[u-boot.lds文件详解](http://blog.csdn.net/xinyuwuxian/article/details/8931650)

分类： [Uboot Study](http://blog.csdn.net/xinyuwuxian/article/category/1403008)2013-05-15 18:04 375人阅读 [评论](http://blog.csdn.net/xinyuwuxian/article/details/8931650#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/xinyuwuxian/article/details/8931650#report)

[uboot](http://www.csdn.net/tag/uboot)

**u-boot.lds文件详解**

网上大部分u-boot.lds文件的分析大部分都是千遍一律，例如下面就是本人在网上找到的关于u-boot.lds的资料。

OUTPUT\_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")

/\*指定输出可执行文件是elf格式,32位ARM指令,小端\*/  
OUTPUT\_ARCH(arm)

/\*指定输出可执行文件的平台为ARM\*/  
ENTRY(\_start)

/\*指定输出可执行文件的起始代码段为\_start\*/  
SECTIONS  
{

/\*指定可执行image文件的全局入口点，通常这个地址都放在ROM(flash)0x0位置。必须使编译器知道这个地址，通常都是修改此处来完成\*/  
 . = 0x00000000;/\*;从0x0位置开始\*/  
 . = ALIGN(4);/\*代码以4字节对齐\*/  
 .text :  
 {  
  cpu/arm920t/start.o (.text)

    /\*代码的第一个代码部分\*/    
  \*(.text)

  /\*下面依次为各个text段函数\*/  
 }  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 .rodata : { \*(SORT\_BY\_ALIGNMENT(SORT\_BY\_NAME(.rodata\*))) }

 /\*指定只读数据段\*/  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 .data : { \*(.data) }  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 .got : { \*(.got) }

/\*指定got段, got段是uboot自定义的一个段, 非标准段\*/  
 . = .;  
 \_\_u\_boot\_cmd\_start = .;

/\*把\_\_u\_boot\_cmd\_start赋值为当前位置, 即起始位置\*/  
 .u\_boot\_cmd : { \*(.u\_boot\_cmd) }

 /\*指定u\_boot\_cmd段, uboot把所有的uboot命令放在该段.\*/  
 \_\_u\_boot\_cmd\_end = .;

 /\*把\_\_u\_boot\_cmd\_end赋值为当前位置,即结束位置\*/  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 \_\_bss\_start = .;

 /\*把\_\_bss\_start赋值为当前位置,即bss段的开始位置\*/  
 .bss (NOLOAD) : { \*(.bss) . = ALIGN(4); }

/\*指定bss段,告诉加载器不要加载这个段\*/  
 \_\_bss\_end = .;

/\*把\_end赋值为当前位置,即bss段的结束位置\*/  
}

看完上面的解析思路本来应该是很清晰的，于是乎编译u-boot，查看一下System.map,

30100000 T \_start

30100020 t \_undefined\_instruction

30100024 t \_software\_interrupt

30100028 t \_prefetch\_abort

3010002c t \_data\_abort

30100030 t \_not\_used

30100034 t \_irq

30100038 t \_fiq

发现 \_start 的链接地址不是u-boot.lds中.text 的当前地址0x00000000，而是0x30100000，这就产生很多疑问了：

(1)     为什么u-boot.lds指定的 .text 的首地址不起作用？

(2)     0x30100000是什么地址，由谁指定.text的首地址是0x30100000的呢？

(3)     假如有其他动作改变了 .text 的首地址，那么该动作跟u-boot.lds的优先级又是怎么决定的呢？

其实这三个问题都在Makefile的LDFLAGS 变量和u-boot.lds 中找到答案。我们不妨试着修改一下u-boot.lds，把u-boot.lds修改成如下(红色字体部分为修改过部分)：

OUTPUT\_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")

/\*指定输出可执行文件是elf格式,32位ARM指令,小端\*/  
OUTPUT\_ARCH(arm)

/\*指定输出可执行文件的平台为ARM\*/  
ENTRY(\_start)

/\*指定输出可执行文件的起始代码段为\_start\*/  
SECTIONS  
{

/\*指定可执行image文件的全局入口点，通常这个地址都放在ROM(flash)0x0位置。必须使编译器知道这个地址，通常都是修改此处来完成\*/  
 . = 0x30000000;/\*;从0x0位置开始\*/  
 . = ALIGN(4);/\*代码以4字节对齐\*/

.rodata : { \*(SORT\_BY\_ALIGNMENT(SORT\_BY\_NAME(.rodata\*))) }  
 . = ALIGN(4);

/\*代码以4字节对齐\*/

 .text :  
 {  
  cpu/arm920t/start.o (.text)

    /\*代码的第一个代码部分\*/    
  \*(.text)

  /\*下面依次为各个text段函数\*/  
 }

 /\*指定只读数据段\*/  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 .data : { \*(.data) }  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 .got : { \*(.got) }

/\*指定got段, got段是uboot自定义的一个段, 非标准段\*/  
 . = .;  
 \_\_u\_boot\_cmd\_start = .;

/\*把\_\_u\_boot\_cmd\_start赋值为当前位置, 即起始位置\*/  
 .u\_boot\_cmd : { \*(.u\_boot\_cmd) }

 /\*指定u\_boot\_cmd段, uboot把所有的uboot命令放在该段.\*/  
 \_\_u\_boot\_cmd\_end = .;

 /\*把\_\_u\_boot\_cmd\_end赋值为当前位置,即结束位置\*/  
 . = ALIGN(4);

/\*代码以4字节对齐\*/  
 \_\_bss\_start = .;

 /\*把\_\_bss\_start赋值为当前位置,即bss段的开始位置\*/  
 .bss (NOLOAD) : { \*(.bss) . = ALIGN(4); }

/\*指定bss段,告诉加载器不要加载这个段\*/  
 \_\_bss\_end = .;

/\*把\_end赋值为当前位置,即bss段的结束位置\*/  
}

上面对u-boot.lds主要做了两点修改

(1)     把0x00000000 改成 0x30000000。

(2)     把 .text 和 .rodata 存放的地址调换了位置。

重新编译 u-boot, 查看System.map

30000000 R version\_string

30000028 r C.27.2365

.

.

.

30100000 T \_start

30100020 t \_undefined\_instruction

.

.

.

从上面的System.map部分内容可以看出:

(1)     u-boot.lds设定的地址(0x00000000或0x30000000)是有效的。

(2)     .text的地址仍然是30100000

跟着我们查看Makefile中的LDFLAGS变量，发现一条指令

LDFLAGS += -Ttext $(TEXT\_BASE)  其中TEXT\_BASE 是在u-boot根目录的board文件夹的对应的开发板名字的子目录下的config.mk文件中定义的

TEXT\_BASE = 0x30100000

看到这里我们应该明白为什么\_start，也就是.text的首地址总是等于0x30100000了，在连接的时候ld命令会把参数-Ttext指定的地址赋给.text,所以.text在u-boot.lds中的默认地址(当前地址)不起作用了。