

u-boot中环境变量与ulmage中MTD的分区关系

分区只是内核的概念，就是说A~B地址放内核，C~D地址放文件系统，(也就是规定哪个地址区间放内核或者文件系统)等等。

一般我们只需要分3-4个区，第一个为boot区，一个为boot参数区(传递给内核的参数),一个为内核区，一个为文件系统区。（但是有的内核就会有很多分区，比如内核参数会有两个，还有会Logo的地址）

而对于bootloader中只要能将内核下载到A~B区的A地址开始处就可以，C~D区的C起始地址下载文件系统.....这些起始地址在MTD的分区信息中找到。所以bootloader对分区的概念不重要，只要它能把内核烧到A位置，把文件系统烧到C位置即可。

所以，在bootloader对Flash进行操作时，哪块区域放什么是以内核为主（内核中MTD的分区信息可以从内核的代码中看到）。传递给u-boot的参数只要和内核中MTD分区信息一致即可。

而为了方便操作，bootloader类似也引入分区的概念。例如，可以使用“nandwrite 0x3000000 kernel 200000”命令将ulmage烧到kernel分区，而不必写那么长：nand write 3000000 A 200000,也就是用分区名来代替具体的地址

只要在u-boot中设置mtdparts变量，就相当于对应的Nand flash进行了分区，十分方便。而且可以在之后的nand read/write/erase命令中直接使用分区的名字而无需指定绝对地址。

```
set bootargs "noinitrd console=ttySAC0 root=/dev/
mtdblock3 rootfstype=jffs2 mtdparts=nand_flash:128k(u-boot)ro,64k(u-
boot envs),3m(kernel),30m(root.jffs2),30m(root.yaffs) "
```

内核通过bootargs找到文件系统，bootargs中的mtdblockx即代表分区，block0，1，2代表哪个分区是如何确定的。

事实上,bootargs中的"root=/dev/mtdblockx"只是告诉内核,root fs从第x个(x=0,1,2...)MTD分区挂载, mtblock0对应第一个分区,mtblock1对应第二个分区,以此类推.

至于这个分区对应MTD device(NAND Flash)的哪一段范围,取决于内核读到的MTD分区信息,这个分区信息可以通过:

- 1) 写死在MTD层的NAND Controller驱动或者内核其他部分代码里
- 2) 通过U-boot传递给内核的命令行中的mtdparts=...部分解析得出,解析的规则同u-boot中mtdparts变量的赋值规则
- 3) 其他可以让内核知道分区信息的任何办法

在u-boot中给nand分区后是否要对应修改kernel的代码?

如果你用的是通过内核命令行给MTD层传递u-boot中的MTD分区信息,那是不需要的,在这种情况下,内核读取到的分区信息始终和u-boot中的保持一致(推荐的做法)

如果你用的是把分区信息写死在内核源代码里的方法,那最好保证它和u-boot中的保持一致,即同步修改内核的相关部分代码。

mtdparts参数

要想这个参数起作用,内核中的mtd驱动必须要支持,即内核配置时需要选上
Device Drivers ---> Memory Technology Device (MTD) support --->
Command line partition table parsing

mtdparts的格式如下:

```
mtdparts  =<mtddef>[;<mtddef>
<mtddef> := <mtd-id>:<partdef>[,<partdef>]
<partdef> := <size>[@offset][<name>][ro]
<mtd-id>  := unique id used in mapping driver/device
<size>    := standard linux memsize OR "-" to denote all remaining space
<name>    := (NAME)
```

因此你在使用的时候需要按照下面的格式来设置:

mtdparts=mtd-

id:<size1>@<offset1>(<name1>),<size2>@<offset2>(<name2>)

这里面有几个必须要注意的:

a. mtd-id 必须要跟你当前平台的flash的mtd-id一致, 不然整个mtdparts会失效

怎样获取到当前平台的flash的mtd-id?

在bootargs参数列表中可以指定当前flash的mtd-id, 如指定

mtdids:nand0=gen_nand.1, 前面的nand0则表示第一个flash

b. size在设置的时候可以为实际的size(xxM,xxk,xx), 也可以为'-'这表示剩余的所有空间。

相关信息可以查看drivers/mtd/cmdlinepart.c中的注释找到相关描述。