MySQL交叉编译说明

1. 环境说明

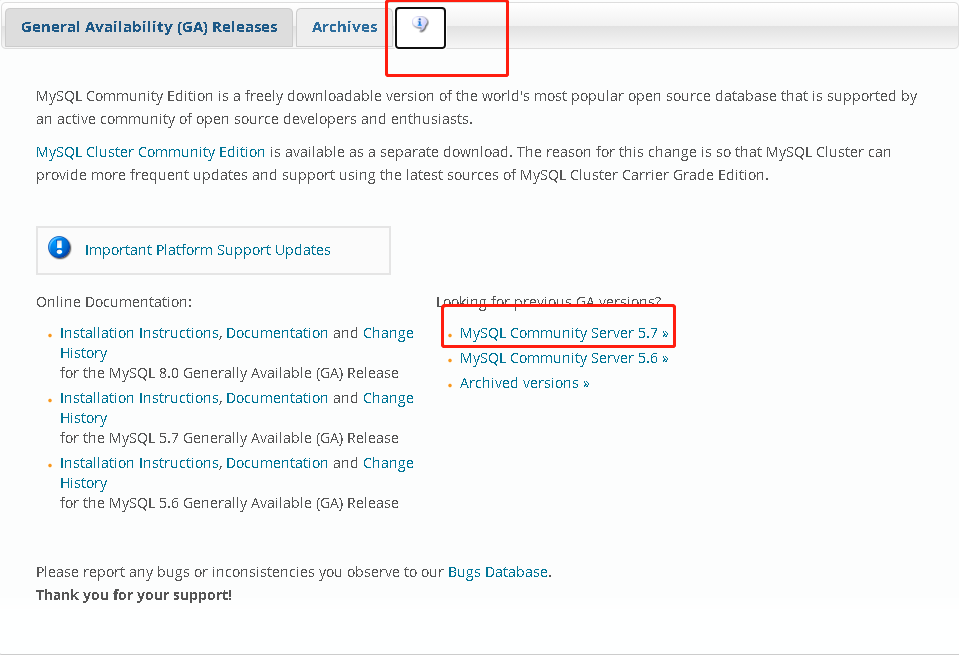
**主机平台： UBUNTU18.04  
硬件平台：力合A9主控板（HSD6405）  
内核版本 ：linux-4.1.15  
交叉编译链：arm-poky-linux-gnueabi**

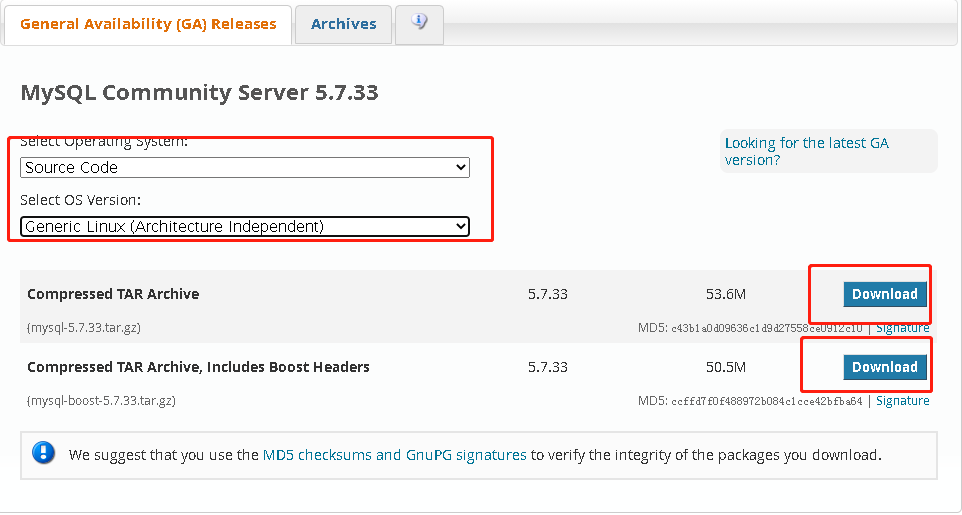
**工具链：fsl-imx-x11-glibc-x86\_64-meta-toolchain-qt5-cortexa9hf-neon-toolchain-4.1.15-2.1.0.sh**

1. 下载源码

官网下载 MySql 源码包：<https://dev.mysql.com/downloads/mysql/，此处下载5.7.29>版本

Github官方地址：<https://github.com/mysql/mysql-server/tree/5.7>





注：最好选择mysql-boost-5.7.33.tar.gz版本，编译需要boost源码。 Mysql编译依赖ncurses，openssl库，但是imx6的工具链已包含，此处无需进行交叉编译。

1. 本机编译

MySql交叉编译需要用到本机编译的一些文件

1) extra/comp\_err

2) scripts/comp\_sql

3) sql/gen\_lex\_hash

4) sql/gen\_lex\_token

5) extra/protobuf/protoc

6) libmysql/libmysql\_api\_test

解压MySql

$ tar zxvf mysql-boost-5.7.29.tar.gz  
$ cd mysql-5.7.29

配置MySql生成makefile

$ mkdir x86build

$ cd x86build

$ cmake ../ -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DMYSQL\_DATADIR=/usr/local/mysql/data \

-DWITH\_BOOST= /home/cftc/Downloads/mysql5-9/mysql-community-5.7.29-1.el7.src/mysql-5.7.29 /boost\_1\_59\_0 \

-DSYSCONFDIR=/etc \

-DEFAULT\_CHARSET=utf8mb4 \

-DDEFAULT\_COLLATION=utf8mb4\_general\_ci \

-DENABLED\_LOCAL\_INFILE=1

其中尤其注意 boost 路径：-DWITH\_BOOST=上面记录下来的路径，以下为其他配置：  
-DCMAKE\_INSTALL\_PREFIX：安装路径；  
-DMYSQL\_DATADIR：数据存放目录；  
-DWITH\_BOOST：boost源码路径；  
-DSYSCONFDIR：my.cnf配置文件目录；  
-DEFAULT\_CHARSET：数据库默认字符编码；  
-DDEFAULT\_COLLATION：默认排序规则；  
-DENABLED\_LOCAL\_INFILE：允许从本文件导入数据；  
-DEXTRA\_CHARSETS：安装所有字符集；

make #可以不install

1. 交叉编译

参考链接：<https://blog.csdn.net/changqing1990/article/details/83027384>

<https://blog.csdn.net/yangxiangzhan/article/details/105059380>

编译过程出现unsported platform错误以及X86工具引用问题参考链接：

https://github.com/jwzl/mysql-server/commit/4baad6dc0075f24c79acb08922c0c4eb6aa8211e

**主要参考链接1进行操作。**

* 交叉编译通用配置

cmake 要做交叉编译，必须要有一个用于指定交叉编译链的配置文件，根据cmake 的官方描述，该内容大致如下demo：

################################################

####### General crosscompile toolchain file

################################################

SET(CMAKE\_SYSTEM\_NAME Linux)

SET(CMAKE\_SYSTEM\_PROCESSOR arm)

SET(CMAKE\_CROSSCOMPILING 1)

SET(CMAKE\_SYSROOT **/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi**)

SET(**CMAKE\_FIND\_ROOT\_PATH /home/cftc/Downloads/mysql5-9/mysql-community-5.7.29-1.el7.src/mysql-5.7.29 /opt/fsl-imx-x11/4.1.15-2.1.0**)

#LINK\_DIRECTORIES(/opt/fsl-imx-x11/4.1.15-2.0.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi)

SET(CMAKE\_C\_COMPILER arm-poky-linux-gnueabi-gcc)

SET(CMAKE\_CXX\_COMPILER arm-poky-linux-gnueabi-g++)

SET(CMAKE\_C\_COMPILER\_ARG1 **"-march=armv7-a -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a9 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi**")

SET(CMAKE\_CXX\_COMPILER\_ARG1 "**-march=armv7-a -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a9 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi**")

SET(CMAKE\_EXE\_LINKER\_FLAGS "${CMAKE\_EXE\_LINKER\_FLAGS} **--sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi**")

SET(CMAKE\_MODULE\_LINKER\_FLAGS "${CMAKE\_EXE\_LINKER\_FLAGS} **--sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa9hf-neon-poky-linux-gnueabi**")

SET(CMAKE\_FIND\_ROOT\_PATH\_MODE\_PROGRAM NEVER)

SET(CMAKE\_FIND\_ROOT\_PATH\_MODE\_LIBRARY ONLY)

SET(CMAKE\_FIND\_ROOT\_PATH\_MODE\_INCLUDE ONLY)

SET(CMAKE\_FIND\_ROOT\_PATH\_MODE\_PACKAGE ONLY)

**注意和自身编译器参数匹配！！！！！**

* mysql相关配置

################################################################

####### The special configuration MACRO for mysql

################################################################

SET(WITH\_UNIT\_TESTS OFF) ## disable unit\_test

SET(DEFAULT\_CHARSET utf8) ## the default charset

SET(DEFAULT\_COLLATION utf8\_general\_ci)

SET(EXTRA\_CHARSETS all) ## Which extra character sets to include:

SET(STACK\_DIRECTION 1) ## Stack grows direction

## If LOAD DATA LOCAL is disabled, either in the server or the client, a client that attempts

## to issue such a statement receives the following error message:

## ERROR 1148: The used command is not allowed with this MySQL version

SET(ENABLED\_LOCAL\_INFILE 1)

#SET(WITH\_EMBEDDED\_SERVER TRUE)

SET(HAVE\_LLVM\_LIBCPP OFF)

SET(HAVE\_FALLOC\_PUNCH\_HOLE\_AND\_KEEP\_SIZE 1)

SET(HAVE\_IB\_GCC\_SYNC\_SYNCHRONISE 1)

SET(HAVE\_IB\_GCC\_ATOMIC\_THREAD\_FENCE 1)

SET(HAVE\_IB\_GCC\_ATOMIC\_COMPARE\_EXCHANGE 1)

SET(HAVE\_IB\_ATOMIC\_PTHREAD\_T\_GCC 1)

SET(HAVE\_IB\_LINUX\_FUTEX 1)

SET(HAVE\_GCC\_ATOMICS\_WITH\_ARCH\_FLAG 1)

SET(HAVE\_FUNC\_IN\_CXX 1)

SET(HAVE\_\_\_BUILTIN\_FFS 1)

IF(CMAKE\_COMPILER\_IS\_GNUCC)

SET(HAVE\_C\_FLOATING\_POINT\_FUSED\_MADD 1)

ENDIF()

IF(CMAKE\_COMPILER\_IS\_GNUCXX)

SET(HAVE\_CXX\_FLOATING\_POINT\_FUSED\_MADD 1)

ENDIF()

IF(CMAKE\_COMPILER\_IS\_GNUCC)

SET(HAVE\_C\_SHIFT\_OR\_OPTIMIZATION\_BUG 1)

ENDIF()

IF(CMAKE\_COMPILER\_IS\_GNUCXX)

SET(HAVE\_CXX\_SHIFT\_OR\_OPTIMIZATION\_BUG 1)

ENDIF()

该部分和通用配置组合成imx.cmake文件，作为CMAKE配置工程时的参数输入。

* 交叉编译的其他修正

前面提到过，交叉编译时需要把6个中间工具copy 到对应的目录，并touch它，然后进行编译。这样比较麻烦的是，当你make clean 后，有得重新copy && touch 它。这是比较麻烦的，为了修改这一点，我们直接搜索定义这个命令的CMakeList.txt.

libmysql/CMakeLists.txt。

# Verify that libmysql\_api\_test runs OK

ADD\_CUSTOM\_COMMAND(TARGET libmysql\_api\_test POST\_BUILD

COMMAND libmysql\_api\_test ## 改为 /usr/bin/libmysql\_api\_test

> ${CMAKE\_CURRENT\_BINARY\_DIR}/libmysql\_api\_test.out

)

将命令改到/usr/bin下，这样就不用每次clean后都都copy && touch了。我们只要把X86 版本的6个命令都copy到/usr/bin/下，然后一一按照上面的逻辑修改这6个命令CMakeList.txt。

$ cp extra/comp\_err /usr/bin/

$ cp scripts/comp\_sql /usr/bin/

$ cp sql/gen\_lex\_hash /usr/bin/

$ cp sql/gen\_lex\_token /usr/bin/

$ cp extra/protobuf/protoc /usr/bin/

$ cp libmysql/libmysql\_api\_test /usr/bin/

* 编译

$ mkdir armbuild

$ cd armbuild

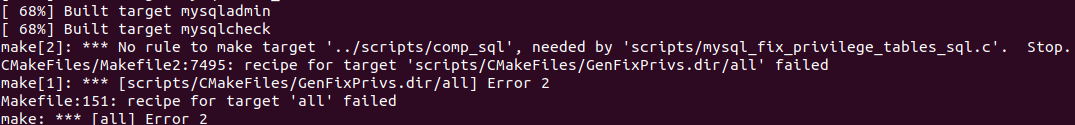
$ cmake ../ -DENABLE\_DOWNLOADS=1 -DDOWNLOAD\_BOOST=1 -DWITH\_BOOST=../boost\_1\_59\_0 -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DCMAKE\_TOOLCHAIN\_FILE=../imx.cmake

**注意：可能到最后会提示Configuring incomplete, errors occurred!，再进行一次cmake以上命令就行了。**

make

sudo make install

如果如下错误：



cp ../x86build/scripts/\* ./scripts/ -r

1. 目标平台安装、运行

* 安装部署

cd /usr/local/

tar czvf mysql.tar.gz /mysql

将生成的压缩包mysql.tar.gz通过FTP工具拷贝至HSD6405主板上，并解压。

tar xzvf /root/mysql.tar.gz -C /usr/local/mysql

* 配置运行

#!/bin/bash

MYSQL\_BASE=/usr/local/mysql

echo "Setup the mysql-5.7.23....."

echo "Create the data path ${MYSQL\_BASE}/data..."

mkdir -p ${MYSQL\_BASE}/data

echo "Initailize the data directory and create the MySQL grant tables"

${MYSQL\_BASE}/bin/mysqld --user=root --basedir=${MYSQL\_BASE} --datadir=${MYSQL\_BASE}/data --initialize-insecure ##--initialize

echo "copy mysqld config file to /etc/ "

cp my.cnf /etc/my.cnf

echo "Create link to /usr/bin"

ln -s ${MYSQL\_BASE}/bin/mysqld /usr/bin/mysqld

ln -s ${MYSQL\_BASE}/bin/mysql /usr/bin/mysql

ln -s ${MYSQL\_BASE}/bin/my\_print\_defaults /usr/bin/my\_print\_defaults

ln -s ${MYSQL\_BASE}/bin/mysqladmin /usr/bin/mysqladmin

ln -s ${MYSQL\_BASE}/bin/mysqldump /usr/bin/mysqldump

echo "Starting and Stopping MySQL Automatically.... "

cp ${MYSQL\_BASE}/support-files/mysql.server /etc

echo "No passwd for mysql, you can use the following command to login: "

echo "mysql -u root --skip-password "

echo "[Done]"

exit 0;

脚本中涉及到my.conf内容如下：

########################################################

###### mysql configure file

########################################################

[mysqld]

user=root

port=3306

datadir=/usr/local/mysql/data

basedir=/usr/local/mysql

socket=/tmp/mysql.sock

pid-file=/var/run/mysqld.pid

general-log-file=/var/log/mysqld.log

slow-query-log-file=/var/log/mysqld.log

log-error=/var/log/mysqld.log

[client]

port=3306

socket=/tmp/mysql.sock

* 测试

启动mysql：

/etc/mysql.server start

通过shell窗口执行mysql，可以实现无密码登录。

修改密码：

mysqladmin –u root password cx125

打开远程登录：

use mysql;

update user set host = '%' where user = 'root';

FLUSH PRIVILEGES;

此时可能出现本地连接不可用的情况，

use mysql;

delete from user where User is NULL;

或者直接上位机工具连接后，手工删除mysql数据user表中的其他用户，只保留%用户。