内核加载地址和入口地址

如果使用 mkimage 生成内核镜像文件的话，会在内核的前头加上了 64 bytes 的信息头，供建立 tag 之用。bootm 命令会首先判断 bootm xxx 这个指定的地址 xxx 与 -a 指定的加载地址是否相同。

如果不同的话会从这个地址开始提取出这个 64 bytes 的头部，对其进行分析，然后把去掉头部的内核复制到 -a 指定的加载地址去运行;  
如果相同的话那就让其原封不同的放在那，但 -e 指定的入口地址会推后 64 bytes，以跳过这 64 bytes 的头部。  
我们来看看这两种不同的情况:

1) mkimage -A arm -O linux -T kernel -C none -a 0x30008000 -e 0x30008040 -n linux-3.0.2 -d zImage uImage

这种情况，只能把 uImage download 到 0x30008000（头部地址） 的位置上，否则从 0x30008040 是启动不了的。这是下载，和从flash读取不同。

原因：如果将 uImage (加了头的镜像文件)下载到不同于指定加载地址的地方，则会进行上面的操作，将去掉头部的内核拷贝到指定的加载地址，此时加地载址和入口地址需要是相同的，因为已经没有镜像头了，所以此时入口地址也应该为 0x30008000，而不应该再加上 64 个字节。所以在构建镜像头部中的加载地址和入口地址时千万要考虑下载的地址，否则将会启动失败。

2) mkimage -A arm -O linux -T kernel -C none -a 30008000 -e 30008000 -n linux-3.0.2 -d zImage uImage

韦东山uboot使用这种情况

这种情况 download 地址随便。 还是按上面说的，因为将加载地址和入口地址设置成同样的地址，在下载到任意地址时，将去掉头部的内核镜像拷贝到指定加载地址后，可以直接从加载地址开始启动。但是要是下载地址和指定加载地址相同呢？也就是下面的:

如果 tftp 下载地址为 0x30008000 , 此时因为下载地址和指定加载地址相同，所以就不会搬动，内核直接从指定加载地址自解压，但是因为指定的入口地址也是 0x30008000，这样的话内核就不会正常启动，所以还得将入口地址往后推后 64 个字节从 0x30008040 启动就能 OK 。

所以在配置下载地址和入口地址是就有以下两种情况：

1. mkimage -n 'linux' -A arm -O linux -T kernel -C none -a 0x30008000 -e

0x30008000 -d zImage uImage

加载地址和入口地址相同

tftp 0x31000000 uImage

bootm 0x31000000

下载地址可以任意放(除了下载地址与加载地址相同的情况)。

韦东山的0x30007fc0，不知道需不需要搬运代码？

2) mkimage -n 'linux' -A arm -O linux -T kernel -C none -a 0x30008000 -e 0x30008040 -d zImage uImage

入口地址在加载地址后面64个字节

tftp 0x30008000 uImage

bootm 0x30008000

下载地址一定要在指定的加载地址上。

Uboot运行地址和链接地址

Uboot.lds链接脚本中指定的地址0x00000000为uboot运行起始地址。而-Ttext指定的地址0x3000f8000为连接地址（代码最终的运行地址）。编译时所有的符号地址都是以链接地址为基准的。因此程序从flash启动时由于运行地址不在链接地址处，因此重定位之前的代码必须使用位置无关码。uboot1.1.6属于该种情况。

uboot2012 链接脚本指定的地址0x00000000为uboot运行起始地址，-T指定的地址也是0x00000000。因此uboot可以直接从flash运行而不需要使用位置无关码,但是因为norflash不能直接写，如果遇到写数据也会出问题。因此仍然需要拷贝到内存执行。但是当代码被拷贝到内存后。由于链接地址和运行地址不一致，那么必须重新修改所有符号的地址到最终的运行地址。编译时加上pie选项可以生成符合信息，最终根据这些信息修改符号

地址。arm-linux-ld –pie –T u-boot.lds –Bstatic –Text 0x0 $UNDEF\_SYM arch /arm/cpu/arm920t/start.o

1.1.6在内存中运行地址是直接设置的连接地址。而uboot2012在内存中的地址则是在程序中计算.

board/100ask24x0/config.mk中的 TEXT\_BASE = 0x33F80000指定链接地址。