2440	.bin	Nor Flash	SRAM	SDRAM	imx6ull	I
Boot Rom		3-1.2 IMX6ULL			100ask_imx6ull	

```
./tools/mkimage -n ./tools/imximage.cfg.cfgtmp -T imximage -e **0x80100000** -d relocate.bin
relocate.imx
```

DDR3 .bin 0x80100000 CPU

9.3.2 data

	data	data			
002_clean_bss		data	g_charA	** Git	NoosProgramProject/(9_
main.c					

```
37 int main (void)
38 {
39     Uart_Init();    //  uart
40
41     printf("\n\r");
42     /*      g_charA */
43     while (1)
44     {
45         PutChar( g_charA);
46         g_charA++;
47         delay(1000000);
48     }
49
50     return 0;
51 }
```

9.3.2.1 1 4-1.4

9.3.2.2 2 3-1.4

g_charA

ABCDEF GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{| }

123456789; ; <=>?@A

CPU	DDR3	g_charA	DDR3		
data	imx6ull	RAM	CPU	RAM	DDR3
data	** Git	NoosProgramProject/(9_	/004_manual_relocate_data)**		

9.3.2.3 1 RAM

Chapter 2: Memory Maps

RAM 0x900000 ~ 0x91FFFF .data 0x900000

9.3.2.4 2

.data

```
. = ALIGN(4);
.rodatabin : { *(.rodatabin) }

. = ALIGN(4);

data_load_addr = .; /0x8010xxxx
```

.data (runtime address) 0x900000 data_load_addr .bin .data .rodatabin

```
.data 0x900000 : AT(data_load_addr)
```

.data data_start .data data_end

```
{
    data_start = . ; /addr = 0x900000
    *(.data)
    data_end = . ; /addr = 0x900000+SIZEOF(.data)
}
```

imx6ull.lds

```
SECTIONS {
    . = 0x80100000;

    . = ALIGN(4);
    .text :
    {
        *(.text)
    }
```

```

. = ALIGN( 4);
.rodata : { *(.rodata) }

. = ALIGN( 4);

data_load_addr = .; [0][0][0][0]
.data 0x900000 : AT( data_load_addr)
{
    data_start = . ;
    *(.data)
    data_end = . ;
}

. = ALIGN( 4);
__bss_start = .;
.bss : { *(.bss) *(.COMMON) }
__bss_end = .;
}

```

CPU

RAM

.data

RAM

.data

DDR3

.data

image not found or type unknown



9.3.2.5 3 .data

copy_data data

start.S

```

/* */
ldr sp,=0x80200000

/* data */
bl copy_data

/* bss */
bl clean_bss

```

copy_data

start.S

```
copy_data:
    /* data */
    ldr r1, =data_load_addr/* data , , 0x8010xxxx */
    ldr r2, =data_start/* data , , 0x900000 */
    ldr r3, =data_end/* data , , 0x90xxxx */

cpy:
    ldr r4, [r1]/* r1 r4 */
    str r4, [r2]/* r4 r2 */
    add r1, r1, #4/* r1+1 */
    add r2, r2, #4/* r2+1 */
    cmp r2, r3/* r2 r3 */
    bne cpy/* */

    mov pc, lr/* copy_data */
```

9.3.2.6 3 4-1.4

9.3.2.7 4 4-1.4

** Git NoosProgramProject/ (9_ /003_without_relocation)
NoosProgramProject/ (9_ /004_manual_relocate_data)** data

9.4 C data bss

data bss C data bss

9.4.1

Git NoosProgramProject/(9_ /005_relocate_data_with_c)

9.4.1.1 1

start.S copy_data, clean_bss C C r0~r4 C

start.S

```

.text
.global _start

_start:

    /*      */
    ldr sp, =0x80200000

    /*  data  */
    ldr r0, =data_load_addr/* data      (0x8010...) */
    ldr r1, =data_start/* data      , 0x900000 */
    ldr r2, =data_end/* data      (      0x90...) */
    sub r2, r2, r1/* r2  data      */

    bl copy_data/*      copy_data  r1,r2,r3      */

    /*  bss  */
    ldr r0, =__bss_start
    ldr r1, =__bss_end

    bl clean_bss/*      clean_bss  r0, r1      */

    /*      */
    bl main

halt:
    b  halt

```

9.4.1.2 2 init.c copy_data, clean_bss

init.c

```

/*      src, dest, len  */
void copy_data (volatile unsigned int *src, volatile unsigned int *dest, unsigned int len)
{
    unsigned int i = 0;

    while (i < len)
    {
        *dest++ = *src++;
    }
}

```

```

        i += 4;
    }
}

```

```

/*      start, end */
void clean_bss (volatile unsigned int *start, volatile unsigned int *end)
{
    while (start <= end)
    {
        *start++ = 0;
    }
}

```

- copy_data src, dest, len r1, r2, r3
- clean_bss start, end r0, r1

9.4.1.3 3 Makefile

Makefile init.c init.o

Makefile

```

08 relocate.img : start.S  uart.c main.c my_printf.c init.c
09      $(CC) -nostdlib -g -c -o start.o start.S
10      $(CC) -nostdlib -g -c -o uart.o uart.c
11      $(CC) -nostdlib -g -c -o main.o main.c
12      $(CC) -nostdlib -g -c -o my_printf.o my_printf.c
13      $(CC) -nostdlib -g -c -o init.o init.c
14
15      $(LD) -T imx6ull.lds -g start.o uart.o main.o my_printf.o init.o -o relocate.elf -lgcc
-L/home/book/100ask_imx6ull-sdk/ToolChain/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-
gnueabihf/lib/gcc/arm-linux-gnueabihf/6.2.1

```

9.4.1.4 3 4-1.4

9.4.1.5 4 4-1.4


```

12 void copy_data (void)
13 {
14     /*          data_load_addr, data_start, data_end */
15     extern int data_load_addr, data_start, data_end;
16
17     volatile unsigned int *dest = (volatile unsigned int *)&data_start;
18     volatile unsigned int *end = (volatile unsigned int *)&data_end;
19     volatile unsigned int *src = (volatile unsigned int *)&data_load_addr;
20
21     /*          */
22     while (dest < end)
23     {
24         *dest++ = *src++;
25     }
26 }

```

```

38 void clean_bss(void)
39 {
40     /* lds      __bss_start, __bss_end */
41     extern int __bss_end, __bss_start;
42
43     volatile unsigned int *start = (volatile unsigned int *)&__bss_start;
44     volatile unsigned int *end = (volatile unsigned int *)&__bss_end;
45
46     while (start <= end)
47     {
48         *start++ = 0;
49     }
50 }

```

9.4.2.3 3 4-1.4

9.4.2.4 4 4-1.4

ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}



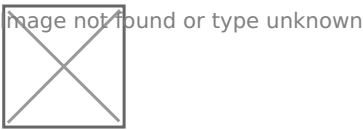
9.4.3 : C

C

1. C extern extern int _start;
2. (&) int * p = & _start; //p lds _start

C

C int g_i = 10; 4 g_i .bin C



&g_i addr
&a1 value

9.5

9.5.1 C

100ask_imx6ull DDR3 S3C2440 .bin
.bin RAM

Git NoosProgramProject/(9_ /007_relocate_all_with_c)

RAM RAM 128KB

9.5.1.1 1

1. 0x900000
2. .data

3. _load_addr Makefile entry C

imx6ull.lds

```
01 SECTIONS {
02     _load_addr = 0x80100000;
03
04     . = 0x9000000;
05
06     . = ALIGN(4);
07     .text      :
08     {
09         *(.text)
10     }
11
12     . = ALIGN(4);
13     .rodata : { *(.rodata) }
14
15     . = ALIGN(4);
16     .data : { *(.data) }
17
18     . = ALIGN(4);
19     __bss_start = .;
20     .bss : { *(.bss) *(.COMMON) }
21     __bss_end = .;
22 }
```

9.5.1.2 2 init.c

.data

copy_data

init.c

```
12 void copy_data (void)
13 {
14     /*          _start, __bss_start, */
15     extern int _load_addr, _start, __bss_start;
16
17     volatile unsigned int *dest = (volatile unsigned int *)&_start;    //_start =
```

```

0x900000
18     volatile unsigned int *end = (volatile unsigned int *)&__bss_start;    //__bss_start =
0x9xxxxx
19     volatile unsigned int *src = (volatile unsigned int *)&_load_addr;    //_load_addr =
0x80100000
20
21     /*      */
22     while (dest < end)
23     {
24         *dest++ = *src++;
25     }
26 }

```

9.5.1.3 3

ldr pc, = xxx

start.S

```

16     /*      */
17     // bl main[ ]/*      DDR3      */
18     ldr pc, =main[ ]/*      RAM      */

```

9.5.1.4 3 4-1.4

9.5.1.5 4 4-1.4

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ[ \]^_`abcdefghijklmnopqrstuvwxyz{| }
123456789: ; <=>?@A

```

9.5.2

copy_data

RAM

RAM

relocate.dis

```

07 00900000 <_start>:
08 900000: e59fd00c ldr sp, [pc, #12] ; 900014 <halt+0x4>
09 900004: fa00016f blx 9005c8 <copy_data>
10 900008: fb000180 blx 900612 <clean_bss>
11 90000c: e59ff004 ldr pc, [pc, #4] ; 900018 <halt+0x8>

```

CPU 0x80100000 entry

9 10 blx pc + offset 0x9005c8

1. 0x80100000 pc = 0x80100000 + 8 = 0x80100008

2. offset = 0x9005c8 + 8 - 0x900004 = 0x5cc 0x80100004 0x8010

3. copy_data clean_bss ldr pc, =main pc = 0x90000c + 8 + 4 = 0x900018

b bl

-

a) static

b) rodata

- ldr pc = xxx runtime address

Revision #1

Created 3 March 2022 02:41:48 by

Updated 3 March 2022 02:42:06 by