

关于zynq petalinux 2020.2版本交叉编译环境工具链的搭建和使用备忘

原创 lrmrm 于 2021-04-14 17:09:59 发布 5449 收藏 49 版权

分类专栏: zynq 嵌入式 文章标签: zynq petalinux 交叉编译 工具链 gcc



zynq 同时被 2 个专栏收录

7 订阅 8 篇文章

订阅专栏

用过xilinx zynq petalinux的人都知道petalinux编译一次非常慢，即使下载了ssstate和download包之后编译还是很慢很慢，真是让人难以接受了。。。。

so....经过我一番艰苦研究之后终于弄明白了如何使用petalinux生成的sdk来编译自己的驱动程序，步骤如下：

第一步：**vivado** 搭建硬件环境，编译出xsa，在petalinux下创建工程，按照官方指导步骤最后 petalinux-build 编译一次生成整个linux工程

第二步：使用petalinux-build --sdk, petalinux-package --sysroot命令编译出 **交叉编译 工具链**，这样只会在images/linux/sdk文件夹下就生成了编译工具和sysroot，到此步骤之后，如果你**只想编译linux 应用程序**的话

就打开终端，输入source xxxx/images/linux/sdk/environment-setup-cortexa9t2hf-neon-xilinx-linux-gnueabi 导入sdk到环境变量，然后 输入 arm-xilinx-linux- 按tab键就可以看到petalinux的gcc交叉编译环境已经有了，

之后按照常规方法编译你自己的应用程序就可以了，但是这样只能编译linux 应用程序，要开发petalinux 驱动程序的话是无法编译的，编译提示错误，找不到一堆头文件，要使用petalinux 2020.2的交叉编译器还要下面几个步骤

第三步：**如果你想用petalinux 2020.2自带的sdk交叉编译环境编译自己的驱动程序**的话，makefile需要指定一个**预编译好的内核文件目录** LINUX_KERNEL_PATH := xxx，由于petalinux build出来的文件分散放在不同目录，因此直接使用petalinux工程目录 xxxx/components/yocto/workspace/sources/linux-xlnx下的源码是不行的，编译起来会提示你没有config过，让你重新编译一次，虽然拷贝一份内核代码到别的目录后自己make menuconfig一次能解决，但是这样的话会有一个问题，就是最终即使能正常编译出自己的 ko 文件，但是加载到zynq开发板上是不能 insmod 的，因为zynq上运行的内核和你编译ko的源码内核不是同一个环境编译的，因此不能加载ko。知道原因就好办，想办法把petalinux环境编译内核时生成的目标文件和目录找出来，将文件复制到内核源码下面的对应路径即可，经过一番研究，发现petalinux2020.2编译出的内核文件目录在 xxxx/build/tmp/work-shared/zynq-generic/kernel-build-artifacts这目录下，只要按照这个目录下的文件夹名称，将里面的东西复制到linux内核源码目录下对应名字的文件夹里面即可，经过我试验，确认这个方法是ok的，使用这个方法之后每次编译自己的驱动程序只要几秒钟，原来使用petalinux-build一次5分钟以上，简直爽爆了！

Makefile写法参考如下

```
obj-m := fuckpetalinux.o

CURRENT_PATH := $(shell pwd)
LINUX_KERNEL_PATH :=/xxx/Petalinux_2020.2/linux-xlnx （linux源码目录）

all:
    make ARCH=arm CROSS_COMPILE=arm-xilinx-linux-gnueabi- -C $(LINUX_KERNEL_PATH) M=$(CURRENT_PATH) modules
clean:
    make -C $(CURRENT_PATH) M=`pwd` clean
```

比较好的做法不应该直接复制文件，因为直接复制的话如果要重新编译内核的话，又要重新复制一次，非常麻烦，后面我直接创建petalinux编译后的文件超链接，将超链接复制到linux内核源码目录下，这样之后即使改了硬件，重新编译了内核代码都可以不用修改就能直接编译驱动程序了，记得每次打开终端时source一次即可