Tpm1交叉编译

1. 准备的文件

参考博文:

<https://blog.csdn.net/mkpci/article/details/31355541?utm_source=blogxgwz5>

<https://www.cnblogs.com/hpl20155329/p/9064175.html>

1.1 tpm工具源码准备

解压**kx\_shared.rar** 得到以下三个文件

**gmp-6.1.2**

**tpm-emulator-master**

**trousers-0.3.14**

**tpm-tools\_1.3.8.orig.tar**

存放路径：[\\10.255.21.167\share\zhangchao\tpm1](file:///\\10.255.21.167\share\zhangchao\tpm1)

1.2 交叉编译链准备

**gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi**

存放路径: [\\10.255.21.167\share\zhangchao\crosstools](file:///\\10.255.21.167\share\zhangchao\crosstools)

拷贝解压到虚拟机目录 **/opt** 下

解压后，参考**environment-setup-arm-linux-gnueabi**文件进行环境变量设置，该路径下交叉编译链已经设置。

export SDKTARGETSYSROOT=/opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/arm-linux-gnueabi/libc

#export PATH=/opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/bin:/opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/arm-linux-gnueabi/bin:$PATH

export PATH=/opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/bin:$PATH

export CC="arm-linux-gnueabi-gcc --sysroot=$SDKTARGETSYSROOT"

export CXX="arm-linux-gnueabi-g++ --sysroot=$SDKTARGETSYSROOT"

export CPP="arm-linux-gnueabi-gcc -E --sysroot=$SDKTARGETSYSROOT"

export AS="arm-linux-gnueabi-as "

export LD="arm-linux-gnueabi-ld --sysroot=$SDKTARGETSYSROOT"

export GDB=arm-linux-gnueabi-gdb

export STRIP=arm-linux-gnueabi-strip

export RANLIB=arm-linux-gnueabi-ranlib

export OBJCOPY=arm-linux-gnueabi-objcopy

export OBJDUMP=arm-linux-gnueabi-objdump

export AR=arm-linux-gnueabi-ar

export NM=arm-linux-gnueabi-nm

export CONFIGURE\_FLAGS="--target=arm-linux-gnueabi --host=arm-linux-gnueabi --build=x86\_64-linux --with-libtool-sysroot=$SDKTARGETSYSROOT"

export CFLAGS=" -O2 -pipe -g -feliminate-unused-debug-types "

export CXXFLAGS=" -O2 -pipe -g -feliminate-unused-debug-types "

export LDFLAGS="-Wl,-O1 -Wl,--hash-style=gnu -Wl,--as-needed"

export CPPFLAGS=""

export KCFLAGS="--sysroot=$SDKTARGETSYSROOT"

export ARCH=arm

export CROSS\_COMPILE=arm-linux-gnueabi-

该配置存放在：

/opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi**/environment-setup-arm-linux-gnueabi**

1. **编译**

**2.1 编译编译gmp-6.1.2**

**环境变量设置**

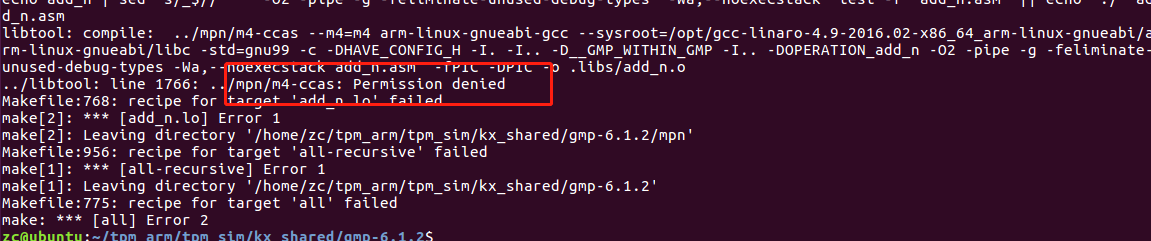
source /opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi**/environment-setup-arm-linux-gnueabi**

**chmod a+x configure**

**./configure --host=arm-linux-gnueabi --target=arm-linux-gnueabi**

**make**

**若报以下的错误**



**m4-ccas 无权限**

**使用**

**chmod a+x mpn/m4-ccas**

**继续执行: make**

**mkdir out make install DESTDIR=输出文件的绝对路径**

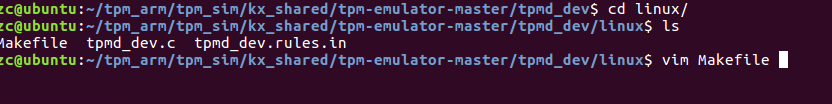
**DESTDIR:指向安装的路径，这个路径是绝对路径，指的是你想存储的路径**

**2.2编译tpm-emulator-master**

2.2.1 首先需要先编译内核。

2.2.2 修改依赖kernel的部分

**cd kx\_shared/tpm-emulator-master/tpmd\_dev/linux**

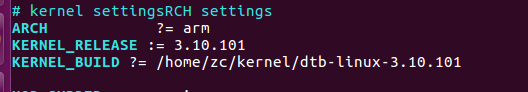
**vim Makefile**

**ARCH：arm**

**KERNEL\_RELEASE= kenel版本号**

**KERNEL\_BUILD = KERNEL的路径**

**修改后示例如下：**



2.2.3 搜索修改链接器

回到tpm\_emulator\_master根目录，进行CMAKE编译

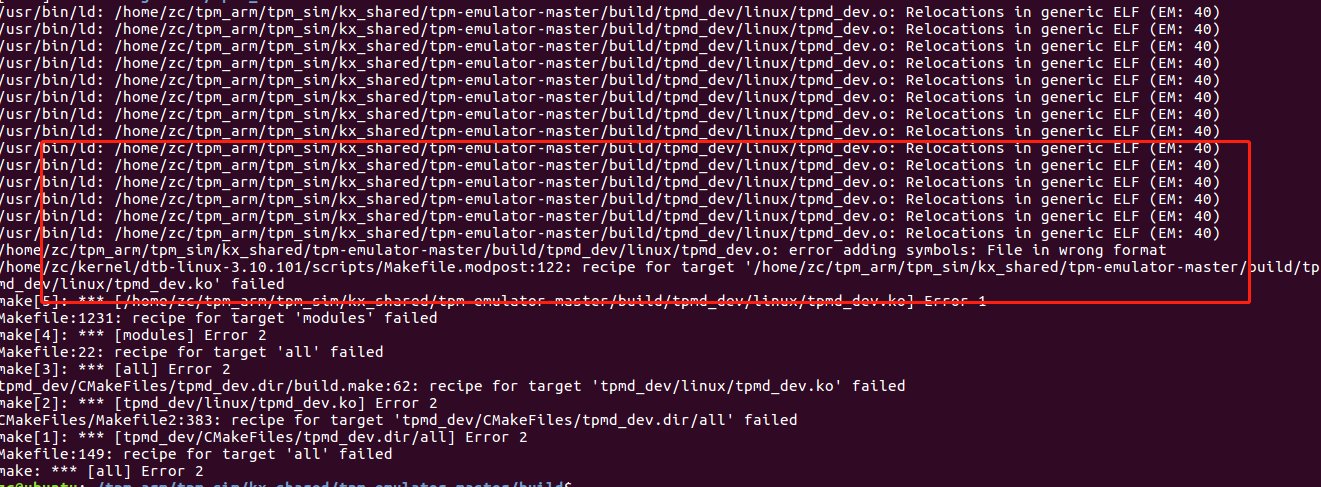
**mkdir build**

**cd build**

**cmake ..**

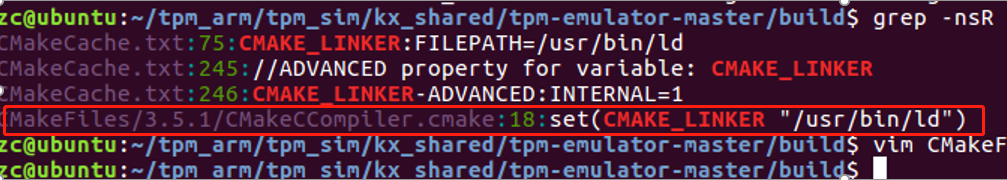
**make**

若make链接出以下错误



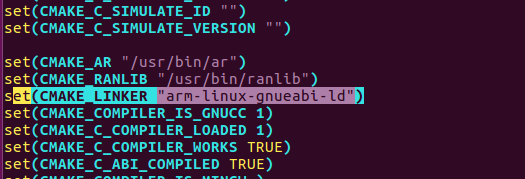
在build文件夹中搜索

**grep -nsR CMAKE\_LINKER**



修改CMakeCCompiler.cmake第18行 为以下示例：

改为：**arm-linux-gnueabi-ld 如下：**



执行：

make

可以创建一个输出目录，mkdir out

make install DESTDIR=**输出目录的绝对路径**

进入输出路径：应当会有：



将输出的所有目录压缩

**tar -czvf tpm\_dev.taf.gz \***

**2.3 编译trousers-0.3.14**

环境变量设置：

进入目录：

**source /opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/environment-setup-arm-linux-gnueabi**

**若configure没有权限**

**chmod a+x ./configure**

**./configure --host=arm-linux-gnueabi**

**make**

**若提示找不着openssl相关的头文件**

**执行：**

./configure –with-openssl=/home/zc/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/arm-linux-gnueabi/libc/i --host=arm-linux-gnueabi

再：make

make install DESTDIR=**输出的绝对路径**

进入输出路径：

将输出的 usr 压缩

**tar –czvf tpm\_dev.tar.gz usr**

**2.4 编译tpm-tools**

进入tpm-tools-1.3.8 路径执行以下指令：

**source /opt/gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabi/environment-setup-arm-linux-gnueabi**

**./configure --host=arm-linux-gnueabi**

**make**

**make install DESTDIR=输出的绝对路径**

进入输出路径：

将输出的 usr 压缩

**tar –czvf tpm\_tools.tar.gz usr**

1. **运行**

将2中编译的四个包：

tpm-tools.tar.gz

gmp-1.6.2.tar.gz

tss1\_tar.gz

tpm\_dev.tar.gz

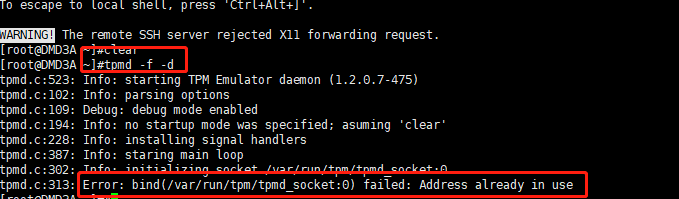
使用winscp传输进板子，并全部解压至根目录

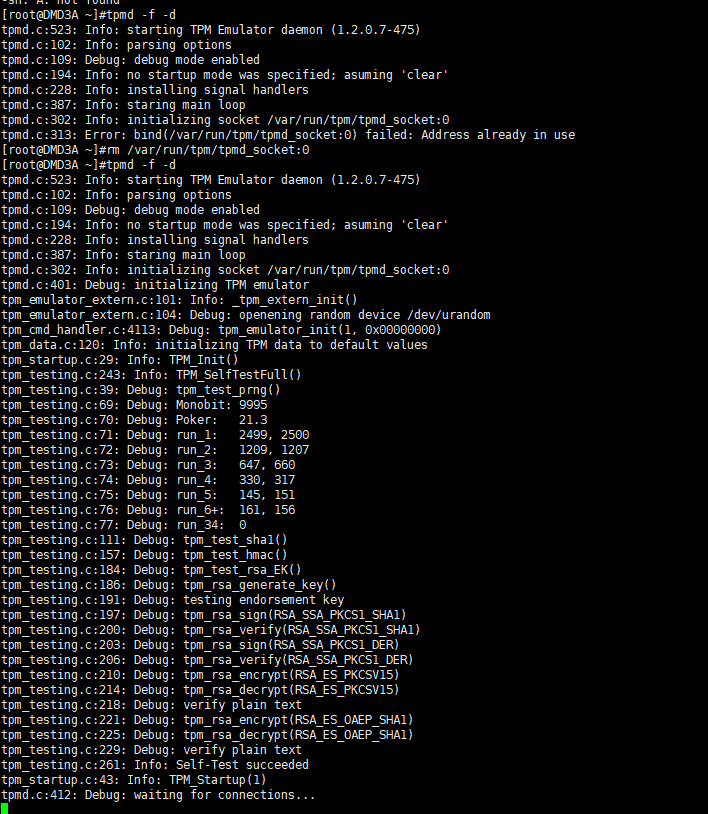
1. 加载tpmd\_dev

**find ./ -name tpmd\_dev**

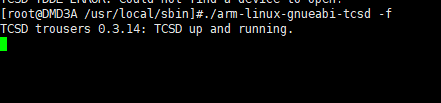
tpmd\_dev保存路径： ./lib/modules/3.10.101/extra/tpmd\_dev.ko

**insmode tpmd\_dev**



**tpmd -f –d 启动模块**

**tcsd -e –f 启动TPM TSS协议栈**



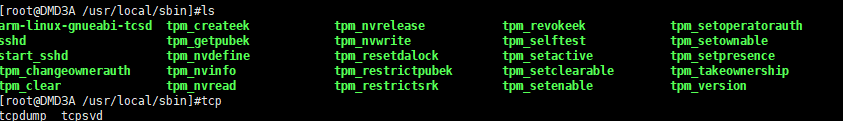
**可以运行tpm相关的指令**

**vi /etc/hosts**

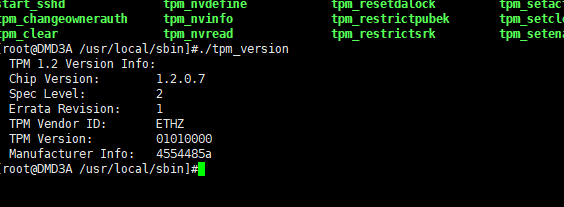
**写入以下值：**

**127.0.0.1 localhost**

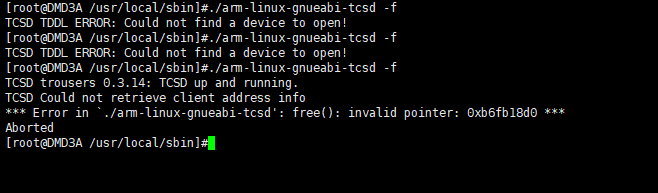
TPM命令：



运行：



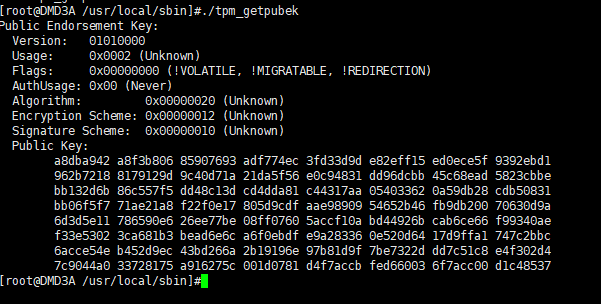
此时会报出一个bug 还未解决：



再次启动，tcsd

./arm-linux-gnueabi-tcsd -f –d

运行：**./tpm\_getpubek 获得公钥**



此时TPM1的模拟环境基本已经搭建完成，tpm设备位置 /dev/tpm

Tpm协议栈以及tpm-tools均可正常工作。