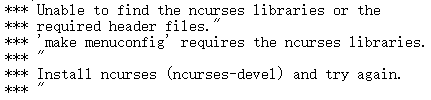
make menuconfig与库ncurses与pkg-config

如果在linux内核源码下执行make menuconfig时，出现与ncurseslibrariessh相关的错误：



那么需要从linux内核源码执行make menuconfig时自动生成的文件scripts/kconfig/lxdialog/check-lxdialog.sh查找原因，发现上述错误就是由该文件输出的。

在编译内核时，如果没有设置交叉编译环境前执行make menuconfig时正常，然而设置环境变量source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi 后在出现这种错误时，可以使用 make menuconfig ARCH=arm CROSS\_COMPILE=arm-poky-linux-gnueabi- 来解决，最好的方式是安装好库 ncursesw和 ncurses，执行以下两句安装：

apt-get install libncursesw5-dev

apt-get install libncurses5-dev

文件 scripts/kconfig/lxdialog/check-lxdialog.sh的作用是检查make menuconfig所依赖头文件<ncurses.h>和库ncurses或ncursesw是否存在，检查的方法就是带上这些编译参数编译一个文件：$cc -x c - -o $tmp 2>/dev/null <<'EOF' #include CURSES\_LOC main() {} EOF，具体参见后面的第二点。

文件 scripts/kconfig/lxdialog/check-lxdialog.sh的内容和注释如下：

#!/bin/sh

# Check ncurses compatibility

# What library to link

ldflags()

{

pkg-config --libs ncursesw 2>/dev/null && exit

/\*

一、符号&&的作用：比如 a && b,如果执行a语句成功（查询echo $?为0则表示成功）成立，那么就么接着执行b语句。

二、2>/dev/null：将标准错误输出2的信息重定向到设备文件/dev/null 中。

三、工具pkg-config可用与列举出某个库的相关信息，比如此库的路径、相关头文件路径等，工具pkg-config会首先在环境变量PKG\_CONFIG\_PATH定义的目录下查找库信息文件ncurses.pc（pc是pkg-config的缩写），如果没有找到，那么继续在路径/usr/lib/, /usr/share/, /usr/local/lib/ and /usr/local/share/中查找，可以使用 man pkg-config查看pkg-config的具体用法.

在设置环境变量source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi 后，

例如PKG\_CONFIG\_PATH=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi/usr/lib/pkgconfig，

执行pkg-config --libs ncursesw的作用是首先在目录PKG\_CONFIG\_PATH下查找名为ncursesw.pc的文件，执行选项为--libs，所以pkg-config会查找文件ncursesw.pc中的变量Libs的值，并返回，执行的结果为-lncursesw；执行echo $? 得到的结果为 0 。

\*/

pkg-config --libs ncurses 2>/dev/null && exit

/\*工具pkg-config会首先在环境变量PKG\_CONFIG\_PATH定义的目录下查找库信息文件ncurses.pc，并返回变量Libs的值。\*/

for ext in so a dll.a dylib ; do

for lib in ncursesw ncurses curses ; do

$cc -print-file-name=lib${lib}.${ext} | grep -q /

/\*

一、grep -q / 的作用是查找字符 “/”，如果有找到则保持安静quiet，而不输出，

-q 参数，本意是 Quiet; do not write anything to standard output.  Exit immediately with zero status if any match is found, even if an error was detected.   中文意思为，安静模式，不打印任何标准输出。如果有匹配的内容则立即返回状态值0，即echo $?的值为 0。

二、这里变量 $cc=$HOSTCC=gcc，参见本文后面的第二点，gcc -print-file-name=libxxx 的作用是打印出库libxxx的完整路径，例如：

root@ubuntu:/# gcc -print-file-name=libncurses.so

/usr/lib/gcc/x86\_64-linux-gnu/5/../../../x86\_64-linux-gnu/libncurses.so

选项-print-file-name的作用参见 man gcc：

-print-file-name=library

Print the full absolute name of the library file library that would be used when linking---and don't do anything else. With this option, GCC does not compile or link anything; it just prints the file name.

\*/

if [ $? -eq 0 ]; then

echo "-l${lib}"

exit

fi

done

done

exit 1

}

# Where is ncurses.h?：找出头文件ncurses.h的路径。

ccflags()

{

if pkg-config --cflags ncursesw 2>/dev/null; then

echo '-DCURSES\_LOC="<ncurses.h>" -DNCURSES\_WIDECHAR=1'

/\*在没有设置交叉编译环境时，工具pkg-config /usr/lib/ 目录下查找库信息文件ncurses.pc，并返回变量Cflags的值 -D\_GNU\_SOURCE，例如：

root@ubuntu:/# cat /usr/lib/x86\_64-linux-gnu/pkgconfig/ncurses.pc

Libs: -L${libdir} -lncurses -ltinfo

Libs.private:

Cflags: -D\_GNU\_SOURCE

\*/

elif pkg-config --cflags ncurses 2>/dev/null; then

echo '-DCURSES\_LOC="<ncurses.h>"'

elif [ -f /usr/include/ncursesw/curses.h ]; then

echo '-I/usr/include/ncursesw -DCURSES\_LOC="<curses.h>"'

echo ' -DNCURSES\_WIDECHAR=1'

elif [ -f /usr/include/ncurses/ncurses.h ]; then

echo '-I/usr/include/ncurses -DCURSES\_LOC="<ncurses.h>"'

elif [ -f /usr/include/ncurses/curses.h ]; then

echo '-I/usr/include/ncurses -DCURSES\_LOC="<curses.h>"'

elif [ -f /usr/include/ncurses.h ]; then

echo '-DCURSES\_LOC="<ncurses.h>"'

else

echo '-DCURSES\_LOC="<curses.h>"'

fi

}

# Temp file, try to clean up after us

tmp=.lxdialog.tmp

trap "rm -f $tmp" 0 1 2 3 15

# Check if we can link to ncurses

check() {

$cc -x c - -o $tmp 2>/dev/null <<'EOF' #$cc -x c - -o编译<<'EOF'…EOF 包围的代码。

#include CURSES\_LOC

main() {}

EOF

if [ $? != 0 ]; then

echo " \*\*\* Unable to find the ncurses libraries or the" 1>&2

echo " \*\*\* required header files." 1>&2

echo " \*\*\* 'make menuconfig' requires the ncurses libraries." 1>&2

echo " \*\*\* " 1>&2

echo " \*\*\* Install ncurses (ncurses-devel) and try again." 1>&2

echo " \*\*\* " 1>&2

exit 1

fi

}

usage() {

printf "Usage: $0 [-check compiler options|-ccflags|-ldflags compiler options]\n"

/\*

脚本 check-lxdialog.sh 的用法：

在内核源码根目录下的文件scripts/kconfig/Makefile会调用执行文件 scripts/kconfig/lxdialog/check-lxdialog.sh，

check-lxdialog := $(srctree)/$(src)/lxdialog/check-lxdialog.sh

HOST\_EXTRACFLAGS += $(shell $(CONFIG\_SHELL) $(check-lxdialog) -ccflags) \

HOSTLOADLIBES\_mconf = $(shell $(CONFIG\_SHELL) $(check-lxdialog) -ldflags $(HOSTCC))

$(obj)/dochecklxdialog:

$(Q)$(CONFIG\_SHELL) $(check-lxdialog) -check $(HOSTCC) $(HOST\_EXTRACFLAGS) $(HOSTLOADLIBES\_mconf)

其中的变量$HOSTCC 在linux 内核源码下的Makefile的第302行定义：HOSTCC = gcc，也就是说变量$HOSTCC的固定为gcc，只与编译内核的主机安装的gcc版本有关，并不会因为设置了交叉编译环境而改变。

打印出的信息：

/bin/sh imx-kernel-3.14/scripts/kconfig/lxdialog/check-lxdialog.sh -check gcc -I/usr/include/ncursesw -DCURSES\_LOC="<curses.h>" -DNCURSES\_WIDECHAR=1 -DLOCALE -lncursesw -ltinfo

\*/

}

if [ $# -eq 0 ]; then

usage

exit 1

fi

cc=""

case "$1" in

"-check")

shift

cc="$@" #这里变量$cc的值参见后面的第二点

check

;;

"-ccflags")

ccflags

;;

"-ldflags")

shift

cc="$@"

ldflags

;;

"\*")

usage

exit 1

;;

esac

1. 工具pkg-config的作用和用法

工具pkg-config可用与列举出某个库的相关信息，比如此库的路径、相关头文件路径等，工具pkg-config会优先在环境变量PKG\_CONFIG\_PATH定义的目录下查找库信息文件.pc（pc是pkg-config的缩写），如果没有找到，那么继续在路径/usr/lib/, /usr/share/, /usr/local/lib/ and /usr/local/share/中查找，可以使用 man pkg-config查看pkg-config的具体用法：

pkg-config looks in /usr/lib/pkgconfig, /usr/share/pkgconfig, /usr/local/lib/pkgconfig and /usr/local/share/pkgconfig for these files.

It will additionally look in the colon-separated (on Windows, semicolon-separated) list of directories specified by the PKG\_CONFIG\_PATH environment variable.

pkg-config常用参数：

--list-all 列出所有已安装的共享库

--cflags 列出指定共享库的预处理和编译flag，读取\*.pc文件中的变量 Cflags 的值。

--libs 列出指定共享库的链接flag，读取\*.pc文件中的变量 Libs 的值。

如果在环境变量 PKG\_CONFIG\_PATH 没有指定的路径，那么pkg-config会在/usr/lib/, /usr/share/, /usr/local/lib/ and /usr/local/share/中查找，如果还是没有找到就会报错：

root@ubuntu:/usr/lib/x86\_64-linux-gnu/pkgconfig# pkg-config --libs ncursesw

Package ncursesw was not found in the pkg-config search path.

Perhaps you should add the directory containing `ncursesw.pc'

to the PKG\_CONFIG\_PATH environment variable

No package 'ncursesw' found

在设置环境变量source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi 后，

例如PKG\_CONFIG\_PATH=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi/usr/lib/pkgconfig，

执行pkg-config --libs ncursesw的作用是首先在目录PKG\_CONFIG\_PATH下查找名为ncursesw.pc的文件，执行选项为--libs，所以pkg-config会查找文件ncursesw.pc中的变量Libs的值，并返回，执行的结果为-lncursesw；执行pkg-config --cflags ncursesw的作用是首先在目录PKG\_CONFIG\_PATH下查找名为ncursesw.pc的文件，执行选项为-- cflags，所以pkg-config会查找文件ncursesw.pc中的变量Cflags的值，并返回，执行的结果为空；文件ncursesw.pc有如下的内容：

[admin@localhostpkgconfig]$ catncursesw.pc

prefix=/usr

exec\_prefix=/usr

libdir=/usr/lib

includedir=/usr/include

major\_version=5

version=5.9.20110404

Name: ncursesw

Description: ncurses 5.9 library

Version: ${version}

Requires: tinfo

Libs: -lncursesw

Cflags:

工具pkg-config的参考资料：

https://blog.csdn.net/absurd/article/details/599813

https://blog.csdn.net/bryce\_xiao/article/details/70787810

https://blog.csdn.net/mantis\_1984/article/details/52847435

1. 解释： $cc -x c - -o $tmp 2>/dev/null <<'EOF' #include CURSES\_LOC main() {} EOF

这句话的作用是编译一个内容为 #include CURSES\_LOC main() {} 的C文件，宏 CURSES\_LOC 的值由编译选项-DCURSES\_LOC 指定，例如 -DCURSES\_LOC="<curses.h>"，则这里的 CURSES ＝ <curses.h>。变量$gcc的内容包含了ncurses 的库和头文件的路径，如果能编译成功，那么变量 $? ＝ 0，则检测库 ncurses 成功。编译出的文件名为$tmp，2>/dev/null的作用是将编译过程中输出的标准错误输出2的内容重定向到特殊设备文件。

在内核源码根目录下的文件scripts/kconfig/Makefile会调用执行脚本文件 ：

scripts/kconfig/lxdialog/check-lxdialog.sh。

脚本文件scripts/kconfig/lxdialog/check-lxdialog.sh 的被调用过程如下：

check-lxdialog := $(srctree)/$(src)/lxdialog/check-lxdialog.sh

HOST\_EXTRACFLAGS += $(shell $(CONFIG\_SHELL) $(check-lxdialog) -ccflags) \

HOSTLOADLIBES\_mconf = $(shell $(CONFIG\_SHELL) $(check-lxdialog) -ldflags $(HOSTCC))

$(obj)/dochecklxdialog:

$(Q)$(CONFIG\_SHELL) $(check-lxdialog) -check $(HOSTCC) $(HOST\_EXTRACFLAGS) $(HOSTLOADLIBES\_mconf)

其中的变量$HOSTCC 在linux 内核源码下的Makefile的第302行定义：HOSTCC = gcc，也就是说变量$HOSTCC的固定为gcc，只与编译内核的主机安装的gcc版本有关，并不会因为设置了交叉编译环境而改变。

调用执行脚本文件 scripts/kconfig/lxdialog/check-lxdialog.sh 的语句为：

$(Q)$(CONFIG\_SHELL) $(check-lxdialog) -check $(HOSTCC) $(HOST\_EXTRACFLAGS) $(HOSTLOADLIBES\_mconf)

根据make menuconfig V=1 打印出的信息：

/bin/sh imx-kernel-3.14/scripts/kconfig/lxdialog/check-lxdialog.sh -check gcc -I/usr/include/ncursesw -DCURSES\_LOC="<curses.h>" -DNCURSES\_WIDECHAR=1 -DLOCALE -lncursesw –ltinfo

再根据 case "$1" in

"-check")

shift

cc="$@"

check

其中的 $@ 是脚本或函数的所有参数，由此可以知道语句 $cc -x c - -o $tmp的$cc= gcc -I/usr/include/ncursesw -DCURSES\_LOC="<curses.h>" -DNCURSES\_WIDECHAR=1 -DLOCALE -lncursesw –ltinfo 。

选项 –x c 的作用参见 man gcc：

-x language

Specify explicitly the language for the following input files (rather than letting the compiler choose a default based on the file name suffix). This option applies to all following input files until the next -x option. Possible values for language are:

c c-header cpp-output

c++ c++-header c++-cpp-output

。。。。（省略）

还有一个比较奇怪的是选项 -x c - -o 中 –x c后面的横杠 – 的作用，根据下面的实际编译情况可以猜测横杠 – 的作用是替代 << 'EOF' … EOF的组成的内容：

root@ubuntu:/tmp# gcc -x c -o confuse << 'EOF'

> #include <stdio.h>

> void main() {

> printf("i am confused.\n");}

>

> EOF

gcc: warning: ‘-x c’ after last input file has no effect

gcc: fatal error: no input files

compilation terminated.

root@ubuntu:/tmp#

root@ubuntu:/tmp#

root@ubuntu:/tmp# gcc -x c - -o confuse << 'EOF'

> #include <stdio.h>

> void main() {

> printf("i am confused.\n");}

>

> EOF

root@ubuntu:/tmp# ./confuse

i am confused.

root@ubuntu:/tmp#

1. 综合以上

在设置好交叉编译环境source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi后，在执行make menuconfig 时，

mxl@ubuntu:~/work/kc532s/testmenuconfig/linux-imx$ make menuconfig V=1

make -f ./scripts/Makefile.build obj=scripts/basic

rm -f .tmp\_quiet\_recordmcount

make -f ./scripts/Makefile.build obj=scripts/kconfig menuconfig

/bin/bash ./scripts/kconfig/lxdialog/check-lxdialog.sh -check gcc -D\_GNU\_SOURCE -DCURSES\_LOC="<ncurses.h>" -DNCURSES\_WIDECHAR=1 -DLOCALE -lncursesw -ltinfo

\*\*\* Unable to find the ncurses libraries or the

\*\*\* required header files.

\*\*\* 'make menuconfig' requires the ncurses libraries.

\*\*\*

\*\*\* Install ncurses (ncurses-devel) and try again.

\*\*\*

scripts/kconfig/Makefile:201: recipe for target 'scripts/kconfig/dochecklxdialog' failed

make[1]: \*\*\* [scripts/kconfig/dochecklxdialog] Error 1

Makefile:546: recipe for target 'menuconfig' failed

make: \*\*\* [menuconfig] Error 2

mxl@ubuntu:~/work/kc532s/testmenuconfig/linux-imx$

出现上面找不到库ncurses 的错误时，可以将文件scripts/kconfig/lxdialog/check-lxdialog.sh 的语句

由 $cc -x c - -o $tmp 2>/dev/null <<'EOF' 修改为如下：

$cc -x c - -o $tmp <<'EOF'

这样编译过程中产生的错误可以输出显示出来：

mxl@ubuntu:~/work/kc532s/testmenuconfig/linux-imx$ make menuconfig V=1

make -f ./scripts/Makefile.build obj=scripts/basic

rm -f .tmp\_quiet\_recordmcount

make -f ./scripts/Makefile.build obj=scripts/kconfig menuconfig

/bin/bash ./scripts/kconfig/lxdialog/check-lxdialog.sh -check gcc -D\_GNU\_SOURCE -DCURSES\_LOC="<ncurses.h>" -DNCURSES\_WIDECHAR=1 -DLOCALE -lncursesw -ltinfo

<stdin>:2:1: warning: return type defaults to ‘int’ [-Wimplicit-int]

/usr/bin/ld: cannot find -lncursesw

collect2: error: ld returned 1 exit status

\*\*\* Unable to find the ncurses libraries or the

\*\*\* required header files.

\*\*\* 'make menuconfig' requires the ncurses libraries.

\*\*\*

\*\*\* Install ncurses (ncurses-devel) and try again.

\*\*\*

scripts/kconfig/Makefile:201: recipe for target 'scripts/kconfig/dochecklxdialog' failed

make[1]: \*\*\* [scripts/kconfig/dochecklxdialog] Error 1

Makefile:546: recipe for target 'menuconfig' failed

make: \*\*\* [menuconfig] Error 2

mxl@ubuntu:~/work/kc532s/testmenuconfig/linux-imx$

注意其中的错误信息：/usr/bin/ld: cannot find –lncursesw，说明没有找到库文件 ncursesw，原因是在设置好交叉编译环境source /opt/poky/1.7/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi后，指定了环境变量PKG\_CONFIG\_PATH，

PKG\_CONFIG\_PATH=/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi/usr/lib/pkgconfig，查看路径PKG\_CONFIG\_PATH 路径下文件ncursesw.pc的内容如下：

mxl@ubuntu:/opt/poky/1.7/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi/usr/lib/pkgconfig$ cat ncursesw.pc

prefix=/usr

exec\_prefix=/usr

libdir=/usr/lib

includedir=/usr/include

major\_version=5

version=5.9.20110404

Name: ncursesw

Description: ncurses 5.9 library

Version: ${version}

Requires: tinfo

Libs: -lncursesw

Cflags:

但是在目录/usr/lib 下并没有找到库ncursesw，

root@ubuntu:/# cd /

root@ubuntu:/# find -name "\*ncursesw\*"

./var/lib/dpkg/info/libncursesw5:amd64.triggers

./var/lib/dpkg/info/libncursesw5:amd64.md5sums

./var/lib/dpkg/info/libncursesw5:amd64.symbols

./var/lib/dpkg/info/libncursesw5:amd64.list

./var/lib/dpkg/info/libncursesw5:amd64.shlibs

./lib/x86\_64-linux-gnu/libncursesw.so.5

./lib/x86\_64-linux-gnu/libncursesw.so.5.9

有三种解决方案，第一种是将check-lxdialog.sh 的语句 pkg-config --libs ncursesw 2>/dev/null && exit 注释掉，

第二种方法是环境变量PKG\_CONFIG\_PATH 指定路径下的文件ncursesw.pc 的内容根据库 ncursesw 实际所在的路径进行设置，即修改变量prefix、exec\_prefixlibdir、includedir=/usr/include的值。

第三种方法是安装ncursesw来解决，在系统RedHat/Fedora/CentOS中执行： sudo yum install ncurses-devel

在系统Debian/Ubuntu中执行： sudo apt-get install libncursesw5-dev。

安装完成后查找库 ncursesw ：  
mxl@ubuntu:~$ cd /

mxl@ubuntu:/$ find -name "\*ncursesw\*"

./var/lib/dpkg/info/libncursesw5-dev:amd64.md5sums

./var/lib/dpkg/info/libncursesw5-dev:amd64.list

./var/lib/dpkg/info/libncursesw5:amd64.triggers

./var/lib/dpkg/info/libncursesw5:amd64.md5sums

./var/lib/dpkg/info/libncursesw5:amd64.symbols

./var/lib/dpkg/info/libncursesw5:amd64.list

./var/lib/dpkg/info/libncursesw5:amd64.shlibs

./lib/x86\_64-linux-gnu/libncursesw.so.5

./lib/x86\_64-linux-gnu/libncursesw.so.5.9

./usr/include/ncursesw

./usr/lib/x86\_64-linux-gnu/libncursesw.a

./usr/lib/x86\_64-linux-gnu/pkgconfig/ncursesw.pc

./usr/lib/x86\_64-linux-gnu/libncursesw.so

./usr/bin/ncursesw5-config

./usr/share/doc/libncursesw5

./usr/share/doc/libncursesw5-dev

./usr/share/man/man1/ncursesw5-config.1.gz