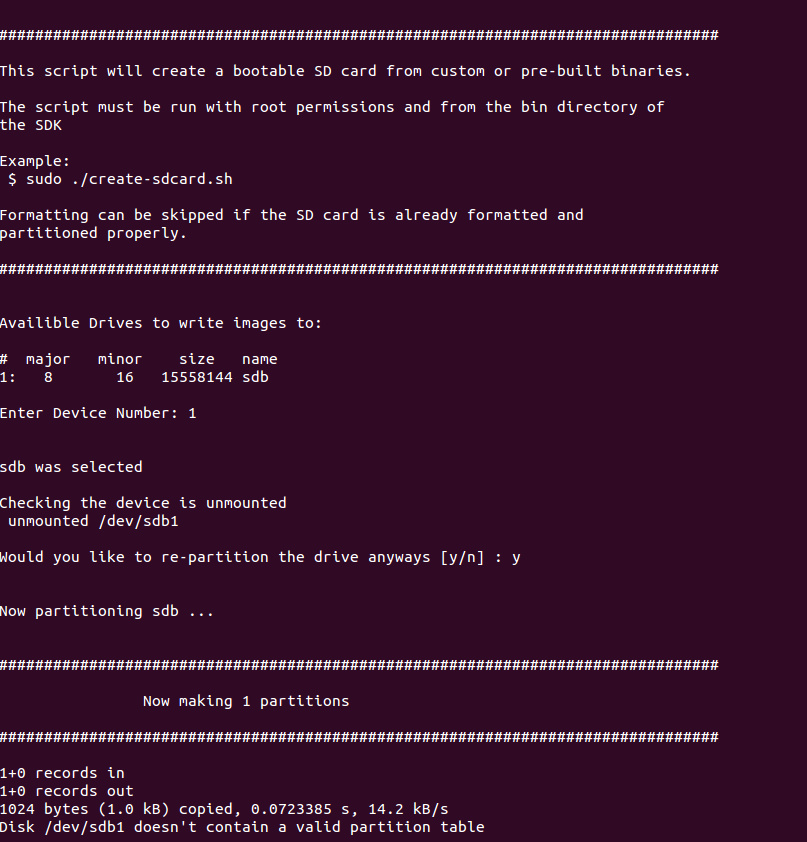
Ok6818 uboot移植学习

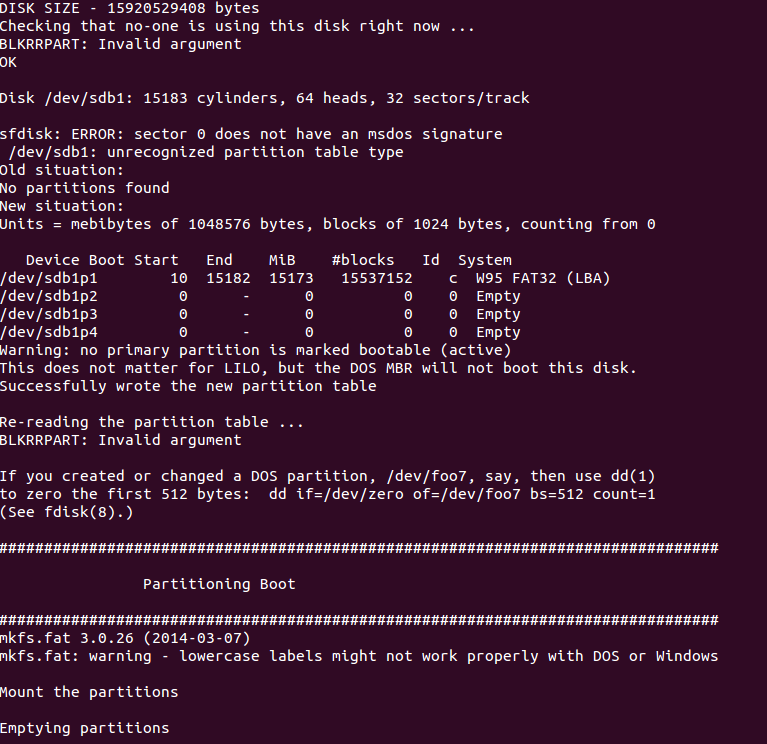
1. SD卡烧写记录
   1. SD制作

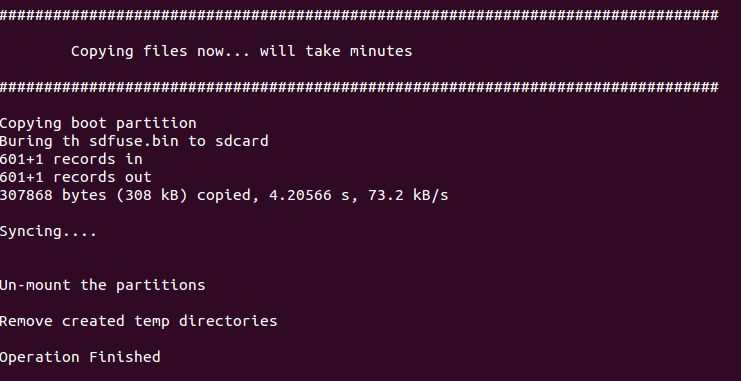
Q1:在ubuntu16.04-64bit中，SD卡被识别为mmcblk0p1，无法通过SD卡制作工具mksdcard.sh,sdfuse.bin(原制作工具的SD卡识别为sdb1，经过修改脚本后，可识别到mmcblk0p1)，原因是？

S1：换到ubuntu12.04-32bit中，SD卡可正常识别为sdb，通过SD卡制作工具可以成功格式化SD卡，并完成ok6818开发板的SD卡烧写。

SD制作信息正确的打印信息如下：







* 1. 烧写串口打印

<sd烧写串口打印(2017.11.28).txt>

* 1. uboot基础
     1. 补丁(patch)文件结构
        1. 补丁头

补丁头是分别由---/+++开头的两行，用来表示要打补丁的文件。---开头表示旧文件，+++开头表示新文件。一个补丁文件中可能包含很多---/+++的补丁头，每一补丁头用来打一个补丁，所以在一个补丁文件中可以包含多个补丁。例如：

--- libmng\_display.c.orig 2007-04-20 10:41:58.000000000 +0200

+++ libmng\_display.c 2007-07-14 00:00:00.000000000 +0200

* + - 1. 块

块是补丁中要修改的地方。它通常由一部分不用修改的东西开始和结束。他们只是用来表示要修改的位置。他们通常以@@开始，结束于另一个块的开始或者一个新的补丁头。

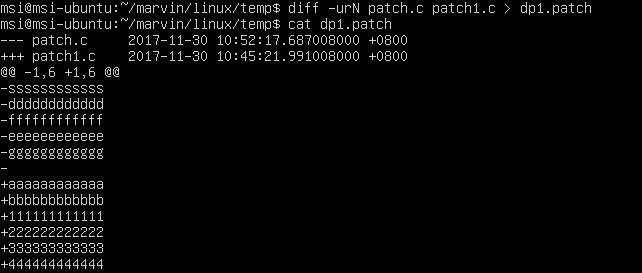
块会缩进一列，这一列是用来表示这一行是要增加还是要删除的。

+号表示这一行是要加上的。

-号表示这一行是要删除的。

没有加号也没有减号表示这里只是引用的而不需要修改

* + - 1. 文件补丁及文件夹补丁
         1. 文件补丁



@@表示一段补丁的开始

-1,6：表示源文件需要修改的行数

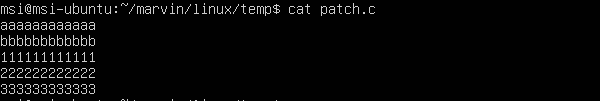
+1,6：表示需要加入的补丁行数。

@@下面表示修改的规则，如上图所示，源文件的1到6行全部删除，并将补丁的1到6行全部加入到源文件中。

* + - * 1. 文件夹补丁
        2. Patch常见问题分析

此处实验是.c文件，未测试文件夹

源文件patch.c：



修改文件patch1.c:

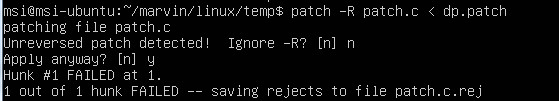


生成的补丁文件dp.patch：

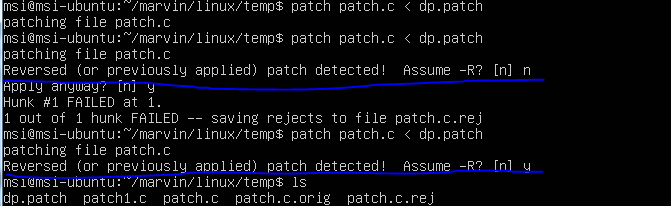


生成命令：diff –urN patch.c patch1.c > dp.patch

1. 对没有patch的源文件(patch.c)，执行：patch –R命令，即撤销补丁操作，会出现以下错误提示:



1. 对已经patch的源文件，再次patch同一个补丁文件，会提示以下错误：



蓝色线条标注的是当错误发生时，是否进行“Assume -R”（翻译：试试撤销操作？）的不同结果。

如果是n,表示不试撤销，结果和1）一样抱错，并生成一个rej文件；当然，如果是y，则表示你同意试试撤销动作，此时再打开源文件，就会发现之前打的补丁被还原了，就相当于对已打补丁的文件（patch.c）执行了一次：patch –R操作。

* + 1. 顶层Makefile分析
    2. 链接脚本u-boot.lds

*.text :*

*{*

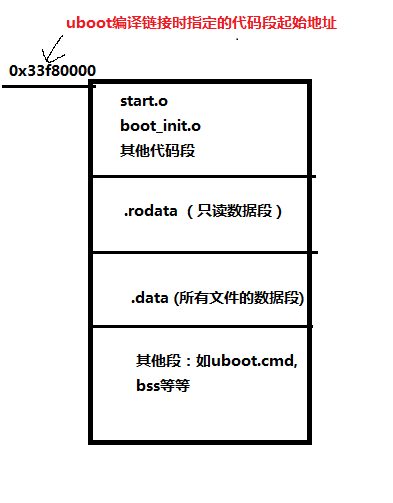
*\*(.\_\_image\_copy\_start)*

*SDIR/start.o (.text\*)*

*\*(.text\*)*

*}*

如图所示，将lds链接脚本定义的段转换成内存图后，各段在sram中的关系：



* 1. 4818/6868 uboot分析
     1. 启动分析

（1）芯片上电后首先自动运行IROM里的代码，IROM是芯片中的一个包含固化代码的ROM存储空间，该段代码主要侦测启动脚的配制，从相应启动介质中抓取16KB代码（Block1-Block32）到ISRAM（0xFFFF0000）中运行。

（2）运行内部ISRAM中的2ndboot.bin代码，复制u-boot.bin到DDR中。

（3）在DDR中运行u-boot.bin。

NanoPi2 MMC镜像分布：

