

嵌入式Linux启动流程

1.硬件上电

2.一级启动

Boot ROM是厂商都无法进行更改，固化在芯片内部的**ROM**：

作用：

确定用于启动的设备

从启动设备的位置搬移一小段代码(4K/8K/16K)到RAM中运行，即**SPL**

3.二级启动：SPL + uboot

1.SPL：Security Program Loader:

作用：

初始化**DDR**：

从启动设备搬移第二段代码到RAM中，加载**uboot**

(有些SOC是将bootloader复制到RAM后，在进行DDR初始化)

2.uboot：Universal boot loader:

U-Boot从启动设备上面读取，分析环境变量获得**Kernel**和**rootfs**存储位置，以及所需的**Kernel command Line**；

自动检测系统RAM和eMMC/Nand Flash容量和参数；

设置以太网口**MAC**地址，并配置好硬件准备加载**Linux kernel**；

加载**Linux kernel**到RAM，至此系统控制权则转移到**kernel**来处理；

4.Linux Kernel

系统**Kernel**初始化Linux，加载**rootfs**，最后启动'**init**'来初始化Linux user space

5.不同的OS

Android、Yocto、ubuntu、fedora等

补充:

一个嵌入式Linux系统从软件的角度看通常可以分为四个层次：

1.引导加载程序。包括固化在固件(firmware)中的boot代码(可选)，和BootLoader两大部分。（bootROM, uboot）

2.Linux内核。特定于嵌入式板子的定制内核以及内核的启动参数。（linux kernel）

3.文件系统。包括根文件系统和建立于Flash内存设备之上文件系统。通常用ramdisk来作为rootfs。

4.用户应用程序。特定于用户的应用程序。有时在用户应用程序和内核层之间可能还会包括一个嵌入式图形用户界面。常用的嵌入式GUI有：**MicroWindows**和**MiniGUI**等。

BootLoader是一段小程序，主要功能是引导操作系统启动。可以把它想象成PC机Linux上的**GRUB/LILO**引导程序，只不过在嵌入式linux中，没有**BIOS**，而是直接从**flash**中运行，来装载内核。它可以初始化硬件设备，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统做好准备。