

浙江大学 实验报告

课程名称： 嵌入式系统 指导老师： 蔡铭 学生姓名： 李磊

实验名称： 任务 37: WRTnode 的交叉编译环境

实验类型： 操作实践 学生学号： 3110102782

一、实验目的和要求

在树莓派或 Acadia 上实现一个 C 语言的交叉编译环境，能编译产生 WRTnode 用的 MIPS 程序。

二、实验内容和原理

在树莓派或 Acadia 上实现一个 C 语言的交叉编译环境，能编译产生 WRTnode 用的 MIPS 程序。

三、主要仪器设备

1. Raspberry Pi 主板一块；
2. 8G SD 卡一块；
3. 5V/1A 电源一个；
4. microUSB 线一根；
5. USB-TTL 串口线一根（PL2303 芯片）；
6. PC（Windows/Mac OS/Linux）一台；
7. 以太网线一根；
8. 路由器一台。

四、操作方法和实验步骤

```
1. 安装 bison、flex、texinfo、libncurses5-dev、bc
$ sudo apt-get update
$ sudo apt-get install bison
$ sudo apt-get install flex
$ sudo apt-get install texinfo
$ sudo apt-get install libncurses5-dev
$ sudo apt-get install bc
```

```
pi@raspberrypi: ~  
忽略 http://mirrordirector.raspbian.org wheezy/main Translation-en  
忽略 http://mirrordirector.raspbian.org wheezy/non-free Translation-zh_CN  
忽略 http://mirrordirector.raspbian.org wheezy/non-free Translation-zh  
忽略 http://mirrordirector.raspbian.org wheezy/non-free Translation-en  
忽略 http://mirrordirector.raspbian.org wheezy/rpi Translation-zh_CN  
忽略 http://mirrordirector.raspbian.org wheezy/rpi Translation-zh  
忽略 http://mirrordirector.raspbian.org wheezy/rpi Translation-en  
下载 7,132 kB, 耗时 3分 0秒 (39.4 kB/s)  
正在读取软件包列表... 完成  
pi@raspberrypi ~ $ sudo apt-get install bison  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
将会安装下列额外的软件包:  
  libbison-dev m4  
建议安装的软件包:  
  bison-doc  
下列【新】软件包将被安装:  
  bison libbison-dev m4  
升级了 0 个软件包, 新安装了 3 个软件包, 要卸载 0 个软件包, 有 54 个软件包未被升级。  
需要下载 1,204 kB 的软件包。  
解压缩后会消耗掉 2,501 kB 的额外空间。  
您希望继续执行吗? [Y/n]
```

```
pi@raspberrypi: ~  
正在处理用于 man-db 的触发器...  
正在设置 m4 (1.4.16-3) ...  
正在设置 libbison-dev:armhf (1:2.5.dfsg-2.1) ...  
正在设置 bison (1:2.5.dfsg-2.1) ...  
update-alternatives: using /usr/bin/bison.yacc to provide /usr/bin/yacc (yacc) in  
n 自动模式  
pi@raspberrypi ~ $ sudo apt-get install flex  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
下列【新】软件包将被安装:  
  flex  
升级了 0 个软件包, 新安装了 1 个软件包, 要卸载 0 个软件包, 有 54 个软件包未被升级。  
需要下载 321 kB 的软件包。  
解压缩后会消耗掉 910 kB 的额外空间。  
获取: 1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main flex armhf 2.5.  
35-10.1 [321 kB]  
下载 321 kB, 耗时 2秒 (134 kB/s)  
Selecting previously unselected package flex.  
(正在读取数据库 ... 系统当前共安装有 78617 个文件和目录。)  
正在解压缩 flex (从 .../flex_2.5.35-10.1_armhf.deb) ...  
正在处理用于 man-db 的触发器...
```

```
pi@raspberrypi: ~  
下载 321 kB, 耗时 2秒 (134 kB/s)  
Selecting previously unselected package flex.  
(正在读取数据库 ... 系统当前共安装有 78617 个文件和目录。)  
正在解压缩 flex (从 .../flex 2.5.35-10.1_armhf.deb) ...  
正在处理用于 man-db 的触发器...  
正在处理用于 install-info 的触发器...  
正在设置 flex (2.5.35-10.1) ...  
pi@raspberrypi ~ $ sudo apt-get install texinfo  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
建议安装的软件包:  
texlive-base texlive-latex-base texlive-generic-recommended  
texinfo-doc-nonfree  
下列【新】软件包将被安装:  
texinfo  
升级了 0 个软件包, 新安装了 1 个软件包, 要卸载 0 个软件包, 有 54 个软件包未被升  
级。  
需要下载 960 kB 的软件包。  
解压缩后会消耗掉 2,379 kB 的额外空间。  
获取: 1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main texinfo armhf 4  
.13a.dfsg.1-10 [960 kB]  
下载 960 kB, 耗时 1秒 (587 kB/s)
```

```
pi@raspberrypi: ~/Desktop/ncurses-5.9  
文件(E) 编辑(E) 标签(T) 帮助(H)  
bin directory: /usr/bin  
lib directory: /usr/lib  
include directory: /usr/include  
man directory: /usr/man  
terminfo directory: /usr/share/terminfo  
pi@raspberrypi ~/Desktop/ncurses-5.9 $ apt-get install libncurses5-dev  
E: 无法打开锁文件 /var/lib/dpkg/lock - open (13: 权限不够)  
E: 无法对状态列表目录加锁(/var/lib/dpkg/), 请查看您是否正以 root 用户运行?  
pi@raspberrypi ~/Desktop/ncurses-5.9 $ sudo apt-get install libncurses5-dev  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
将会安装下列额外的软件包:  
libtinfo-dev  
建议安装的软件包:  
ncurses-doc  
下列【新】软件包将被安装:  
libncurses5-dev libtinfo-dev  
升级了 0 个软件包, 新安装了 2 个软件包, 要卸载 0 个软件包, 有 54 个软件包未被升  
级。  
需要下载 292 kB 的软件包。  
解压缩后会消耗掉 1,110 kB 的额外空间。  
您希望继续执行吗? [Y/n]
```

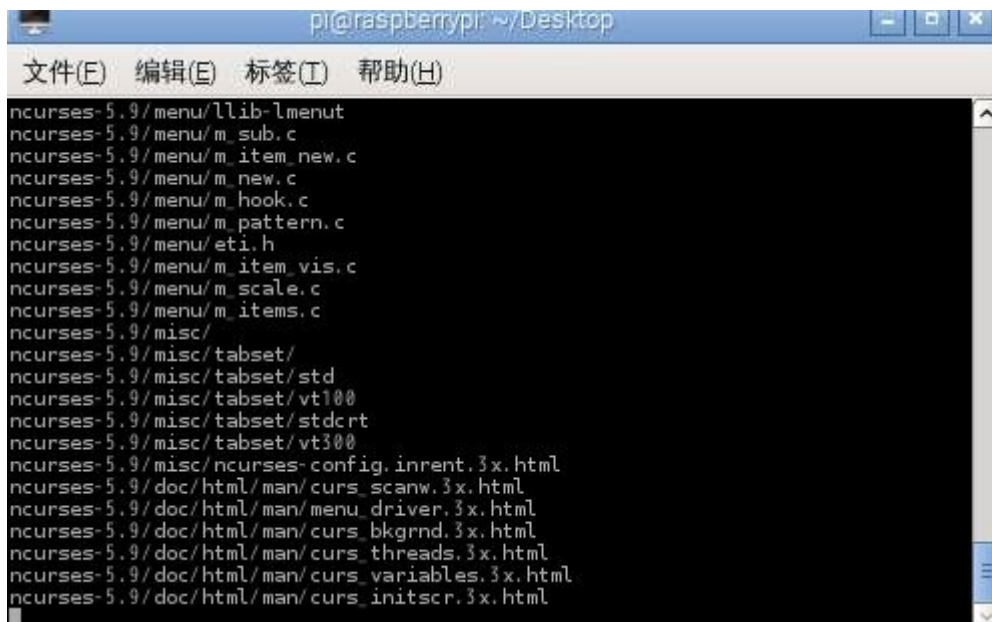


```
pi@raspberrypi: ~/Desktop/buildroot
文件(E) 编辑(E) 标签(T) 帮助(H)
onfig BR2_EXTERNAL=support/dummy-external SKIP_LEGACY= /home/pi/Desktop/buildroot/output/build/buildroot-config/conf --silentoldconfig Config.in
#
# configuration written to /home/pi/Desktop/buildroot/.config
#
make[1]: Leaving directory '/home/pi/Desktop/buildroot'
You must install 'bc' on your build machine
support/dependencies/dependencies.mk:28: recipe for target 'core-dependencies' failed
make: *** [core-dependencies] Error 1
pi@raspberrypi ~/Desktop/buildroot $ sudo apt-get install bc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  bc
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 54 个软件包未被升级。
需要下载 106 kB 的软件包。
解压缩后会消耗掉 257 kB 的额外空间。
获取：1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main bc armhf 1.06.95-2 [106 kB]
下载 106 kB，耗时 1秒 (98.1 kB/s)
```

2. 下载 ncurses 并安装

从 <http://mirrors.ustc.edu.cn/gnu/ncurses/ncurses-5.9.tar.gz> 下载 ncurses-5.9.tar.gz 解包，并安装：

```
$ tar zxvf ncurses-5.9.tar.gz
$ cd ncurses-5.9
```



```
pi@raspberrypi: ~/Desktop
文件(E) 编辑(E) 标签(T) 帮助(H)
ncurses-5.9/menu/llib-lmenut
ncurses-5.9/menu/m_sub.c
ncurses-5.9/menu/m_item_new.c
ncurses-5.9/menu/m_new.c
ncurses-5.9/menu/m_hook.c
ncurses-5.9/menu/m_pattern.c
ncurses-5.9/menu/eti.h
ncurses-5.9/menu/m_item_vis.c
ncurses-5.9/menu/m_scale.c
ncurses-5.9/menu/m_items.c
ncurses-5.9/misc/
ncurses-5.9/misc/tabset/
ncurses-5.9/misc/tabset/std
ncurses-5.9/misc/tabset/vt100
ncurses-5.9/misc/tabset/stdcrt
ncurses-5.9/misc/tabset/vt300
ncurses-5.9/misc/ncurses-config.inrent.3x.html
ncurses-5.9/doc/html/man/curs_scanw.3x.html
ncurses-5.9/doc/html/man/menu_driver.3x.html
ncurses-5.9/doc/html/man/curs_bkgnd.3x.html
ncurses-5.9/doc/html/man/curs_threads.3x.html
ncurses-5.9/doc/html/man/curs_variables.3x.html
ncurses-5.9/doc/html/man/curs_initscr.3x.html
```

进行配置：

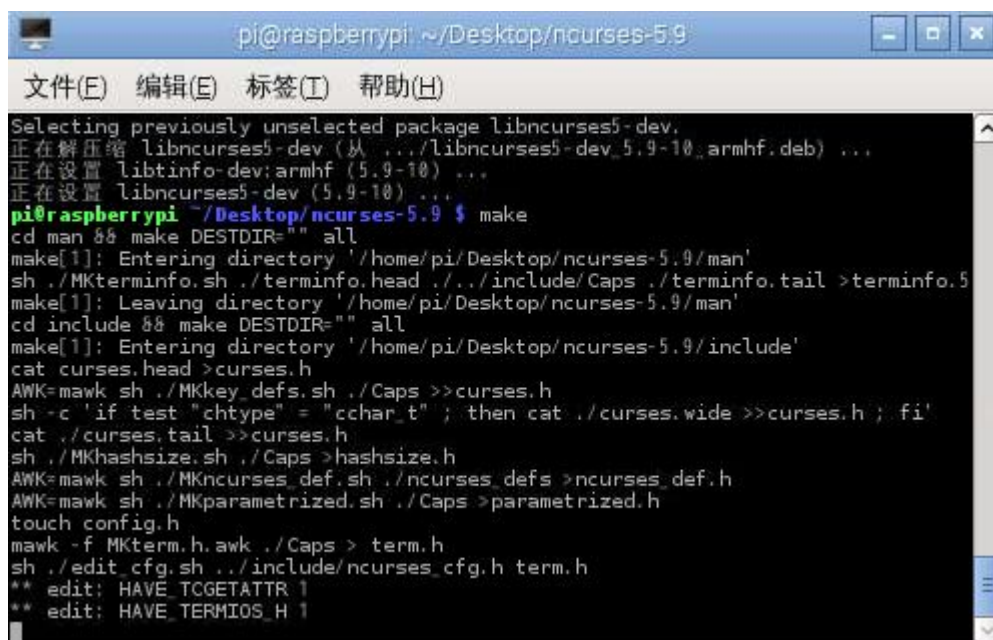
```
$ ./configure
```


A terminal window titled 'pi@raspberrypi: ~/Desktop/ncurses-5.9' with a menu bar containing '文件(E)', '编辑(E)', '标签(T)', and '帮助(H)'. The terminal shows the output of the './configure' command. It lists various configuration options like 'INSTALL', 'install-sh', 'Makefile.in', 'Makefile.os2', 'man', 'mk-2nd.awk', 'mk-dlls.sh.in', 'mk-hdr.awk', 'ncurses', 'NEWS', 'test', and 'T0-D0'. It then proceeds to check for 'egrep', 'grep -E', and 'gcc'. It configures for 'ABI 5' on 'Fri Apr 17 06:46:15 UTC 2015' for 'armv6l-unknown-linux-gnueabi'. It checks for 'prefix' (/usr), 'gcc', 'C compiler default output' (a.out), 'whether the C compiler works' (yes), 'whether we are cross compiling' (no), 'executable suffix' (o), 'object suffix' (o), 'whether we are using the GNU C compiler' (yes), 'whether gcc accepts -g' (yes), 'version of gcc' (4.6.3), 'how to run the C preprocessor' (gcc -E), and 'whether gcc needs -traditional' (no).

```
pi@raspberrypi: ~/Desktop/ncurses-5.9
文件(E) 编辑(E) 标签(T) 帮助(H)
c++          INSTALL          mk-2nd.awk    test
config.guess install-sh      mk-dlls.sh.in T0-D0
config.sub   Makefile.in    mk-hdr.awk
configure    Makefile.os2   ncurses
configure.in man            NEWS
pi@raspberrypi ~/Desktop/ncurses-5.9 $ ./configure
checking for egrep... grep -E
Configuring NCURSES 5.9 ABI 5 (Fri Apr 17 06:46:15 UTC 2015)
checking build system type... armv6l-unknown-linux-gnueabi
checking host system type... armv6l-unknown-linux-gnueabi
checking target system type... armv6l-unknown-linux-gnueabi
Configuring for linux-gnueabi
checking for prefix... /usr
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for executable suffix... o
checking for object suffix... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking version of gcc... 4.6.3
checking how to run the C preprocessor... gcc -E
checking whether gcc needs -traditional...
```

Make:

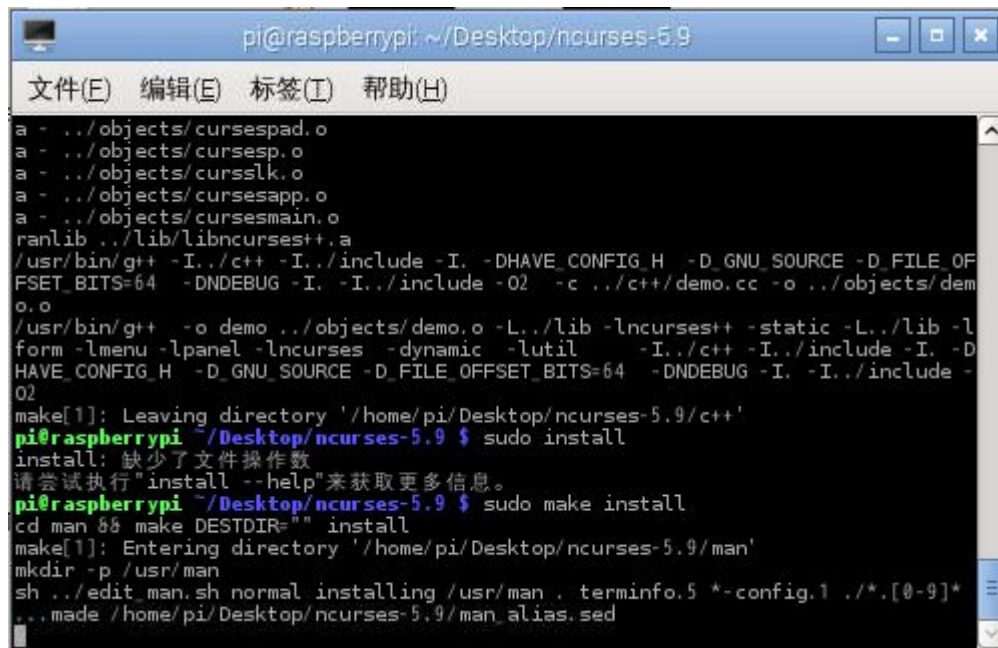
```
$ make
```

A terminal window titled 'pi@raspberrypi: ~/Desktop/ncurses-5.9' with a menu bar containing '文件(E)', '编辑(E)', '标签(T)', and '帮助(H)'. The terminal shows the output of the 'make' command. It starts by selecting previously unselected packages like 'libncurses5-dev', 'libtinfo-dev:armhf (5.9-10)', and 'libncurses5-dev (5.9-10)'. It then enters the 'make' command. The output shows the make process for 'man', 'include', and 'curses.h'. It lists various files and commands like 'MKkey_defs.sh', 'Caps', 'term.h', 'edit_cfg.sh', 'include/ncurses_cfg.h', 'term.h', 'edit: HAVE_TCGETATTR 1', and 'edit: HAVE_TERMIOS_H 1'.

```
pi@raspberrypi ~/Desktop/ncurses-5.9 $ make
cd man && make DESTDIR="" all
make[1]: Entering directory '/home/pi/Desktop/ncurses-5.9/man'
sh ./MKterminfo.sh ./terminfo.head ../include/Caps ./terminfo.tail >terminfo.5
make[1]: Leaving directory '/home/pi/Desktop/ncurses-5.9/man'
cd include && make DESTDIR="" all
make[1]: Entering directory '/home/pi/Desktop/ncurses-5.9/include'
cat curses.head >curses.h
AWK=mawk sh ./MKkey_defs.sh ./Caps >>curses.h
sh -c 'if test "ctype" = "cchar_t" ; then cat ./curses.wide >>curses.h ; fi'
cat ./curses.tail >>curses.h
sh ./MKhashsize.sh ./Caps >hashsize.h
AWK=mawk sh ./MKncurses_def.sh ./ncurses_defs >ncurses_def.h
AWK=mawk sh ./MKparametrized.sh ./Caps >parametrized.h
touch config.h
mawk -f MKterm.h.awk ./Caps > term.h
sh ./edit_cfg.sh ../include/ncurses_cfg.h term.h
** edit: HAVE_TCGETATTR 1
** edit: HAVE_TERMIOS_H 1
```

安装:

```
$ sudo make install
```



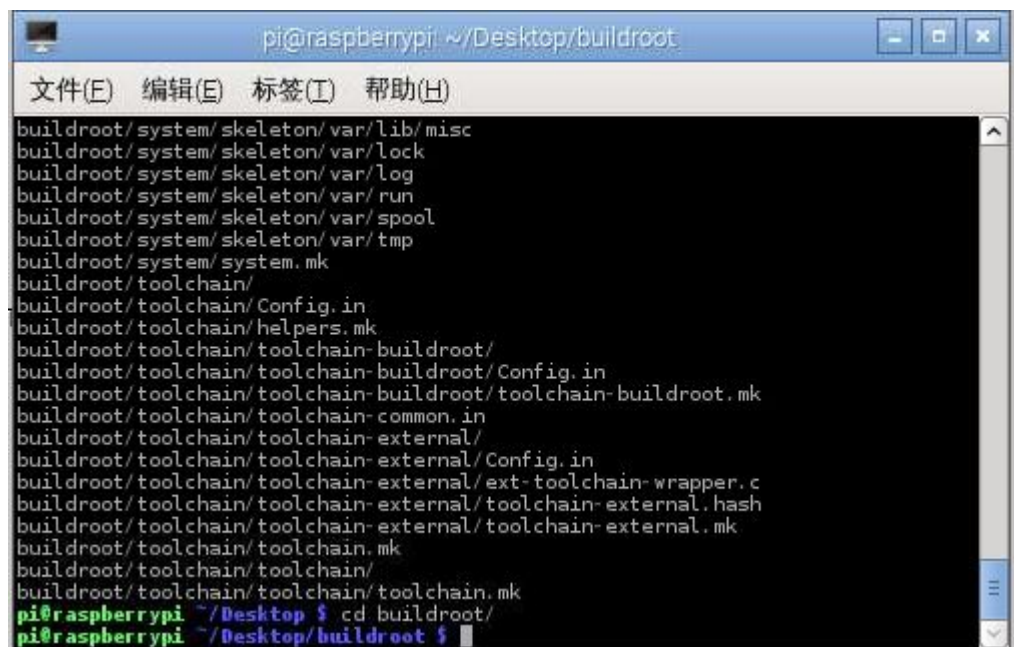
```
pi@raspberrypi: ~/Desktop/ncurses-5.9
文件(E) 编辑(E) 标签(T) 帮助(H)
a - ../objects/cursespad.o
a - ../objects/curses.o
a - ../objects/cursslk.o
a - ../objects/cursesapp.o
a - ../objects/cursesmain.o
ranlib ../lib/libncurses++.a
/usr/bin/g++ -I../c++ -I../include -I. -DHAVE_CONFIG_H -D_GNU_SOURCE -D_FILE_OFFSET_BITS=64 -DNCDEBUG -I. -I../include -O2 -c ../c++/demo.cc -o ../objects/demo.o
/usr/bin/g++ -o demo ../objects/demo.o -L../lib -lncurses++ -static -L../lib -lform -lmenu -lpanel -lncurses -dynamic -lutil -I../c++ -I../include -I. -DHAVE_CONFIG_H -D_GNU_SOURCE -D_FILE_OFFSET_BITS=64 -DNCDEBUG -I. -I../include -O2
make[1]: Leaving directory '/home/pi/Desktop/ncurses-5.9/c++'
pi@raspberrypi ~/Desktop/ncurses-5.9 $ sudo make install
install: 缺少了文件操作数
请尝试执行"install --help"来获取更多信息。
pi@raspberrypi ~/Desktop/ncurses-5.9 $ sudo make install
cd man && make DESTDIR="" install
make[1]: Entering directory '/home/pi/Desktop/ncurses-5.9/man'
mkdir -p /usr/man
sh ../edit_man.sh normal installing /usr/man . terminfo.5 *-config.1 ../.[0-9]*
...made /home/pi/Desktop/ncurses-5.9/man/alias.sed
```

3. 安装配置 buildroot

从 <http://buildroot.uclibc.org/downloads/snapshots/buildroot-snapshot.tar.bz2> 下载 buildroot。

解包 buildroot:

```
$ tar -jxvf buildroot-snapshot.tar.bz2
$ cd buildroot
```



```
pi@raspberrypi: ~/Desktop/buildroot
文件(E) 编辑(E) 标签(T) 帮助(H)
buildroot/system/skeleton/var/lib/misc
buildroot/system/skeleton/var/lock
buildroot/system/skeleton/var/log
buildroot/system/skeleton/var/run
buildroot/system/skeleton/var/spool
buildroot/system/skeleton/var/tmp
buildroot/system/system.mk
buildroot/toolchain/
buildroot/toolchain/Config.in
buildroot/toolchain/helpers.mk
buildroot/toolchain/toolchain-buildroot/
buildroot/toolchain/toolchain-buildroot/Config.in
buildroot/toolchain/toolchain-buildroot/toolchain-buildroot.mk
buildroot/toolchain/toolchain-common.in
buildroot/toolchain/toolchain-external/
buildroot/toolchain/toolchain-external/Config.in
buildroot/toolchain/toolchain-external/ext-toolchain-wrapper.c
buildroot/toolchain/toolchain-external/toolchain-external.hash
buildroot/toolchain/toolchain-external/toolchain-external.mk
buildroot/toolchain/toolchain.mk
buildroot/toolchain/toolchain/
buildroot/toolchain/toolchain/toolchain.mk
pi@raspberrypi ~/Desktop $ cd buildroot/
pi@raspberrypi ~/Desktop/buildroot $
```

配置 buildroot:

```
$ make clean
$ make menuconfig
```

```
pi@raspberrypi: ~/Desktop/buildroot
文件(E) 编辑(E) 标签(T) 帮助(H)

| hello hello - 副本.c ncurses-5.9.tar.gz
| hello.c lab3 Test.class
| buildroot snapshot.tar.bz2 hellopi ncurses-5.9 Test.java
pi@raspberrypi ~/Desktop $ cd buildroot/
pi@raspberrypi ~/Desktop/buildroot $ make clean
rm -rf /home/pi/Desktop/buildroot/output/target /home/pi/Desktop/buildroot/output
t/images /home/pi/Desktop/buildroot/output/host \
/home/pi/Desktop/buildroot/output/build /home/pi/Desktop/buildroot/output
t/staging \
/home/pi/Desktop/buildroot/output/legal-info /home/pi/Desktop/buildroot/
output/graphs
pi@raspberrypi ~/Desktop/buildroot $ make menuconfig
mkdir -p /home/pi/Desktop/buildroot/output/build/buildroot-config/lxdialog
PKG_CONFIG_PATH="" make CC="/usr/bin/gcc" HOSTCC="/usr/bin/gcc" \
obj=/home/pi/Desktop/buildroot/output/build/buildroot-config -C support/kcon
fig -f Makefile.br mconf
make[1]: Entering directory '/home/pi/Desktop/buildroot/support/kconfig'
/usr/bin/gcc -DCURSES_LOC="" -DLOCALE -I/home/pi/Desktop/buildroot
/output/build/buildroot-config -DCONFIG_="" -MM *.c > /home/pi/Desktop/buildr
oot/output/build/buildroot-config/.depend 2>/dev/null || :
/usr/bin/gcc -DCURSES_LOC="" -DLOCALE -I/home/pi/Desktop/buildroot
/output/build/buildroot-config -DCONFIG_="" -c conf.c -o /home/pi/Desktop/bu
ildroot/output/build/buildroot-config/conf.o
```

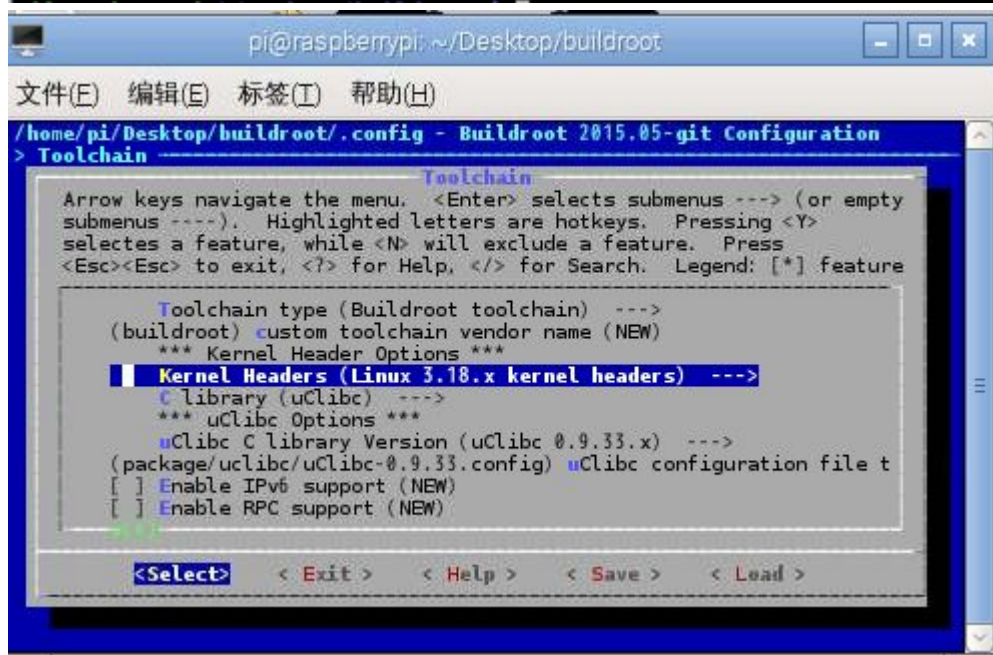


选择“Target Architecture”，改成 MIPS（little endian）



选择“Toolchain”，务必将“Kernel Headers”的 Linux 版本改成主机的 Linux 版本，本机是 3.18.x

```
pi@raspberrypi ~/Desktop/buildroot $ uname -a
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l GNU/Linux
```



进行编译：

```
$ sudo make
```

```
pi@raspberrypi ~/Desktop/buildroot $ sudo make
```




```
pi@raspberrypi: ~/Desktop/buildroot
文件(E) 编辑(E) 标签(T) 帮助(H)
/usr/bin/gcc -DCURSES_LOC="" -DLOCALE -I/home/pi/Desktop/buildroot
/output/build/buildroot-config -DCONFIG="" /home/pi/Desktop/buildroot/output
t/build/buildroot-config/conf.o /home/pi/Desktop/buildroot/output/build/buildroo
t-config/zconf.tab.o -o /home/pi/Desktop/buildroot/output/build/buildroot-confi
g/conf
make[2]: Leaving directory '/home/pi/Desktop/buildroot/support/kconfig'
BR2_DEFCONFIG=' KCONFIG_AUTOCONFIG=/home/pi/Desktop/buildroot/output/build/buil
droot-config/auto.conf KCONFIG_AUTOHEADER=/home/pi/Desktop/buildroot/output/buil
d/buildroot-config/autoconf.h KCONFIG_TRISTATE=/home/pi/Desktop/buildroot/output
/build/buildroot-config/tristate.config BR2_CONFIG=/home/pi/Desktop/buildroot/.c
onfig BR2_EXTERNAL=support/dummy-external SKIP_LEGACY= /home/pi/Desktop/buildroo
t/output/build/buildroot-config/conf --silentoldconfig Config.in
make[1]: Leaving directory '/home/pi/Desktop/buildroot'
>>> host-binutils 2.24 Downloading
--2015-04-17 08:03:15-- http://ftp.gnu.org/pub/gnu/binutils/binutils-2.24.tar.b
z2
正在解析主机 ftp.gnu.org (ftp.gnu.org)... 208.118.235.20, 2001:4830:134:3::b
正在连接 ftp.gnu.org (ftp.gnu.org)[208.118.235.20]:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：22716802 (22M) [application/x-bzip2]
正在保存至："/home/pi/Desktop/buildroot/output/build/.binutils-2.24.tar.bz2.wfRp
KE/output"
57% [=====]> ] 13,121,876 87.4K/s 估时 1m 47s
```

4. 使用编译好的工具链

进入工具链目录：

```
$ cd /home/pi/Desktop/buildroot/host/usr/bin
```

使用工具链进行编译：

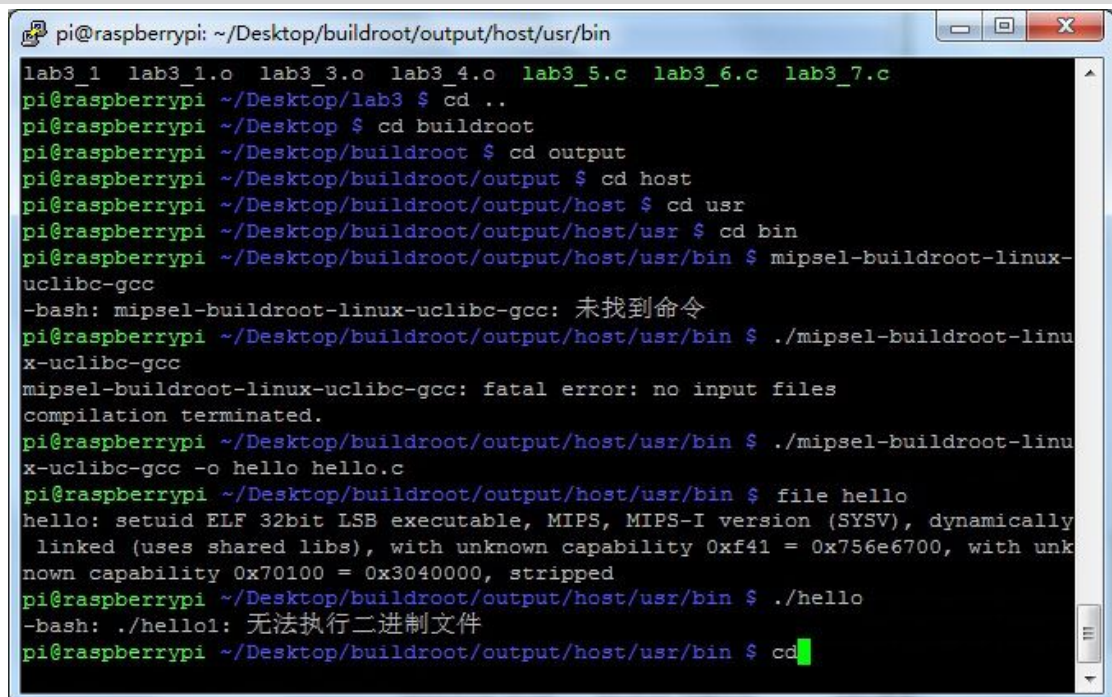
```
$ ./mipsel-buildroot-linux-uclibc-gcc -o hello hello.c
```

查看编译后的版本信息：

```
$ file hello
```

尝试运行：

```
$ ./hello
```



```
pi@raspberrypi: ~/Desktop/buildroot/output/host/usr/bin
lab3_1 lab3_1.o lab3_3.o lab3_4.o lab3_5.c lab3_6.c lab3_7.c
pi@raspberrypi ~/Desktop/lab3 $ cd ..
pi@raspberrypi ~/Desktop $ cd buildroot
pi@raspberrypi ~/Desktop/buildroot $ cd output
pi@raspberrypi ~/Desktop/buildroot/output $ cd host
pi@raspberrypi ~/Desktop/buildroot/output/host $ cd usr
pi@raspberrypi ~/Desktop/buildroot/output/host/usr $ cd bin
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ mipsel-buildroot-linux-
uclibc-gcc
-bash: mipsel-buildroot-linux-uclibc-gcc: 未找到命令
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ ./mipsel-buildroot-linu
x-uclibc-gcc
mipsel-buildroot-linux-uclibc-gcc: fatal error: no input files
compilation terminated.
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ ./mipsel-buildroot-linu
x-uclibc-gcc -o hello hello.c
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ file hello
hello: setuid ELF 32bit LSB executable, MIPS, MIPS-I version (SYSV), dynamically
linked (uses shared libs), with unknown capability 0xf41 = 0x756e6700, with unk
nown capability 0x70100 = 0x3040000, stripped
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ ./hello
-bash: ./hello1: 无法执行二进制文件
pi@raspberrypi ~/Desktop/buildroot/output/host/usr/bin $ cd
```

五、实验数据记录和处理

数据记录在四中完成

六、实验结果与分析

实验成功。成功在树莓派上编译了交叉编译工具链，并编译了 MIPS 指令的程序。

七、讨论、心得

本次实验进行的相当艰苦，首先是树莓派的系统，缺乏很多组件，在编译工具链的过程中一直报错，报错后根据报错的内容对组件进行一个个的补全，由于树莓派性能不足，编译的过程也相当漫长，并且在这过程中由于提升性能对树莓派进行了超频，导致了有一次编译过程中机器死机重启。不过经过了两天一夜的艰苦编译，终于编译出了 MIPS 指令程序。