# Zebos编译

以下是在ubuntu12.04实验并编译通过（理论上适用ubuntu10.04）

## 安装编译需要的工具

注：./configure的操作已经被加到build.sh、build\_platform.sh脚本中，不需要自己手动执行。 可以直接调到H20RN181编译章节。

如果在执行build.sh、build\_platform.sh脚本过程中出现下面提示的问题，请参照解决。

正常情况下使用svn上下载的ubuntu10.04不会出现下面的问题；

1. ./configure 报错

configure: error: GNU awk is required for lib/memtype.h made by memtypes.awk.

BSD awk complains: awk: gensub doesn't support backreferences (subst "\1")

|  |
| --- |
| $ sudo apt-get install gawk |

2．configure: error: vtysh needs libreadline but was not found and usable on your system.

|  |
| --- |
| $ sudo apt-get install libreadline5 libreadline5-dev |

如果readline5装不了，装readline6

|  |
| --- |
| $ sudo apt-get install libreadline6 libreadline6-dev |

3.WARNING: `automake-1.11' is missing on your system.  You should only need it if

         you modified `Makefile.am', `acinclude.m4' or `configure.ac'.

         You might want to install the `Automake' and `Perl' packages.

         Grab them from any GNU archive site.

|  |
| --- |
| $ sudo apt-get install automake perl |

## 编译配置

注： build.sh、build\_platform.sh脚本，已经传到svn库上，build.sh不需要修改，可以直接使用；build\_platform.sh需要根据实际项目调整一下编译器路径；

build\_platform.sh仅用来编译平台代码，多用于本地编译；可以在build\_ platform.sh所在路径直接执行；

build.sh用于统一编译时，编译平台代码，需要从项目的根目录通过工具调用，不能在build.sh所在路径直接执行；

添加build.sh脚本

#!/bin/bash

export PATH=$PATH:/home/kevin/Documents/mips-tools/usr/bin:/home/kevin/Documents/mips-tools/bin

mkdir Output

./configure \

CC=mips\_4KC-gcc \

--host=mipsel-linux \

--prefix=/home/huahuan/zebra/Output \

--enable-usr=root \

--enable-group=root

make

make install

configure配置文件解释

--build:执行代码编译的主机，正常的话就是你的主机系统。这个参数一般由config.guess来猜就可以。当然自己指定也可以。  
--host:编译出来的二进制程序所执行的主机，因为绝大多数是如果本机编译，本机执行。所以这个值就等于build。只有交叉编译的时候（也就是本机编译，其他系统机器执行）才会build和host不同。用host指定运行主机。  
--target:这个选项只有在建立交叉编译环境的时候用到，正常编译和交叉编译都不会用到。他用build主机上的编译器，编译一个新的编译器（binutils, gcc,gdb等），这个新的编译器将来编译出来的其他程序将运行在target指定的系统上。

--prefix 指定安装目录（这个目录有两个用途，一个是用来存放编译的输出文件，一个是程序在目标板上运行时，这个目录会作为工作目录，所以目标板上需要创建相同的目录）

注：

1.编译时要根据自己的交叉编译目录修改PATH变量和--prefix

2.编译时，缺少了一个宏定义，在/zebos/debra/rt\_netlink.c 516行，为了编译通过临时注掉

//case ARPHRD\_IEEE802154: return ZEBRA\_LLT\_IEEE802154;

## 三、目标板上执行

1.将输出的目录Output整体复制到目标板上，放到前面—prefix指定的目录

需要创建这个目录/home/huahuan/zebra/Output

2. 文件夹下Output/etc有一个zebra.conf.sample文件,需将该文件copy一份

cp zebra.conf.sample zebra.conf

3.执行./zebra -d -u root -g root

该步骤会查找/home/huahuan/zebra/Output/etc下是否有zebra.conf文件，如报错检查—prefix指定的目录与当前目录是否一致

执行完后可以ps看一下是否起来了

4. telnet localhost 2601

默认密码zebra

## 四、H20RN181编译

产品的编译分为kernel，sdk，zebos，product

### 1.使用脚本一次编译所有模块

请参考下面这两个文档

svn://172.16.95.117/project\_manager/项目文件管理/开发部/数通产品通用类/版本文件/IPRAN\_V2.4/compile/IPRAN 代码编译及sys打包方法.doc

svn://172.16.95.117/project\_manager/项目文件管理/开发部/数通产品通用类/版本文件/IPRAN\_V2.4/compile/IPRAN 代码编译脚本说明.doc

### 2.使用脚本单独编译

单独编译时要确定该模块所依赖的模块已经编译

正确的编译顺序为kernel—>sdk—>zebos—> product

这几个模块的编译命令

$ ./hh\_cmd H20RN181 target kernel

$ ./hh\_cmd H20RN181 target driver sdk-xgs-robo-6.4.2 user-mode CARD1610

$ ./hh\_cmd H20RN181 target platform

$ ./hh\_cmd H20RN181 target product

对应的clean命令

$ ./hh\_cmd H20RN181 clean kernel

$ ./hh\_cmd H20RN181 clean driver sdk-xgs-robo-6.4.2

$ ./hh\_cmd H20RN181 clean platform

$ ./hh\_cmd H20RN181 clean product

### 3.单独编译

单独编译时需要通过脚本或命令传递参数

**zebos编译**使用zebos目录下的build\_platform.sh脚本

按需更改build\_platform.sh里面的配置和参数，然后执行./ build\_platform.sh进行编译

**product编译**使用cmake命令，进入project/H20RN181/**product**目录执行下面命令

cmake

-DZEBOS\_PREFIX:STRING=${ZEBOS\_PREPATH} -DZEBOS\_SRC\_DIR:STRING=${ROOT\_DIR}/platform/zebos -DZEBOS\_LIB\_DIR:STRING=${ROOT\_DIR}/platform/zebos/lib -DHSL\_SDK\_SRC\_DIR:STRING=${ROOT\_DIR}/driver/sdk-xgs-robo-6.4.2 -DHSL\_SDK\_LIB\_DIR:STRING=${ROOT\_DIR}/target/driver/sdk-xgs-robo-6.4.2/H20RN181/CARD1610/unix-user/H20PN1610-2\_6 ../../../hsl

${ ZEBOS\_PREPATH }是zebos编译时的prefix目录

${ROOT\_DIR}是顶层目录

执行完cmake后还需要执行命令

make

make install

**hal编译**使用cmake命令，进入project/H20RN181目录执行下面命令

cmake -DZEBOS\_PREFIX:STRING=${ZEBOS\_PREPATH} -DZEBOS\_SRC\_DIR:STRING=${ROOT\_DIR}/platform/zebos -DZEBOS\_LIB\_DIR:STRING=${ROOT\_DIR}/platform/zebos/lib -DHAL\_HSL\_SRC\_DIR:STRING=${ROOT\_DIR}/hsl -DHAL\_SDK\_SRC\_DIR:STRING=${ROOT\_DIR}/driver/sdk-xgs-robo-6.4.2 -DHAL\_SDK\_LIB\_DIR:STRING=${ROOT\_DIR}/target/driver/sdk-xgs-robo-6.4.2/H20RN181/CARD1610/unix-user/H20PN1610-2\_6 -DHAL\_LIB\_ZEBRA=aa -DHAL\_LIB\_HSL=aa -DHAL\_LIB\_HSL\_MUX=aa -DHAL\_LIB\_HSL\_DRIVERS=aa ../../../hal

${ ZEBOS\_PREPATH }是zebos编译时的prefix目录

${ROOT\_DIR}是顶层目录

同样的执行完cmake后还需要执行命令

make

make install

可能会遇到下面的错误：

CMake Error: The following variables are used in this project, but they are set to NOTFOUND.

Please set them or make sure they are set and tested correctly in the CMake files:

HAL\_LIB\_HSL

    linked by target "hal" in directory work/hal HAL\_LIB\_HSL\_DRIVERS

    linked by target "hal" in directory work/hal HAL\_LIB\_HSL\_MUX

    linked by target "hal" in directory work/hal HAL\_LIB\_ZEBRA

    linked by target "hal" in directory work/hal

这个错误是因为在编译hal前没有正确编译zebos或hsl，hal编译时要查找zebos和hsl的库，不要忘记编译玩hsl后需要make install