

Linux[®] Kernel Configuration
中文手册(2.6.28)

刘少侠

中国·December 10, 2008

前言

这是关于 Linux 内核配置菜单选项的书。

Linux 发展到今天，已经是版本2.6.28。为了让广大 Linux 爱好者能够更加清楚的了解 Linux 内核的配置，笔者在业余时间里翻译了它的部分配置菜单。

由于本人水平有限，且时间仓促，不足之处在所难免。欢迎广大读者批评指正，共同提高。

作者：刘少侠
December 10, 2008

Contents

I	总览	1
II	开始配置菜单	7
1	General setup 常规设置	9
1.1	Prompt for development and/or incomplete code/drivers	9
1.2	Local version - append to kernel release	9
1.3	Automatically append version information to the version string . . .	9
1.4	Support for paging of anonymous memory (swap)	10
1.5	System V IPC	10
1.6	POSIX Message Queues	10
1.7	BSD Process Accounting	11
1.8	Export task/process statistics through netlink	11
1.9	Auditing support	12
1.10	Kernel .config support	12
1.11	Kernel log buffer size	12
1.12	Control Group support	13
1.13	Group CPU scheduler	14
1.14	Simple CPU accounting cgroup subsystem	14
1.15	Resource counters	15
1.16	Memory Resource Controller for Control Groups	15
1.17	Create deprecated sysfs file	15
1.18	Kernel→user space relay support (formerly relayfs)	15
1.19	Namespaces support	15
1.20	Initial RAM filesystem and RAM disk support	16
1.21	Optimize for size	17
1.22	Configure standard kernel features (for small systems)	17
1.23	Disable heap randomization	17
1.24	Choose SLAB allocator	18
1.25	Profiling support	18

1.26	Activate markers	18
1.27	OProfile system profiling	18
1.28	Kprobes	18
2	Loadable module support 可加载模块支持	21
2.1	Enable loadable module support	21
2.2	Forced module loading	21
2.3	Module unloading	21
2.4	Module versioning support	22
2.5	Source checksum for all modules	22
3	Enable the block layer 允许块设备	23
3.1	Support for Large Block Devices	23
3.2	Support for tracing block io actions	23
3.3	Support for Large Single Files	23
3.4	Block layer SG support v4	24
3.5	Block layer data integrity support	24
3.6	I/O Schedulers	24
4	Processor type and features 处理器类型及特性	25
4.1	Tickless System	25
4.2	High Resolution Timer Support	25
4.3	Symmetric multi-processing support (SMP)	25
4.4	Enable MPS table	26
4.5	Subarchitecture Type	26
4.6	RDC R-321x SoC	26
4.7	Single-depth WCHAN output	27
4.8	Paravirtualized guest support	27
4.9	Memtest	27
4.10	Processor family	27
4.11	Generic x86 support	28
4.12	Branch Trace Store	29
4.13	HPET Timer Support	29
4.14	Maximum number of CPUs	29
4.15	SMT (Hyperthreading) scheduler support	29
4.16	Multi-core scheduler support	29
4.17	Preemptible Model	30
4.18	Local APIC support on uniprocessors	30
4.19	Machine Check Exception	31

4.20	Toshiba Laptop support	31
4.21	Dell laptop support	31
4.22	Enable X86 board specific fixups for reboot	32
4.23	/dev/cpu/microcode - microcode support	32
4.24	/dev/cpu/*/msr - Model-specific register support	33
4.25	/dev/cpu/*/cpuid - CPU information support	33
4.26	High Memory Support	33
4.27	PAE Support	34
4.28	Add LRU list to track non-evictable pages	34
4.29	Check for low memory corruption	34
4.30	Reserve low 64K of RAM on AMI/Phoenix BIOSen	34
4.31	Math emulation	35
4.32	MTRR (Memory Type Range Register) support	35
4.33	EFI runtime service support	36
4.34	Enable kernel irq balancing	36
4.35	Enable seccomp to safely compute untrusted bytecode (SECCOMP)	36
4.36	Timer frequency	36
4.37	kexec system call	37
4.38	Alignment value to which kernel should be aligned	37
4.39	Compat VDSO support (COMPAT_VDSO)	38
4.40	Built-in kernel command line	38
5	Power management options 电源管理选项	39
5.1	Power Management support	39
5.2	Suspend to RAM and standby	39
5.3	Hibernation	40
5.4	ACPI Support	41
5.5	APM (Advanced Power Management) BIOS Support	44
5.6	CPU Frequency scaling	45
5.7	CPU idle PM support	48
5.8	Memory power savings	48
6	Bus options 总线选项	49
6.1	PCI support	49
6.2	PCI Express support	50
6.3	Message Signaled Interrupts(MSI and MSI-X)	50
6.4	Enable deprecated pci_find_* API	50
6.5	PCI Debugging	50
6.6	Interrupts on hypertransport devices	50

6.7	ISA support	51
6.8	MCA support	51
6.9	NatSemi SCx200 support	51
6.10	One Laptop Per Child support	51
6.11	PCCARD (PCMCIA/CardBus) support	51
6.12	PCI Hotplug Support	52
7	Executable file formats 可执行文件格式	53
7.1	Kernel support for ELF binaries	53
7.2	Write ELF core dumps with partial segments	53
7.3	Kernel support for a.out and ECOFF binaries	54
7.4	Kernel support for MISC binaries	54
8	Networking support 网络支持	55
8.1	Networking options	55
8.2	Amateur Radio support	67
8.3	CAN bus subsystem support	67
8.4	IrDA (infrared) subsystem support	67
8.5	Bluetooth subsystem support	67
8.6	Phonet protocols family	68
8.7	Wireless	68
8.8	RF switch subsystem support	69
9	Device Drivers 设备驱动	71
9.1	Generic Driver Options	72
9.2	Connector - unified userspace <-> kernelspace linker	73
9.3	Memory Technology Devices (MTD) support	73
9.4	Parallel port support	74
9.5	Plug and Play support	75
9.6	Block devices	75
9.7	Misc devices	79
9.8	ATA/ATAPI/MFM/RLL support	80
9.9	SCSI device support	87
9.10	Serial ATA and Parallel ATA drivers	89
9.11	Multiple devices driver support(RAID and LVM)	91
9.12	Fusion MPT device support	92
9.13	IEEE 1394 (FireWire) support	92
9.14	I2O device support	93
9.15	Macintosh device drivers	94

9.16	Network device support	94
9.17	ISDN support	97
9.18	Telephony support	97
9.19	Input device support	97
9.20	Character devices	100
9.21	I2C support	104
9.22	SPI support	104
9.23	GPIO Support	104
9.24	Dallas's I-wire support	105
9.25	Power supply class support	105
9.26	Hardware Monitoring support	105
9.27	Generic Thermal sysfs driver	106
9.28	Watchdog Timer Support	106
9.29	Sonics Silicon Backplane	106
9.30	Multifunction device drivers	107
9.31	Voltage and Current regulators	107
9.32	Multimedia devices	107
9.33	Graphics support	108
9.34	Sound	110
9.35	HID Devices	114
9.36	USB support	115
9.37	Ultra Wideband devices	121
9.38	MMC/SD card support	121
9.39	Sony MemoryStick card support	121
9.40	LED Support	121
9.41	Accessibility support	122
9.42	InfiniBand support	122
9.43	Real Time Clock	122
9.44	DMA Engine support	122
9.45	Userspace I/O drivers	122
9.46	Staging drivers	123
10	Firmware Drivers 固件驱动	125
10.1	BIOS Enhanced Disk Drive calls determine boot disk	125
10.2	EFI Variable Support via sysfs	125
10.3	BIOS update support for DELL systems via sysfs	125
10.4	Dell Systems Management Base Driver	126
10.5	Export DMI identification via sysfs to userspace	126
10.6	iSCSI Boot Firmware Table Attributes	126

11 File systems 文件系统	127
11.1 Second extended fs support	127
11.2 Ext3 journalling file system support	128
11.3 The Extended 4 (ext4) filesystem	129
11.4 JBD (ext3) debugging support	130
11.5 JBD2 (ext4) debugging support	130
11.6 Reiserfs support	130
11.7 JFS filesystem support	132
11.8 XFS filesystem support	132
11.9 GFS2 file system support	133
11.10 OCFS2 file system support	134
11.11 Dnotify support	135
11.12 Inotify file change notification support	135
11.13 Quota support	135
11.14 Kernel automounter support	136
11.15 Kernel automounter version 4 support	136
11.16 Filesystem in Userspace support	137
11.17 CD-ROM/DVD Filesystems	137
11.18 DOS/FAT/NT Filesystems	138
11.19 Pseudo filesystems	140
11.20 Miscellaneous filesystems	141
11.21 Network File Systems	147
11.22 Partition Types	154
11.23 Native Language Support	154
11.24 Distributed Lock Manager (DLM)	157
12 Kernel hacking 内核调试	159
12.1 Show timing information on printk	159
12.2 Enable __deprecated logic	159
12.3 Enable __must_check logic	159
12.4 Warn for stack frames larger than (2048)	159
12.5 Magic SysRq key	160
12.6 Enable unused/obsolete exported symbols	161
12.7 Debug Filesystem	162
12.8 Run 'make headers_check' when building vmlinux	162
12.9 Kernel debugging	162
12.10 SLUB debugging on by default	163
12.11 Enable SLUB performance statistics	164
12.12 RT Mutex debugging, deadlock detection	164

12.13 Built-in scriptable tester for rt-mutexes	164
12.14 Spinlock and rw-lock debugging: basic checks	164
12.15 Mutex debugging: basic checks	164
12.16 Lock debugging: detect incorrect freeing of live locks	164
12.17 Lock debugging: prove locking correctness	165
12.18 Lock usage statistics	165
12.19 Spinlock debugging: sleep-inside-spinlock checking	165
12.20 Locking API boot-time self-tests	165
12.21 kobject debugging	166
12.22 Compile the kernel with debug info	166
12.23 Debug VM	166
12.24 Debug VM translations	166
12.25 Debug filesystem writers count	166
12.26 Debug linked list manipulation	167
12.27 Debug SG table operations	167
12.28 Compile the kernel with frame pointers	167
12.29 Delay each boot printk message by N milliseconds	167
12.30 torture tests for RCU	167
12.31 Check for stalled CPUs delaying RCU grace periods	168
12.32 Kprobes sanity tests	168
12.33 Self test for the backtrace code	168
12.34 Force extended block device numbers and spread them	168
12.35 Linux Kernel Dump Test Tool Module	169
12.36 Fault-injection framework	169
12.37 Latency measuring infrastructure	169
12.38 Sysctl checks	169
12.39 Tracers	169
12.40 Remote debugging over FireWire early on boot	170
12.41 Enable dynamic printk() call support	170
12.42 Sample kernel code	170
12.43 Filter access to /dev/mem	171
12.44 Enable verbose x86 bootup info messages	171
12.45 Early printk	171
12.46 Check for stack overflows	171
12.47 Stack utilization instrumentation	171
12.48 Debug page memory allocations	171
12.49 Debug access to per_cpu maps	171
12.50 Export kernel pagetable layout to userspace via debugfs	172

12.51	Write protect kernel read-only data structures	172
12.52	Testcase for the NX non-executable stack feature	172
12.53	Use 4Kb for kernel stacks instead of 8Kb	172
12.54	Memory mapped IO tracing	172
12.55	IO delay type	173
12.56	Debug boot parameters	173
12.57	CPA self-test code	174
12.58	Allow gcc to uninline functions marked ‘inline’	174
13	Security options 安全选项	175
13.1	Enable access key retention support	175
13.2	Enable different security models	175
13.3	File POSIX Capabilities	176
13.4	Low address space to protect from user allocation	176
13.5	NSA SELinux Support	177
13.6	Simplified Mandatory Access Control Kernel Support	179
14	Cryptographic API 密码接口	181
14.1	Sequence Number IV Generator	181
14.2	Cryptographic algorithm manager	181
14.3	HMAC support	181
14.4	XCBC support	181
14.5	Null algorithms	182
14.6	MD4 digest algorithm	182
14.7	MD5 digest algorithm	182
14.8	SHA1 digest algorithm	182
14.9	SHA224 and SHA256 digest algorithm	182
14.10	SHA384 and SHA512 digest algorithms	182
14.11	Whirlpool digest algorithms	183
14.12	Tiger digest algorithms	183
14.13	GF(2^{128}) multiplication functions	183
14.14	ECB support	183
14.15	CBC support	183
14.16	PCBC support	184
14.17	LRW support	184
14.18	XTS support	184
14.19	CTR support	184
14.20	GCM/GMAC support	184
14.21	CCM support	184

14.22 Software async crypto daemon	185
14.23 DES and Triple DES EDE cipher algorithms	185
14.24 FCrypt cipher algorithm	185
14.25 Blowfish cipher algorithm	185
14.26 Twofish cipher algorithm	185
14.27 Twofish cipher algorithms (i586)	185
14.28 Serpent cipher algorithm	186
14.29 AES cipher algorithms	186
14.30 AES cipher algorithms (i586)	186
14.31 CAST5 (CAST-128) cipher algorithm	186
14.32 CAST6 (CAST-256) cipher algorithm	186
14.33 TEA, XTEA and XETA cipher algorithms	186
14.34 ARC4 cipher algorithm	187
14.35 Khazad cipher algorithm	187
14.36 Anubis cipher algorithm	187
14.37 SEED cipher algorithm	187
14.38 Salsa20 stream cipher algorithm	188
14.39 Salsa20 stream cipher algorithm (i586)	188
14.40 Deflate compression algorithm	188
14.41 Michael MIC keyed digest algorithm	188
14.42 CRC32c CRC algorithm	188
14.43 Camellia cipher algorithms	188
14.44 Testing module	189
14.45 Authenc support	189
14.46 LZO compression algorithm	189
14.47 Hardware crypto devices	189
15 Virtualization 虚拟机	191
15.1 Kernel-based Virtual Machine (KVM) support	191
15.2 Linux hypervisor example code	192
15.3 PCI driver for virtio devices	192
15.4 Virtio balloon driver	192
16 Library routines 例行库	193
16.1 CRC-CCITT functions	193
16.2 CRC16 functions	193
16.3 CRC calculation for the T10 Data Integrity Field	193
16.4 CRC ITU-T V.41 functions	193
16.5 CRC32 functions	194
16.6 CRC7 functions	194
16.7 CRC32c Cyclic Redundancy-Check	194

Part I

总览

内核特性和选项说明

一些内核特性可以直接编译进内核。一些可以被编译为可装载的模块。一些则可以完全被一起移除。这里还有一些内核参数，它不是真正的特性，但是必须输入有效的十进制或者十六进制或者一些文本。

菜单项目中，以这些符号开始的表示特性可以：

[]	编译或者移除
< >	编译或者模块化或者移除
{ }	编译或者模块化（被其他特性选择）
- -	被其他特性选择

其中：

文字中的 Y 表示选择进内核。M 表示编成模块。N 表示不选择。

在 make menuconfig 下，*表示 Y，M 表示 M，空白表示 N。

为了改变这些特性，你需要用光标键移动使之高亮显示，同时按 Y 将其编译，按 M 模块化，按 N 将它移除。你可能也需要按空格键来在几个选择中循环（Y→N→M→Y）。

一些额外的键盘提示

菜单

使用上下方向键来高亮你想要改变的项目或者子菜单，然后按回车键。子菜单显示为→

小技巧：按选项中高亮的那个字母（热键）。按热键多次可以在使用了这个热键的选项中循环。

你可以用上下翻页键来翻页，以观察看不到的选项。

要退出菜单，使用方向键来高亮 < EXIT > 按钮，然后按回车。

小技巧：按 < ESC > < ESC > 或者 E 或者 X，如果这时没有选项使用这些热键。也可以只按一次 < ESC >，但是由于延迟，可能会让你很烦。

同时，< TAB > 键和方向键将会在 < Select >，< EXIT > 和 < HELP > 中循环。

要得到选项的提示，使用方向键高亮 < HELP > 然后按回车。

小技巧：按 < H > 或者 < ? >。

列表选择

使用方向键来选择相应选项，按 < S > 或者按空格键。

小技巧：按选项的第一个字母，然后按 < S > 或者空格键。

要查看可用的帮助，使用方向键高亮 < HELP >，然后按回车。

小技巧：按 H 或者 < ? >。

同时，< TAB > 键和方向键将会在 < SELECT > 和 < HELP > 间循环。

数据输入

输入需要的信息，然后按 `< ENTER >`。如果你在输入十六进制数据，没有必要添加前缀 `'0X'`。

需要帮助时，用 `< TAB >` 或者方向键高亮帮助菜单后按 `< ENTER >`。也可以按 `< TAB > < H >`。

文本框（帮助窗口）

使用方向键来调整方向。也可以使用 VI 的方向键：`h`，`j`，`k`，`l` 来完成，还可以使用空格和 `< B >` 来进行，这是 `less` 和 `lynx` 的键位。

按 `< E > < X > < ENTER >` 或者 `< ESC > < ESC >` 退出。

可选配置文件

支持可选配置文件，这样可以方便地在各种内核配置之间切换。

在主菜单的最后，你会发现两个选项。一个是储存当前配置文件，另一个是载入已经保存好的配置文件。

即使你不使用可选的配置文件，你在进行配置的过程中，可能发现你已经完全搞乱了你的设置，这时你可以使用“Load Alternate...”选项来恢复你之前的设置。这样不需要重新启动配置程序。

其他信息

如果你在 XTERM 窗口中使用配置程序 Menuconfig，确保你的环境变量 `$TERM` 指向了一个支持颜色的 X 终端。否则，菜单看起来将会很丑。配置程序在 RXVT 窗口中无法正常显示，因为它只支持一种颜色。

配置程序在显示模式高于 25 行和 80 列的显示器上，将会显示得更大。为使之工作，命令：`stty size` 应该可以显示出屏幕的实际行列数。我强烈建议你不要在 SHELL 变量中设置 `LINES` 和 `COLUMNS`。一些发行版通过 `/etc/profile` 设置了这些环境变量。一些程序可能会出错，因为这些变量没有真实反映屏幕的尺寸。

可选个人设置

如果你喜欢把内核的所有选项列在一个单独的菜单上，而不是默认的多级菜单，那么运行配置程序时设置 `MENUCONFIG_MODE` 为单独的菜单。比如：

```
make MENUCONFIG_MODE=single_menu menuconfig
```

`< ENTER >` 将会把相应内容融合和展开。

注意这种模式与普通模式相比会导致 CPU 的额外开销（特别是有大量的没有融合的内容）。

不同的可用颜色面板

通过 MENUCONFIG_COLOR 可以设置不同的颜色面板。

选择一个面板输入：

```
make MENUCONFIG_COLOR=< theme > menuconfig
```

可用的 theme 有：

mono = > 用于单色显示。

blackbg = > 背景为黑色。

classic = > 背景为蓝色，传统的界面。

bluetitle = > LCD 友好界面（默认）。

内核 2.6.28 的大改动

- 正式加入对 ext4 文件系统的支持。

Part II

开始配置菜单

Chapter 1

General setup 常规设置

1.1 Prompt for development and/or incomplete code/drivers

对开发中的或者未完成的代码和驱动进行提示

Linux 下的很多东西，比如网络设备、文件系统、网络协议等等，它们的功能、稳定性、或者测试等级等等还不能够符合大众化的要求，还处于开发之中。这就是所谓的 α 版本（阿尔法版本）：最初开发版本；接下来的是 β 版本（BETA 版本），公开测试版本。如果这是阿尔法版本，那么开发者为了避免收到诸如“为何这东西不工作”的信件的麻烦，常常不会让它发布出去。但是，积极的测试和使用阿尔法版本对软件的开发是非常好的。你只需要明白它未必工作得很好，在某些情况有可能会出问题。汇报详细的出错情况对开发者很有帮助。

这个选项同样会让一些老的驱动可用。很多老驱动在后来的内核中已经被代替或者被移除。除非你想要帮助软件的测试，或者开发软件，或者你的机器需要这些特性，否则你可以选 N，那样你会在配置菜单中得到较少的选项。如果你选了 Y，你将会得到更多的阿尔法版本的驱动和代码的配置菜单。

1.2 Local version - append to kernel release

本地版本 - 附加内核发行版本

在你的内核版本后面加上一串字符来表示版本。这些字符在你使用 `uname -a` 命令时会显示出来。你在这设置的版本字符将会出现在文件的目录和内容中，如果这些文件调用了内核的版本号。你的字符最多不能超过 64 位。

1.3 Automatically append version information to the version string

自动生成版本信息

这个选项会尝试自动探测目前的编译树列是否是一个发行版本树列¹，这通过寻找 `git`² 标记来完成，它归属于目前的树列修正版顶层。

一个类似格式 `-gxxxxxxx` 的字符串将会添加在本地版本的后面，如果 `git` 树列找到了的话。这个字符串也将在所有匹配的 `localversion*` 文件后面添加，如果 `CONFIG_LOCALVERSION` 值设置了的话。

在这使用的字符串，实际上是运行了以下命令得到的前 8 位字符：

```
$ git rev-parse --verify HEAD
```

这个命令是通过在“`scripts/setlocalversion`”的脚本完成的。

1.4 Support for paging of anonymous memory (swap)

内存随机页面支持 (swap)

这个选项可以让你选择是否在内核中使用 `swap` 设备或者 `swap` 文件，以提供比实际 RAM 内存更多的虚拟内存。

如果不清楚，选 Y。

1.5 System V IPC

System V 中间过程连接

中间过程连接是一组功能和系统调用，使得进程能够同步和交换信息。这通常来说是好事，有一些程序只有你选择了 Y 才能运行。特别地，你想在 Linux 下运行 DOS 仿真程序，你必须选 Y。

你可以用 `info ipc` 命令来了解 IPC。

1.6 POSIX Message Queues

POSIX 信息队列

POSIX 信息队列变量是 IPC 的一部分。在 POSIX 信息队列中，每个信息有一个优先权，这可以决定进程处理它的连续性³。如果你想要编译和运行比如写给 Solaris 的程序，它们使用了 POSIX 信息队列，那么要在这选 Y。

POSIX 信息队列对于文件系统是可见的，称为‘`mqueue`’，也可以被挂载，如果你打算这么做的话。

如果不清楚，选 Y。

¹因为有的人可能会更改和调试内核，导致这个内核不是标准的发行版内核。

²Linus Trovald 制作的一个内核版本控制工具

³意即越连续，则占用时间会越多

1.7 BSD Process Accounting

BSD 进程统计

如果你选 Y，用户级别的程序就可以通过特殊的系统调用方式来通知内核把进程统计信息记录到一个文件：当这个进程存在的时候，信息就会被内核记录进文件。信息通常包括建立时间、所有者、命令名称、内存使用、控制终端等。这对用户级程序非常有用。所以通常选 Y 是一个好主意。

1.7.1 BSD Process Accounting version 3 file format

BSD 进程统计 v3 文件格式

如果你选 Y，进程统计信息将以一种新的文件格式写入，它把每个父进程的 ID 当作日志写入文件。注意这种文件格式和以前的 v0/v1/v2 文件格式不兼容，所以你需要升级工具软件以处理它。

1.8 Export task/process statistics through netlink

通过网联 (netlink)⁴ 输出工作/进程的相关数据

通过通用的网联接口输出工作/进程的相应数据，和 BSD 进程统计不同的是，这些数据在进程运行的时候就可以通过相关命令访问。和 BSD 进程统计类似，数据将在进程结束时送入用户空间。

如果不清楚，选 N。

1.8.1 Enable per-task delay accounting

允许单任务延迟统计

通过任务等待系统资源的方式收集信息，这会收起时间的花销。系统的资源如 cpu、同步 I/O 复合块和交换页面等等。这些统计信息可以帮助设置一个任务的 cpu/io/iss 优先级等。

如果不清楚，选 N。

1.8.2 Enable extended accounting over taskstats

允许通过任务状态进行扩展统计

收集扩展任务统计数据，并把它们发送至用户空间，以便可以用任务状态接口进行处理。

如果不清楚，选 N。

⁴要注意区分网联和普通的网络连接的区别。前者是 Linux 新版本内核在内核空间与用户空间之间进行信息传输的方式，它在实质上与网络的服务器和客户端的连接是一样的，所以这个连接被形象化为 netlink，直译为网络连接，笔者翻译为网联以区别。

- Enable per-task storage I/O accounting
允许单任务储存 I/O 统计
收集任务引发的储存 I/O 上的一定数量的字节信息。
如果不清楚，选 N。

1.9 Auditing support

审计支持

允许审计的基础结构能够被其他内核子系统使用，比如 SELinux，它需要这个来进行登录时的声音和视频输出。没有 CONFIG_AUDITSYSCALL 时（即下一个选项）无法进行系统调用。

1.9.1 Enable system-call auditing support

允许系统调用审计支持

允许低层的系统调用审计构架，这可以独立使用，或者用于其他的内核子系统，比如 SELinux。要使用审计的文件系统监视特性，请确保 INOTIFY 被配置。

1.10 Kernel .config support

内核.config 支持

这个选项允许已经完成的.config 文件⁵保存在内核当中。

它提供正在运行中的或者还在硬盘中的内核的相关配置选项。可以通过内核镜像文件 kernel image file 用脚本 scripts/extract-ikconfig 来提取出来，作为当前内核重编译或者另一个内核编译的参考。如果你的内核在运行中，可以通过 /proc/config.gz 文件来读取。下一个选项提供这项支持。

1.10.1 Enable access to .config through /proc/config.gz

允许通过 /proc/config.gz 访问 .config

可以通过 /proc/config.gz 访问当前内核的 .config。

1.11 Kernel log buffer size

内核日志缓冲尺寸

选择内核日志缓冲尺寸。

例子：

⁵即编译 Linux 时的配置文件

17=128KB

16=64KB

15=32KB

14=16KB

13=8KB

12=4KB

1.12 Control Group support

组控制支持

这个选项会让你像使用多 CPU 一样使用处理器组子系统。

如果不清楚，选 N。

1.12.1 Example debug cgroup subsystem

调试组控制子系统范例

这个选项开启一个简单的组控制子系统，输出有用的关于组控制框架的调试信息。

如果不清楚，选 N。

1.12.2 Namespace cgroup subsystem

命名空间组控制子系统

提供一个简单的命名空间组控制子系统，以提供分层次的设置命名，用于虚拟服务器实例和检查点/重启任务。

1.12.3 control group freezer subsystem

组控制冻结子系统

提供一种方式用于冻结和解冻控制组中的所有任务。

1.12.4 Device controller for cgroups

组控制系统设备控制器

提供一个组控制工具设备白名单⁶，以便组中的进程可以建立节点或者打开它。

⁶与黑名单相对，指优良者名单

1.12.5 Cpuset support

Cpuset 支持

这个选项将使你可以建立和管理 CPUSET，它允许动态地把一个系统分区至多个 CPU 组和内存节点组中，并分配和运行仅位于这些组之中的任务。这对于大型的 SMP 和 NUMA 系统非常有用。

如果不清楚，选 N。

1.13 Group CPU scheduler

CPU 组调度器

这个特性让 CPU 调度器识别任务组，并控制 CPU 带宽定位至那些任务组。

1.13.1 Group scheduling for SCHED_OTHER

SCHED_OTHER 组调度器

1.13.2 Group scheduling for SCHED_RR/FIFO

SCHED_RR/FIFO 组调度器

1.13.3 Basis for grouping tasks

基本组任务

- user id 用户 id

这个选项选择用户 id 作为基本组任务，因此对每个用户提供相同的 CPU 带宽。

- Control groups 控制组

这个选项允许你建立任意的任务组，通过使用“cgroup”伪文件系统来实现，这可以控制 cpu 带宽定位至每个任务组。

1.14 Simple CPU accounting cgroup subsystem

简单 CPU 统计组控制子系统

提供一个简单的资源控制器，通过在组控制中的任务监视所有 CPU 的花销。

1.15 Resource counters

资源计数器

这个选项开启独立的资源计数器构架，它与组控制系统共同工作。

1.16 Memory Resource Controller for Control Groups

控制组的内存资源控制器

提供一个内存资源控制器，以管理随机内存和页面缓存。

1.17 Create deprecated sysfs file

建立不需要的系统文件

这个选项建立一些不需要的链接，比如“device”链接，`< subsystem > : < name >` 链接和 `< bus >` 链接。它将给 Uevent 环境增加不需要的关键字。

在今天，上述的特性没有一个被使用了。因为它们把驱动核心细节输送到了用户空间，或者把不稳定的内核属性给输出了。

如果可能的话，这个选项将把一些设备的结构层次进行移动，回到 `/sys/class` 体系中，以适应老的 `udev` 版本和用户空间程序。

如果你正在使用新的用户空间包，那么在这选 `N` 是安全的。

1.18 Kernel→user space relay support (formerly relayfs)

内核空间和用户空间传递通讯支持

这个选项在特定的文件系统（比如 `debugfs`）中提供数据传递接口支持，它可以提供从内核空间到用户空间的大批量的数据传递工具和设施。

如果不清楚，选 `N`。

1.19 Namespaces support

命名空间支持

给任务提供一种途径，以便同样 `id` 的任务能够指向不同的对象。例如相同的 `IPC id` 将会指向不同的对象或者相同的用户 `id` 或 `pid` 将指向不同的任务，当在不同的命名空间使用时。

1.19.1 UTS Namespaces

UTS 命名空间

在这个命名空间，任务通过 `uname()` 系统调用来查看不同的信息。

1.19.2 IPC Namespaces

IPC 命名空间

在这个命名空间，任务用 IPC id 来工作。IPC id 通过不同命名空间的不同 IPC 对象来进行通信。

1.19.3 User namespace

用户命名空间

这里允许容器，比如 `vservers`，使用用户命名空间来为不同的服务器提供不同的用户信息。如果不清楚，选 N。

1.19.4 PID Namespaces

PID 命名空间

支持进程 id 命名空间。这允许多进程拥有相同的 pid，只要它们位于不同的 pid 命名空间。这是一个容器的编译块。

除非你知道如何用这项实验性质的特性来工作，否则选 N。

1.20 Initial RAM filesystem and RAM disk support

初始 RAM 文件系统和 RAM 磁盘支持

初始 RAM 文件系统是一种 ram 文件系统，它被启动器装载（比如 `lilo`），在通常的启动程序前以 `root` 形式挂载。通常它用来装载模块，这在挂载真正的 `root` 文件系统时使用。

如果 RAM 磁盘支持也被选入，这将允许初始 RAM 磁盘支持（`initrd`），这将给内核尺寸增大 15KB（在一些体系上，可能会更大）。

如果不清楚，选 Y。

1.20.1 Initramfs source files

初始 ram 文件系统源文件

这里可以是一个单独的 `cpio` 压缩包，有 `.cpio` 的后缀。或者是一个目录和文件的列表，用来编译初始 ram 文件系统镜像。`cpio` 压缩包应该包含 ram 文件系

统镜像所需的文件系统包。目录列表应该包含所有文件的层次，文件列表应该包含“usr/gen_init_cpio”中提到的内容。

当多重的目录和文件确定时，初始 ram 文件系统镜像将把它们聚合起来。

如果不清楚，将它留空。

1.21 Optimize for size

尺寸优化

这个选项将在 GCC 命令后用“-Os”代替“-O2”参数，这样可以得到更小的内核。

如果不清楚，选 N。

1.22 Configure standard kernel features (for small systems)

配置标准内核特性（对小系统）

这个选项可以让内核的基本选项和设置无效或者扭曲。这是用于特定环境中的，它允许“非标准”内核。你要是选它，你一定要明白自己在干什么。

1.22.1 Load all symbols for debugging/ksymoops

加载所有的标记用于调试/ksymoops

在这选 Y 可以让内核打印所有的崩溃信息标记和栈回溯信息。这里会增大内核的一些尺寸，因为所有的标记都被编译进了内核镜像。

- Include all symbols in kallsyms

在 kallsyms 中包含所有标记

- Do an extra kallsyms pass

进行一个额外的 kallsyms 过程

1.23 Disable heap randomization

禁止随机堆

随机堆会让堆的开发更加困难，它同样会破坏传统的二进制文件（基于 libc5）。这个选项使系统启动时变为禁止随机堆，在运行中，可以通过 /proc/sys/kernel/randomize_va_space 的值改为 2 来修改。

对于非传统的发行版（2000年后的），选 N 是安全的。

1.24 Choose SLAB allocator

选择 SLAB 分配符

这个选项允许你选择一个 slab 分配符。

1.24.1 SLAB

SLAB

这是传统的，它已经被证实在所有的环境中都工作得很好。它用每个 CPU 和每个节点队列来组织每个热对象。这是默认的分配符。

1.24.2 SLUB

非队列分配符

这个分配符最小化了缓冲，它不像 SLAB 那样使用队列缓冲对象。每个 CPU 的缓冲被认为是使用 slab 对象，而不是一个队列。SLUB 可以高效使用内存，还有加强的诊断。

1.25 Profiling support

剖面法支持

在这选 Y 以支持扩展剖面支持架构，它被 OProfile 等剖面器使用。

1.26 Activate markers

动态标记

在每个标记点放置一个空函数调用。能够被动态修改，用于探测功能。

1.27 OProfile system profiling

OProfile 系统剖面器

OProfile 是一个剖面分析系统，能够对整个系统进行剖面图分析，包括内核、内核模块、库文件和应用程序。

如果不清楚，选 N。

1.28 Kprobes

K 探测器

Kprobes 允许你追踪几乎所有内核的地址和运行一个反馈函数。register_kprobe() 函数定制了一个追踪点并指定了反馈方式。Kprobes 对于内核调试、非侵入式设备测试是相当有用的。

如果不清楚，选 N。

Chapter 2

Loadable module support 可加载模块支持

2.1 Enable loadable module support

允许可装载模块支持

这个选项可以让你的内核支持模块。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)。在启动时不用到的功能，编成模块是最有效的方式¹。你可以查看 MAN 手册来了解：`modprobe`, `lsmod`, `modinfo`, `insmod` 和 `rmmod`。

如果你选了这项，你可能需要运行“`make modules_install`”命令来把模块添加到 `/lib/modules/` 目录下，以便 `modprobe` 可以找到它们。

如果不清楚，选 Y。

2.2 Forced module loading

强制模块加载

允许在没有版本信息的时候加载模块（比如 `modprobe -force`）。这项功能设置了‘F’尾标记，这通常是一个坏主意。

2.3 Module unloading

模块卸载

¹一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，否则无法启动系统。

这个选项可以让你卸载不再使用的模块，如果选 N 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)。如果选 N，可以让你的内核尺寸减小一些，内核更简单一些。

如果不清楚，选 Y。

2.3.1 Forced module unloading

强制模块卸载

这个选项允许你强行卸除模块，即使内核认为这不安全。内核将会立即移除模块，而不管是否有人在使用它（用 `rmmod -f` 命令）。这通常不是什么好主意。

如果不清楚，选 N。

2.4 Module versioning support

模块版本支持

有时候，你需要编译模块。选这项会添加一些版本信息，来给编译的模块提供独立的特性，以使不同的内核在使用同一模块时区别于它原有的模块。这有时可能会有点用。

如果不清楚，选 N。

2.5 Source checksum for all modules

对所有模块进行源代码检查

模块中有一个变量 `MODULE_VERSION`，它还有一个额外的“srcversion”域，嵌入了它们的模块信息中，其中包含着源代码的 `sum`²。这能够帮助所有者明确地知道他使用了那个源文件来编译这个模块（因为有的人在编译模块时不小心更改了内核模块的源代码但忘记更改版本号）。这个选项将会在所有模块建立“srcversion”域。

如果不清楚，选 N。

²—一种检测码，例如：`md5sum`

Chapter 3

Enable the block layer 允许块设备

为内核提供块设备支持。

如果你不选此项，那么将移除相应的功能。这对于嵌入式设备来说是有用的。

如果不选此项，那么 block device 文件将不可用，一些文件系统比如 ext3 将不可用。同时，这也会禁止 SCSI 字符设备和 USB 储存设备，因为它们使用各种块设备终端和工具。

选 Y，除非你知道你确实不需要挂载硬盘和其他类似的设备。

3.1 Support for Large Block Devices

对大型块设备的支持

提供对于 2TB 或者更大的块设备的支持。

这个选项对于大型的块设备是必须的。这些设备通常都大于 2TB，包括 RAID、磁盘、网络块设备、逻辑卷管理器 LVM 和回送。

或者举例来说，RAID 设备通常比最大的个人硬盘还要大很多。

这个选项通常是没有用的，如果你有一个人的磁盘总共有 2TB +，同时你不打算把它当作大型块设备（比如 RAID 或者 LVM）。

如果不清楚，选 N。

3.2 Support for tracing block io actions

对块设备进行跟踪

选 Y，如果你想要跟踪你的块设备情况，这通常用一个给定的队列来显示。

这种跟踪允许你观察在块设备队列上任何堵塞的发生。

如果不清楚，选 N。

3.3 Support for Large Single Files

大文件支持。

如果你准备处理的文件大于 2TB，选 Y。
如果不清楚，选 N。

3.4 Block layer SG support v4

块设备层 SG¹ v4 支持

在这选 Y 将为所有块设备开启通用 SG v4 支持。

不同于 SG v3，SG v4 能够处理复杂的 SCSI 命令：标记变量长度 cdb's，使用双向数据传输和通用请求/回复协议（比如任务管理器功能和串行 SCSI² SMP 功能）。

如果不清楚，选 N。

3.5 Block layer data integrity support

块设备层数据完整性支持

一些储存设备允许额外信息的储存和找回，以便保护数据。块设备层数据完整性选项提供了相应的挂钩，这可以用于文件系统中，以确保更好的数据完整性。

选 Y，如果你有一个储存设备，它提供 T10/SCSI 数据完整域或者 T13/ATA 扩展路径保护功能。

如果不清楚，选 N。

3.6 I/O Schedulers

I/O 调度器

Anticipatory I/O scheduler 抢先式 I/O 调度方式对于大多数环境通常是比较好的选择。但是它和 Deadline I/O 调度器相比有点大和复杂，它有时在数据调入时会比较慢。

Deadline I/O scheduler Deadline（期限式）I/O 调度器简单而又紧密，在性能上和抢先式调度器不相上下，在一些数据调入时工作得更好。至于在单进程 I/O 磁盘调度上，它的工作方式几乎和抢先式调度器相同，因此也是一个好的选择。

CFQ I/O scheduler CFQ 调度器尝试为所有进程提供相同的带宽。它将提供平等的工作环境，对于桌面系统很合适。这也是默认的 I/O 调度器。

Default I/O scheduler 选择默认的 I/O 调度器 建议选 CFQ I/O scheduler

¹通用 SCSI

²串行 SCSI 就是 SAS

Chapter 4

Processor type and features 处理器类型及特性

4.1 Tickless System

微标记系统（动态标记）

这个选项允许一个微标记系统：时钟中断只会在需要的前提下才触发—要么是系统繁忙的时候，要么是系统空闲的时候。

4.2 High Resolution Timer Support

高分辨率时钟支持

这个选项开启高分辨率时钟支持。如果你的硬件不够好，那么这个选项仅仅是增加内核的尺寸¹。

4.3 Symmetric multi-processing support (SMP)

对称多处理器支持

这将支持有多 CPU 的系统。如果你的系统只有一个 CPU，选 N。反之，选 Y。

如果你选 N，内核将会在单个或者多个 CPU 的机器上运行，但是只会使用一个 CPU。如果你选 Y，内核可以在很多（但不是所有）单 CPU 的机器上运行，在这样的机器，你选 N 会使内核运行得更快。

注意如果你选 Y，然后在 Processor family 选项中选择“586”或“Pentium”，内核将不能运行在 486 构架的机器上。同样的，多 CPU 的运行于 PPro 构架上的内核也无法在 Pentium 系列的板上运行。

¹也就是无用

使用多 CPU 机器的人在这里选 Y，通常也会在后面的选项“Enhanced Real Time Clock Support”中选 Y。如果你在这选 Y，“Advanced Power Management”的代码将不可用。

如果不清楚，选 N。

4.4 Enable MPS table

允许 MPS 表

老的 smp 系统没有合适的 acpi 支持。更新的系统（尤其是 64 位 CPU 的）有了 acpi 支持，MADT 和 DSDT 将代替它。

4.5 Subarchitecture Type

子构架类型

- PC-compatible

PC-兼容机

选这个，如果你的机器是标准 PC 或者兼容机。

- AMD Elan

如果你的机器是 AMD Elan，选这项。

注意，如果你是 K6/Athlon/Opteron 处理器不要选这项，而是选择 PC-compatible。

- Generic architecture

通用体系结构

这个选项编译了 NUMAQ, Summit, bigsmp, ES7000, 及其他默认子构架。它作为一个通用的二进制内核。如果你把它们都选上了，内核将按顺序一个个检测，并将回到默认值。

4.6 RDC R-321x SoC

这个选项用于 RDC R-321x 系统芯片，也被叫做 R-8610-(G)。

如果你没有这种芯片，那么选 N。

4.7 Single-depth WCHAN output

单精度 WCHAN² 输出

计算简单的 `/proc/<PID>/wchan` 的值。如果这个选项被禁止，那么 `wchan` 值将会返回并影响调用的程序。这里提供更精确的 `wchan` 值，不过会造成一点点的调度开销。

如果不清楚，选 Y。

4.8 Paravirtualized guest support

半虚拟化客户支持

在这选 Y，可以看到更多的有关在各种管理程序下运行 Linux 的选项。这个选项本身不增加任何内核代码。

如果你选 N，所有的子选项将会禁止。

4.9 Memtest

内存测试

这个选项增加一个内核参数 `'memtest'`，这允许设置内存测试。

<code>memtest=0,</code>	表示禁止；默认值
<code>memtest=1,</code>	表示做 1 次测试模式；
<code>...</code>	
<code>memtest=4,</code>	表示做 4 次测试模式。

如果不清楚，选 N。

4.10 Processor family

处理器类型

这里是处理器的类型。这里的信息主要目的是用来优化。为了让内核能够在所有 X86 构架的 CPU 上运行（虽然不是最佳速度），在这你可以选 386。

内核不会运行在比你选的构架还要老的机器上。比如，你选了 Pentium 构架来优化内核，它将不能在 486 构架上运行。

如果你不清楚，选 386。

- 386
- 486

²WCHAN:Waiting Channel，表示进程或系统调用等待时的地址。

- 586/K5/5x86/6x86/6x86MX
- Pentium-Classic
- Pentium-MMX
- Pentium-Pro
- Pentium-II/Celeron(pre-Coppermine)
- Pentium-III/Celeron(Coppermine)/Pentium-III Xeon
- Pentium M
- Pentium-4/Celeron(P4-based)/Pentium-4 M/older Xeon
- K6/K6-II/K6-III
- Athlon/Duron/K7
- Opteron/Athlon64/Hammer/K8
- Crusoe
- Efficeon
- Winchip-C6
- Winchip-2
- Winchip-2A/Winchip-3
- GeodeGX1
- Geode GX/LX
- CyrixIII/VIA-C3
- VIA C3-2(Nehemiah)
- VIA C7
- Core 2/newer Xeon

4.11 Generic x86 support

通用 X86 支持

除了对上面你选择的 X86 CPU 类型进行优化（比如 PII，Crusoe 或者 Athlon），它还对更多类型 X86 CPU 的进行优化。这将会使内核在其他的 X86 CPU 上运行得更好。

对于供应商来说，他们非常需要这些功能，因为他们需要更通用的优化支持。

4.12 Branch Trace Store

分支追踪储存

这里增加一个 ptrace 接口到硬件的分支追踪储存。

测试人员将可以使用它来收集所调试程序的运行追踪信息，以应对这个问题：“我如何才能到达这里？”测试人员将追踪用户模式和内核模式。

选 Y，除非你的机器上不进行任何的程序开发，你想要减小代码的尺寸。

4.13 HPET Timer Support

HPET 时钟支持

允许内核使用 HPET。HPET 是代替当前 8254 的下一代时钟，全称叫作高精度事件定时器。你可以安全地选 Y。但是，HEPT 只会在支持它的平台和 BIOS 上运行。如果不支持，8254 将会激活。

选 N，将继续使用 8254 时钟。

4.14 Maximum number of CPUs

最大支持的 CPU 数量

这里允许你指定内核支持的最大 CPU 数量。最大值为 255，最小为 2。

这里纯粹是为了节约内存 - 每多支持一个 CPU，内核镜像将增大近似 8KB。

4.15 SMT (Hyperthreading) scheduler support

超线程调度器支持

超线程调度器在某些情况下将会对 Intel Pentium 4 HT 系列有较好的支持。但是对于别的构架，可能会减慢速度。

如果你不清楚，选 N。

4.16 Multi-core scheduler support

多核调度器支持

多核调度将会对多核的 CPU 系列有较好的支持。但是在别的环境下可能会增加开销。

如果不清楚，选 N。

4.17 Preemptible Model

抢先式模块

4.17.1 No Forced Preemption (Server)

非强迫式抢先（服务器）

这是传统的 Linux 抢先式模型，针对于高吞吐量设计。它同样在很多时候会提供很好的响应，但是也可能会有较长的延迟。

如果你是建立服务器或者用于科学运算，选这项，或者你想要最大化内核的原始运算能力，而不理会调度上的延迟。

4.17.2 Voluntary Kernel Preemption (Desktop)

自动式内核抢先（桌面系统）

这个选项通过向内核添加更多的“清晰抢先点”来减少内核延迟。这些新的抢先点以降低吞吐量的代价，来降低内核的最大延迟，提供更快应用程序响应。这通过允许低优先级的进程自动抢先来响应事件，即使进程在内核中进行系统调用。这使得应用程序运行得更“流畅”，即使系统已经是高负荷运转。

如果你是桌面系统编译内核，选这项。

4.17.3 Preemptible Kernel (Low-Latency Desktop)

可抢先式内核（低延迟桌面）

这个选项通过使所有内核代码（非致命部分）编译为“可抢先”来降低内核延迟。

这通过允许低优先级进程进行强制抢先来响应事件，即使这些进程正在进行系统调用或者未达到正常的“抢先点”。这使得应用程序运行得更加“流畅”即使系统已经是高负荷运转。代价是吞吐量降低，内核运行开销增大。

选这项如果你是桌面或者嵌入式系统编译内核，需要非常低的延迟。

4.18 Local APIC support on uniprocessors

对特定处理器的本地 APIC 支持

本地的 APIC³是一个综合的 CPU 中断控制器。如果你有一个单 CPU 系统，它有一个本地的 APIC，那么你可以在这选 Y。如果你选了 Y，但是你的机器根本没有本地 APIC 支持，那么内核运行起来也不会变慢。本地 APIC 支持 CPU 生成自我中断（时间、计数器），也支持 NMI 看门狗探测硬件锁定。

³高级可编程中断控制器

4.19 Machine Check Exception

机器例外检查

机器例外检查允许处理器在检测到问题（比如过热、组件错误）时通知内核。内核根据问题的严重程度来决定下一步行为，比如在命令行上打印告警信息，或者关机。你的处理器必须是Pentium 或者更新版本才能支持这个功能。用 `cat /proc/cpuinfo` 来检测你的 CPU 是否有 `mce` 标志。

注意一些老的Pentium 系统存在设计缺陷，会提供假的 MCE 事件，所以在所有 P5 处理器上 MCE 被禁用，除非在启动选项上明确“`mce`”参数。同样地，如果 MCE 被编译入内核并在非标准的机器上导致错误，你可以用“`nomce`”启动参数来禁用 MCE 。

MCE 功能会自动忽视非 MCE 处理器，比如 386 和 486 ，所以几乎所有人都可以在这里选 Y 。

4.19.1 Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium 4

检测 AMD Athlon/Duron / Intel Pentium 4 的非致命错误

允许这项特性，系统将会启动一个计时器，每 5 秒进行检测。非致命问题会自动修正（但仍然会记录下来），如果你不想看到这些信息，选 N 。这些信息可以让你发现要损坏的硬件，或者是非标准规格硬件（比如：超频的）。

这个功能只会在特定的 CPU 上起作用。

4.19.2 check for P4 thermal throttling interrupt.

检测 P4 节能器中断

当 P4 进入节能状态时，打印信息。

4.20 Toshiba Laptop support

东芝笔记本支持

这个选项增加一个驱动，它可以安全地访问带有智能 BIOS 的 Toshiba 笔记本系统管理模式。它不会工作在 Phoenix BIOS 模块上。系统管理模式用于给 Toshiba 笔记本设置 BIOS 和节能选项。

选 Y 如果你打算在 Toshiba 笔记本上运行 Linux 。否则，选 N 。

4.21 Dell laptop support

DELL 笔记本支持

这个选项增加一个驱动，它可以安全地访问 Dell Inspiron 8000 系列 CPU 系统管理模式。这个系统管理模式用于读取 cpu 温度和冷却风扇状态，并控制风扇转速。

这个驱动仅仅在 Inspiron 8000 上进行测试，它也可能工作在其他的 Dell 笔记本上。你可以在模块加载时设置参数‘force=1’以强制加载模块，不过这是很冒险的。

如果你打算在 Dell Inspiron 8000 上使用 Linux，选 Y。否则，选 N。

4.22 Enable X86 board specific fixups for reboot

X86 主板重启修复功能

这将打开芯片或者主板上的重启修复功能，从而能够使之正常工作。这功能仅仅在一些硬件和 BIOS 的特定组合上需要。需要这项功能的征兆是重启时使系统卡死或者挂起。

目前，这个修复功能仅仅支持 Geode 机器上使用的 CS5530A/CS5536 芯片以及 RDC R-321x SoC。

选 Y 如果你需要这项功能，目前，选 Y 是安全的，即使你不需要它。否则，选 N。

4.23 /dev/cpu/microcode - microcode support

/dev/cpu/microcode - 支持微代码

这个选项将让你可以更新 Intel 和 AMD 特定处理器的微代码，Intel 支持的是 IA32 家族，比如 Pentium Pro, Pentium II, Pentium III, Pentium 4, Xeon 等等。AMD 支持的是 0x10 和 0x11 家族的处理器，比如 Opteron, Phenom 和 Turion 64 Ultra。显然你需要提供最新的微处理器二进制代码，Linux 内核本身不提供这些代码。

这个选项选择的仅仅是通用的模块，你需要选择至少一个供应商的特殊模块才能工作。

选 M 编译为模块，名为 microcode。

4.23.1 Intel microcode patch loading support

Intel 微代码补丁装载支持

4.23.2 AMD microcode patch loading support

AMD 微代码补丁装载支持

4.24 /dev/cpu/*/msr - Model-specific register support

/dev/cpu/*/msr - 特殊模块寄存器支持

这个设备通过 x86 特殊模块寄存器访问特定的处理器。这是一个字符设备，主字节为 202，次字节为 0 - 31，从 /dev/cpu/0/msr 至 /dev/cpu/31/msr。MSR 访问在多处理器系统上对于某个特定的 CPU 是直接的。

4.25 /dev/cpu/*/cpuid - CPU information support

/dev/cpu/*/cpuid - CPU 信息支持

这个设备提供处理器访问，使 x86 CPUID 指令运行于一个特定的处理器上。它是一个字符设备，主字节为 203，次字节从 0 至 31，从 /dev/cpu/0/cpuid 至 /dev/cpu/31/cpuid。

4.26 High Memory Support

大容量内存支持

Linux 能够在 X86 系统中使用 64GB 的物理内存。但是，32 位地址的 X86 处理器只能支持到 4GB 大小的内存。这意味着，如果你有大于 4GB 的物理内存，并非都能被内核“永久映射”。这些非永久映射内存就称为“高阶内存”。

如果你编译的内核永远都不会运行在高于 1G 内存的机器上，选 OFF（默认选项，适合大多数人）。这将会产生一个“3GB/1GB”的内存空间划分，3GB 虚拟内存被内核映射以便每个处理器能够“看到”3GB 的虚拟内存空间，这样仍然能够保持 4GB 的虚拟内存空间被内核使用，更多的物理内存能够被永久映射。

如果你有 1GB - 4GB 之间的物理内存，选 4GB 选项。如果超过 4GB，那么选择 64GB。这将打开 Intel 的物理地址延伸模式（PAE）。PAE 将在 IA32 处理器上执行 3 个层次的内存页面。PAE 是被 Linux 完全支持的，现在的 Intel 处理器（Pentium Pro 和更高级的）都能运行 PAE 模式。注意：如果你选 64GB，那么在不支持 PAE 的 CPU 上内核将无法启动。

你机器上的内存能够被自动探测到，或者你可以用类似于“mem=256M”的参数强制给内核指定内存大小。

- off 如果不清楚，选 OFF。
- 4GB 选这项如果你用的是 32 位的处理器，内存存在 1-4GB 之间。
- 64GB 选这项如果你用的是 32 位的处理器，内存大于 4GB。

4.27 PAE Support

PAE⁴支持

PAE 需要 NX 支持，此外能够支持更大的交换空间，用于非过量使用的目的。它消耗更多的锁定页面表，同样会在每个处理器消耗更多的页面表空间。

4.28 Add LRU list to track non-evictable pages

增加 LRU⁵列表以追踪非回收表页面

对于活动的和非活动的页面输出列表，保持非回收表页面的关闭状态，这可以使 kswapd⁶不浪费 CPU 时间，或者使它在它扫描这些页面的时候保持平衡丢弃算法。选择这项将使用一个页面标记，并将增加一些内核代码。

选 Y，除非你知道你在做什么。

4.29 Check for low memory corruption

检查低端内存损坏

周期性地检查低端内存损坏，这种损坏可能是由 BIOS 导致。即使在这里选了 Y，它仍然不会在内核运行时执行。要开启它，你需要在内核启动命令行设置“memory_corruption_check=1”。默认地，它每隔 60 秒扫描内存的低 64K 字节。

当以默认选项开启时，这个选项几乎不造成额外开销，因为它保留了一个相关的小内存块，并且不是经常去扫描内存。它探测损坏，并使得损坏的地方不影响到运行中的系统。

但是，它仅作为一个探测工具。如果源于 BIOS 的问题不断地损伤相同地址的内存，你可以使用参数 memmap=来阻止内核使用那些内存。

4.30 Reserve low 64K of RAM on AMI/Phoenix BIOSen

在 AMI/Phoenix BIOSen 上保留 RAM 的低端 64K 字节

在 BIOS 上保留第一个 64K 物理 RAM，它已知是潜在性损坏的内存范围。很多 BIOS 在挂起/恢复的时候使用这个区域的内存，因此它必须不能被内核使用。

在这选 N，如果你非常确信你的 BIOS 已经正确使用了所有的内存区域。

如果你对你的 BIOS 有怀疑（比如，挂起/恢复操作总是不成功，或者在某些硬件热插入时，会导致内核崩溃），但是它又不是 AMI 或 Phoenix 的类型，那么

⁴物理地址扩展

⁵LRU 是 Least Recently Used 的缩写，即最近最少使用页面置换算法，为虚拟页面存储管理服务。

⁶swap 后台进程

你可能需要开启选项：X86_CHECK_BIOS_CORRUPTION=y 来允许内核进行传统的损伤类型检查。

如果不清楚，选 Y。

4.31 Math emulation

数学仿真

Linux 可以仿真一个数学协处理器（用来进行浮点运算），如果你没有的话。486DX 和 Pentium 处理器内建有数学协处理器。486SX 和 386 的没有，除非你专门加过 487DX 或者 387 协处理器。所有人都需要协处理器或者这个仿真。

如果你没有数学协处理器，你需要在这选 Y。如果你有了协处理器还在这选 Y，你的协处理器仍然被用到。这意味着如果你打算把编译的内核用在不同的机器上，选 Y 是明智的选择。

如果不清楚，选 Y，这将使内核增加 66KB，无伤大雅。

4.32 MTRR (Memory Type Range Register) support

内存类型区域寄存器支持

在 Intel P6 系列处理器(Pentium Pro, Pentium II 和更新的)上，MTRR 将会用来规定和控制处理器访问某段内存区域的策略。

如果你在 PCI 或者 AGP 总线上有 VGA 卡，这将非常有用。例如可将 MTRR 设为在显存的地址范围上使用“write-combining”策略，这样 CPU 可以在 PCI/AGP 总线爆裂之前将多次数据传输集成一个大的数据传输，这样可以提升图像的传送速度 2.5 倍以上。选 Y，会生成文件/proc/mtrr，它可以用来操纵你的处理器的 MTRR。典型地，X server 会用到。

这段代码有着通用的接口，其他 CPU 的寄存器同样能够使用该功能。Cyrix 6x86, 6x86MX 和 M II 处理器有 ARR，它和 MTRR 有着类似的功能。AMD K6-2/K6-3 有两个 MTRR，Centaur C6 有 8 个 MCR 允许复合写入。所有这些处理器都支持这段代码，你可以选 Y 如果你有以上处理器。

选 Y 同样可以修正 SMP BIOS 的问题，它仅为第一个 CPU 提供 MTRR，而不为其他的提供。这会导致各种各样的问题，所以选 Y 是明智的。

你可以安全地选 Y，即使你的机器没有 MTRR。这会给内核增加 9KB。

4.32.1 MTRR cleanup support

MTRR 清扫支持

把连续的 MTRR 层转换为离散的，这可以使 X 驱动增加写反馈项目。

可以在内核命令行通过 disable_mtrr_cleanup 禁止它。最大的 mtrr 连续项目尺寸可以通过 mtrr_chunk_size 来设置。

如果不清楚，选 Y。

4.32.2 x86 PAT support

x86 PAT 支持

使用 PAT 的属性来设置页面层次缓冲控制器。

PAT 是现代的等效于 MTRR 的东西，它比 MTRR 更有灵活性。

选 N，如果你在启动内核时发生了问题（启动崩溃、启动挂起、自动重启）或者显卡不能工作。

如果不清楚，选 Y。

4.33 EFI runtime service support

EFI 运行服务支持

这里允许内核在 EFI 平台上使用储存于 EFI 固件中的运行服务。

这个选项只有在有 EFI 固件的系统上 useful。另外，你必须使用最新的 ELILO 登录器才能使内核采用 EFI 的固件设置来启动。即使你没有 EFI，却选了这个选项，内核同样可以启动。

4.34 Enable kernel irq balancing

允许内核中断平衡

默认选 Y，将允许内核进行中断载入平衡控制。选 N 将使内核不进行中断平衡。

4.35 Enable seccomp to safely compute untrusted bytecode (SECCOMP)

允许 SECCOMP（快速计算）安全地运算非信任代码

这个内核特性在程序出现数码错误，需要重新对非信任的代码进行运算时非常有效。它使用管道或者其他传输方式，使文件描述进程支持读/写的系统调用，这样可以利用 SECCOMP 隔离那些程序本身的空间。

一旦 seccomp 通过 `/proc/<pid>/seccomp` 运行，它将不能停止，任务也只能进行一些安全的被 seccomp 认证的系统调用。

如果不清楚，选 Y。只有嵌入式系统选 N。

4.36 Timer frequency

时钟频率

允许设置时钟频率。

这是用户定义的时钟中断频率 100HZ-1000HZ，不过 100 HZ 对服务器和 NUMA 系统更合适，它们不需要很快速的响应用户的要求，因为时钟中断会导致总线争用和缓冲打回。注意在 SMP 环境中，时钟中断由变量 `NR_CPUS * Hz` 定义在每个 CPU 产生。

- 100 HZ

100 HZ 是传统的对服务器、SMP 和 NUMA 的系统选项。这些系统有比较多的处理器，可以在中断较集中的时候分担中断。

- 250 HZ

250 HZ 对服务器是一个好的折衷的选项，它同样在 SMP 和 NUMA 系统上体现出良好的反应速度。如果你想要在 NTSC 视频或者多媒体上使用，选择 300HZ。

- 300HZ

300 HZ 对服务器是一个好的折衷的选项，它同样在 SMP 和 NUMA 系统上体现出良好的反应速度。它在 PAL 和 NTSC 构架的视频和多媒体工作上表现非常优秀。

- 1000 HZ

1000 HZ 对于桌面和其他需要快速事件反应的系统是非常棒的。

4.37 kexec system call

kexec 系统调用

kexec 是一个用来关闭你当前内核，然后开启另一个内核的系统调用。它和重启很像，但是它不访问系统固件。由于和重启很像，你可以启动任何内核，不仅仅是 Linux。

kexec 这个名字是从 exec 系统调用来的。它只是一个进程，可以确定硬件是否正确关闭，所以如果这段代码没能正确为你进行初始化工作，请不要奇怪。它对设备的热拔插会有点帮助。由于它对硬件接口会乱写点东西，所以我没什么好的建议给你。

4.38 Alignment value to which kernel should be aligned

内核对准值

这个值在物理地址上做了一个对准限制，内核将从这里开始启动。内核将会以这个对准值为基点进行编译。

如果启动器⁷在非对准的地址加载内核，同时变量 `CONFIG_RELOCATABLE`⁸ 被设置，内核将会把自己移动到最接近启动器要求的地方进行启动。

如果启动器在非对准的地址加载内核，而 `CONFIG_RELOCATABLE` 没有被设置，内核将无视运行时的加载地址，而是根据自己在编译时已经做好的对准值来进行加载。这个编译时做好的值已经和标准的对准值很接近了。因此结果是内核在标准的对准值物理地址上进行启动。

不要改变这个值，除非你知道你在做什么。

4.39 Compat VDSO support (COMPAT_VDSO)

Compat VDSO 支持

如果你运行的是最新的 glibc (GNU C 函数库) 版本 (2.3.3 或更新)，选 N，这样可以移除高阶的 VDSO 映射，使用随机的 VDSO。

如果不清楚，选 Y。

4.40 Built-in kernel command line

内嵌内核命令行

允许你在编译内核的时候指定特定的内核启动参数。在一些系统上（比如嵌入式系统），它是必需的，或者对于提供一些或者全部的内核自启动参数（即不依赖启动器传递给它的参数）是非常方便的。

要编译命令行参数进内核，在这选 Y，然后在 `CONFIG_CMDLINE` 中填写相应的启动参数。

如果你的系统有功能完备的启动器（比如，非嵌入式的），那么应该在这选 N。

⁷Grub 或者 Lilo 等等

⁸意即“可重定位的”

Chapter 5

Power management options 电源管理选项

5.1 Power Management support

电源管理支持

电源管理意味着你电脑上的某一部分在不用的时候可以关闭或者休眠。这领域有两个竞争标准：APM 和 ACPI。如果你需要两者之一，请把这里选上，再把下面的相关内容选上。

电源管理对于使用电池的笔记本相当重要。如果你有笔记本，请参照相关网站上的说明。

注意，即使你在这选 N，在 X86 构架的机器上，如果没有任务，Linux 会发出 hlt 指令，因此会让处理器休眠，达到节电的目的。

5.1.1 Power Management Debug Support

电源管理调试支持

这个选项提供详细的电源管理调试信息。当你调试和报告电源管理漏洞的时候非常有用，有点像电源管理的“中断”支持。

- Verbose Power Management debugging

详细电源管理调试信息

5.2 Suspend to RAM and standby

内存挂起和待机功能

允许系统进入休眠状态，同时主内存仍然带电并保存相关信息，就如同内存挂起的状态（比如 ACPI S3 状态）。

5.2.1 Test suspend/resume and wakealarm during bootup

在启动时测试挂起/恢复和苏醒警告

这个选项将会在启动时挂起机器，然后在几秒之后苏醒，这通过使用 RTC 苏醒警告来实现。要开启这个选项，使用内核命令行如“test_suspend=mem”。

你可能需要你系统的 RTC 驱动进行静态链接，并确保进行这项测试时，RTC 可用。

5.3 Hibernation

休眠（也称为磁盘挂起）

允许磁盘挂起¹，在用户界面通常称为“休眠”或者更有诗意点，叫“冬眠”。STD 检查系统和将自己关闭，同时把检查点储存起来，在启动时恢复。

你可以这样挂起你的机器：echo disk>/sys/power/state，在此之前你需要在内核启动器的配置文件输入相关命令：resume=/dev/swappartition²。

原则上，这并不需要 ACPI 或者 APM，虽然 ACPI 会在最后一个环节使用，如果它可用的话。使用软件挂起的一个原因就是，固件在使用挂起功能时，比如挂起内存，常常无法可靠的在 Linux 下完成工作。

它会在你的 swap 里建立一个镜像。在下一次启动时，参数 resume=/dev/swappartition 会让内核检测已经储存的镜像，然后恢复系统的状态，并继续运行。如果你不希望之前的状态被加载，那么你可以使用 noresume 内核命令。注意，fsck³会在你的文件系统上运行，你将需要运行 mkswap⁴以使得 swap 分区能够用于挂起。

它也可以使用交换文件来达到这个目的。

现在你也许可以在启动时使用相关参数不进行挂起恢复，而且可以在后面能够用 resume 来恢复，但是此时你不可以使用 swap 分区/文件来恢复。因为你不能在恢复前使用文件系统；特别地，你绝对不可在挂起恢复之前，挂载任何的日志文件系统，否则它们会以各种莫名其妙的方式损坏！

5.3.1 Default resume partition

默认恢复分区

默认恢复分区是磁盘挂起工具软件所要查找的挂起镜像所处的分区。

在这指定的分区对每个人来说都不一样。它应该是一个合法的 swap 分区（至少现在是这样），并应在挂起前被开启。

分区的指定值可以通过下面的方式来覆盖：

resume=/dev/<other device>

¹简称为 STD

²即你的 swap 分区，比如/dev/hda8

³文件系统检查程序

⁴swap 分区建立程序

这将会把恢复分区设置为上述指定的分区。

注意目前还不能指定哪个设备用于储存挂起镜像。它只是简单地选择第一个可用的 swap 设备。

5.4 ACPI Support

高级电源配置接口支持

高级电源设置接口 (ACPI) 支持需要整合了 ACPI 的平台 (固件/硬件), 并且这个平台要支持操作系统和电源管理软件的安装。这个选项会给你的内核增加 70KB。

Linux ACPI 提供了相当强大的电源接口, 甚至可以取代一些传统的设置和电源管理接口, 包括 PNP BIOS (即插即用 BIOS) 规范, MPS (多处理器规范), 和 APM (高级电源管理) 规范。如果 ACPI 和 APM 同时被选上, 先被系统调用的起作用。

5.4.1 Deprecated /proc/acpi files

废弃的/proc/acpi 文件

为了保证对老版本的兼容, 这个选项允许使用已经废弃不用的/proc/acpi 文件。目前它们已经被/sys 的相应功能取代了。废弃的文件以及相应的取代包括:

/proc/acpi/sleep (/sys/power/state)

/proc/acpi/info (/sys/modules/acpi/parameters/acpica_version)

/proc/acpi/dsdt (/sys/firmware/acpi/tables/DSDT)

/proc/acpi/fadt (/sys/firmware/acpi/tables/FACP)

/proc/acpi/debug_layer (/sys/module/acpi/parameters/debug_layer)

/proc/acpi/debug_level (/sys/module/acpi/parameters/debug_level)

这个选项对于/proc/acpi 的文件不起什么作用, 如果相应功能没有在/sys 中存在的话。

选 N 删除/proc/acpi 文件, 它们已经移动到了/sys/下。

5.4.2 Deprecated power /proc/acpi directories

废弃的/proc/acpi 子目录

为了保证对老版本的兼容, 这个选项允许使用已经废弃不用的/proc/acpi 子目录。目前它们已经被/sys 的相应功能取代了。废弃的子目录以及相应的取代包括:

/proc/acpi/battery/* (/sys/class/power_supply/*)

/proc/acpi/ac_adapter/* (/sys/class/power_supply/*)

这个选项对于/proc/acpi 的目录不起什么作用, 如果相应功能没有在/sys 中存在的话。

选 N 删除/proc/acpi 子目录, 它们已经移动到了/sys/下。

5.4.3 Future power /sys interface

新能源管理接口/sys

选 N 将禁止能源/sys 接口。

5.4.4 Deprecated /proc/acpi/event support

废弃的/proc/acpi/event 支持

这是一个用户空间的后台程序，acpi，通常会去读取/proc/acpi/event 的内容，以应付所有的 ACPI 子系统产生的事件。

这些事件现在通过输入层，或者网络事件来发送到用户空间。

这个编译选项允许老的代码，以运行于传统的用户空间。不久以后，这将会在 CONFIG_ACPI_PROCS 下移动并删除。

在这选 Y 以使用老的行为。选 N，如果你的用户空间比内核 2.6.23 要新。

5.4.5 AC Adapter

AC 交流电源适配器

这个驱动给 AC 交流电源适配器提供支持，它指示出系统是否在 AC 下工作。如果你的系统可以在 AC 和电池状态下切换，选 Y。

5.4.6 Battery

电池

这个驱动通过/proc/acpi/battery 提供电池信息。如果你有使用电池的移动系统，选 Y。

5.4.7 Button

按钮

这个驱动通过电源、休眠、锁定按钮来提交事件。后台程序读取 /proc/acpi/event 来运行用户要求的事件，比如关机。这对软件控制关机是必要的。

5.4.8 Fan

风扇

这个驱动对 ACPI 风扇设备提供支持，允许用户模式的程序进行风扇的基本控制（开、关、状态显示）。

5.4.9 Processor

处理器

这个驱动以空闲管理者方式给 Linux 安装 ACPI，使用 ACPI C2 和 C3 处理器状态来节约电能，如果你的系统支持的话。一些 CPU 频率调节的驱动需要这个功能。

- Thermal Zone

温控区域

ACPI 温控区域驱动。大多数笔记本和台式机支持 ACPI 温控区域。强烈要求你选 Y，否则你的处理器可能会坏掉。

5.4.10 ASUS/Medion Laptop Extras

华硕笔记本扩展支持

这个驱动提供对 ASUS 笔记本的 ACPI 兼容性额外特性支持。对于 ASUS 厂家制造的一些 Medion 笔记本，它可能提供了相应的支持（比如 9675）。它使得所有的扩展按钮产生标准的 ACPI 事件，这通过 `/proc/acpi/events` 来实现；它还可以对更改显示亮度和输出、切换 LCD 背光开关提供支持；更为重要的，它允许你通过 LED 发光闪烁的情况来显示无线网络和邮件传输状态。

注意：显示切换部分的代码目前还是实验性的，对那些值的更改可能会锁住你的电脑。

所有的更改都可以通过目录 `/proc/acpi/asus` 的项目来完成。相应的权限可以通过设置 `asus_uid` 和 `asus_gid` 参数来实现。

如果你有一台 ASUS 笔记本，上面有扩展 ACPI 兼容功能，那么在这选 Y 或者 M。这个驱动仍然在研发之中，所以要是你的笔记本出现了意料之外的问题，请停用这个驱动并向我们报告（acpi4asus-user@lists.sourceforge.net）。

注意：这个驱动现在已经废弃不用，将很快被移除，并使用 `asus-laptop` 驱动来代替。

5.4.11 Toshiba Laptop Extras

Toshiba 笔记本扩展支持

这个驱动增加对传统的“自由式”Toshiba 笔记本特定系统设置的访问支持。这些笔记本能够通过少量的 BIOS 配置菜单项和 APM 支持来进行识别。

在这些机器上，所有的能源系统配置是通过 ACPI 来完成的。这个访问控制驱动没有涵盖通用的 ACPI 驱动，比如 LCD 亮度、视频输出等等。

这个驱动与非 ACPI Toshiba 笔记本驱动⁵有所不同。配置的访问是通过读写 `/proc` 树列下的文本文件来完成的，而不是 `/dev`。将来，将没有一个能源管理的功能暴露在外，它们将通过通用的 ACPI 驱动来处理。

如果你有一个传统自由式 Toshiba 笔记本（比如 Libretto L1 系列），那么选 Y。

⁵位于“Processor type and features”

5.4.12 Disable ACPI for systems before Jan 1st this year

在这年元旦不使用 ACPI⁶

输入一个 4 位数字，比如 2001，以使得 ACPI 在该年⁷的 1 月 1 日禁止使用。“acpi=force”选项可以覆盖这个设置。

输入 0 将禁用这个构架，这将使 ACPI 在任何年份使用，这也是默认值。

5.4.13 Debug Statements

调试语句

ACPI 驱动可以自定义报告详细的错误信息。选 Y 开启这项功能，这将让你的内核增加 50KB。

5.4.14 PCI slot detection driver

PCI 插槽探测驱动

这个驱动将尝试探索你系统上所有的 PCI 插槽，并在 `/sys/bus/pci/slots/` 上建立相关项目。这个特性可以帮助你关联 PCI 总线地址至你的插槽物理地址。如果不清楚，选 N。

5.4.15 Smart Battery System

袖珍电池系统

这个驱动对袖珍电池系统提供支持，依赖于 I2C (在选项 Device Drivers —> I2C support)。袖珍电池非常古老，也非常稀少，对于今天的 ACPI 支持的电池规范来说。

5.5 APM (Advanced Power Management) BIOS Support

高级电源管理 (APM) BIOS 支持

APM 是一个 BIOS 的规范，它通过多种不同的技术来节能。这对于电池供能的笔记本电脑来说尤其有用，通常它上面已经有内嵌 APM 的 BIOS 系统。如果你在这选 Y，系统时间将会在 RESUME 操作后重置，`/proc/apm` 设备将会提供电池状态信息，用户空间的程序会收到 APM 事件通告（比如电池状态改变）。

如果你在这选 Y，你可以通过 `apm=off` 选项来在内核启动时禁止 APM BIOS 的实际作用。

注意 APM 支持目前已经在多 CPU 的机器上被禁止。

⁶防止老机器的千年虫问题

⁷针对 DMI BIOS 时间

要使用 APM，你还需要相应软件的支持。

这个驱动不会降低磁盘的转速，也不会关闭 VESA 融合的“绿色”显示器。

这个驱动不支持 TI 4000M TravelMate 和 ACER 486/DX4/75，因为它们没有相配的 BIOS。很多绿色桌面机器同样也没有相应的 BIOS，因此这个驱动很可能会导致这些机器在启动时崩溃。

通常的，如果你的机器没有电池，那么没有太大必要使用这个驱动，你应该选 N。如果你得到随机内核 OOPS 信息或者莫名其妙地重启，试试禁止/允许这个选项（或者在你的 BIOS 禁止/允许 APM）。

如果你碰到很多诡异的问题，你可以试试：

- 确保你有足够的 swap 空间，并且已经开启。
- 给内核设置“no-hlt”选项。
- 打开内核浮点运算器，并给内核设置“no387”选项。
- 给内核设置“floppy=nodma”。
- 给内核设置“mem=4M”（因此会禁止使用除了首部 4MB 内存以外的空间）。
- 确定 CPU 没有超频。
- 访问网站 <http://www.bitwizzard.nl/sigll/>
- 在 BIOS 设置中禁止缓冲。
- 给显卡加装风扇，或者更换显卡 RAM。
- 给 CPU 装更好的风扇。
- 更换内存。
- 更换主板。

选 M 编译为模块，名为：apm。

5.6 CPU Frequency scaling

CPU 变频控制

CPU 变频控制允许你在运行中改变 CPU 的时钟速度。这是对于节约电能来说是一个不错的主意，因为 CPU 频率越低，它消耗的电能越少。

注意这个驱动不会自动改变 CPU 的时钟速度，你要么允许动态的频率调节器（看下面），要么使用用户工具。

如果不清楚，选 N。

5.6.1 Enable CPUfreq debugging

是否允许调试 CPU 改变主频的功能，如果要调试，还需要在启动时加上参数。

内核命令行：cpufreq.debug=<value>

其中，相应的值有：

- 1：变频技术的内核调试
- 2：变频技术的驱动调试
- 3：变频技术的调节器调试

5.6.2 CPU frequency translation statistics

CPU 频率统计功能

这个驱动通过 sysfs 文件系统输出 CPU 频率统计信息。

选 M 编译为模块，名为：cpufreq_stats。

如果不清楚，选 N。

- CPU frequency translation statistics details

CPU 频率统计功能（详细）

这里通过 sysfs 文件系统显示详细的 CPU 频率表。

如果不清楚，选 N。

5.6.3 Default CPUFreq governor (performance)

默认主频调节方式

默认的主频调节方式，圆括号内的是你选择的结果，这里表示以性能为主。

- performance 性能优先

这个调节器把 CPU 的频率设置得尽可能高。

选 M 编译为模块，名为：cpufreq-performance。

如果不清楚，选 Y。

- powersave 节能

这个调节器把 CPU 频率调节得尽可能低。

选 M 编译为模块，名为：cpufreq-powersave。

如果不清楚，选 Y。

- userspace 用户定义

如果你想要自己定义 CPU 频率，或者有一个用户空间的程序可以动态设置 CPU 频率，那么选 Y。

选 M 编译为模块，名为：cpufreq-userspace。

如果不清楚，选 Y。

- ondemand 根据需要

这个驱动增加一个动态的 CPU 频率策略调节器。这个调节器进行周期性的轮询，然后根据 CPU 的利用情况来改变频率。这个支持依赖于 CPU 的性能，以能够快速切换频率（比如非常低延迟地转换频率）。

选 M 编译为模块，名为：cpufreq-ondemand。

如果不清楚，选 N。

- conservative 传统

这个驱动有点像 ondemand 调节器，无论是在源代码上还是在目的上。不同点是它是为电池环境优化的。频率将会非常好的增加或者减少，而不是突然跳到 100%。

如果你是一个桌面系统，那么你应该认真考虑一下 ondemand 调节器。如果你使用的是笔记本电脑、PDA 或者 AMD64 位的电脑（在 CPU 的最低和最高频率之间的分步调节可能会导致问题），那么你可能需要使用这个调节器。

选 M 编译为模块，名为：cpufreq-conservative。

如果不清楚，选 N。

5.6.4 CPUFreq processor drivers

变频驱动模块

- ACPI Processor P-States driver ACPI 处理器状态驱动
- AMD Mobile K6-2/K6-3 PowerNow! AMD 移动版 K6 处理器的变频驱动
- AMD Mobile Athlon/Duron PowerNow! AMD 移动版毒龙、雷鸟的变频驱动
- AMD Opteron/Athlon64 PowerNow!
- Cyrix MediaGX/NatSemi Geode Suspend Modulation Cyrix 处理器的变频驱动。
- Intel Enhanced SpeedStep Intel 的移动变频技术支持。
 - Use ACPI tables to decode valid frequency/voltage pairs 使用 BIOS 中的主频/电压参数。
 - Built-in tables for Banias CPUs 迅驰一代的主频/电压参数。
- Intel Speedstep on ICH-M chipsets (ioport interface) Intel ICH-M 移动南桥芯片组的支持

- Intel Pentium 4 clock modulation P4 处理器的时钟模块支持。
- Transmeta LongRun Transmeta 处理器的支持。
- VIA Cyrix III Longhaul VIA Cyrix 处理器的支持。
- VIA C7 Enhanced PowerSaver 节能增强型 VIA C7 支持

5.6.5 shared options

共享选项

- /proc/acpi/processor/../performance interface
/proc/acpi/processor/../performance 接口

这里开启已经废弃不用的/proc/acpi/processor/../performance 接口。这对于调试很有用，通用的、交叉构架的 CPU 变频接口可能会用上它。

如果不清楚，选 N。

5.7 CPU idle PM support

CPU 空闲 PM⁸支持

CPU 空闲是一个通用的框架，用于支持软件控制的处理器空闲时电源管理。它包括模块化的交叉平台调节器，能够在运行时进行交换。

如果你使用的是 ACPI 平台，那么你应该要在这选 Y。

5.8 Memory power savings

内存节能

⁸即 Power Management

Chapter 6

Bus options 总线选项

一般来说，很多总线的选项一定要编译进内核，而不能编成模块。否则会导致内核启动失败。

6.1 PCI support

PCI 总线支持

找找你的主板资料，看看你用的是不是 PCI 主板。PCI 是总线系统的名称，是 CPU 用来与其他设备进行通信的通道。其他总线系统有 ISA、EISA、MCA 和 VESA。如果你有 PCI，选 Y。否则，选 N。

6.1.1 PCI access mode

PCI 访问模式

在 PCI 系统中，BIOS 可以检测 PCI 设备和确定它们的设置。但是，一些老的 PCI 主板有 BIOS 问题，如果这里选上会让系统当机。同时，一些嵌入式的基于 PCI 系统没有任何 BIOS。Linux 可以在不使用 BIOS 的情况下尝试直接检测 PCI 硬件。

选上这个以后，你可以设定 Linux 如何检测 PCI 设备。如果你选择“BIOS”，BIOS 会用到。你选“Direct”，BIOS 不会用到。如果你选“MMConfig”，PCI 加速的 MMCONFIG 会用到。如果你选“Any”，内核先用 MMCONFIG，然后“Direct”，最后才是“BIOS”如果前面的都无法工作。如果不清楚，选“Any”。

- BIOS
- MMConfig
- Direct
- Any

6.2 PCI Express support

PCI Express 支持

这里自动支持 PCI Express 端口总线。用户可以选择 Native Hot-Plug support, Advanced Error Reporting support, Power Management Event support, Virtual Channel support 4 个选项来支持 PCI Express 端口（启动或者切换）。

6.3 Message Signaled Interrupts(MSI and MSI-X)

信息信号中断 (MSI 和 MSI - X)

这里允许设备驱动使用 MSI。它允许一个设备通过写在自己 PCI 总线内存模块里的代码产生中断，而不是使用一个设备 IRQ 针脚。

在内核启动时，可以禁止 PCI MSI，通过在内核启动参数设置‘pci=noms’。这在整个系统禁用 MSI。

如果你不知道怎么做，选 N。

6.4 Enable deprecated pci_find_* API

允许废弃的 pci_find_*接口

在这选 Y，如果你想要支持已经废弃不用的接口 pci_find_slot() 和 pci_find_device()。很多驱动已经转化为使用正确的热插入接口，所以这个选项在 include/exclude 的内容只有少数驱动会用到。

6.5 PCI Debugging

PCI 调试

选 Y，如果你希望 PCI 核心产生大量的调试信息至系统日志。如果你在 PCI 支持上有问题，想要看看到底发生了什么的话，在这选 Y。

如果不清楚，选 N。

6.6 Interrupts on hypertransport devices

超传输设备中断

这里允许本地的超传输设备使用中断。

如果不清楚，选 Y。

6.7 ISA support

ISA 总线支持

看看你主板上是否有 ISA 插槽。ISA 是比较老的总线，现已基本被 PCI 取代。如果你没有老式的 ISA 设备，可以选 N，否则，选 Y。

6.8 MCA support

MCA 支持

IBM PS/2 上的总线，现已淘汰。微通道总线 IBM 的台式机和笔记本上可能会有这种总线，包括它的 p 系列、e 系列、z 系列机器上都用到了这种总线。

6.9 NatSemi SCx200 support

NatSemi SCx200 支持

这里提供对松下（现在是 AMD）的半导体处理器 Geode 的基本支持。这个驱动探测一些芯片设备上的 PCI-ID，所以它是其他 scx200_*驱动的不错的通用选项。

如果编译为模块，名为：scx200。

6.10 One Laptop Per Child support

人人电脑¹支持

对于 OLPC XO 硬件的一些特殊属性增加支持。

6.11 PCCARD (PCMCIA/CardBus) support

PCCARD(PCMCIA/CardBus)支持

在这选 Y，如果你想要在你的 Linux 电脑上连接 PCMCIA 或者 PC 卡。这些卡大小和信用卡相当，就如同网卡、modem 或者硬设备一样经常使用于笔记本电脑上。目前有两种种类的卡：16 位的 PCMCIA 和 32 位的 CardBus。

选 M 编译为模块，名为 pcmcia_core。

¹OLPC 就是所谓的人人电脑，它是由美国麻省理工大学的尼葛洛庞帝教授提出的。这种电脑是为了满足广大发展中国家的中小學生而设计的，其最吸引人的一点便是它的售价只有 100 美元，也就是折合人民币 800 元，这对发展中尤其是中国的学生有很大的吸引力。

6.12 PCI Hotplug Support

PCI 热插拔支持

选 Y，如果你的主板有 PCI 热拔插控制器，这允许你热拔插 PCI 卡。

选 M，将编译为模块，叫做 `pci_hotplug`。

如果不清楚，选 N。

Chapter 7

Executable file formats 可执行文件格式

7.1 Kernel support for ELF binaries

ELF 二进制文件内核支持

ELF（可执行和可链接格式）是一种用来连接不同架构和操作系统的可执行文件、库函数格式。选 Y，你的内核可以运行 ELF 二进制文件，这也使你的内核增大 13KB。

ELF 现在基本代替了传统的 a.out 格式（QMAGIC and ZMAGIC 用到），因为它是可移植的（可移植不代表它可以直接运行在不同构架和操作系统上），而且建立相关运行库文件非常容易。很多新的可执行文件都用 ELF 格式发布，你在这里当然要选择 Y。

7.2 Write ELF core dumps with partial segments

用部分参数写入 ELF 核心信息转储

ELF 核心信息转储文件描述了每个崩溃进程的内存页面，它能够保持或者省略每个内存的内容。默认的，不可修改的文本映射内容将被省略。

对于一个 ELF 对象中不可修改的文本映射，包括文件的第一页，核心转储使得它可以识别文件中的编译 ID 字段，而不需要花费 i/o 和磁盘空间来转储所有的文本。但是，GDB¹ 6.7 之前的版本会被 ELF 核心转储文件的格式弄混。

核心转储行为能够通过使用伪文件 /proc/PID/coredump_filter 进行每个进程的控制。这些设置是可继承的。

这个配置选项在机器启动时修改了 coredump_filter 的默认设置。如果不清楚，选 N。

¹gdb 是 Linux 下的程序调试工具。

7.3 Kernel support for a.out and ECOFF binaries

对 a.out 和 ECOFF 二进制文件的支持

A.out (Assembler.OUTPUT)是一种二进制文件格式，它用在最早的 UNIX 版本中。Linux 在 QMAGIC 和 ZMAGIC 两个镜像中使用 A.out，直到它最近被 ELF 取代。ELF 的转变开始于 1995 年。这个选项主要是给研究历史的人提供感兴趣的信息，或者你要是有那个年代的文件，你需要这个选项。

大多数人在这可以选 N。如果你认为你可能会用到这个格式，选 M 编译成模块。模块名为 `binfmt_aout`。如果你系统的关键部件（比如 `/sbin/init` 或者 `/lib/ld.so`）仍然是 a.out 格式的话，你要在这选 Y。

7.4 Kernel support for MISC binaries

内核对 MISC 二进制文件的支持

如果你在这选 Y，它将可以将 wrapper-driven 二进制格式嵌入内核。当你使用一些程序的解释器时，比如 Java, Python, .NET 或者 Emacs-Lisp，或者当你经常通过 DOS 仿真器运行 DOS 程序时，它将非常有用。当你在这个选项选 Y，你可以简单地通过在 shell 打相应命令运行以上的程序，Linux 可以自动匹配正确的格式。

要使用 `binfmt_misc` 你可能需要挂载它：

```
mount binfmt_misc -t binfmt_misc /proc/sys/fs/binfmt_misc
```

你可以选 M 作为模块，以后再加载，模块名为 `binfmt_misc`。如果你不知道怎么办，选 Y。

Chapter 8

Networking support 网络支持

网络支持

你在这应该选 Y，除非你真正知道你在干什么。原因是一些程序必须要内核的网络支持才能工作，即使你的机器没有真正连上网络。

如果你是从老的内核升级的，你要考虑升级你的网络工具，因为内核中有所更改，而相应的工具总是联合其中，依赖于内核的。

8.1 Networking options

8.1.1 Packet socket

包套接字¹

包协议被很多程序用来直接连接网络设备，而不需要内核执行中间网络协议，比如 tcpdump。如果你要它们工作，选 Y。

选 m 编译成模块。模块名：af_packet。

如果不清楚，选 Y。

- Packet socket: mmaped IO

包套接字：mmaped IO

如果你选 Y，包协议驱动将使用一个 IO 结构，这可以加速连接。

如果不清楚，选 N。

8.1.2 Unix domain sockets

UNIX 域套接字

如果你在这选 Y，你将编入 UNIX 域套接字支持。套接字是标准 UNIX 结构，用来建立和访问网络连接。很多程序比如 X Window 系统和系统信息使用这些套接

¹计算机为 TCP/IP 数据转换所保留的一系列存贮地址

字即使你的机器未连接上网络。除非你在一个嵌入式系统上工作，你必须要在这选 Y。

选 M 编译成模块。模块名： `unix` 。注意一些重要的服务不会正确工作如果你选 M ，却忘记加载模块。

选 Y ，除非你知道在干什么。

8.1.3 Transformation user configuration interface

用户设置接口转换

对用户设置接口转换功能进行支持，比如本地 Linux 工具用到的 IPsec 。

如果不清楚，选 Y 。

8.1.4 PF_KEY sockets

PF_KEY 套接字

PF_KEYv2 套接字家族，与 KAME 相一致。如果你要使用 KAME 端口的 IPsec 工具，你需要这个选项。

选 Y ，除非你知道在干什么。

8.1.5 TCP/IP networking

TCP/IP 网络

这些是在 Internet 和本地以太网使用最多的协议。强烈推荐选 Y （这会让内核增大 144KB ），因为一些程序，比如 X window 系统使用 TCP/IP ，即使你的机器没有连上网络。

你会得到自回环设备（ LOOP - BACK ），这样你可以 ping 自己（有趣！）

如果你选 Y ，并在后面的“/proc file system support”和“Sysctl support”也选上，你将可以通过 `/proc/sys/net/ipv4/*` 的虚拟文件来改变 TCP/IP 代码的工作方式。

简之：选 Y 。

- IP: multicasting

IP: 多播

这是在一个时间里访问多个地址的代码，将会增大内核 2KB 。你需要多播功能，如果你想要加入 MBONE （多路广播主干网），一个高带宽的，有声音和影像的高层广播网络。

对大多数人，在这可以安全地选 N 。

- IP: advanced router

IP: 高级路由

如果你想让你的 Linux 机器作成路由，比如，用来转信和回环网络包，选 Y。你将可以得到更多的选项来精确地控制你的路由进程。

在这的回答不会直接影响到内核。选 N 只是让配置程序跳过所有的关于高级路由的问题。

注意你的机器只可以作为路由如果你在内核允许 IP 转信。你可以通过在下面的选项选 Y “/proc file system support” 和 “Sysctl support”，然后在启动时，在挂载 /proc 后执行命令

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

如果你开启 IP 转信，你可能需要 `rp_filter`，它可以自动拒绝输入包，如果来源地址不在信任列表中。这有着良好的安全性，因为它可以防止 IP 欺骗，但是也可能会导致问题，比如你使用的是不对称路由（路由到主机的信息和主机到路由的信息使用不同的线路），或者你操作的非路由主机有多个 IP 地址。

要关闭 `rp_filter` 用：

```
echo 0 > /proc/sys/net/ipv4/conf/<device>/rp_filter
```

或者

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

如果不清楚，选 N。

- IP: policy routing

IP: 路由策略

通常地，一个路由器描述了如何处理接收到的基于唯一终点地址的包。如果你在这选 Y，Linux 路由将能够考虑包的源地址。此外，包的 TOS²域可以用于路由描述。

如果不清楚，选 N。

- IP: equal cost multipath

IP: 多路径等同花销

通常地，路由表指定了一个单独的行为，以决定和处理给定的包。如果你在这选 Y，它将可以用几种行为处理一个包类型，表现出来的结果就是对这些包指定了几种可选的传输路径。路由器把这些路径当作是同等的开销，同时非确定性地选择它们的其中之一，当一个匹配的包到达的时候。

- IP: verbose route monitoring

IP: 详细路由监视

如果你在这选 Y（这也是推荐的作法），内核将可以打印详细的路由信息，比如接收到陌生包时的警告，这可以作为被攻击的证据，或者表明系统的某个地方没有配置好。这个信息由后台程序 `klogd`³ 负责处理，它对于内核的信息可以作出响应。

²服务类型

³“man klogd”查看更多信息

- IP: kernel level autoconfiguration

IP : 内核级别自动配置

这里允许在内核启动时对 IP 地址和路由表进行自动配置，由内核的命令行或者由 BOOTP 或 RARP 协议提供信息。你需要选 Y，如果你磁盘空间很少，但是又需要进行网络访问（同时你需要在后面的“Root file system on NFS”中选 Y），因为其他机器在启动脚本中（而不是内核中）进行网络配置。

- IP: DHCP support

IP : DHCP 支持

如果你想让你的 Linux 机器在启动时通过网络的 NFS 协议挂载其他机上的整个启动文件系统（包括根目录/），同时你想要让你的 IP 地址在启动时被 DHCP 协议（一种专门用来在启动时自动获取 IP 地址的协议）自动获取，选 Y。假如网卡的启动 ROM 模块已经设计为给 Linux 提供 DHCP 支持，提供了足够的内核命令信息，你可以在这选 N。

如果不清楚，选 Y。注意如果你要用 DHCP，DHCP 服务器必须要在网络中可用。

- IP: BOOTP support

IP: BOOTP 支持

如果你想让你的 Linux 机器在启动时通过网络的 NFS 协议挂载其他机上的整个启动文件系统（包括根目录/），同时你想要让你的 IP 地址在启动时被 BOOTP 协议（一种专门用来在启动时自动获取 IP 地址的协议）自动获取，选 Y。假如网卡的启动 ROM 模块已经设计为给 Linux 提供 BOOTP 支持，提供了足够的内核命令信息，你可以在这选 N。如果不清楚，选 Y。注意如果你要用 BOOTP，BOOTP 服务器必须要在网络中可用。

- IP: RARP support

IP: RARP 支持

如果你想让你的 Linux 机器在启动时通过网络的 NFS 协议挂载其他机上的整个启动文件系统（包括根目录/），同时你想要让你的 IP 地址在启动时被 RARP 协议（一种专门用来在启动时自动获取 IP 地址的协议，比 DHCP 和 BOOTP 更老）自动获取，选 Y。假如网卡的启动 ROM 模块已经设计为给 Linux 提供 RARP 支持，提供了足够的内核命令信息，你可以在这选 N。如果不清楚，选 Y。注意如果你要用 RARP，RARP 服务器必须要在网络中可用。

- IP: tunneling

IP:隧道传输

隧道传输意味着将数据从一个协议类型压缩到另一个协议类型中，并通过理解这个压缩协议的一个隧道进行传输。这个特殊的隧道驱动包含 IP 到 IP 的压缩，看起来没什么意义，但是当你需要你的（或者别人的）机器在不同的网络上使用，或者使用移动 IP 设备（可以让你的笔记本在不同的网络使用，而不需要改变 IP）的时候很有用。

选 *y* 会生成两个模块（可以在运行中的内核嵌入和移除）。大多数人不需要这个选项，可以选 *N*。

- IP: GRE tunnels over IP

IP : GRE 隧道

隧道传输意味着将数据从一个协议类型压缩到另一个协议类型中，并通过理解这个压缩协议的一个隧道进行传输。这个特殊的隧道驱动包含 GRE（通用路由压缩），可以将 IPv4 或者 IPv6 压缩到现有的 IPv4 基本组织中。这个驱动很有用，如果网络终点是 Cisco 路由。Cisco 有点像 GRE，但是比其他的 Linux 隧道驱动更好（看上面的选项）。另外，GRE 允许多播通过隧道重新分配。

- IP: broadcast GRE over IP

IP : 广播 GRE

GRE/IP 的一个用途是建立广播 WAN（广域网），它有点像普通的以太 LAN，但是可以传递给整个 Internet。如果你要这样，选 *Y* 和下面的“IP multicast routing”。

- IP: multicast routing

IP:多播路由

如果你想要你的机器变成一个多终端的 IP 包路由，选这项。在 MBONE 需要这功能。MBONE 是一个高带宽的网络，在 Internet 的上层传播大量的声音和视频广播。为了达到这个目的，你可以要运行 `mroute`。如果你没有听说过它，你不需要它。

- IP: TCP syncookie support (disabled per default)

IP : TCP cookie 同步支持（默认关闭）

普通 TCP/IP 网络对于“同步泛滥”攻击是开放式的，没有阻碍。这种拒绝服务式的攻击让合法的远程用户无法连接到你的计算机。这种攻击只需要简单的工作就可以从网络的任意地方进行。

同步 cookies 针对这种攻击提供了保护。如果你选 *Y*，TCP/IP 栈会使用一种“SYN cookies”的密码协议，使得合法的用户能继续保持连接，即使你的机器已经遭受攻击。对于合法的用户，没有必要更改他们的 TCP/IP 软件，SYN cookies 对于它们是透明的。

如果你被同步泛滥攻击，内核报告的源地址很可能被攻击者伪造。它仅仅是追踪实际源用的包，而不是真实的。

SYN cookies 将会在服务器过负荷的时候制止客户端对于错误报告的纠错行为。如果这经常发生，最好将它关闭。

如果你选 *Y*，注意 SYN cookies 不会自动运行。你可以在选项“/proc file system support”和“Sysctl support”中选 *Y*，然后在系统启动时，挂载了 /proc 后运行相关命令来运行它：

```
echo 1 >/proc/sys/net/ipv4/tcp_syncookies
```

如果不清楚，选 N。

- IP: AH transformation

IP : AH 转换。

给 IPsec AH 提供支持。

如果不清楚，选 Y。

- IP: ESP transformation

IP : ESP 转换。

给 IPsec ESP 提供支持。

如果不清楚，选 Y。

- IP: IPComp transformation

IP : IPCOMP 转换

支持 IP 负荷压缩协议，IPsec 一般需要这个协议。

如果不清楚，选 Y。

- IP: IPsec transport mode

IP : IPsec 传输模式。

支持 IPsec 传输模式。

如果不清楚，选 Y。

- IP: IPsec tunnel mode

IP : IPsec 隧道模式

支持 IPsec 隧道模式。

如果不清楚，选 Y。

- IP: IPsec BEET mode

IP: IPsec BEET 模式

支持 IPsec BEET 模式

如果不清楚，选 Y。

- Large Receive Offload (ipv4/tcp)

大型接收卸载(ipv4/tcp)

支持大型接收卸载(ipv4/tcp)

如果不清楚，选 Y。

- INET: socket monitoring interface

INET : 套接口监控界面

对于一些 Linux 工具比如 ss 提供 INET (TCP, DCCP 等)套接口监控界面支持。

如果不清楚，选 Y。

- TCP: advanced congestion control

TCP 高级拥挤控制

对于各种 TCP 拥塞控制模块提供选择支持

几乎所有的用户可以安全地在这里选 N，安全的默认选项将会设置。

如果不清楚，选 N。

- The IPv6 protocol

IPv6 协议

这里支持 IPv6 本。你将仍然可以使用传统的 IPv4 网络。

选 M 编译成模块，模块名为：ipv6。

- IPv6: Privacy Extensions support

IPv6: 隐私扩展支持

这是对自动配置 IPV6 地址时的隐私扩展支持。有这个选项时，额外的、定期改变的、随机虚拟的、共用作用域的、单一传播的地址将会指派给你的接口。

我们使用我们标准的随机虚拟算法生成随机的接口 ID，类似的算法有 RFC 3041。

默认地，内核不会生成临时地址，要使用临时地址，输入命令：

```
echo 2 >/proc/sys/net/ipv6/conf/all/use_tempaddr
```

- IPv6: Router Preference (RFC 4191) support

IPv6: 路由首选项 (RFC 4191) 支持

路由首选项是可选的扩展功能，它可以让路由宣告相关信息，以改进主机的能力。它使主机能够选择更合适的路由，特别是主机在多用户的网络中时尤其有效。

如果不清楚，选 N。

- IPv6: Route Information (RFC 4191) support

IPv6: 路由信息(RFC 4191)支持

这是实验中的路由信息的支持项。

如果不清楚，选 N。

- IPv6: AH transformation

IPv6: AH 转换

支持 IPsec AH。

如果不清楚，选 Y。

- IPv6: ESP transformation
IPv6: ESP 转换
支持 IPsec ESP。
如果不清楚，选 Y。
 - IPv6: IPComp transformation
IPv6: IPComp 转换
支持 IPsec IPComp。
如果不清楚，选 Y。
 - IPv6: IPsec transport mode
IPv6 : IPsec 传输模式。
支持 IPsec 传输模式。
如果不清楚，选 Y。
 - IPv6: tunnel transformation
IPv6 : IPsec 隧道转换。
支持 IPsec 隧道转换。
如果不清楚，选 Y。
 - IPv6: IPsec BEET mode
IPv6: IPsec BEET 模式
支持 IPsec BEET 模式
如果不清楚，选 Y。
 - IPv6: IPv6-in-IPv4 tunnel (SIT driver)
IPv6: IPv6 到 IPv4 隧道
隧道传输意味着将数据从一个协议类型压缩到另一个协议类型中，并通过理解这个压缩协议的一个隧道进行传输。这个特殊的隧道驱动包含 IPv6 到 IPv4 的套接口。如果你想要将两个 IPv6 网络通过唯一的 IPv4 路径进行连接的话，这非常有用。
选 M 生成模块，名叫 sit.ko。如果不清楚，选 Y。
 - IPv6: IP-in-IPv6 tunnel
IPv6: IP 到 IPv6 隧道
支持 IPv6 到 IPv6 隧道和 IPv4 到 IPv6 隧道，这在 RFC 2473 中描述。
如果不清楚，选 N。
- NetLabel subsystem support
网络标签子系统支持
网络标签提供清晰的网络包标签协议，比如 CIPSO 和 RIPS0。
如果不清楚，选 N。

8.1.6 Security Marking

这个选项给网络套接字提供安全标志，有点像 `nfmark`，但主要目的是安全。如果不清楚，选 `N`。

8.1.7 Network packet filtering framework

网络包过滤框架

网络过滤器是一个用于过滤和切除网络传输包的构架。最普遍的用法就是以防火墙的形式运行 `Linux`，用于保护连接在 `Internet` 上的局域网。

内核提供的防火墙叫做“包过滤器”，它可以通过包类型、来源地、终点等来拒绝相应的网络包。另外一种防火墙是基于代理的。它更安全，但是更强制（经常打断你），安装更麻烦。它审查网络通讯更直接，能够修改包，能够理解更高层次的协议。这是包过滤器所做不到的。

你可以选 `Y`，如果你想要让你的 `Linux` 机器做局域网到 `Internet` 的网关，但是你的机器又没有通用有效的 `IP` 地址。这叫做“伪装”：比如如果局域网的一台机器想要对外发送信息，你的机器可以“伪装”成那台机器。它传输信息到相应的终点，但是修改了包，使它看起来像是通过防火墙本身过来的一样。它通过两种方式工作：如果外部主机有响应，`Linux` 会静默地传输信息到相应的内网机器。在这种方式下，内网的机器对外网来说是完全不可见的，即使它们可以与外网通讯。另一种方式是你运行通用的可见的服务器，通过在伪装的局域网络中使用一种叫做 `portforwarding`（埠传输）的构架。伪装也被叫做 `NAT`（网络地址转换）。

另外一种包过滤器的用法是“透明代理”：如果一个内网机器试图连接外部主机，你的 `Linux` 机器可以透明地转信至内网的服务器，通常是一个缓冲代理服务器。

还有一种用法是建立桥接式防火墙。使用桥接式网络包过滤可以使 `IP` 表“看到”桥接传输。为了在底层网络和以太网协议实现桥接式过滤，你要使用 `ebtables`（桥接网络过滤配置）

这里有很多模块，可以代替以前的伪装、包过滤、透明代理和埠传输工具。

注意你要在下面的“`Fast switching`”选项中选 `N`，如果你要在这选 `Y`。因为目前 `Fast switching` 会忽略包过滤器。

如果你编译内核的目的之一是为了当路由，选 `Y`。如果仅仅是做为平常的主机，选 `N`。如果不清楚，选 `N`。

- Network packet filtering debugging

网络包过滤调试

- Advanced netfilter configuration

高级网络过滤配置

- Core Netfilter Configuration

核心网络过滤架配置

- IP virtual server support
IP 虚拟服务器支持
- IP: Netfilter Configuration
IP:网络过滤架配置
- IPv6: Netfilter Configuration
IPv6:网络过滤架配置

8.1.8 Asynchronous Transfer Mode (ATM)

异步传输模式

ATM 是一种高速度的网络技术，它应用于局域网和广域网中。它使用修改过的包尺寸，而且它的连接是定向的，只会占用非常小的带宽。

为了加入 ATM 网络，你的 Linux 机器需要 ATM 网络卡。如果你有，在这选 Y，并在下面相关卡的驱动选 Y。

注意你需要一组用户空间的程序来真正运行 ATM。

8.1.9 802.1d Ethernet Bridging

802.1d 以太网桥接

如果你选 Y，你的 Linux 机器将可以做为以太网桥接，对于用户来说，它可以让连接上的不同的以太网段表现为一个段。几个桥接段还可以通过 IEEE 802.1 生成树算法再连接为更大的网络。这是一个标准，Linux 可以用这个标准来与其他第三方的桥接产品进行协作。

为了使用以太网桥接，你需要一些相关的配置工具。

如果你同时开启了 IP 表支持和桥接支持，你将让你的桥变成桥接式 IP 防火墙。IP 表将可以看到被桥接的 IP 包，所以你在设置防火墙规则时，需要把这考虑进去。桥接时允许 ARP 表支持将让 ARP 表在连接链中看到被桥接的 ARP 包。

选 M 编译成模块，名为：bridge。

如果不清楚，选 N。

8.1.10 802.1Q VLAN Support

802.1Q VLAN 支持

选上这个，你可以在你的以太网接口中建立 802.1Q VLAN 接口。802.1Q VLAN 支持大多数以太网接口，包括防火墙、桥接，当然也包括 IP 传输。你需要 'vconfig' 的一个工具来使用 VLAN。

选 M 编译成模块，名为：8021q。

如果不清楚，选 N。

8.1.11 DECnet Support

DEC 网络支持

DECnet 网络协议在 Digital(现在是 Compaq)的很多产品上使用。它提供可靠的流和顺序包连接，可以与 TCP/IP 进行通信。

当你使用 DECnet 的时候，确保你在“/proc file system support”和“Sysctl support”中选择 Y，因为它运行的时候，你需要 sysctl 的支持才能更改配置。

选 M 编译成模块，名为 decnet。

8.1.12 ANSI/IEEE 802.2 LLC type 2 Support

ANSI/IEEE 802.2 LLC type 2 支持。

这是一个逻辑的连接层，提供定向的连接支持。如果你需要 PF LLC 套接口的支持，选 Y。

8.1.13 The IPX protocol

IPX 协议

这是对 Novell 的网络协议，IPX 的支持选项。IPX 在局域网的 Windows 机器上广泛使用。如果你要访问 Novell NetWare 文件或者打印服务，你需要这个选项。为了保证工作，你还需要在下面的选项“NCP file system support”中选 Y。

IPX 类似于 IP，而运行于 IPX 上层的 SPX 类似于 TCP。这里同样有实验中的对 SPX 的支持选项（看下面的“SPX networking”）。

IPX 驱动会让你的内核增大 16KB。选 M 编译为模块，叫做 ipx。除非你真的需要让你的 Linux 机器连接上 Novell 网络，否则可以选 N。

- IPX: Full internal IPX network

IPX：完全内部 IPX 网络

每个 IPX 网络都有自己的地址标识符。有时给你的 Linux 机器加上一个 IPX 网络地址是比较有用的（比如如果你的机器在不同的 IPX 网络中作为文件服务器：它将可以从各个地方用一个地址进行访问）。它工作的方式是在你的机器内部建立虚拟的“网络”，并且给它指派一个 IPX 地址。如果你想要这么做，选 Y。

完全内部 IPX 网络允许你在内部网络的不同节点指派套接字。这通过在相关的绑定软件调用中（BIND，一个域名管理软件）给域 sipx_node 赋予套接地地址值来实现。所以当绑定一个套接字到主网络的时候，程序总可以对节点域进行零值初始化。在这种情况下，当内部网络建立时，内核会给出默认节点，从而指派给套接字。为了使完全内部 IPX 网络工作，包监听主网络的功能将被禁止，这些包是特殊的套接字到套接字之间的交互式包。这可能会破坏一些程序，特别是 RIP/SAP 后台程序。

如果不清楚，选 N。

8.1.14 Appletalk protocol support

Appletalk 协议支持

AppleTalk 是苹果电脑用于连接网络的协议。如果你的 Linux 机器要连接这样的网络，选 Y。你将要用到 netatalk 包，这样你的机器可以在 Macs 和苹果网络中做为文件和打印服务器。

EtherTalk 是 AppleTalk 的以太网版本，LocalTalk 是 AppleTalk 的专有的串行连接版本，它更慢，当然更便宜。Linux 完全支持 EtherTalk 和 LocalTalk。

选 M 编译为模块，名为 appletalk。通常你肯定要编译它为模块，这样你可以重启你的 AppleTalk 栈，而不需要重启电脑。我听说苹果公司对 GNU 的抵制已经结束，所以政治上端正的人（应该是对苹果公司的相关政策有意见的人）都可以在这里选 Y。

8.1.15 QoS and/or fair queueing

QoS and/or 平衡队列

当内核有几个包要通过网络设备传输，它必须要决定哪个先传送，哪个后传送，哪个要丢弃。这就是队列规则，一些不同的算法就是用来使之传送得更为“平衡”。

如果你选 N，你将使用标准的包调度器，叫做 FIFO（先到先提供服务）。如果你选 Y，你将可以选择一些算法，可以附属于不同的网络设备。这非常有用，比如你的一些网络设备是实时设备，它将需要最小化的数据浮动比率；或者你需要限制数据最大流量，用于匹配某些标准。这段代码是处于实验中的。

服务质量 (QoS) 支持可以使你在你的 Linux 路由中使用分化型的服务 (diffserv) 和资源保留协议 (RSVP)，如果你在相应的选项选 Y 的话。

如果你在这选 Y，同时在“/proc file system”也选 Y，你将可以通过文件 /proc/net/psched 来阅读包调度器的状态。

可用的调度器在下面的选项都列了出来，你可以在喜欢的地方选 Y。如果不清楚，选 N。

8.1.16 Network testing

网络测试。

- Packet Generator (USE WITH CAUTION)

包产生器（小心使用）

这个模块通过设置的频率和特定的接口，向预制包中注入数据。它用于网络接口的负荷测试和程序分析。如果你不知道上面说了什么，你不需要它，选 N。

选 M 编译为模块，名为：pktgen。

8.2 Amateur Radio support

业余广播支持

如果你想要让你的机器连接上业余的广播台，选 Y。

注意这个回答不会直接影响到内核：选 N 只是会让编译器跳过所有与业余广播相关的选项。

8.3 CAN bus subsystem support

CAN 总线子系统支持

控制区域网络 (CAN) 是一个慢速 (可以到 1MB 每秒) 的串行通讯协议，它是 Bosch 开发出来的，当初主要目的是用于汽车，现在在潜艇 (NMEA2000)、工业和医疗软件上广泛应用。

如果你需要 CAN 支持，那么在这选 Y，同时在下方的子选项中选上相应的控制器。

8.4 IrDA (infrared) subsystem support

红外子系统支持

选 Y，如果你需要内核支持 IrDA (TM) 协议 (红外通讯协议)。红外数据通讯是用于无线红外连接的标准，它支持大多数的笔记本和 PDA。要使用红外通讯协议，你还需要一些用户空间的实用程序，比如 irattach。如果你要和 PDA 交换比特数据，你需要安装一些 OBEX 程序，比如 OpenObex。

选 M 编译为模块，名为：irda。

8.5 Bluetooth subsystem support

蓝牙子系统支持

蓝牙是低花费、低能耗、短距离的无线技术。它设计的目的是用来取代电缆和其他的短距离技术，比如红外技术。蓝牙的作用范围一般在 10 米。

Linux 蓝牙子系统包括一些组件：

Bluetooth Core，蓝牙核心支持 (HCI 设备和连接管理、调度)

HCI Device drivers，HCI 设备驱动 (硬件接口)

SCO Module，SCO 模块 (SCO 声讯连接)

L2CAP Module，L2CAP 模块 (逻辑连接控制和适配器协议)

RFCOMM Module，RFCOMM 模块 (RFCOMM 协议)

BNEP Module，BNEP 模块 (蓝牙网络封包协议)

CMTP Module，CMTP 模块 (CAPI 信息传输协议)

HIDP Module , HIDP 模块 (人性化接口设备协议)

选 Y 编译时内核, 或者选 M 编译为模块, 名为: bluetooth。

要使用 Linux 蓝牙子系统, 你需要一些用户空间的实用程序如 hciconfig 和 hcid。这些程序和内核模块升级部件在 BlueZ 包中提供。

8.6 Phonet protocols family

网络电话协议家族

网络电话协议 (PhoNet) 是一个包定向的通讯协议, 它由 Nokia 开发, 用于它们的 modem。

对于 Maemo, 使用元数据连接 (如果可用) 的话是需要这个协议的。它还可以用于在 Linux 系统中控制 Nokia 电话, 虽然用 AT 命令可能会更容易。

选 M 编译为模块, 名为: phonet。

如果不清楚, 选 N。

8.7 Wireless

无线支持

8.7.1 Improved wireless configuration API

改进型无线配置接口

8.7.2 Wireless extensions

无线扩展

这个选项允许传统的无线扩展 (通过 ioctl 进行无线网络接口配置)。

无线扩展将会被 cfg80211 取代, 也将只被传统的无线扩展驱动器需要。这个选项不会影响到在 cfg80211 中的无线扩展的代码。

选 N (如果可以的话) 除非你知道你确实需要无线扩展来支持外部模块。

8.7.3 Generic IEEE 802.11 Network Stack(mac80211)

通用 IEEE 802.11 网络栈(mac80211)

这个选项允许硬件依赖 IEEE 802.11 网络栈。

8.7.4 Generic IEEE 802.11 Network Stack

通用 IEEE 802.11 网络栈(已经废弃不用)

这个标准已经被上面的 mac80211 所取代。

8.8 RF switch subsystem support

RF 转换子系统支持

选 Y，如果你想要控制你的 RF 转换器，它通常用在很多 WiFi 和蓝牙卡上。

选 M 编译为模块，名为：rfkill。

Chapter 9

Device Drivers 设备驱动

说明：

这一章的内容在所有内核配置菜单选项里是最多的，它涵盖了内核所支持的所有设备驱动。因此笔者也不能——列出和详细说明所有的菜单选项。

但是，这一章的内容又是极为关键的，如果急躁的读者乱配置的话，可能编译好的内核根本不能启动，或者启动后老是缺这个功能少那个功能。甚至会出现两个功能互相抵触的情况。

因此，要配置好这一章，你需要认真仔细地查看相关选项。本人认为比较重要的选项，都做了说明，不重要的，只是列举标题。但是这也并不一定就适合你的机器。

其中要重点说明的一个地方就是磁盘的配置。读者要明白的几点如下：

- 所有的磁盘都是块设备。因此你必定要选择块设备选项。
- IDE 磁盘是 ATA 磁盘的一种，称为 ATA-1。
- SATA 和 PATA 也是 ATA 的一种。
- SCSI 和 IDE 没有关系，但是和 SATA 和 PATA 有关系：SATA 和 PATA 有部分功能依赖 SCSI，因此不要认为你机器上没有 SCSI 盘，SCSI 菜单下的所有选项都可以不选。
- SCSI 还和 USB 储存盘有关系。你要是没有选择 SCSI 的相关选项，将无法使用 U 盘。
- 根目录所在的磁盘类型，不能编译为模块。
- 根目录所在的磁盘系统的主适配器，不能编译为模块。

还要说明一点的是显卡，很多人的显卡老是出问题，关键就是你的内核配置不对。

需要注意的就是 AGP 和 Direct Rendering 的关系。

AGP 是通用的驱动，没有它你的 AGP 显卡无法正常使用。Direct Rendering 是直接渲染技术，它依赖于 AGP，而 OPENGL 等 3D 引擎又依赖它。

在 AGP 和 Direct Rendering 子菜单里，你需要选择和你机器上对应的显卡。如果你的显卡非常新，找不到相应的驱动项目，你可以试试同一类别的驱动项目，或者是通用的驱动项目。要做好内核无法使用你的显卡的心理准备。

到你的显卡厂商那看看，他们有没有提供 Linux 的驱动，并了解安装方法。

读者需要认真看好每个选择，同时必须非常了解自己的机器硬件配置，并选择相应的设备驱动。如果搞错了内核启动不了，那么注意屏幕信息，看看到底在哪里出了问题。要是解决不了，上网求助。

9.1 Generic Driver Options

通用驱动选项

9.1.1 path to uevent helper

uevent 帮助程序路径

uevent 帮助程序的路径，每个 uevent 由内核叉产生。

9.1.2 Select only drivers that don't need compile-time external firmware

选择不需要外部固件编译时间的驱动

选 Y，如果你的驱动没有使用魔术般¹的固件。

如果不清楚，选 Y。

9.1.3 Prevent firmware from being built

阻止编译固件

选 Y 可以阻止编译固件。固件通常和驱动相连，只有在升级固件后，才需要重新编译。

如果不清楚，选 Y。

9.1.4 Userspace firmware loading support

用户空间固件装载支持

这个选项用于非内核的模块对固件的装载支持，它需要使用用户空间。内核本身的模块不一定提供相关支持。

¹指智能的、可升级的、奇特的

9.1.5 Driver Core verbose debug messages

驱动核心详细调试信息

选 Y，如果你想要驱动核心产生一大堆调试信息至系统日志。如果你在驱动核心上有问题，想要看看到底发生了什么，选 Y。

如果不清楚，选 N。

9.1.6 Managed device resources verbose debug messages

管理设备资源详细调试信息

这个选项开启内核参数 `devres.log`。如果设置为非零，`devres` 调试信息将打印出来。如果你在 `devres` 的使用上出现问题或者想要对一个管理设备调试资源管理，那么选 Y。`devres.log` 可以在系统文件节点中切换开启和关闭状态。

如果不清楚，选 N。

9.2 Connector - unified userspace <-> kernelspace linker

连接器 -- 用户空间<-> 内核空间统一连接

这是在网络连接套接口协议上运行的用户空间<-> 内核空间统一连接。

连接器可以编译为模块，名为 `cn.ko`。

9.2.1 Report process events to userspace

向用户空间报告进程事件

让连接器向用户空间报告进程事件。比如派生指令、执行指令、ID 变化和退出指令。

9.3 Memory Technology Devices (MTD) support

内存技术设备支持

内存技术设备是 `flash`、`RAM` 和类似的芯片，经常用于嵌入式设备中的连续文件系统。这个选项将提供对 MTD 驱动的通用支持，使之注册在内核之中，并且对潜在的用户列举出相关设备以便使用。它同样允许你对于特别的硬件和 MTD 设备用户选择个性化的驱动。如果不清楚，选 N。

- Debugging

调试

这为 MTD 子系统打开底层的调试。通常，你应该选 N。

- MTD concatenating support

MTD 并置支持

将并置的几个 MTD 设备整合成一个（虚拟的）设备。这允许你的 JFFS (2)文件系统生成多重的物理的闪存。如果不清楚，选 Y。

9.3.1 MTD partitioning support

MTD 分区支持

如果你有一个设备，它需要将它自己的闪存分成几个复合的“分区”，每个对用户来说好像都是独立的 MTD 设备，选 Y。如果不清楚，选 Y。

注意，对于 DiskOnChip 设备，你不需要这个选项。在 NFTL 设备上分区又是不同的情况：那是对块设备的“普通”的分区应用。

- RedBoot partition table parsing

RedBoot 分区表分析

RedBoot 是一个 ROM 监视器和启动装载器，它对于多重的在闪存设备中的“镜像”建立一个表，这个表存在于设备的可擦除块中，有点像分区表，对每个在闪存中的镜像提供偏移量、长度、名字。

如果你需要探测和分析这个表格的代码，并且对表格中的每个镜像声明 MTD 分区响应，在这选 Y

你将同样需要相关的分析功能，它不会自动生成。比如，SA1100 页驱动为此提供了一些选项。

- Include unallocated flash regions

包含未分配闪存区域

如果你需要声明每个未分配的闪存区域，作为 MTD 的“分区”，在这选 Y。

- Force read-only for RedBoot system images

对 RedBoot 系统镜像强制只读

如果你需要对‘RedBoot’、‘RedBoot Config’和‘FIS directory’镜像强制只读，选 Y。

9.4 Parallel port support

并行端口支持

如果你想要使用连接到你机器的并行端口²上的设备，比如打印机、ZIP 磁盘、PLIP 连接器³等等，在这选 Y。

²通常是 25 个孔的接口

³并行网络协议线路，主要用于建立一个小型的网络，它通过两台机器的并行端口互联。

你可以让一个端口共享给几个设备使用，同时把所有的相关驱动编译进内核是安全的。选 M 编译为模块，名为：parport。

如果不清楚，选 Y。

9.5 Plug and Play support

即插即用支持

即插即用是周边设备的一种标准。它允许这些周边设备能够在接入时被软件配置，通过中断或者其他参数。你的设备上不需要设置任何的跳线，因为相应的值是由 BIOS、操作系统或者相关软件工具提供的。

如果你想要你的即插即用设备工作，那么在这选 Y。你同样也要在下面所有的协议上选 Y。另外，你也可以在这选 N，然后通过用户空间的工具来配置你的即插即用设备，比如 isapnptools 工具包。

如果不清楚，选 Y。

9.5.1 PNP debugging messages

PNP 调试信息

如果你想要 PNP 层输出调试信息，那么选 Y。信息将在启动时通过内核参数 pnp.debug 开启。

这个选项允许你储存一些字节的空间，如果你不想要信息编译进入内核。

如果不清楚，选 Y。

9.6 Block devices

块设备

在这选 Y 可以看到各种块设备的驱动。这个选项本身不增加任何内核代码。

如果你选 N，所有的子选项将会禁用；不要这样做，除非你真正知道你在干什么。

9.6.1 Normal floppy disk support

普通软盘支持

如果你想要使用软盘，在这选 Y。

选 M 编译为模块，名为 floppy。

9.6.2 Compaq SMART2 support

Compaq SMART2 支持

这是针对 Compaq SMART2 的阵列控制器驱动支持。

9.6.3 Compaq Smart Array 5xxx support

Compaq Smart 5xxx 阵列支持

这是针对 Compaq SMART 5xxx 的阵列控制器驱动支持。

9.6.4 Mylex DAC960/DAC1100 PCI RAID Controller support

Mylex DAC960/DAC1100 PCI RAID 控制器支持

这个驱动支持 Mylex DAC960, AcceleRAID 和 eXtremeRAID PCI RAID 控制器。

选 M 编译为模块，名为：DAC960。

9.6.5 Loopback device support

回送设备支持

在这选 Y 将允许你像使用一个块设备一样使用一个常规文件。你可以在这个块设备上建立一个文件系统，并且可以像挂载硬盘、光盘、软盘那样去挂载它。这种回送设备是特殊的文件块设备，主号为 7，传统上被称为 /dev/loop0，/dev/loop1 等等。

这个功能是非常有用的。比如你想在刻录一个光盘前，要检查硬盘上的 ISO 9660 文件，或者你想要使用一个虚拟的软盘文件，但是又不想把它写在软盘上。此外，一些 Linux 的发行版把它们的启动文件放在了 DOS FAT 文件中，并通过使用这个回送设备驱动的方式来访问，这样可以避免使用 Linux 的专用分区，以提高兼容性。

回送设备驱动也可以用于在硬盘分区、软盘或者常规文件中“隐藏”文件系统。它要么通过完整加密程序（加密所有相应数据），要么是速记式加密（隐藏数据的低字节，通常用于声音文件）。这些做法使得在远程文件服务器上的文件都非常安全。

有很多种方法来加密磁盘，一些需要内核的补丁。vanilla 内核提供了密码至回送设备的选项和一个设备映射目标（它支持所有的文件系统，是一个上层系统）。如果你需要使用密码至回送的功能，那么在 LOOP 和 CRYPTOLOOP 都要选 Y，并确保你有个比较新的（2.12 或者更新）Linux 工具版本。另外，注意密码至回送功能对于储存日志文件系统很不安全。

注意这个回送设备和网络上使用的自回环设备没有任何关系，那是用于与自己连接的方式。

选 M 编译为模块，名为 loop。

大多数用户可以在这选 N。

- Cryptoloop Support

密码至回送支持

选 Y，如果你需要这项由加密接口提供的功能来用于回送的传输。这可以用于硬盘加密。

警告：这个设备对于日志文件系统非常不安全，比如 ext3 和 Reiserfs。如果你使用这些文件系统，那么请你使用 Device Mapper crypto 模块来代替，它可以配置得和密码回送设备兼容。

9.6.6 Network block device support

网络块设备支持

选 Y 将允许你的计算机成为网络块设备的客户机，比如它将可以使用由服务器提供的块设备（在上面挂载文件系统，等等）。客户机和服务器通过 TCP/IP 网络进行通信，但是对于客户机来说这是隐藏的：它看起来像一个常规的本地文件，通过 `/dev/nd0` 这样的块设备方式来进行访问。

网络块设备同样允许你在用户平台运行一个块设备，这使得服务器和客户机看起来在物理上就是一个同样的计算机，它们使用回环网络设备进行通信。

注意这里和网络文件系统 NFS 或者 Coda 没有任何关系。你可以在这选 N，即使你想要使用 NFS 或者 Coda。

选 M 编译为模块，名为 `nbd`。

如果不清楚，选 N。

9.6.7 Promise SATA SX8 support

Promise SATA SX8 支持

选 Y 或者 M 支持 Promise SATA SX8 控制器。

设备为 `/dev/sx8/$N` 和 `/dev/sx8/$Np$M`。

9.6.8 Low Performance USB Block driver

低端 USB 块驱动行为

这个驱动支持特定的 USB 连接储存设备，比如 flash key。

如果你开启这个选项，那么建议你开启 `USB_LIBUSUAL` 以避免和 `usb-storage` 选项的冲突。

如果不清楚，选 N。

9.6.9 RAM block device support

RAM 块设备支持

选 Y 将允许你使用你 RAM 内存的一部分作为一个块设备，这样你就可以在上面建立文件系统，进行读写操作和其他相关的操作，就好比你在使用一个常规的块设备（如硬盘）一样。它经常用于在 Linux 初始化安装的时候装载和储存软盘上的一个小型启动文件系统。

注意现在的内核命令选项“ramdisk=XX”已经不用了。

选 M 编译为模块，名为：rd。

大多数普通的用户不会用到 RAM 磁盘的功能，因此可以在这选 N。

- Default number of RAM disks

默认 RAM 磁盘数量

默认值是 16。如果你要改变它，你要确定你在做什么。如果你启动的文件系统需要在内存中解压缩，那么你需要至少一个 RAM 磁盘（比如 root on cramfs）。

- Default RAM disk size (kbytes)

默认 RAM 磁盘尺寸

默认值为 4096KB。如果要修改它，你要明白你在做什么。

- Support XIP filesystems on RAM block device

在 RAM 块设备支持 XIP 文件系统

在 RAM 块设备的顶层支持 XIP 文件系统（比如有 XIP 支持的 ext2）。这将会稍微地增大内核，也会阻止 RAM 块设备在高层内存上定位和储存（在大内存系统上会存在问题）。

9.6.10 Packet writing on CD/DVD media

CD/DVD 媒体包写入

如果你有一个支持包写入的 CDROM/DVD，那么在这选 Y 以提供相应支持。它也可以工作在任何 MMC/Mt Fuji 兼容的 ATAPI 或者 SCSI 驱动器上，这是比较新的 DVD/CD 刻录机。

目前只能写入 CD-RW, DVD-RW, DVD+RW 和 DVD-RAM 盘。

DVD-RW 盘必须要在限制覆盖模式。

选 M 编译为模块，名为 pktcdvd。

9.6.11 ATA over Ethernet support

以太网 ATA 支持

这个驱动支持 ATA 以太网块设备，比如 Coraid EtherDrive Storage Blade。

9.6.12 Very old hard disk (MFM/RLL/IDE) driver

非常老的硬盘驱动

这里是非常老的硬盘驱动，它们比起新的硬盘驱动缺少增强型的功能。

对于古董级别的 MFM/RLL/ESDI 驱动，这个选项是必需的。

如果不清楚，选 N。

9.7 Misc devices

杂类设备

在这选 Y 可以看到各种杂类设备目录。这个选项本身不增加任何内核代码。
如果你选 N, 所有的子选项将会禁用。

9.7.1 Sensable PHANToM

Sensible PHANToM

9.7.2 EEPROM 93CX6 support

EEPROM 93CX6 支持

9.7.3 SGI IOC4 Base IO support

SGI IOC4 Base IO 支持

9.7.4 Fujitsu Laptop Extras

Fujitsu 笔记本扩展

9.7.5 MSI Laptop Extras

MSI 笔记本扩展

9.7.6 Sony Laptop Extras

Sony 笔记本扩展

9.7.7 ThinkPad ACPI Laptop Extras

ThinkPad ACPI 笔记本扩展

9.7.8 Thermal Management driver for Intel menlow platform

Intel menlow 平台温控管理驱动

9.7.9 Enclosure Services

Enclosure 服务

9.7.10 Channel interface driver for HP iLO/iLO2 processor

HP iLO/iLO2 处理器隧道接口驱动

9.8 ATA/ATAPI/MFM/RLL support

ATA/ATAPI/MFM/RLL 支持

如果你在这选 Y，你的内核将可以管理大量的储存设备单元，比如 ATA/(E)-IDE 和 ATAPI 单元。最广泛使用的就是 IDE 硬盘驱动和 ATAPI CD - ROM 驱动。

如果你的系统是纯种的 SCSI，完全不使用上述的任何接口，那么你可以在这选 N。

集成式磁盘电子接口（IDE，也称为 ATA - 1）是一个连接标准，它用于通用储存单元，比如硬盘上。它在 1984 年被 Western Digital 和 Compaq Computer 设计出来，同时命名为 ST506。相当多的磁盘使用的是 IDE 接口。

AT Attachment(ATA)是 IDE 的超集。IDE 也被称为 ATA - 1。

Fast-IDE（快速 IDE）是 ATA - 2（也被称为 Fast ATA），Enhanced IDE（强化 IDE）(EIDE)是 ATA - 3。它提供对更大的磁盘的支持（根据 LBA 的标准，大于 8.4GB），更多的磁盘（以前是 2 个，现在支持到 4 个）和对其他通用储存单元如磁带机和光盘机的支持。UDMA/33（也称为 UltraDMA/33）是 ATA - 4，它提供更快的（更友好的 CPU）传输模式。而之前的 ATA/IDE 标准使用的 PIO（可编程处理器输入输出）相比起 DMA 来就慢了很多。

ATA 包接口（ATAPI）是一个协议，它用于 EIDE 磁带机和光盘机，有点类似于 SCSI 的协议。

SMART IDE（自监视、分析和报告技术）通过检测硬件的失败条件（温度、访问时间或者类似的）来防止数据损失和磁盘损坏。从 1995 年起制作的磁盘很多都遵守这个标准。内核本身不管这些，但是有很多用户程序比如 smart 可以从磁盘驱动中队列化 SMART 参数。

选 M 编译为模块，名为 ide。

如果不清楚，选 Y。

9.8.1 Support for SATA

SATA 支持⁴

有两个驱动支持串行 ATA 控制器。

⁴已经废弃不用，与 libata SATA 驱动冲突

主驱动，“libata”，使用 SCSI 子系统，它支持大多数现代的 SATA 控制器。要使用它，你可以看看选项：“Serial ATA (prod) and Parallel ATA drivers”。

IDE 驱动（你正在配置的这个）支持少量的第一批 SATA 控制器。

为了使这两个系统不冲突，这里提供对 SATA 的 IDE 驱动支持。通常这个选项是关闭的，最好的方式是这个系统主要提供对 PATA 的支持，而 libata 提供对对 SATA 的支持。

如果不清楚，选 N。

9.8.2 generic ATA/ATAPI disk support

通用 ATA/ATAPI 磁盘支持

支持 ATA/ATAPI 磁盘（包括 ATAPI 软盘）。

选 M 编译为模块，名为：ide-gd_mod。

如果不清楚，选 Y。

- ATA disk support

ATA 磁盘支持

这里支持 ATA 硬盘。

如果不清楚，选 Y。

- ATAPI floppy support

ATAPI 软盘支持

这里支持 ATAPI 软盘驱动器（比如 Iomega ZIP 或者 MKE LS-120）。

如果不清楚，选 N。

9.8.3 PCMCIA IDE support

PCMCIA IDE 支持

9.8.4 Cardbus IDE support (Delkin/ASKA/Workbit)

Cardbus IDE 支持

9.8.5 Include IDE/ATA-2 DISK support

IDE/ATA-2 支持

这里提供对 MFM/RLL/IDE 硬盘的加强支持。如果你有上述的一个磁盘，并且也没有特殊的情况要在下面的“老式磁盘驱动”选项上选 Y，那么你可以在这选 Y。如果你的系统是纯 SCSI 系统，那么你可以在这选 N。

选 M 编译为模块，名为 ide-disk。

不要把它编译为模块，如果你的启动文件系统（包含/目录的系统）定位于一个 IDE 设备上。

如果不清楚，选 Y。

9.8.6 Use multiple sector mode for Programmed Input/Output by default

默认使用多重扇区进行程序化输入/输出

这个设置和大多数使用直接内存访问，不提供多重扇区模式的 IDE 磁盘无关。多重扇区模式是一个现代 IDE 硬盘的特性，它允许多重扇区传输的程序化输入/输出中断，而不是像以前那样的每个扇区一个中断。当这个特性开始时，它可以减少操作系统用于磁盘程序化输入/输出的开销。在其他的一些系统上，它也可以提高数据的传输速度。不过一些设备，在这个特性下有可能反而运行得更慢。一些驱动器声称支持多扇区模式，但是在某些情况下会丢失数据。在极少的情况下，这些失误可能会导致文件系统的大面积崩溃。

如果你得到下面的错误提示，尝试在这选 Y：

```
hda: set_multmode: status=0x51 { DriveReady SeekComplete Error }
```

```
hda: set_multmode: error=0x04 { DriveStatusError }
```

如果不清楚，选 N。

9.8.7 Include IDE/ATAPI CDROM support

IDE/ATAPI CDROM 支持

如果你有一个使用 ATAPI 协议的 CDROM，选 Y。ATAPI 是一个用于 IDE CDROM 和磁带机的更新的协议，类似 SCSI 协议。大多数新的 CDROM 驱动器使用 ATAPI，包括有 NEC-260, Mitsumi FX400, Sony 55E，和所有的非 SCSI 双倍（2X）或者更快速度的驱动器。

如果你选 Y，CDROM 驱动器将会在启动时和其他的 IDE 设备一样被识别，可能被命名为“hdb”或者“hdc”，或者其他类似的名称（使用 dmesg 来检查启动信息）。如果这是你唯一的光驱，那么你可以在其他的 CDROM 选项上选 N，但是要在选项“ISO 9660 CD-ROM file system support”中选 Y 或者 M。

选 M 编译为模块，名为：ide-cd。

9.8.8 Include IDE/ATAPI TAPE support

IDE/ATAPI 磁带机支持

选 Y，如果你有一个 IDE/ATAPI 磁带机。

9.8.9 Include IDE/ATAPI FLOPPY support

IDE/ATAPI 软驱支持

选 Y，如果你有一个 IDE/ATAPI 软驱。

9.8.10 SCSI emulation support

SCSI 仿真支持

9.8.11 IDE ACPI support

IDE ACPI 支持

为通用的 IDE 设备执行 ACPI。在现在的设备中，需要 ACPI 支持来应付 ACPI S3 状态。

9.8.12 IDE Taskfile Access

IDE 工作文件访问

9.8.13 legacy /proc/ide/ support

传统的/proc/ide/支持

这个选项允许对/proc/ide/目录下的多种文件的支持。在 Linux2.6 这种做法已经被 sysfs 取代，但是很多老的程序依赖旧的做法。

如果不清楚，选 Y。

9.8.14 generic/default IDE chipset support

通用/默认 IDE 芯片支持

如果不清楚，选 N。

9.8.15 Platform driver for IDE interfaces

IDE 接口平台驱动

这是 IDE 平台驱动，主要用于内存映射 IDE 设备，比如在真实 IDE 模式下运行的 Compact Flash。

如果不清楚，选 N。

9.8.16 CMD640 chipset bugfix/support

CMD640 芯片错误修正/支持

CMD 技术 CMD640 IDE 芯片用于很多通用 486 和 Pentium 主板上，通常和“Neptune”或者“SIS”芯片绑定。不幸的是，它有一大堆相当糟糕的设计缺陷，可能会在很多常规情况下导致数据错误。在这选 Y，将尝试自动探测和校正 Linux 下的问题。这个选项也允许在某些基于 CMD640 的系统对第二 IDE 端口的访问。

这个驱动将在基于 PCI 的系统下⁵自动运行。但是，如果你的系统使用的是 VESA 本地总线（VLB）而不是 PCI，你必须提供一个内核启动参数以开启 CMD640 错误修复功能的支持：“cmd640.probe_vlb”。尝试“man bootparam”或者查看你的启动器手册，看看应该如何给内核增加参数。

CMD640 芯片同样用于一些 Acculogic 附加卡上，和一些人使用的“CSA-6400E PCI to IDE controller”上。

9.8.17 PNP EIDE support

即插即用 EIDE 支持

9.8.18 Probe IDE PCI devices in the PCI bus order

按照 PCI 总线的顺序探测 IDE PCI 设备

9.8.19 Generic PCI IDE Chipset Support

通用 PCI IDE 芯片支持

9.8.20 RZ1000 chipset bugfix/support

RZ1000 芯片错误修复/支持

9.8.21 AEC62XX chipset support

AEC62XX 芯片支持

9.8.22 ALI M15x3 chipset support

ALI M15x3 芯片支持

9.8.23 AMD and nVidia IDE support

AMD 和 nVidia IDE 支持

9.8.24 ATI IXP chipset IDE support

ATI IXP 芯片 IDE 支持

⁵大多数新系统有 PCI 插槽

9.8.25 CMD64{3|6|8|9} chipset support

CMD64{3|6|8|9}芯片支持

9.8.26 Compaq Triflex IDE support

Compaq Triflex IDE 支持

9.8.27 CY82C693 chipset support

CY82C693 芯片支持

9.8.28 Cyrix/National Semiconductor CS5530 MediaGX chipset support

Cyrix/National 半导体 CS5530 MediaGX 芯片支持

9.8.29 AMD CS5535 chipset support

AMD CS5535 芯片支持

9.8.30 HPT34X chipset support

HPT34X 芯片支持

9.8.31 HPT36X/37X chipset support

HPT36X/37X 芯片支持

9.8.32 JMicron JMB36x support

JMicron JMB36x 支持

9.8.33 National SCx200 chipset support

National SCx200 芯片支持

9.8.34 Intel PIIX/ICH chipsets support

Intel PIIX/ICH 芯片支持

这个驱动增加对 Intel PIIX 和 ICH 芯片的详细支持。这允许内核更改 PIO , DMA 和 UDMA 速度, 并配置芯片以达到最优化速度。

9.8.35 IT8213 IDE support

IT8213 IDE 支持

9.8.36 IT821X IDE support

IT821X IDE 支持

9.8.37 NS87415 chipset support

NS87415 芯片支持

9.8.38 PROMISE PDC202{46|62|65|67} support

PROMISE PDC202{46|62|65|67}支持

9.8.39 PROMISE PDC202{68|69|70|71|75|76|77} support

PROMISE PDC202{68|69|70|71|75|76|77}支持

9.8.40 ServerWorks OSB4/CSB5/CSB6 chipsets support

ServerWorks OSB4/CSB5/CSB6 芯片支持

9.8.41 Silicon Image chipset support

Silicon Image 芯片支持

9.8.42 SiS5513 chipset support

SiS5513 芯片支持

9.8.43 SLC90E66 chipset support

SLC90E66 芯片支持

9.8.44 Tekram TRM290 chipset support

Tekram TRM290 芯片支持

9.8.45 VIA82CXXX chipset support

VIA82CXXX 芯片支持

9.8.46 Toshiba TC86C001 support

Toshiba TC86C001 支持

9.9 SCSI device support

SCSI 设备支持

9.9.1 RAID Transport Class

RAID 传输类

9.9.2 SCSI device support

SCSI 设备支持

如果你想要使用 SCSI 硬盘、磁带机、光驱或者其他 SCSI 设备，需要在这选 Y，并且要选上相应的 SCSI 主适配器（在你电脑里使用 SCSI 协议的卡，也称为 SCSI 控制器）。

如果你要使用一些 SCSI 协议的设备，比如 USB 储存设备、火线储存设备和 IDE-SCSI 仿真驱动，那么也要在这选 Y。

选 M 编译为模块，名为：scsi_mod。

如果你的启动文件系统（包含/目录）在 SCSI 设备里，那么就不能把它编译为模块。

9.9.3 legacy /proc/scsi/ support

传统/proc/scsi/支持

9.9.4 SCSI disk support

SCSI 磁盘支持

如果你要使用 SCSI 硬盘、Fibre 隧道磁盘、SATA 磁盘或者 PATA 硬盘、USB 储存设备、IOMEGA ZIP 设备等等，你需要在这选 Y。

选 M 编译为模块，名为：sd_mod。

如果你的启动文件系统（包含/目录）在 SCSI 设备里，那么就不能把它编译为模块，也不要吧 SCSI 主适配器驱动编译为模块。

9.9.5 SCSI tape support

SCSI 磁带支持

9.9.6 SCSI OnStream SC-x0 tape support

SCSI OnStream SC-x0 磁带支持

9.9.7 SCSI CDROM support

SCSI CDROM 支持

9.9.8 Enable vendor-specific extensions (for SCSI CDROM)

允许销售商的特殊扩展 (对 SCSI CDROM)

9.9.9 SCSI generic support

SCSI 通用支持

9.9.10 SCSI media changer support

SCSI 媒质变换器支持

9.9.11 Probe all LUNs on each SCSI device

在每个 SCSI 设备探测所有 LUN

9.9.12 Verbose SCSI error reporting (kernel size +=12K)

详细 SCSI 错误报告 (内核尺寸增大 12K)

9.9.13 SCSI logging facility

SCSI 日志工具

9.9.14 Asynchronous SCSI scanning

异步 SCSI 扫描

9.9.15 SCSI Transports

SCSI 传输

9.9.16 SCSI low-level drivers

SCSI 底层驱动

9.10 Serial ATA and Parallel ATA drivers

串行 ATA 和并行 ATA 驱动

如果你想要使用一个 ATA 硬盘、ATA 磁带机、ATA 光盘机或者其他任何的 ATA 设备，在这选 Y，同时确保你知道你的 ATA 主适配器的名字（在你的电脑中的一张卡，它提供了 ATA 协议，也称为 ATA 控制器），因为你要提供它的信息。

注意：ATA 提供了基本的 SCSI 支持，但是，SCSI 下的相关选项也是必需的，比如 SCSI 磁盘支持、SCSI 磁带支持或者 SCSI 光盘支持等，这取决于你的系统硬件配置。

9.10.1 ATA ACPI Support

ATA ACPI 支持

9.10.2 SATA Port Multiplier support

SATA 端口乘法器支持

9.10.3 AHCI SATA support

AHCI SATA 支持

9.10.4 Silicon Image 3124/3132 SATA support

Silicon Image 3124/3132 SATA 支持

9.10.5 ATA SFF support

ATA SFF 支持

这个选项增加对 ATA 控制器的 SFF 或者其他类似接口的兼容支持。

SFF 是传统的 IDE 接口，从发明开始就被拥护。几乎所有的 PATA 控制器都有一个 SFF 接口。很多 SATA 控制器有一个 SFF 接口，当配置为与传统模式兼容时将起作用。

对于使用更先进控制器的用户，比如 AHCI，Silicon Image 3124，或者 Marvell 6440，你可以选择禁止这项没有必要的 SFF 支持。

如果不清楚，选 Y。

- ServerWorks Frodo / Apple K2 SATA support
ServerWorks Frodo / Apple K2 SATA 支持
- Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA support
Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA 支持
- NVIDIA SATA support
- Pacific Digital ADMA support
- Pacific Digital SATA QStor support
- Promise SATA TX2/TX4 support
- Silicon Image SATA support
- SiS 964/965/966/180 SATA support
- ULi Electronics SATA support
- VIA SATA support
- VITESSE VSC-7174 / INTEL 31244 SATA support
- Initio 162x SATA support
- ACPI firmware driver for PATA
- AMD/NVidia PATA support
- ARTOP 6210/6260 PATA support
- ATI PATA support
- CMD64x PATA support
- CS5510/5520 PATA support
- EFAR SLC90E66 support
- Generic ATA support
- HPT 366/368 PATA support
- HPT 343/363 PATA support
- IT8211/2 PATA support

- JMicron PATA support
- Compaq Triflex PATA support
- Marvell PATA support via legacy mode
- Intel PATA MPIIX support
- Intel PATA old PIIX support
- NETCELL Revolution RAID support
- PC Tech RZ1000 PATA support
- SERVERWORKS OSB4/CSB5/CSB6/HT1000 PATA support
- Promise PATA 2027x support
- CMD / Silicon Image 680 PATA support
- VIA PATA support
- Winbond SL82C105 PATA support
- Intel SCH PATA support

9.11 Multiple devices driver support(RAID and LVM)

多重设备驱动支持 (RAID 和 LVM)

支持用单逻辑设备管理多重物理设备的方式， RAID 和 LVM 都需要它。

9.11.1 RAID support

RAID 支持

9.11.2 Device mapper support

设备映射支持

- Device mapper debugging support 设备映射调试支持
- Crypt target support 加密目标支持
- Snapshot target 快照目标
- Mirror target 镜像目标
- Zero target 零目标
- Multipath target 多路径目标

9.12 Fusion MPT device support

Fusion⁶ MPT 设备支持

在这选 Y 以得到 Fusion 信息处理技术 (MPT) 驱动的相关选项。这个选项本身不增加内核代码。

如果你选 N , 所有的子选项将被禁用。

9.12.1 Fusion MPT ScsiHost drivers for SPI

SPI 的 Fusion MPT ScsiHost 驱动

9.12.2 Fusion MPT ScsiHost drivers for FC

FC 的 Fusion MPT ScsiHost 驱动

9.12.3 Fusion MPT ScsiHost drivers for SAS

SAS 的 Fusion MPT ScsiHost 驱动

9.12.4 Maximum number of scatter gather entries (16 - 128)

分散收集序列最大值

9.12.5 Fusion MPT misc device (ioctl) driver

Fusion MPT misc device (ioctl)驱动

9.12.6 Fusion MPT logging facility

Fusion MPT 日志工具

9.13 IEEE 1394 (FireWire) support

IEEE 1394 (火线) 支持

9.13.1 Stable FireWire stack

稳定火线栈

⁶AMD 的一种处理品, 称为: 融合。

9.13.2 OHCI-1394 controllers

OHCI-1394 控制器

9.13.3 Storage devices (SBP-2 protocol)

储存设备 (SBP-2 协议)

9.13.4 raw1394 userspace interface

raw1394 用户空间接口

9.13.5 video1394 userspace interface

video1394 用户空间接口

9.13.6 dv1394 userspace interface

dv1394 用户空间接口

9.13.7 Excessive debugging output

过度调试输出

9.14 I2O device support

I2O 设备支持

智能输入输出 (I2O) 体系允许硬件驱动分成两个部分：一个是操作系统指定模块，称为 OSM，另一个是硬件指定模块，称为 HDM。OSM 可以与 HDM 的所有队列对话，同时 HDM 的队列并不是 OS 的依赖。这样允许同样的 HDM 驱动使用于不同的操作系统中，如果相关的 OSM 已经安装了的话。为保证这些可以工作，你需要有一个 I2O 接口适配器卡。这张卡包含一个特别的 I/O 处理器 (IOP)，因此可以在 CPU 不需要应付 I/O 的情况下仍然保持高速度。

如果你在这选 Y，你将会得到一个菜单，供你选择接口适配器的驱动和 OSM 的相关信息。

选 M 编译为模块，名为：i2o_core。

如果不清楚，选 N。

9.15 Macintosh device drivers

Mac 设备驱动

选 Y 可以看到 Mac 设备的相关驱动。这个选项本身不增加任何内核代码。

如果你选 N，那么所有的子选项将被禁止。

9.15.1 Support for mouse button 2+3 emulation

2 + 3 鼠标按键仿真支持

9.16 Network device support

网络设备支持

你可以在这选 N，如果你不打算让你的机器连接到其他的机器。

如果你有一个网卡，并打算在 Linux 下使用它的话，你必须要在这选 Y。如果你打算运行 SLIP 或者 PPP 电话或 modem 拨号的话，你也需要在这选 Y。使用 PLIP 通过两台机器的并口连接的话也需要这个选项，因为 AX.25/KISS 是通过业余电台连接来发送 Internet 堵塞的。

如果不清楚，选 Y。

9.16.1 Netdevice multiple hardware queue support

网络设备多重硬件队列支持

9.16.2 Dummy net driver support

伪网络驱动器支持

9.16.3 Bonding driver support

捆绑驱动器支持

9.16.4 EQL (serial line load balancing) support

EQL (串行线路加载平衡) 支持

9.16.5 Universal TUN/TAP device driver support

通用 TUN/TAP 设备驱动支持

9.16.6 Virtual ethernet pair device

虚拟以太网对设备

9.16.7 General Instruments Surfboard 1000

通用网络冲浪板 1000 方法

9.16.8 ARCnet support

ARCnet 支持

9.16.9 PHY Device support and infrastructure

PHY 设备支持和基础结构

9.16.10 Ethernet (10 or 100Mbit)

以太网 (10 或 100MB)

以太网 (也称为 IEEE 802.3 或者 ISO 8802-2) 是大学和公司的局域网 (LAN) 最通用的方式。

通用的以太网种类有：

- 10BASE-2 or Thinnet
- 10BASE-T or twisted pair
- 10BASE-F
- 100BASE-TX
- 100BASE-T4
- 100BASE-FX
- Gigabit Ethernet

如果你的 Linux 机器将要连接到以太网，同时你有一个以太网卡 (NIC) ，那么在这选 Y 。你还需要在下面的子菜单中选择相应的网卡类型。

注意这里的选择不会直接影响到内核：选 N 只是让配置器跳过所有的以太网卡选择。如果不清楚，选 N 。

9.16.11 Ethernet (1000 Mbit)

以太网 (1000MB)

9.16.12 Ethernet (10000 Mbit)

以太网 (10000MB)

9.16.13 Token Ring driver support

令牌驱动支持

9.16.14 Wireless LAN

无线 LAN

9.16.15 USB Network Adapters

USB 网络适配器

9.16.16 Wan interfaces support

WAN 接口支持

9.16.17 FDDI driver support

FDDI 驱动支持

9.16.18 PPP (point-to-point protocol) support

PPP (点对点协议) 支持

PPP (点对点协议) 是一个更新更好的 SLIP 协议。它实现与 SLIP 同样的目的：通过电话线（或者其他串行线）来传送网络信息。请找你的网络供应商，否则你无法使用这项功能。大多数的网络供应商提供的是 PPP 而不是 SLIP。

要使用 PPP，你还需要一个额外的软件，称为 `pppd`，并确保你的软件版本能够与最新的 PPP 协议对应。PPP 选项会使内核增大 16KB。

实际上有两种 PPP 版本：一种是传统的 PPP，通过异步线路传输，比如常规的电话线；另一种是同步 PPP，它通过数码 ISDN 线路传输。如果你想要通过电话线路或者其他的异步串行线路使用 PPP，你需要在这选 Y 或者 M，并在下面的选项“PPP support for async serial ports”选 Y。如果你要使用同步线路的 PPP，那么在下面的选项“Support synchronous PPP”选 Y。

如果你在上面的选项“Version information on all symbols”选了 Y，那么你将不能够把 PPP 驱动编译进内核，只能编译为模块。模块名为：`ppp-generic`。

9.16.19 SLIP (serial line) support

SLIP (串行线) 支持

9.16.20 Fibre Channel driver support

Fibre 隧道驱动支持

9.17 ISDN support

ISDN 支持

ISDN (综合数码网络服务, 在法国称为 RNIS) 是一种特殊的数码电话服务类型。它用于将你的电脑连接到你的 Internet 服务供应商 (用 SLIP 或者 PPP)。主要的优势在于, 连接的速度要快于通常的 modem/电话连接, 而且在下载东西的时候, 仍然可以进行通话。只有在你的电脑上有一张 ISDN 卡的时候, 它才会工作, 而且需要你和你的网络供应商从电信部门购买专用的 ISDN 线路。

选 Y, 如果你想要你的内核支持 ISDN。

9.18 Telephony support

电话通讯支持

选 Y, 如果你有一张电话卡, 它可以让你使用通常的电话在网络上通过声音 - IP 程序进行通话。

注意: 这里和 modem 没有任何关系。如果你使用 modem 的话, 你没有任何必要在这选 Y。

选 M 编译为模块, 名为: phonedev。

9.19 Input device support

输入设备支持

9.19.1 Generic input layer

通用输入层

在这选 Y, 如果你使用任何的输入设备 (鼠标、键盘、按钮、手柄、方向盘等等) 连接到你的系统, 并且想要它们能够被程序使用。这里包含了标准的 PS/2 键盘和鼠标。

如果你有一个微标题系统 (没有显示器、没有键盘), 那么你可以选 N。

如果不清楚, 选 Y。

选 M 编译为模块, 名为: input。

9.19.2 Support for memoryless force-feedback devices

支持小内存力反馈设备

9.19.3 Polled input device skeleton

轮询输入设备框架

9.19.4 Mouse interface

鼠标接口

在这选 Y ，如果你想要你的鼠标被仿真为一个字符设备，这通常在 13:32+ - /dev/input/mouseX 和 13:63 - /dev/input/mice 中。通过这种方式，任何用户空间的程序（包括 SVGAlib ， GPM 和 X ）将可以使用你的鼠标。

如果不清楚，选 Y 。

选 M 编译为模块，名为： mousedev 。

- Provide legacy /dev/psaux device 提供传统的/dev/psaux 设备

在这选 Y ，你的鼠标同样会仿真为另一个字符设备： 10:1 - /dev/psaux 。通过这种方式反馈的数据和同/dev/input/mice 反馈的数据是一样的。

如果不清楚，选 Y 。

- (1024) Horizontal screen resolution 水平屏幕分辨率

如果你在使用数字转换器，或者一个图形化的按钮，并且想要像使用鼠标一样使用它们，那么鼠标驱动器需要知道你所使用的 X window 屏幕策略以正确地按比例转换数据。如果你没有使用数字转换器，这个值将被忽略。

- (768) Vertical screen resolution 垂直屏幕分辨率

9.19.5 Joystick interface

手柄接口

9.19.6 Event interface

事件接口

如果你想要你的输入设备事件能够通过字符设备 13:64+ - /dev/input/eventX 访问（这是通用的方式），那么选 Y 。

选 M 编译为模块，名为： evdev 。

9.19.7 Event debugging

事件调试

9.19.8 Keyboards

键盘

在这选 Y，所支持的键盘列表将会显示出来。这个选项本身不增加任何内核代码。

如果不清楚，选 Y。

- Sun Type 4 and Type 5 keyboard
- DECstation/VAXstation LK201/LK401 keyboard
- XT keyboard
- Newton keyboard
- Stowaway keyboard

9.19.9 Mice

鼠标

选 Y，将会看到所支持的鼠标列表。这个选项本身不增加内核代码。

如果不清楚，选 Y。

- PS/2 mouse

PS/2 鼠标

选 Y，如果你使用的是 PS/2 鼠标。这里包含了标准的 2 键和 3 键 PS/2 鼠标，同时也包含了滚轮和额外按钮的支持，兼容 Microsoft, Logitech, Genius 等等。

如果不清楚，选 Y。

- Serial mouse
- Apple USB Touchpad support
- DEC VSXXX-AA/GA mouse and VSXXX-AB tablet

9.19.10 Joysticks/Gamepads

手柄/游戏板

9.19.11 Tablets

按钮⁷

⁷Tablet 是 Tablet PC 上的一种快捷按钮，用于执行常用任务。

9.19.12 Touchscreens

触摸屏

9.19.13 Miscellaneous devices

多种设备

9.19.14 Hardware I/O ports

硬件 I/O 端口

- Serial I/O support

串行 I/O 支持

在这选 Y，如果你的输入设备使用了串行的 I/O 来连接到系统。这其中包含标准的 AT 键盘和 PS/2 鼠标，同时也包含了串行鼠标、Sun 键盘、一些手柄和 6dof 设备等等。

如果不清楚，选 Y。

选 M 编译为模块，名为：serio。

- Serial port line discipline
- ct82c710 Aux port controller
- PCI PS/2 keyboard and PS/2 mouse controller
- Raw access to serio ports

- Gameport support

9.20 Character devices

字符设备

9.20.1 Virtual terminal

虚拟终端

如果你在这选 Y，你将可以得到终端设备的支持，它具有显示和键盘输入设备。这些叫做“虚拟”因为你可以在一个终端设备上运行几个终端，这同样也叫做虚拟控制台。这相当有用，比如你可以在一个终端收集系统信息，另一个作为用户文本输入界面，再一个做为 X 会话，这些都是并行开展的。在每个终端之间切换通常用组合键进行，比如 Alt+Ctrl+功能键。

命令：setterm 可以用于改变一个虚拟终端的属性（比如颜色或者声音）。命令 setfont 可以改变字体；loadkeys 改变键绑定，等等。

你需要至少一个虚拟终端设备，以能够使用你的显示器和键盘。因此，只有少数的编译嵌入式系统的人可以在这选 N，这样可以节约内存。这种系统要登录进去的话，只能通过网络或者并行端口的连接才能实现。

如果不清楚，选 Y，否则你将没法在你新编译的 Linux 系统上做任何事情:-)

9.20.2 Support for binding and unbinding console drivers

支持绑定和解除绑定控制台驱动

9.20.3 /dev/kmem virtual device support

/dev/kmem 虚拟设备支持

9.20.4 Non-standard serial port support

非标准串行端口支持

9.20.5 Serial drivers

串行驱动

- 8250/16550 and compatible serial support

8250/16550 和兼容串行支持

这里包含标准的串行端口。标准的回答是 Y。那些选 N 的人，要么是设置以太网 WWW/FTP 服务器，要么是使用非串行口的鼠标，他们不想要机器上的标准串行端口做任何事情。注意 Cyclades 和 Stallion 多重串行端口驱动器不需要这个驱动也可以工作。

选 M 编译为模块，名为：8250。

注意：如果你使用的是非标准串行端口，不要把这个驱动编译为模块。因为在模块卸载的时候，所有的配置信息将会丢失。这个限制在不久的将来会解决。

顺便：如果你有一个串行鼠标没有被 X window 系统识别出来，试试先运行 gpm。

顺便：如果你尝试使用一个软件 modem（也叫 Winmodem），建议是忘记它吧。这些 modem 是残疾的，它们需要的驱动只有在 Windows 下才提供。

大多数人可以在这选 Y 或者 M，这样他们可以使用串行鼠标、modem 和其他类似的连接到串行端口的设备。

– Console on 8250/16550 and compatible serial port

- Maximum number of 8250/16550 serial ports

- Number of 8250/16550 serial ports to register at runtime
- Extended 8250/16550 serial driver options
- Digi International NEO PCI Support

9.20.6 Legacy (BSD) PTY support

传统 (BSD) PTY 支持

伪终端 (PTY) 是一个由两部分组成的软件设备：一个为主设备，一个为从设备。从设备表现为一个物理终端，主设备则用于读取和写入从设备，因此表现为仿真终端。典型的主设备端软件为 telnet 服务器和 xterms。

Linux 已经使用了类似于 BSD 风格的伪终端，`/dev/ptyxx` 对于主设备，以及 `/dev/ttyxx` 对于从设备。这个方案有一些问题，包括安全问题。这个选项开启这些传统的设备；对于大多数系统，选 N 是安全的。

9.20.7 (256) Maximum number of legacy PTY in use

传统 PTY 最大使用数量

9.20.8 IPMI top-level message handler

IPMI 顶层信息处理器

9.20.9 Hardware Random Number Generator Core support

硬件随机数生成器核心支持

- Intel HW Random Number Generator support
Intel HW 随机数生成器支持
- AMD HW Random Number Generator support
AMD HW 随机数生成器支持
- AMD Geode HW Random Number Generator support
AMD Geode HW 随机数生成器支持
- VIA HW Random Number Generator support
VIA HW 随机数生成器支持

9.20.10 `/dev/nvram` support

`/dev/nvram` 支持

9.20.11 Enhanced Real Time Clock Support

加强型实时时钟支持

9.20.12 Siemens R3964 line discipline

西门子 R3964 线方法

9.20.13 Applicom intelligent fieldbus card support

Applicom 智能区域总线卡支持

9.20.14 ACP Modem (Mwave) support

ACP Modem(Mwave)支持

9.20.15 NatSemi PC8736x GPIO Support

NatSemi PC8736x GPIO 支持

9.20.16 NatSemi Base GPIO Support

NatSemi Base GPIO 支持

9.20.17 AMD CS5535/CS5536 GPIO

AMD CS5535/CS5536 GPIO

9.20.18 RAW driver

RAW 驱动

9.20.19 (256) Maximum number of RAW devices to support

支持的最大 RAW 设备数量

9.20.20 HPET - High Precision Event Timer

HPET - 高精度事件时钟

9.20.21 Allow mmap of HPET

允许 HPET 映射

9.20.22 Hangcheck timer

挂起检查时钟

9.21 I2C support

I2C 支持

I2C(发音：I-square-C) 是一个慢速的串行总线协议，它用于很多微型控制器程序中，是由 Philips 开发的。SMBus 或者系统管理总线是 I2C 协议的子协议。

I2C 和 SMBus 在这都提供了支持。你将需要这些来提供硬件检测器支持和图像支持。

如果你需要 I2C 支持，你应该在这选 Y，同时在下方的子菜单中选上你自己的总线驱动适配器。

这个驱动同样可以编译为模块，名为：i2c-core。

9.22 SPI support

SPI 支持

串行周边接口 (SPI) 是一个底层的同步协议。支持 SPI 的芯片能够将自己的传输速度提高到几十 MB 每秒。芯片在定位时配备有控制器和一个芯片选择器。很多 SPI 从设备不支持动态设备恢复，有的甚至是只写或者只读。

SPI 在微型控制器上广泛使用来与探测器、eeprom 和闪存、codecs 和其他多种控制器芯片、分析器数码转换 (d-to-a)、以及其他多种设备对话。MMC 和 SD 卡能够被 SPI 协议访问。使用 MMC 插槽的数据闪存卡，也需要使用 SPI。

SPI 是使用 4 线接口的协议 (选择、时钟、数据输入、数据输出)，这点上它与 Microwire、SSP、SSI 和 PSP 很相象。这个驱动框架能够在类似的设备和控制器上进行使用。

9.23 GPIO Support

GPIO 支持

这里开启 GPIO 支持，它使用通用的 GPIO 库。如果你想要 GPIO 扩展卡驱动支持的话，在这选 Y。

如果不清楚，选 N。

9.24 Dallas's l-wire support

达拉斯⁸单线支持

达拉斯单线总线在连接慢速的只有一个针的设备上是很有用的。这些设备通常有按键和温度探测器。

如果你需要 W1 支持，你应该在这选 Y。

这个 W1 支持也可以编译为模块，名为：wire.ko。

9.25 Power supply class support

能源供应类支持

选 Y 以开启能源供应类支持。这允许通过用户空间的 sysfs 和 uevent（如果可用的话）和 APM 内核接口（如果在下面选上的话）监视能源供应（电池、交流电源、USB）。

9.25.1 Power supply debug

能源供应调试

9.25.2 Generic PDA/phone power driver

通用 PDA/phone 能源驱动

9.25.3 DS2760 battery driver (HP iPAQ & others)

DS2760 电池驱动（HP iPAQ 和其他）

9.26 Hardware Monitoring support

硬件监视支持

硬件监视设备允许你监视系统中的硬件情况。很多现代的主板都有这些设备。它包含温度探测器、电压探测器、风扇转速探测器和各种额外的特性，比如可以控制风扇的转速。如果你需要这些支持，那么在这选 Y，同时在下面的相应探测器驱动上选 Y。

要知道你需要哪个驱动，你可以使用 sensors-detect 脚本来检测，它在 1m-sensors 包中。

选 M 编译为模块，名为：hwmon。

⁸一个通讯公司

9.27 Generic Thermal sysfs driver

通用温控 sysfs 驱动

通用温控 sysfs 驱动为温度管理提供通用的构架。通常它用于一个或者多个温控区域，以使设备冷却。

每个温控区域包含它自己的温度、差错点、冷却设备。

所有的 ACPI 平台上的温控支持都可以使用这个驱动。如果你需要的话，选 Y 或者 M。

9.28 Watchdog Timer Support

看门狗时钟支持

如果你在这选 Y（和下面的其中一个选项），并且在 `/dev/watchdog` 中建立一个字符，使用 `mknod` 命令确定主数据为 10 和从数据为 130 的话，你将可以得到一个看门狗。比如：接下来你打开一个文件，并且在 1 分钟之内写入失败的话，将导致系统重启。这在网络工作的机器上很有用，它需要在锁定后通过线路迅速反馈。在软件中也有相应的看门狗（有时会重启失败），在硬件中也有相应的看门狗主板和驱动，这个更加强大，还能够追踪你机器的温度。

看门狗通常与看门狗后台程序结合起来使用。这些后台程序同样能够监视 NFS 连接，并在处理表满负荷时重启机器。

如果不清楚，选 N。

9.29 Sonics Silicon Backplane

Sonics Silicon Backplane 支持

这里支持 Sonics Silicon Backplane 总线。如果你在为一个有这样总线的嵌入系统编译内核的话，需要选 Y。

它将会被自动选择，如果其他环境需要的话。

模块名为 `ssb`。

如果不清楚，选 N。

9.29.1 Support for SSB on PCI-bus host

支持 PCI-bus host 上的 SSB

9.29.2 SSB debugging

SSB 调试

9.29.3 SSB PCI core driver

SSB PCI 核心驱动

9.30 Multifunction device drivers

多功能设备驱动

9.30.1 Support for Silicon Motion SM501

支持 Silicon Motion SM501

9.30.2 HTC PASIC3 LED/DS1WM chip support

HTC PASIC3 LED/DS1WM 芯片支持

9.31 Voltage and Current regulators

电压和电流调节器支持

9.31.1 Voltage and Current Regulator Support

电压和电流调节器支持

9.31.2 Virtual regulator consumer support

虚拟调节器消耗支持

9.31.3 TI bq24022 Dual Input 1-Cell Li-Ion Charger IC

9.32 Multimedia devices

多媒体设备

9.32.1 Video For Linux

Linux 录像机

9.32.2 DVB for Linux

Linux DVB

9.32.3 DAB adapters

DAB 适配器

9.32.4 DABUSB driver

DABUSB 驱动

9.33 Graphics support

图形支持

9.33.1 /dev/agpgart (AGP Support)

/dev/agpgart (AGP 支持)

AGP (图形加速端口) 是一个总线系统, 它可以在你的 3D 渲染视频卡上使用 AGP 特性。这个代码表现为“AGP 驱动”的一种方式, 在主板芯片看来。

如果你需要比你的 AGP (通常支持到 256MB, 实际上根据内核定位事件, 一般只用到 64 或 128MB) 更多的结构内存, 你可以使用 PCI 访问管理, 这样可以使使用双倍的结构内存。

注意这只是在 AGP 总线上使用 MTRR 来支持 X/GLX 的整体写入。如果没有这个功能, OpenGL 直接渲染将会变得非常慢, 不过仍然比 PIO 要快。

选 M 编译为模块, 名为 agpgart。

你需要在这选 Y, 如果你要使用 GLX 或者 DRI 的话。

如果不清楚, 选 N。

- ALI chipset support
- ATI chipset support
- ATI 芯片支持
- AMD Irongate, 761, and 762 chipset support
- AMD Opteron/Athlon64 on-CPU GART support
- Intel 440LX/BX/GX, I8xx and E7x05 chipset support

Intel 芯片支持

- NVIDIA nForce/nForce2 chipset support
nVidia 芯片支持
- SiS chipset support
- Serverworks LE/HE chipset support
- VIA chipset support
- Transmeta Efficeon support

9.33.2 Direct Rendering Manager

直接渲染管理器

在内核级别支持直接渲染接口（DRI），这在新的 X 系统里推荐使用。如果你在这选 Y，那么需要在下面的子菜单中选择相应的显卡模块。这些模块提供了同步、安全和 DMA 传输。你同时也需要开启 AGP 支持。

9.33.3 Lowlevel video output switch controls

底层图像输出转换控制器

9.33.4 Support for frame buffer devices

支持帧缓冲设备

9.33.5 Backlight & LCD device support

背光和 LCD 设备支持

9.33.6 Display device support

显示设备支持

9.33.7 Console display driver support

控制台显示驱动支持

- VGA text console

VGA 文本控制台

在这选 Y，你将可以使用 Linux 文本模式，它通过通用的 VGA 标准来显示。可以说每个人都需要这个功能。

如果你有 SVAGTextMode 工具包，将可以发挥 SVGA 显卡文本模式的潜在功能。

选 Y。

- Enable Scrollback Buffer in System RAM
 - * Scrollback Buffer Size (in KB)
- Video mode selection support

9.33.8 Bootup logo

启动 logo

开启和选择帧缓冲启动 logo.

- Standard black and white Linux logo 标准黑白 Linux logo.
- Standard 16-color Linux logo 标准 16 色 Linux logo.
- Standard 224-color Linux logo 标准 224 色 Linux logo.

9.34 Sound

声音

9.34.1 Advanced Linux Sound Architecture

高级 Linux 声音体系

- Sequencer support

顺序发音器支持

选 Y 或者 M 以支持 MIDI 音序器和路由器。这个特性允许传递和排序 MIDI 事件。事件能够在给定时间里处理。

很多程序需要这个特性，所以你应该开启它，除非你知道你在做什么。

- Sequencer dummy client

- OSS Mixer API

OSS⁹混音接口

这里允许 OSS 混音接口仿真支持 (/dev/mixer*)。

很多程序用到它，所以应该选 Y。

选 M 编译为模块，名为：snd-mixer-oss。

⁹开放声音系统

- OSS PCM (digital audio) API

OSS PCM (数码音效) 接口

这里开启 OSS PCM 仿真支持 (/dev/dsp*) 。

很多程序会用到这个特性，所以你应该选 Y 。

选 M 编译为模块，名为：snd-pcm-oss 。

– OSS PCM (digital audio) API - Include plugin system

- OSS Sequencer API

OSS 顺序发音器接口

选 Y 开启 OSS 顺序发音器仿真 (/dev/sequencer 和 /dev/music) 。

很多程序会用到这个特性，所以你应该选 Y 。

选 M 编译为模块，名为：snd-seq-oss 。

- RTC Timer support
- Dynamic device file minor numbers
- Support old ALSA API
- Verbose procfs contents
- Verbose printk
- Debug
- Generic devices
- Dummy (/dev/null) soundcard
- Virtual MIDI soundcard
- MOTU MidiTimePiece AV multiport MIDI
- UART16550 serial MIDI driver
- Generic MPU-401 UART driver
- PCI devices

PCI 设备

- Analog Devices AD1889
- Avance Logic ALS300/ALS300+
- Avance Logic ALS4000
- ALi M5451 PCI Audio Controller

- ATI IXP AC97 Controller
- ATI IXP Modem
- Aureal Advantage
- Aureal Vortex
- Aureal Vortex 2
- Emagic Audiowerk 2
- Bt87x Audio Capture
- SB Audigy LS / Live 24bit
- C-Media 8338, 8738, 8768, 8770
- C-Media 8788 (Oxygen)
- Cirrus Logic (Sound Fusion) CS4281
- Cirrus Logic (Sound Fusion) CS4280/CS461x/CS462x/CS463x
- CS5530 Audio
- CS5535/CS5536 Audio
- (Echoaudio) Darla20
- (Echoaudio) Gina20
- (Echoaudio) Layla20
- (Echoaudio) Darla24
- (Echoaudio) Gina24
- (Echoaudio) Layla24
- (Echoaudio) Mona
- (Echoaudio) Mia
- (Echoaudio) 3G cards
- (Echoaudio) Indigo
- (Echoaudio) Indigo IO
- (Echoaudio) Indigo DJ
- Emu10k1 (SB Live!, Audigy, E-mu APS)
- Emu10k1X (Dell OEM Version)
- (Creative) Ensoniq AudioPCI 1370
- (Creative) Ensoniq AudioPCI 1371/1373
- ESS ES1938/1946/1969 (Solo-1)
- ESS ES1968/1978 (Maestro-1/2/2E)
- ForteMedia FM801
- Intel HD Audio

- RME Hammerfall DSP Audio
 - TempoTec HiFier Fantasia
 - ICEensemble ICE1712 (Envy24)
 - ICE/VT1724/1720 (Envy24HT/PT)
 - Intel/SiS/nVidia/AMD/ALi AC97 Controller
Intel/SiS/nVidia/AMD/ALi AC97 控制器
选 Y 以支持相应主板的 AC97 声效卡控制器。
选 M 编译为模块，名为：snd-intel8x0。
 - Intel/SiS/nVidia/AMD MC97 Modem
 - Korg 1212 IO
 - ESS Allegro/Maestro3
 - Digigram miXart
 - NeoMagic NM256AV/ZX
 - Digigram PCXHR
 - Conexant Riptide
 - RME Digi32, 32/8, 32 PRO
 - RME Digi96, 96/8, 96/8 PRO
 - SiS 7019 Audio Accelerator
 - S3 SonicVibes
 - Trident 4D-Wave DX/NX; SiS 7018
 - VIA 82C686A/B, 8233/8235 AC97 Controller
 - VIA 82C686A/B, 8233 based Modems
 - Asus Virtuoso 100/200 (Xonar)
 - Digigram VX222
 - Yamaha YMF724/740/744/754
- USB devices
 - USB Audio/MIDI driver
 - Tascam US-122, US-224 and US-428 USB driver
 - Native Instruments USB audio devices
 - System on Chip audio support
 - LSA for SoC audio support
 - ALSA SoC audio for Freescale SOCs
 - SoC Audio for the Texas Instruments OMAP

9.34.2 Open Sound System

开放式声音系统（废弃不用）

9.35 HID Devices

HID 设备

选 Y 将可以看到各种计算机 - 人性化接口设备驱动。这个选项本身不增加内核代码。

如果你选 N，所有的子选项将被禁止。

9.35.1 Generic HID support

通用 HID 支持

人性化接口设备是一种计算机设备，它们相互直接作用，并从人的输入中获取信息。术语“HID”最通常的意义是指 USB-HID，但是其他的设备（比如，蓝牙设备，不严格区分的话）也是设计为使用 HID（这里包含一些特定的键盘、鼠标、按钮等等）。这个选项把通用的 HID 支持代码（分析、使用功能等等）编译进入内核，这样可以被一些特定的 HID 工具设备使用（比如 USB 或者蓝牙）。

如果不清楚，选 Y。

- HID debugging support HID 调试支持
- `/dev/hidraw` raw HID device support `/dev/hidraw` raw HID 设备支持

9.35.2 USB Human Interface Device (full HID) support

USB 人性化接口设备（完全 HID）支持

选 Y，如果你需要在连接 USB 键盘、鼠标、手柄、图形按钮或者其他基于 HID 设备时得到完全 HID 支持。这也支持 UPS（不间断电源）和监视器控制设备。

如果不清楚，选 Y。

选 M 编译为模块，名为：`usbhid`。

- Enable support for Apple laptop/aluminum USB special keys 支持 Apple 笔记本/铝¹⁰USB 特殊按键
- `/dev/hiddev` raw HID device support `/dev/hiddev` raw HID 设备支持

¹⁰苹果笔记本的一种，用大块铝做外壳，被称为砖头做的机器。

9.36 USB support

USB 支持

这个选项增加对通用串行总线（USB）的核心支持。你同时也需要子菜单中的相关驱动来使用每个 USB 设备。

9.36.1 Support for Host-side USB

支持主机侧 USB

通用串行总线（USB）是一个特殊的串行总线子系统，它与传统的 PC 串行端口相比，能提供更高速度和更多的特性。这种总线提供对周边设备的能源，并允许热插拔和切换。多达 127 个 USB 周边设备可以通过树形结构被连接到一个 USB 主机侧。

USB 主机是这个树形结构的根，那些周边设备是叶子，内节点是特殊的 USB 设备，叫做 hub。大多数 PC 有 USB 主机侧，用于连接各种周边设备，比如扫描仪、键盘、鼠标、modem、摄像头、磁盘、闪存、网络连接、打印机等等。

在这选 Y，如果你的计算机有一个 USB 主机侧端口，并且你想要使用 USB 设备。你还需要在下面的选项“Host Controller Driver (HCD)”中至少选择一个驱动。选择一个 USB 1.1 控制器，比如选择“UHCI HCD support”或“OHCI HCD support”，以及“EHCI HCD (USB2.0) support”，如果你使用的不是老系统（没有 USB2.0）。如果你不确定，全部选上也没有问题。

如果你的系统有一个设备侧¹¹USB 端口，使用周边的 USB 协议的话，那么请查看“USB Gadget”选项结构。

选 M 编译为模块，名为：usbcore。

- USB verbose debug messages
- USB announce new devices
- USB device filesystem
- USB device class-devices
- Dynamic USB minor allocation
- USB selective suspend/resume and wakeup

USB 选择性挂起/恢复和苏醒

如果你在这选 Y，你将可以使用驱动调用功能，或者 sysfs 的“power/level”文件来挂起和恢复 USB 周边设备，也可以开启和关闭自动挂起功能。

¹¹主机侧的意思是，你的系统可以作为 USB 的主机使用，它提供 USB 服务。设备侧的意思是，你的系统可作为 USB 设备，享受 USB 主机给你提供的服务。这与 server 和 client 的关系比较类似。

同时，USB “远程苏醒”信号在此也提供支持，这可以让一些 USB 设备（比如键盘和网卡）能够唤醒他们的父集线器。这种唤醒串联至 USB 树之中，可以通过一系列状态，比如内存挂起，唤醒整个系统。

如果你不清楚，选 N。

- USB Monitor
- Support WUSB Cable Based Association (CBA)
- Cypress C67x00 HCD support
- EHCI HCD (USB 2.0) support

EHCI HCD (USB 2.0) 支持

增强主控制器接口（EHCI）是 USB 2.0 的标准，它是高速（480Mbit/sec, 60Mbyte/sec）主控制器硬件。如果你的 USB 主控制器支持 USB 2.0，你将需要配置这个主控制器接口。此时，主要的 EHCI 工具芯片是 NEC 生产的，广泛用于外加的 PCI 卡，但是其他的生产厂商也提供这些执行工具，比如 Intel 和 Philips。很多主板也开始支持 USB 2.0。

EHCI 控制器以“控制器伴侣”的形式打包在 USB 1.1 主控制器（OHCI 或 UHCI）中，以处理连接到根集线器端口的 USB 1.1 设备。如果端口接上的是高速设备，那么它将连接至 EHCI，否则它将连接至另一个伴侣控制器¹²。如果你配置了 EHCI，你应该也要同时配置 OHCI¹³和 UHCI¹⁴。

选 M 编译为模块，名为：ehci-hcd.

– Root Hub Transaction Translators

根集线器处理转换器

一些 EHCI 芯片有厂商专门提供的扩展功能，用于处理 USB 转换，这样就不需要 OHCI 或者 UHCI 伴侣控制器。在这选 Y 是安全的，即使你的控制器不支持这个特性。

这支持通常由 ARC 提供的 EHCI 执行工具，似乎还更新了好几次。

- ISP116X HCD support
- OHCI HCD support

OHCI HCD 支持

开放式主控制器接口（OHCI）是 USB 1.1 主控制器硬件标准。很多硬件都使用了这种接口，多于 Intel 自己的 UHCI 接口。如果你的 USB 主控制器使用 OHCI，那么在这选 Y。在大多数非 x86 系统，和不使用 Intel 及 VIA 的 x86 硬件，选 Y 是非常合适的。如果你的主控制器不使用 PCI，在这选 Y 同样也是合适的。对基于 PCI 的系统，如果你不确定的话，用“lspci -v”命令可以在项目“prog-if”中显示正确的 USB 控制器：EHCI、OHCI 或者 UHCI。

选 M 编译为模块，名为：ohci-hcd.

¹²即 OHCI 或 UHCI

¹³用于 NEC 和一些厂商的 USB 主控制器驱动

¹⁴用于 Via 主板 USB 主控制器驱动

- UHCI HCD (most Intel and VIA) support

UHCI HCD (大多数 Intel 和 VIA) 支持

通用主控制器接口 (UHCI) 是 Intel 使用的 USB 硬件标准, 用于访问 PC 中的 USB 设备, 也称为 USB 主控制器。如果你的 USB 主控制器使用这个标准, 那么在这选 Y。所有的近期使用 Intel PCI 芯片的主板 (比如 intel 430TX, 440FX, 440LX, 440BX, i810, i820) 使用这个标准。同样所有的 VIA PCI 芯片也使用这个标准 (比如 VIA VP2, VP3, MVP3, Apollo Pro, Apollo Pro II, Apollo Pro 133) 。

如果不清楚, 选 Y。

选 M 编译为模块, 名为: uhci-hcd.

- SL811HS HCD support
- R8A66597 HCD support

9.36.2 USB Modem (CDC ACM) support

USB Modem 支持

9.36.3 USB Printer support

USB 打印机支持

9.36.4 USB Wireless Device Management support

USB 无线设备管理支持

9.36.5 USB Test and Measurement Class support

USB 测试和测量类支持

9.36.6 USB Mass Storage support

USB 通用储存支持

选 Y, 如果你想要连接 USB 通用储存设备至你电脑上的 USB 端口。这里提供了 USB 软盘、USB 硬盘、USB 磁带机、USB 光驱、USB 闪存设备、内存棒和其他类似设备的驱动支持。这个驱动也可以用于一些数码相机和读卡器。

这个选项依赖于 SCSI 支持¹⁵, 因此你需要选择‘SCSI device support: SCSI disk support’以支持大多数的 USB 储存设备。

选 M 编译为模块, 名为: usb-storage。

¹⁵因此要使用 USB 储存设备, 你需要在前面的 SCSI 相关选项中选 Y。

- USB Mass Storage verbose debug
- Datafab Compact Flash Reader support
- Freecom USB/ATAPI Bridge support
- ISD-200 USB/ATA Bridge support
- Microtech/ZiO! CompactFlash/SmartMedia support
- USBAT/USBAT02-based storage support
- SanDisk SDDR-09 (and other SmartMedia) support
- SanDisk SDDR-55 SmartMedia support
- Lexar Jumpshot Compact Flash Reader
- Olympus MAUSB-10/Fuji DPC-R1 support
- Support OneTouch Button on Maxtor Hard Drives
- Support for Rio Karma music player
- SAT emulation on Cypress USB/ATA Bridge with ATACB

9.36.7 The shared table of common (or usual) storage devices

常规 (或者常用) 储存设备共享表

9.36.8 USB Mustek MDC800 Digital Camera support

USB Mustek MDC800 数码相机支持

9.36.9 Microtek X6USB scanner support

Microtek X6USB 扫描仪支持

9.36.10 USB Serial Converter support

USB 串行转换器支持

9.36.11 EMI 6|2m USB Audio interface support

EMI 6|2m USB 声音接口支持

9.36.12 EMI 2|6 USB Audio interface support

EMI 2|6 USB 声音接口支持

9.36.13 ADU devices from Ontrak Control Systems

Ontrak 控制系统 ADU 设备

9.36.14 USB Auerswald ISDN support

USB Auerswald ISDN 支持

9.36.15 USB Diamond Rio500 support

USB Diamond Rio500 支持

9.36.16 USB Lego Infrared Tower support

USB Lego Infrared Tower 支持

9.36.17 USB LCD driver support

USB LCD 支持

9.36.18 USB BlackBerry recharge support

USB BlackBerry 充电支持

9.36.19 USB LED driver support

USB LED 驱动支持

9.36.20 Cypress CY7C63xxx USB driver support

Cypress CY7C63xxx USB 驱动支持

9.36.21 Cypress USB thermometer driver support

Cypress USB thermometer 支持

9.36.22 USB Phidgets drivers

USB Phidgets 驱动

9.36.23 Siemens ID USB Mouse Fingerprint sensor support

西门子 ID USB 鼠标手指打印感应器支持

9.36.24 Elan PCMCIA CardBus Adapter USB Client

Elan PCMCIA CardBus 适配器 USB 客户机

9.36.25 Apple Cinema Display support

Apple 电影显示支持

9.36.26 USB 2.0 SVGA dongle support (Net2280/SiS315)

USB2.0 SVGA dongle 支持 (Net2280/SiS315)

9.36.27 USB LD driver

USB LD 驱动

9.36.28 PlayStation 2 Trance Vibrator driver support

PlayStation 2 Trance Vibrator 驱动支持

9.36.29 IO Warrior driver support

IO Warrior 驱动支持

9.36.30 USB testing driver

USB 测试驱动

9.36.31 iSight firmware loading support

iSight 固件加载支持

9.36.32 USB Gadget Support

USB Gadget 支持

9.37 Ultra Wideband devices

超宽波段设备 (UWB)

UWB 是一个高波段、低能耗、点对点的无线电技术，它使用宽波谱 (3.1—10.6GHz)。它多用于室内 (2 米内 480Mbps，10 米内 110Mbps 传输速度)。它作为其他协议的一个传输层，比如无线 USB、无线 IP 和即将来临的蓝牙系统和 1394 火线。

这个技术是点对点的；但是，更高层的协议 (比如无线 USB) 会强加一个主/从关系。

选 Y，如果你的电脑有 UWB 无线控制器。你还需要在下面的子菜单开启相应的选项。你也可以把它们都选上，这没什么害处。

选 M 编译为模块。

9.38 MMC/SD card support

MMC/SD 卡支持

MMC 是多媒体卡总线协议。

如果你需要 MMC 支持，你需要在这选 Y，同时下面的选项中为你的 MMC 接口选择特定的驱动。

9.39 Sony MemoryStick card support

索尼 MemoryStick 卡支持

Sony MemoryStick 是一个专用的储存/扩展卡协议。

如果你需要 MemoryStick 支持，你需要在这选 Y，同时下面的子选项中选择相应的驱动。

9.40 LED Support

LED¹⁶支持

选 Y 支持 Linux LED。这允许在用户空间和可选的内核事件中 (触发器) 对相关的 LED 进行控制。

这里与键盘上的 LED 没有任何关系，它们是被输入系统进行控制的。

¹⁶发光二极管

9.41 Accessibility support

易用性支持

易用性支持是为残疾人设置的功能，它对各种硬件设备或者软件适配器进行调整，以帮助身体有残疾的人士（比如盲人）使用电脑。

其中包含盲人字法设备、语音合成、键盘重映射等等。

在这选 Y，可以看到易用性支持的选项。这个选项本身不增加内核代码。

如果你选 N，所有的子选项将被禁用。

如果不清楚，选 N。

9.42 InfiniBand support

InfiniBand 支持

对 InfiniBand（IB）的核心支持。确保你在下面的子选项中选择与你硬件相匹配的驱动。

9.43 Real Time Clock

实时时钟（RTC）

通用 RTC 类支持。如果你在这选 Y，你将可以在你的系统插入一个或者多个 RTC。你需要在下面的子选项中允许一个或者几个接口。

选 M 编译为模块，名为：rtc-core。

9.44 DMA Engine support

DMA 引擎支持

DMA 引擎可以在不通过主 CPU 的情况进行异步数据传输。目前，这个框架可以用于在网络栈中卸下内存拷贝和在 MD 驱动器中开展 RAID 操作。

9.45 Userspace I/O drivers

用户空间 I/O 驱动

选 Y 以允许用户空间驱动的核心代码。这个代码允许用户空间程序容易地访问内核中断和内存位置，允许一些驱动在用户空间中写入。注意一个小内核驱动同样需要这个功能来确保正确地中断相关工作。

如果你不清楚，选 N。

9.46 Staging drivers

补充驱动

这个选项允许你选择更多的驱动。这些驱动不是常规内核级别的，它们放在这里的目的是给更多读者使用。注意这些驱动正在开发之中，很可能无法工作，也可能所包含的用户空间接口将在不久后改变。

使用这些驱动将会让你的内核变得不纯净，这将影响到相关的支持选项，通常这些支持由社区群体和各种商业组织提供。

如果你打算在这些驱动上工作，要改进它们，或者报告相应的问题，请查看文件 `driver_name.README` 中的 `drivers/staging` 目录获得更多信息。

如果不清楚，选 N。

Chapter 10

Firmware Drivers 固件驱动

固件就是你板上的 BIOS、各种显卡芯片之类的已经固化好的记录某些特定数据的东西。

10.1 BIOS Enhanced Disk Drive calls determine boot disk

BIOS 加强磁盘功能，确定启动盘

选 Y 或 M，如果你要使用 BIOS 加强磁盘服务功能来确定 BIOS 用哪个磁盘来启动。启动后这个信息会反映在系统文件中。

这个选项是实验性的，而且已经被确认在某些未测试选项下会启动失败。很多磁盘控制器的 BIOS 供应商都不支持这个特性。

10.2 EFI Variable Support via sysfs

通过 sysfs 的 EFI 变量支持

如果你在这选 Y，你将可以通过 sysfs 得到 EFI（扩展固件接口）变量信息。通过这个接口，你可以读、写、建立和删除 EFI 变量。

10.3 BIOS update support for DELL systems via sysfs

用于 DELL 机器的 BIOS 升级支持

如果你想要支持你的 Dell 机器的 BIOS 升级，选 M。注意你需要 Dell 相关的软件包支持（开放式管理器或者升级包）以便与 BIOS 会话。

10.4 Dell Systems Management Base Driver

DELL 系统管理器的基本驱动

Dell 系统管理基本驱动提供一个 `sysfs` 接口，这使得系统管理软件在相应的 Dell 系统上能够生成系统管理中断（SMI）和主控制行为（系统电源队列或者在操作系统关闭后关闭机箱电源）。

10.5 Export DMI identification via sysfs to userspace

通过 `sysfs` 输出 DMI 认证至用户空间

选 Y，如果你想要从用户空间的 `/sys/class/dmi/id/` 队列化 SMBIOS/DMI 系统识别信息或是想要基于 DMI 的模块能够自动装载。

10.6 iSCSI Boot Firmware Table Attributes

iSCSI 启动固件表属性

这个选项允许内核寻找 iSCSI 启动固件表的内存区域。这对于 iSCSI 启动固件表属性模块的正常工作很有必要。

Chapter 11

File systems 文件系统

11.1 Second extended fs support

第二扩展文件系统支持

Ext2 是标准的 Linux 硬盘文件系统。

选 m 编译为模块，名为：ext2。如果不清楚，选 Y。

11.1.1 Ext2 extended attributes

Ext2 扩展属性

Ext2 扩展属性是名字：值的配对属性，与内核或者用户的节点关联。

如果不清楚，选 N。

- Ext2 POSIX Access Control Lists

Ext2 POSIX 访问控制清单

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

- Ext2 Security Labels

Ext2 安全标签

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 ext2 文件系统中的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.1.2 Ext2 execute in place support

Ext2 适合运行支持

适合运行可以用于后台内存块设备。如果你允许这个选项，你可以选择挂载块设备，它们可以不通过使用分页缓存来实现这个功能。

如果你不使用具有以上功能的块设备，或者不清楚，选 N。

11.2 Ext3 journalling file system support

Ext3 日志文件系统

这是第二扩展文件系统的日志版本（经常叫 ext3），实际上的标准 Linux 硬盘文件系统（为了有效组织储存设备上的文件）。

这个驱动中包含的日志代码，意味着你不需要在文件系统崩溃后运行 e2fsck（文件系统检查工具）。日志保持着更改的痕迹，这些痕迹在系统崩溃时造成。它能够确保你的文件系统保持和原来的一致，而不需要进行长时间的检查。

除了增加日志到文件系统，ext3 的磁盘文件格式和 ext2 相同。这使得你可以自由地在两者之间进行转换，只要你干净地卸载了文件系统，或者在文件系统中用 e2fsck 运行检查。

要给现有的 ext2 文件系统增加日志，或者更改 ext3 文件系统的行为，你可以使用实用工具 tune2fs（“man tune2fs”查看用法）。要更改 ext3 文件的文件和目录属性，使用 chattr（“man chattr”查看用法）。你需要使用 e2fsprogs（1.20 版本或更高）来建立 ext3 日志。

选 m 编译为模块，名为：ext3。

11.2.1 Ext3 extended attributes

Ext3 扩展属性

扩展属性是名字：值的配对属性，与内核或者用户的节点关联。

如果不清楚，选 N。

你需要选 Y 以保证 POSIX ACL 对 ext3 的支持。

- Ext3 POSIX Access Control Lists

Ext3 POSIX 访问控制清单

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

- Ext3 Security Labels

Ext3 安全标签

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 ext3 文件系统中的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.3 The Extended 4 (ext4) filesystem

ext4 扩展文件系统

这是下一代 ext3 文件系统。

不像从 ext2 到 ext3 的变化，ext4 在磁盘上的格式已经不再和 ext3 兼容：它基于大范围的映像，并且支持 48 位的物理块数量。这些整合在磁盘格式上的变化将使 ext4 能够管理超过 16 TB 的文件系统容量 - ext3 无法逾越的障碍，如果不改变它的格式的话。

ext4 还有其他的新特性，比如支持延迟分配，持续的预分配，高分辨率的时间戳，和更大的文件支持等，它还有很多的特性来改善和加速 fsck¹ 时间。

ext4 文件系统还将支持挂载 ext3 文件系统。由于将可能获得延迟分配和内节点表抢先读等优势，最好的行为就是开启 ext4 特性支持，或者格式化一个新的 ext4 初始化文件系统。

选 m 编译为模块，名为 ext4。

如果不清楚，选 N。

11.3.1 Enable ext4dev compatibility

开启 ext4dev 兼容

从内核 2.6.28 开始，ext4 文件系统正式由之前的 ext4dev 命名为 ext4。不过不幸的是，有很多用户空间的程序（比如 klibc 的文件系统）在硬代码上写上了“ext4dev”。

为了保持兼容性，以便系统仍然使用 ext4dev 来挂载 ext4 文件系统，在这选 Y。这个特性将会在 2.6.31 版本的内核移除，所以你最好安排在恰当的时候修复你的用户空间程序！

11.3.2 Ext4 extended attributes

Ext4 扩展属性

扩展属性是名字：值的配对属性，与内核或者用户的节点关联。

如果不清楚，选 N。

你需要选 Y 以保证 POSIX ACL 对 ext4 的支持。

- Ext4 POSIX Access Control Lists

Ext4 POSIX 访问控制清单

¹文件系统检查工具

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

- Ext4 Security Labels

Ext4 安全标签

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 ext4 文件系统中的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.4 JBD (ext3) debugging support

JBD(ext3)调试支持

如果你正在使用 ext3 日志文件系统（或者其他文件系统/设备可能会潜在使用 JBD），这个选项可以让你在系统运行时开启调试输出，以便追踪任何错误。默认地这些调试输出是关闭的。

如果你在这选 Y，那么你将可以打开调试，使用“echo N > /sys/kernel/debug/jbd/jbd-debug”，其中 N 是从 1-5 的数字，越高产生的调试输出越多。要再次关闭，使用“echo 0 > /sys/kernel/debug/jbd/jbd-debug”。

11.5 JBD2 (ext4) debugging support

JBD2(ext4)调试支持

如果你正在使用 ext4 日志文件系统（或者其他文件系统/设备可能会潜在使用 JBD2），这个选项可以让你在系统运行时开启调试输出，以便追踪任何错误。默认地这些调试输出是关闭的。

如果你在这选 Y，那么你将可以打开调试，使用“echo N > /sys/kernel/debug/jbd2/jbd2-debug”，其中 N 是从 1-5 的数字，越高产生的调试输出越多。要再次关闭，使用“echo 0 > /sys/kernel/debug/jbd2/jbd2-debug”。

11.6 Reiserfs support

Reiserfs 支持

不仅储存文件名字，而且在一个平衡的树中包含文件本身。使用日志。

平衡树在结构构架上比传统的文件系统更为有效。

通常地，ReiserFS 和 ext2 一样快，但是在目录和小文件上更为有效。在 NFS 和配额上，你需要额外的补丁。

比起基于块配置的文件系统来说，这更加容易扩展现有的数据库和关键字搜寻系统的特性。下一个版本将会变得如此可扩展，也将会支持插件，就如同我们的口号：“开源软件需要更多的许可。”

如果你喜欢它，你可以支付一定的费用来增加你需要的新的特性，购买一个支持合约，或者嵌入另一个操作系统。

11.6.1 Enable reiserfs debug mode

允许 reiserfs 调试模式

如果你选 Y，ReiserFS 将会在操作过程中进行它所能进行的各项检查。它在实质上会变得很慢。我们不只一次地忘记了这个选项是打开的，从而导致我们对于最新版本的基准相当沮丧。:-)使用这个功能，我们的队伍可以进行一致地全面检查，而不需考虑它对于终端用户的影响。如果你濒于发送错误报告，那么选 Y，你将能够得到有用的出错信息。几乎每个人都可以选 N。

11.6.2 Stats in /proc/fs/reiserfs

在 /proc/fs/reiserfs 中建立文件分级体系，显示各种 ReiserFS 统计数据和内部数据，这将导致你的内核和模块变大(+8 KB)。这也在每项挂载时增加内核的内存花销。除了 ReiserFS 开发者、喜欢切换 reiserfs 或者追踪错误的人，其他人都可以选 N。

11.6.3 ReiserFS extended attributes

ReiserFS 扩展属性

扩展属性是名字：值的配对属性，与内核或者用户的节点关联。

如果不清楚，选 N。

- ReiserFS POSIX Access Control Lists

ReiserFS POSIX 访问控制清单

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

- ReiserFS Security Labels

ReiserFS 安全标签

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 ReiserFS 文件系统中的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.7 JFS filesystem support

JFS 文件系统支持

这是 IBM 的日志文件系统端口。

如果你不打算使用 JFS 文件系统，选 N。

11.7.1 JFS POSIX Access Control Lists

JFS POSIX 访问控制清单

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

11.7.2 JFS Security Labels

JFS 安全标签

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 JFS 文件系统中的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.7.3 JFS debugging

JFS 调试

如果你在 JFS 文件系统中试验某些问题，选 Y。这将会在系统日志中增加调试信息。在一般情况下，这会导致一点点小的开销。

11.7.4 JFS statistics

JFS 统计数据

选 Y，将会在 /proc/fs/jfs/ 目录中为用户生成 JFS 文件系统统计数据。

11.8 XFS filesystem support

XFS 文件系统支持

XFS 是一个高性能的日志文件系统，它在 SGI IRIX 平台上产生。它是完全的多线程，支持大文件和大的文件系统、扩展属性、多种块尺寸，是广泛的基准，被广泛用于 Btrees（目录、范围、自由空间）来同时实现性能和可伸缩性。

选 m 编译为模块，名为：xfs。但是要注意，如果你的启动分区的文件系统被编译为模块，那么你需要使用初始储存盘来启动。

11.8.1 XFS Quota support

XFS 配额支持

如果你选 Y，那么你将可以在 XFS 文件系统下对每个用户或者每个群组进行磁盘使用量的限制。XFS 把配额信息当作文件系统的元数据，并使用日志来保证高级别的一致性。硬盘上的配额数据格式与 IRIX 平台下的 XFS 格式一致，这就允许文件系统在不作任何转换的情况下在 Linux 和 IRIX 平台之间迁移。

如果不清楚，选 N。无论是否有通用的配额支持，XFS 配额都可以使用。它们是完全独立的子系统。

11.8.2 XFS Security Label support

XFS 安全标签支持

安全标签支持可选的访问控制模块，这些模块被例如 SELinux 中的安全模块执行。这个选项允许对 XFS 文件系统上的文件安全标签进行扩展属性管理。

如果你不是在使用一个安全模块（它需要使用文件安全标签扩展属性），那么选 N。

11.8.3 XFS POSIX ACL support

XFS POSIX 访问控制清单

POSIX 访问控制清单 (ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

11.8.4 XFS Realtime subvolume support

XFS 实时子容量支持

如果你选 Y，那么你将能够挂载包含实时子容量的 XFS 文件系统。实时子容量是一个独立的磁盘空间区域，那里只储存文件的数据。它原先被设计为提供决定性的数据比率以匹配多媒体流程序，但是它同样作为一个通用的机制以确定数据。元数据/日志/IO 是完全独立的。常规的 I/O 文件与独立的设备在所有的响应上相隔离，而它能够非常透明地实施程序功能，通过继承的实时目录内节点标记。

如果不清楚，选 N。

11.9 GFS2 file system support

GFS2 文件系统支持

一个簇文件系统。

它允许计算机的簇同时使用一个共享给它们的块设备（包含 FC, iSCSI, NBD，等等）。GFS 像本地的文件系统一样读写块设备，但是使用一个锁定的模块，

来使计算机协调它们的 IO，以便文件系统能够保持一致性。其中一个非常精巧的特性就是对一台机器上文件系统的完美地一致性 – 更改，能够很快地通过簇，反映到其他的机器上。

要使用 GFS 文件系统，你需要开启一个或者多个以下被锁定的模块。

11.9.1 GFS2 “nolock” locking module

GFS2 “nolock” 锁定模块

GFS2 的单节点锁定模块。

使用这个模块，如果你想要把 GFS2 用于单节点，而不使用它的簇特性。你仍然可以利用大文件支持，升级至完全的簇特性，如果在后面需要的话。

如果你仅仅把 GFS2 用于簇模式，那么你不需要这个模块。

11.9.2 GFS2 DLM locking module

GFS2 DLM 锁定模块

GFS2 多节点锁定模块。

大多数的 GFS2 用户将需要这个模块。它提供 GFS2 和 DLM 之间的锁定接口，这需要 GFS2 运行在簇环境下。

11.10 OCFS2 file system support

OCFS2 文件系统支持

OCFS2 是一个通用的基于共享磁盘的簇文件系统，很多地方类似于 ext3。它支持 64 位内部节点数量，能够自动扩展元数据组，这样可以让它使用在非簇的环境。

你将需要安装 ocfs2-tools 包，以便至少能够使用 mount.ocfs2。

11.10.1 OCFS2 logging support

OCFS2 记录支持

ocfs2 文件系统有一个广泛的记录系统。这个系统允许事件选择在文件 `/sys/o2-cb/logmask/` 中进行记录。这个选项将增大你的内核，但是允许你调试 ocfs2 文件系统的关键。

11.10.2 OCFS2 expensive checks

OCFS2 昂贵检查

这个选项允许进行大花销的一致性检查。选这里仅仅是为了调试，它会影响文件系统的性能。

11.11 Dnotify support

Dnotify 支持

Dnotify 是一个基于目录的每个文件描述变化通知系统，它使用事件信号来与用户空间连接。现在有更好的选择，但是一些程序可能仍然响应 dnotify。

11.12 Inotify file change notification support

Inotify 文件变化通知支持

选 `y` 来开启 Inotify 支持。它是一个文件变化通知系统，代替了 dnotify 的位置。Inotify 修正了 dnotify 中的很多缺点，并且提出一些新的特性，包括多重文件事件、一次性支持、卸载通知。

如果不清楚，选 `Y`。

11.12.1 Inotify support for userspace

Inotify 用户空间支持

选 `Y`，以开启 Inotify 用户空间支持，包括相关的系统调用。Inotify 允许监视文件和目录，通过一个单独的开放的文件描述。事件通过文件描述进行阅读，用 `select()` 和 `poll()` 也可以实现。

11.13 Quota support

配额支持

如果你选 `Y`，你将可以设置每个用户在磁盘使用上的空间限制（也叫磁盘配额）。现在，它在 `ext2`, `ext3`, 和 `reiserfs` 文件系统中使用。`ext3` 也支持日志配额，这样你在意外关机时不需要运行 `quotacheck(8)`。

配额支持很可能只是对多用户系统使用。如果不清楚，选 `N`。

11.13.1 Report quota messages through netlink interface

通过网络连接接口报告配额信息

如果你选 `Y`，配额的警告（关于超过软件限制，寻找硬件限制等）将会通过网络连接接口进行报告。如果不清楚，选 `Y`。

11.13.2 Print quota warnings to console

在控制台打印配额警告（废弃不用）

如果你选 `Y`，配额的警告（关于超过软件限制，寻找硬件限制等）将会在处理器控制终端打印出来。注意这个行为目前是被反对的，将来会被移除。请使用通过网络连接套接字通知来代替。

11.13.3 Old quota format support

老配额格式支持

这个配额格式在2.4.22 版本的内核之前使用。如果你正在使用老的配额，而现在又不想转换到新的格式，那么选 Y。

11.13.4 Quota format v2 support

配额格式 v2 支持

这个配额格式使用 32 位 UIDs/GIDs 。如果你需要这个功能，选 Y。

11.14 Kernel automounter support

内核自动挂载支持

自动挂载是一个工具，它在需要时自动挂载远程文件系统。这种执行方式是部分基于内核的，目的是在已经挂载的容器中减少花销。这不同于 BSD 自动挂载 (amd)，那是纯粹的用户空间后台程序。

要使用自动挂载你需要使用 autofs 包中的用户空间工具。你同样需要在下面的“NFS 文件系统支持”选项中选 Y。

如果你想要使用更新版本更多功能的自动挂载，在这选 N，然后在下面的“Kernel automounter v4 support”选项中选 Y。

选 m 编译为模块，名为：autofs。

如果你不是很大的、广泛分发网络的一部分，你可能不需要自动挂载，可以在这选 N。

11.15 Kernel automounter version 4 support

内核自动挂载 V4 支持

自动挂载是一个工具，它在需要时自动挂载远程文件系统。这种执行方式是部分基于内核的，目的是在已经挂载的容器中减少花销。这不同于 BSD 自动挂载 (amd)，那是纯粹的用户空间后台程序。

要使用自动挂载你需要使用 autofs 包中的用户空间工具。你同样需要在下面的“NFS 文件系统支持”选项中选 Y。

选 m 编译为模块，名为 autofs4，你将需要增加“alias autofs autofs4”语句到你的模块配置文件中。

如果你不是很大的、广泛分发网络的一部分，或者不是一个笔记本，需要在局域网络中动态地重新配置，你可能不需要自动挂载，可以在这选 N。

11.16 Filesystem in Userspace support

文件系统用户空间支持 (FUSE)

在 FUSE 支持下, 你将可以使一个完全功能化的文件系统在用户空间生效。

这里同时有一个配套的库: libfuse 。

如果你想要开发一个用户空间文件系统, 或者你想要使用基于 FUSE 的文件系统, 选 Y 或者 M 。

11.17 CD-ROM/DVD Filesystems

CD-ROM/DVD 文件系统

11.17.1 ISO 9660 CDROM file system support

ISO 9660 CDROM 文件系统支持

这是 CD-ROM 中使用的标准文件系统。它在之前被称做“高锯齿文件系统”, 在其他的 Unix 系统中简称为“hsfs”。所谓的“岩石山脊”扩展, 允许支持 Unix 长文件名和符号连接。在这个驱动中也同样支持。如果你有一个 CD-ROM 驱动器, 而除了听 CD 音乐和看着二极管灯发光以外, 你还想做点别的事情, 那么在这选 Y, 这会使你的内核增大 27KB。否则, 选 N。

选 m 编译为模块, 名为 isofs 。

- Microsoft Joliet CDROM extensions

Microsoft Joliet CDROM 扩展

Joliet 是 Microsoft 对于 ISO 9660 CD-ROM 文件系统的扩展, 它支持 unicode 长文件名格式 (unicode 是一种新的 16 位的字符代码, 是 ASCII 的接替者, 几乎包含世界上的所有语言的所有字符)。选 Y, 如果你想要在 Linux 下阅读 Joliet CD-ROM 。

- Transparent decompression extension

透明解压扩展

这是 Linux 典型的针对 RockRidge (岩石山脊) 的扩展功能, 它可以让你在 CD-ROM 中储存压缩数据, 同时在访问时透明地解压出来。

选 Y, 如果你想要阅读上述压缩的 CD-ROM 。

11.17.2 UDF file system support

UDF 文件系统支持

这是在 CD-ROM 和 DVD 中使用的新的文件系统, 选 Y, 如果你想要以包模式挂载 DVD 或者 CDRW, 或者以其他 UDF 实用程序写盘, 如 DirectCD 。

选 m 编译为模块, 名为 udf 。

如果不清楚, 选 N 。

11.18 DOS/FAT/NT Filesystems

DOS/FAT/NT 文件系统

11.18.1 MSDOS fs support

MSDOS 文件系统支持

这允许你挂载硬盘和软盘中的 MSDOS 分区（除非它们是压缩的；要在 Linux 下访问压缩的 MSDOS 分区，你可以使用 DOS 仿真器 DOSEMU，或者试试 dmsdosfs 工具。如果你想要在非压缩的 MSDOS 分区上使用 dosemu，在这选 Y）。这意味着文件访问变得透明，也就是说，MSDOS 文件看起来和执行起来和其他的 Unix 文件一样。

如果你有 Windows 95 或者 Windows NT 安装在你的 MSDOS 分区，你将要使用 VFAT 文件系统（在下面的“VFAT fs support”选项中选 Y），否则你将不能看到由 Windows 95 / Windows NT 生成的长文件名。

这个选项将会使你的内核增大 7 KB。如果不清楚，选 Y。你要在“DOS FAT fs support”选项中也选 Y，这项功能才会工作。选 M 编译为模块，名为：msdos。

11.18.2 VFAT (Windows-95) fs support

VFAT (Windows-95) 文件系统支持

这个选项提供对常规 Windows 长文件名文件系统的支持。其中包括非压缩的基于 FAT 的文件系统，如 Windows 95, Windows 98, Windows NT 4.0，以及来自于 mtools 包的 Unix 程序。

VFAT 支持会增大你的内核 10 KB，你只有在上面的选项“DOS FAT fs support”选 Y，它才会工作。如果不清楚，选 Y。

选 m 编译为模块，名为 vfat。

- Default codepage for FAT

默认 FAT 代码页

这个选项要根据你的 FAT 文件系统来设置，它能够通过“codepage”挂载选项来修改。

- Default iocharset for FAT

FAT 默认 IO 字符设置

这里设置你想要 FAT 使用的默认 IO 字符。它需要与你的 FAT 文件系统相匹配，能够通过“iocharset”挂载选项来修改。注意“utf8”不推荐于 FAT 文件系统。如果不清楚，你不要在这设置“utf8”。

11.18.3 NTFS file system support

NTFS 文件系统支持

NTFS 是 Microsoft Windows NT, 2000, XP 和 2003 使用的文件系统。

选 `y` 或者 `M`，允许你进行读操作。也会提供部分的，但是安全的写操作支持。要进行写操作，你需要在下面的选项“NTFS write support”选 `Y`。

这里也提供很多用户空间工具，称为 `ntfsprogs`。里面包含 `ntfsundelete` 和 `ntfsresize`，能够在内核没有 NTFS 的支持下工作。

这是 Linux NTFS 支持的草稿的重写，它代替了老的 Linux 2.5.11 开始使用的 NTFS 代码。对于 Linux 2.4 内核系列的后台接口，在我们的工程网站上作为补丁包单独提供。

选 `m` 编译为模块，名为 `ntfs`。如果除了 Linux 之外，你不使用 Windows NT, 2000, XP 或者 2003，那么选 `N`。

- NTFS debugging support

NTFS 调试支持

如果你在试验 NTFS 文件系统的各种问题，在这选 `Y`。这将导致进行额外的一致性检查，这由驱动程序执行，并做为额外的调试信息写入系统日记。注意这些调试信息默认是禁止的。要开启它，需要在内核启动时在内核命令行明确 `debug_msgs=1`，或者做为模块插入的选项，当你载入 `ntfs` 模块时。当这个驱动激活时，你可以以超级用户身份进行信息调试：`echo 1 > /proc/sys/fs/ntfs-debug`

把 `1` 改为 `0`，将禁止调试信息。

如果你禁止调试信息，这只会造成小小的开销；如果你开启调试信息，将会严重影响系统的性能。

当要报告 BUG 时，请尝试将电脑中的关于错误发生的完整调试信息不经处理转移出来。

- NTFS write support

NTFS 写支持

这里允许对 NTFS 驱动器部分的，但是安全的写支持。

唯一的支持操作是修改已经存在的文件，并且不更改文件的大小。它不会建立新的文件或者目录，也不会删除或者重命名。注意只有非常驻的文件能够被写入，所以你会发现一些非常小的文件（小于 500 字节左右）将不会被写入。

我们不能保证它一定不会损伤任何数据，但是目前我们也没有接到任何单独的报告，反映驱动器遭到了破坏。所以我们认为使用它是非常安全地。

注意：这个版本的写支持是安全的（是 NTFS 支持草稿的重写），但是你要注意老的 NTFS 写支持，包括 Linux 2.5.10 和更早版本的（始于 1997 年），是不安全的。

现在对于 TopologiLinux 比较有用。TopologiLinux 运行于 DOS/Microsoft Windows 系统之上，而不分区你的硬盘。与其他 Linux 发行版本不同的是，它不需要自己的分区。

在这选 `n` 是最安全的。

11.19 Pseudo filesystems

伪文件系统

11.19.1 /proc file system support

/proc 文件系统支持

这是一个虚拟的文件编译，它提供系统状态信息。“虚拟”意味着，它不占用你硬盘上的空间：文件是在你试图访问的时候，由内核动态建立。同样，你无法使用老版本的 LESS 来阅读这些文件，你需要 `more` 或者 `cat`。

这非常酷：比如：“`cat /proc/interrupts`”提供目前正在使用的不同的 IRQ 信息（在你的计算机中有一小部分的中断命名序列，它们被相关联的设备使用，以获取 CPU 的注意——通常是资源的争用，如果两个设备错误地配置为使用同样的中断）。`procinfo` 程序用来显示你系统从 /proc 文件系统中获取的信息。

在你使用 /proc 文件系统前，你需要挂载它，意思是它需要有一个目录结构的位置。这个位置应该是 /proc。命令如下：“`mount -t proc proc /proc`”或者在 /etc/fstab 中等价的行也可以工作。

这个选项将使你的内核增大 67 KB。一些程序依赖这个功能，所以每个人应该在这选 `Y`。

- /proc/kcore support
/proc/kcore 支持
- /proc/vmcore support
/proc/vmcore 支持

11.19.2 Virtual memory file system support

虚拟内存文件系统支持

Tmpfs 是一个把所有文件保持在虚拟内存里面的文件系统。

所有在 tmpfs 里的都是临时的，意味着没有文件会在你的硬盘中建立。这些文件在内存和 swap 空间中存在。如果你卸载 tmpfs 实例，所有储存在里面的东西将会消失。

- Tmpfs POSIX Access Control Lists
Tmpfs POSIX 访问控制清单

POSIX 访问控制清单(ACLs) 通过 owner/group/world 方案支持用户和群组许可。

如果你不知道什么是 POSIX 访问控制清单，选 N。

11.19.3 HugeTLB file system support

HugeTLB 文件系统支持

hugetlbfs 文件系统是 HugeTLB 页的后台支持，基于 ramfs。

如果不清楚，选 N。

11.19.4 Userspace-driven configuration filesystem

被动用户空间配置文件系统

configs 一个基于 ram 的文件系统，它提供与 sysfs 相对应的功能。sysfs 是一个查看内核对象的文件系统，而 configs 则是一个管理内核对象的文件系统，或者配置系统。

sysfs 和 configs 两者能够并且应该在一个系统中同时存在。一个无法代替另一个工作。

11.20 Miscellaneous filesystems

多种文件系统

11.20.1 ADFS file system support

ADFS 文件系统支持

Acorn（橡果）磁盘填充系统是 RiscOS 操作系统的标准文件系统，它在 Acorn 基于 ARM 的 Risc Pc 系统上运行，也运行在 Acorn 阿基米德范围内的机器上。如果你选 Y，Linux 将可以阅读硬盘上和软盘上的 ADFS 分区及格式。如果你想要写这些设备，在下面的“ADFS write support”选 Y。

ADFS 分区应该是你的第一个分区。

选 m 编译为模块，名为 adfs。

如果不清楚，选 N。

- ADFS write support

ADFS 写支持（危险）

如果你选 Y，你将可以写硬盘和软盘上的 ADFS 分区及格式。这是试验性的代码，所以你要是不清楚，选 N。

11.20.2 Amiga FFS file system support

Amiga FFS 文件系统支持

快速文件系统(FFS) 是 amiga(tm)操作系统上使用的通用文件系统，它从 AmigaOS V1.3 (34.20)开始使用。选 Y，如果你想要读写你硬盘上的 Amiga FFS 分区。Amiga 软盘无法用这个驱动来访问，因为它的软盘标准与 PC 和工作站相比，是另一个不兼容的标准。

这个驱动同样能让你挂载 Bernd Schmidt 使用的 Un*X Amiga Emulator 文件系统。

如果你想要这么做，你需要在上面的“Loop device support”选 Y 或者 M。

选 m 编译为模块，名为 affs。如果不清楚，选 N。

11.20.3 eCrypt filesystem layer support

eCrypt 文件系统层支持

加密文件系统在 VFS 层上进行操作。

选 m 编译为模块，名为 ecryptfs。

11.20.4 Apple Macintosh file system support

苹果文件系统支持

如果你选 Y，你可以挂载苹果格式的软盘和硬盘分区，拥有完全的读写权限。

选 m 编译为模块，名为 hfs。

11.20.5 Apple Extended HFS file system support

苹果扩展 HFS 文件系统支持

如果你选 Y，你可以挂载扩展的苹果格式硬盘分区，拥有完全的读写权限。

这个文件系统通常称为 HFS+，在 MacOS 8 中推荐使用。它包含所有的 Mac 专有文件系统数据，比如数据叉和建立者代码，但是它同样有一些 UNIX 风格的特性，比如文件拥有者和权限。

11.20.6 BeOS file system (BeFS) support

eOS 文件系统(BeFS) 支持 (只读)

BeOS 文件系统(BeFS) 是 Be, Inc's BeOS 的本地文件系统。对于文件和目录的任意属性，以及数据库的索引选择，都有着卓越的表现（也要注意，这个驱动不会包含上述的特性）。它是 64 位的文件系统，所以它支持相当大的卷标和文件。

如果你使用这个文件系统，你需要在下面的至少一个 NLS（本地语言支持）中选 Y。

如果不清楚，选 N。

选 m 编译为模块，名为：befs。

- Debug BeFS

BeFS 调试

如果你选 Y，你可以使用挂载选项 'debug' 来开启对于驱动器的调试。

11.20.7 BFS file system support

BFS 文件系统支持

启动文件系统(BFS) 是在 SCO UnixWare 下使用的文件系统，它允许启动器在系统启动时访问内核镜像和其他的重要文件。它通常被挂载于 /stand，相当于 UnixWare 分区下的“STAND”片标记。如果你想要读写位于 /stand 片下的文件，你需要选 Y。你同样要在下面的选项“UnixWare slices support”中选 Y。

如果不清楚，选 N。

选 m 编译为模块，名为： bfs。注意你的启动分区文件系统（包含/目录）不能被编译成模块。

11.20.8 EFS file system support

EFS 文件系统支持（只读）

EFS 是在 non-ISO9660 CD-ROM 和硬盘上使用的老的文件系统，它用于 SGI's IRIX 操作系统下（ IRIX 6.0 以及更新的使用 XFS 文件系统）。

这里仅仅提供只读访问。如果你不清楚，选 N 是安全的。

选 m 编译为模块，名为： efs。

11.20.9 Journalling Flash File System v2 (JFFS2) support

日志闪存文件系统 v2 支持(JFFS2)

JFFS2 是第二代的日志闪存文件系统，它用于无磁盘的设备。它提供改良的保持层、压缩和硬连接支持。你不能使用于普通的块设备，仅能用于'MTD'设备。

- JFFS2 debugging verbosity (0 = quiet, 2 = noisy)

JFFS2 调试冗长（0 = 安静，2 = 喧闹）

这里控制 JFFS2 代码生成的大量调试信息。对于商用系统，设置为 0。对于鉴定、测试和调试，设置为 1 是明智的。这将会允许一些主张的功能，也将在 KERN_DEBUG 日志等级上打印调试信息，平时这些信息是不可见的。设置为 2 看起来不是很有用 - 它允许在特定的区域开展额外的调试工作，当某个点需要调试的时候。但是当错误被定位和修复后，详细的信息将归类于第 2 等级。

如果要报告错误，请试着将完整的信息不加修改地第 1 等级下转移出来，当错误发生的时候。

- JFFS2 write-buffering support

JFFS2 写缓冲支持

这里允许 JFFS2 写缓冲支持。

这个功能在以下几种闪存设备类型中，需要这个功能：

- NAND 闪存
- NOR 闪存（带有透明的 ECC）
- 数据闪存
- Verify JFFS2 write-buffer reads

验证 JFFS2 写缓冲阅读

这使得 JFFS2 返回阅读通过写缓冲写入的页，并检查错误。

- JFFS2 summary support

JFFS2 摘要支持

这个特性能够使用摘要信息，以便更快地挂载。

摘要信息能够嵌入一个文件系统镜像中，通过使用实用工具'sumtool'。

如果不清楚，选 N。

- JFFS2 XATTR support

JFFS2 扩展属性支持

扩展属性是名字：值的配对属性，与内核或者用户的节点关联。

如果不清楚，选 N。

- Advanced compression options for JFFS2

JFFS2 高级压缩选项

这个选项允许你明确地选择在 JFFS2 使用哪个压缩模块，如果有的话。移除压缩罐，意味着你不能阅读已经存在的文件系统，允许试验性的压缩罐，意味着你写入的文件系统不能够被标准的内核阅读。

如果不清楚，你应该明确选 N。

11.20.10 Compressed ROM file system support

压缩 ROM 文件系统支持(cramfs)

选 Y，将包括对压缩 ROM 文件系统支持(cramfs)的支持。cramfs 是一个简易的、小巧的、压缩的文件系统，主要用在基于嵌入 ROM 的系统上。CramFs 是只读的，受 256MB 文件系统限制（含 16MB 文件），并且不支持 16/32 位的 uid/gid、硬连接和时间戳。

选 m 编译为模块，名为：cramfs。注意启动文件系统（包含/目录）不能够被编译为模块。

如果不清楚，选 N。

11.20.11 FreeVxFS file system support

FreeVxFS 文件系统支持

FreeVxFS 文件系统支持 VERITAS VxFS(TM)文件系统格式。VERITAS VxFS(TM)是SCO UnixWare (或者可能有其他)的标准文件系统,并且是 Sunsoft Solaris, HP-UX 和其他操作系统的可选项。目前这里仅能够提供只读访问。

注意:文件系统类型,通过 mount(1), mount(2) 和 fstab(5)使用的是'vxfs',它被描述为文件系统格式,不是实际的驱动。

选 m 编译为模块,名为: freevxfs。如果不清楚,选 N。

11.20.12 Minix file system support

Minix 文件系统支持

Minix 是一个简单的操作系统,在很多阶层使用。minix 文件系统(设计来组织硬盘分区或者软盘上的文件)曾经是 Linux 的常规文件系统,但是已经被第二扩展文件系统 ext2fs 取代。你不需要在你的硬盘上使用 minix 文件系统,因为在编译嵌入时的限制,但是有时在老的 Linux 软盘上会有这个东西。这个选项会使你的内核增大28 KB。如果不清楚,选 N。

选 m 编译为模块,名为: minix。注意启动文件系统(包含/目录)不能够被编译为模块。

11.20.13 OS/2 HPFS file system support

OS/2 HPFS 文件系统支持

OS/2 是 IBM 在 PC 上的操作系统,如同 Warp, HPFS 文件系统是用来组织 OS/2 硬盘分区上的文件的。选 Y,如果你想要读写你硬盘 OS/2 HPFS 分区上的文件。OS/2 软盘采用了常规的 MSDOS 格式,所以你不需要这个选项来阅读它。

选 m 编译为模块,名为: hpfs。如果不清楚,选 N。

11.20.14 QNX4 file system support

QNX4 文件系统支持(只读)

这是用在实时操作系统 QNX 4 和 QNX 6 上的文件系统(后者也叫 QNX RTP)。

选 Y,如果你想要挂载 QNX 硬盘或者软盘。你只能以只读方式访问文件系统,除非你在下面的“QNX4FS read-write support”选项中选 Y。

选 m 编译为模块,名为: qnx4。

如果你不知道你是否需要,那说明你不需要它,选 N。

11.20.15 ROM file system support

ROM 文件系统支持

这是一个非常小的只读文件系统，主要目的是初始化安装磁盘的 ram 磁盘，但是它同样能够用于其他的多媒体只读环境中。

选 m 编译为模块，名为：romfs。注意启动文件系统（包含/目录）不能够被编译为模块。

如果你不知道你是否需要，那说明你不需要它，选 N。

11.20.16 System V/Xenix/V7/Coherent file system support

System V/Xenix/V7/Coherent 文件系统支持

SCO, Xenix and Coherent 是针对 Intel 机器的商业 Unix 系统。它的第 7 个版本使用于 DEC PDP-11。选 Y，你将可以阅读它们的硬盘分区和软盘。

如果你有上述的硬盘分区和软盘，那么它们很可能也包含有二进制文件。为了运行这些二进制文件，你需要安装 linux-abi，它是内核模块的组件之一。它能够使你在 Linux 下运行 SCO, Xenix, Wyse, UnixWare, Dell Unix 和 System V 程序。

注意：那只能在基于 Intel 的机器上运行相应程序。使用 PDP 的估计要等到某人把 Linux 开发到第 11 个版本才能运行了;-)

如果你仅仅是需要挂载网络某处的 UNIX 文件，它使用 NFS 文件系统，那么你不需要 System V 文件系统支持（但是你需要 NFS 文件系统支持，显然地）。

注意在软盘中通常不需要这个选项，因为有非常好的移植方法，在各 UNIX 系统（甚至其他系统）中转移文件和目录，这通过 tar 程序来完成（“man tar”或者尽量“info tar”）。同时也要注意这个选项和上面的“System V IPC”选项没有任何关系。

选 m 编译为模块，名为：sysv。

如果你没有听说过上面的东西，选 N 是安全的。

11.20.17 UFS file system support

UFS 文件系统支持

BSD 和 Unix 衍生物（如 SunOS, FreeBSD, NetBSD, OpenBSD 和 NeXTstep）使用的文件系统叫做 UFS。一些 System V Unix 也能够使用这种文件系统建立和挂载硬盘分区和软盘。选 Y，你将可以阅读这些分区。如果你想要改写它，那么在下面的试验性的选项“UFS file system write support”中选 Y。

目前 UFS2 发行版本的变体（用于 FreeBSD 5.x）仅提供只读支持。

如果你仅仅是需要挂载网络某处的 UNIX 文件，它使用 NFS 文件系统，那么你不需要 UFS 文件系统支持（但是你需要 NFS 文件系统支持，显然地）。

注意在软盘中通常不需要这个选项，因为有非常好的移植方法，在各 UNIX 系统（甚至其他系统）中转移文件和目录，这通过 tar 程序来完成（“man tar”或者尽量“info tar”）。

当你要访问 NeXTstep 文件时，你可能需要把它们从 NeXT 字符模式转化为拉丁字符模式，使用程序 recode（“info recode”）达到以上目的。

选 m 编译为模块，名为：ufs。

- UFS file system write support

UFS 文件系统写支持（危险）

选 Y，如果你想要写 UFS 分区。这是试验性的，所以你应该事先备份你的 UFS 分区。

- UFS debugging

UFS 调试

如果你正在试验 UFS 文件系统的任何问题，选 Y。这将生成很多额外的调试信息写入系统日志。

11.21 Network File Systems

网络文件系统

选 Y，以便看到关于网络文件系统的选项，以及与之相关的代码，比如 NFS 守护进程和 PRCSec 安全模块。

这个选项本身不会增加任何内核代码。

如果你选 N，所有的子选项将被跳过和禁用；如果不清楚，选 Y。

11.21.1 NFS file system support

NFS 文件系统支持

如果你连接到其他的（通常是本地的）Unix 电脑（使用 SLIP, PLIP, PPP 或者 Ethernet），并且想要挂载在那台主机上的文件，而它使用了网络文件共享协议（NFS 主机），那么在这选 Y。“挂载文件”意味着客户端可以通过通常的 UNIX 命令来访问文件，就如同它们是位于客户端的硬盘上一样。为使之工作，服务器必须安装有程序 `nfsd` 和 `mountd` （但是它的内核中不需要有 NFS 文件系统支持）。

Coda 文件系统比 NFS 更优秀，但是使用面没那么广。看下面的选项“Coda file system support”。

如果你在这选 Y，你应该已经在上面的 TCP/IP 网络中选了 Y。这个选项会让你的内核增大 27KB。

选 m 编译为模块，名为：NFS。

如果你在配置无盘的机器，它将在启动时挂载 NFS 文件系统，那么在这里，上面的“Kernel level IP autoconfiguration”，下面的“Root file system on NFS”都要选 Y。在这种情况下，你不能将这个驱动编译为模块。

如果不清楚，选 N。

- Provide NFSv3 client support

提供 NFSv3 客户端支持

选 Y，如果你想要你的 NFS 客户端能够发出第 3 版本的 NFS 协议。

- Provide client support for the NFSv3 ACL protocol extension

提供 NFSv3 ACL 扩展协议客户端支持

实行 NFSv3 ACL 扩展协议来操控 POSIX 访问控制清单。服务器也应该编译有 NFSv3 ACL 扩展协议；请查看 `CONFIG_NFSD_V3_ACL` 选项。

如果不清楚，选 N。

- Provide NFSv4 client support

提供 NFSv4 客户端支持

选 Y，如果你想要你的 NFS 客户端能够发出第 4 版本的 NFS 协议。

如果不清楚，选 N。

- Allow direct I/O on NFS files

允许 NFS 文件直接 I/O

这个选项允许程序使用 `O_DIRECT` `open()` 标记，在 NFS 文件系统中不通过缓冲直接 I/O。当 `O_DIRECT` 设置给一个文件时，它的数据将不会通过系统缓冲页进行缓冲。数据将直接与用户级别的缓存进行沟通。与本地磁盘文件系统不同的地方是，NFS `O_DIRECT` 没有校准限制。

除非你的程序是为使用 `O_DIRECT` 设计的，你最好不要让 NFS 客户端来帮你管理数据缓冲。滥用 `O_DIRECT` 会导致糟糕的服务器表现或者网络风暴。这个内核选项默认是关闭的，以防系统管理员不小心更改潜在的危险的特性。

如果不清楚，选 N。这会减少 NFS 客户端的尺寸，同时让 `open()` 返回 `EINVAL`，如果一个 NFS 文件被以 `O_DIRECT` 标记打开。

11.21.2 NFS server support

NFS 服务器支持

如果你想要你的 Linux 机器做为 NFS 服务器，以便其他在局域网的机器能够通过 NFS 透明地访问你机器上的指定目录，那么你有两个选择：你可以使用自包容用户空间程序 `nnfsd`，这样你就可以在这选 N；或者在这选 Y，使用基于内核的 NFS 服务器。基于内核的好处是，它更快。

如果你选 Y，你将得到第 2 版本的 NFS 协议支持(NFSv2)，如果你想要 NFSv3，在下面的问题中也选 Y。

选 m 编译为模块，名为 `nfdsd`。如果不清楚，选 N。

- Provide NFSv3 server support

提供 NFSv3 服务器支持

如果你除了 NFSv2，还想要 NFSv3 服务器，在这选 Y。如果不清楚，选 Y。

- Provide server support for the NFSv3 ACL protocol extension

提供 NFSv3 ACL 扩展协议服务器支持

实行 NFSv3 ACL 扩展协议来在导出的文件系统上操控 POSIX 访问控制清单。客户端也应该编译有 NFSv3 ACL 扩展协议；请查看 CONFIG_NFSD_V3_ACL 选项。

如果不清楚，选 N。

- Provide NFSv4 server support

提供 NFSv4 服务器支持

如果除了 NFSv2 和 NFSv3，你还要 NFSv4 服务器，在这选 Y。这个特性是试验性的，只有你对测试 NFSv4 感兴趣时才使用。如果不清楚，选 N。

- Provide NFS server over TCP support

通过 TCP 提供 NFS 服务器支持

如果你想你的 NFS 服务器支持 TCP 连接，在这选 Y。

TCP 通常表现得比默认的 UDP 要好一些，在网络有损耗和拥挤时。如果不清楚，选 Y。

11.21.3 Support for rpcbind versions 3 & 4

对 rpcbind 第 3 和 4 版本的支持

通过第 3 和 4 版本的 rpcbind 协议，提供内核询问 rpcbind 服务器的支持。内核将自动回用第 2 版本，如果远程的 rpcbind 服务不支持第 3 和 4 版本。

如果不清楚，选 N 来得到传统的表现（仅为第 2 版本的 rpcbind 响应）。

11.21.4 Secure RPC: Kerberos V mechanism

安全 RPC：Kerberos V 结构

提供安全 RPC 调用，依靠基于 Kerberos V5 的 gss-api 结构。NFSv4 需要这个功能。

如果不清楚，选 N。

11.21.5 Secure RPC: SPKM3 mechanism

安全 RPC：SPKM3 结构

提供安全 RPC 调用，依靠基于 SPKM3 公共钥匙的 gss-api 结构。

如果不清楚，选 N。

11.21.6 SMB file system support

SMB 文件系统支持（废弃的，请用 CIFS）

SMB（信息块服务）是 Workgroups(WfW), Windows 95/98, Windows NT 和 OS/2 网络管理版使用的协议窗口，用来在局域网中共享文件和打印机。选 Y，你可以挂载它们的操作系统（在文字上通常叫做“共享”），并且可以像访问其他 Unix 目录一样访问它们。现在，这仅仅在 Windows 机器使用 TCP/IP，而不是 NetBEUI，作为底层传输协议时使用。

注意：如果你仅仅想要你的机器作为一个 SMB 服务器，为 Windows 客户端（它们需要有 TCP/IP 栈）提供文件和打印机服务，你不需要在这选 Y；你可以使用程序 SAMBA 来实现。

选 m 编译为模块，名为 smbfs。大多数人可以选 N。

- Use a default NLS

使用默认的 NLS

选 Y，会使 smbfs 使用默认的 NLS 转换。你需要在 nls 的设置中明确说明本地的字符设置(CONFIG-NLS-DEFAULT)，同时你需要在 CONFIG-SMB-NLS-REMOTE 明确 SMB 服务器的默认 nls。

nls 的设置能够在挂载的时候更改，如果你的 smbmount 支持的话，可以使用 codepage 和 iocharset 参数来实现。

samba 2.2.0 或更高版本中的 smbmount 支持这个功能。

11.21.7 CIFS support

CIFS 支持（高级网络文件系统，优于 SMBFS）

这是 VFS 客户端模块，支持通用网络文件系统（CIFS）协议，它是服务器信息块协议（SMB）的替代者，是本地文件共享构架，用于很多早期的 PC 操作系统。CIFS 协议和 Samba（为 Linux 和其他操作系统提供优秀的 CIFS 服务器支持）一样被文件服务器完全支持，比如 Windows 2000（包含 Windows 2003, NT 4 和 Windows XP）。对于 OS/2 和 Windows ME 以及类似的服务器提供有限支持。

cifs 模块提供一个高级的文件系统客户端，以挂载到 CIFS 适用的服务器。它支持 DFS（分层命名空间），通过 Kerberos 或 NTLM 或 NTLMv2 建立安全用户会话，安全发布缓冲(oplock)，可选包标记，Unicode 和其他国际化的进步。

如果你需要从这台机器挂载 Samba 或 Windows，选 Y。

- CIFS statistics

CIFS 统计数据

开启这个选项，将会生成统计数据。通过 cifs 客户端挂载的每个共享服务器信息将显示在 /proc/fs/cifs/Stats。

- Support legacy servers which use weaker LANMAN security

使用较低的 LANMAN 安全继承服务器支持

现代的 CIFS 服务器包含有 Samba 和大多数 Windows 版本（始于 1997 年），支持更严格的 NTLM（甚至是 NTLMv2 和 Kerberos）安全构架。比起老的使用 SMB 协议的 LANMAN 版本，它们将密码切割得更安全。但是 LANMAN 基础下的验证是需要的，以便与老的 SMB 服务器建立会话。

开启这个选项，将允许 cifs 模块挂载到基于老 LANMAN 的服务器，比如 OS/2

和 Windows 95，但是这些挂载的安全性将会降低，比起使用 NTLM 或者目前更加安全的构架，如果你在公共网络的话。除非你非常需要访问老的 SMB 服务器（并且在私有网络上），你应该选 N 比较好。即使这个支持在内核构建中提供，LANMAN 验证也不会自动使用。在系统运行时，LANMAN 挂载是禁止的，但你可以在 `/proc/fs/cifs` 设置为允许（或可选），也可以通过挂载命令选项来实现。这个支持默认是关闭的，目的是为了减少下层的潜在攻击。

如果不清楚，选 N。

- CIFS extended attributes

CIFS 扩展属性

扩展属性是名字：值的配对属性，与内核或者用户的内节点关联。CIFS 通过扩展属性起始的 SMB/CIFS EA 前缀的用户命名空间来映射命名。EA 在 Windows 服务器上储存，它没有用户命名空间前缀，但是它们的名字在 Linux cifs 客户端看来是以用户命名空间前缀起头的。系统命名空间（一些文件系统使用来储存 ACL）在这个时候不被支持。

如果不清楚，选 N。

- Enable additional CIFS debugging routines

允许额外 CIFS 调试例行程序

开启这个选项，将给 cifs 代码增加一些调试例行程序，这将会轻微地增加 cifs 模块的尺寸，也可能导致额外的调试信息记录导出到一些错误的路径，并减慢运行速度。这个选项能够关闭，除非你在调试 cifs 问题。如果不清楚，选 N。

- CIFS Experimental Features

CIFS 试验性的特性

允许 CIFS 试验性的特性。这些特性是试验性的，目前包含 DFS 支持和目录更改通告 ie `fcntl(F_DNOTIFY)`，也包含上层构架调用，这用于 Kerberos 协商会话和 uid 重映射。一些特性依赖于在伪文件 `/proc/fs/cifs/Experimental` 中设置值为 1（默认为关闭）。如果不清楚，选 N。

11.21.8 NCP file system support

NCP 文件系统支持 (挂载 NetWare 卷标)

NCP (NetWare 核心协议)是 IPX 上使用的协议, Novell NetWare 客户端用来与文件服务器对话。它对于 IPX 就如同 NFS 对于 TCP/IP, 如果这么解释有用的话。在这选 Y, 你可以挂载 NetWare 文件服务卷标, 并且像访问其他 Unix 目录一样访问它们。

如果你想要你的 Linux 机器作为 Novell NetWare 文件服务器, 你不需要在这选 Y。

选 m 编译为模块, 名为: ncpfs。选 N, 除非你要连接 Novell 网络。

- Packet signatures

包签名

NCP 允许包签名, 以确保高级别安全性。如果你需要安全, 选 Y。普通用户可以不管它。为了使用包签名, 你必须使用 2.0.12 版本以上的 ncpfs。

- Proprietary file locking

专有文件锁定

允许锁定远程卷标的结果。选 N, 除非你有特别的程序, 它利用这个锁定方案。

- Clear remove/delete inhibit when needed

当需要时清除移动/删除限制

允许操作文件移动/删除限制标记。要使用这个特性, 你需要在挂载卷标时用上 ncpmount 参数“-s” (ncpfs-2.0.12 和更高版本)。选 Y, 除非你不用 -f 444 参数挂载卷标。

- Use NFS namespace if available

如果可行, 使用 NFS 命名空间

允许你在 NetWare 服务器上使用 NFS 命名空间。它给你大小写敏感的文件名。选 Y。你可以在挂载时用参数“-N nfs”关闭它。

- Use LONG (OS/2) namespace if available

如果可行, 使用长命名空间(OS/2)

允许你 NetWare 服务器上使用 OS2/LONG 命名空间。在这个命名空间里文件名被限制为 255 个字符, 它们是大小写不敏感的, 大小写名字是保留的。选 Y。你可以在挂载时用参数“-N os2”关闭它。

- Lowercase DOS filenames

小写 DOS 文件名

如果你在这选 Y, 在 NetWare 服务器卷标上的每个文件名, 包括使用 OS2/LONG 命名空间的, 在 DOS 下建立的, 或者使用 DOS 命名空间的, 将会转化为小写字母。

选 N，那些文件名将会以大写字母形式给你。

这仅仅是一个装饰用的选项，因为 OS2/LONG 命名空间是不区分大小写的。提供这个选项唯一的主要理由是给予逆向的兼容性，当你从 DOS 移动到 OS2/LONG 命名空间时。长文件名（Win95 建立）不会受影响。

这个选项不能解决 Linux 和 Windows 下的文件名不同的问题。因为 Windows 在客户端做了一个额外的转换。你可以通过在“Allow using of Native Language Support”选项中选 Y 来达到类似的目的。

- Use Native Language Support

使用本地语言支持

允许你使用代码页和 I/O 字符设置，在服务器文件系统和输入/输出之间翻译文件名。这是非常有用的，如果你想要用其他操作系统访问服务器，比如 Windows 95。看 NLS 取得更多信息。

你需要使用 ncpfs-2.2.0.13 或更高版本才能选择代码页和 I/O 字符设置。

- Enable symbolic links and execute flags

允许符号连接和执行标记

这里允许在 NCPFS 使用符号连接和执行许可字节。文件服务器不需要长命名空间，或者载入 NFS 命名空间来使之运行。

要使用新特性，推荐在 ncpmount 命令行使用标记 '-f 600 -d 755'。

11.21.9 Coda file system support

Coda 文件系统支持（高级网络文件系统）

Coda 是一个高级网络文件系统。它与 NFS 文件系统类似，可以让你挂载远程服务器的文件系统，并且使用常规的 Unix 命令访问它们，就如同它们好像安装在你的硬盘上一样。Coda 在一些地方比 NFS 更有优势：支持离线操作（比如笔记本），读/写服务器响应，验证和加密的安全模式，持久客户端缓存和写反馈缓存等。

如果你选 Y，你的 Linux 机器将可以做为 Coda 客户端。你同时需要客户端和服务端的用户级别的代码。服务器目前是用户级别的，也就是说，不需要内核支持。

选 m 编译为模块，名为 coda。

- Use 96-bit Coda file identifiers

使用 96 位 Coda 文件验证器

这是一个新的内核 - 用户空间 API，推荐在 Coda v6.0 使用，为了在新的领域支持更大的 128 位文件验证。

不过这个新的 API 不兼容老的客户端。如果你真的需要运行老的 Coda 用户空间缓存管理器，那么选 Y。

在很多情况下，你都可以选 N。

11.21.10 Andrew File System support (AFS)

Andrew 文件系统支持

如果你选 Y，你将得到试验性的 Andrew 文件系统驱动。它目前只支持非安全的只读 AFS 访问。

如果不清楚，选 N。

- AFS dynamic debugging

AFS 动态调试

选 Y，以使运行时的控制调试信息可见。

如果不清楚，选 N。

11.22 Partition Types

分区类型

11.22.1 Advanced partition selection

高级分区选择

选 Y，如果你想要在 Linux 下使用由别的操作系统分区的硬盘，它使用了其他的构架系统。

注意这个选项的选择不会直接影响内核：选 N 只会让配置器跳过所有的外来分区方案的选项。

如果不清楚，选 N。

11.23 Native Language Support

本地语言支持

基本的本地语言支持。很多文件系统依赖它（比如 FAT、JOLIET、NT、BEOS 文件系统），有一些文件系统的功能需要使用它（比如 NCP、SMB）。

如果不清楚，选 Y。

选 M 编译为模块，名为：nls_base。

- Default NLS Option

默认语言选项

当挂载文件系统时默认使用的 NLS。注意，这是你的控制台使用的 NLS，而不是特定文件系统储存数据（文件名）于磁盘时使用的 NLS（如果不同）。

目前，合法的值有：

big5, cp437, cp737, cp775, cp850, cp852, cp855, cp857, cp860, cp861, cp862, cp863, cp864, cp865, cp866, cp869, cp874, cp932, cp936, cp949, cp950, cp1251, cp1255, euc-jp, euc-kr, gb2312, iso8859-1, iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7, iso8859-8, iso8859-9, iso8859-13, iso8859-14, iso8859-15, koi8-r, koi8-ru, koi8-u, sjis, tis-620, utf8.

如果你指定了一个错误值，那么将会使用内建的 NLS，它与 iso8859-1 兼容。

如果不清楚，指定它为“iso8859-1”。

- Codepage 437 (United States, Canada)

代码页 437：美国、加拿大

微软的 FAT 文件系统家族用内部的语言字符设置来处理文件名。这些字符设置储存在所谓的 DOS 代码页中。你需要包含恰当的代码页，如果你想要在 DOS/Windows 分区上正确地读/写这些文件名。这些代码页仅仅是适用于文件名，而不是文件的内容。

你可以包含多种代码页；在这选 Y，如果你想要包含 DOS 代码页 437，它使用在美国和加拿大的部分地区。这也是建议值。

- Codepage 737 (Greek) 希腊
- Codepage 775 (Baltic Rim) 波罗的海
- Codepage 850 (Europe) 欧洲
- Codepage 852 (Central/Eastern Europe) 中/东欧
- Codepage 855 (Cyrillic) 西里尔
- Codepage 857 (Turkish) 土耳其
- Codepage 860 (Portuguese) 葡萄牙
- Codepage 861 (Icelandic) 冰岛语
- Codepage 862 (Hebrew) 希伯来
- Codepage 863 (Canadian French) 法属加拿大
- Codepage 864 (Arabic) 阿拉伯
- Codepage 865 (Norwegian, Danish) 挪威，丹麦
- Codepage 866 (Cyrillic/Russian) 西里尔/俄语
- Codepage 869 (Greek) 希腊
- Simplified Chinese charset (CP936, GB2312) 简体中文
- Traditional Chinese charset (Big5) 繁体中文

- Japanese charsets (Shift-JIS, EUC-JP) 日语
- Korean charset (CP949, EUC-KR) 韩国语
- Thai charset (CP874, TIS-620) 泰国语
- Hebrew charsets (ISO-8859-8, CP1255) 希伯来字符设置
- Windows CP1250 (Slavic/Central European Languages) 斯拉夫/中欧语言
- Windows CP1251 (Bulgarian, Belarusian) 保加利亚，白俄罗斯
- ASCII (United States) 美国
- NLS ISO 8859-1 (Latin 1; Western European Languages) 拉丁 1；西欧
- NLS ISO 8859-2 (Latin 2; Slavic/Central European Languages) 拉丁 2；斯拉夫/中欧语言
- NLS ISO 8859-3 (Latin 3; Esperanto, Galician, Maltese, Turkish) 拉丁 3；世界语，加里西亚，马尔他，土耳其
- NLS ISO 8859-4 (Latin 4; old Baltic charset) 拉丁 4：老波罗的海
- NLS ISO 8859-5 (Cyrillic) 西里尔
- NLS ISO 8859-6 (Arabic) 阿拉伯
- NLS ISO 8859-7 (Modern Greek) 现代希腊
- NLS ISO 8859-9 (Latin 5; Turkish) 拉丁 5：土耳其
- NLS ISO 8859-13 (Latin 7; Baltic) 拉丁 7：波罗的海
- NLS ISO 8859-14 (Latin 8; Celtic) 拉丁 8：凯尔特语
- NLS ISO 8859-15 (Latin 9; Western European Languages with Euro) 拉丁 9：西欧语言
- NLS KOI8-R (Russian) 俄语
- NLS KOI8-U/RU (Ukrainian, Belarusian) 乌克兰，白俄罗斯
- NLS UTF8
- NLS UTF8

如果你想要从微软 FAT 文件系统家族或者从 JOLIET CD-ROM 中正确地在显示器上显示文件名，你需要包含恰当的输入/输出字符设置。在这选 Y 包含 UTF-8 编码，用于 Unicode/ISO9646 通用字符设置。

11.24 Distributed Lock Manager (DLM)

发行版锁定管理器 (DLM)

一个通用的发行版锁定管理器，用于内核或者用户空间程序。

Chapter 12

Kernel hacking 内核调试

12.1 Show timing information on printk

在 `printk` 上显示时间信息

选择这个选项，将可以在 `printk` 输出上显示时间信息。这允许你在内核的操作之间测量时间间隔，包括启动操作。这对于标识内核启动时的长延迟非常有用。

12.2 Enable `__deprecated` logic

允许 `__deprecated` 逻辑

在内核编译时允许 `__deprecated` 逻辑。禁止这个选项，将不会得到这个信息：
warning: 'foo' is deprecated (declared at kernel/power/somefile.c:1234)。

12.3 Enable `__must_check` logic

允许 `__must_check` 逻辑、

在内核编译时允许 `__must_check` 逻辑。禁止这个选项，将不会得到这个信息：
warning: ignoring return value of 'foo', declared with attribute warn_unused_result
。

12.4 Warn for stack frames larger than (2048)

警告堆栈帧超过规定数值（需要 `gcc 4.4`）

让 `gcc` 在编译时，在堆栈帧数超过规定数值时发出警告。把这个值设置过低会导致很多的警告。把它设置为零则禁用警告。

需要 `gcc 4.4`。

12.5 Magic SysRq key

魔术键 SysRq

如果你选 Y，你将可以控制整个系统，即使系统已经崩溃。比如在内核调试的时候（你需要清除磁盘缓存，快速重启系统，转移一些状态信息）。这通过按下 SysRq (Alt+PrintScreen)的同时，按不同的一些按键来实现。它在串行的控制台上同样可以工作（至少在 PC 硬件上），如果你发出一个中断，并且在 5 秒内发出按键命令。不要选 Y，除非你知道怎么做。

相关组合键：

摘自：《Linux 新手管理员手册》，我最喜欢的一本关于 Linux 的书，打印出来放在床头看。里面写得非常好，我非常佩服老外的认真，也非常感谢翻译人员。这本书比较老，但是就是老的书才有味道，特别是你想尝试文本控制台的时候，它非常有用。不过人的口味不一样，合适我的未必合适你。有兴趣的可以去网上搜索下载。

< Alt > < SysRq > < command.key >

这是在 Linux 的内核（底层内核）上支持的一组组合键。它意味着这些组合键在大部分情况下都是有效的。组合键主要是针对开发人员的程序调试目的或者在紧急场合下才使用；你也可以使用其他的办法，安全第一。< SysRq > 键在 PC 上指的是 PrintScreen 键。组合键的功能可以通过设置相对应的内核参数来激活或者取消，比如：

```
echo "1" > /proc/sys/kernel/sysrq
```

- < Alt > < SysRq > < k >

终止在当前虚拟终端上运行的所有进程（包括 X 窗口）。这个组合键被称为安全访问键（SAK）。

- < Alt > < SysRq > < e >

发送 Term 信号（结束信号）到除了 init 进程以外的所有运行进程，要求其退出。

- < Alt > < SysRq > < i >

发送 kill（终止信号）到除了 init 进程以外的所有运行进程，这个组合键比上一个更加有效，但是也可能引起一些程序被异常终止。

- < Alt > < SysRq > < l >

发送 kill 命令到所有运行进程（包括 init），系统将不再能够使用。

- < Alt > < SysRq > < s >

在所有的挂载文件系统上运行紧急同步（缓存写）。这可以保护数据丢失。

- < Alt > < SysRq > < u >

以只读模式重新挂载所有已加载的文件系统。这和上面的同步组合键有相同的功能，但是另外有一个好处：如果操作顺利完成，下一次硬件重新启动时，fsck 将不会重新检查所有的文件系统。

- `< Alt > < SysRq > < r >`

关闭键盘的原始模式。当你的 X 窗口程序挂住没有响应时特别有用。使用这个组合键之后，你可能会接着使用 `< Ctrl > < Alt > < Del >` 程序启动。

- `< Alt > < SysRq > < b >`

不进行同步和卸载就立刻重新启动系统。之后你可能会碰到一下错误信息。

- `< Alt > < SysRq > < o >`

关机（如果正确设置以后）。

- `< Alt > < SysRq > < p >`

导出当前的寄存器和标志到控制台。

- `< Alt > < SysRq > < t >`

导出当前任务和相关信息到控制台。

- `< Alt > < SysRq > < m >`

导出内存的内容到控制台。

- `< Alt > < SysRq > < 数字 >`

“数字”从 0 到 9。设置控制台记录的级别，用于控制哪一些内核信息将被显示在控制台上。举例，

“0”意味着只有象 PANIC 和 OOPS 之类的紧急信息才被显示在控制台上。

- `< Alt > < SysRq > < h >`

显示帮助信息。还有，任何其他不被支持的 `< Alt > < SysRq > < key >` 组合键将显示同样的帮助。

12.6 Enable unused/obsolete exported symbols

允许不使用的/废弃的导出符号

不使用的导出符号会使内核不必要地变大。所以很多不使用的导出都被移除了。这个选项主要是为了一些外部内核模块提供过渡，以防它会使用到这些符号。如果你遇到了这样的情况，你需要考虑是否使用了正确的接口（解释：由于没有人在内核模块中使用它，它实际上已经是错误的应用接口）。如果你真的需要这些符号，请你发个电子邮件到 Linux 内核的官方邮箱上，说明你需要哪些符号，为什么需要，以及对于在新的内核中融合这些符号的计划。

12.7 Debug Filesystem

调试文件系统

调试文件系统是一个虚拟的文件系统。内核开发者用来放入要调试的文件。允许这个选项，你将可以读写这些文件。

如果不清楚，选 N。

12.8 Run 'make headers_check' when building vmlinux

在 vmlinux 编译时运行 make headers_check

这个选项在编译内核时解压用户可见的内核头文件，并且进行基本的检查与判断，以确保要解压的文件是用户需要的文件，等等。

如果你正在修改与用户空间相关联的头文件，选 Y，并检查解压到\$(INSTALL_HDR_PATH)（通常在你的编译树 usr/include 中）的头文件，确认它们是否合适。

12.9 Kernel debugging

内核调试

选 Y，如果你正在开发驱动程序，或者尝试调试和鉴定内核问题。

12.9.1 Debug shared IRQ handlers

调试共享的 IRQ 中断处理器

选 Y，在共享的中断处理器被注册后，或者取消注册前，将会生成伪中断。设备应该能够在相关节点处理中断。一些中断需要/不需要发生。

12.9.2 Detect Soft Lockups

探测软件锁定

选 Y，允许内核探测软件锁定，这是一种 BUG，它使内核陷入死循环，在 10 秒以上都不会响应其他的工作任务。

当一个软件锁定被探测到时，内核将会打印出目前的堆栈痕迹（你需要报告这些情况），但是系统将保持锁定。这个选项特性造成微不足道的花销。

注意硬件锁定是另一种类型的 BUG，可以通过 NMI-watchdog 探测到，它在相关支持的平台上使用。

12.9.3 Collect scheduler debugging info

收集调度器调试信息

如果你选 Y ，将会生成 `/proc/sched_debug` 文件，有助你调试调度器。这个选项在运行时的花销是很小的。

12.9.4 Collect scheduler statistics

收集调度器统计信息

如果你在这选 Y ，额外的代码将会嵌入到调度器和相关的标准程序当中，以便收集调度器的行为统计信息，并在 `/proc/schedstat` 文件中提供。这些统计信息可能会在调整和调试调度器时有用。如果你不想调试调度器或者尝试调整一个特别的程序，你可以选 N ，以避免增加一点点小的花销。

12.9.5 Collect kernel timers statistics

收集内核时间器统计信息

如果你在这选 Y ，额外的代码将会嵌入时间器的标准程序，以便收集内核时间器重调的统计信息。这些信息可以通过 `/proc/timer_stats` 阅读。信息统计工作通过向 `/proc/timer_stats` 写入 1 开始，写入 0 结束。这个特性在收集时间器在内核和用户空间工作模式的信息时很有用。这个特性是轻量级的，如果在内核中允许配置，但是不激活它（它默认在系统启动时不激活。仅当一些程序，如 `powertop` ，明确要求时，它才被激活）。

12.9.6 Debug object operations

调试对象操作

如果你选 Y ，额外的代码将加入内核，以追踪各种对象的生存时间，和有效地操作那些对象。

12.10 SLUB debugging on by default

SLUB 默认调试

默认在启动时调试。SLUB 在启动时默认关闭调试功能。选这项相当于在启动时明确 `slub_debug` 参数值。这里没能很好地通过类似 `slub_debug=xxx` 的方式进行调试控制支持。

SLUB 调试可能在内核编译时关闭，通过明确“`slub_debug=-`”来改变 `CONFIG_SLUB_DEBUG_ON` 参数。

12.11 Enable SLUB performance statistics

允许 SLUB 执行统计信息

SLUB 统计信息对于调试 SLUB 分配行为，优化分配器有着良好的作用。这项功能应该不会选上，因为它会让你的分配器效率降低一些百分比。命令：slabinfo 支持判定最为动态的片，以便决定哪些片对特殊的装载有关联。

尝试运行：slabinfo -DA

SLAB：传统的内存管理方式。

SLUB：从 2.6.22 内核开始引入了 SLUB 的内存管理方式，默认代替 SLAB。

12.12 RT Mutex debugging, deadlock detection

RT Mutex 调试，探测死锁定

这里允许自动探测和报告 rt mutex 的语法错误和关联的死锁定。

12.13 Built-in scriptable tester for rt-mutexes

嵌入式 rt-mutexes 脚本测试器

这个选项开启 rt-mutex 测试器。

12.14 Spinlock and rw-lock debugging: basic checks

旋转锁定和读写锁定调试：基本检查

在这选 Y，将会使 SMP 获取遗失的旋转锁定初始值和旋转锁定造成的其他类型的通用错误。这在与 NMI watchdog 结合中大量使用，以便旋转锁定造成的死锁定也能被调试。

12.15 Mutex debugging: basic checks

Mutex 调试：基本检查

这个特性允许检查和报告 mutex 语义错误。

12.16 Lock debugging: detect incorrect freeing of live locks

锁定调试：探测不正确的活动锁定释放

这个特性将检查每个保持的锁定(spinlock, rwlock, mutex or rwsem) 是否被内核不正确地释放, 这个检查通过各种常规的内存释放方法(kfree(), kmem_cache_free(), free_pages(), vfree(), 等等)。它还检查每个活动的锁定是否被不正确地重置, 这一检查通过 spin_lock_init()/mutex_init()等实现, 或者检查在任务退出后, 是否还保持有锁定。

12.17 Lock debugging: prove locking correctness

锁定调试：验证锁定正确性

这个特性允许内核验证所有在内核运行时发生的锁定都是绝对正确的：在任何情况下, 任意（没有触发的）整合的可观察到的锁定队列（在任意数量的 CPU , 运行任意数量的任务和处理任意的中断）都不会导致死锁定错误。

简单地说, 这个特性允许内核在死锁定发生前就报告锁定的关系。

这个验证不依赖于一个死锁定的触发方案有多困难和多复杂：这个触发会需要多少个特殊的 CPU , 任务和中断内容等信息。它也不依赖于时间：如果一个竞争和一个有结果的死锁定在理论上是可能的（无论这个竞争的方案有多么的不可能），它将会被验证并立即通过内核进行报告（一旦事件被观察, 将使死锁定理论上可能）。

如果一个死锁定不可能（例如, 锁定的规则, 在内核看来是绝对正确的），那么内核将什么都不报告。

注意：这个特性同样用在 rwlocks, mutexes 和 rwsems 上 - 以便在这些不同的锁定变量中的所有依赖也被观察和映像。同样, 这项正确性验证将检查任意整合的各种锁定变量。

12.18 Lock usage statistics

锁定用法统计信息

这个功能允许追踪锁定争用点。

12.19 Spinlock debugging: sleep-inside-spinlock checking

旋转锁定调试：内部睡眠旋转锁定检查

如果你选 Y , 很多常规的程序如果在休眠时被保持的旋转锁定调用的话, 将会发出很大的声音。

12.20 Locking API boot-time self-tests

锁定接口启动自检

选 Y，如果你想要内核在启动时运行一个小的自检程序的话。自检程序将检查一般的锁定 BUG 类型是否被调试机构检测到（当然，如果你禁止了锁定调试，那些 BUG 将不会被检测）。锁定接口包含以下几个：spinlocks、rwlocks、mutexes 和 rwsems。

12.21 kobject debugging

kobject 调试

如果你选 Y，一些额外的 kobject 调试信息将会发至系统信息文件中。

12.22 Compile the kernel with debug info

编译内核包含调试信息

如果你选 Y，编译好的内核中将会包含调试信息，这也使得内核的镜像会变大。

它增加调试符号到内核和模块(gcc -g)中，如果你想要使用内核错误卸载工具或者二进制对象工具，如 crash, kgdb, LKCD, gdb 等，它将非常有用。

选 Y，如果你想要调试内核。

12.23 Debug VM

调试虚拟内存

这里会对虚拟内存系统进行额外的检查，也会影响系统性能。

如果不清楚，选 N。

12.24 Debug VM translations

调试 VM 转换

允许在虚拟内存页面进行一些额外的完整性检查。这可以用 virt_to_page() 和同类函数捕捉错误。

如果不清楚，选 N。

12.25 Debug filesystem writers count

调试文件系统写计数

开启这项以在 vfsmount 架构中截获写计数的使用错误。这将使每个文件架构增加 32 字节。

如果不清楚，选 N。

12.26 Debug linked list manipulation

调试链路表操作

选 Y，可以打开额外的链路表标准路径检查。

如果不清楚，选 N。

12.27 Debug SG table operations

调试 SG 表操作

选 Y，可以对 SG 表（散射 - 集合表）进行检查。这可以帮助程序查找驱动，它们在初始化时没有能够正确地找到 SG 表。

如果不清楚，选 N。

12.28 Compile the kernel with frame pointers

编译内核时加入构架指针

如果你选 Y，那么编译好的内核镜像将会有点大和慢，但是它很可能很有用，如果你想得到一些体系构架的信调试息，或者使用额外的调试器的话。

如果你不调试内核，选 N。

12.29 Delay each boot printk message by N milliseconds

对每个启动时的 printk 信息造成 N 毫秒的延迟

这个选项允许你读取内核的每个启动信息，它在这些信息中间加入了一些延迟。这个延迟值以毫秒计算，可以通过内核命令行 `boot_delay=N` 来修改。

看起来你需要使用 `lpj=M` 来预先设置“loops per jiffie”的值。

看看前一个启动信息关于“lpj”值的描述，然后在设置“`boot_delay=N`”之前先设置 `lpj=M`。

注意：使用这个选项将会对多 CPU 系统不利，第一个 CPU 以外的 CPU 将不会正确启动。

`BOOT_PRINTK_DELAY` 参数同样会引起参数 `DETECT_SOFTLOCKUP` 来探测它认为需要探测的锁定条件。

12.30 torture tests for RCU

RCU 严格测试

这个选项提供一个内核模块，它对 RCU 的基础构架进行严格的测试。这个模块将在内核设计为测试时建立。

选 M 如果你想要 RCU 严格测试编译为模块。

选 n 如果你不清楚。

12.31 Check for stalled CPUs delaying RCU grace periods

检查失速的 CPU 延迟 RCU 宽限周期

这个选项使 RCU 在每个延迟于宽限周期之内的 CPU 上用 printk 打印信息，但是宽限周期必须在时间周期之内。

选 Y，如果你需要 RCU 进行此项检查。

如果不清楚，选 N。

12.32 Kprobes sanity tests

K 探测器稳健测试

这个选项用于在启动时测试基本的 k 探测器系列功能。k 探测器的样本，jprobe 和 kretprobe 将插入和检验这些功能。

如果不清楚，选 N。

12.33 Self test for the backtrace code

对后台追踪代码自检

这个选项将提供一个内核模块，它可以用来检查内核堆栈的后台追踪代码。这个选项对于发行内核或者常规内核没有多大用处，但是对于内核开发者致力于体系构架的研究来说比较有用。

选 N，如果你不清楚。

12.34 Force extended block device numbers and spread them

强迫额外块设备数量和扩展它们

警告：开启这个选项将可能会在某些发行版导致崩溃！请不要开启它，除非你知道你在干什么。

如果不清楚，选 N。

12.35 Linux Kernel Dump Test Tool Module

Linux 内核转储测试工具模块

这个模块开启对不同的转储机制的测试，这通过包含系统预定义崩溃点来实现。

如果你不需要它，选 N。

选 M 编译为模块，名为：lkdtm。

12.36 Fault-injection framework

故障注入结构

提供故障注入结构。

12.37 Latency measuring infrastructure

延迟测量基础设施

选 Y，如果你想要使用延迟工具来寻找哪个用户空间被内核操作给堵塞了。

12.38 Sysctl checks

Sysctl 检查

sys.sysctl 使用二进制路径，这已经被发现与维护和使用相冲突。这里开启相关的检查，以帮助你确保一切正常。

12.39 Tracers

追踪器

- Kernel Function Tracer
内核功能追踪器
- Interrupts-off Latency Tracer
中断关闭延迟追踪器
- Sysprof Tracer
Sysprof 追踪器
- Scheduling Latency Tracer
调度延迟追踪器

- Trace process context switches
追踪处理环境切换
- Trace boot initcalls
追踪启动初始化调用
- Trace max stack
追踪最大堆栈

12.40 Remote debugging over FireWire early on boot

在启动时通过火线开启远程调试

如果你想要调试一些在启动的早期导致内核挂起或者崩溃的程序，同时这个崩溃的机器有一个火线接口的话，那么你可以使用这个特性，通过火线接口来远程访问崩溃机器的内存。这个远程访问直接内存存储是 OHCI1394 标准的一部分，现在已经是火线控制器的标准。

通过远程的 DMA，你可以使用 firescope 远程监视 printk 缓存，可以使用 gdb 下的工具 fireproxy 访问所有 4GB 以下的内存。甚至控制内核调试器都可以采用远程 DMA。

用法：

如果 ohci1394_dma=early 被用于启动参数，它将初始化所有的 OHCI1394 控制器，这些可以在 PCI 设置空间里找到。

由于所有对于火线总线的改变，比如允许和禁止设备，会导致总线重置，从而会对所有设备禁止远程 DMA。因此你在调试前要确信在调试对象启动前，电缆插上，火线在调试主机上开启。

这个代码（1K）在启动后被释放。到那时，火线堆栈将处于 OHCI-1394 控制器的控制之下，它将代替该代码使用。

12.41 Enable dynamic printk() call support

允许动态 printk() 调用支持

12.42 Sample kernel code

内核代码范例

你可以在这构建和测试内核代码范例。

12.43 Filter access to /dev/mem

过滤器访问 /dev/mem

12.44 Enable verbose x86 bootup info messages

允许详细 x86 启动信息

12.45 Early printk

预先 printk

12.45.1 Early printk via EHCI debug port

通过EHCI调试端口预先 printk

12.46 Check for stack overflows

检查堆栈溢出

这个选项，将在自由堆栈空间超过下限时打印相关信息。

12.47 Stack utilization instrumentation

堆栈利用操作设备

允许在 sysrq-T 和 sysrq-P 的调试输出中显示应用程序使用过的自由堆栈的最小值。

12.48 Debug page memory allocations

调试分页内存配置

通过 free_pages()来解除内核中的线性映射页。这会导致较大的速度降低，但是可以帮助找到内存损坏的类型。

12.49 Debug access to per_cpu maps

调试每 CPU 映射访问

12.50 Export kernel pagetable layout to userspace via debugfs

通过 debugfs 输出内核表至用户空间

12.51 Write protect kernel read-only data structures

写入保护内核只读数据结构

使内核的只读数据在分页表中变为写保护，这是为了捕获偶然的（或者不正确的）对于这个常数的写入。这是推荐选项，这样我们可以很快地找到内核 BUG。如果不清楚，选 Y。

12.51.1 Testcase for the DEBUG_RODATA feature

DEBUG_RODATA 特性测试盒

这个选项开启 DEBUG_RODATA 测试盒，这是 `change_page_attr()` 的基本设施。如果不清楚，选 N。

12.52 Testcase for the NX non-executable stack feature

Nx 非运行堆栈特性测试盒

这个选项开启 CPU NX 功能测试盒，同时提供一些相应的设置软件。

如果不清楚，选 N。

12.53 Use 4Kb for kernel stacks instead of 8Kb

使用 4Kb 内核堆栈代替 8Kb

如果你选 Y，内核将会使用 4Kb 的堆栈尺寸，它关系到每个进程和线程。这使得在系统中运行更多的线程更为容易，同时在更高层次的分配中减轻 VM 子系统的压力。这个选项也会使用 IRQ 堆栈来补偿减少的堆栈空间。

12.54 Memory mapped IO tracing

内存映射 IO 追踪

12.55 IO delay type

IO 延迟类型

12.55.1 port 0x80 based port-IO delay

port-IO 基准 0x80 端口延迟 (推荐)

这是传统的 Linux IO 延迟，用于 in/out_p。它经过最多的测试，所以是最安全的选择。

12.55.2 port 0xed based port-IO delay

port-IO 基准 0xed 端口延迟

使用 0xed 端口作为 IO 延迟。这将释放 0x80 端口，因为它经常被用于硬件调试端口。

12.55.3 udelay based port-IO delay

port-IO 基准 udelay 延迟

使用 udelay(2) 作为 IO 延迟。这在没有 IO 端口空间可供使用的时候来提供延迟。

12.55.4 no port-IO delay

没有 port-IO 延迟

没有 IO 端口延迟。这将使老的机器无法使用，因为它在具体操作上需要 IO 端口延迟。这在新的机器上可以工作。

12.55.5 Build kobject examples

构建 kobject 例子

这个配置选项允许你构建很多不同的 kobject 范例模块，可以展示如何正确使用 kobjects、ksets 和 ktypes。

12.56 Debug boot parameters

调试启动参数

12.57 CPA self-test code

CPA 自检查代码

每 30 秒用 `change_page_attr()` 进行自检。

12.58 Allow gcc to uninline functions marked ‘inline’

允许 gcc 对标记为“内联”的函数取消内联

Chapter 13

Security options 安全选项

13.1 Enable access key retention support

允许保持访问钥匙

这个选项可以在内核中保留访问的钥匙和验证标志。它同样提供一些方法，使得一个进程，如网络文件系统、加密支持等可以连接上这些钥匙。

而且，它还提供一个特殊的钥匙，它运作起来像一个钥匙链：一个可查询的钥匙序列。每个进程都配备了可访问的 5 个标准钥匙链：UID, GID, session, process 和 thread。

如果你不清楚这个功能是否会用上，选 N。

13.1.1 Enable the /proc/keys file by which keys may be viewed

生成/proc/keys 文件以便钥匙可以被访问

这个选项将生成/proc/keys 文件，里面包含所有的钥匙列表，可以被相应进程访问到。

在表中的钥匙是授权给访问进程查看的，无论这些进程是否能够控制钥匙。注意 LSM 安全检查仍然有效，而且可能会过滤掉没有授权给目前进程访问的钥匙。表中仅仅显示钥匙的属性，更多的量没有包括在结果表中。

如果你不清楚是否需要这项功能，选 N。

13.2 Enable different security models

允许不同的安全模式

这里允许你选择不同的安全模式，以配置进你的内核。

如果你没有选这个选项，Linux 将使用默认的安全模式。

如果你不清楚如何选择，选 N。

13.2.1 Socket and Networking Security Hooks

套接字和网络安全钩

这里允许套接字和网络安全钩。如果选上了，一个安全模块可以使用这些钩来实施套接字和网络访问控制。

如果不清楚如何选择，选 N。

- XFRM (IPSec) Networking Security Hooks

XFRM (IPSec)网络安全钩

这里允许 XFRM (IPSec)网络安全钩。如果你选上了，一个安全模块可以使用这些钩来实施每个包的访问控制，通常这些包的传递基于 IPSec 规则。非 IPSec 的通讯将会定义为未标识的，也只有被授权进行未标识通讯的套接字能够不通过 IPSec 进行传输。

如果不清楚如何选择，选 N。

13.2.2 Default Linux Capabilities

默认 Linux 性能

这里允许默认的 Linux 性能与功能。

如果不清楚如何选择，选 Y。

13.3 File POSIX Capabilities

POSIX 文件性能

这里允许文件系统性能，允许你在不使用 ROOT 用户的 ID 下，赋予二进制超级用户子集的权利。

如果不清楚，选 N。

13.4 Low address space to protect from user allocation

低端地址空间保护用户配置

这里是低端虚拟内存的一部分，以保护用户空间的配置。防止用户改写低端内存能够减少内核受到空指针 BUG 的冲击。

大多数用户拥有 65536 以上的内存空间，这是合理的，可以减少问题的发生。那些使用 vm86 功能的程序，要么需要额外的许可来访问 LSM 或者功能模块，要么要禁止掉这项保护功能。

这个值可以在系统启动后，在 `/proc/sys/vm/mmap_min_addr` 中调节。

13.5 NSA SELinux Support

NSA SELinux 支持

这里选择 NSA SELinux (Linux 安全加强) 支持。

你将同样需要一个规则选项和一个已标识的文件系统。

你可以获取规则编译器 (checkpolicy)，一个用于标识文件系统的实用工具(setfiles)，和一些配置的例子。

如果不清楚，选 N。

13.5.1 NSA SELinux boot parameter

NSA SELinux 启动参数

这个选项增加一个内核参数“selinux”，它允许 SELinux 在启动时运行或者不运行。如果这里选上了，SELinux 功能将根据内核命令行 selinux=0 而禁止。这个目的是允许一个单独的内核镜像能够在发布时嵌入 SELinux，但是也不一定需要。

如果不清楚，选 N。

- NSA SELinux boot parameter default value

NSA SELinux 启动参数默认值

这个选项设置内核参数 selinux 的默认值，这样可以让 SELinux 在系统启动时不运行。如果这个值是 0，那么 SELinux 内核参数将默认为 0，在系统启动时不运行。如果值是 1，那么 SELinux 内核参数将默认为 1，在系统启动时运行。

如果不清楚，选 1。

13.5.2 NSA SELinux runtime disable

NSA SELinux 运行时禁止

这个选项允许在一个 SELinux 文件节点中写入 disable 值，它使得系统在载入规则之前，禁止 SELinux 运行。SELinux 将一直保持禁止状态，直到下次系统启动。这个选项有点类似于 selinux=0 启动参数，但是支持在运行时禁止 SELinux，比如从/sbin/init 中实现，这样是为了在不同平台中的更好的可移植性，因为有的地方用启动参数很难实现。

13.5.3 NSA SELinux Development Support

NSA SELinux 开发支持

这里是 NSA SELinux 开发支持的选项。它对于测试 SELinux 和开发规则非常有用。如果不清楚，选 Y。内核将以宽容模式启动（记录所有东西，不拒绝任何东西），除非你在内核命令行中指定 enforcing=1。你可以通过/selinux/enforce 自由地在宽容模式和严格模式中切换（只要是规则允许）。

13.5.4 NSA SELinux AVC Statistics

NSA SELinux AVC 统计

这个选项统计 AVC（访问向量缓存）至 `/selinux/avc/cache_stats` 中，这可以通过 `avcstat` 等工具进行监视。

13.5.5 NSA SELinux checkreqprot default value

NSA SELinux checkreqprot 默认值

这个选项为 checkreqprot 标记设置默认值。这个标记的作用是决定 SELinux 通过程序还是通过内核响应进行安全检查（包含所有的隐含可执行文件，只读 - 隐含 - 可执行），这一检查通过 `mmap` 和 `mprotect` 调用实现。如果这个值设置为 0，SELinux 将通过内核响应进行安全检查。如果这个值设置为 1，SELinux 将通过程序响应进行安全检查。checkreqprot 标记可以在“`checkreqprot=`”启动参数中修改默认值。它同样可以在系统运行时通过 `/selinux/checkreqprot` 修改，只要规则允许。

如果你不清楚，选 1。

13.5.6 NSA SELinux enable new secmark network controls by default

NSA SELinux 默认允许新的网络辅助标记控制

这个选项决定 NSA SELinux 是否默认允许新的网络辅助标记控制。如果不是，老的包控制将会默认选择，保留老的行为。

如果你允许新的控制，你需要升级 SELinux 用户空间库、工具和规则。一般地，你的 Linux 发布版本将提供这些和允许在内核中执行新控制，它们同样也要进行发布。

注意这个选项能够通过 `selinux.compat.net` 参数设置来使之在系统启动时无效（启动后通过 `/selinux/compat.net` 设置）。

如果你允许新的网络控制，你将同样需要确定参数 `SECMARK` 和 `CONNSECMARK` 的目标，如同每个控制台你需要指定协议一样。

如果不清楚，选 N。

13.5.7 NSA SELinux maximum supported policy format version

NSA SELinux 最大支持的协议格式版本

这个选项允许设置 NSA SELinux 最大支持的协议格式版本个数。这个值在用户空间通过 `/selinux/policyvers` 反映，并且在协议载入时使用。它可以向下调整，以支持老的用户空间，这些老的空间不能正确地处理内核和支持更新的协议版本。

举例：

Fedora Core 3 和 4 版本的 Linux 发布版本，允许这个选项和通过下个选项进行设置相关值。Fedora Core 5 和以后的版本，将不需要这个选项。

如果不清楚，选 N。

13.6 Simplified Mandatory Access Control Kernel Support

简化强制访问控制内核支持

这里选择简化强制访问控制内核 (Smack)。Smack 对于敏感的、完整的和各种其他强制安全构架非常有用。

如果你不清楚，选 N。

Chapter 14

Cryptographic API 密码接口

这个功能提供核心的密码接口。

14.1 Sequence Number IV Generator

Number IV 序列生成器

这个 IV 生成器通过对一个序列数字进行逻辑异或处理来生成一个 IV 号码。这个算法主要应用在 CTR 和类似的模式中。

14.2 Cryptographic algorithm manager

密码算法管理

生成默认的密码模板范例。比如 CBC 加密算法。

14.3 HMAC support

HMAC 支持。

HMAC : 信息散列验证。IPSec 需要用到这项功能 (IPSec 的介绍见网络部分)。

14.4 XCBC support

XCBC 支持

XCBC : 散列加密算法

14.5 Null algorithms

空算法

这是被 IPsec 使用的“空”算法。它表示什么也不做。

14.6 MD4 digest algorithm

MD4 摘要算法

MD4 信息摘要算法。

14.7 MD5 digest algorithm

MD5 摘要算法

MD5 信息摘要算法。

MD5 : 在 Linux 中可以用 md5sum 命令来校验。

14.8 SHA1 digest algorithm

SHA1 摘要算法

SHA-1 安全散列标准。

安全散列算法 SHA (Secure Hash Algorithm , SHA) 是美国国家标准和技术局发布的国家标准 FIPS PUB 180-1 , 一般称为 SHA-1 。其对长度不超过 264 二进制位的消息产生 160 位的消息摘要输出。SHA 是一种数据加密算法, 该算法经过加密专家多年来的发展和改进已日益完善, 现在已成为公认的最安全的散列算法之一, 并被广泛使用。

14.9 SHA224 and SHA256 digest algorithm

SHA224 和 SHA256 摘要算法

SHA256 安全散列标准(DFIPS 180-2)。

这个版本执行 256 位的 SHA 散列, 它包含 128 位安全标准, 以应对冲突攻击。

这段代码也包含了 SHA-224 , 包含 112 位安全标准的 224 位散列, 以应对冲突攻击。

14.10 SHA384 and SHA512 digest algorithms

SHA384 和 SHA512 摘要算法

SHA512 安全散列标准(DFIPS 180-2)。

这个版本执行 512 位的 SHA 散列，它包含 256 位安全标准，以应对冲突攻击。

这段代码也包含了 SHA-384，包含 192 位安全标准的 384 位散列，以应对冲突攻击。

14.11 Whirlpool digest algorithms

Whirlpool (旋涡) 摘要算法

Whirlpool512，384 和 256 位的散列算法。

Whirlpool-512 是较早的“尼斯湖”加密算法之一。它将被列入 ISO/IEC 标准。

14.12 Tiger digest algorithms

Tiger (老虎) 摘要算法

Tiger192，160 和 128 位的散列算法。

老虎算法主要是针对 64 位处理器进行的优化算法，它同样适用于 32 位处理器。它是由 Ross Anderson 和 Eli Biham 开发的。

14.13 $\text{GF}(2^{128})$ multiplication functions

$\text{GF}(2^{128})$ 乘法功能 (试验中，新)

在 $\text{GF}(2^{128})$ 域中执行的有效乘法表。有的密码模式中需要这项功能。如果你选择了类似的密码模式，那么这项功能将会自动选上。如果你想要增加一个额外模块，而它需要这项功能，那么选 Y。

14.14 ECB support

ECB 支持 (新)

ECB：电子编码本模式

这是最简单的块加密算法。它简单地按次序加密输入的块。

14.15 CBC support

CBC 支持 (新)

CBC：块运算链路模式

这个算法在 IPSec 中使用。

14.16 PCBC support

PCBC 支持 (新)

PCBC : 第二代块运算链路模式

这个算法在 RxRPC 中使用。

14.17 LRW support

LRW 支持 (新)

LRW : 一种扭曲的, 非延展性的, 非移动性的狭小块运算模式, 它在数据管理加密中运用。使用时需要一个特别的字串模式 `aes-lrw-benbi` , 密钥必须是256, 320 或384 位。密钥前面的128, 192 或256 位用于 AES , 余下的用来将每个加密块捆绑到一起, 并放到它们的逻辑位置。

14.18 XTS support

XTS 支持 (新)

XTS : IEEE1619/D16 标准中的狭小块运算模式, 它的字串是 `aes-xts-plain` 。密钥长度 256, 384 或 512 位。目前当磁盘的扇区不是 16 的倍数时, 这个算法无法使用。

14.19 CTR support

CTR 支持

CTR:计数模式

这个块运算算法在 IPsec 中需要。

14.20 GCM/GMAC support

GCM/GMAC 支持

对伽罗瓦/计数模式(GCM)和伽罗瓦信息验证代码(GMAC)的支持。IPsec 需要。

14.21 CCM support

CCM 支持

对 CBC MAC 计数支持。IPsec 需要。

14.22 Software async crypto daemon

软件异步加密后台程序（新）

这是通用的软件异步加密后台程序。它将任意的同步的软件加密算法转换成异步的加密算法，这个过程在内核线程中执行。

14.23 DES and Triple DES EDE cipher algorithms

DES 和多重 DES EDE 加密算法

DES 加密算法，以及多重 EDS EDE。

14.24 FCrypt cipher algorithm

Fcrypt 加密算法

RxRPC 使用的一种算法。

14.25 Blowfish cipher algorithm

Blowfish 加密算法

Blowfish（河豚）加密算法，是 Bruce Schneier 开发的。

这种算法的密钥是可变的，可以使用 32 位至 448 位的密钥长度。它非常迅速而且简单，主要是为大型的处理器的设计。

14.26 Twofish cipher algorithm

Twofish 加密算法

双鱼加密算法使用于计数窗格系统，它被研究者纳入 AES（高级加密标准）的算法之一。它采用 16 个循环块加密的方式，支持的密钥长度有 128、192 和 256 位。

14.27 Twofish cipher algorithms (i586)

Twofish 加密算法(i586)

用于 586 处理器的双鱼加密算法。

14.28 Serpent cipher algorithm

Serpent (巨蛇) 加密算法

这个算法由 Anderson, Biham 和 Knudsen 开发。它的密钥由以前的 8 位变成 0 - 256 位，其中也包括 'Tnepres' (Serpent 这个单词的反写) 算法，顾名思义，就是反向的巨蛇加密算法，这是为了兼容老的内核代码。

14.29 AES cipher algorithms

AES (高级加密标准) 加密算法

AES 加密算法(FIPS-197)，它使用 Rijndael 算法。无论用于反馈系统还是非反馈系统，Rijndael 算法在硬件和软件中都有着好的表现，即便运用于大的运算环境。它的密钥生成时间非常优秀，而且密钥的灵活性很好。Rijndael 在严格的环境中运行得非常好，这得益于它的内存需求量很低。在面对电路和计数器类型的攻击时，它的操作非常简单。

AES 使用三种密钥长度：128、192 和 256 位。

14.30 AES cipher algorithms (i586)

用于 586 处理器的 AES 加密算法。

14.31 CAST5 (CAST-128) cipher algorithm

CAST5 (CAST-128)加密算法

CAST5 加密算法 (等同于 128 位 CAST) 在 RFC2144 中提及。

14.32 CAST6 (CAST-256) cipher algorithm

CAST6 (CAST-256)加密算法

CAST6 加密算法 (等同于 256 位 CAST) 在 RFC2612 中提及。

14.33 TEA, XTEA and XETA cipher algorithms

TEA, XTEA 和 XETA 加密算法

TEA (小型加密算法) 是一种简单的运算，它用于很多安全环境，它非常小巧，占用的内存很少。

XTEA (扩展小型加密算法) 是 TEA 的升级，它改善了密钥的潜在弱点。

XETA 是 XTEA 的兼容模式，以便一些环境无法使用 XTEA 时使用。

14.34 ARC4 cipher algorithm

ARC4 加密算法

ARC4 是一种加密流，使用的密钥从 8 位到 2048 位。这种算法主要运用于基于驱动器的 WEP（详见下文），但是它在其他场合由于算法的弱点，不被广泛使用。

WEP-Wired Equivalent Privacy 加密技术，WEP 安全技术源自于名为 RC4 的 RSA 数据加密技术，以满足用户更高层次的网络安全需求。

WEP 是 Wired Equivalent Privacy 的简称，有线等效保密（WEP）协议是对在两台设备间无线传输的数据进行加密的方式，用以防止非法用户窃听或侵入无线网络。

802.11b 标准里定义的一个用于无线局域网(WLAN)的安全性协议。WEP 被用来提供和有线 lan 同级的安全性。LAN 天生比 WLAN 安全，因为 LAN 的物理结构对其有所保护，部分或全部网络埋在建筑物里面也可以防止未授权的访问。

经由无线电波的 WLAN 没有同样的物理结构，因此容易受到攻击、干扰。WEP 的目标就是通过对无线电波里的数据加密提供安全性，如同端-端发送一样。WEP 特性里使用了 rsa 数据安全性公司开发的 rc4 prng 算法。如果你的无线基站支持 MAC 过滤，推荐你连同 WEP 一起使用这个特性(MAC 过滤比加密安全得多)。

14.35 Khazad cipher algorithm

Khazad 加密算法

Khazad 是初始的 NESSIE（尼斯湖）竞赛中的决赛算法。它针对 64 位处理器进行了优化，同时它在 32 位处理器也运行得很好。它使用 128 位的密钥。

14.36 Anubis cipher algorithm

Anubis 加密算法

Anubis 使用可变的 128 至 320 位的密钥。它是尼斯湖竞赛的参赛者之一。

NESSIE（尼斯湖竞赛）是欧洲委员会的信息社会技术(IST)规划中所支持的一项工程，支持年限三年(2000 年 1 月至 2002 年 12 月)，投资 33 亿欧元。它对各种计算机加密算法分析性能指标及运行结果，并做出评价。

14.37 SEED cipher algorithm

SEED（种子）加密算法

它采用 128 位的密钥块运算。这种算法由 KISA（朝鲜信息安全机构）开发，并成为朝鲜共和国的国家标准算法。

14.38 Salsa20 stream cipher algorithm

Salsa20 流加密算法

Salsa20 流加密算法。

Salsa20 流加密算法是由 eSTREAM（流加密工程项目）提出。它由 Daniel J. Bernstein 设计。

14.39 Salsa20 stream cipher algorithm (i586)

Salsa20 流加密算法(i586)

用于 586 处理器的 Salsa20 流加密算法。

14.40 Deflate compression algorithm

Deflate 压缩算法

这是 Deflate 压缩算法(RFC1951)，运用于 IPComp 协议下的 IPSec。

如果你使用 IPSec，那么你很有可能会需要它。

14.41 Michael MIC keyed digest algorithm

Michael MIC 密钥摘要算法

Michael MIC 用于 TKIP，主要是为了保证信息的完整性。但是它在其他地方不是很常用，因为算法本身存在弱点。

14.42 CRC32c CRC algorithm

CRC32c CRC 算法

循环冗余校验算法，被 iSCSI 和很多场合运用于数据校验。这个算法执行 lib/libcrc32c，模块名为 crc32c。

14.43 Camellia cipher algorithms

Camellia 加密算法

Camellia（茶花）是一种对称密钥块运算，由 NTT 和三菱电气公司联合开发。它采用三种密钥长度：128、192 和 256 位。

14.44 Testing module

测试模块

快速和不正规的加密测试模块。

14.45 Authenc support

Authenc 支持

Authenc : 为 IPsec 而整合封装的模式。这在 IPsec 中会用到。

14.46 LZO compression algorithm

LZO 压缩算法

这里是 LZO 算法。

14.47 Hardware crypto devices

硬件加密设备

选 Y，你将能看到硬件加密设备和处理器的选项。这个选项本身不会增加任何的内核代码。

如果你选 N，所有的子选项将被跳过和禁用。

14.47.1 Support for VIA PadLock ACE

VIA（威盛，一个厂家）高级密码引擎锁支持

一些 VIA 处理器使用整合的加密引擎（所以叫做高级密码引擎），它提供非常快的密码指令操作，以支持各种加密算法。

这些指令只有在 CPU 支持下才能运行，否则只是软件级别上的加密。

- PadLock driver for AES algorithm

AES 算法锁驱动

VIA 锁的 AES 算法驱动。

在 VIA C3 和更新的 CPU 上可以使用。

如果不确定，选 M。模块名为 padlock-aes.ko。

- PadLock driver for SHA1 and SHA256 algorithms

SHA1 和 SHA256 算法锁驱动

VIA 锁的 SHA1/SHA256 算法驱动。

在 VIA C7 和更新的 CPU 上可以使用。

如果不确定，选 M。模块名为 padlock-sha.ko。

14.47.2 Support for the Geode LX AES engine

Geode LX AES 引擎支持

选 Y，你将可以使用固化在 AMD Geode LX 处理器主板上的 AES 引擎来执行 AES 加密算法。

选 m 编译为模块，名为 geode-aes。

14.47.3 Driver HIFN 795x crypto accelerator chips

HIFN 795x 加密加速芯片驱动

这个选项为 HIFN 795x 加密适配器提供支持。

Chapter 15

Virtualization 虚拟机

在这选 Y，你将可以看到一些相关的选项，以使用你的 Linux 主机通过虚拟机（客户）模式运行其他的操作系统。

这个选项本身不增加任何的内核代码。

如果你选 N，所有的子选项将跳过和禁止。

15.1 Kernel-based Virtual Machine (KVM) support

基于内核的虚拟机（KVM）支持

通过硬件扩展，来实现主机完全地对客户机的虚拟化。你将需要一个不错的较新的处理器，上面配置有虚拟化扩展。你还需要在下面的选项选择 1 个或者更多的处理器模块。

这些模块提供对硬件功能的访问权，这通过 `/dev/kvm` 中的一个设备节点字符来实现。

选 m 编译为模块，名为 `kvm`。

15.1.1 KVM for Intel processors support

对 Intel 处理器的 KVM 支持

对 intel 处理器提供 KVM 支持。这些处理器上配备有 VT 扩展。

15.1.2 KVM for AMD processors support

对 AMD 处理器的 KVM 支持

对 AMD 处理器提供 KVM 支持。这些处理器上配备有 AMD-V(SVM)扩展。

15.1.3 KVM trace support

KVM 追踪支持

这个选项允许通过 `relayfs` 读取 `kvm_related` 事件追踪。注意 ABI 并不认为是稳定的，将会在不久的版本中修改。

15.2 Linux hypervisor example code

Linux 高级伪装范例代码

这是非常简单的模块，允许你运行多重的 Linux 内核实例。它使用“`lguest`”命令（在 `Documentation/lguest` 目录中有介绍）。注意“`lguest`”是“`fell quest`”（击倒用户）的谐音，而不是“`rustyvisor`”（生锈的伪装）。

如果不清楚，选 N。如果好奇，选 M。如果喜欢受虐，选 Y。

15.3 PCI driver for virtio devices

virtio 虚拟设备的 PCI 驱动

这个驱动为 PCI 设备提供基于 virtio 技术的虚拟支持。这需要你的 VMM 有合适的对于 PCI virtio 的支持。大多数基于 VMM 的 QEMU 应该支持这些设备（如 KVM 或者 Xen）。

virtio，一个公司名，主要在各个平台开发虚拟和仿真软件。

VMM：英特尔的 Vanderpool 技术引入了一个软件层，即虚拟机监视器（Virtual Machine Monitor，简称 VMM）。

QEMU：是一套由 Fabrice Bellard 所编写的模拟操作系统的自由软件。它与 Bochs，PearPC 近似，但其具有某些后两者所不具备的特性，如高速度及跨平台的特性。经由 `kqemu` 这个非自由的加速器，QEMU 能模拟至接近真实电脑的速度。

15.4 Virtio balloon driver

Virtio 气泡驱动

这个驱动支持增加和减少 KVM 客户的内存。

如果不清楚，选 M。

Chapter 16

Library routines 例行库

16.1 CRC-CCITT functions

CRC-CCITT 功能(CRC_CCITT)

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC-CCITT 功能，那么你要在这选 M。

16.2 CRC16 functions

16 位 CRC 校验功能

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC16 功能，那么你要在这选 M。

16.3 CRC calculation for the T10 Data Integrity Field

CRC 计算 T10 数据完整域

这个选项只有在模块不处于内核树中，需要计算 CRC 检查时用到，它用于 SCSI 数据完整性子系统。

16.4 CRC ITU-T V.41 functions

CRC ITU-T V.41 功能

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC ITU-T V.41 功能，那么你要在这选 M。

16.5 CRC32 functions

32 位 CRC 功能

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC32 功能，那么你要在这选 M。

16.6 CRC7 functions

CRC7 功能

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC7 功能，那么你要在这选 M。

16.7 CRC32c Cyclic Redundancy-Check

32 位 CRC-C 检测

这个功能是为一些需要的模块而提供的，通常这些模块不是编译在内核之中。如果这些模块需要用到 CRC32c 功能，那么你要在这选 M。模块名为 libcrc32c。

注：

CRC（循环冗余检查）——一种数据传输检错功能，对数据进行多项式计算，并将得到的和数附在帧的后面。接收设备也执行类似的算法。根据应用环境与习惯的不同，CRC 又可分为以下几种标准：

- ① CRC-12 码；
- ② CRC-16 码；
- ③ CRC-CCITT 码；
- ④ CRC-32 码。

致谢

感谢Ubuntu和LFS在Linux上对我的帮助。

感谢 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$ ，用它来排版真是很好玩。