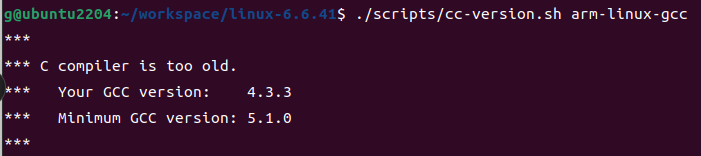
最近想尝试把新的Linux内核移植到tq2440的开发板上，看看还能不能顺利的跑起来。我的基础版本是买板子的时候提供的2.6.30版本，编译器版本是4.3.3.。随着内核版本的升级，编译器的版本也需要升级，写稿时最新的编译器版本是13.3。但是编译器版本也不是越高越好，太新的编译器和老的内核也会出现兼容性问题。

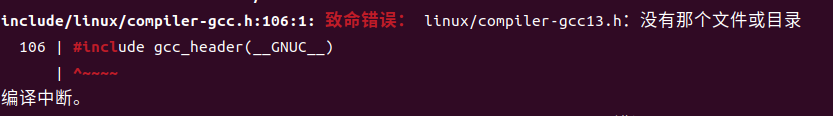
新的内核有个脚本./scripts/cc-version.sh可以检查编译器是否OK，



它调用了另一个脚本./scripts/min-tool-version.sh显示内核支持的最低编译器版本。



旧内核我没找到对应的工具，不过编译器都会提示的，比如版本过高就会出现找不到文件的错误



上面的错误就是我用13.3的编译器编译linux3.4的内核出现的，换成4.3.3的编译器就OK了。

下载linux源码，源码的官方网站是<https://www.kernel.org/>，官方有多个git分支，可以根据自己的需求选择，我选择stable版本：

<https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/>，推荐这个版本，分支比较全。

也可以下载Linus Torvalds的分支：

<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/>。这个分支不多，tag多。

用git可以非常方便的在各个版本之间切换，非常适合实验。

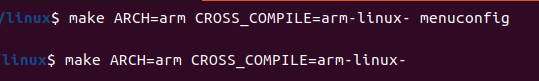
编译前需要配置好环境变量，将编译器的bin路径添加到PATH中，可以修改~/.bashrc或者/etc/profile，这种方式是长期生效的，开机即用，也可以直接在命令行窗口export PATH=[编译器bin路径]:$PATH的方式配置，这种方式是针对此命令行窗口临时的，关掉后就失效了。关于环境变量更详细的内容请自行搜索学习，我这里就不再罗嗦了。

编译内核还需要下载一些必要的工具，

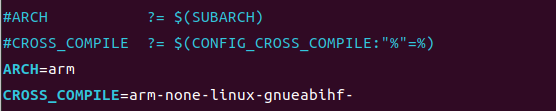
sudo apt install bc binutils bison dwarves flex gcc git gnupg2 gzip libelf-dev libncurses5-dev libssl-dev make openssl pahole perl-base rsync tar xz-utils

这里只是一个例子，随着内核的发展，或许有些工具不需要了，或许版本不对，或者需要其他一些新的工具，不要担心，编译的过程中如果缺乏工具编译器会报错的，通过错误信息我们就知道安装什么了。

前期工作做好了之后就可以进入linux源码目录了，交叉编译需要明确给出两个环境变量ARCH和CROSS\_COMPILE，否则会默认编译x86版本的内核。可以在make命令参数上给出：



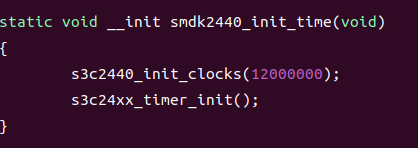
也可以直接写入Makefile文件，我选择这种方式，比较方便；



第二幅图的编译器是最新的，随着版本的更新，编译器也跟着升级，宏的具体值根据你自己版本的情况填入即可。

下面的修改是针对我的板子的，你的可能不需要修改，需要你根据你的板子自行判断，不能盲目照搬。

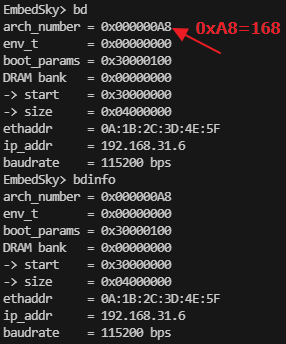
修改系统时钟，我的板子用的12M晶振，内核默认是16.9344M的，这里不改会导致log输出乱码。修改的文件是arch/arm/mach-s3c/mach-smdk2440.c，老的版本是arch/arm/mach-s3c24xx/mach-smdk2440.c或arch/arm/mach-s3c2440/mach-smdk2440.c，需要灵活应对。



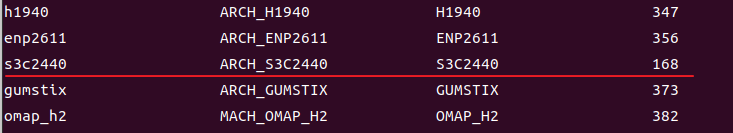
不同的版本这个函数也是有区别的，抓住关键2440和clocks就不会出错。

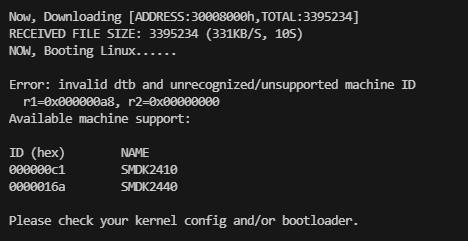
还有个地方需要修改，我的uboot将s3c2440的机器码改成了168，所以Linux这边也得改，这里非常关键，内核之所以知道跑哪个芯片的哪个平台就是通过这个机器码决定的。

想要知道uboot传入的是哪个机器码，可以在uboot的命令行输入bd或bdinfo查看



内核修改文件arch/arm/tools/mach-types，找到s3c2440，将362改成168

如果机器码不对，内核会提示下面的信息

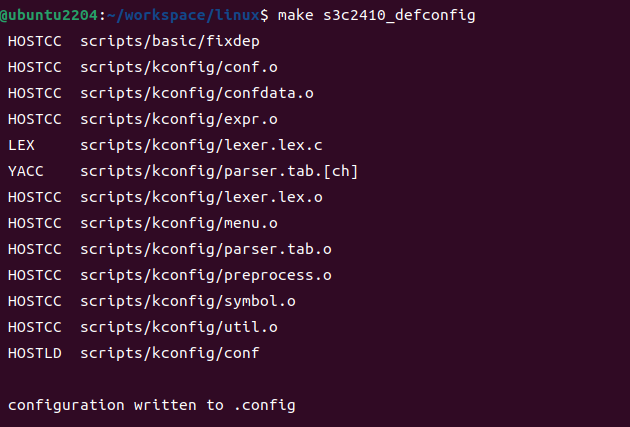


列出的machine跟你的配置有关。

以上的修改对每个版本都要做，我以后就不特别指出了。

接下来准备.config文件，这是内核编译依赖的配置文件，整个内核相关的宏都在这个文件中，我们menuconfig的操作就是在改变这个文件，如果你不确定你想要的配置是否生效或者关闭，就可以在.config中搜索查看。（注意config之前有个英文句号，默认是隐藏文件）

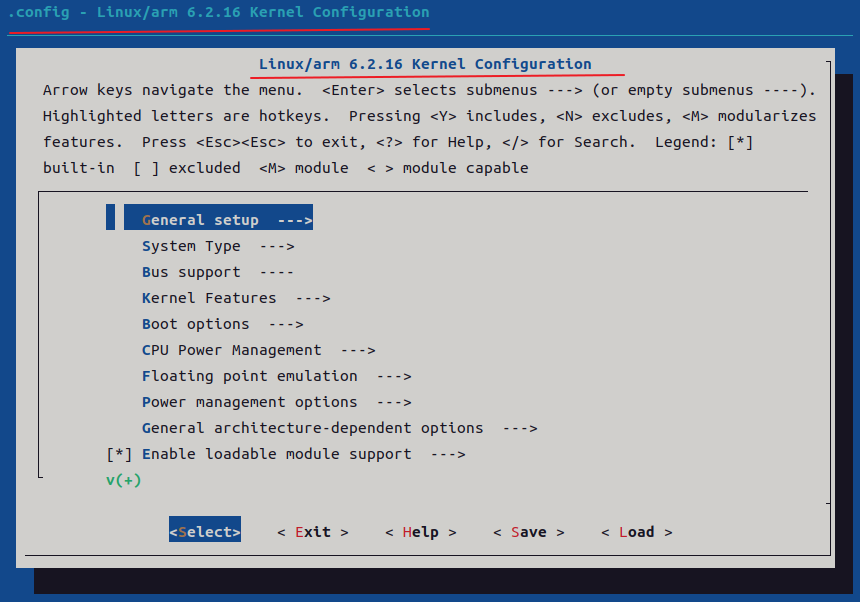
Linux内核提供了一些预定义文件，我们可以拿一个作为基础开始，执行下面的命令，复制预定义文件为.config：



此时已经可以编译了，我的目标是做最少的修改，内核能引导起来就行，也就是内核能运行起来，并且能够输出log。所以我先用默认配置编译，下载到板子看效果，然后再根据情况做进一步的操作。下面我会按照版本讲述不同版本遇到的问题和解决办法，以供大家参考。

（实际做移植肯定要尽可能裁剪掉不需要的东西，否则编译费时间，编译出来的文件还大）

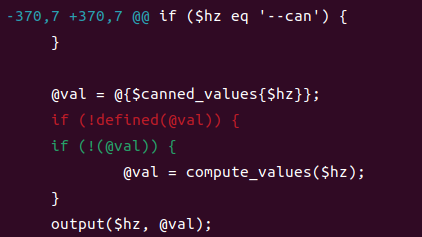
在开始之前，我们先通过命令make menuconfig见一见linux内核的配置界面



最上面一行可以看到使用的配置文件、平台和内核版本，这里就可以确认你的交叉编译环境是否配置正确。这个界面不同的版本会有所不同，需要大家灵活应对。

2.6.30-2.6.34

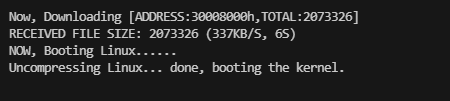
这几个版本用4.3.3编译需要额外修改kernel/timeconst.pl



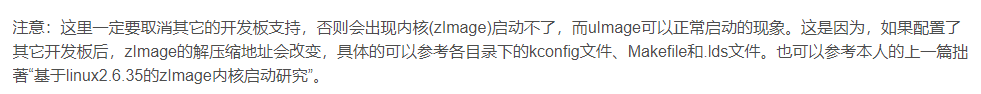
编译通过后下载到板子可以直接引导运行，没什么特别需要注意的，比较顺利。

2.6.35

到这个版本就出现问题了，跟上面通用的方法，这次运行卡在了booting the kernel这里



之后也没有任何输出，然后在网上找到了一个blog的这一段话

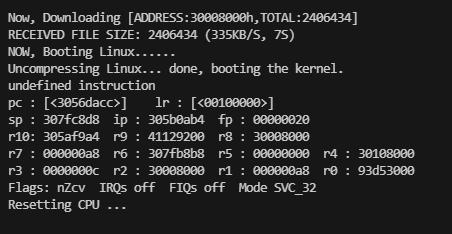


尝试后发现真的可以了。可见，linux旧内核可以的行为在新内核不一定可以。当然这里进一步的原因我没有深究，因为后边linux的升级版本又恢复到不需要去掉其他板子也可以运行的状态了。

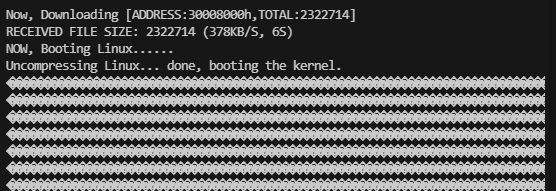
（对于上面的文章是转载的，我没有找到原出处，我就不列出转载源了，感兴趣自己搜索吧）

3.4.y

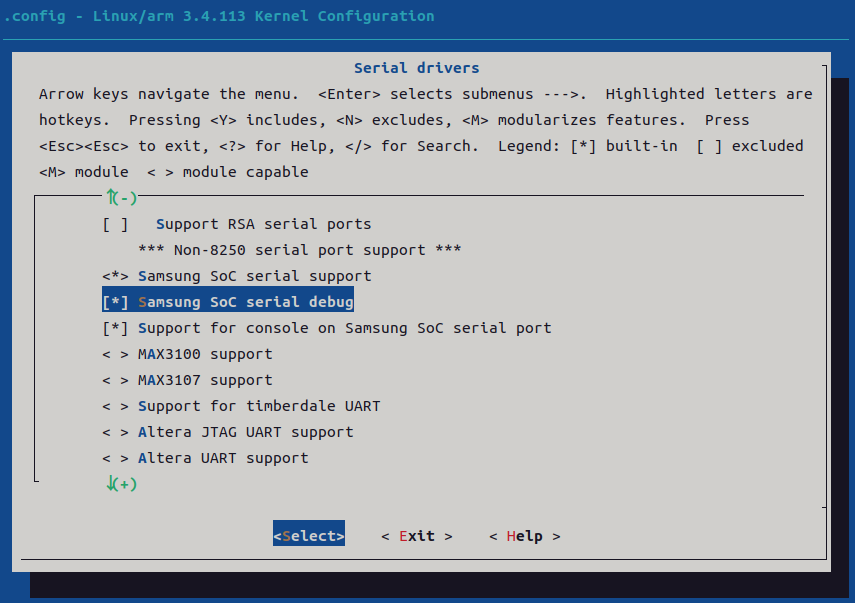
这个版本更绝，上来就是未定义的指令，直接重启。



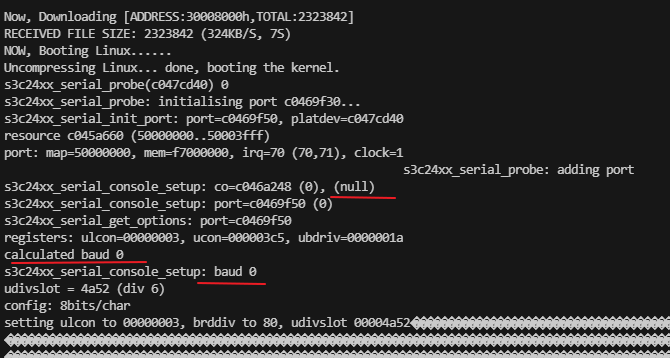
按照之前的思路去掉其他不用的板子，然后成这样了



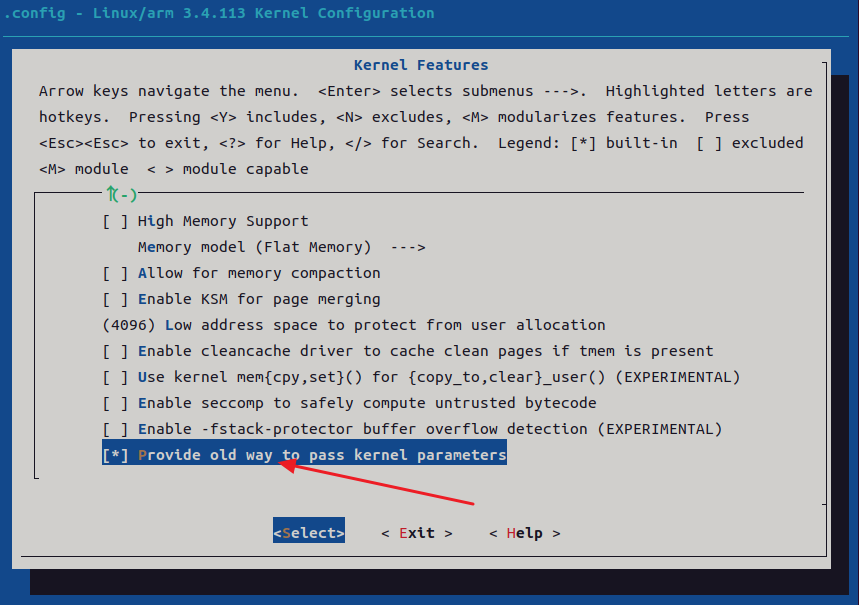
遇到这种情况不要慌，内核提供了调试信息，我们打开调试信息或许能获得线索，之前的版本或许也可以，不过我到这个版本才发现这个配置，你如果是之前的版本可以试试。



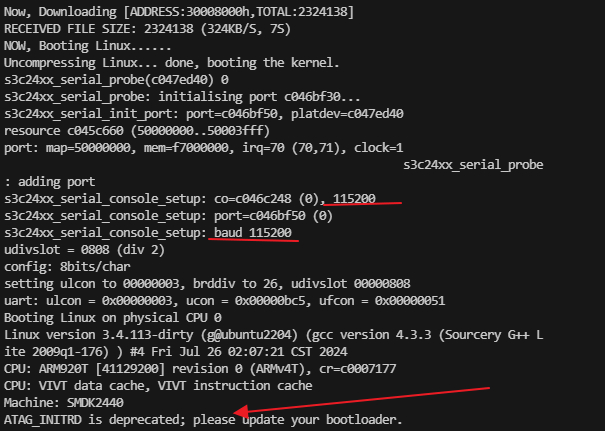
编译下载



这里可以看到波特率是0，很明显有问题。我在uboot的bootargs传入了115200，但是这里没有收到。不卖关子了，这个版本内核加入了设备树，uboot传参的方式已经发生了变化，而我的uboot是旧的，所以需要配置内核用旧的方式传参。



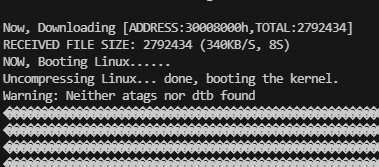
之后再次编译下载就能正常输出了，说明uboot的传参成功了



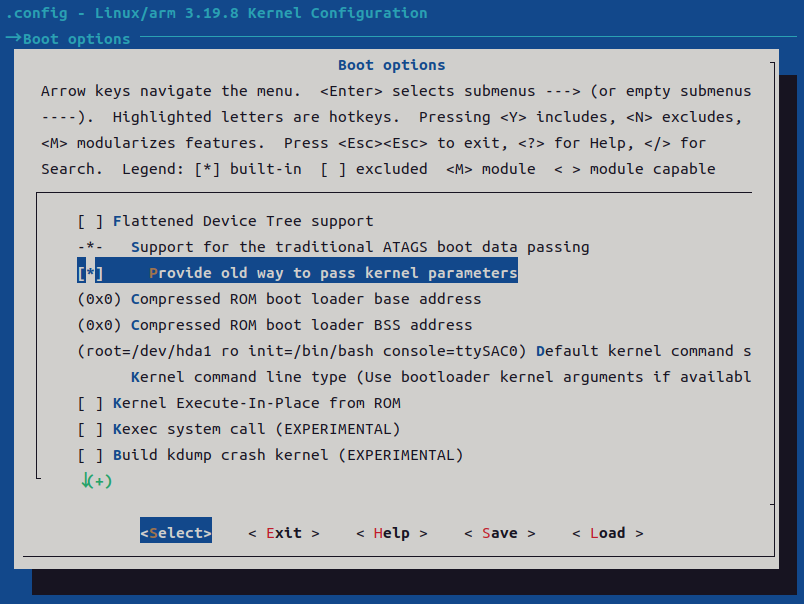
当然内核在这里也提示这种方式过时了，赶紧升级uboot吧。

3.19

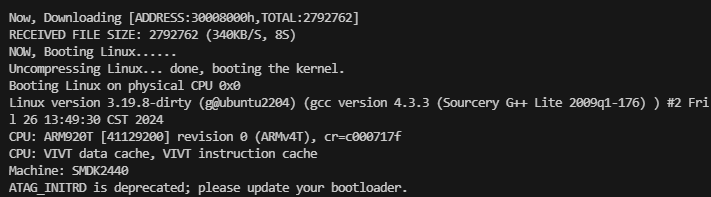
这个版本用默认预定义配置不会出现未定义指令，直接能输出乱码，那看来就是传参问题了。



开启旧方式传参，这个版本把这个选项放到了boot options下，



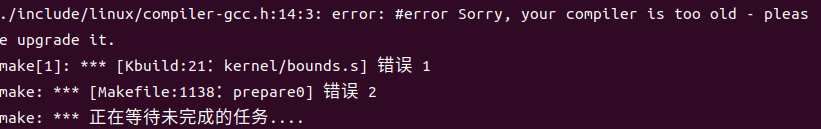
编译下载，这次启动就OK了。



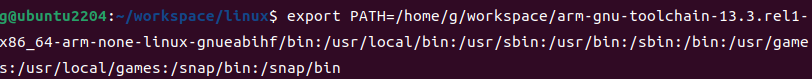
到这里可以发现内核的兼容性又变好了，不需要过多的裁剪也可以正常运行，可见内核升级就是一个不断出现问题又改进问题的过程。

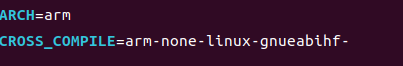
4.19.y

这个版本make时开始提示编译器太老了，需要升级。



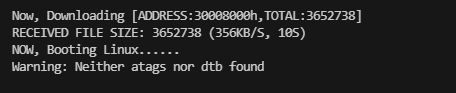
换成最新的编译器



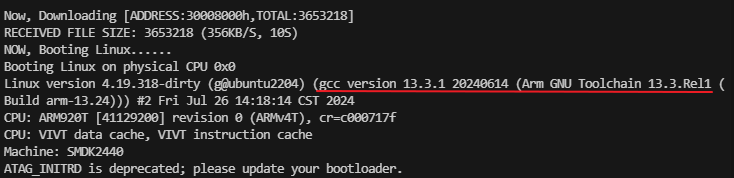


继续编译就OK了。

默认下载进去启动不了

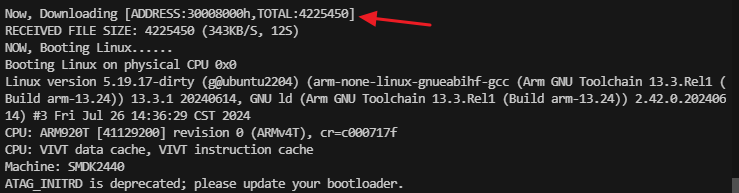


但是之前已经确认传参有问题，先改成旧的方式，编译下载后可以了。



5.19.y

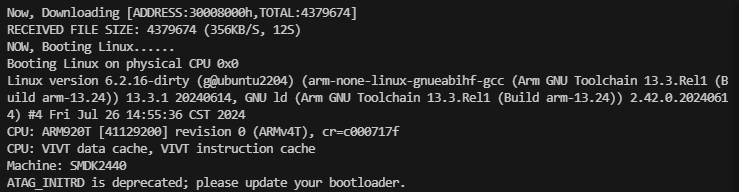
这次在s3c2410\_defconfig的基础上直接开启用旧的方式传参，编译下载。



直接可以启动。注意看，此时内核的大小是4M，而最开始2.6.35的才2M，直接翻倍了。从log也可以看出来内核结构发生了巨大的变化。

6.2.y

修改和配置同5.19，编译引导也正常。



Linux内核6.3以后就不支持s3c2440这颗芯片了。当然旧版本的内核也够用了，这么老的芯片很多都用旧内核，没有特殊需求没有升级新内核的必要，这或许也是新版本不再支持的原因吧。

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=ff0c7e18629b