

附录三 内核配置详细说明

对 linux 内核进行编译之前，首先要使用命令 ‘make menuconfig’ 对 linux 的编译选项进行配置。

下面详细讲解使用这个命令时所遇到的重要配置参数：

第一部分

Code maturity level options ---> 代码成熟等级选项

[] Prompt for development and/or incomplete code/drivers

默认情况下是选择的，这将会在设置界面中显示还在开发或者还没有完成的代码与驱动。你应该选择它，因为有许多设备可能必需选择这个选项才能进行配置，实际上它是安全的。

[] Select only drivers expected to compile cleanly

选择这个选项你将不会看到一些已知的存在问题的驱动程序选项，默认的情况下也是选择的。如果你有设备没有找到驱动选项，你可以将这一项去掉，或许就可以找到相关驱动了，不过它可能是有 BUG 的。

第二部分

General setup ---> 常规安装选项

Local version - append to kernel release

[*] Support for paging of anonymous memory (swap)

这个选项将使你的内核支持虚拟内存，也就是让你的计算机好象拥有比实际内存更多多的内存空间用来执行很大的程序。默认是选择的。

[*] System V IPC

是组系统调用及函数库，它能让程序彼此间同步进行交换信息。某些程序以及 DOS 模拟环境都需要它。为进程提供通信机制，这将使系统中各进程间有交换信息与保持同步的能力。有些程序只有在选 Y 的情况下才能运行，所以不用考虑，这里一定要选。

[] POSIX Message Queues

POSIX 消息队列，这是 POSIX IPC 中的一部分

[] BSD Process Accounting

一般用户所执行的程序，可通知内核将程序统计数据写成文件，详细记录相关信息。

[*] Sysctl support

此项功能可在不重新编译内核或是重新开机的前提下，动态变更某些特定的内核参数和变量，赋予内核较大的弹性。除非内存太少或是编译出来的内核是给安装、救援磁盘所使用，否则这个选项一定要选上。这将提供一个接口让你可以动态的更改一些核心参数与变量，而不需要重新启动系统。打开这个选项将会增加内核的体积至少 8KB。如果你的内核仅用制作安装与恢复系统系统盘那么可以不选，以减少对内存的占用。

[] Auditing support

审计支持，用于和内核的某些子模块同时工作，例如 SELinux。只有选择此项及它的子项，才能调用有关审计的系统调用。

--- Support for hot-pluggable devices

是否支持热插拔的选项，肯定要选上。不然 USB、PCMCIA 等这些设备都用不了。

[*] Kernel Userspace Events

内核中分为系统区和用户区，这里系统区和用户区进行通讯的一种方式，选上。

[] Kernel .config support

这将会把内核的配置信息与相关的文档说明编译进内核中，以后可以使用一些工具来提取它用来重新构建内核，一般不用选它。

[] Configure standard kernel features (for small systems) --->

这是为了编译某些特殊用途的内核使用的，例如引导盘系统。通常你可以不选择这一选项，你也不用对它下面的子项操心了。

第三部分

Loadable module support ---> 可引导模块支持 建议作为模块加入内核

[*] Enable loadable module support

这个选项可以让你的内核支持模块，模块是什么呢？模块是一小段代码，编译后可在系统内核运行时动态的加入内核，从而为内核增加一些特性或是对某种硬件进行支持。一般一些不常用到的驱动或特性可以编译为模块以减少内核的体积。在运行时可以使用 modprobe 命令来加载它到内核中去(在不需要时还可以移除它)。一些特性是否编译为模块的原则是，不常使用的，特别是在系统启动时不需要的驱动可以将其编译为模块，如果是一些在系统启动时就要用到的驱动比如说文件系统，系统总线的支持就不要编为模块了，否在无法启动系统。

[*] Module unloading

这个选项可以让你卸载不再使用的模块，如果不选的话你将不能卸载任何模块(有些模块一旦加载就不能卸载，不管是否选择了这个选项)。不选择这个选项会让你的内核体积减小一点。

[*] Forced module unloading

允许强制卸载正在使用中的模块(比较危险)

[] Module versioning support (EXPERIMENTAL)

允许使用其他内核版本的模块(可能会出问题)

[] Source checksum for all modules

这个功能是为了防止更改了内核模块的代码但忘记更改版本号而造成版本冲突。我估计现在没有哪家公司在开发中还没使用版本控制工具，所以不需要这项了。如果你不是自己写内核模块，那就更不需要这一选项了。

[*] Automatic kernel module loading

一般情况下，如果用户的内核在某些任务中要使用一些被编译为模块的驱动或特性时，我们要先使用 modprobe 命令来加载它，内核才能使用。不过，如果你选择了这个选项，在

内核需要一些模块时它可以自动调用 modprobe 命令来加载需要的模块，这是个很棒的特性，当然也要选 Y 喽。

第四部分

System Type ---> 系统类型

ARM system type (Samsung S3C2410) ---> ARM 系统，基于 S3C2410
S3C24XX Implementations ---> 基于 S3C24xx 架构的实现

--- S3C2410 Boot

--- S3C2410 Setup

[*] S3C2410 DMA support 支持 DM 功能

[] S3C2410 DMA support debug 支持 DMA 调试功能

(0) S3C2410 UART to use for low-level messages

--- Processor Type 处理器类型

--- Processor Features 处理器特性

[*] Support Thumb user binaries 支持 Thumb 指令集

[] Disable I-Cache 禁止指令缓存 (Instruction Cache , I-

Cache) 。

[] Disable D-Cache 禁止数据缓存 (Data Cache , D-Cache)

[] Force write through D-cache 强制回写数据缓存

第五部分

Bus support ---> 总线支持

PCCARD (PCMCIA/CardBus) support --->

< > PCCard (PCMCIA/CardBus) support

PCMCIA 卡支持

第六部分

Kernel Features

- ☐ Symmetric Multi-Processing (EXPERIMENTAL)
- ☐ Preemptible Kernel (EXPERIMENTAL)

第七部分

Boot options --->

- (0x0) Compressed ROM boot loader base address
- (0x0) Compressed ROM boot loader BSS address
- (console=ttyS0 , 115200) Default kernel command string
- ☐ Kernel Execute-In-Place from ROM

第八部分

Floating point emulation --->

--- At least one emulation must be selected

- ☒ NWFPE math emulation
- ☐ Support extended precision
- ☐ FastFPE math emulation (EXPERIMENTAL)

第九部分

Userspace binary formats ---> 支持的可执行文件格式

- ☒ Kernel support for ELF binaries

ELF 是开放平台下最常用的二进制文件格式，支持动态连接，支持不同的硬件平台.除非你知道自己在做什么，否则必选

< > Kernel support for a.out and ECOFF binaries

这是早期 UNIX 系统的可执行文件格式，目前已经被 ELF 格式取代。

< > Kernel support for MISC binaries

允许插入二进制的封装层到内核中，使用 Java，.NET，Python，Lisp 等语言编写的程序时需要它

< > RISC OS personality

第十部分

Power management options ---> 电源管理选项

[] Power Management support

电源管理有 APM 和 ACPI 两种标准且不能同时使用.即使关闭该选项，X86 上运行的 Linux 也会在空闲时发出 HLT 指令将 CPU 进入睡眠状态

第十一部分

Device Drivers ---> 设备驱动程序

Generic Driver Options --->

驱动程序通用选项

[*] Select only drivers that don't need compile-time external firmware

只显示那些不需要内核对外部设备的固件作 map 支持的驱动程序，除非你有某些怪异硬件，否则请选上

[*] Prevent firmware from being built

不编译固件.固件一般是随硬件的驱动程序提供的，仅在更新固件的时候才需要重新编译.建议选上

<*> Hotplug firmware loading support

加载热插拔固件支持， 在内核树之外编译的模块可能需要它

[] Driver Core verbose debug messages

让驱动程序核心在系统日志中产生冗长的调试信息， 仅供调试

Memory Technology Devices (MTD) --->

特殊的存储技术装置， 如常用于数码相机或嵌入式系统的闪存卡

Parallel port support --->

并口支持(传统的打印机接口)

Plug and Play support --->

即插即用支持， 若未选则应当在 BIOS 中关闭"PnP OS".这里的选项与 PCI 设备无关

Block devices --->

ATA/ATAPI/MFM/RLL support --->

这个是有关各种接口的硬盘/光驱/磁带/软盘支持的， 内容太多了， 使用缺省的选项吧，
如果你使用了比较特殊的设备， 比如 PCMCIA 等， 就到里面自己找相应的选项吧

SCSI device support --->

SCSI 设备

Multi-device support (RAID and LVM) --->

多设备支持(RAID 和 LVM).RAID 和 LVM 的功能是使多个物理设备组建成一个单独的逻辑磁盘

Fusion MPT device support --->

Fusion MPT 设备支持

IEEE 1394 (FireWire) support --->

IEEE 1394(火线)

I2O device support --->

I2O(智能 IO)设备使用专门的 I/O 处理器负责中断处理/缓冲存取/数据传输等烦琐任务以减少 CPU 占用， 一般的主板上没这种东西

Networking support --->

网络支持

ISDN subsystem --->

综合业务数字网(Integrated Service Digital Network)

Input device support --->

输入设备

Character devices --->

字符设备

I2C support --->

I2C 是 Philips 极力推动的微控制应用中使用的低速串行总线协议, 可用于监控电压/风扇转速/温度等.SMBus(系统管理总线)是 I2C 的子集.除硬件传感器外"Video For Linux"也需要该模块的支持

Misc devices --->

Multimedia devices --->

多媒体设备

Graphics support --->

图形设备/显卡支持

Sound --->

声卡

USB support --->

USB 支持

MMC/SD Card support --->

MMC/SD 卡支持

第十二部分

File systems ---> 文件系统

<*> Second extended fs support

Ext2 文件系统是 Linux 的标准文件系统，擅长处理稀疏文件

[*] Ext2 extended attributes

Ext2 文件系统扩展属性(与 inode 关联的 name:value 对)支持

[*] Ext2 POSIX Access Control Lists

POSIX ACL(访问控制列表)支持，可以更精细的针对每个用户进行访问控制，需要外部库和程序的支持

[] Ext2 Security Labels

安全标签允许选择使用不同的安全模型实现(如 SELinux)的访问控制模型，如果你没有使用需要扩展属性的安全模型就别选

<*> Ext3 journalling file system support

Ext3 性能平庸，使用 journal 日志模式时数据完整性非常好(但怪异的是此时多线程并发读写速度却最快)

[*] Ext3 extended attributes

Ext3 文件系统扩展属性(与 inode 关联的 name:value 对)支持

[] Ext3 POSIX Access Control Lists

POSIX ACL(访问控制列表)支持，可以更精细的针对每个用户进行访问控制，需要外部库和程序的支持

[] Ext3 Security Labels

安全标签允许选择使用不同的安全模型实现(如 SELinux)的访问控制模型，如果你没有使用需要扩展属性的安全模型就别选

[] JBD (ext3) debugging support 仅供开发者使用

< > Reiserfs support

性能几乎全面超越 Ext2(处理稀疏文件比 Ext2 慢), 小文件(小于 4k)性能非常突出, 创建和删除文件速度最快, 处理大量目录和文件(5k-20k)时仍然非常迅速.日志模式建议使用 Ordered, 追求极速可使用 Writeback 模式, 追求安全可使用 Journal 模式.建议使用 noatime, notail 选项挂载分区以提高速度和避免 bug.用于 NFS 和磁盘限额时需要额外的补丁

< > JFS filesystem support

IBM 的 JFS 文件系统

XFS support --->

碎片最少, 多线程并发读写最佳, 大文件(>64k)性能最佳, 创建和删除文件速度较慢.由于 XFS 在内存中缓存尽可能多的数据且仅当内存不足时才会将数据刷到磁盘, 所以应当仅在确保电力供应不会中断的情况下才使用 XFS

< > Minix fs support

老古董文件系统

< > ROM file system support

用于嵌入式系统的内存文件系统的支持

[] Quota support

磁盘配额支持, 限制某个用户或者某组用户的磁盘占用空间, Ext2/Ext3/Reiserfs 都支持它

<*> Kernel automounter support

内核自动加载远程文件系统(v3, 就算选也不选这个旧的)

< > Kernel automounter version 4 support (also supports v3)

新的(v4)的内核自动加载远程文件系统的支持, 也支持 v3

CD-ROM/DVD Filesystems --->

CD-ROM/DVD 文件系统

DOS/FAT/NT Filesystems --->

DOS/Windows 的文件系统

Pseudo filesystems --->

伪文件系统

Miscellaneous filesystems --->

非主流的杂项文件系统

Network File Systems --->

网络文件系统

Partition Types --->

高级磁盘分区类型，不确定可以全不选

Native Language Support --->

本地语言支持.如果你仅仅使用几种主流的 Linux 文件系统(ext2/3/4 , Reiserfs , JFS , XFS) , 就不需要这个东西.但是如果你需要使用 FAT/NTFS 分区的话 , 就需要这个东西了.

第十三部分

Profiling support --->

[] Profiling support (EXPERIMENTAL)

对系统的活动进行分析，仅供内核开发者使用

第十四部分

Kernel hacking ---> 内核 hack 选项，普通用户是用不着这个功能的

[] Show timing information on printk

在 printk 的输出中包含时间信息，可以用来分析内核启动过程各步骤所用时间

[*] Kernel debugging

不是内核开发者的别选

☐ Magic SysRq key

不懂的千万别选

(14) Kernel log buffer size (16 => 64KB , 17 => 128KB)

☐ Collect scheduler statistics

☐ Debug memory allocations

☐ Spinlock debugging

☐ Sleep-inside-spinlock checking

☐ kobject debugging

☐ Compile the kernel with debug info

☐ Debug Filesystem

☒ Verbose user fault messages

☐ Wait queue debugging

☐ Verbose kernel error messages

☒ Kernel low-level debugging functions

☐ Kernel low-level debugging via EmbeddedICE DCC channel

☒ Kernel low-level debugging messages via S3C2410 UART

(0) S3C2410 UART to use for low-level debug

第十五部分

Security options --->安全选项，这里的选项不明白的建议不要选，否则有可能弄巧成拙。

☐ Enable access key retention support

在内核中保留 authentication token 和 access key

☐ Enable different security models

允许内核选择不同的安全模型，如果未选中则内核将使用默认的安全模型

第十六部分

Cryptographic options ---> 加密选项

[*[*] Cryptographic API

提供核心的加密 API 支持.这里的加密算法被广泛的应用于驱动程序通信协议等机制中.子选项可以全不选,内核中若有其他部分依赖它,会自动选上

[] HMAC support

为 IPsec 所必须,可为 PPPoE 提供压缩支持

< > Null algorithms

NULL 加密算法(什么也不做),用于 IPsec 协议的封装安全载荷模块(ESP)

< > MD4 digest algorithm

老旧的摘要算法,已经过时

<*> MD5 digest algorithm

主流摘要算法,128 位(已被中国山东大学王小云攻破,可以快速找到碰撞)

< > SHA1 digest algorithm

主流摘要算法,160 位(已被中国山东大学王小云攻破,可以快速找到碰撞),速度与 MD5 相当

< > SHA256 digest algorithm

更好的摘要算法,256 位,速度较 SHA1 稍慢

< > SHA384 and SHA512 digest algorithms

更好的摘要算法,384/512 位,速度大约只有 SHA1 的 40-50%

< > Whirlpool digest algorithms

最安全的摘要算法,512 位,已被列入 ISO 标准,目前最新版本为 3.0(2003 年发布)

< > Tiger digest algorithms

号称最快的摘要算法,192 位,专门为 64 位 CPU 进行了优化

<*> DES and Triple DES EDE cipher algorithms

老迈的(DES)和尚佳的(Triple DES)对称加密算法

< > Blowfish cipher algorithm

又老又慢的对称加密算法

< > Twofish cipher algorithm

很强的对称加密算法，使用较广

< > Serpent cipher algorithm

很强的对称加密算法

< > AES cipher algorithms

最佳的对称加密算法(Rijndael)，128/192/256 位，强度最高，快速且节省内存

< > CAST5 (CAST-128) cipher algorithm

对称加密算法

< > CAST6 (CAST-256) cipher algorithm

对称加密算法

< > TEA and XTEA cipher algorithms

较弱的对称加密算法

< > ARC4 cipher algorithm

脆弱的流对称加密算法

< > Khazad cipher algorithm

对称加密算法

< > Anubis cipher algorithm

对称加密算法

< > Deflate compression algorithm

压缩算法，当在 IPsec 中使用 IPCOMP 协议时才需要

< > Michael MIC keyed digest algorithm

摘要算法，仅仅用于校验 iSCSI 设备传输的数据，因为算法本身比较脆弱

< > CRC32c CRC algorithm

摘要算法，可用于校验 iSCSI 设备传输的数据

< > Testing module

快速且丑陋的测试模块

Hardware crypto devices --->

仅有 VIA C7 系列处理器支持硬件加密(VIA PadLock 高级加密引擎)

第十七部分

Library routines ---> 库子程序

--- CRC-CCITT functions

传送 8-bit 字符，欧洲标准

--- CRC32 functions

用于点对点的同步数据传输中，传输网络数据包所必须的

<*> CRC32c (Castagnoli , et al) Cyclic Redundancy-Check

用于点对点的同步数据传输中，比如 iSCSI 设备

第十八部分

Load an Alternate Configuration File

读入一个外部配置文件

Save Configuration to an Alternate File

将配置保存到一个外部文件