Linux 内核文档

ARM/启动

原文：Documentation/arm/Booting
翻译：tekkamanninja@gmail.com
翻译完成时间：2012 年 4 月 2 日星期一

V1.0

目录

启动 ARM Linux
==========

作者：Russell King
日期：2002 年 5 月 18 日

以下文档适用于 2.4.18-rmk6 及以上版本.

为了启动 ARM Linux，你需要一个引导装载程序（boot loader），它是一个在主内核启动前运行的一个小程序。引导装载程序需要初始化各种设备，并最终调用 Linux 内核，将信息传递给内核。

从本质上讲，引导装载程序应提供（至少）以下功能：

1. 设置和初始化 RAM。
2. 初始化一个串口。
3. 检测机器的类型（machine type）。
4. 设置内核标签列表（tagged list）。
5. 调用内核映像。

1、设置和初始化 RAM

现有的引导加载程序:　　强制
新开发的引导加载程序:　　强制

引导装载程序应该找到并初始化系统中所有内核用于保持系统变量数据的 RAM。这个操作的执行是设备依赖的。(她可能使用内部算法来自动定位和计算所有 RAM，或可能使用对这个设备已知的 RAM 信息，还可能使用任何引导装载程序设计者想到的匹配方法。)

2. 初始化一个串口

现有的引导加载程序:　　可选、建议
新开发的引导加载程序:　　可选、建议

引导加载程序应该初始化并使能一个目标板上的串口。这允许内核串口驱动自动检测哪个串口用于内核控制台。(一般用于调试或与目标板通信。)

作为替代方案，引导加载程序也可以通过标签列表传递相关的'console='选项给内核以指定某个串口，而串口数据格式的选项在以下文档中描述：Documentation/kernel-parameters.txt.

3. 检测机器类型

现有的引导加载程序:　　可选
新开发的引导加载程序:　　强制

引导加载程序应该通过某些方式检测自身所处的机器类型。这是一个硬件代码或通过查看所连接的硬件用某些算法得到，这些超出了本文档的范围。引导加载程序最终必须能提供一个 MACH_TYPE_xxx 值给内核. (详见

linux/arch/arm/tools/mach-types)。

4. 设置启动数据

现有的引导加载程序:　　可选、强烈建议
新开发的引导加载程序:　　强制

引导加载程序必须提供标签列表或者 dtb 映像以传递配置数据给内核。启动数据的物理地址通过寄存器 r2 传递给内核。

4a. 设置内核标签列表

---

bootloader 必须创建和初始化内核标签列表。一个有效的标签列表以 ATAG_CORE 标签开始，并以 ATAG_NONE 标签结束。ATAG_CORE 标签可以是空的，也可以是非空。一个空 ATAG_CORE 标签其 size 域设置为 '2' (0x00000002)。ATAG_NONE 标签的 size 域必须设置为 '0'。

在列表中可以保存任意数量的标签。对于一个重复的标签是追加到之前标签所携带的信息之后，还是会覆盖原来的信息，是未定义的。某些标签的行为是前者，其他是后者。

bootloader 必须传递一个系统内存的位置和最小值，以及根文件系统位置。因此，最小的标签列表如下所示：

```
基地址  ----> +-----------------------+
              | ATAG_CORE       |       |
              +-----------------------+       |
              | ATAG_MEM        |       | 地址增长方向
              +-----------------------+       |
              | ATAG_NONE       |       |
              +-----------------------+       v
```

标签列表应该保存在系统的 RAM 中。

标签列表必须置于内核自解压和 initrd'bootp'程序都不会覆盖的内存区。建议放在 RAM 的头 16KiB 中。

4b. 设置设备树

---

bootloader 必须以 64bit 地址对齐的形式加载一个设备树映像 (dtb) 到系统 RAM 中，并用启动数据初始化它。dtb 格式在文档 Documentation/devicetree/booting-without-of.txt 中。内核将会在 dtb 物理地址处查找 dtb 魔数值（0xd00dfeed），以确定 dtb 是否已经代替标签列表被传递进来。

bootloader 必须传递一个系统内存的位置和最小值，以及根文件系统位置。dtb 必须置于内核自解压不会覆盖的内存区。建议将其放置于 RAM 的头 16KiB 中。但是不可将其放置于“0”物理地址处，因为内核认为：r2 中为 0，意味着没有标签列表和 dtb 传递过来。

5. 调用内核映像

现有的引导加载程序:　　强制
新开发的引导加载程序:　　强制

调用内核映像 zImage 有两个选择。如果 zImge 保存在 flash 中，且是为了在 flash 中直接运行而被正确链接的。这样引导加载程序就可以在 flash 中直接调用 zImage。

zImage 也可以被放在系统 RAM（任意位置）中被调用。注意：内核使用映像基地址的前 16KB RAM 空间来保存页表。建议将映像置于 RAM 的 32KB 处。

对于以上任意一种情况，都必须符合以下启动状态：

- 停止所有 DMA 设备，这样内存数据就不会因为虚假网络包或磁盘数据而被破坏。这可能可以节省你许多的调试时间。

- CPU 寄存器配置

```
r0 = 0，
r1 = （在上面 3 中获取的）机器类型码。
r2 = 标签列表在系统 RAM 中的物理地址，或
     设备树块(dtb)在系统 RAM 中的物理地址
```

- CPU 模式

```
所有形式的中断必须被禁止 (IRQs 和 FIQs)
CPU 必须处于 SVC 模式。(对于 Angel 调试有特例存在)
```

- 缓存，MMUs

```
MMU 必须关闭。
指令缓存开启或关闭都可以。
数据缓存必须关闭。
```

- 引导加载程序应该通过直接跳转到内核映像的第一条指令来调用内核映像。