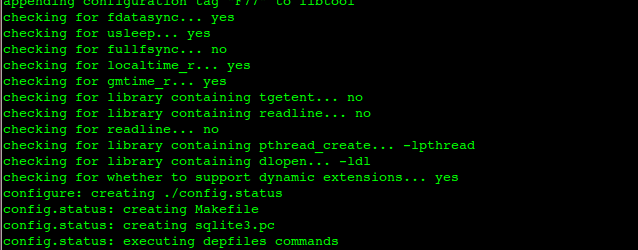
**SQLite向arm-linux的移植**

SQLite是一个开源的嵌入式关系数据库，我们用它来管理从M0端接收的数据，将检测到的仓库信息存储起来。

1. SQLite环境的建立：
2. 首先从下载最新版本的sqlite源码包,我下载的是sqlite-amalgamation-3.7.3.tar.gz，并将下载的文件解压。
3. 在解压生成的 sqlite-3.7.3 目录下执行./configure--host=arm-none-linux-gnueabi--prefix=/home/linux/project/sqlite-arm，其中host为设置交叉编译器，prefix为设置存放编译生成文件的目录，sqlite-arm为跟sqlite-3.7.3同级的目录。
4. 

|  |
| --- |
| arm-none-linux-gnueabi: 编译器的版本要和内核版本一致，所以要根据自己的情况来决定。  /home/linux/project/sqlite-arm: 路径可根据自己的需要来修改。 |

1. 执行上步操作之后会生成Makefile,打开Makefile文件，找到-DPACKAGE\_STRING=\"sqlite\ 3.7.6.2\"处，把3.7.6.2前面的“\”和空格删除。（我使用的这个版本有个错误，其他的版本不一定有），然后直接执行make（要是这样meke有错的话，可以在make之前先make clean一下，然后在make），生成可执行文件，然后再执行make install，配置环境变量。sqlite-arm目录下会生成bin、include、share和lib四个目录。
2. SQLite数据库的移植：
3. 去掉/home/linux/project/sqlite-arm/bin目录下的sqlite3的调示信息（瘦身）：arm-none-linux-gnueabi-strip sqlite3。（为了在PC环境下调试，最好备份一下sqlite3，一下瘦身的文件同理）
4. 将sqlite3下载到开发板的/usr/bin目录：

cp sqlite3 /source/rootfs/usr/bin

3、在/home/linux/project/sqlite-arm/lib 中找到libsqlite3.so.0.8.6这个库文件，去掉调示信息后将libsqlite3.so.0.8.6拷贝到开发板目录/usr**/lib**下，在开发板目录/usr**/lib**下创建一个符号链接：ln -s libsqlite3.so.0.8.6 libsqlite3.so.0。这样你交叉编译的程序就可以使用sqlite3数据库了。

指令如下：

在/home/linux/project/sqlite-arm/lib 下：

arm-none-linux-gnueabi-strip libsqlite3.so.0.8.6

cp libsqlite3.so.0.8.6 /source/rootfs/usr/lib

在开发板目录/usr/lib下：

ln -s libsqlite3.so.0.8.6 libsqlite3.so.0

注：libsqlite3.so，libsqlite3.so.0是链接到libsqlite3.so.0.8.6的.

三、SQLite的测试：

开发板上输入：sqlite3。看是否能进入。

查看“简单数据库测试程序”。（不兼容）

测试程序中的重点，回调函数的使用。