IMG_256



步骤：

1. 创建Vivado硬件平台，导出xsa文件到Ubuntu中。使用过vivado工具的都知道，最后可以生成一个硬件描述文件，也就是xsa文件，我们在使用petalinux在设计的时候，需要把这个文件导入到Linux中。由于本讲重点在是使用petalinux，所以不再讲解如何使用vivado进行硬件的设计。而是直接选择xsa文件导入到Ubuntu中。

这个文件在资料盘中有提供。

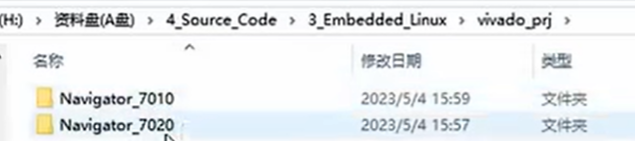
文件路径为：

资料A盘/4\_Source\_Code/3\_Embedded\_Linux 这里面全是压缩包：我们主要解压两个：vivado\_prj.zip 和 zynq\_petalinux:



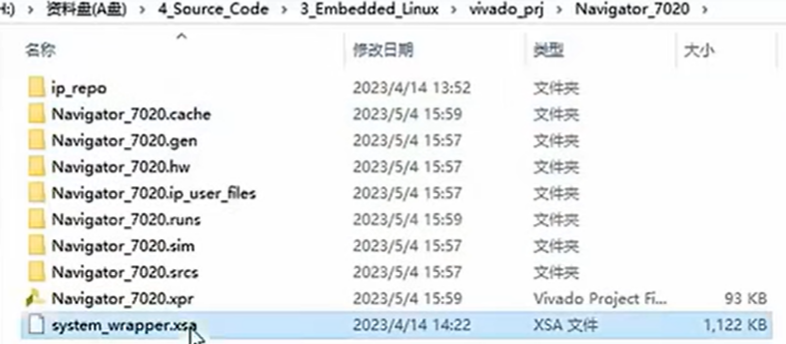
vivado\_prj:

领航者有两块：一个7010一个7020，视频以7020为例。



里面就有xsa文件：

（这整个实际上是一个vivado的工程）



我们把xsa文件复制到共享文件夹下面的 新建的 7020\_xsa文件夹下面：

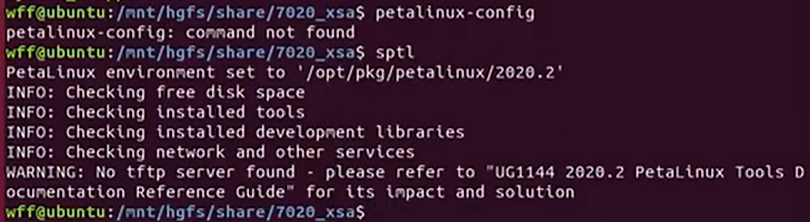


1. 设置petalinux的环境变量：  
   由于我们直接使用petalinux进行定制的时候，直接输入petalinux-config会显示命令找不到: 那是因为没有设置好环境变量。

IMG_256

导致每次打开一个终端都需要设置一次环境变量。在上两节课程中有介绍设置变量别名：sptl。

所以，此时输入sptl就可以设置环境变量。



再次输入petalinux-config就不会提示命令没有找到了。取而代之的是命令指导，指导我们使用这个命令:

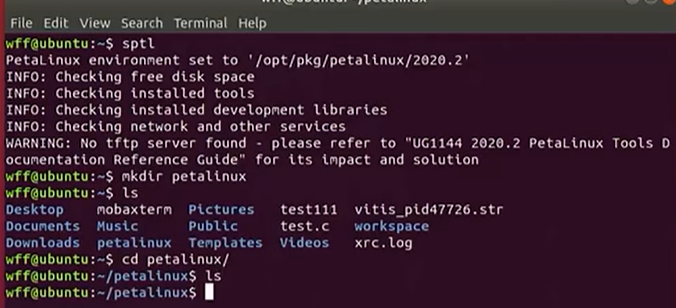


到这里就已经设置好了环境变量了。

1. 创建petalinux工程：

在创建petalinux工程之前，建议先创建一个工程文件夹，用来保存我们所有petalinux工程。

他在用户目录下使用命令：mkdif petalinux。创建了一个petalinux文件夹。然后进入该文件夹：



后续所有的petalinux工程都保存在这个目录下面。本节课就在该文件夹下面创建本课程的petalinux工程。

完整命令如下（这里仅参考命令，命令所在目录是随便找的，应结合实际工程项目位置输入命令）：

petalinux-create -t project --template zynq -n Zynq-7020

IMG_256

\*命令后面的-t是指类型，我们指定的是project，即创建工程类型。

\*命令后面跟的 --template 是指定一个模板的意思。

由于本课程使用的是领航者,是zynq，所以使用的zynq:

IMG_256

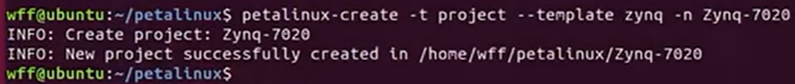
如果使用的是P4、P5,他们是zynqMPSoc，所以使用的就是zynqMP

（注意MP需要大写）:

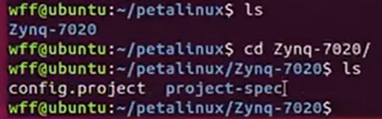
IMG_256

\*命令后面跟的 -n 是指定名字的意思：

创建成功后命令行显示:



进入该文件夹下面：cd Zynq-7020/ 并且可以看到它已经创建了一些文件。



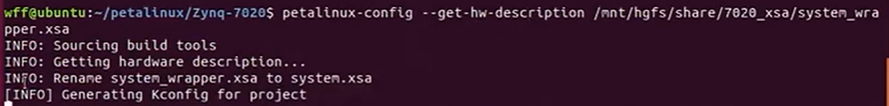
1. 配置petalinux工程：

我们知道，之前共享文件夹已经有了xsa文件，现在就需要进行一些相关的配置。

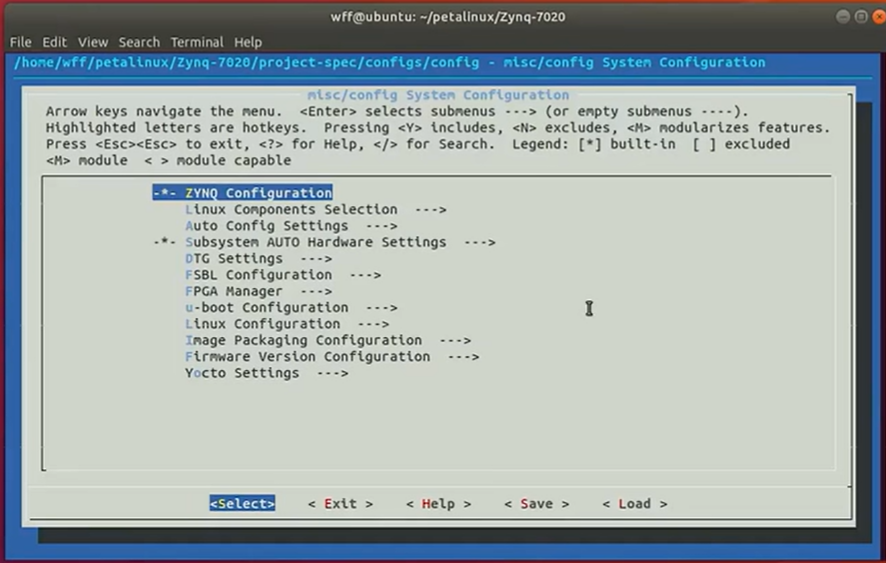
使用petalinux-config命令：

petalinux-config --get-hw-description /mnt/hgfs/share/7020\_xsa/system\_wrapper.xsa

前面部分表示我们要获取硬件描述文件，后面的表示硬件描述文件的路径。他放在共享文件夹下面的。



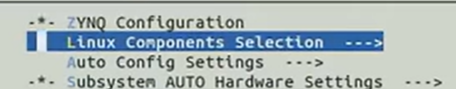
输入命令后的等待的这个过程，它是在生成config，即生成如下图的图形化界面：



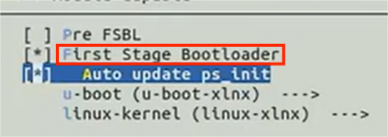
另外，对于该menuconfig的使用上，由于其蓝色标记了首字母，我们直接按首字母，他就会跳到对应的名称上面，如：按L，他就会在如图的两个L开头的选项上面跳转。

（1）

这里，我们进入：

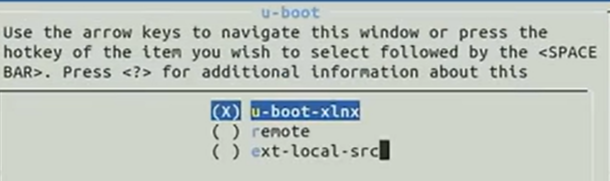


可以看到里面有：FSBL（First Stage Bootloader，我们选中这个）



其次还有自动更新PS\_init（Auto update ps\_init）,这个是默认选中的。

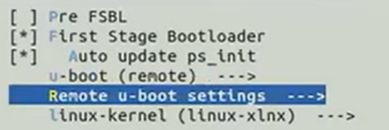
再下面是u-boot，这个是指定u-boot的来源的，我们可以选中看一下:



这里有3个来源：

第一个就是petalinux默认的u-boot。

第二个remote就是远程的,如果选中它，则在上一级就会多一个选项：

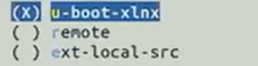


再进入Remote u-boot setting进行远程u-boot的一些设置：

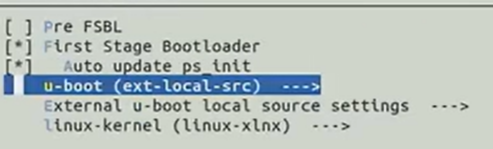


git URL 表示我们可以使用URL网址来选中控制u-boot的获取。

这里还是使用默认的u-boot:

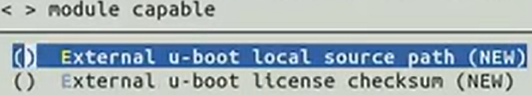


第三个选项ext-local-src表示：使用本地的资源文件，进入可见：



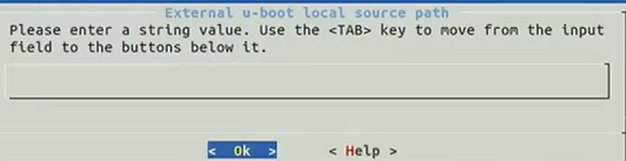
Uboot(ext...)表示外部的uboot

External...表示本地的u-boot,如果本地有u-boot就直接使用本地uboot,就直接选中这一项：

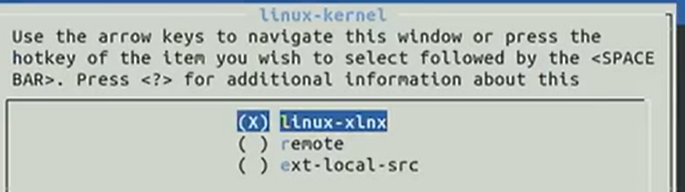


里面:

第一步是路径：回车后就需要输入一个绝对路径。

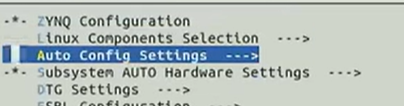


Linux-Kernel表示内核，进入后三个选项同上：

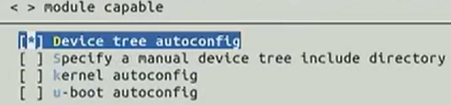


（2）

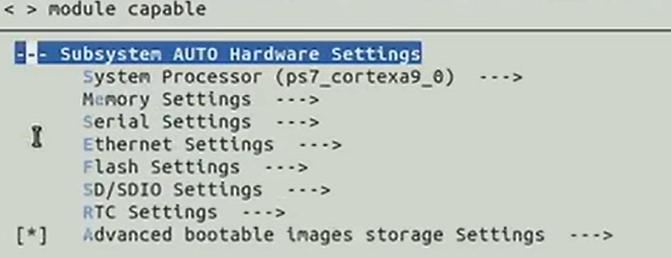
我们进入Auto Config Settings:



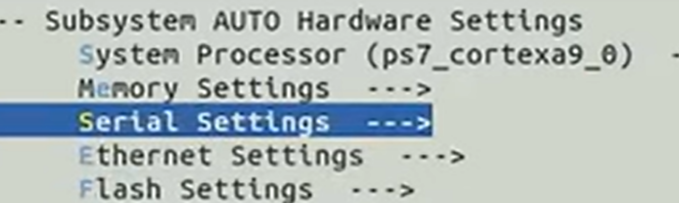
这里面进行设备树以及uboot、内核的一些自动配置。这里就不过多介绍了



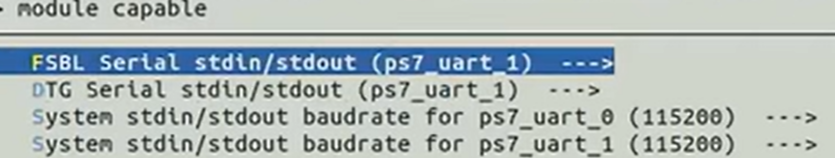
1. Subsystem AUTO Hardware Settings子系统以及一些硬件配置，进入有：



对内存、以太网、Flash、SD、RTC等简单的外设的一些配置。我们主要关注对串口的一些配置：

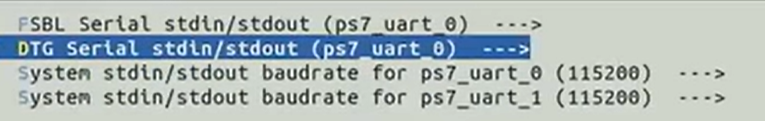


进入后，可看到：

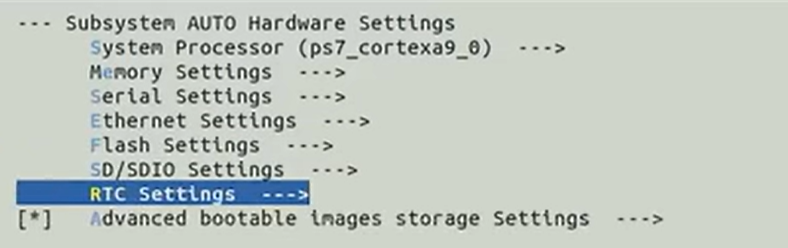


前两个选项指定PS7\_uart\_1,我们需要修改成uart\_0。这是针对领航者开发板的一些修改。因为领航者的开发板uart\_1是PL端的串口。我们这里需要修改成PS端的端口，所以为uart0。

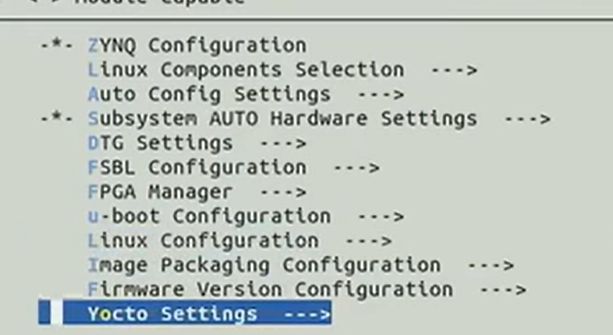
如果是其他的开发板可能就不需要修改。需要结合具体开发板到底是PS还是PL端的串口来配置。



返回上一级：其他的配置就不需要看了，直接使用默认的即可。

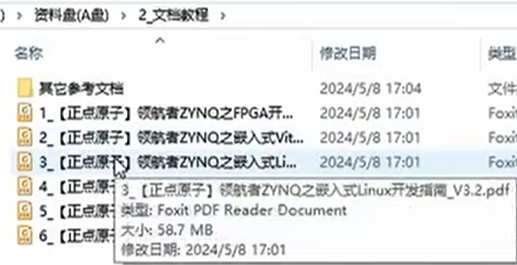


1. 我们再来看Yocto Settings:

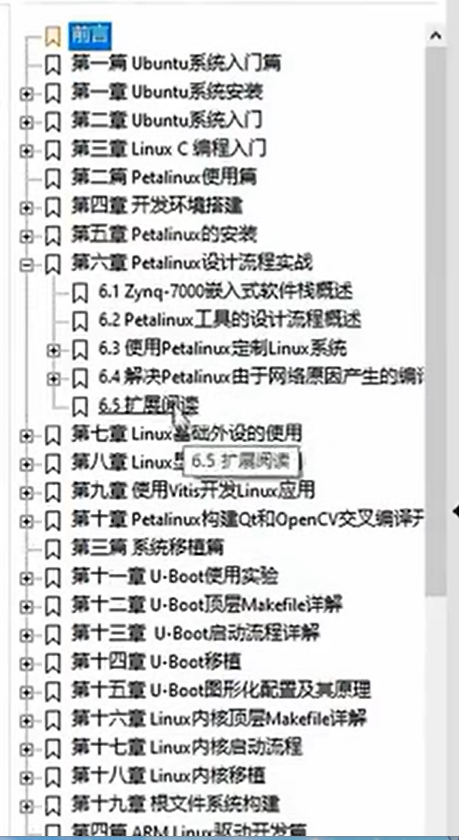


对于这个Setting，我们上两节下载了几个工具包（sstate\_arm 和 downloads），而工具包的用处主要就是在这个配置里面去修改。我们需要用到那几个工具包来帮我们解决报错。建议在第一次配置的时候就把工具包设置好。后续就不会遇到因网络原因导致的编译报错了。

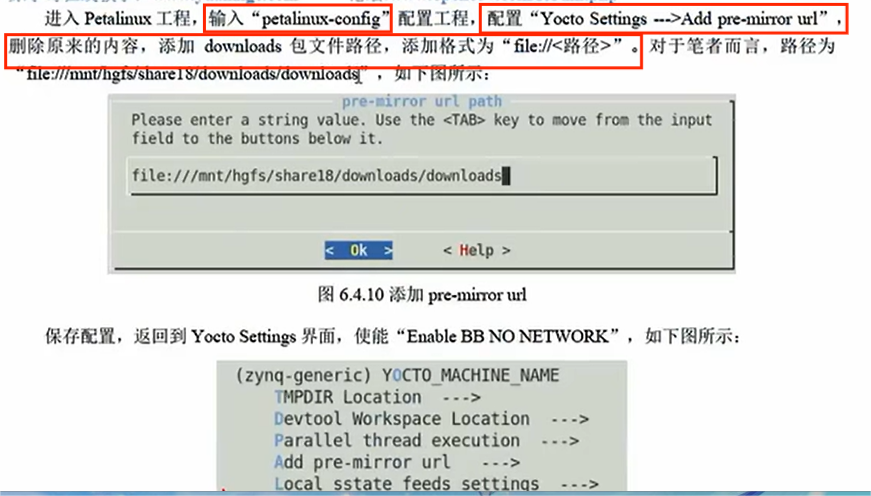
具体如何修改，建议参考Linux开发指南。（领航者ZYNQ：A盘/文档教程/嵌入式Linux开发指南）



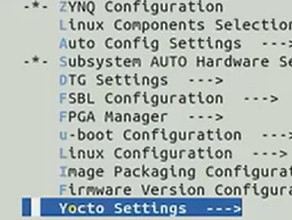
在第6章设计流程中：6.4列了常见的第一次编译的编译报错



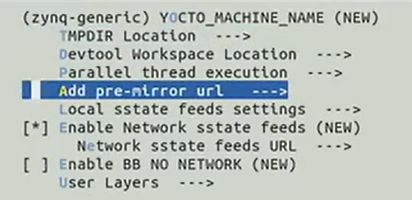
在这一步开始介绍如何配置修改：



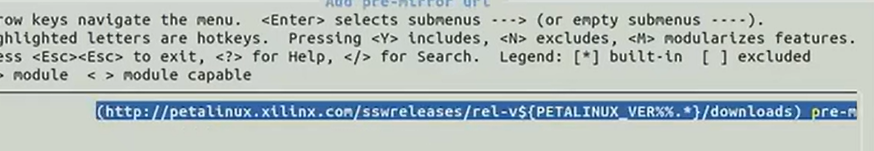
即：进入：



再选中：



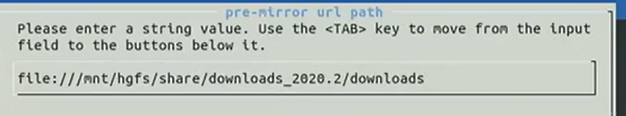
在这里他给了个奇怪的网址，默认去那里下载。



这里修改为：

file: + // + 绝对路径

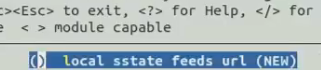
例如：



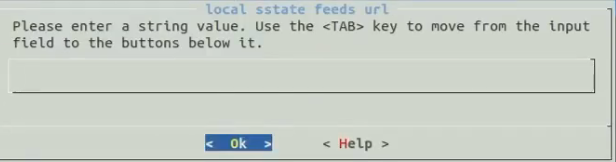
回到上一级：我们知道上一节还下载了ssatate。这里选中如下：



进入后显示：

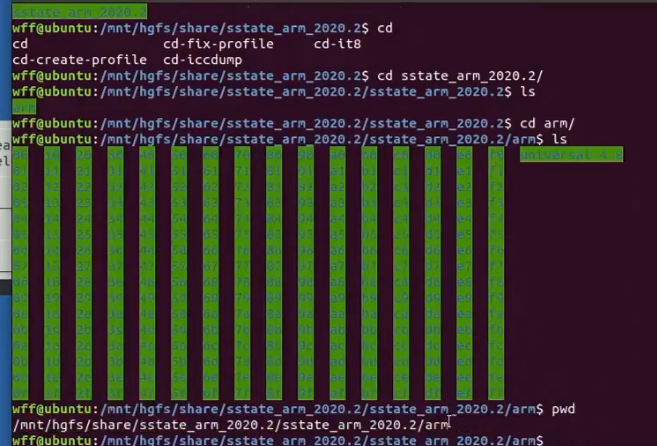


在回车进入：

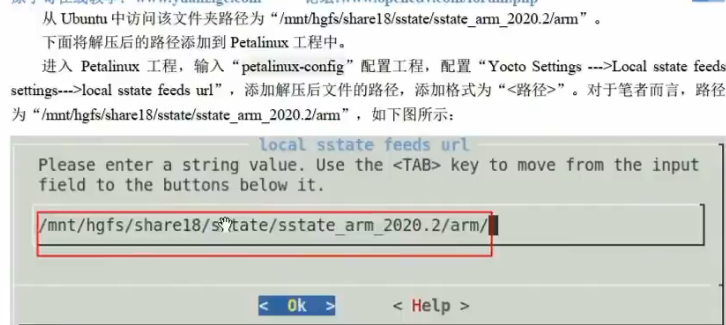


这里同样要输入一个路径。

我们先看一下ssatate文件夹：

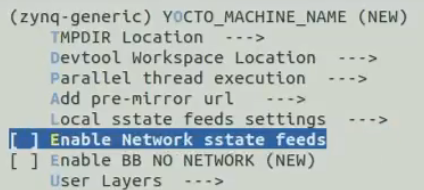


来到上图的路径，选择绝对路径。复制到方框里面。粘贴后直接回车即可，无需再加file。



返回上上级：

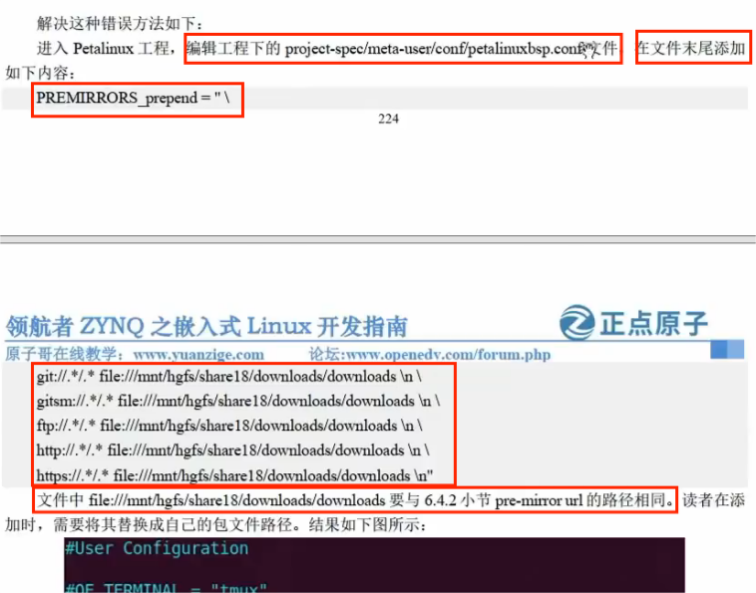
在这里需要取消下面的Enable Network选项：因为指定后他会从默认的网址去下载。



然后一直返回，保存退出。后续开发，每新建一个项目都要如此配置避免报错。

此外，我们可能还需要解决一个报错：





图中的路径就是Donwload的路径，需要结合实际修改

1. 配置Linux内核：

这一讲将在下一节进行赘述。