

实验 04_Makefile 编译

4.1 本章导读

在前面入门视频第十期“搭建编译环境 uboot_linux_Android”中，只介绍了输入 Make 命令就可以编译出内核，并没有介绍它是如何运行的。

在向内核中添加驱动的时候要完成 3 项工作，包括：

- 1) 在 Kconfig 中添加新代码对应项目的编译条件，这一部分在实验 3 已经介绍过；
- 2) 将驱动源码添加到对应的目录中，这一步比较好理解，在本章实验中给大家演示一下；
- 3) 在目录 Makefile 中文件中增加针对新代码的编译条目，这一部分将在本章节介绍到。

4.1.1 工具

4.1.1.1 硬件工具

- 1) PC 机

4.1.1.2 软件工具

- 1) 虚拟机 Vmware
- 2) Ubuntu12.04.2
- 3) Ubuntu 系统下解压生成的 Linux 源码

4.1.2 预备课程

入门视频“01-烧写、编译以及基础知识视频”

→ “实验 10-搭建编译环境 uboot_linux_Android” 或者使用手册 “五 Android 开发环境搭建以及编译”；

实验三 Menuconfig_Kconfig

4.1.3 视频资源

本节配套视频为 “视频 04_Makefile 编译”

4.2 学习目标

本章需要学习以下内容：

掌握 Linux 内核编译命令

掌握编译器路径设置的方法

理解环境变量路径、编译器、源码 Makefile 文件中编译器路径三者之间的关系

理解 Makefile 文件结构

掌握向 Makefile 文件中添加脚本命令的方法，能够看懂 Makefile 脚本

4.3 编译器路径的设置

在前面入门知识中，只介绍了在什么目录解压编译器，在环境变量文件中添加路径，就可以编译出内核，并没有介绍他们之间是如何对应的。

如下图所示，按照前面提供的基础教程，配置好 Ubuntu 中的编译器和库文件，输入命令 “#make”，就可以编译生成内核的二进制镜像。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib           REPORTING-BUGS
binary        drivers       MAINTAINERS   samples
block         firmware     Makefile      scripts
config_for_android  fs           mm           security
config_for_android_2M  include     modem.patch   sound
config_for_linux    init        modules.builtin  System.map
config_for_ubuntu   ipc         modules.order  tools
config_for_ubuntu_hdmi  Kbuild     Module.symvers  usr
COPYING          Kconfig     net           virt
CREDITS          kernel      pull_log.bat   vmlinux
crypto           kernel_readme.txt  README        vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# make
```

如下图所示，使用命令 “#cd” ，使用命令 “#vim .bashrc” 打开环境变量文件 “.bashrc”。

```
root@ubuntu: ~
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib           REPORTING-BUGS
binary        drivers       MAINTAINERS   samples
block         firmware     Makefile      scripts
config_for_android  fs           mm           security
config_for_android_2M  include     modem.patch   sound
config_for_linux    init        modules.builtin  System.map
config_for_ubuntu   ipc         modules.order  tools
config_for_ubuntu_hdmi  Kbuild     Module.symvers  usr
COPYING          Kconfig     net           virt
CREDITS          kernel      pull_log.bat   vmlinux
crypto           kernel_readme.txt  README        vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# cd
root@ubuntu: ~# vim .bashrc
```

如下图所示，进入最底行，可以看到，在环境变量文件中编译器路径设置为 “export PATH=\$PATH:/usr/local/arm/arm-2009q3/bin” 。

```
root@ubuntu: ~  
alias ll='ls -alF'  
alias la='ls -A'  
alias l='ls -CF'  
  
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then  
#    . /etc/bash_completion  
#fi  
  
#export PATH=$PATH:/usr/local/arm/4.3.2/bin  
#export PATH=$PATH:/usr/local/arm/4.4.1/bin  
export PATH=$PATH:/usr/local/arm/arm-2009q3/bin
```

103,1 Bot

然后使用命令 “#cd /usr/local/arm/” ，进入解压编译器 “arm-2009q3.tar.bz2” 的文件目录。

```
root@ubuntu: /usr/local/arm  
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls  
arch          Documentation  lib            REPORTING-BUGS  
binary        drivers        MAINTAINERS    samples  
block         firmware      Makefile       scripts  
config_for_android fs             mm            security  
config_for_android_2M include        modem.patch    sound  
config_for_linux init           modules.builtin System.map  
config_for_ubuntu ipc            modules.order  tools  
config_for_ubuntu_hdmi Kbuild        Module.symvers usr  
COPYING       Kconfig       net            virt  
CREDITS       kernel        pull_log.bat   vmlinux  
crypto        kernel_readme.txt README          vmlinux.o  
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# cd  
root@ubuntu: ~# vim .bashrc  
root@ubuntu: ~# cd /usr/local/arm/  
root@ubuntu: /usr/local/arm# ls  
4.3.2  arm-2009q3  arm-linux-4.4.1.tar.gz  
4.4.1  arm-2009q3.tar.bz2  arm-linux-gcc-4.3.2.tar.gz  
root@ubuntu: /usr/local/arm#
```

可以明显的看到，环境变量中设置的路径和解压的路径是对应的。

这里回到内核源码的目录，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
config_for_ubuntu_hdmi  Kbuild          Module.symvers  usr
COPYING                Kconfig          net              virt
CREDITS                kernel           pull_log.bat    vmlinux
crypto                 kernel_readme.txt README          vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# cd
root@ubuntu: ~# vim .bashrc
root@ubuntu: ~# cd /usr/local/arm/
root@ubuntu: /usr/local/arm# ls
4.3.2  arm-2009q3      arm-linux-4.4.1.tar.gz
4.4.1  arm-2009q3.tar.bz2 arm-linux-gcc-4.3.2.tar.gz
root@ubuntu: /usr/local/arm# cd /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib             REPORTING-BUGS
binary        drivers        MAINTAINERS     samples
block         firmware      Makefile        scripts
config_for_android  fs            mm             security
config_for_android_2M  include      modem.patch     sound
config_for_linux      init         modules.builtin System.map
config_for_ubuntu     ipc          modules.order  tools
config_for_ubuntu_hdmi Kbuild       Module.symvers  usr
COPYING            Kconfig      net            virt
CREDITS            kernel       pull_log.bat    vmlinux
crypto             kernel_readme.txt README          vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```

如下图所示，使用命令 “#vim Makefile” 打开内核目录下的 Makefile 文件。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
COPYING            Kconfig          net              virt
CREDITS            kernel           pull_log.bat    vmlinux
crypto             kernel_readme.txt README          vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# cd
root@ubuntu: ~# vim .bashrc
root@ubuntu: ~# cd /usr/local/arm/
root@ubuntu: /usr/local/arm# ls
4.3.2  arm-2009q3      arm-linux-4.4.1.tar.gz
4.4.1  arm-2009q3.tar.bz2 arm-linux-gcc-4.3.2.tar.gz
root@ubuntu: /usr/local/arm# cd /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib             REPORTING-BUGS
binary        drivers        MAINTAINERS     samples
block         firmware      Makefile        scripts
config_for_android  fs            mm             security
config_for_android_2M  include      modem.patch     sound
config_for_linux      init         modules.builtin System.map
config_for_ubuntu     ipc          modules.order  tools
config_for_ubuntu_hdmi Kbuild       Module.symvers  usr
COPYING            Kconfig      net            virt
CREDITS            kernel       pull_log.bat    vmlinux
crypto             kernel_readme.txt README          vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# vim Makefile
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```

这里通过 vim 编辑器查找参数 “CROSS_COMPILE”，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
VERSION = 3
PATCHLEVEL = 0
SUBLEVEL = 15
EXTRAVERSION =
NAME = Sneaky Weasel

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

# Do not:
# o use make's built-in rules and variables
#   (this increases performance and avoids hard-to-debug behaviour);
# o print "Entering directory ...";
MAKEFLAGS += -R --no-print-directory

# Avoid funny character set dependencies
unexport LC_ALL
LC_COLLATE=C
LC_NUMERIC=C
export LC_COLLATE LC_NUMERIC
/CROSS_COMPILE
```

可以查到参数 “CROSS_COMPILE” ，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
# When performing cross compilation for other architectures ARCH shall be set
# to the target architecture. (See arch/* for the possibilities).
# ARCH can be set during invocation of make:
# make ARCH=ia64
# Another way is to have ARCH set in the environment.
# The default ARCH is the host where make is executed.

# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# are prefixed with $(CROSS_COMPILE).
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=ia64-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# A third alternative is to store a setting in .config so that plain
# "make" in the configured kernel build directory always uses that.
# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile
export KBUILD_BUILDHOST := $(SUBARCH)
ARCH ?= arm
CROSS_COMPILE ?= /usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-
#CROSS_COMPILE ?= /usr/local/arm/4.5.1/bin/arm-none-linux-gnueabi-
CROSS_COMPILE ?= $(CONFIG_CROSS_COMPILE: "%"=%)

199,0-1 11%
```

如上图所示，可以看到这个参数是

“/usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-” 。

再看一下参数“CROSS_COMPILE”的下一行，可以看出这里有修改过的痕迹。很容易推断出，原来三星是用的“4.5.1”版本的编译器，不过这个没关系，只要能编译通过就可以了。

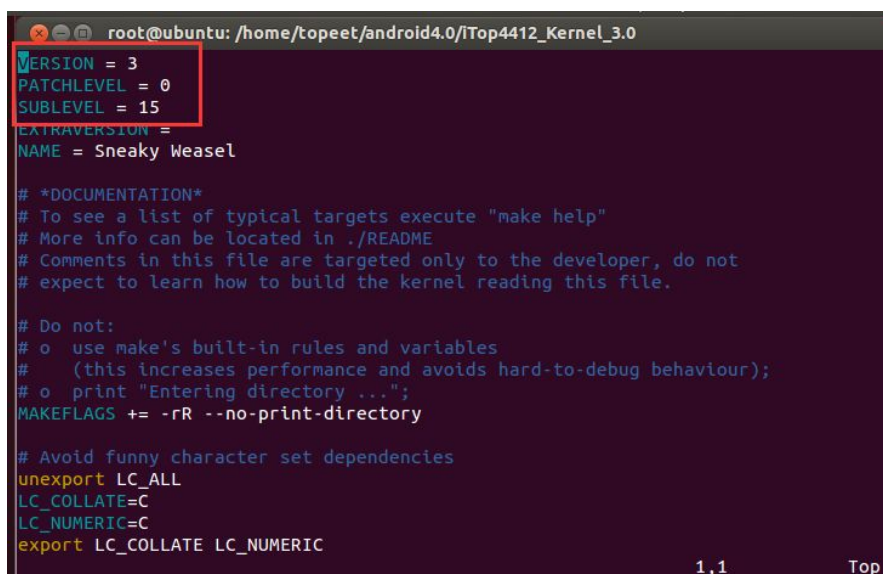
一般说来，拿到源码之后的第一步，是要先要将源码编译通过。

设置上图中的变量“CROSS_COMPILE”以及环境变量还有编译器实际解压路径三者对应。这三者对应之后，就能确保执行 Make 命令之后，系统能够找到这个编译器。

在编译命令执行的过程中，会提示一些错误，然后根据提示的错误，挨个去排查修改，去添加库文件或者修改库文件。这个过程有长有短，考验的只是耐心，并没有太多技巧。由于已经将缺少的库文件写成脚本，这样大家在编译器执行以下脚本就能够编译通过。

针对内核目录下的 Makefile 文件，提醒大家一下，这个文件中内容很多，除了上面编译器路径变量以外的其它部分几乎不用关注。

另外还有一个地方需要了解一下，如下图所示，在该文件的第一行，可以看到内核的版本。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
VERSION = 3
PATCHLEVEL = 0
SUBLEVEL = 15
EXTRAVERSION =
NAME = Sneaky Weasel

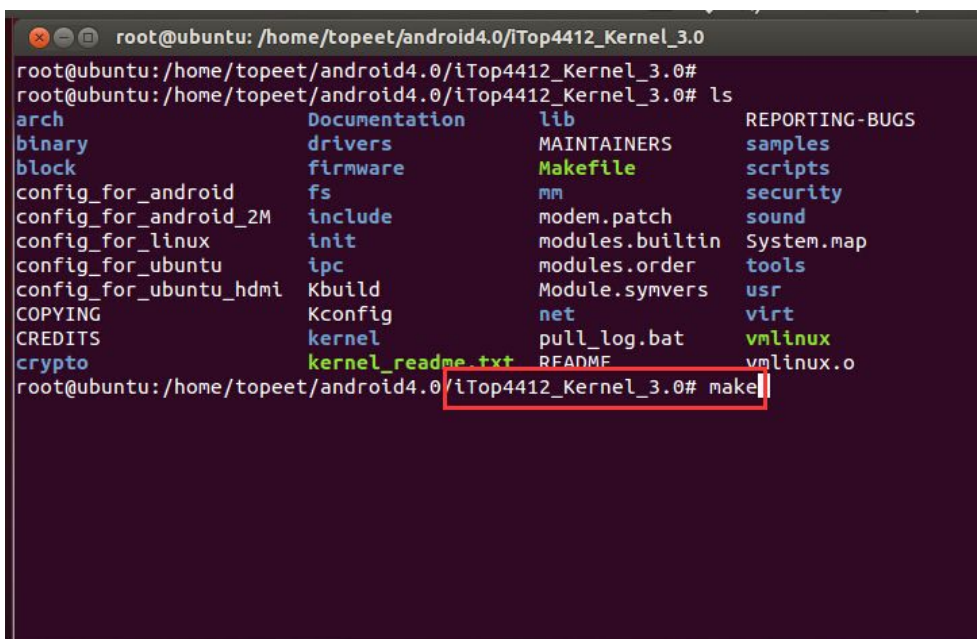
# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

# Do not:
# o use make's built-in rules and variables
#   (this increases performance and avoids hard-to-debug behaviour);
# o print "Entering directory ...";
MAKEFLAGS += -rR --no-print-directory

# Avoid funny character set dependencies
unexport LC_ALL
LC_COLLATE=C
LC_NUMERIC=C
export LC_COLLATE LC_NUMERIC
```

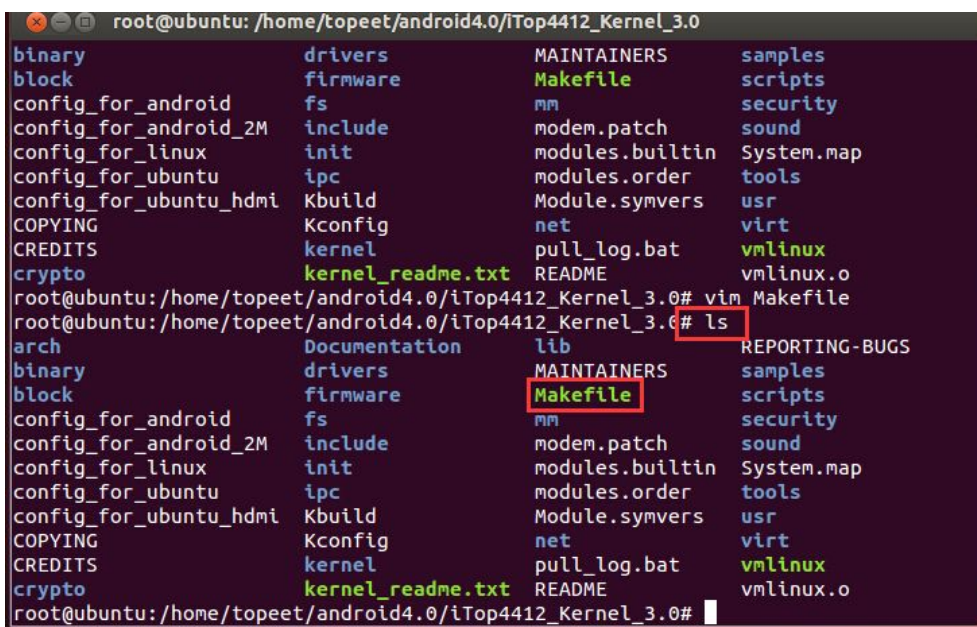

4.4 Make 内核编译命令执行过程简介

如下图所示，编译内核需要执行 make 命令，在执行 make 命令之后，并没有介绍它是如何执行的。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib           REPORTING-BUGS
binary        drivers        MAINTAINERS  samples
block         firmware      Makefile     scripts
config_for_android  fs           mm           security
config_for_android_2M  include      modem.patch  sound
config_for_linux    init         modules.builtin  System.map
config_for_ubuntu   ipc          modules.order  tools
config_for_ubuntu_hdmi  Kbuild      Module.symvers  usr
COPYING          Kconfig       net           virt
CREDITS           kernel        pull_log.bat  vmlinux
crypto            kernel_readme.txt  README       vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# make
```

在 make 命令执行之后，它会首先找到当前目录下的 “Makefile” 文件，如下图所示。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# vim Makefile
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls
arch          Documentation  lib           REPORTING-BUGS
binary        drivers        MAINTAINERS  samples
block         firmware      Makefile     scripts
config_for_android  fs           mm           security
config_for_android_2M  include      modem.patch  sound
config_for_linux    init         modules.builtin  System.map
config_for_ubuntu   ipc          modules.order  tools
config_for_ubuntu_hdmi  Kbuild      Module.symvers  usr
COPYING          Kconfig       net           virt
CREDITS           kernel        pull_log.bat  vmlinux
crypto            kernel_readme.txt  README       vmlinux.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```


运行 Makefile 文件之后，它会在 Makefile 文件中找到编译器的路径

“/usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-” 如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
# ARCH can be set during invocation of make:
# make ARCH=ia64
# Another way is to have ARCH set in the environment.
# The default ARCH is the host where make is executed.

# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# are prefixed with $(CROSS_COMPILE).
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=ia64-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# A third alternative is to store a setting in .config so that plain
# "make" in the configured kernel build directory always uses that.
# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile
export KBUILD_BUILDHOST := $(SUBARCH)
ARCH ?= arm
CROSS_COMPILE ?= /usr/local/arm/arm-2009q3/bin/arm-none-linux-gnueabi-
#CROSS_COMPILE ?= /usr/local/arm/4.5.1/bin/arm-none-linux-gnueabi-
CROSS_COMPILE ?= $(CONFIG_CROSS_COMPILE: "%"=%)

# Architecture as present in compile.h
UT_MACHINE := $(ARCH)
```

然后系统根据环境变量找到 “export PATH=\$PATH:/usr/local/arm/arm-2009q3/bin”

编译器的路径，如下图所示。

```
root@ubuntu: ~
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

#export PATH=$PATH:/usr/local/arm/4.3.2/bin
#export PATH=$PATH:/usr/local/arm/4.4.1/bin
export PATH=$PATH:/usr/local/arm/arm-2009q3/bin
```

如下图所示，使用命令 “ls /usr/local/arm/arm-2009q3/bin/” 可以查看到 arm2009q3 解压之后有哪些具体编译器。

```

root@ubuntu: ~
config_for_linux      init                modules.builtin      System.map
config_for_ubuntu     ipc                modules.order         tools
config_for_ubuntu_hdmi Kbuild            Module.symvers        usr
COPYING               Kconfig           net                   virt
CREDITS               kernel            pull_log.bat          vmlinux
crypto                kernel_readme.txt README                 vmlinux.o

root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# vim Makefile
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# cd
root@ubuntu: ~#
root@ubuntu: ~# vim bashrc
root@ubuntu: ~# ls /usr/local/arm/arm-2009q3/bin/
arm-none-linux-gnueabi-addrd2line  arm-none-linux-gnueabi-gprof
arm-none-linux-gnueabi-ar          arm-none-linux-gnueabi-ld
arm-none-linux-gnueabi-as          arm-none-linux-gnueabi-nm
arm-none-linux-gnueabi-c++         arm-none-linux-gnueabi-objcopy
arm-none-linux-gnueabi-c++filt     arm-none-linux-gnueabi-objdump
arm-none-linux-gnueabi-cpp         arm-none-linux-gnueabi-ranlib
arm-none-linux-gnueabi-g++         arm-none-linux-gnueabi-readelf
arm-none-linux-gnueabi-gcc         arm-none-linux-gnueabi-size
arm-none-linux-gnueabi-gcc-4.4.1  arm-none-linux-gnueabi-sprite
arm-none-linux-gnueabi-gcov        arm-none-linux-gnueabi-strings
arm-none-linux-gnueabi-gdb         arm-none-linux-gnueabi-strip
arm-none-linux-gnueabi-gdbtui
root@ubuntu: ~#

```

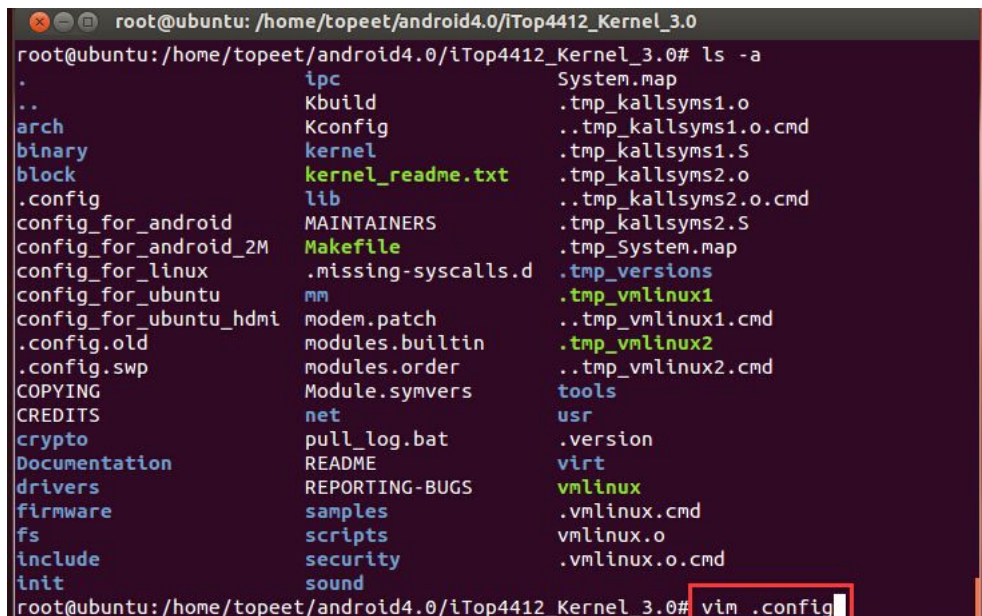
如上图，系统找到编译器之后，同时基础的库文件也是和编译器在一起的，有时候编译一个新的内核，还有可能需要修改库文件。

编译执行前还需要找到 “.config” 文件，默认是放在源码目录下的。使用命令 “ls -a” 就可以看到，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls -a
.                ipc                System.map
..               Kbuild            .tmp_kallsyms1.o
arch             Kconfig           ..tmp_kallsyms1.o.cmd
binary           kernel            .tmp_kallsyms1.S
block            kernel_readme.txt .tmp_kallsyms2.o
.config          lib               ..tmp_kallsyms2.o.cmd
config_for_android  MAINTAINERS      .tmp_kallsyms2.S
config_for_android_2M Makefile          .tmp_System.map
config_for_linux   .missing-syscalls.d .tmp_versions
config_for_ubuntu  mm               .tmp_vmlinuxx1
config_for_ubuntu_hdmi modem.patch       ..tmp_vmlinuxx1.cmd
.config.old        modules.builtin  .tmp_vmlinuxx2
.config.swp        modules.order    ..tmp_vmlinux2.cmd
COPYING           Module.symvers   tools
CREDITS           net              usr
crypto            pull_log.bat     .version
Documentation      README           virt
drivers            REPORTING-BUGS   vmlinux
firmware          samples          .vmlinux.cmd
fs                scripts          vmlinux.o
include           security         .vmlinux.o.cmd
init              sound

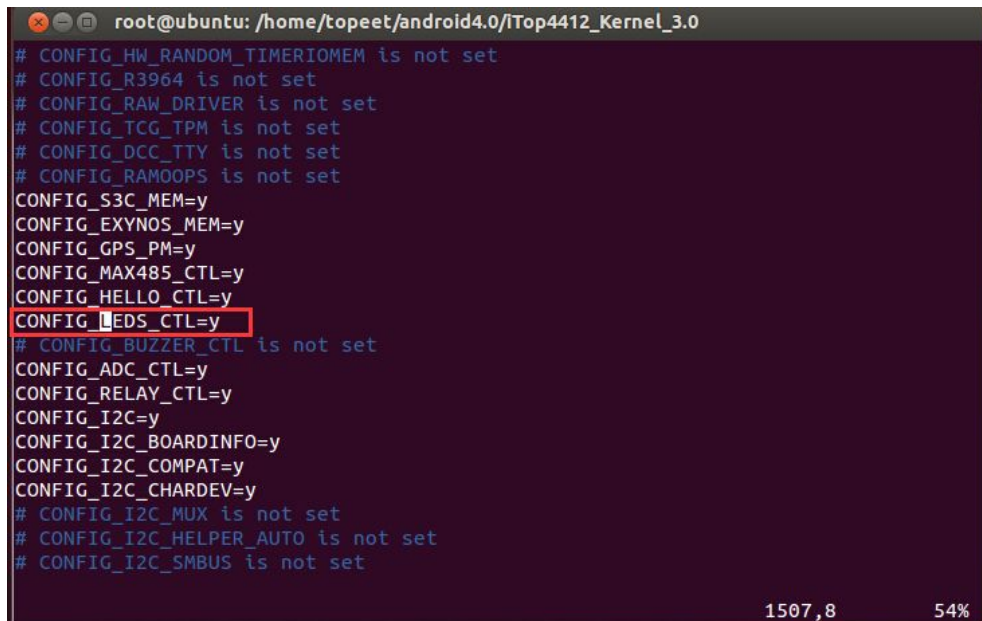
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```

从前面的实验 3.5 小节，学习到这个 “.config” 是通过 menuconfig 工具生成的，里面只是一些宏定义，如下图所示，打开这个文件。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls -a
.                ipc                System.map
..               Kbuild             .tmp_kallsyms1.o
arch             Kconfig            ..tmp_kallsyms1.o.cmd
binary           kernel              .tmp_kallsyms1.S
block            kernel_readme.txt  .tmp_kallsyms2.o
.config          lib                 ..tmp_kallsyms2.o.cmd
config_for_android  MAINTAINERS        .tmp_kallsyms2.S
config_for_android_2M  Makefile           .tmp_System.map
config_for_linux      .missing-syscalls.d .tmp_versions
config_for_ubuntu      mm                  .tmp_vmlinux1
config_for_ubuntu_hdmi  modem.patch         ..tmp_vmlinux1.cmd
.config.old          modules.builtin     .tmp_vmlinux2
.config.swp          modules.order       ..tmp_vmlinux2.cmd
COPYING            Module.symvers      tools
CREDITS            net                 usr
crypto             pull_log.bat        .version
Documentation       README              virt
drivers             REPORTING-BUGS      vmlinux
firmware            samples             vmlinux.cmd
fs                  scripts             vmlinux.o
include             security            vmlinux.o.cmd
init                sound
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# vim .config
```

查找宏定义“LEDS”，“”是编译 led 驱动的宏定义“LEDS_CTL”，如下图所示。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
# CONFIG_HW_RANDOM_TIMERIOMEM is not set
# CONFIG_R3964 is not set
# CONFIG_RAW_DRIVER is not set
# CONFIG_TCG_TPM is not set
# CONFIG_DCC_TTY is not set
# CONFIG_RAMOOPS is not set
CONFIG_S3C_MEM=y
CONFIG_EXYNOS_MEM=y
CONFIG_GPS_PM=y
CONFIG_MAX485_CTL=y
CONFIG_HELLO_CTL=y
CONFIG_LEDS_CTL=y
# CONFIG_BUZZER_CTL is not set
CONFIG_ADC_CTL=y
CONFIG_RELAY_CTL=y
CONFIG_I2C=y
CONFIG_I2C_BOARDINFO=y
CONFIG_I2C_COMPAT=y
CONFIG_I2C_CHARDEV=y
# CONFIG_I2C_MUX is not set
# CONFIG_I2C_HELPER_AUTO is not set
# CONFIG_I2C_SMBUS is not set

1507,8      54%
```

现在系统找到这个宏定义“LEDS_CTL”，在编译具体中间文件的时候会用到。

具体是怎么实现的，继续看下一小节的内容就可以搞清楚这个宏定义有什么用处。

4.5 Makefile 文件

接前一小节的内容，继续介绍系统是如何一步一步编译出内核镜像的。

在这里仍然以 LEDS 小灯为例。

4.5.1 宏定义 LEDS_CTL 的使用

如下图所示，led 驱动属于字符驱动，字符驱动一般是在源码目录 “drivers/char/” 下。

```
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  mem.o              raw.c
apm-emulation.c    hpet.c             misc.c             rtc.c
applicom.c          hw_random           misc.o             s3c_mem.c
applicom.h          i8k.c              mmtimer.c          s3c_mem.h
bfin-otp.c          ipmi                modules.builtin    s3c_mem.o
briq_panel.c        itop4412_adc.c      modules.order       scc.h
bsr.c               itop4412_adc.o      msm_smd_pkt.c       scx200_gpio.c
built-in.o          itop4412_buzzer.c   mspec.c            snsc.c
dcc_tty.c           itop4412_buzzer.o   mwave              snsc_event.c
ds1302.c            itop4412_leds.c     nsc_gpio.c          snsc.h
ds1620.c            itop4412_leds.o     nvram.c             sonypi.c
dsp56k.c            itop4412_relay.c    nwbutton.c          tb0219.c
dtlk.c              itop4412_relay.o    nwbutton.h          tlclk.c
efirtc.c            Kconfig             nwflash.c           toshiba.c
exynos_mem.c        lp.c                pc8736x_gpio.c      tpm
exynos_mem.o        Makefile             pcncia              ttyprintk.c
generic_nvram.c     max485_ctl.c         ppdev.c             uv_mmtimer.c
genrtc.c            max485_ctl.o         ps3flash.c          viotape.c
gps.c               mbc.c               ramoops.c           virtio_console.c
gps.h               mbc.h               random.c             xilinx_hwicap
gps.o               mem.c               random.o
```

如上图所示，这里要关注的文件只有框框中的三个 “itop4412_leds.c” ，

“itop4412_leds.o” ， “Makefile” 。

itop4412_leds.c ：比较好理解，就是 led 驱动的源码

itop4412_leds.o ：这个是生成最终 zImage 二进制的中间文件

Makefile ：就是 Make 命令所需要的文件

这里不关心 itop4412_leds.c 中的代码，只需先知道 itop4412_leds.c 的文件名即可。在

源码目录中使用命令 “vim drivers/char/Makefile” ，如下图所示。


```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
agp                hangcheck-timer.c  misc.o             s3c_mem.c
apm-emulation.c    hpet.c             mmtimer.c          s3c_mem.h
applicom.c         hw_random          modules.builtin    s3c_mem.o
applicom.h         i8k.c             modules.order      scc.h
bfin-otp.c         ipmi               msm_snd_pkt.c      scx200_gpio.c
briq_panel.c       itop4412_adc.c     mspec.c            snsc.c
bsr.c              itop4412_adc.o     mwave              snsc_event.c
built-in.o         itop4412_buzzer.c  nsc_gpio.c         snsc.h
dcc_tty.c          itop4412_leds.c    nvram.c            sonypi.c
ds1302.c           itop4412_relay.c   nwbutton.c         tb0219.c
ds1620.c           itop4412_relay.o   nwbutton.h         tlclk.c
dsp56k.c           Kconfig            nwflash.c          toshiba.c
dtlk.c             lp.c               pc8736x_gpio.c     tpm
efirtc.c           Makefile           pcmcia              ttyprintk.c
exynos_mem.c       max485_ctl.c       ppdev.c            uv_mmtimer.c
exynos_mem.o       max485_ctl.o       ps3flash.c         viotape.c
generic_nvram.c    mbcs.c             ramoops.c          virtio_console.c
genrtc.c           mbc.h              random.c            xilinx_hwicap
gps.c              mem.c              random.o
gps.h              mem.o              raw.c
gps.o              misc.c             rtc.c
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# vim drivers/char/Makefile
```

打开“vim drivers/char/Makefile”文件之后，找到和 itop4412_leds.c 相关的文件，搜索“itop4412_leds”，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
obj-$(CONFIG_PCMCIA)      += pcmcia/
obj-$(CONFIG_IPMI_HANDLER) += ipmi/

obj-$(CONFIG_HANGCHECK_TIMER) += hangcheck-timer.o
obj-$(CONFIG_TCG_TPM)      += tpm/

obj-$(CONFIG_DCC_TTY)      += dcc_tty.o
obj-$(CONFIG_PS3_FLASH)    += ps3flash.o
obj-$(CONFIG_RAMOOOPS)     += ramoops.o

obj-$(CONFIG_JS_RTC)      += js-rtc.o
js-rtc-y = rtc.o

obj-$(CONFIG_S3C_MEM)     += s3c_mem.o
obj-y += gps.o

obj-$(CONFIG_MAX485_CTL)   += max485_ctl.o
obj-$(CONFIG_LEDS_CTL)     += itop4412_leds.o
obj-$(CONFIG_BUZZER_CTL)   += itop4412_buzzer.o
obj-$(CONFIG_ADC_CTL)     += itop4412_adc.o
obj-$(CONFIG_RELAY_CTL)   += itop4412_relay.o

obj-$(CONFIG_EXYNOS_MEM)  += exynos_mem.o
/itop4412_leds
```

如上图方框所示，就是编译 itop4412_leds.c 的脚本。这个脚本大家也只需要仿照写即可，严格的按照已有的例子来写即可。

```
obj-$(CONFIG_MAX485_CTL) += max485_ctl.o
obj-$(CONFIG_LEDS_CTL) += itop4412_leds.o
obj-$(CONFIG_BUZZER_CTL) += itop4412_buzzer.o
obj-$(CONFIG_ADC_CTL) += itop4412_adc.o
obj-$(CONFIG_RELAY_CTL) += itop4412_relay.o
```

如果想添加类似的字符驱动，就可以在这个目录下添加。

Kconfig、Makefile、menuconfig、“.config”文件大家就可以联系起来了。

4.5.2 Makefile 脚本语法简介

本小节介绍的语法是对内核源码子目录中 Makefile 进行简单的介绍，这部分是经常会用到的，也是需要掌握的。

常用的强制编译写法，还是在“drivers/char/Makefile”中，在 Top 行，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
# Makefile for the kernel character device drivers.
#
obj-y += mem.o random.o
obj-$(CONFIG_TTY_PRINTK) += ttyprintk.o
obj-y += misc.o
obj-$(CONFIG_ATARI_DSP50K) += dsp50k.o
obj-$(CONFIG_VIRTIO_CONSOLE) += virtio_console.o
obj-$(CONFIG_RAW_DRIVER) += raw.o
obj-$(CONFIG_SGI_SNSC) += snsc.o snsc_event.o
obj-$(CONFIG_MSM_SMD_PKT) += msm_smd_pkt.o
obj-$(CONFIG_MSPEC) += mspec.o
obj-$(CONFIG_MMTIMER) += mmtimer.o
obj-$(CONFIG_UV_MMTIMER) += uv_mmtimer.o
obj-$(CONFIG_VIOTAPE) += viotape.o
obj-$(CONFIG_IBM_BSR) += bsr.o
obj-$(CONFIG_SGI_MBCS) += mbc.o
obj-$(CONFIG_BRIQ_PANEL) += briq_panel.o
obj-$(CONFIG_BFIN_OTP) += bfin-otp.o
obj-$(CONFIG_PRINTER) += lp.o

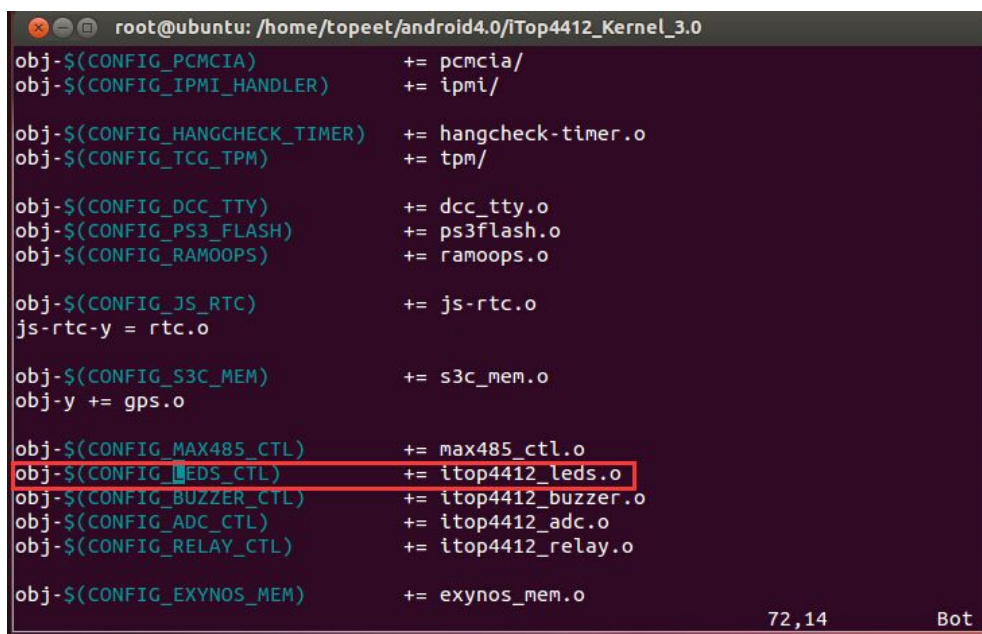
1,1 Top
```

上图中的这一行 “obj-y += misc.o” , 注意这里需要和内核自带的代码对齐。

“加等号” 右边的 “misc.o” 表示要编译 Makefile 目录下的 “misc.c” 文件 ,

“加等号” 左边的 “obj-y ” 表示要编译 , 并链接进内核 (链接进内核是 linux 源码中自带的工具处理的 , 驱动工程师完全不用关心具体是怎么实现的)

如下图所示 , 就是 LEDS_CTL 的条件编译 , 也就是需要在 Kconfig 中定义 , 在 menuconfig 中配置之后 , 编译器运行的时候找到对应的宏变量 LEDS_CTL 之后才会编译。

A terminal window showing the configuration of kernel objects. The prompt is 'root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0'. The output lists various configuration options and their corresponding object files. The line 'obj-\$(CONFIG_LEDS_CTL) += itop4412_leds.o' is highlighted with a red box. The terminal also shows 'js-rtc-y = rtc.o' and 'obj-y += gps.o'. The bottom right corner of the terminal window displays '72,14' and 'Bot'.

上图比较好理解 , 因为一直是以这个 LEDS_CTL 为例来讲解的。这种方式在写驱动的时候会经常使用到。

最后介绍一下目录层次的迭代，如下图所示，使用命令“vim drivers/Makefile”打开“drivers/char”上一层目录的 Makefile 文件。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
bfin-otp.c      ipmi      msm_smd_pkt.c  scx200_gpio.c
briq_panel.c   itop4412_adc.c  mspec.c       snsc.c
bsr.c          itop4412_adc.o  mwave         snsc_event.c
built-in.o     itop4412_buzzer.c nsc_gpio.c    snsc.h
dcc_tty.c      itop4412_leds.c nvram.c       sonypi.c
ds1302.c       itop4412_relay.c nwbutton.c    tb0219.c
ds1620.c       itop4412_relay.o nwbutton.h    tlclk.c
dsp56k.c       Kconfig        nwflash.c     toshiba.c
dtlk.c         lp.c           pc8736x_gpio.c tpm
efirtc.c       Makefile       pcmcia        ttyprintk.c
exynos_mem.c   max485_ctl.c   ppdev.c       uv_mmtimer.c
exynos_mem.o   max485_ctl.o   ps3flash.c    viotape.c
generic_nvram.c mbc.c         ramoops.c     virtio_console.c
genrtc.c       mbc.h         random.c      xilinx_hwicap
gps.c          mem.c         random.o
gps.h          mem.o         raw.c
gps.o          misc.c        rtc.c
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# vim drivers/char/Makefile
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0#
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# vim drivers/char/Makefile
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# vim drivers/Makefile
```

搜索关键词“char”，因为前面介绍到的字符变量在“char”目录下，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
obj-$(CONFIG_RAPIDIO) += rapidio/
obj-y                 += video/
obj-y                 += idle/
obj-$(CONFIG_ACPI)   += acpi/
obj-$(CONFIG_SFI)    += sfi/
# PnP must come after ACPI since it will eventually need to check if acpi
# was used and do nothing if so
obj-$(CONFIG_PNP)    += pnp/
obj-$(CONFIG_ARM_AMBA) += amba/
# Many drivers will want to use DMA so this has to be made available
# really early.
obj-$(CONFIG_DMA_ENGINE) += dma/

obj-$(CONFIG_VIRTIO) += virtio/
obj-$(CONFIG_XEN)    += xen/

# regulators early, since some subsystems rely on them to initialize
obj-$(CONFIG_REGULATOR) += regulator/

# tty/ comes before char/ so that the VT console is the boot-time
# default.
obj-v                 += tty/
obj-y                 += char/
/char
```

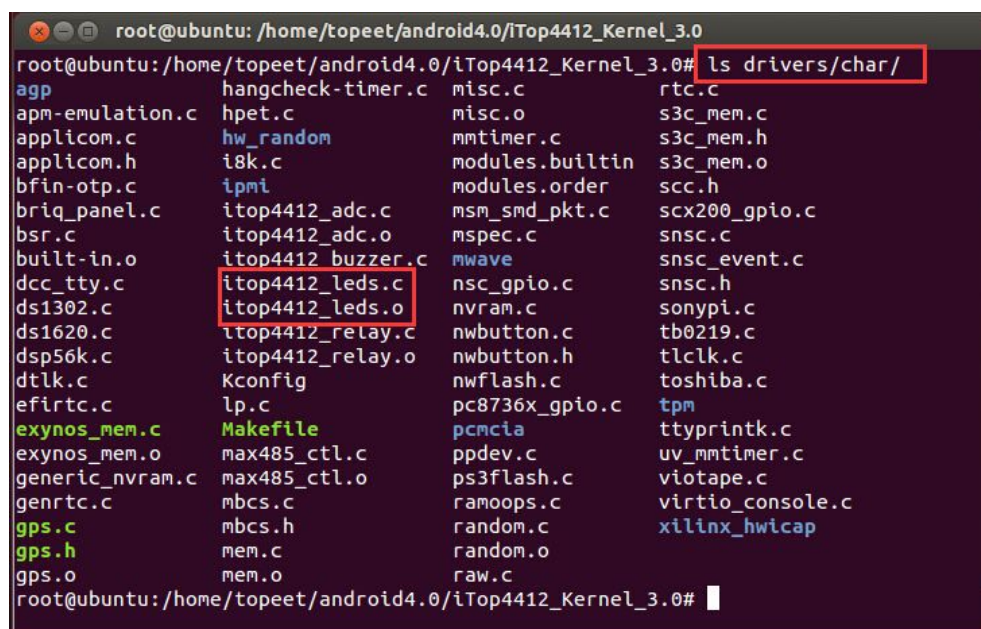

上图中的语法也是很简单，只不过“加等号”右边有文件变为了文件夹。这里表示强制编译当前目录“/drivers”的下一级目录“/char”。在执行编译命令执行到这一句的时候，就会先跳转到“/char”目录下的“Makefile”文件。

4.6 Makefile 测试

本小节给大家做个简单的小实验，通过配置 menuconfig 中的 LEDS，来将内核编译进内核或者不编译进内核。

4.6.1 将 LEDS 驱动不编译进内核

如下图所示，在给大家提供的源码中，解压之后默认状态就有将“itop4412_leds.c”编译进内核的中间文件“itop4412_leds.o”，如下图所示。



```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  misc.o             rtc.c
apm-emulation.c    hpet.c             mmtimer.c          s3c_mem.c
applicom.c          hw_random           modules.builtin    s3c_mem.h
applicom.h          i8k.c              modules.order      s3c_mem.o
bfin-otp.c          ipmi                msm_smd_pkt.c      scc.h
briq_panel.c        itop4412_adc.c     mspec.c            scx200_gpio.c
bsr.c               itop4412_adc.o     mwave              snsc.c
built-in.o          itop4412_buzzer.c  nsc_gpio.c         snsc_event.c
dcc_tty.c           itop4412_leds.c    nvram.c            snsc.h
ds1302.c            itop4412_leds.o    nwbutton.c         sonypi.c
ds1620.c            itop4412_relay.c   nwbutton.h         tb0219.c
dsp56k.c            itop4412_relay.o   nwflash.c          tlclk.c
dtlk.c              Kconfig            pc8736x_gpio.c     toshiba.c
efirtc.c            lp.c               pcmcia              tpm
exynos_mem.c        Makefile            ppdev.c            ttyprintk.c
exynos_mem.o        max485_ctl.c        ps3flash.c         uv_mmtimer.c
generic_nvram.c     max485_ctl.o        ramoops.c          viotape.c
genrtc.c            mbc.c               random.c           virtio_console.c
gps.c               mbc.h              random.o           xilinx_hwicap
gps.h               mem.c               raw.c
gps.o               mem.o
```

如下图所示，使用命令“rm -rf drivers/char/itop4412_leds.o”删除掉 LEDS 驱动的中间文件“itop4412_leds.o”。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  misc.c             rtc.c
apm-emulation.c   hpet.c             misc.o             s3c_mem.c
applicom.c         hw_random          mmtimer.c          s3c_mem.h
applicom.h         i8k.c              modules.builtin    s3c_mem.o
bfin-otp.c         ipmi                modules.order       scc.h
briq_panel.c       itop4412_adc.c     msm_smd_pkt.c      scx200_gpio.c
bsr.c              itop4412_adc.o     mspec.c            snsc.c
built-in.o         itop4412_buzzer.c  mwave              snsc_event.c
dcc_tty.c          itop4412_leds.c    nsc_gpio.c         snsc.h
ds1302.c           itop4412_leds.o    nvram.c            sonypi.c
ds1620.c           itop4412_relay.c   nwbutton.c         tb0219.c
dsp56k.c           itop4412_relay.o   nwbutton.h         tlclk.c
dtlk.c             Kconfig            nwflash.c          toshiba.c
efirtc.c           lp.c               pc8736x_gpio.c     tpm
exynos_mem.c       Makefile            pcmcia              ttyprintk.c
exynos_mem.o       max485_ctl.c        ppdev.c            uv_mmtimer.c
generic_nvram.c    max485_ctl.o        ps3flash.c         viotape.c
genrtc.c           mbc.c              ramoops.c          virtio_console.c
gps.c              mbc.h              random.c           xilinx_hwicap
gps.h              mem.c               random.o
gps.o              mem.o               raw.c
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# rm -rf drivers/char/ito
p4412_leds.o
```

如下图所示，使用命令“ls drivers/char/”查看一下，发现已经没有了“itop4412_leds.o”

文件。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
p4412_leds.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  misc.o             s3c_mem.c
apm-emulation.c   hpet.c             mmtimer.c          s3c_mem.h
applicom.c         hw_random          modules.builtin    s3c_mem.o
applicom.h         i8k.c              modules.order       scc.h
bfin-otp.c         ipmi                msm_smd_pkt.c      scx200_gpio.c
briq_panel.c       itop4412_adc.c     mspec.c            snsc.c
bsr.c              itop4412_adc.o     mwave              snsc_event.c
built-in.o         itop4412_buzzer.c  nsc_gpio.c         snsc.h
dcc_tty.c          itop4412_leds.c    nvram.c            sonypi.c
ds1302.c           itop4412_relay.c   nwbutton.c         tb0219.c
ds1620.c           itop4412_relay.o   nwbutton.h         tlclk.c
dsp56k.c           Kconfig            nwflash.c          toshiba.c
dtlk.c             lp.c               pc8736x_gpio.c     tpm
efirtc.c           Makefile            pcmcia              ttyprintk.c
exynos_mem.c       max485_ctl.c        ppdev.c            uv_mmtimer.c
exynos_mem.o       max485_ctl.o        ps3flash.c         viotape.c
generic_nvram.c    mbc.c              ramoops.c          virtio_console.c
genrtc.c           mbc.h              random.c           xilinx_hwicap
gps.c              mem.c               random.o
gps.h              mem.o               raw.c
gps.o              misc.c              rtc.c
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```

然后使用命令“make menuconfig”，打开配置工具，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
p4412_leds.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  misc.o             s3c_mem.c
apm-emulation.c    hpet.c             mmtimer.c          s3c_mem.h
applicom.c          hw_random           modules.builtin    s3c_mem.o
applicom.h          i8k.c              modules.order       scc.h
bfin-otp.c          ipmi                msm_smd_pkt.c      scx200_gpio.c
briq_panel.c        itop4412_adc.c      mspec.c            snsc.c
bsr.c               itop4412_adc.o      mwave              snsc_event.c
built-in.o          itop4412_buzzer.c   nsc_gpio.c         snsc.h
dcc_tty.c           itop4412_leds.c     nvram.c            sonypi.c
ds1302.c            itop4412_relay.c    nwbutton.c         tb0219.c
ds1620.c            itop4412_relay.o    nwbutton.h         tlclk.c
dsp56k.c            Kconfig             nwflash.c          toshiba.c
dtlk.c              lp.c                pc8736x_gpio.c     tpm
efirtc.c            Makefile             pcmcia              ttyprintk.c
exynos_mem.c        max485_ctl.c         ppdev.c            uv_mmtimer.c
exynos_mem.o        max485_ctl.o         ps3flash.c         viotape.c
generic_nvram.c     mbcs.c              ramoops.c          virtio_console.c
genrtc.c            mbcs.h              random.c           xilinx_hwicap
gps.c               mem.c               random.o
gps.h               mem.o               raw.c
gps.o               misc.c              rtc.c
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# make menuconfig
```

打开之后，如下图所示。找到 “Device Drivers ---> ” 选项。

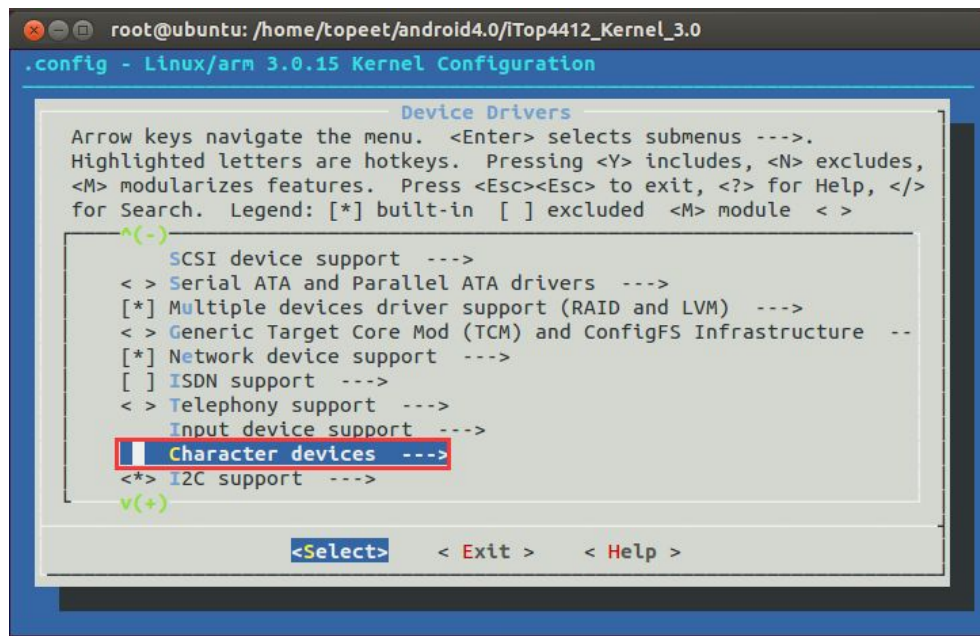
```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
.config - Linux/arm 3.0.15 Kernel Configuration

Linux/arm 3.0.15 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

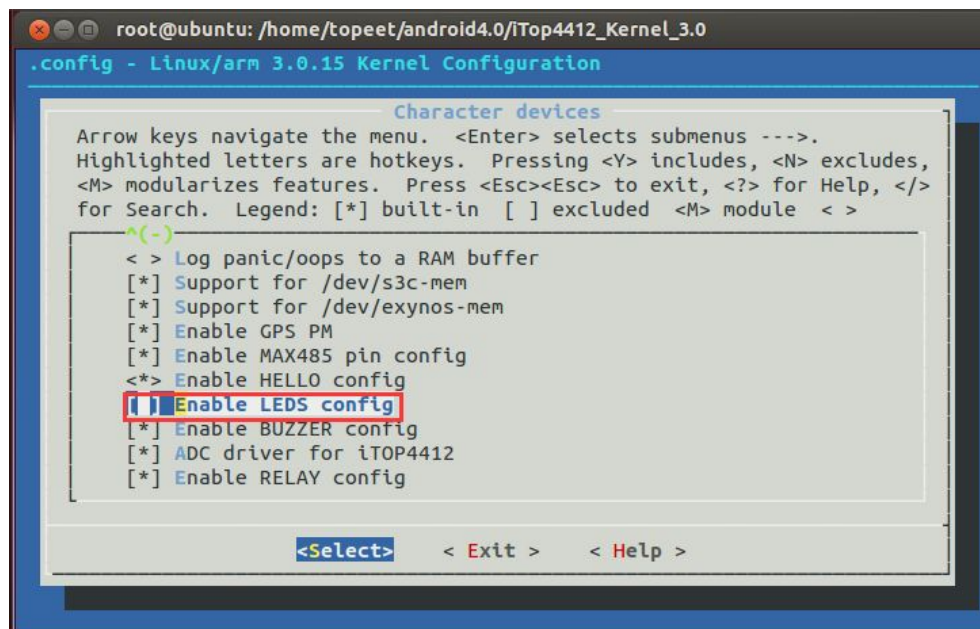
^(-)
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
v(+)

<Select>  < Exit >  < Help >
```

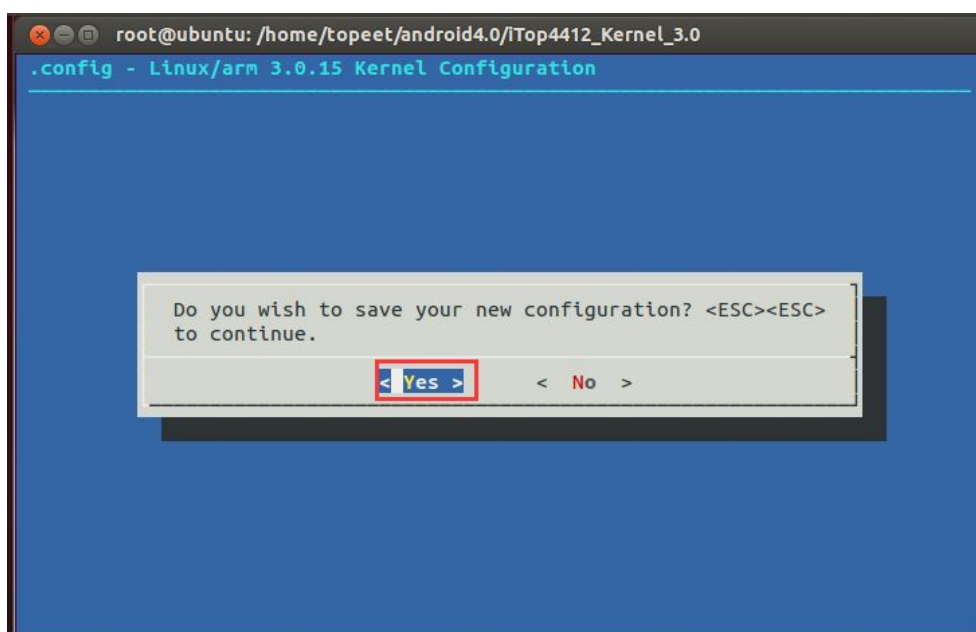
如下图所示，进入到 “Device Drivers ---> ” 选项，找到选项 “Character devices ---> ”。



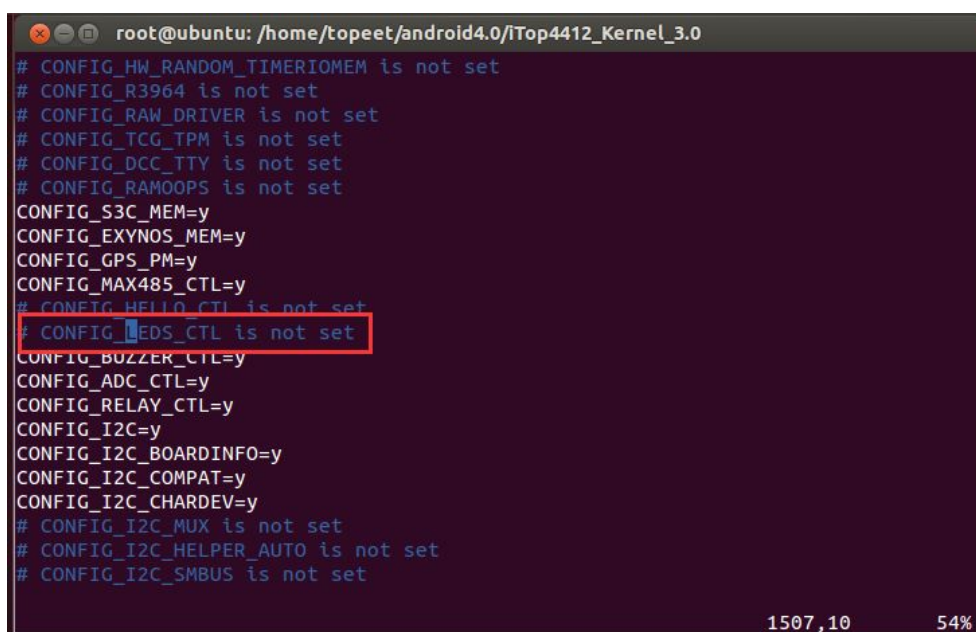
如下图所示，找到选项“Enable LEDS config”，将 LEDS 的配置关闭



退出，保存，生成新的“.config”文件。



保存退出后，打开 “.config” 文件，可以发现 “CONFIG_LEDS_CTL” 没有配置。



关闭 “.config” 文件，然后执行编译内核的命令 “make”，如下图所示。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
dcc_tty.c      itop4412_leds.c      nvram.c        sonypi.c
ds1302.c      itop4412_relay.c     nwbutton.c     tb0219.c
ds1620.c      itop4412_relay.o     nwbutton.h     tlclk.c
dsp56k.c      Kconfig              nwflash.c     toshiba.c
dtlk.c        lp.c                 pc8736x_gpio.c tpm
efirtc.c      Makefile             pcmcia         ttyprintk.c
exynos_mem.c  max485_ctl.c         ppdev.c        uv_mmtimer.c
exynos_mem.o  max485_ctl.o         ps3flash.c     viotape.c
generic_nvram.c mbc.c                ramoops.c      virtio_console.c
genrtc.c      mbc.h                random.c        xilinx_hwicap
gps.c         mem.c                 random.o
gps.h         mem.o                 raw.c
gps.o         misc.c                rtc.c
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# make menuconfig
scripts/kconfig/mconf Kconfig
#
# configuration written to .config
#
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# make
```

如下图所示，编译完成。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
UPD      include/generated/compile.h
CC        init/verston.o
LD        init/built-in.o
LD        .tmp_vmlinux1
KSYM      .tmp_kallsyms1.S
AS        .tmp_kallsyms1.o
LD        .tmp_vmlinux2
KSYM      .tmp_kallsyms2.S
AS        .tmp_kallsyms2.o
LD        vmlinux
SYSMAP    System.map
SYSMAP    .tmp_System.map
OBJCOPY   arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
GZIP      arch/arm/boot/compressed/piggy.gzip
AS        arch/arm/boot/compressed/piggy.gzip.o
SHIPPED   arch/arm/boot/compressed/lib1funcs.S
AS        arch/arm/boot/compressed/lib1funcs.o
LD        arch/arm/boot/compressed/vmlinux
OBJCOPY   arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
Building modules, stage 2.
MODPOST 3 modules
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0#
```

如下图所示，使用命令“ls drivers/char”查看 LEDS 驱动所在目录，发现并没有产生了中间文件。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
MODPOST 3 modules
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char
agp                hangcheck-timer.c  misc.c             rtc.c
apm-emulation.c    hpet.c             misc.o             s3c_mem.c
applicom.c          hw_random           mmtimer.c          s3c_mem.h
applicom.h          i8k.c              modules.builtin    s3c_mem.o
bfin-otp.c          ipmi                modules.order      scc.h
briq_panel.c        itop4412_adc.c     msm_smd_pkt.c      scx200_gpio.c
bsr.c               itop4412_adc.o     mspec.c            snsc.c
built-in.o          itop4412_buzzer.c  mwave              snsc_event.c
dcc_tty.c           itop4412_buzzer.o  nsc_gpio.c         snsc.h
ds1302.c            itop4412_leds.c    nvram.c            sonypi.c
ds1602.c            itop4412_relay.c   nwbutton.c         tb0219.c
dsp56k.c            itop4412_relay.o   nwbutton.h         tlclk.c
dtlk.c              Kconfig            nwflash.c          toshiba.c
efirtc.c            lp.c               pc8736x_gpio.c     tpm
exynos_mem.c        Makefile            pcmcia              ttyprintk.c
exynos_mem.o        max485_ctl.c        ppdev.c            uv_mmtimer.c
generic_nvram.c     max485_ctl.o        ps3flash.c         viotape.c
genrtc.c            mbc.c              ramoops.c          virtio_console.c
gps.c               mbc.h              random.c            xilinx_hwicap
gps.h               mem.c               random.o
gps.o               mem.o               raw.c
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0#
```

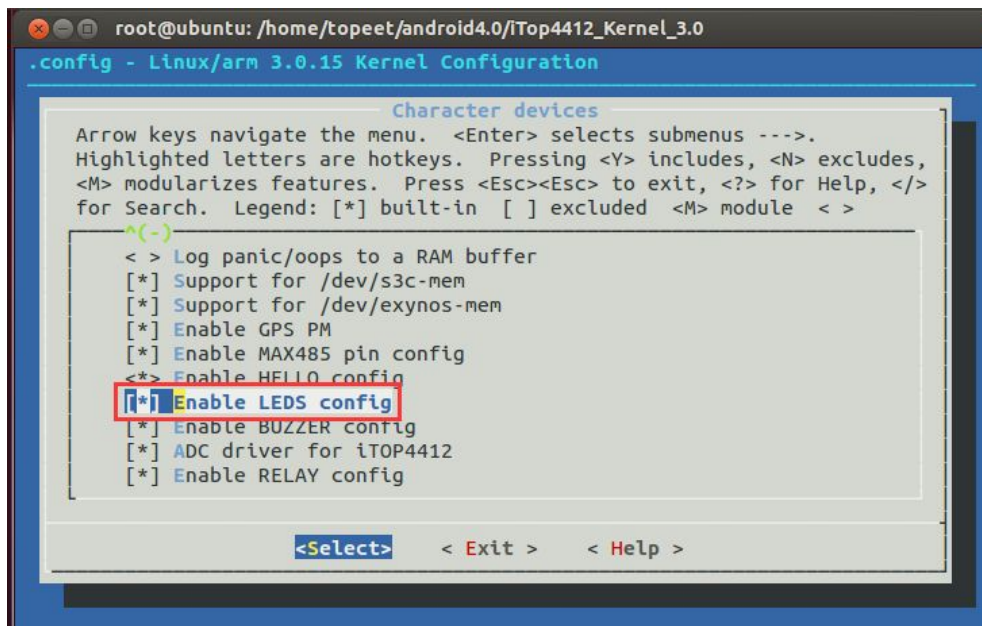
最后也可以将生成的文件“zImage”文件下载到开发板，会发现“/dev”目录中没有产生LEDS驱动的设备节点“leds”。

4.6.2 将LEDS驱动编译进内核

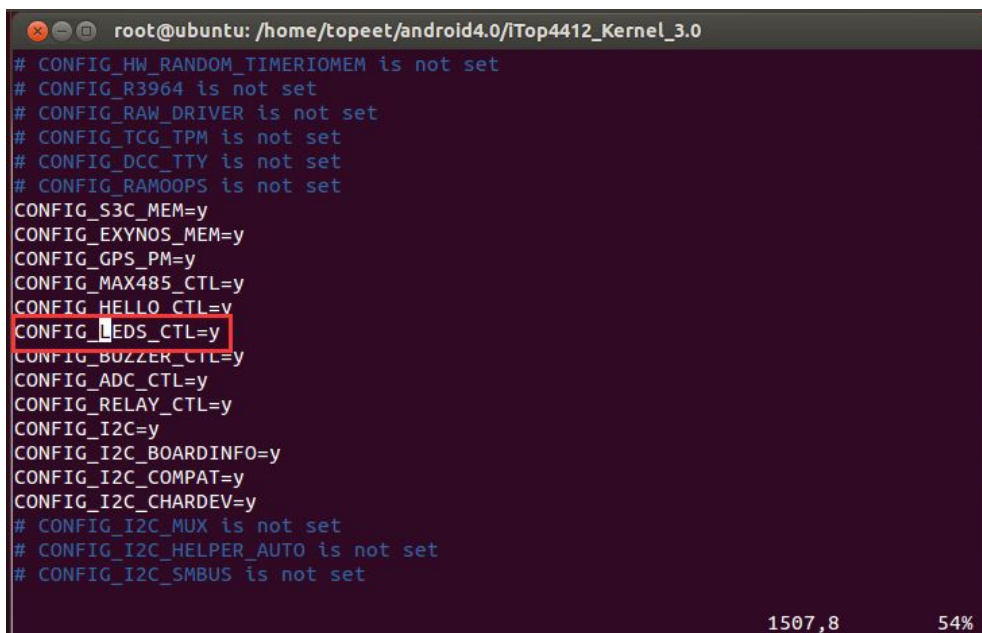
接上一小节，如下图所示，使用命令“make menuconfig”，打开menuconfig配置工具。

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
root@ubuntu:/home/topeet/android4.0/iTop4412_Kernel_3.0# make menuconfig
```

进入 “ Device Drivers ---> ” 选项，然后进入 “Character devices ---> ” 选项，然后如下图所示，配置上 “Enable LEDS config” 。



保存退出。打开 “.config” 文件，可以发现 “CONFIG_LEDS_CTL=y” 已经配置。



执行编译命令 “make” ，如下图所示，编译完成之后，使用命令 “ls drivers/char/” 可以看到 “itop4412_leds.o” 已经生成。

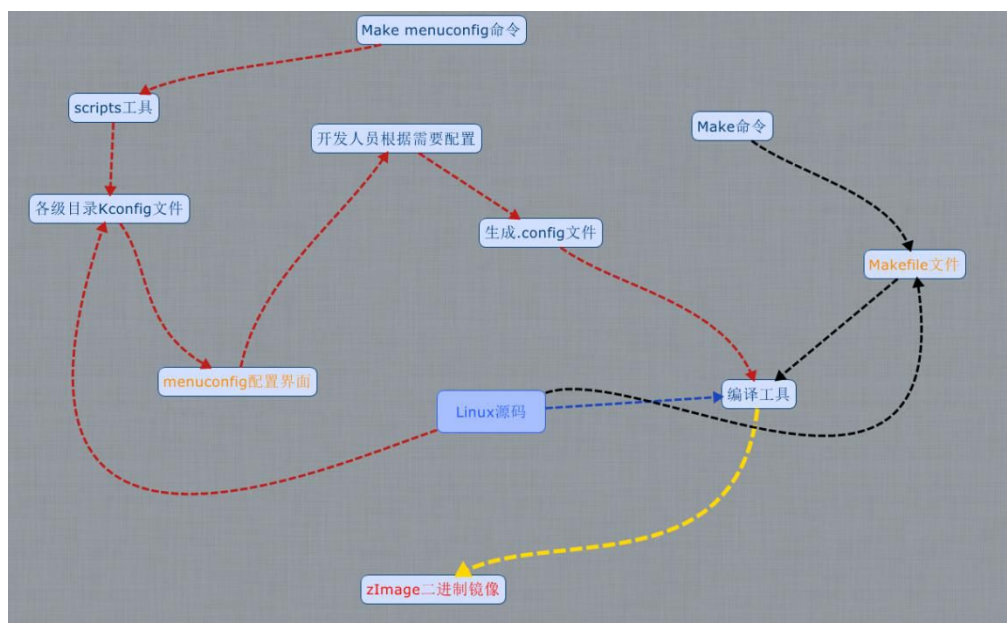

```
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0
MODPOST 3 modules
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0# ls drivers/char/
agp                hangcheck-timer.c  mem.o              raw.c
apm-emulation.c    hpet.c             misc.c             rtc.c
applicom.c          hw_random           misc.o             s3c_mem.c
applicom.h          i8k.c              mmtimer.c          s3c_mem.h
bfin-otp.c          ipmi                modules.builtin    s3c_mem.o
briq_panel.c        itop4412_adc.c      modules.order       scc.h
bsr.c               itop4412_adc.o      msm_snd_pkt.c      scx200_gpio.c
built-in.o          itop4412_buzzer.c   mspec.c            snsc.c
dcc_tty.c           itop4412_buzzer.o  mwave              snsc_event.c
ds1302.c            itop4412_leds.c     nsc_gpio.c         snsc.h
ds1620.c            itop4412_leds.o     nvram.c            sonypi.c
dsp56k.c            itop4412_relay.c    nwbutton.c         tb0219.c
dtlk.c              itop4412_relay.o    nwbutton.h         tclk.c
efirtc.c            Kconfig             nwflash.c          toshiba.c
exynos_mem.c        lp.c                pc8736x_gpio.c     tpm
exynos_mem.o        Makefile             pcmcia              ttyprintk.c
generic_nvram.c     max485_ctl.c         ppdev.c            uv_mmtimer.c
genrtc.c            max485_ctl.o         ps3flash.c         viotape.c
gps.c               mbc.c               ramoops.c          virtio_console.c
gps.h               mbc.h               random.c            xilinx_hwicap
gps.o               mem.c               random.o
root@ubuntu: /home/topeet/android4.0/iTop4412_Kernel_3.0#
```

最后也可以将生成的文件“zImage”文件下载到开发板，会发现“/dev”目录中产生 LEDS 驱动的设备节点“leds”，LED 灯的控制又恢复正常了。

4.7 编译流程图解

现在内核编译相关的知识已经全部介绍完，本节做一个小节。

下图已经将内核编译的全部过程包含。



如上图所示。

红色的线条表示配置文件 Kconfig 这一部分，在 Kconfig 中要定义针对具体驱动文件的宏变量。然后使用 menuconfig 工具生成新的 ".config" 文件。

黑色的线条表示编译文件 Makefile 这一部分，在 Makefile 中针对宏变量编译驱动文件。

执行 make 命令之后，调用 ".config" 文件，配合各级目录中的 Makefile 文件编译具体的驱动源代码，将源代码编译成 ".o" 中间文件。

当中间文件全部编译完成之后，编译工具会生成一个非常精炼的 "zImage" 二进制文件。