

# 嵌入式 linux 系统下交叉编译 tcpdump 源码总结 V1.0.0

文档版本: V1.0.0

更新日期: 2015-01-21

## 版权声明

版权所有©深圳市广和通实业发展有限公司 2013。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 商标申明



为深圳市广和通实业发展有限公司的注册商标，由所有人拥有。

## 版本记录

文档版本	更新日期	说明
V1.0.0	2015-01-21	初始版本

## 适用型号

序号	产品型号	说明

# 目录

1	前言 .....	5
2	交叉编译工具准备 .....	6
2.1	设置交叉编译环境 .....	6
3	Tcpdump 交叉编译: .....	7
3.1	源码下载.....	7
3.2	源码交叉编译 .....	7
3.2.1	安装 libpcap-1.0.0 .....	7
3.2.2	安装 tcpdump-4.0.0.....	7

# 1 前言

鉴于嵌入式linux下不能使用通用的tcpdump工具，所以就针对嵌入式linux环境总结了一下交叉编译tcpdump源码的方法。

## 2 交叉编译工具准备

### 2.1 设置交叉编译环境

找到对应嵌入式 linux 系统使用的交叉编译工具，即 GCC 工具。

本文的工具是直接找客户要的：toolchains.zip

解压后目录为：

```
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ls
cross.install gcc-3.4.3-uClibc-0.9.28 gcc-3.4.3-uClibc-0.9.28.tar.bz2 lib.stdclib.tgz lib.uClibc.tgz
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ls gcc-3.4.3-uClibc-0.9.28
arm-hisi-linux bin build-time include info lib libexec man share uClibc++ usr
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ls gcc-3.4.3-uClibc-0.9.28/bin/
arm-hisi-linux-addr2line arm-hisi-linux-c++filt arm-hisi-linux-gcc-3.4.3 arm-hisi-linux-gdbtui arm-hisi-linux-objcopy arm-hisi-linux-size
arm-hisi-linux-ar arm-hisi-linux-cpp arm-hisi-linux-gccbug arm-hisi-linux-g++-uc arm-hisi-linux-objdump arm-hisi-linux-strings
arm-hisi-linux-as arm-hisi-linux-g++ arm-hisi-linux-gcov arm-hisi-linux-ld arm-hisi-linux-ranlib arm-hisi-linux-strip
arm-hisi-linux-c++ arm-hisi-linux-gcc arm-hisi-linux-gdb arm-hisi-linux-nm arm-hisi-linux-readelf
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$
```

其中 gcc-3.4.3-uClibc-0.9.28 即为交叉编译工具，查看里面的 bin 目录即为最终使用的工具名称前缀为：arm-hisi-linux-

现在需要设置编译环境

编辑 /etc/bash.bashrc 文件，在文件的末尾添加下述语句：

```
printf "%s: command not found\n" "$1" >&2
return 127

fi

}

fi
Export PATH=$PATH:/home/ght/liuqf/Patch/arm-linux/toolchains/toolchains/gcc-3.4.3-uClibc-0.9.28/bin
```

保存退出后，执行 source /etc/bash.bashrc

```
arm-hisi-linux bin build-time include info lib libexec man share uClibc++ usr
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ vim /etc/bash.bashrc
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ source /etc/bash.bashrc
```

执行完后，在终端输入 ar 再按两次 Tab 键补全命令，发现有 arm-hisi-linux- 前缀的信息，即表示交叉编译环境已经设置成功

```
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ls
cross.install gcc-3.4.3-uClibc-0.9.28 gcc-3.4.3-uClibc-0.9.28.tar.bz2 lib.stdclib.tgz lib.uClibc.tgz
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ls gcc-3.4.3-uClibc-0.9.28
arm-hisi-linux bin build-time include info lib libexec man share uClibc++ usr
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ vim /etc/bash.bashrc
ght@fibocom:~/liuqf/Patch/arm-linux/toolchains/toolchains$ ar
ar
arch
arecord
arecordmidi
arm2hpd1
arm-hisi-linux-addr2line
arm-hisi-linux-ar
arm-hisi-linux-as
arm-hisi-linux-c++
arm-hisi-linux-c++filt
arm-hisi-linux-cpp
arm-hisi-linux-g++
arm-hisi-linux-gcc
arm-hisi-linux-gcc-3.4.3
arm-hisi-linux-gccbug
arm-hisi-linux-gcov
arm-hisi-linux-gdb
arm-hisi-linux-gdbtui
arm-hisi-linux-g++-uc
arm-hisi-linux-ld
arm-hisi-linux-nm
arm-hisi-linux-objcopy
arm-hisi-linux-objdump
arm-hisi-linux-ranlib
arm-hisi-linux-readelf
arm-hisi-linux-size
arm-hisi-linux-strings
arm-hisi-linux-strip
arp
arpd
arping
```

## 3 Tcpdump 交叉编译:

### 3.1 源码下载.

在 <http://www.tcpdump.org> 下载 libpcap-1.0.0.tar.gz 和 tcpdump-4.0.0.tar.gz 两个文件。

### 3.2 源码交叉编译

#### 3.2.1 安装 libpcap-1.0.0

进入 libpcap 目录，打开 configure。将下面两端代码注释掉

```
#if test -z "$with_pcap" && test "$cross_compiling" = yes; then
# { { echo "$as_me:$LINENO: error: pcap type not determined when cross-compiling; use
--with-pcap=..." >&5
# echo "$as_me: error: pcap type not determined when cross-compiling; use --with-pcap=..." >&2;}
# { (exit 1); exit 1; }; }
#fi

.....
# if test $ac_cv_linux_vers = unknown ; then
# { { echo "$as_me:$LINENO: error: cannot determine linux version when cross-compiling" >&5
# echo "$as_me: error: cannot determine linux version when cross-compiling" >&2;}
# { (exit 1); exit 1; }; }
# fi
```

运行 `./configure --host=arm-linux CC=arm-hisi-linux-gcc --prefix=/xxx/libpcap-1.0.0/_INSTALL`

(xxx 为 libpcap-1.0.0 所有目录，请根据实际情况添加，\_\_INSTALL 目录为手动创建的，用来保存交叉编译后的文件)

(如果不注释掉上面两段代码，可能会出现 determine linux version when cross-compiling 或 pcap type not determined when cross-compiling 导致无法 configure)。

配置之后，会生成 Makefile。打开 Makefile 发现 CC= arm-hisi-linux-gcc，说明交叉编译配置成功。

然后 make，make install，执行 make install 后\_\_INSTALL 目录将生成所需要的编译文件

#### 3.2.2 安装 tcpdump-4.0.0

(1) 进入 tcpdump 目录，打开 configure，将下面一段代码注释掉

```
# if test $ac_cv_linux_vers = unknown ; then
# { { echo "$as_me:$LINENO: error: cannot determine linux version when
cross-compiling" >&5
# echo "$as_me: error: cannot determine linux version when cross-compiling" >&2;}
# { (exit 1); exit 1; }; }
# fi
```

运行 `./configure --host=arm-linux CC=arm-hisi-linux-gcc --prefix=/xxx/tcpdump-4.0.0/_INSTALL`

(xxx 为 libpcap-1.0.0 所有目录，请根据实际情况添加，\_\_INSTALL 目录为手动创建的)

然后 make, make install。在\_\_INSTALL 目录下的 sbin 目录有个 tcpdump 的二进制文件，这个就是交叉编译成功的 tcpdump，将此文件拷贝至嵌入式 linux 系统执行 tcpdump -h 命令，输出帮助命令即表示编译成功，如图：

```
tcpdump version 4.6.2
libpcap version 1.6.2
OpenSSL 1.0.1 14 Mar 2012
Usage: tcpdump [-aAbdDefhHIJKLlNOpqRStuUvxxX#] [-B size] [-c count]
               [-C file_size] [-E algo:secret] [-F file] [-G seconds]
               [-i interface] [-j tstamptype] [-M secret] [--number]
               [-Q in|out|inout]
               [-r file] [-s snaplen] [--time-stamp-precision precision]
               [-T type] [--version] [-V file]
               [-w file] [-W filecount] [-y datalinktype] [-z command]
               [-Z user] [expression]
```