# Linux内核学习笔记

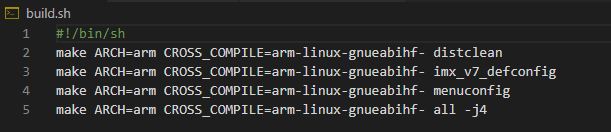
## 学习目标

Linux内核编译

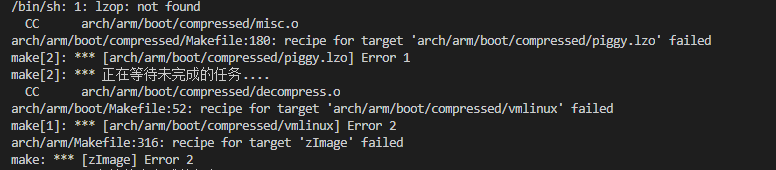
Linux内核目录

## Linux内核编译

与uboot的步骤一样



编译遇到错误



apt-get install lzop 进行解决

Linux内核的默认配置文件保存在arch/arm/configs

## Linux内核目录





arch/arm/boot/ 这个目录，Linux内核编译完成以后，在此目录下生成image，zImage。

arch/arm/boot/dts这个目录，存放Linux内核所有设备树文件，包括dts编译后对应的.dtb文件。

## Linux内核启动流程

从arch/arm/kernel/vmlinux.lds分析Linux内核第一行启动代码。找到ENTRY(stext)

入口函数是stext，image和zImage是经过压缩的，Linux内核会先进行解压缩，解压缩完成以后就要运行Linux内核。要求：

1、MMU关闭

2、D cache关闭

3、I cache无所谓

4、r0 = 0。

5、r1 = machine nr

6、r2=atags 或设备树

Linux内核第一阶段：

\_\_vet\_atags 函数验证atags或dtb是否有效，如果使用设备树的话就是dtb。

\_\_create\_page\_tables 创建页表。

ldr r13, =\_\_mmap\_switched 也就是r13保存\_\_mmap\_switched。

\_\_enable\_mmu 使能MMU

-> \_\_turn\_mmu\_on

->\_mmap\_switched

->start\_kernel 启动内核。

：

Linux内核第二阶段:

start\_kernel

-> rest\_init

-> kernel\_thread(kernel\_init, NULL, CLONE\_FS); 创建kernel\_init进程。也就是

init进程，PID=1

-> kernel\_thread(kthreadd, NULL, CLONE\_FS | CLONE\_FILES); 创建kthreadd

进程，进程PID为2。

-> cpu\_startup\_entry 进入空闲进程，也就是主进程退化为空闲进程，idle。

kernel\_init 进程：

kernel\_init

-> kernel\_init\_freeable

-> 设置标准输入、标准输出、标准错误使用console，比如ttymxc0

-> ramdisk\_execute\_command = "/init";

-> 检查/init是否存在，存在的话就运行。

-> uboot传递给Linux内核的bootargs可以自定init=xxx，或者叫命令行参数。

有一些开发板会设置init=linuxrc。

-> 试着运行/sbin/init。

-> /etc/init

-> /bin/init

-> /bin/sh