

menuconfig配置菜单

内核编译选项

- 1、General setup
- 2、Enable loadable module support
- 3、Enable the block layer
- 4、Processor type and features (处理器类型及特点)
- 5、Power management and ACPI options
- 6、Bus options(PCI etc.)
- 7、Executable file formats / Emulations
- 8、Networking support
- 9、Device Drivers
- 10、Firmware Drivers
- 11、File systems
- 12、Kernel hacking
- 13、Security options
- 14、Virtualization
- 15、Library routines

内核的编译

menuconfig配置菜单

在选择相应的配置时，有三种选择方式，它们分别代表的含义如下：

Y—将该功能编译进内核

N—不将该功能编译进内核

M—将该功能编译成可以在需要时动态插入到内核中的模块

使用的是 `make menuconfig`，需要使用空格键进行选取。在每一个选项前都有一个括号，有的是中括号有的是尖括号，还有圆括号。用空格键选择时可以发现，中括号里要么是空，要么是“*”，而尖括号里可以是空，“*”和“M”这表示前者对应的项要么不要，要么编译到内核里；后者则多一样选择，可以编译成模块。而圆括号的内容是要你在所提供的几个选项中选择一项。

(注：其中有不少选项是目标板开发人员加的，对于陌生选项，自己不知道该选什么时建议使用默认值)

下面我们来看看具体配置菜单，进入内核所在目录，键入 `make menuconfig` 你就会看到配置菜单具有如下一些项：

内核编译选项

1、General setup

代码成熟度选项，它又有子项：

1.1、prompt for development and/or incomplete code/drivers

该选项是对那些还在测试阶段的代码，驱动模块等的支持。一般应该选这个选项，除非你只是想使用 LINUX 中已经完全稳定的东西。但这样有时对系统性能影响挺大。

1.2、Cross-compiler tool prefix

交叉编译工具前缀，例如：Cross-compiler tool prefix值为：(arm-linux-)

1.3、Local version - append to kernel release

内核显示的版本信息，填入 64字符以内的字符串，你在这里填上的字符串可以用uname -a命令看到。

1.4、 Automatically append version information to the version string

自动在版本字符串后面添加版本信息, 编译时需要有perl以及git仓库支持

1.5、Kernel compression mode (Gzip) --->

有四个选项，这个选项是说内核镜像要用的压缩模式，回车一下，可以看到gzip, bzip2, lzma, lzo, 一般可以按默认的gzip, 如果要用bzip2, lzma, lzo要先装上支持

1.6、Support for paging of anonymous memory (swap)

使用交换分区或交换文件来做为虚拟内存，一定要选上。

1.7、System V IPC

表示系统的进程间通信Inter Process Communication，它用于处理器在程序之间同步和交换信息，如果不选这项，很多程序运行不起来，必选。

1.8、POSIX Message Queues

POSIX标准的消息队列，它同样是一种IPC。建议你最好将它选上

1.9、BSD Process Accounting

用户进程访问内核时将进程信息写入文件中。通常主要包括进程的创建时间/创建者/内存占用等信息。建议最好选上。

BSD Process Accounting version 3 file format

使用新的第三版文件格式, 可以包含每个进程的PID和其父进程的PID, 但是不兼容老版本的文件格式。

1.11、Export task/process statistics through netlink (EXPERIMENTAL)

通过netlink接口向用户空间导出任务/进程的统计信息, 与BSD Process Accounting的不同之处在于这些统计信息在整个任务/进程生存期都是可用的

Enable per-task delay accounting (EXPERIMENTAL)

在统计信息中包含进程等候系统资源(cpu, IO同步, 内存交换等)所花费的时间

Enable extended accounting over taskstats (EXPERIMENTAL)

在统计信息中包含扩展进程所花费的时间

1.12、Auditing support

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

1.13、Enable system-call auditing support

支持对系统调用的审计

1.14、IRQ subsystem --->

中断子系统

Support sparse irq numbering

<== 支持稀有的中断编号，关闭

1.15、RCU Subsystem --->

非对称读写锁系统 是一种高性能的kernel 锁机制，适用于读多写少环境

RCU Implementation (Tree-based hierarchical RCU) --->

RCU 实现机制 Tree(X) Tree-based hierarchical RCU 基本数按等级划分

Enable tracing for RCU

激活跟踪

(32) Tree-based hierarchical RCU fanout value

基本数按等级划分分列值

Disable tree-based hierarchical RCU auto-balancing

1.16、< > Kernel .config support

这个选项允许.config文件（即编译Linux时的配置文件）保存在内核当中

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

1.18、[] Control Group support --->

cgroups 支持， 文档资料 ，cgroups 主要作用是给进程分组，并可以动态调控进程组的CPU 占用率。比如A 进程分到apple 组，给予20%CPU 占用率，E 进程分easy 组，给予50%CPU 占用率，最高100% 。我目前没有此类应用场景，用到时会选择将其编译进去。

CPU bandwidth provisioning for FAIR_GROUP_SCHED

此选项允许用户定义的CPU带宽速率（限制）在公平的组调度运行的任务。组没有限制设置被认为是无约束和运行没有限制。

Group scheduling for SCHED_RR/FIFO

此功能可以让您显式地分配真实的CPU带宽任务组。

1.19、-*- Namespaces support --->

命名空间支持，允许服务器为不同的用户信息提供不同的用户名空间服务

[*] UTS namespace

通用终端系统的命名空间。它允许容器，比如Vservers利用UTS命名空间来为不同的服务器提供不同的UTS。如果不清楚，选N。

[*] IPC namespace

IPC命名空间，不确定可以不选

[*] User namespace (EXPERIMENTAL)

User命名空间，不确定可以不选

[*] PID Namespaces

PID命名空间，不确定可以不选

[*] Network namespace

1.20、Automatic process group scheduling 自动进程组调度

1.21、[] enable deprecated sysfs features to support old userspace tools

1.22、*- Kernel->user space relay support (formerly relayfs)

在某些文件系统上(比如debugfs)提供从内核空间向用户空间传递大量数据的接口,我目前没有此类应用场景

1.23、[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support

用于在真正内核装载前,做一些操作(俗称两阶段启动),比如加载module, mount 一些非root 分区,提供灾难恢复 shell 环境等, 资料,我是期望直接从kernel image 直接启动,所以没选它

1.24、Initramfs source file(s)

initrd已经被initramfs取代,如果你不明白这是什么意思,请保持空白

1.25、Optimize for size

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。没必要选。选上了有时会产生错误的二进制代码。

1.26、Enable full-sized data structures for core: 在内核中使用全尺寸的数据结构.禁用它将使得某些内核的数据结构减小以节约内存,但是将会降低性能。

1.27、Enable futex support: 快速用户空间互斥体可以使线程串行化以避免竞态条件,也提高了响应速度.禁用它将导致内核不能正确的运行基于glibc的程序。

1.28、Enable eventpoll support: 支持事件轮循的系统调用。

1.29、Use full shmem filesystem: 除非你在很少的内存且不使用交换内存时,才不要选择这项。后面的这四项都是在编译时内存中的对齐方式,0 表示

编译器的默认方式。使用内存对齐能提高程序的运行速度,但是会增加程序对内存的使用量。内核也是一组程序呀。

Enable VM event counters for /proc/vmstat: 允许在/proc/vmstat中包含虚拟内存事件计数器。

[*] Disable heap randomization

禁用随机heap (heap堆是一个应用层的概念,即堆对CPU是不可见的,它的实现方式有多种,可以由OS实现,也可以由运行库实现,如果你愿意,你也可以在一个栈中来实现一个堆)

1.30、Choose SLAB allocator (SLAB) --->

选择内存分配管理器(强烈推荐使用SLUB)

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

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。这是为了编译某些特殊用途的内核使用的,例如引导盘系统。配置标准的内核特性(为小型系统)

Enable 16-bit UID system calls: 允许对UID系统调用进行过时的16-bit包装。

Sysctl syscall support 几乎使用不到这一选项,不选它可以轻微使内核变小

Include all symbols in kallsyms: 在kallsyms中包含内核知道的所有符号,内核将会增大300K。

Enable support for printk: 允许内核向终端打印字符信息,在需要诊断内核为什么不能运行时选择。

BUG() support: 显示故障和失败条件(BUG 和WARN),禁用它将可能导致隐含的错误被忽略。

Enable ELF core dumps: 内存转储支持,可以帮助调试ELF格式的程序。

1.32、[*] Profiling support

不选剖面支持，用一个工具来扫描和提供计算机的剖面图。支持系统评测（对于大多数用户来说并不是必须的）

1.33、OProfile system profiling

OProfile评测和性能监控工具

1.35、[] OProfile multiplexing support (EXPERIMENTAL)

1.34、Kprobes

调试内核除非开发人员，否则不选

1.35、Optimize trace point call sites

1.36、GCOV-based kernel profiling

[] Enable gcov-based kernel profiling 不选

2、Enable loadable module support

2.1 Forced module loading

允许强制加载模块

2.2 Module unloading

允许卸载已经加载的模块

2.3 Forced module unloading

允许强制卸载正在使用中的模块(比较危险)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）。这主要是针对开发者和冲动的用户提供的功能。如果不清楚，选N。

2.4 Module versioning support

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。允许使用其他内核版本的模块(可能会出问题)

2.5 Source checksum for all modules

为所有的模块校验源码, 如果你不是自己编写内核模块就不需要它这个功能是为了防止你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而造成版本冲突。如果不清楚，选N。

3、Enable the block layer

块设备支持, 使用硬盘/USB/SCSI设备者必选这项使得块设备可以从内核移除。如果不选，那么 `blockdev` 文件将不可用，一些文件系统比如 `ext3` 将不可用。这个选项会禁止 SCSI 字符设备和 USB 储存设备，如果它们使用不同的块设备。选Y，除非你知道你不需要挂载硬盘和其他类似的设备。不过此项无可选项

3.1 Support for large (2TB+) block devices and files

仅在使用大于2TB的块设备时需要

3.2 Block layer SG support v4

通用scsi块设备第4版支持

3.3 Block layer data integrity support

块设备数据完整性支持

3.4 Block layer bio throttling support

可用于限制设备的IO速度

3.5 Partition Types

Advanced partition selection

如果你想要在linux上使用一个在其他的介质上运行着操作系统的硬盘时，选择Y，如果你不确定时可以选N

3.6 IO Schedulers

IO调度器I/O是输入输出带宽控制，主要针对硬盘，是核心的必须的东西。这里提供了三个IO调度器。

Deadline I/O scheduler

使用轮询的调度器,简洁小巧,提供了最小的读取延迟和尚佳的吞吐量,特别适合于读取较多的环境(比如数据库)Deadline 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

使用QoS策略为所有任务分配等量的带宽,避免进程被饿死并实现了较低的延迟,可以认为是上述两种调度器的折中.适用于有大量进程的多用户系统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。

Default I/O scheduler (CFQ)

默认IO调度器我这样理解上面三个IO调度器:

抢先式是传统的,它的原理是一有响应,就优先考虑调度。如果你的硬盘此时在运行一项工作,它也会暂停下来先响应用户。期限式则是:所有的工作都有最终期限,在这之前必须完成。当用户有响应时,它会根据自己的工作能否完成,来决定是否响应用户。CFQ则是平均分配资源,不管你的响应多急,也不管它的工作量是多少,它都是平均分配,一视同仁的。

(*) Deadline

() CFQ

() No-op

4、Processor type and features (处理器类型及特点)

4.1 DMA memory allocation support

该选项允许小于32位地址的设备使用前16MB的地址空间,如果不缺定的话,选Y

4.2 Symmetric multi-processing support

对称多处理器支持,如果你有多个CPU或者使用的是多核CPU就选上.此时"Enhanced Real Time Clock Support"选项必须开启,"Advanced Power Management"选项必须关闭如果你选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 ” or “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。

4.3 Enable MPS table

MPS多处理器规范，不选

4.4 Support for big SMP systems with more than 8 CPUs

默认情况下为不选

4.5 Support for extended (non-PC) x86 platforms

如果选的话，你将可以选择支持如下32位X86的平台。

AMD Elan, NUMAQ (IBM/Sequent), RDC R-321x SoC, SGI 320/540 (Visual Workstation), STA2X11-based (e.g. Northville), Summit/EXA (IBM x440), Unisys ES7000 IA32 series

Moorestown MID devices如果你有这样的系统，或者你想要构建一个这样的通用的分布式，选择Y，否则选择N

4.6 intel MID platform support

Medfield MID platform

4.7 RDC R-321x SoC

嵌入式相关，不选

4.8 Support non-standard 32-bit SMP architectures

非标准的32位SMP结构支持，不选

4.9 Eurobraille/Iris poweroff module

来自eurobraille的iris机器不支持APM和ACPI来适时关闭自己，此模块在内核中起到这一作用。这是用于eurobraille的iris机子，不确定的话，不选。

4.10 Single-depth WCHAN output

跟 proc 相关的最好不要关，选Y

4.11 paravirt-ops debugging

4.12 Memtest

这一选项使内核增加一个“memtest”（内核测试）的参数，这将允许设置memtest。如果你不知道如何回答这个问题，选择N

4.13 Processor family (Pentium-Pro)处理器系列, 请按照你实际使用的CPU选择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 () 586/K5/5x86/6x86/6x86MX () Pentium-Classic () Pentium-MMX(系统默认选项) 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()
Intel Atom

4.14 Generic x86 support 这一选项针对x86系列的CPU使用更多的常规优化。如果你在上面一项选的是i386、i586之类的才选这个通用x86支持,如果你的CPU能够在上述“Processor family”中找到就别选除了对上面你选择的X86 CPU进行优化,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。这个选项提供了对X86系列CPU最大的兼容性,用来支持一些少见的x86构架的CPU。如果你的CPU能够在上面的列表中找到,就里就不用选了。

4.15 PentiumPro memory ordering errata workaround

旧的PentiumPro多处理器系统有勘误能力,可能会导致在少数的情况下,违反x86的排序标准内存操作。启用此选项将尝试解决一些(但不是全部)此类问题,但将以spinlock和内存为代价。

4.16 HPET Timer Support (HPET时钟支持)

允许内核使用HPET。HPET是替代8254芯片的新一代定时器,i686及以上级别的主板都支持,可以安全的选上。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。

4.17 Enable DMI scanning

选择Y除非你已经证明当进入DMI时不影响你的配置。PNP BIOS代码需要这一项的支持。

4.18 (8) Maximum number of CPUs

支持的最大CPU数,每增加一个内核将增加8K体积

4.19 SMT (Hyperthreading) scheduler support

支持Intel的超线程(HT)技术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。如果你不清楚,选N

4.20 Multi-core scheduler support

针对多核CPU进行调度策略优化多核调度机制支持,双核的CPU要选。多核心调度在某些情况下将会对多核的CPU系列有较好的支持。如果你不清楚,选N

4.21 Fine granularity task level IRQ time accounting

如果不确定的话,选N,默认为不选。

4.22 Preemption Model (Voluntary Kernel Preemption (Desktop))

内核抢占模式一些优先级很高的程序可以先让一些低优先级的程序执行,即使这些程序是在核心态下执行。从而减少内核潜伏期,提高系统的响应。当然在一些特殊的点的内核是不可抢先的,比如内核中的调度程序自身在执行时就是不可被抢先的。这个特性可以提高桌面系统、实时系统的性能。

No Forced Preemption (Server)

适合服务器环境的禁止内核抢占 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,而不会理会调度上的延迟。

(默认选项) Voluntary Kernel Preemption (Desktop)

适合普通桌面环境的自愿内核抢占 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,提供更快的应用程序响应。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。

Preemptible Kernel (Low-Latency Desktop)

适合运行实时程序的主动内核抢占 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。 如果你要最快的响应，选第三项。我认为万物是平衡的，低延迟意味着系统运行不稳定，因为过多来响应用户的要求，所以我选第二个。

4.23 Reroute for broken boot IRQs

防止同时收到多个boot IRQ（中断）时，系统混乱

4.24 Machine Check / overheating reporting

让CPU 检测到系统故障时通知内核，以便内核采取相应的措施（如过热关机等）

Intel MCE features

AMD MCE features

Support for old Pentium 5 / WinChip machine checks

4.25 Machine check injector support

让CPU检测到系统故障时通知内核，以便内核采取相应的措施（如过热关机等）（不明白此项与上一项的区别，原来的配置中是做模块加入内核的）

4.26 Enable VM86 support

这一选项用于支持在像DOSEMU一样的程序在x86的处理器上运行16位的legacy代码。也可能像XFree86这样的软件通过BIOS初始化声卡的时候会用到。

4.27 Toshiba Laptop support

东芝笔记本模块支持，不选

4.28 Dell laptop support

Dell笔记本模块支持，不选

4.29 Enable X86 board specific fixups for reboot

修正某些旧x86主板的重起bug，这种主板基本绝种了，我认为可以不选择这一项

4.30 dev/cpu/microcode - microcode support

是否支持Intel IA32架构的CPU。这个选项将让你可以更新Intel IA32系列处理器的微代码，显然你需要到网上去下载最新的代码，Linux不提供这些代码。当然你还必须在文件系统选项中选择/dev file system support才能正常的使用它。如果你把它译为模块，它将是 microcode。 IA32主要用于高于4GB的内存。详见下面的“高内存选项”。使用不随Linux内核发行的IA32微代码，你必需有IA32微代码二进制文件，仅对Intel的CPU有效。

Intel microcode patch loading support

AMD microcode patch loading support

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

是否打开CPU特殊功能寄存器的功能。这个选项桌面用户一般用不到，它主要用在Intel的嵌入式CPU中的，这个寄存器的作用也依赖与不同的CPU类型而有所不同，一般可以用来改变一些CPU原有物理结构的用途，但不同的CPU用途差

别也很大。在多cpu系统中让特权CPU访问x86的MSR寄存器。

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

是否打开记录CPU相关信息功能。这会在/dev/cpu中建立一系列的设备文件，用以让过程去访问指定的CPU。能从/dev/cpu/x/cpuid获得CPU的唯一标识符(CPUID)。

4.33 High Memory Support (4GB)

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。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。

() off (X) 4GB () 64GB

4.34 Memory model (Flat Memory)

一般选“Flat Memory”，其他选项涉及内存热插拔(X) Flat Memory（平坦内存模式）

这个选项允许你改变内核在内部管理内存的一些方式。大多数用户在这只会有一个选项：Flat Memory。这是普遍的和正确的选项。一些用户的机器有更高级的特性，比如 NUMA 和内存热拔插，那将会有不同的选项。Discontiguous Memory（非接触式内存模式）是一个更成熟、更好的测试系统。但是对于内存热拔插系统不太合适，会被“Sparse Memory”代替。如果你不清楚“Sparse Memory”和“Discontiguous Memory”的区别，选后者。如果不清楚，就选Flat Memory。

Sparse Memory（稀疏内存模式）

这对某些系统是唯一选项，包括内存热拔插系统。这正常。对于其他系统，这将会被Discontiguous Memory选项代替。这个选项提供潜在的更好的特性，可以降低代码复杂度，但是它是新的模式，需要更多的测试。如果不清楚，选择“Discontiguous Memory”或“Flat Memory”。

4.35 Allow for memory compaction

4.36 Page migration（选择Y）

Enable KSM for page merging

(4096) Low address space to protect from user allocation

4.37 Enable recovery from hardware memory errors

选择Y

4.38 Transparent Hugepage Support

Transparent Hugepage Support sysfs defaults

1.always 2.madvise（默认选项）

4.39 Cross Memory Support

4.40 Enable cleancache driver to cache clean pages if tmem is present

4.41 Enable frontswap to cache swap pages if tmem is present (这三部分不太了解)

4.42 Allocate 3rd-level pagetables from highmem

在内存很多 (大于 4G) 的机器上将用户空间的页表放到高位内存区, 以节约宝贵的低端内存。

4.43 Check for low memory corruption

低位内存脏数据检查, 默认是每60 秒检查一次。一般这种脏数据 是因某些Bios 处理不当引起的。

4.44 (64) Amount of low memory, in kilobytes, to reserve for the BIOS

4.45 Math emulation

数学协处理器仿真, 486DX以上的cpu就不要选它了

4.46 MTRR (Memory Type Range Register) support (内存类型区域寄存器)

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。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。Cyrrix 6×86, 6×86MX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。打开它可以提升PCI/AGP总线上的显卡2倍以上的速度, 并且可以修正某些BIOS错误。

4.47 MTRR cleanup support

MTRR清理 (2.6.27内核新增功能, 不确定可以不选)

MTRR cleanup enable value (0-1)

MTRR cleanup spare reg num (0-7)

x86 PAT support

x86 architectural random number generator

4.48 EFI runtime service support (EFI启动支持)

这里允许内核在EFI平台上使用储存于EFI固件中的系统设置启动。这也允许内核在运行时使用EFI的相关服务。这个选项只有在有EFI固件的系统上有用, 它会使内核增加8KB。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 (GRUB和LILO完全不知道EFI是什么东西)。即使你没有EFI, 却选了这个选项, 内核同样可以启动。大家应该用的是GRUB, 所以选上这个也没什么用。除非你的系统支持EFI (一种可代替传统BIOS的技术) 否则不选。

4.49 Enable seccomp to safely compute untrusted bytecode

只有嵌入式系统可以不选

4.50 Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)

4.51 Timer frequency (1000 HZ)

内核时钟频率桌面1000服务器100或250, 允许设置时钟频率。这是用户定义的时钟中断频率 100HZ-1000 HZ, 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。其实和前面的抢先式进程差不多, 就是多少频率来响应用户要求。我选了250HZ的。要快点的可以选1000HZ的。但是还是那句话, 一切是平衡的。机

器过快响应你，它自己的活就不知道做得好不好了。() 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。() 300 HZ (X) 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。

4.52 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，Linus本人都没话说，估计是受害不浅。我们当然不能上当，选N！提供kexec系统调用，可以不必重启而切换到另一个内核，如果需要就选择，对大多数用户来说并不需要。

4.53 kernel crash dumps

内核崩溃时，dump 运行时信息。就算 crash 了，我也不会去调试内核的core dump

4.54 (0x1000000) Physical address where the kernel is loaded

4.55 Build a relocatable kernel

官方说明（建立一个移动的内核，并增加10%的内核尺寸，运行时会被丢弃），我认为没实质性的作用

4.56 (0x100000) Alignment value to which kernel should be aligned

4.57 Support for hot-pluggable CPUs

对SMP休眠和热插拔CPU提供支持

4.58 Compat VDSO support

如果Glibc版本大于等于2.3.3就不选，否则就选上，目前的版本基本上都大于2.3.3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。

4.59 Built-in kernel command line（不选）

5、Power management and ACPI options

5.1、Power Management support

5.2、Power Management Debug Support

不想调试ACPI

这个你现在可以勾掉，不勾也没事，稍侯会在kernel-hacking这一节勾掉调试，这里也就没

5.3、Suspend to RAM and standby

待机

5.4、Hibernation (aka 'suspend to disk')

休眠

5.5、Run-time PM core functionality

5.6、ACPI (Advanced Configuration and Power Interface) Support --->

见附1

5.7、SFI (Simple Firmware Interface) Support --->

5.8、APM (Advanced Power Management) BIOS support --->

选acpi就不用apm，一般你也只用acpi&

5.9、CPU Frequency scaling --->

[*] CPU Frequency scaling

[] Enable CPUfreq debugging <== 我不需要调试 CPUfreq

< > CPU frequency translation statistics

Default CPUFreq governor (performance) ---> <== 默认用 performance 高性能的CPU 调频方式

-*- 'performance' governor

< > 'powersave' governor

< > 'userspace' governor for userspace frequency scaling

< > 'ondemand' cpufreq policy governor <== " 周期性的考察CPU 负载并自动的动态调整cpu 频率"，我只用 performance

< > 'conservative' cpufreq governor

*** CPUFreq processor drivers ***

< > Processor Clocking P-stat driver

<*> ACPI Processor P-States driver

< > AMD Mobile K6-2/K6-3 PowerNow !

< > AMD Mobile Athlon/Duron PowerNow !

< > AMD Opteron/Athlon64 PowerNow !

< > Cyrix MediaGX /NatSemi Geode Suspend Modulation

< > Intel Enhanced SpeedStep (deprecated)

< > Intel Speedstep on ICH-M chipsets (ioport interface)

< > Intel Pentium 4 clock modulation

< > Transmeta LongRun

< > VIA Cyrix III Longhaul

5.10、CPU idle PM support

5.11、Cpuidle Driver for Intel Processors

6、Bus options(PCI etc.)

6.1、PCI support (这个必须选)

6.2、PCI access mode (Any)

() BIOS

() MMConfig

() Direct

(X) Any

6.3、Read CNB20LE Host Bridge Windows

没有公共规范的芯片组，此功能已知是不完整的。如果你不知道需不需要它，请选择N

6.4、PCI Express support

如果你的主板支持PCI Express，请选择Y

6.5、PCI Express Hotplug driver 选Y

6.6、Root Port Advanced Error Reporting support

硬件驱动会负责发送错误信息

6.7、PCI Express ECRC settings control

如果怀疑，请选择N

6.8、PCIe AER error injector support 同上，选N

6.9、PCI Express ASPM control

这使得OS控制的PCI Express ASPM（活动状态电源管理）和时钟电源管理。 ASPM支持状态L0/L0s/L1，选Y

6.10、Debug PCI Express ASPM 选N

Default ASPM policy

(X) BIOS default

() powersave

() performance

6.11、Message Signaled Interrupts (MSI and MSI-X)

这使得设备驱动能够使用MIS（消息信号中断）选Y

6.12、PCI Debugging

我认为这里没有必要选

6.13、Enable PCI resource re-allocation detection

当PCI资源重新分配时，如果你需要PCI核心来检测的话，选择Y，同时你可以用pci=realloc=on和pci=realloc=off来覆盖它，如果你不确定的话，选择N

6.14、PCI Stub driver

选择Y或者M，如果你想要：当一个设备去注册其他的客户操作系统时需要保留该PCI设备

6.15、Interrupts on hypertransport devices

这将允许高速传输设备使用中断，如果不明确的话，选择Y

6.16、PCI IOV support

I / 0虚拟化是由一些设备支持的PCI功能，这使得他们能够创建虚拟设备共享其物理资源。如果不确定的话，选择N

6.17、PCI PRI support

RIP就是PCI页面请求接口，如果不确定的话，选择N

6.18、PCI PASID support 不确定的话选择N

6.19、PCI IO-APIC hotplug support 选Y

6.20、ISA support (以及之后的EISA)

查看你的主板上是否有ISA插槽。ISA是总线系统的名称，它是一个老的系统，现已被PCI取代。新的主板已经不支持它，如果你还有，选择Y，否则，选择N

6.21、NatSemi SCx200 support 选择编译为模块

6.22、One Laptop Per Child support 不选

6.23、PCEngines ALIX System Support (LED setup)

6.24、Soekris Engineering net5501 System Support (LEDS, GPIO, etc)

6.25、Traverse Technologies GEOS System Support (LEDS, GPIO, etc)

6.26、RapidIO support

RapidIO主要应用于嵌入式系统内部互连。如果你选择Y，内核中将包含支持RapidIO设备连接的驱动和设施

6.27、PCCard (PCMCIA/CardBus) support

一般笔记本电脑会配备PCCard 接口(无线网卡之类的)，看你的硬件和使用场景吧。虽然我也是NB，但我从来不用PCMCIA

16-bit PCMCIA support

Load CIS updates from userspace (EXPERIMENTAL)

32-bit CardBus support

*** PC-card bridges ***

CardBus yenta-compatible bridge support

Cirrus PD6729 compatible bridge support

i82092 compatible bridge support

6.28、Support for PCI Hotplug

支持热拔插PCI 设备

Fake PCI Hotplug driver

Compaq PCI Hotplug driver

Save configuration into NVRAM on Compaq servers

IBM PCI Hotplug driver

ACPI PCI Hotplug driver

ACPI PCI Hotplug driver IBM extensions

CompactPCI Hotplug driver

SHPC PCI Hotplug driver

7、Executable file formats / Emulations

7.1、Kernel support for ELF binaries 选择Y

7.2、Write ELF core dumps with partial segments 不选

7.3、Kernel support for a.out and ECOFF binaries 编译成模块

7.4、Kernel support for MISC binaries 编译成模块

8、Networking support

8.1、Networking options

<Y> Packet socket

<Y> Unix domain sockets

<M> Transformation user configuration interface

[] Transformation sub policy support (EXPERIMENTAL)

[] Transformation migrate database (EXPERIMENTAL)

[] Transformation statistics (EXPERIMENTAL)

<M> PF_KEY sockets

[] PF_KEY MIGRATE (EXPERIMENTAL)

[Y] TCP/IP networking

[Y] IP: multicasting

[Y] IP: advanced router

Choose IP: FIB lookup algorithm (choose FIB_HASH if unsure) (FIB_HASH) |

[Y] IP: policy routing

[Y] IP: equal cost multipath

[Y] IP: verbose route monitoring

[] IP: kernel level autoconfiguration

<M> IP: tunneling

< M> IP: GRE demultiplexer

[Y] IP: multicast routing

[] IP: multicast policy routing

[Y] IP: PIM-SM version 1 support

[Y] IP: PIM-SM version 2 support

[] IP: ARP daemon support

[Y] IP: TCP syncookie support

抵抗SYN flood 攻击，我是开发机，暂不考虑安全特性

<M> IP: AH transformation

<M> IP: ESP transformation

<M> IP: IPComp transformation

<M> IP: IPsec transport mode

<M> IP: IPsec tunnel mode

<*> IP: IPsec BEET mode

{*} Large Receive Offload (ipv4/tcp)

<M> INET: socket monitoring interface

[*] TCP: advanced congestion control --->

高级拥塞控制，如果没有特殊需求（比如无线网络）就别选了

[] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)

< > The IPv6 protocol --->

我暂时没有要支持IPV6 的需求

[] NetLabel subsystem support

NetLabel 子系统，为诸如CIPSO 与RIPSO 之类能够在分组信息上添加标签的协议提供支持，我用不到

-*- Security Marking

对网络包进行安全标记，类似于nfmark，但主要是为安全目的而设计，安全特性，我暂时不考虑

[] Timestamping in PHY devices

[] Network packet filtering framework (Netfilter) --->

我不打算使用防火墙，要用到时再编译进去

<M> The DCCP Protocol (EXPERIMENTAL) --->

-M- The SCTP Protocol (EXPERIMENTAL) --->

<M> The RDS Protocol (EXPERIMENTAL)

< > RDS over Infiniband and iWARP

< > RDS over TCP

[] RDS debugging messages

<M> The TIPC Protocol (EXPERIMENTAL) --->

<M> Asynchronous Transfer Mode (ATM)

<M> Classical IP over ATM

[] Do NOT send ICMP if no neighbour

<M> LAN Emulation (LANE) support

< > Multi-Protocol Over ATM (MPOA) support

<M> RFC1483/2684 Bridged protocols

[] Per-VC IP filter kludge

< > Layer Two Tunneling Protocol (L2TP) --->

<M> 802.1d Ethernet Bridging

[*] IGMP/MLD snooping

[] Distributed Switch Architecture support --->

<M> 802.1Q VLAN Support

[] GVRP (GARP VLAN Registration Protocol) support

< > DECnet Support

< > ANSI/IEEE 802.2 LLC type 2 Support

< > The IPX protocol

< > Appletalk protocol support

< > CCITT X.25 Packet Layer (EXPERIMENTAL)

< > LAPB Data Link Driver (EXPERIMENTAL)

< > Acorn Econet/AUN protocols (EXPERIMENTAL)

< > WAN router

< > Phonet protocols family

< > IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support (EXPERI |

[] QoS and/or fair queueing --->

通过IPRoute 切换网络设备上的Qos 策略，我不打算使用IP 路由

[] Data Center Bridging support

-*- DNS Resolver support

< > B. A. T. M. A. N. Advanced Meshing Protocol

Network testing --->

8.2、[] Amateur Radio support --->

我没有无线电

8.3、< > CAN bus subsystem support --->

8.4、< > IrDA (infrared) subsystem support --->

8.5、<M> Bluetooth subsystem support --->

8.6、< > RxRPC session sockets

8.7、- - Wireless --->

我没有使用无线网卡

8.8、< > WiMAX Wireless Broadband support --->

8.9、< > RF switch subsystem support --->

我没有RF 切换设备

8.10、< > Plan 9 Resource Sharing Support (9P2000) (Experimental) --->

8.11、< > CAIF support --->

8.12、< > Ceph core library (EXPERIMENTAL)

9、Device Drivers

9.1、Generic Driver Options --->

9.1.1、() path to uevent helper

9.1.2、[] Maintain a devtmpfs filesystem to mount at /dev

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

9.1.4、[*] Prevent firmware from being built

9.1.5、~~-*- Userspace firmware loading support~~

9.1.6、[*] Include in-kernel firmware blobs in kernel binary

9.1.7、() External firmware blobs to build into the kernel binary

9.1.8、[] Driver Core verbose debug messages

9.1.9、[] Managed device resources verbose debug messages

管理设备资源的冗长调试信息，我不需要

9.2、<*> Connector - unified userspace <-> kernelspace linker --->

内核空间与用户空间的信道

9.2.1、[*] Report process events to userspace

报告处理时间给用户空间

9.3、< > Memory Technology Device (MTD) support --->

9.4、< > Parallel port support --->

9.5、~~-*- Plug and Play support~~ --->

[] PNP debugging messages

调试信息，老规矩

9.6、[] Block devices --->

我没有想要支持的块设备，比如ramdisk，磁盘阵列，CD/DVD 刻录等，详见内部选项

9.7、[] Misc devices --->

没有需要支持的杂项设备

9.8、< > ATA/ATAPI/MFM/RLL support (DEPRECATED) --->

9.9、SCSI device support --->

< > RAID Transport Class

~~-*- SCSI device support~~

[] legacy /proc /scsi / support <=== 我没有SCSI 设备

*** SCSI support type (disk, tape, CD-ROM) ***

```

<*) SCSI disk support // 就算你用SATA，此选项也必选
< > SCSI tape support
< > SCSI OnStream SC-x0 tape support
< > SCSI CDROM support <=== 我没有SCSI 设备
< > SCSI generic support <=== 我没有SCSI 设备
< > SCSI media changer support
[ ] Probe all LUNs on each SCSI device
[ ] Verbose SCSI error reporting (kernel size +=12K) <=== 我没有SCSI 设备
[ ] SCSI logging facility
[ ] Asynchronous SCSI scanning
SCSI Transports --->
< > Parallel SCSI (SPI) Transport Attributes <=== 我没有SCSI 设备
< > FiberChannel Transport Attributes
< > iSCSI Transport Attributes
< > SAS Domain Transport Attributes
< > SRP Transport Attributes
[ ] SCSI low-level drivers --->
< > SCSI Device Handlers --->
< > OSD-Initiator library

```

9.10、<M> Serial ATA and Parallel ATA drivers --->

```

[*] Verbose ATA error reporting
[*] ATA ACPI Support
[ ] SATA Port Multiplier support <=== 我只有一枚SATA 设备，没有使用 多路SATA/SATA Hub 的需求。Port
Multiplier 是南桥芯片提供的一种支持多块SATA 设备，并共享总带宽的技术。
<*) AHCI SATA support
< > Platform AHCI SATA support
< > Initio 162x SATA support
< > Silicon Image 3124/3132 SATA support
[*] ATA SFF support // 选择自己硬件对应的驱动即可
< > ServerWorks Frodo / Apple K2 SATA support
<*) Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA support // Intel ICH，G 系列chipset driver
< > Marvell SATA support
< > 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
< > ALi PATA support
< > AMD/Nvidia PATA support <=== 我用的是SATA，取消PATA 支持

```

- < > ARTOP 6210/6260 PATA support
- < > ATI PATA support
- < > CMD64x PATA support
- < > CS5510/5520 PATA support
- < > CS5530 PATA support
- < > CS5536 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 <=== 我用的是SATA，取消PATA 支持
- < > Intel PATA old PIIX support <=== 我用的是SATA，取消PATA 支持
- < > NETCELL Revolution RAID support
- < > Nat Semi NS87410 PATA support
- < > Nat Semi NS87415 PATA support
- < > Older Promise PATA controller support
- < > PC Tech RZ1000 PATA support
- < > SC1200 PATA support
- < > SERVERWORKS OSB4/CSB5/CSB6/HT1000 PATA support
- < > Promise PATA 2027x support
- < > CMD / Silicon Image 680 PATA support
- < > SiS PATA support
- < > VIA PATA support
- < > Winbond SL82C105 PATA support
- < > Intel SCH PATA support <=== 我用的是SATA，取消PATA 支持

9.11、[] Multiple devices driver support (RAID and LVM) --->

暂时没有要使用Raid（磁盘阵列）和LVM（逻辑卷管理器，添加，删除逻辑分区）的需求

9.12、[] Fusion MPT device support --->

9.13、IEEE 1394 (FireWire) support --->

9.14、< > I2O device support --->

9.15、[] Macintosh device drivers --->

Mac 系统硬件设备驱动，没什么好说的，关

9.16、*- Network device support --->

- < > Dummy net driver support
- < > Bonding driver support
- < > EQL (serial line load balancing) support

```

< > Universal TUN/TAP device driver support
< > Virtual ethernet pair device
< > General Instruments Surfboard 1000
< > ARCnet support --->
-*- PHY Device support and infrastructure ---> <=== PHY (物理层控制芯片) , 里面没有我对应的硬件

[ ] Ethernet (10 or 100Mbit) ---> <=== 如果你是百 M 卡, 请自行选择
[*] Ethernet (1000 Mbit) ---> // 选择自己对应的硬件
[ ] Ethernet (10000 Mbit) ---> <=== 如果你是万M 卡, 请自行选择
< > Token Ring driver support ---> <=== IBM 的令牌环网, 用以太网的忽略
[ ] Wireless LAN ---> <=== 不用无线网络
    *** Enable WiMAX (Networking options) to see the WiMAX drivers ***
    USB Network Adapters --->
[ ] Wan interfaces support --->
< > FDDI driver support <=== 光纤卡驱动, 相信没几个人能用上这玩意
< > PPP (point-to-point protocol) support
< > SLIP (serial line) support
[ ] Fibre Channel driver support
[ ] Network console logging support
[ ] VMware VMXNET3 ethernet driver

```

9.17、[] ISDN support --->

9.18、< > Telephony support --->

9.19、Input device support --->

```

-*- Generic input layer (needed for keyboard, mouse, ...)
-*- Support for memoryless force-feedback devices
< > Polled input device skeleton <=== 一种周期性轮询硬件状态的驱动, 去掉后没什么副作用
    *** Userland interfaces ***
-*- Mouse interface
[ ] Provide legacy /dev /psaux device
(1024) Horizontal screen resolution
(768) Vertical screen resolution
< > Joystick interface
< * > Event interface // 将输入设备的事件存储到/dev /input/eventX 供应用程序读取
< > Event debugging
    *** Input Device Drivers ***
-*- Keyboards --->
[*] Mice --->
[ ] Joysticks/Gamepads ---> <=== 游戏设备
[ ] Tablets ---> <=== 平板PC
[ ] Touchscreens ---> <=== 触摸屏
[ ] Miscellaneous devices ---> <=== 杂七杂八的驱动, 扬声器, 笔记本扩展按键等
Hardware I/O ports --->

```

9.20、Character devices --->

-*- Virtual terminal

[*] Support for binding and unbinding console drivers // 在某些系统上可以使用多个控制台驱动程序(如 framebuffer 控制台驱动程序), 该选项使得你可以选择其中之一, 我一般只用默认的虚拟终端

[] /dev /kmem virtual device support <=== 支持/dev /kmem 设备, 很少用

[] Non-standard serial port support <=== 我没有非标准的串口设备

Serial drivers --->

< > 8250/16550 and compatible serial support <=== 兼容一些老式的串口设备, 我一般不用

*** Non-8250 serial port support ***

< > Digi International NEO PCI Support

-*- Unix98 PTY support

[] Support multiple instances of devpts

[] Legacy (BSD) PTY support

< > IPMI top-level message handler --->

<*> Hardware Random Number Generator Core support

< > Timer IOMEM HW Random Number Generator support

<*> Intel HW Random Number Generator support

< > AMD HW Random Number Generator support <=== 我是intel 主板

< > AMD Geode HW Random Number Generator support <=== 我是intel 主板

< > VIA HW Random Number Generator support <=== 我是intel 主板

< > /dev /nvram support <=== 直接存取CMOS, 太危险, 关

< > Siemens R3964 line discipline

< > Applicom intelligent fieldbus card support

< > ACP Modem (Mwave) support

< > NatSemi PC8736x GPIO Support

< > NatSemi Base GPIO Support

< > AMD CS5535/CS5536 GPIO (Geode Companion Device)

< > RAW driver (/dev /raw/rawN)

[*] HPET - High Precision Event Timer

[] Allow mmap of HPET

< > Hangcheck timer

9.21、{M} I2C support --->

感知硬件状态, 比如温度, 风扇转速

9.22、[] SPI support --->

9.23、PPS support --->

9.24、[] GPIO Support --->

9.25、{*} Power supply class support --->

9.26、{*} Hardware Monitoring support --->

9.27、-*- Generic Thermal sysfs driver --->

9.28、[*] Watchdog Timer Support --->

系统监视程序，我一般不用

9.29、Sonics Silicon Backplane --->

9.30、< > Multimedia support --->

9.31、[] Voltage and Current Regulator Support --->

9.32、< > Multimedia support --->

< > /dev /agpgart (AGP Support) ---> < --- virtualbox不支持虚拟独立显卡

-*- VGA arbitration

(16) Maximum number of GPU

[] Laptop Hybrid Graphics - GPU switch support

<*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) --->

<> Lowlevel video output switch controls

<> Support for frame buffer devices --->

[] Backlight & LCD device support ---> < --- 支持背光设置，比如pda等。我用不到

Display device support --->

Console display driver support --->

[] Enable Scrollback Buffer in System RAM

9.33、< > Sound card support --->

用不到声卡

9.34、[] HID Devices --->

用不到人力工程学设备

9.35、[] USB support ---> <=== 这个选项，对于跑物理机建议 开启，因为有可能你的键盘是USB 的，我是跑虚拟机的，所以关了

9.36、< > MMC/SD/SDIO card support --->

9.37、< > Sony MemoryStick card support (EXPERIMENTAL) --->

9.38、[] LED Support ---> <=== 发光二极管，应该是跟显示器相关的驱动，由于我运行 的是虚拟机，所以我选择关闭

9.39、[] Accessibility support --->

9.40、< > InfiniBand support --->

9.41、[*] EDAC (Error Detection And Correction) reporting ---> // 硬件故障reporting

9.42、<*> Real Time Clock --->

9.43、[*] DMA Engine support --->

9.44、[] Auxiliary Display support --->

9.45、< > Userspace I/O drivers --->

9.46、TI VLYNQ --->

9.47、[] Staging drivers --->

9.48、[] X86 Platform Specific Device Drivers ---> <=== 一些笔记本的驱动，我没有相关设备

10、Firmware Drivers

< > BIOS Enhanced Disk Drive calls determine boot disk

< > BIOS update support for DELL systems via sysfs

< > Dell Systems Management Base Driver

[*] Export DMI identification via sysfs to userspace // 将BIOS 里的DMI 区信息导出到用户空间，部分系统管理工具可能会用到

[] iSCSI Boot Firmware Table Attributes

11、File systems

< > Second extended fs support

< > Ext3 journalling file system support <=== 我使用的是ext4 FS

<*> The Extended 4 (ext4) filesystem

[] Enable ext4dev compatibility

[*] Ext4 extended attributes

[*] Ext4 POSIX Access Control Lists

[] Ext4 Security Labels <=== 取消 SELinux 支持

[] JBD (ext3) debugging support

[] JBD2 (ext4) debugging support

< > Reiserfs support

< > JFS filesystem support

< > XFS filesystem support

< > OCFS2 file system support

[*] Dnotify support

[*] Inotify support for userspace

[] Quota support <=== 磁盘配额支持，限制某个用户或者某组用户的磁盘占用空间，暂时没这个需求，你可以把它编译成模块

< > Kernel automounter support

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

< > FUSE (Filesystem in Userspace) support

Caches --->

CD-ROM/DVD Filesystems --->

< > ISO 9660 CDROM file system support <=== 在虚拟机内，我不用CDROM

< > UDF file system support


```

    DOS/FAT/NT Filesystems    --->
< > MSDOS fs support <=== 我没有微软fs 的设备
< > VFAT (Windows-95) fs support    <=== 我没有微软fs 的设备
< > NTFS file system support
    Pseudo filesystems    --->
[ ] Miscellaneous filesystems    ---> <=== 如果你没有其他FS 的支持需求, 关
[*] Network File Systems    ---> <=== 如果你没有NFS 的支持需求, 关
    Partition Types    --->
[ ] Advanced partition selection <=== 如果不是和其他系统共存, 可以不选
-* Native language support    ---> // 选上Chinese

```

12、Kernel hacking

```

[ ] Show timing information on printk    <=== 在printk 的输出中包含时间信息, 可以用来分析内核启动过程各步骤所用时间, 我不需要debug 内核
[ ] Enable __deprecated logic
[*] Enable __must_check logic
(2048) Warn for stack frames larger than (needs gcc 4.4)
[ ] Magic SysRq key <=== 一种通过快捷键控制系统方式, 除非你非常清楚这个选项, 官方不推荐选择
[ ] Enable unused/obsolete exported symbols
[ ] Debug Filesystem
[ ] Run 'make headers_check' when building vmlinux
[ ] Kernel debugging <=== 内核调试, 关
[ ] Enable SLUB performance
statistics

[ ] Compile the kernel with frame pointers    <=== 还是跟内核开发有关
[ ] Delay each boot printk message by N milliseconds
< > torture tests for RCU
[ ] Check for stalled CPUs delaying RCU grace periods
< > Self test for the backtrace code
[ ] Force extended block device numbers and spread them
[ ] Fault-injection framework
[ ] Latency measuring infrastructure
[*] Sysctl checks
[ ] Tracers    --->
[ ] Remote debugging over FireWire early on boot    <=== 启动过程中, 允许远程调试内核
[ ] Enable dynamic printk ( ) support
[ ] Enable debugging of DMA-API usage
[ ] Sample kernel code    --->
[ ] Filter access to /dev /mem
[ ] Enable verbose x86 bootup info messages <=== 在内核镜像解压缩阶段输出启动信息, 关闭后相当于无声启动 (Silent Bootup )
-* Early printk
[ ] Early printk via EHCI debug port    <=== 允许printk 通过EHCI 调试端口输出内核日志, 调试的一律关
[ ] Use 4Kb for kernel stacks instead of 8Kb
[ ] Enable IOMMU stress-test mode

```

IO delay type (port 0x80 based port-I/O delay [recommended]) --->
[*] Allow gcc to uninline functions marked 'inline'

13、Security options

安全特性，我选择全关，当然，这些选项不会影响你的日常开发，办公

[] Enable access key retention support <=== 关闭
[] Enable different security models <=== 关闭
[] Enable the securityfs filesystem
[] File POSIX Capabilities <=== 关闭
[] Integrity Measurement Architecture(IMA)

< > Cryptographic API ---> // 加密API，这部分选项会根据此前的优化自动调整，默认即可

14、Virtualization

我的系统已经运行在虚拟机中，不需要再支持虚拟化

15、Library routines

库子程序，这部分选项会根据此前的优化自动调整，默认即可

内核的编译

在完成内核的裁减之后，内核的编译就是一个非常简单的过程。你只要执行以下几条命令就行：

1、make clean

这条命令是在正式编译你的内核之前先把环境给清理干净。有时你也可以用 make realclean或 make mrproper 来彻底清除相关依赖，保证没有不正确的.o 文件存在。

2、make dep

这条命令是编译相关依赖文件。

3、make zImage

这条命令就是最终的编译命令。有时你可以直接用 make(2.6.X 版本上用)或 make bzImage (给 PC 机编译大内核时用)

4、make install

这条命令可以把相关文件拷贝到默认的目录。当然在给嵌入式设备编译时这步可以不要。因为具体的内核安装还需要你手工进行。