



博创科技
WWW.UP-TECH.COM

博创科技 嵌入互动

Linux内核配置、编译分析实验

博创科技

Linux内核配置、编译分析实验

- ◆ Linux内核的配置系统的组成
- ◆ Linux内核配置系统的结构
- ◆ Linux内核的编译
- ◆ Linux内核映象的烧写
- ◆ 在内核配置中添加一个编译模块

Linux内核的配置系统由三个部分组成：

- ◆ **Makefile:** 分布在 Linux 内核源代码中的 Makefile，定义 Linux 内核的编译规则
- ◆ **配置文件（config.in, Kconfig）：** 给用户提供了配置选择的功能
- ◆ **配置工具：**
 - 包括配置命令解释器（对配置脚本中使用的配置命令进行解释）
 - 配置用户界面：
 - ◆ 基于字符的界面 **make config**
 - ◆ 基于 **Ncurses** 的文本模式图形用户界面 **make menuconfig**
 - ◆ 基于 **Xwindows** 图形界面的用户配置界面 **make xconfig**

(1) 基于字符界面的 make config

```
[root@BC kernel-2410s-1netport1]# make config
rm -f include/asm-arm/arch include/asm-arm/proc
(cd include/asm-arm; ln -sf arch-s3c2410 arch; ln -sf proc-armv proc)
rm -f include/asm
(cd include; ln -sf asm-arm asm)
/bin/sh scripts/Configure arch/arm/config.in
#
# Using defaults found in .config
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?] y
Prompt for obsolete code/drivers (CONFIG_OBSOLETE) [N/y/?] N
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?] Y
Set version information on all module symbols (CONFIG_MODVERSIONS) [N/y/?] Y
Kernel module loader (CONFIG_KMOD) [Y/n/?] Y
*
* System Type
*
ARM system type (Anakin, Archimedes/A5000, Cirrus-CL-PS7500FE, CLPS711x/EP721x-based, Co-EBSA285, PX
A250/210-based, EBSA-110, Epxa10db, FootBridge, Integrator, LinkUp-L7200, Motorola-MX1ADS, RiscPC, S
A1100-based, S3C2400-based, S3C2410-based, Shark) [S3C2410-based]
defined CONFIG_ARCH_S3C2410
*
* S3C2410 Implementation
*
SMDK (MERI TECH BOARD) (CONFIG_S3C2410_SMDK) [Y/n/?]
change AIJI (CONFIG_SMDK_AIJI) [Y/n/?]
by threewater--1 (CONFIG_SMDK_THREEWATER1) [Y/n/?]
S3C2410 USB function support (CONFIG_S3C2410_USB) [M/n/y/?]
Support for S3C2410 USB character device emulation (CONFIG_S3C2410_USB_CHAR) [M/n/?]
```

(2) 基于 Ncurses 图形界面的 make menuconfig

Linux Kernel v2.4.18-rmk7-pxa1 Configuration

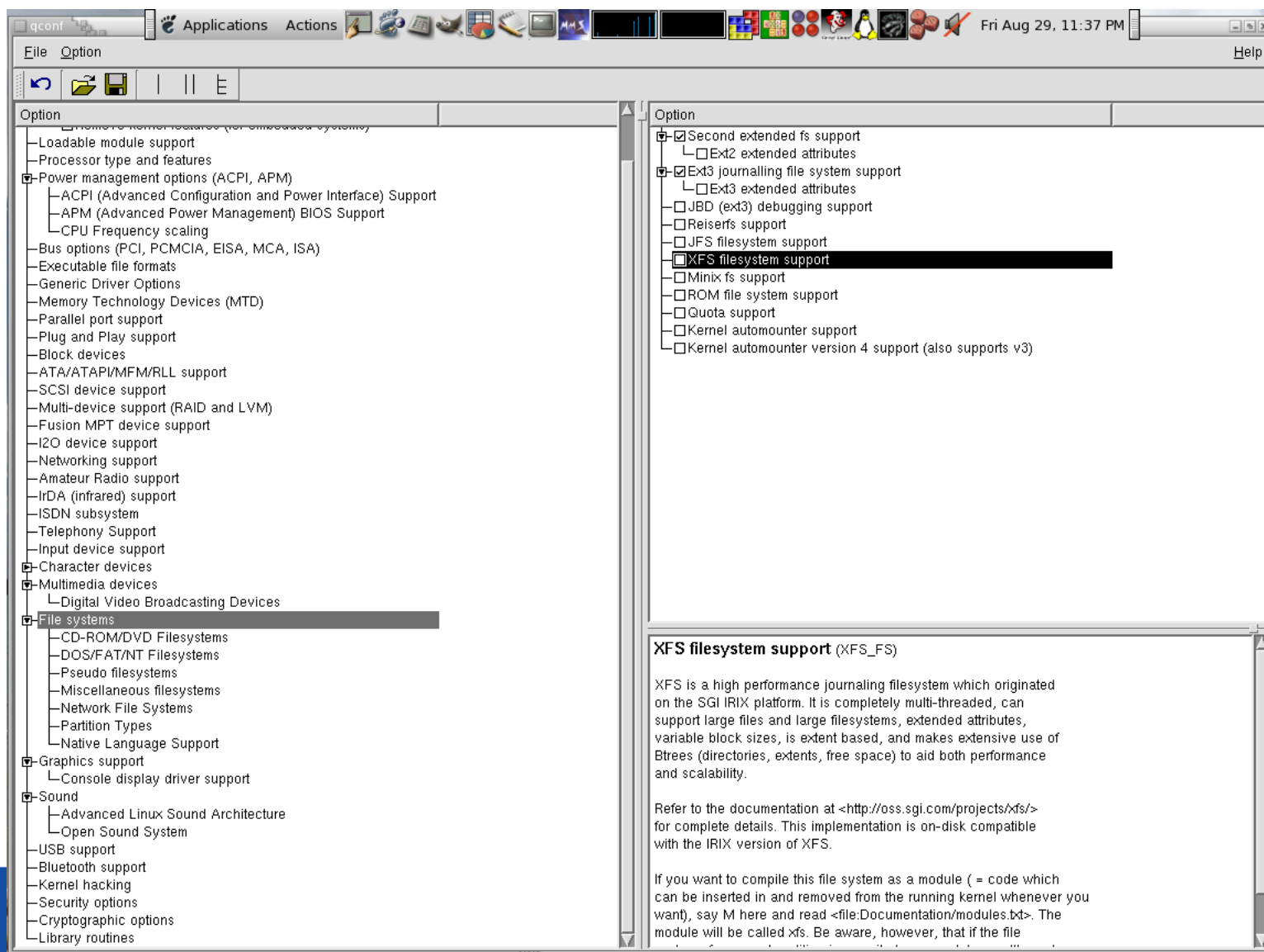
Main Menu

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. Legend: [*] built-in [] excluded <M> module < > module capable

```
Code maturity level options --->
Loadable module support --->
System Type --->
General setup --->
Parallel port support --->
Memory Technology Devices (MTD) --->
Plug and Play configuration --->
Block devices --->
Multi-device support (RAID and LVM) --->
Networking options --->
Network device support --->
Amateur Radio support --->
IrDA (infrared) support --->
ATA/IDE/MFM/RLL support --->
SD/MMC support --->
SCSI support --->
I2O device support --->
ISDN subsystem --->
Input core support --->
Character devices --->
Multimedia devices --->
File systems --->
Console drivers --->
Sound --->
- (+)
```

<Select> < Exit > < Help >

(3) 基于 Xwindows 图形界面的make xconfig



Linux内核配置系统的结构

◆ Makefile:

- ◆ 顶层 Makefile 读取 .config 中的配置选项

`CROSS_COMPILE` `?= /usr/local/arm-linux/bin/arm-linux-` 指定编译内核所使用的编译器

`include arch/(ARCH)/Makefile`, 包含了特定 CPU 体系结构下的 Makefile, 指定了平台相关的信息。

- ◆ 各个子目录下的 Makefile: 比如 `drivers/Makefile`, 负责所在子目录下源代码的编译管理。

- ◆ `.config`: 内核配置文件, 包含由用户选择的配置选项, 用来存放内核配置后的结果 (如 `make menuconfig`)

- ◆ `Rules.make`: 规则文件, 被所有的 Makefile 使用

Linux内核的编译

```
# make menuconfig      ----配置编译选项
# make dep             ----生成变量依赖关系信息.
# make clean           ----删除生成的模块和目标文件.
# make zImage          ----编译内核生成映像.
# make modules         ----编译模块.
# make modules_install ----安装编译完成的模块.
```

压缩内核影像所在路径:

arch/arm/boot/zImage

Linux内核的编译

- ◆ vmlinuz与vmlinux
- ◆ vmlinuz是可引导的、压缩的、可执行的内核。
- ◆ 老的zImage解压缩内核到低端内存（第一个640K），bzImage解压缩内核到高端内存（1M以上）。如果内核比较小，那么可以采用zImage或bzImage之一，两种方式引导的系统运行时是相同的。大的内核采用bzImage，不能采用zImage。
- ◆ vmlinux是未压缩的内核，vmlinuz是vmlinux的压缩文件

内核映像的烧写

◆ 两种烧写方式：

- ◆ 1、使用vivi中提供的xmodem协议下载
- ◆ 2、在开发板的linux系统启动后，使用imagewrite工具：
`imagewrite /dev/mtd/0 zImage:192k`

◆ 实验：编译并烧写内核影像

如何在内核配置中添加一个编译模块 (1)

- ◆ 1、在内核的驱动目录下编写驱动模块代码
本例中kernel-2410s/drivers/char /demo.o
- ◆ 2、在该级目录下的Config.in中添加对该模块的编译条件变量
如：CONFIG_S3C2410_DEMO
该变量可以设置为三种状态：
Y----将该功能模块编译进内核
N ---不将该功能模块编译进内核
M ---将该功能编译成模块的方式，可以在需要时动态插入到内核中的模块

如何在内核配置中添加一个编译模块 (2)

- ◆ 3、在Makefile中将编译选项与具体要编译的代码相关联，根据编译选项变量的状态决定编译
Object file lists.
obj-y :=
obj-m :=
obj-n :=
obj-\$(CONFIG_S3C2410_DEMO) += demo.o
- ◆ 4、使用make menuconfig对该功能模块进行配置，设置完成后，会生成更新的.config文件
- ◆ 拷贝demo.o，运行测试程序test_demo
- ◆ 实验：以[*]和[M] 方式编译demo驱动，并下载测试

具体实验步骤

- ◆ 1. 从ftp://10.61.14.240:5000 下载kernel.iso。
(内核源码下载地址为
<http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.24.tar.gz>)
- ◆ 2. 改变虚拟机的光盘模式为iso光盘挂载模式，并且挂载到/mnt/cdrom
- ◆ 3. 以root 用户名登陆
- ◆ 4. 复制linux-2.4.24.tar.gz到/usr/src目录下

具体实验步骤

- ◆ 5. #解压linux-2.4.24.tar.gz (tar zxvf linux-2.4.24.tar.gz)
- ◆ 6. 进入目录/usr/src/linux-2.4.24,，一定确定是在此目录下操作，否则以后的操作将无功而返。
- ◆ 7. 进行make mrproper操作。
这步操作的目的是清除原先此目录下残留的.config和.o文件（object文件），如果是第一次编译内核，那么这一步就可以省略，但是如果已经编译过多次内核的话，这一步一定要，不然以后出现很多小问题

具体实验步骤

◆ 8. 进行make menuconfig操作

此步骤为配置内核参数的过程。该步是编译内核过程中最繁琐的一步。要选择诸多参数。

“Y”—将该功能编译进内核；

“N”—不将该功能编译进内核；

“M”—将该功能编译成可以在需要时动态插入到内核的模块；

选择的原理是将与内核其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于缩减内核，减少内核消耗的内存。与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。

这里的参数我基本上都设置了，其好处可以从编译完后重启进入的过程中看出来，启动过程中基本上没有什么错误。因此，设置好内核参数很关键。设置的一些主要参数有：

（1）、Loadable module support （2）、General setup （3）、
Networking options （4）、USB Support

具体实验步骤

◆ 9. 进行make dep 操作

对内核源代码的文件进行完整性和依赖性进行检查，确保关键文件在正确的位置。

◆ 10. 进行 make clean 操作 清除一些不必要的文件

◆ 11. 进行 make bzImage 操作 该步骤即生成新内核的步骤。所费时间较长。

◆ 12. 进行 make modules 操作 该步骤把在make menuconfig里边所选择的设置,全部编译成模块。

具体实验步骤

- ◆ 13. 进行 `make modules_install` 操作
安装module的过程。
- ◆ 14. `cp /usr/src/linux/arch/arm/boot/zImage
/tftpboot/ .`
- ◆ 15. 在VIVI下通过串口或tftp协议下载内核映像文件

◇ 参考资料

linux内核编译与裁剪的选项的解释请参考实验word文档，每一项都介绍得很详细。



博创科技
WWW.UP-TECH.COM

博创科技 嵌入互动



感谢您参加博创嵌入式教学培训！