Linux Configuration V6.6

0.1 Einführung

Dieses Dokument dient zur Beschreibung von diversen Einstellungen bei der Konfiguration mittels `make menuconfig` unter Linux.
Es wird nicht näher darauf eingegangen, wie der Kernel kompiliert wird oder welche Voreinstellungen, Programme etc. zum Kompilieren benötigt werden.
Zu Beginn der jeweiligen Konfigurationszeile wird der Standardwert (Default) angezeigt. Mein Vorschlag folgt danach.
Z. B. bei CONFIG_WERROR [=n] **[Y]**
Hier ist der Standarwert ein Nein [n], meine persönliche Einstellung ein Ja [Y].

Wenn Sie Tippfehler finden oder Korrekturen wünschen, dann schicken Sie dies mit Erläuterungen und dem Hinweis auf die obenstehende Version V6.6 an: oe1tkt@gmail.com

1 General setup →

1.1 Compile also drivers which will not load

CONFIG_COMPILE_TEST [=n] [ ]
Kompilieren Sie auch Treiber, die nicht geladen werden können
Einige Treiber können auf einer anderen Plattform kompiliert werden als auf der, für die sie gedacht sind. Obwohl sie dort nicht geladen werden können (oder selbst wenn sie geladen werden können, können sie aufgrund fehlender Hardware-Unterstützung nicht verwendet werden), möchten Entwickler, im Gegensatz zu Distributoren, solche Treiber vielleicht trotzdem kompilieren und testen.

1.2 Compile the kernel with warnings as errors

CONFIG_WERROR [=n] **[Y]**
Den Kernel mit Fehlermeldungen bei Warnungen kompilieren
Ein Build sollte keine Compiler-Warnungen ausgeben, dies aktiviert die Flags '-Werror' (für C) und '-Dwarnings' (für Rust) um diese Regel standardmäßig zu setzen. Bestimmte Warnungen von anderen Tools z. B. der Linker könnte mit dieser Option Fehler generieren. Deaktivieren ist sinnvoll, wenn Sie einen neuen (oder sehr alten) Compiler bzw. Linker mit seltenen, ungewöhnlichen Warnungen haben. Haben Sie auf Ihrer Architektur Probleme, dann müssen Sie diese Konfiguration deaktivieren, um den Kernel erfolgreich zu bauen. Im Zweifelsfall sagen sie Y für Ja.

1.3 Local version – append to kernel release

CONFIG_LOCALVERSION [=] [ ]
Lokale Version – an die Kernelversion anhängen
Type: string
Hängen Sie eine zusätzliche Zeichenkette an das Ende Ihrer Kernelversion an.
Dies wird angezeigt, wenn Sie z. B. `uname` eingeben. Die hier angegebene Zeichenfolge wird an den Inhalt von einem Dateinamen mit `localverion*` als Objekt und im Quellbaum, in dieser Reihenfolge angezeigt. Die Zeichenkette darf maximal 64 Zeichen lang sein.

1.4 Automatically append version information to the version string

CONFIG_LOCALVERSION_AUTO [=y] **[Y]**
Dies versucht automatisch festzustellen, ob der aktuelle Baum ein Release-Tree ist, indem es nach **Git**-Tags sucht, die zur aktuellen Top-of-Tree-Revision gehören.
Eine Zeichenkette des Formats `-gxxxxxxxx` wird der lokalen Version hinzugefügt, wenn ein git-basierter Baum gefunden wird. Die so erzeugte Zeichenkette wird nach allen passenden „localversion*"-Dateien und nach dem in CONFIG_LOCALVERSION eingestellten Wert angehängt. (Die hier tatsächlich verwendete Zeichenkette sind die ersten 12 Zeichen, die durch die Ausführung des Befehls erzeugt werden:

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

der innerhalb des Skripts „scripts/setlocalversion" ausgeführt wird.)

1.5 Build ID Salt

CONFIG_BUILD_SALT [=] [ ]
Type: string
Dies wird verwendet, um die Binaries und ihre Debug-Infos zu verknüpfen. Wenn diese Option gesetzt ist, dann wird dieser Wert in die Berechnung der Build-ID einbezogen. Wird von Distributionen verwendet, die sicherstellen wollen, dass es eineindeutige IDs zwischen verschiedenen Builds gibt. Üblicherweise brauchen wir das nicht.

1.6 Kernel compression mode →

Der Linux-Kernel ist eine Art selbstextrahierende, ausführbare Datei. Es stehen mehrere Kompressionsalgorithmen zur Verfügung, die sich in Effizienz, Kompressions- und Dekompressionsgeschwindigkeit unterscheiden. Die Komprimierungsgeschwindigkeit ist nur bei der Erstellung eines Kernels relevant. Die Dekomprimierungsgeschwindigkeit ist bei jedem Systemstart von Bedeutung. (Eine ältere Version dieser Funktionalität (nur bzip2) für 2.4 wurde von Christian Ludwig bereitgestellt) Hohe Komprimierungsoptionen sind vor allem für Benutzer nützlich, die wenig Festplattenplatz zur Verfügung haben (embedded systems), für die aber die Ram-Größe weniger wichtig ist.
Überblick: Gzip werden von den älteren Kernelversionen unterstützt,
Arch Linux (since Linux/x86 5.9.0) Standard: ZSTD (former: XZ since 4.14.4, predecessor GZIP,XZ)
Debian 11.6: XZ
@TODO Weitere Linux Distributionen

1.6.1 Gzip

CONFIG_KERNEL_GZIP [=n] [ ]
Die alte und bewährte gzip-Kompression. Sie bietet ein gutes Gleichgewicht zwischen Kompressionsrate und Dekompressionsgeschwindigkeit.

1.6.2 Bzip2

CONFIG_KERNEL_BZIP2 [=n] [ ]
Die Kompressionsrate und auch die Geschwindigkeit der ist durchschnittlich. Die Geschwindigkeit der Dekomprimierung ist die langsamste. Größe des Kernels ist etwa 10 % kleiner im Vergleich zu GZIP. Es benötigt auch einen großen Speicherbereich, bei modernen Kerneln benötigt man zumindest 8 MB RAM oder mehr beim Booten.

1.6.3 LZMA

CONFIG_KERNEL_LZMA [=n] [ ]
Dieser Kompressionsalgorithmus hat die höchste Komprimierung. Die Geschwindigkeit der Dekomprimierung liegt zwischen GZIP und BZIP2. Komprimierung ist die langsamste. Kernelgröße beträgt etwa 33 % weniger als mit GZIP.

1.6.4 XZ

CONFIG_KERNEL_XZ [=n] [ ]
XZ verwendet den LZMA2-Algorithmus und befehlssatzspezifische BCJ-Filter, die das Komprimierungsverhältnis des ausführbaren Codes verbessern können. Die Größe des Kernels ist mit XZ im Vergleich zu GZIP etwa 30 % kleiner. Auf Architekturen, für die es einen BCJ-Filter gibt (i386, x86_64, ARM, IA-64, PowerPC und SPARC), erzeugt XZ einen um einige Prozent kleineren Kernel als einfaches LZMA. Die Geschwindigkeit ist in etwa die gleiche wie bei LZMA: Die Dekomprimierungsgeschwindigkeit von XZ ist besser als die von bzip2, aber schlechter als die von gzip und LZO. Die Komprimierung ist langsam.

1.6.5 LZO

CONFIG_KERNEL_LZO [=n] [ ]
Kompressionsrate ist die schlechteste aller anderen. Kernelgröße ist etwa 10 % größer als GZIP. Jedoch ist die Geschwindigkeit beim Komprimieren und Dekomprimieren die höchste.

1.6.6 LZ4

CONFIG_KERNEL_LZ4 [=n] [ ]
LZ4 ist eine LZ77-Typ-Komprimierung mit einer festen, byte-orientierten Enkodierung.
Siehe auch `http://code.google.com/p/lz4`.
Komprimierungsverhältnis ist noch schlechter als LZO. 8 % größere Kernelgröße als bei LZO. Dekomprimierung ist jedoch von der Geschwindigkeit her schneller als LZO.

1.6.7 ZSTD

CONFIG_KERNEL_ZSTD [=y] **[Y]**
ZSTD ist ein Komprimierungsalgorithmus, der auf eine Zwischenkomprimierung mit schneller Dekomprimierungsgeschwindigkeit abzielt. Er komprimiert besser als GZIP und dekomprimiert etwa so schnell wie LZO, ist aber langsamer als LZ4. Sie benötigen mindestens 192 KB RAM oder mehr zum Booten. Das Kommandozeilenprogramm `zstd` ist für die Komprimierung erforderlich.

1.7 Default init path

CONFIG_DEFAULT_INIT [=] [ ]
Diese Option legt den Standard-Init-Pfad für das System fest, wenn in der Kernel-Befehlszeile keine solche init=-Option übergeben wird. Wenn der angeforderte Pfad nicht vorhanden ist, wird trotzdem versucht, weitere Orte zu finden (z. B. /sbin/init usw.). Wenn dieser Pfad leer ist, wird einfach die Fallback-Liste verwendet, wenn init= nicht übergeben wird.

1.8 Default hostname

CONFIG_DEFAULT_HOSTNAME [=archlinux] **[=archlinux]**
Diese Option legt den Standard-Hostnamen des Systems fest, noch bevor der Userspace das Kommando sethostname(2) aufruft. Der Kernel verwendet hier traditionell "(none)", Sie möchten vielleicht eine andere Voreinstellung verwenden, um ein minimales System mit weniger Konfiguration benutzbar zu machen.

1.9 System V IPC

CONFIG_SYSVIPC [=y] **[Y]**
Die Inter-Prozess-Kommunikation IPC ist eine Zusammenstellung aus Bibliotheksfunktionen (libraries) und Systemaufrufen die Prozesse (laufende Programme) synchronisiert und Daten untereinander austauschen kann. Generell ist das eine gute Sache, einige Programme würden auch nicht funktionieren wenn Sie hier kein Y (ja) setzen.

1.10 POSIX Message Queues

CONFIG_POSIX_MQUEUE [=y] **[Y]**
Die POSIX-Variante der Nachrichtenwarteschlangen (message queues) ist ein Teil der IPC. In POSIX-Nachrichtenwarteschlangen hat jede Nachricht eine Priorität, die über die Reihenfolge des Empfangs durch einen Prozess entscheidet. Wenn Sie Programme kompilieren und ausführen wollen, die z. B. für Solaris geschrieben wurden und die POSIX-Warteschlangen (Funktionen mq_) verwenden, sagen Sie hier Y. POSIX-Nachrichtenwarteschlangen sind via Dateisystem als „mqueue“ sichtbar und können irgendwo eingehängt werden, wenn Sie Dateisystemoperationen auf Nachrichtenwarteschlangen durchführen wollen.

1.11 General notification queue

CONFIG_WATCH_QUEUE [=y] **[Y]**
Dies ist eine allgemeine Benachrichtigungswarteschlange für den Kernel, um Ereignisse an den Userspace weiterzuleiten, indem sie in Pipes gesplittet werden. Sie kann in Verbindung mit Watches für Schlüssel-/Schlüsseländerungsbenachrichtigungen (key/keyring) und Gerätebenachrichtigungen verwendet werden. Bemerkung: Bei Debian Bullseye ist dies nicht gesetzt (N).

1.12 Enable process_vm_readv/writev syscalls

CONFIG_CROSS_MEMORY_ATTACH [=y] **[Y]**
Die Aktivierung dieser Option fügt die Systemaufrufe process_vm_readv und process_vm_writev hinzu, die es einem Prozess mit den richtigen Rechten ermöglichen, direkt aus dem Adressraum eines anderen Prozesses zu lesen oder in diesen zu schreiben. Weitere Einzelheiten finden Sie in der Manpage.

1.13 uselib syscall (for libc5 and earlier)

CONFIG_USELIB [=n] **[N]**
Diese Option schaltet den uselib-Systemaufruf ein, der im dynamic-Linker von libc5 und früher verwendet wird. Das aktuelle glibc verwendet diesen Systemaufruf nicht mehr, deshalb kann man diese Option ausschalten wenn sie keine Programme mehr verwenden, die auf libc5 (oder früher) compiliert wurden.
Bemerkung: Debian Bullseye verwendet dies noch (Y).

1.14 Auditing support

CONFIG_AUDIT [=y] **[Y]**
Aktivieren Sie eine Überwachungsinfrastruktur, die mit einem anderen Kernel-Subsystem verwendet werden kann, wie z. B. SELinux (das dies für die Protokollierung der Ausgabe von avc-Nachrichten benötigt). Die Systemaufrufüberprüfung ist auf Architekturen, die sie unterstützen, enthalten.

1.15 IRQ subsystem →

Über diese Schnittstelle kann man Funktionen und Parameter für den Kernelbau auswählen. Merkmale können entweder eingebaut, modularisiert oder ignoriert werden. Parameter müssen als dezimale oder hexadezimale Zahlen oder als Text eingegeben werden.

1.15.1 Expose irq internals in debugfs

CONFIG_GENERIC_IRQ_DEBUGFS [=n] **[N]**
Legt interne Zustandsinformationen über debugfs offen. Hauptsächlich für Entwickler und zur Fehlersuche bei schwer zu diagnostizierenden Interrupt-Problemen.

1.16 Timers subsystem →

1.16.1 Timer tick handling →

Sie müssen aus den folgenden drei Möglichkeiten eine wählen:

1.16.1.1 Periodic timer ticks (constant rate, no dynticks)

CONFIG_HZ_PERIODIC [=n] **[N]**
Diese Option sorgt dafür, dass der Tick periodisch mit einer konstanten Rate läuft, auch wenn die CPU ihn nicht braucht.

1.16.1.2 Idle dynticks system (tickless idle)

CONFIG_NO_HZ_IDLE [=n] **[N]**
Diese Option ermöglicht ein tickloses idle-System (Leerlaufsystem): Timer-Interrupts werden nur bei Bedarf ausgelöst, wenn das System im Leerlauf ist. Dies ist v.a. zum Energiesparen interessant.

1.16.1.3 Full dynticks system (tickless)

CONFIG_NO_HZ_FULL [=y] **[Y]**
Diese Option ermöglicht ein tickloses idle-System (Leerlaufsystem): Timer-Interrupts werden nur bei Bedarf ausgelöst, wenn das System im Leerlauf ist. Dies ist v.a. zum Energiesparen interessant.
Wird bei Linux-Distributionen ausgewählt.

1.16.2 Force user context tracking

CONFIG_CONTEXT_TRACKING_USER_FORCE [=n] **[N]**
Die wichtigste Voraussetzung für das Funktionieren von Full-Dynticks ist die Unterstützung des Subsystems zur Verfolgung des Benutzerkontextes. Es gibt aber auch noch andere Abhängigkeiten, die erfüllt werden müssen, damit die vollständigen Dynticks funktionieren.
Diese Option dient zum Testen, wenn eine Systemarchitektur das Backend für die Benutzerkontextverfolgung implementiert, aber noch nicht alle Anforderungen erfüllt, um die volle Dynticks-Funktion zu ermöglichen. Ohne die vollständigen Dynticks gibt es keine Möglichkeit, die Unterstützung für die Benutzerkontextverfolgung und die Teilsysteme, die darauf angewiesen sind, zu testen: RCU Userspace extended quiescent state und tickless cputime accounting. Diese Option kommt mit dem Fehlen des vollständigen dynticks-Subsystems zurecht, indem sie die Benutzerkontextverfolgung auf allen CPUs im System erzwingt.

Sagen Sie nur dann ja (Y), wenn Sie an der Entwicklung eines Architektur-Backends für die Benutzerkontextverfolgung arbeiten. Sagen Sie ansonsten N, da diese Option einen Overhead mit sich bringt, den Sie in der Praxis nicht haben wollen.

1.16.3 Old Idle dynticks config

CONFIG_NO_HZ [=y] **[N]** *Alte Leerlauf-Dynticks-Konfiguration*
Dies ist der alte Konfigurationseintrag, der Dynticks im Leerlauf aktiviert. ~~Wir behalten ihn noch eine Weile bei, um die Abwärtskompatiblität mit älteren Konfigurationsdateien zu gewährleisten.~~

1.16.4 High Resolution Timer Support

CONFIG_HIGH_RES_TIMERS [=y] **[Y]**
Unterstützung von Timern mit hoher Auflösung
Diese Option aktiviert die Unterstützung hochauflösender Timer. Wenn ihre Hardware dazu nicht in der Lage ist, erhöht diese Option nur die Größe des Kernel-Images.

1.16.5 Clocksource watchdog maximum allowable skew

CONFIG_CLOCKSOURCE_WATCHDOG_MAX_SKEW_US [=100] **[100]**
Maximal zulässige Abweichung der Watchdog-Taktquelle
Geben Sie den maximal zulässigen Wert für den Watchdog-Versatz in Mikrosekunden an, bevor die Clocksource als instabil gemeldet wird. Der Standardwert basiert auf einem Watchdog-Intervall von einer halben Sekunde und der maximalen Frequenzdrift von NTP von 500 Teilen pro Million. Wenn die Clocksource gut genug für NTP ist, ist sie auch gut genug für den Watchdog der Clocksource!
Bereich (Range): 50 – 1000

1.17 BPF subsystem →

Berkeley Packet Filter, Firewall-Filtertechnik im Kernel

1.17.1 Enable bpf() system call

CONFIG_BPF_SYSCALL [=y] **[Y]**
Aktivieren Sie den Systemaufruf bpf(), der es ermöglicht, BPF-Programme und -Maps über Dateideskriptoren zu manipulieren.

1.17.2 Enable BPF Just In Time compiler

CONFIG_BPF_JIT [=y] **[Y]**
BPF-Programme werden normalerweise von einem BPF-Interpreter verarbeitet. Diese Option ermöglicht es dem Kernel, nativen Code zu erzeugen, wenn ein Programm in den Kernel geladen wird. Dadurch wird die Verarbeitung von BPF-Programmen erheblich beschleunigt.
Beachten Sie, dass ein Administrator diese Funktion durch Ändern aktivieren sollte:

```
/proc/sys/net/core/bpf_jit_enable
/proc/sys/net/core/bpf_jit_harden (optional)
/proc/sys/net/core/bpf_jit_kallsyms (optional)
```

1.17.2.1 Permanently enable BPF JIT and remove BPF interpreter

CONFIG_BPF_JIT_ALWAYS_ON [=y] [**Y**]
Aktiviert BPF JIT und entfernt den BPF-Interpreter um spekulative Ausführungen von BPF-Anweisungen durch den Interpreter zu verhindern. Wenn CONFIG_BPF_JIT_ALWAYS_ON eingeschaltet ist, dann wird `/proc/sys/net/core/bpf_jit_enable` permanent auf 1 gesetzt, alle Versuche diese Einstellung auf andere Werte zu legen wird mit einem Fehler zurückgewiesen.

1.17.3 Disable unprivileged BPF by default

CONFIG_BPF_UNPRIV_DEFAULT_OFF [=y] [**Y**]
Deaktiviert die unprivilegierte BPF standardmäßig, indem der entsprechende Eintrag `/proc/sys/kernel/unprivileged_bpf_disabled` auf 2 gesetzt wird. Ein Administrator kann sie immer noch wieder aktivieren, indem er sie später auf 0 setzt, oder sie dauerhaft deaktiviert, indem er sie auf 1 setzt (von wo aus kein weiterer Übergang auf 0 mehr möglich ist).
Unprivilegierte BPF könnte verwendet werden, um bestimmte potenzielle Seitenkanalschwachstellen für spekulative Ausführung auf nicht gemilderter betroffener Hardware auszunutzen. Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollen, antworten Sie mit Y.

1.17.4 Preload BPF file system with kernel specific program and map iterators →

BPF_PRELOAD [=n] [**N**]
Dadurch wird ein Kernelmodul mit mehreren eingebetteten BPF-Programmen erstellt, die als für den Menschen lesbare Dateien in den BPF-FS-Einhängepunkt eingefügt werden, was bei der Fehlersuche und der Untersuchung von BPF-Programmen und -Maps nützlich ist.

1.17.4.1 bpf_preload kernel module

Dies ist nur sichtbar wenn der übergeordnete Punkt aktiviert ist.
CONFIG_BPF_PRELOAD_UMD [=m] [ ]
Dadurch wird ein Kernelmodul mit mehreren eingebetteten BPF-Programmen erstellt, die als für den Menschen lesbare Dateien in den BPF-FS-Einhängepunkt eingefügt werden, was bei der Fehlersuche und der Untersuchung von BPF-Programmen und -Maps nützlich ist.

1.17.5 Enable BPF LSM Instrumentation

CONFIG_BPF_LSM [=y] [**Y**]
Ermöglicht die Instrumentierung der Sicherheitshaken mit BPF-Programmen zur Implementierung dynamischer MAC- und Prüfungsrichtlinien. Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollten, antworten Sie mit N.

1.18 Preemption Model (Preemptible Kernel (Low-Latency Desktop)) →

Eingestellt auf : Low-Latency, d.h. nur kleine Verzögerungen beim Modell des Multitaskings. Es gibt drei Einstellungen:

1.18.1 No Forced Preemption (Server)

CONFIG_PREEMPT_NONE [=n] [**N**]
Das war das traditionelle Linux Modell der Unterbrechungen, das sich auf den Durchsatz konzentrierte. Wird vor allem für den Server-Einsatz verwendet. Es gibt durchaus gute Performance für die Latenz, jedoch keine Garantie dafür und es kann zu zufälligen, längeren Verzögerungszeiten kommen.

Für einen Serverbetrieb wird diese Einstellung empfohlen, damit der maximale Durchsatz an Rechenleistung entsteht.

1.18.2 Voluntary Kernel Preemption (Desktop)

CONFIG_PREEMPT_VOLUNTARY [=n] [**N**]
Diese Einstellung reduziert die Latenz des Kernels durch zusätzliche „explizite Unterbrechungspunkte„ im

Kernel. Diese neuen Unterbrechungspunkte wurden ausgewählt, um die maximale Latenz beim neuerlichen Zuordnen des Schedulers zu reduzieren und dadurch schnelle Reaktionszeiten der Applikationen zu gewährleisten. – Auf Kosten eines geringeren Durchsatzes wird dies erreicht.

1.18.3 Preemptible Kernel (Low-Latency Desktop)

CONFIG_PREEMPT [=y] **[Y]**
Bei dieser Einstellung wird die Latenz des Kernels weiter erniedrigt indem der gesamte Code des Kernels (keine kritischen, geschützten Bereiche) unterbrechbar gemacht wird. Dadurch wird ein reibungsloses Arbeiten mit Applikationen aus Nutzersicht erreicht, sogar unter Volllast. Wähle diese Einstellung, wenn man einen Desktop oder ein Embedded-System mit einer Latenz im Millisekundenbereich möchte. Natürlich geht diese Einstellung mit einem leicht geringerem Durchsatz an Rechenleistung einher.

1.19 Preemtion behaviour defined on boot

CONFIG_PREEMPT_DYNAMIC [=y] **[Y]**
Diese Option ermöglicht es, das Präemptionsmodell über den Kernel-Kommandozeilenparameter zu definieren und damit das während der Kompilierung definierte Standard-Präemptionsmodell außer Kraft zu setzen. Diese Funktion ist vor allem für Linux-Distributionen interessant, die eine vorgefertigte Kernel-Binärdatei bereitstellen, um die Anzahl der angebotenen Kernel-Varianten zu reduzieren und dennoch verschiedene Anwendungsfälle zu ermöglichen.

Der Laufzeit-Overhead ist vernachlässigbar, wenn HAVE_STATIC_CALL_INLINE aktiviert ist, aber wenn Laufzeit-Patching für die spezifische Architektur nicht verfügbar ist, sollte der potenzielle Overhead in Betracht gezogen werden. Interessant wird es, wenn derselbe vorgefertigte Kernel sowohl für Server- als auch für Desktop-Workloads verwendet werden soll.

1.20 Core Scheduling for SMT

CONFIG_SCHED_CORE [=y] **[Y]**
Kern-Scheduling für SMT

Diese Option ermöglicht Core Scheduling, ein Mittel zur koordinierten Auswahl von Aufgaben zwischen SMT-Geschwistern. Wenn diese Option aktiviert ist - siehe prctl (PR_SCHED_CORE) - stellt die Aufgabenauswahl sicher, dass alle SMT-Geschwister eine Aufgabe aus der gleichen „Kerngruppe“ ausführen und den Leerlauf erzwingen, wenn keine passende Aufgabe gefunden wird. Diese Funktion wird unter anderem verwendet:

- Entschärfung einiger (nicht aller) SMT-Seitenkanäle;
- Begrenzung der SMT-Interferenz zur Verbesserung des Determinismus und/oder der Leistung.

SCHED_CORE ist standardmäßig deaktiviert. Wenn es aktiviert und unbenutzt ist, was bei Linux-Distributionen wahrscheinlich der Fall ist, sollte es keine messbaren Auswirkungen auf die Leistung haben.

1.21 CPU/Task time and stats accounting →

1.21.1 Cputime accounting (Full dynticks CPU time accounting) →

1.21.1.1 Full dynticks CPU time accounting

CONFIG_VIRT_CPU_ACCOUNTING_GEN [=y] **[Y]**
Wählen Sie diese Option, um die Berechnung der Task- und CPU-Zeit auf Full-Dynticks-Systemen zu aktivieren. Diese Berechnung wird durch die Überwachung aller Kernel-Benutzer-Grenzen mithilfe des Kontextverfolgungs-Subsystems implementiert.
Die Berechnung erfolgt daher auf Kosten eines erheblichen Overheads.
Im Moment ist dies nur sinnvoll, wenn Sie an der Entwicklung des vollständigen Dynticks-Subsystems arbeiten.

1.21.2 Fine granularity task level IRQ time accounting

CONFIG_IRQ_TIME_ACCOUNTING [=y] **[Y]**
Wählen Sie diese Option aus, um eine fein granulare Berechnung der Task-Irq-Zeit zu aktivieren. Dies geschieht durch das Lesen eines Zeitstempels bei jedem Übergang zwischen dem softirq- und dem hardirq-Zustand, so dass es zu geringen Leistungseinbußen kommen kann.
Im Zweifelsfall sagen Sie hier N für Nein.

1.21.3 BSD Process Accounting

CONFIG_BSD_PROCESS_ACCT [=y] **[Y]**
Wenn Sie hier Y (für Ja) angeben, kann ein Programm auf Benutzerebene den Kernel (über einen speziellen Systemaufruf) anweisen, Prozessabrechnungsinformationen in eine Datei zu schreiben: Jedes Mal, wenn ein Prozess beendet wird, werden Informationen über diesen Prozess vom Kernel an die Datei angehängt. Die Informationen beinhalten Dinge wie die Erstellungszeit, den besitzenden Benutzer, den Befehlsnamen, den Speicherverbrauch, das kontrollierende Terminal usw. (die vollständige Liste kann in der acct-Struktur in <file:include/linux/acct.h> gefunden werden). Es obliegt dem Programm auf Benutzerebene, nützliche Dinge mit diesen Informationen zu tun. Dies ist im Allgemeinen eine gute Idee, also sagen Sie Y für Ja.

1.21.3.1 BSD Process Accounting version 3 file format

CONFIG_BSD_PROCESS_ACCT_V3 [=y] **[Y]**
Wenn Sie hier Y (für Ja) angeben, werden die Prozessabrechnungsinformationen in ein neues Dateiformat geschrieben, das auch die Prozess-IDs der einzelnen Prozesse und ihrer Eltern protokolliert. Beachten Sie, dass dieses Dateiformat nicht mit den früheren v0/v1/v2-Dateiformaten kompatibel ist, so dass Sie aktualisierte Werkzeuge für die Verarbeitung benötigen. Eine vorläufige Version dieser Werkzeuge ist unter `http://www.gnu.org/software/acct/` verfügbar.

1.21.4 Export task/process statistics through netlink

CONFIG_TASKSTATS [=y] **[Y]**
Export ausgewählter Statistiken für Aufgaben/Prozesse über die generische Netlink-Schnittstelle. Im Gegensatz zur BSD-Prozessabrechnung sind die Statistiken während der Lebensdauer von Aufgaben/Prozessen als Antwort auf Befehle verfügbar. Wie BSD-Accounting werden sie beim Beenden von Tasks in den Benutzerbereich gesendet.
Sagen Sie N, wenn Sie unsicher sind.

1.21.4.1 Enable per-task delay accounting

CONFIG_TASK_DELAY_ACCT [=y] **[Y]**
Sammeln Sie Informationen über die Zeit, die eine Task für das Warten auf Systemressourcen wie CPU, synchrone Block-E/A-Abwicklung und Auslagerung von Seiten aufwendet. Solche Statistiken können bei der Festlegung der Prioritäten eines Tasks im Verhältnis zu anderen Tasks für CPU-, IO-, RSS-Limits usw. helfen.
Sagen Sie N, wenn Sie unsicher sind.

1.21.4.2 Enable extended accounting over taskstats

CONFIG_TASK_XACCT [=y] **[Y]**
Sammeln von erweiterten Task-Accounting-Daten und Senden der Daten an das Userland zur Verarbeitung über die Taskstats-Schnittstelle.
Sagen Sie N, wenn Sie unsicher sind.

1.21.4.2.1 Enable per-task storage I/O accounting

CONFIG_TASK_IO_ACCOUNTING [=y] **[Y]**
Sammeln von Informationen über die Anzahl der Bytes an Speicher-E/A, die dieser Task verursacht hat.
Sagen Sie N, wenn Sie unsicher sind.

1.21.5 Pressure stall information tracking

CONFIG_PSI [=y] **[Y]**
Sammeln Sie Metriken, die anzeigen, wie überlastet die CPU-, Speicher- und IO-Kapazität im System sind.

Wenn Sie hier Y angeben, erstellt der Kernel /proc/pressure/ mit die Druckstatistikdateien cpu, memory und io. Diese zeigen den Anteil der Walltime an, in dem einige oder alle Tasks im System aufgrund der Beanspruchung der jeweiligen Ressource verzögert sind.

In Kerneln mit cgroup-Unterstützung verfügen cgroups (nur cgroup2) über cpu.pressure-, memory.pressure- und io.pressure-Dateien, die nur die Druckstaus für die gruppierten Aufgaben zusammenfassen.

Weitere Einzelheiten finden Sie unter Documentation/accounting/psi.rst.
Sagen Sie N, wenn Sie unsicher sind.

1.21.5.1 Require boot parameter to enable pressure stall information tracking

CONFIG_PSI_DEFAULT_DISABLED [=n] **[N]**
Wenn diese Option gesetzt ist, ist die Verfolgung von Druckstauinformationen standardmäßig deaktiviert, kann aber durch die Übergabe von psi=1 auf der Kernel-Befehlszeile beim Booten aktiviert werden.
Diese Funktion fügt dem Task-Wakeup- und Sleep-Pfad des Schedulers etwas Code hinzu. Der Overhead ist zu gering, um gängige planungsintensive Arbeitslasten in der Praxis zu beeinträchtigen (z. B. Webserver, Memcache), aber es zeigt sich in künstlichen Scheduler-Stresstests, wie z. B. Hackbench.
Wenn Sie paranoid sind und nicht sicher, wofür der Kernel verwendet wird, sagen Sie Y für Ja.
Sagen Sie N, wenn Sie unsicher sind.

1.22 CPU isolation

CONFIG_CPU_ISOLATION [=y] **[Y]**
Stellen Sie sicher, dass CPUs, auf denen kritische Aufgaben laufen, nicht durch irgendwelche „Störquellen“ wie ungebundene Workqueues, Timers, kthreads usw. gestört werden.
Ungebundene Aufgaben werden auf Housekeeping-CPUs verlagert. Dies wird durch den Boot-Parameter „isolcpus=“ gesteuert.
Sagen Sie Y für ja, wenn Sie unsicher sind.

1.23 RCU Subsystem →

Read – Copy – Update (Lesen, Kopieren, Aktualisieren)

1.23.1 Make expert-level adjustments to RCU configuration

CONFIG_RCU_EXPERT [=y] **[Y]**
Diese Option muss aktiviert werden, wenn Sie Anpassungen der RCU-Konfiguration auf Expertenebene vornehmen möchten. Standardmäßig können solche Anpassungen nicht vorgenommen werden, was den oft vorteilhaften Nebeneffekt hat, dass „make oldconfig“ Sie davon abhält, alle möglichen detaillierten Fragen darüber zu stellen, wie Sie zahlreiche obskure RCU-Optionen eingerichtet haben möchten.
Sagen Sie Y, wenn Sie Anpassungen an RCU auf Expertenebene vornehmen müssen.
Sagen Sie N, wenn Sie unsicher sind.

1.23.2 Force selection of TASKS_RCU

CONFIG_FORCE_TASKS_RCU [=n] **[N]**
Diese Option erzwingt eine aufgabenbasierte RCU-Implementierung die nur freiwillige Kontextwechsel verwendet (keine Preemption!), Leerlauf und Benutzermodus-Ausführung als Ruhezustände verwendet. Nicht für manuelle Auswahl in den meisten Fällen.

1.23.3 Force selection of Tasks Rude RCU

CONFIG_FORCE_TASKS_RUDE_RCU [=n] **[N]**
Diese Option erzwingt eine Task-basierte RCU-Implementierung, die nur Kontextwechsel (einschließlich Preemption) und die Ausführung im Benutzermodus als Ruhezustand verwendet. Sie erzwingt IPIs und Kontextwechsel auf allen Online-CPUs, auch auf den Idle-CPUs, also mit Vorsicht verwenden. In den meisten Fällen nicht für die manuelle Auswahl geeignet.

1.23.4 Force selection of Tasks Trace RCU

CONFIG_FORCE_TASKS_TRACE_RCU [=n] **[N]**
Diese Option ermöglicht eine Task-basierte RCU-Implementierung, die explizite rcu_read_lock_trace()-Lesemarker verwendet und es ermöglicht, dass diese Leser sowohl in der Leerlaufschleife als auch in den CPU-Hotplug-Codepfaden erscheinen. Es kann IPIs auf Online-CPUs erzwingen, auch auf Idle-CPUs, also mit Vorsicht verwenden. In den meisten Fällen nicht für die manuelle Auswahl geeignet.

1.23.5 Tree-based hierarchical RCU fanout value

CONFIG_RCU_FANOUT [=64] **[64]**
Diese Option steuert den Fanout von hierarchischen Implementierungen von RCU, so dass RCU auf Maschinen mit einer großen Anzahl von CPUs effizient arbeiten kann. Dieser Wert muss mindestens die vierte Wurzel von NR_CPUS sein, wodurch NR_CPUS wahnsinnig groß werden kann. Der Standardwert von RCU_FANOUT sollte für Produktionssysteme verwendet werden, aber wenn Sie die RCU-Implementierung selbst einem Stresstest unterziehen, ermöglichen kleine RCU_FANOUT-Werte das Testen von Codepfaden für große Systeme auf kleinen (kleineren) Systemen.
Wählen Sie eine bestimmte Zahl, wenn Sie RCU selbst testen. Nehmen Sie den Standardwert, wenn Sie unsicher sind.
Symbol: RCU_FANOUT [=64]
Type : integer (Ganzzahl)
Bereich (range) : [2 64]

1.23.6 Tree-based hierarchical RCU leaf-level fanout value

CONFIG_RCU_FANOUT_LEAF [=16] **[16]**
Diese Option steuert das Fanout auf Blattebene bei hierarchischen Implementierungen von RCU und ermöglicht es, Cache-Misses gegen Sperrkonflikte abzuwägen. Systeme, die ihre Scheduling-Clock-Interrupts aus Gründen der Energieeffizienz synchronisieren, werden die Standardeinstellung bevorzugen, da der kleinere Leaf-Level-Fanout die Lock-Contention-Level akzeptabel niedrig hält. Sehr große Systeme (Hunderte oder Tausende von CPUs) werden stattdessen diesen Wert auf den maximal möglichen Wert setzen wollen, um die Anzahl der Cache-Misses zu reduzieren, die während der Initialisierung der RCU-Grace-Periode auftreten. Diese Systeme neigen dazu, CPU-gebunden zu laufen, und werden daher nicht von synchronisierten Interrupts unterstützt, und neigen daher dazu, sie zu verzerren, was den Sperrkonflikt so weit reduziert, dass große Fanouts auf Blattebene gut funktionieren. Das heißt, wenn Sie den Fanout auf Blattebene auf eine große Zahl setzen, wird dies wahrscheinlich zu problematischen Sperrkonflikten auf den rcu_node-Strukturen auf Blattebene führen, es sei denn, Sie booten mit dem Kernelparameter skew_tick.
Wählen Sie eine bestimmte Zahl, wenn Sie die RCU selbst testen.
Wählen Sie den maximal zulässigen Wert für große Systeme, aber bedenken Sie, dass Sie möglicherweise auch den Kernel-Boot-Parameter skew_tick setzen müssen, um Konflikte bei den Sperren der rcu_node-Strukturen zu vermeiden. Nehmen Sie den Standardwert, wenn Sie unsicher sind.
Symbol: RCU_FANOUT_LEAF [=64]
Type : integer (Ganzzahl)
Bereich (range) : [2 64]

1.23.7 Enable RCU priority boosting

CONFIG_RCU_BOOST [=y] **[Y]**
Diese Option erhöht die Priorität von preemptierten RCU-Lesern, die die aktuelle preemptible RCU-Schonfrist zu lange blockieren. Diese Option verhindert auch, dass schwere Lasten den Aufruf von RCU-Callbacks blockieren.
Geben Sie hier Y an, wenn Sie mit Echtzeitanwendungen oder großen Lasten arbeiten.
Sagen Sie hier N ein, wenn Sie unsicher sind.

1.23.7.1 Milliseconds to delay boosting after RCU grace-period start

CONFIG_RCU_BOOST_DELAY [=500] **[500]**
Diese Option gibt die Zeit an, die nach dem Beginn einer bestimmten Karenzzeit gewartet werden soll, bevor die Priorität von RCU-Lesern, die diese Karenzzeit blockieren, erhöht wird.
Beachten Sie, dass jeder RCU-Leser, der eine beschleunigte RCU-Schonfrist blockiert, sofort hochgestuft wird.
Akzeptieren Sie die Standardeinstellung, wenn Sie unsicher sind.
Symbol: RCU_BOOST_DELAY [=500]
Typ : Integer (Ganzzahl)
Bereich : [0 3000]

1.23.7.2 Perform RCU expedited work in a real-time kthread

CONFIG_RCU_EXP_KTHREAD [=n] **[N]**
Verwenden Sie diese Option, um die Latenzzeiten der beschleunigten Neuheitsschonfristen weiter zu reduzieren, was allerdings mit mehr Störungen verbunden ist. Diese Option ist standardmäßig auf PREEMPT_RT=y-Kerneln deaktiviert, die beschleunigte Neuheitsschonfristen nach dem Booten durch die bedingungslose Einstellung rcupdate.rcu_normal_after_boot=1 deaktivieren.
Akzeptieren Sie die Voreinstellung, wenn Sie unsicher sind.

1.23.8 Offload RCU callback processing from boot-selected CPUs

CONFIG_RCU_NOCB_CPU [=y] **[Y]**
Verwenden Sie diese Option, um den Jitter des Betriebssystems für aggressive HPC- oder Echtzeit-Workloads zu reduzieren. Sie kann auch verwendet werden, um RCU-Callback-Aufrufe auf energieeffiziente CPUs in batteriebetriebenen asymmetrischen Multiprozessoren auszulagern. Der Preis für diesen reduzierten Jitter ist, dass der Overhead von call_rcu() ansteigt und dass bei einigen Workloads ein erheblicher Anstieg der Kontextwechselraten zu verzeichnen ist.
Diese Option entlastet den Aufruf von Callbacks von der Gruppe von CPUs, die zur Boot-Zeit durch den rcu_nocbs-Parameter angegeben wird. Für jede dieser CPUs wird ein kthread („rcuox/N“) erstellt, um Callbacks aufzurufen, wobei „N“ die CPU ist, die entlastet wird, und wobei „x“ „p“ für RCU-preempt (PREEMPTION-Kernel) und „s“ für RCU-sched (!PREEMPTION-Kernel) ist. Nichts hindert diesen kthread daran, auf den angegebenen CPUs zu laufen, aber (1) die kthreads können zwischen jedem Callback preempted werden, und (2) Affinität oder cgroups können verwendet werden, um die kthreads zu zwingen, auf jeder gewünschten Gruppe von CPUs zu laufen.
Sagen Sie hier Y, wenn Sie trotz des zusätzlichen Overheads ein geringeres OS-Jitter benötigen.
Sagen Sie hier N, wenn Sie unsicher sind.

1.23.8.1 Offload RCU callback processing from all CPUs by default

CONFIG_RCU_NOCB_CPU_DEFAULT_ALL [=n] **[N]**
Verwenden Sie diese Option, um die Callback-Verarbeitung standardmäßig von allen CPUs zu entlasten, wenn der Boot-Parameter rcu_nocbs oder nohz_full nicht vorhanden ist. Dadurch wird auch die Notwendigkeit vermieden, Boot-Parameter zu verwenden, um den Effekt der Entlastung aller CPUs beim Booten zu erreichen.
Geben Sie hier Y an, wenn Sie alle CPUs standardmäßig beim Booten entlasten wollen.
Sagen Sie hier N, wenn Sie sich nicht sicher sind.

1.23.8.2 Offload RCU callback from real-time kthread

CONFIG_RCU_NOCB_CPU_CB_BOOST [=n] **[N]**
Verwenden Sie diese Option, um ausgelagerte Rückrufe als SCHED_FIFO aufzurufen, um ein Aushungern durch schwere SCHED_OTHER-Hintergrundlast zu vermeiden. Natürlich führt die Ausführung als SCHED_FIFO während Callback Floods dazu, dass die rcuo[ps] kthreads die CPU für Hunderte von Millisekunden oder mehr monopolisieren. Wenn Sie diese Option aktivieren, müssen Sie daher sicherstellen, dass latenzempfindliche Aufgaben entweder mit höherer Priorität oder auf einer anderen CPU ausgeführt werden.
Geben Sie hier Y an, wenn Sie die RT-Priorität für die Auslagerung von kthreads festlegen möchten.
Sagen Sie hier N, wenn Sie einen !PREEMPT_RT-Kernel bauen und sich unsicher sind.

1.23.9 Tasks Trace RCU readers use memory barriers in user and idle

CONFIG_TASKS_TRACE_RCU_READ_MB [=n] **[N]**
Verwenden Sie diese Option, um die Anzahl der IPIs (inter-processor interrupts), die an CPUs gesendet werden, die im Benutzerraum ausgeführt werden oder sich im Leerlauf befinden, während Tasks RCU-Tilgungsfristen verfolgen, weiter zu reduzieren. Da eine vernünftige Einstellung des Kernel-Boot-Parameters rcupdate.rcu_task_ipi_delay solche IPIs für viele Arbeitslasten eliminiert, ist die richtige Einstellung dieser Kconfig-Option vor allem für aggressive Echtzeitinstallationen und für batteriebetriebene Geräte wichtig, daher die oben gewählte Standardeinstellung.
Sagen Sie hier Y, wenn Sie IPIs hassen.
Sagen Sie hier N, wenn Sie leseseitige Speicherbarrieren hassen.
Nehmen Sie die Standardeinstellung, wenn Sie unsicher sind.

1.23.10 RCU callback lazy invocation functionality

CONFIG_RCU_LAZY [=y] [**Y**]
Um Strom zu sparen, sollten Sie RCU-Rückrufe stapeln und nach einer Verzögerung, einem Speicherdruck oder einer zu großen Rückrufliste flushen.

1.23.11 RCU callback-batch backup time check

CONFIG_RCU_DOUBLE_CHECK_CB_TIME [=y] [**Y**]
Verwenden Sie diese Option, um eine präzisere Durchsetzung des Modulparameters rcutree.rcu_resched_ns in Situationen zu ermöglichen, in denen ein einziger RCU-Callback Hunderte von Mikrosekunden lang laufen könnte, wodurch die 32-Callback-Batching-Funktion, die verwendet wird, um die Kosten der feinkörnigen, aber teuren local_clock()-Funktion zu amortisieren, unterlaufen wird.
Diese Option rundet rcutree.rcu_resched_ns auf den nächsten Jiffy auf und setzt die 32-Callback-Batching-Funktion außer Kraft, wenn diese Grenze überschritten wird.
Sagen Sie hier Y, wenn Sie eine strengere Durchsetzung des Rückruflimits benötigen.
Sagen Sie hier N, wenn Sie unsicher sind.

1.24 Kernel .config support

CONFIG_IKCONFIG [=y] [**Y**]
Mit dieser Option kann der gesamte Inhalt der „.config“-Datei des Linux-Kernels im Kernel gespeichert werden. Sie dokumentiert, welche Kernel-Optionen in einem laufenden Kernel oder in einem On-Disk-Kernel verwendet werden. Diese Informationen können mit dem Skript scripts/extract-ikconfig aus der Kernel-Image-Datei extrahiert und als Eingabe verwendet werden, um den aktuellen Kernel neu zu erstellen oder einen anderen Kernel zu bauen. Sie können auch aus einem laufenden Kernel extrahiert werden, indem /proc/config.gz gelesen wird, falls dies aktiviert ist (siehe unten).
Definiert mit init/Kconfig:686

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

CONFIG_IKCONFIG_PROC [=y] [**Y**]
Diese Option ermöglicht den Zugriff auf die Kernelkonfigurationsdatei über /proc/config.gz.

1.25 Enable kernel headers through /sys/kernel/kheaders.tar.xz

CONFIG_IKHEADERS [=m] [**M**]
Diese Option ermöglicht den Zugriff auf die In-Kernel-Header, die während des Build-Prozesses erzeugt werden. Diese können verwendet werden, um eBPF-Tracing-Programme oder ähnliche Programme zu erstellen. Wenn Sie die Header als Modul erstellen, wird ein Modul namens kheaders.ko erstellt, das bei Bedarf geladen werden kann, um Zugriff auf die Header zu erhalten.

1.26 Kernel log buffer size (16 ⇒ 64KB, 17 ⇒ 128KB)

CONFIG_LOG_BUF_SHIFT [=17] [**17**]
Wählen Sie die minimale Größe des Kernel-Protokollpuffers als eine Potenz von 2 aus. Die endgültige Größe wird durch den Konfigurationsparameter LOG_CPU_MAX_BUF_SHIFT beeinflusst, siehe unten. Eine höhere Größe kann auch durch den Boot-Parameter „log_buf_len“ erzwungen werden.
Beispiele:
- 17 ⇒ 128 KB
- 16 ⇒ 64 KB
- 15 ⇒ 32 KB
- 14 ⇒ 16 KB
- 13 ⇒ 8 KB
- 12 ⇒ 4 KB

Symbol: LOG_BUF_SHIFT
Type: Integer (Ganzzahl)
Bereich (range): [12 25]

1.27 CPU kernel log buffer size contribution (13 ⇒ 8 KB, 17 ⇒ 128KB)

CONFIG_LOG_BUF_SHIFT [=12] **[12]**
Diese Option ermöglicht es, die Standardgröße des Ringpuffers entsprechend der Anzahl der CPUs zu erhöhen. Der Wert definiert den Beitrag jeder CPU als eine Potenz von 2. Der beanspruchte Speicherplatz beträgt in der Regel nur wenige Zeilen, kann aber viel mehr sein, wenn Probleme gemeldet werden, z. B. bei Rückverfolgungen. Die erhöhte Größe bedeutet, dass ein neuer Puffer zugewiesen werden muss und der ursprüngliche statische Puffer ungenutzt ist. Dies ist nur auf Systemen mit mehr CPUs sinnvoll. Daher wird dieser Wert nur verwendet, wenn die Summe der Beiträge größer ist als die Hälfte des Standard-Kernel-Ringpuffers, wie durch `LOG_BUF_SHIFT` definiert. Die Standardwerte sind so eingestellt, dass mehr als 16 CPUs erforderlich sind, um die Zuweisung auszulösen. Diese Option wird auch ignoriert, wenn der Kernelparameter „log_buf_len“ verwendet wird, da er eine exakte (Zweierpotenz) Größe des Ringpuffers erzwingt. Die Anzahl der möglichen CPUs wird für diese Berechnung verwendet, wobei Hotplugging ignoriert wird, so dass die Berechnung für das Worst-Case-Szenario optimal ist und gleichzeitig ein einfacher Algorithmus ab dem Hochfahren verwendet werden kann. Beispiele für Verschiebungswerte und ihre Bedeutung:

17 ⇒ 128 KB für jede CPU
16 ⇒ 64 KB für jede CPU
15 ⇒ 32 KB für jede CPU
14 ⇒ 16 KB für jede CPU
13 ⇒ 8 KB für jede CPU
12 ⇒ 4 KB für jede CPU

Symbol: LOG_CPU_MAX_BUF_SHIFT
Type: Integer (Ganzzahl)
Bereich (range): [0 21]

1.28 Printk indexing debugfs interface)

CONFIG_PRINTK_INDEX [=y] **[Y]**
Unterstützung für die Indizierung aller zur Kompilierzeit bekannten printk-Formate unter <debugfs>/printk/index/<module> hinzufügen. Dies kann als Teil der Wartung von Daemonen, die /dev/kmsg überwachen, verwendet werden, da es die Überprüfung der in einem Kernel vorhandenen printk-Formate erlaubt, was die Erkennung von Fällen ermöglicht, in denen überwachte printks geändert oder nicht mehr vorhanden sind.
Es gibt keine zusätzlichen Laufzeitkosten für printk, wenn dies aktiviert ist.

1.29 Scheduler features →

Scheduler-Funktionen

1.29.1 Enable utilization clamping for RT/FAIR tasks

CONFIG_UCLAMP_TASK [=y] **[Y]**
Diese Funktion ermöglicht es dem Scheduler, die geklemmte Auslastung jeder CPU auf der Grundlage der auf dieser CPU geplanten RUNNABLE-Tasks zu verfolgen. Mit dieser Option kann der Benutzer die minimale und maximale CPU-Auslastung angeben, die für RUNNABLE-Aufgaben zulässig ist. Die maximale Auslastung definiert die maximale Häufigkeit, mit der ein Task laufen soll, während die minimale Auslastung die minimale Häufigkeit definiert, mit der er laufen soll.
Sowohl die Minimal- als auch die Maximalwerte für die Auslastung sind Hinweise für den Scheduler, um seine Frequenzauswahl zu verbessern, aber sie erzwingen oder gewähren keine bestimmte Bandbreite für Tasks.
Im Zweifelsfall sagen Sie N für Nein.

1.29.1.1 Number of supported utilization clamp buckets

CONFIG_UCLAMP_BUCKETS_COUNT [=5] **[5]**
Legt die Anzahl der zu verwendenden Klammerbereiche fest. Der Bereich der einzelnen Buckets ist SCHED_CAPACITY_SCALE/UCLAMP_BUCKETS_COUNT. Je höher die Anzahl der Clamp-Buckets, desto feiner die Granularität und desto höher die Präzision der Clamp-Aggregation und -Verfolgung während der Laufzeit. Mit dem minimalen Konfigurationswert haben wir beispielsweise 5 Clamp-Buckets,

die jeweils 20 % Auslastung verfolgen. Eine um 25 % gesteigerte Aufgabe wird im Bucket [20..39]% gezählt und setzt den effektiven Wert der Bucketklemme auf 25 %. Wenn eine zweite, um 30 % erhöhte Aufgabe auf derselben CPU eingeplant wird, wird diese Aufgabe im selben Bucket wie die erste Aufgabe gezählt und erhöht den effektiven Bucket-Clamp-Wert auf 30 %. Der effektive Klemmwert eines Bereichs wird auf seinen Nennwert (20 % im obigen Beispiel) zurückgesetzt, wenn keine weiteren Aufgaben mehr in diesem Bereich gezählt werden. Bei einigen Aufgaben kann eine zusätzliche Verstärkungs-/Kappungsmarge hinzugefügt werden. Im obigen Beispiel wird die 25 %-Aufgabe auf 30 % angehoben, bis sie die CPU verlässt. Sollte dies auf bestimmten Systemen nicht akzeptabel sein, ist es immer möglich, den Spielraum zu verringern, indem die Anzahl der Clamp-Buckets erhöht wird, um den verbrauchten Speicher gegen die Genauigkeit der Laufzeitverfolgung einzutauschen.
Im Zweifelsfall sollten Sie den Standardwert verwenden.

1.30 Memory placement aware NUMA scheduler

CONFIG_NUMA_BALANCING [=y] **[Y]**
Diese Option bietet Unterstützung für die automatische NUMA-kompatible Speicher-/Task-Platzierung. Der Mechanismus ist recht primitiv und basiert darauf, dass Speicher migriert wird, wenn er Referenzen auf den Knoten hat, auf dem die Aufgabe läuft.
Dieses System ist auf UMA-Systemen inaktiv.

1.30.1 Automatically enable NUMA aware memory/task placemnent

CONFIG_NUMA_BALANCING_DEFAULT_ENABLED [=y] **[Y]**
Wenn diese Option gesetzt ist, wird der automatische NUMA-Ausgleich aktiviert, wenn das System auf einem NUMA-Rechner läuft.

1.31 Control Group support →

CONFIG_CGROUPS [=y] **[Y]**
(Unterstützung der Kontrollgruppe)
Diese Option bietet Unterstützung für die Gruppierung von Prozessgruppen zur Verwendung mit Prozesskontrollsubsystemen wie Cpusets, CFS, Speicherkontrolle oder Geräteisolierung.
Siehe

- Dokumentation/scheduler/sched-design-CFS.rst (CFS)
- Documentation/admin-guide/cgroup-v1/ (Funktionen für Gruppierung, Isolierung und Ressourcenkontrolle)

Sagen Sie N, wenn Sie unsicher sind.

1.31.1 Favor dynamic modification latency reduction by default

CONFIG_CGROUP_FAVOR_DYNMODS [=n] **[N]**
Diese Option aktiviert standardmäßig die Einhängeoption „favordynmods“, die die Latenzzeiten dynamischer C-Gruppen-Änderungen wie Task-Migrationen und Controller-Ein-/Ausschaltungen auf Kosten von Hot-Path-Operationen wie Forks und Exits verteuert.
Sagen Sie N, wenn Sie unsicher sind.

1.31.2 Memory controller

CONFIG_MEMCG [=y] **[Y]**
Ermöglicht die Kontrolle über den Speicherbedarf von Tasks in einer cgroup.

1.31.3 IO controller

CONFIG_BLK_CGROUP [=y] **[Y]**
Generische Block IO Controller cgroup Schnittstelle. Dies ist die gemeinsame cgroup-Schnittstelle, die von verschiedenen IO-Kontrollstrategien verwendet werden sollte.

Derzeit wird sie vom CFQ IO Scheduler zur Erkennung von Task-Gruppen und zur Steuerung der Zuweisung von Festplattenbandbreite (proportionale Zeitscheibenzuweisung) an solche Task-Gruppen verwendet. Sie wird auch von der Bio-Throttling-Logik in der Blockschicht verwendet, um eine Obergrenze für die IO-Raten auf einem Gerät einzuführen.
Diese Option aktiviert nur die generische Infrastruktur des Block-IO-Controllers. Man muss auch die tatsächliche IO-Kontrolllogik/-Politik aktivieren. Um die proportionale Aufteilung der Festplattenbandbreite in CFQ zu aktivieren, setzen Sie CONFIG_BFQ_GROUP_IOSCHED=y; für die Aktivierung der Drosselungspolitik setzen Sie CONFIG_BLK_DEV_THROTTLING=y.
Weitere Informationen finden Sie unter Documentation/admin-guide/cgroup-v1/blkio-controller.rst.

1.31.4 CPU controller →

CONFIG_CGROUP_SCHED [=y] **[Y]**
Diese Funktion ermöglicht es dem CPU-Scheduler, Task-Gruppen zu erkennen und die Zuweisung von CPU-Bandbreite an solche Task-Gruppen zu steuern. Er verwendet cgroups, um Tasks zu gruppieren.

1.31.4.1 Group scheduling for SCHED_OTHER

CONFIG_FAIR_GROUP_SCHED [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

1.31.4.1.1 CPU bandwidth provisioning for FAIR_GROUP_SCHED

CONFIG_CFS_BANDWIDTH [=y] **[Y]**
Mit dieser Option können Benutzer CPU-Bandbreitenraten (Limits) für Aufgaben festlegen, die innerhalb des Fair Group Schedulers laufen. Gruppen, für die kein Limit festgelegt wurde, gelten als uneingeschränkt und werden ohne Einschränkung ausgeführt.
Weitere Informationen finden Sie unter Documentation/scheduler/sched-bwc.rst.

1.31.4.2 Group scheduling for SCHED_RR/FIFO

CONFIG_RT_GROUP_SCHED [=n] **[N]**
Mit dieser Funktion können Sie den Task-Gruppen explizit echte CPU-Bandbreite zuweisen. Wenn sie aktiviert ist, wird es auch unmöglich, Echtzeitaufgaben für Nicht-Root-Benutzer zu planen, bis Sie ihnen Echtzeitbandbreite zuweisen.
Weitere Informationen finden Sie unter Documentation/scheduler/sched-rt-group.rst.

1.31.5 Utilization clamping per group of tasks

CONFIG_UCLAMP_TASK_GROUP [=y] **[Y]**
Mit dieser Funktion kann der Scheduler die geklemmte Auslastung jeder CPU auf der Grundlage der RUNNABLE-Tasks, die derzeit auf dieser CPU geplant sind, verfolgen. Wenn diese Option aktiviert ist, kann der Benutzer eine minimale und maximale CPU-Bandbreite angeben, die für jede einzelne Aufgabe in einer Gruppe zulässig ist. Mit der maximalen Bandbreite kann die maximale Frequenz, die ein Task verwenden kann, festgelegt werden, während mit der minimalen Bandbreite eine minimale Frequenz festgelegt werden kann, die ein Task immer verwenden wird. Bei aktivierter aufgabengruppenbasierter Auslastungsbegrenzung wird ein eventuell angegebener aufgabenspezifischer Begrenzungswert durch den von cgroup angegebenen Begrenzungswert eingeschränkt. Sowohl die minimale als auch die maximale Task-Klemmung kann nicht größer sein als die entsprechende auf Task-Gruppen-Ebene definierte Klemmung.
Im Zweifelsfall sagen Sie N.

1.31.6 PIDs controller

CONFIG_CGROUP_PIDS [=y] **[Y]**
Erzwingt die Begrenzung der Prozessanzahl im Bereich einer cgroup. Jeder Versuch, mehr Prozesse zu forken, als in der cgroup erlaubt sind, schlägt fehl. PIDs sind grundsätzlich eine globale Ressource, da es ziemlich trivial ist, eine PID-Erschöpfung zu erreichen, bevor man auch nur eine konservative kmemcg-Grenze erreicht. Infolgedessen ist es möglich, ein System zum Stillstand zu bringen, ohne durch andere cgroup-Richtlinien eingeschränkt zu werden. Der PID-Regler ist dafür ausgelegt, dies zu verhindern.

Es sollte beachtet werden, dass organisatorische Operationen (wie z. B. das Anhängen an eine cgroup-Hierarchie) *nicht* durch den PIDs-Controller blockiert werden, da das PIDs-Limit nur die Fähigkeit eines Prozesses zum Forking, nicht aber zum Anhängen an eine cgroup beeinflusst.

1.31.7 RDMA controller

CONFIG_CGROUP_RDMA [=y] **[Y]**
Ermöglicht die Durchsetzung der vom IB-Stack definierten RDMA-Ressourcen. Es ist relativ einfach für Verbraucher, RDMA-Ressourcen zu erschöpfen, was dazu führen kann, dass Ressourcen für andere Verbraucher nicht mehr verfügbar sind. Der RDMA-Controller ist dafür ausgelegt, dies zu verhindern. Das Anhängen von Prozessen mit aktiven RDMA-Ressourcen an die cgroup-Hierarchie ist erlaubt, auch wenn die Grenze der Hierarchie überschritten werden kann.

1.31.8 Freezer controller

CONFIG_CGROUP_FREEZER [=y] **[Y]**
Ermöglicht das Einfrieren und Aufheben des Einfrierens aller Aufgaben in einer C-Group. Diese Option betrifft die ORIGINAL cgroup-Schnittstelle. Der cgroup2-Speicher-Controller enthält standardmäßig wichtige In-Kernel-Speicherverbraucher.
Wenn Sie cgroup2 verwenden, sagen Sie N.

1.31.9 HugeTLB controller

CONFIG_CGROUP_HUGETLB [=y] **[Y]**
Bietet eine cgroup-Steuerung für HugeTLB-Seiten. Wenn Sie dies aktivieren, können Sie die HugeTLB-Nutzung pro cgroup begrenzen. Die Begrenzung wird während eines Seitenfehlers durchgesetzt. Da HugeTLB keine Seitenrückforderung unterstützt, bedeutet die Durchsetzung des Limits zum Zeitpunkt des Seitenfehlers, dass die Anwendung ein SIGBUS-Signal erhält, wenn sie versucht, über das Limit hinaus auf HugeTLB-Seiten zuzugreifen. Dies setzt voraus, dass die Anwendung im Voraus weiß, wie viele HugeTLB-Seiten sie für ihre Nutzung benötigt. Die Kontrollgruppe wird im dritten Page-lru-Zeiger verfolgt. Dies bedeutet, dass wir die Steuergruppe nicht mit einer riesigen Seite von weniger als 3 Seiten verwenden können.

1.31.10 Cpuset controller

CONFIG_CPUSETS [=y] **[Y]**
Mit dieser Option können Sie CPUSETs erstellen und verwalten, die es ermöglichen, ein System dynamisch in Gruppen von CPUs und Speicherknoten zu partitionieren und Aufgaben zuzuweisen, die nur innerhalb dieser Gruppen ausgeführt werden. Dies ist vor allem auf großen SMP- oder NUMA-Systemen nützlich.
Sagen Sie N, wenn Sie unsicher sind.

1.31.10.1 Include legacy /proc/<pid>/cpuset file

CONFIG_PROC_PID_CPUSET [=y] **[Y]**
This option will let you create and manage CPUSETs which allow dynamically partitioning a system into sets of CPUs and Memory Nodes and assigning tasks to run only within those sets. This is primarily useful on large SMP or NUMA systems.
Say N if unsure.

1.31.11 Device controller

CONFIG_CGROUP_DEVICE [=y] **[Y]**
Bietet einen cgroup-Controller an, der Whitelists für Geräte implementiert, die ein Prozess in der cgroup mknod oder öffnen kann.

1.31.12 Simple CPU accounting controller

CONFIG_CGROUP_CPUACCT [=y] **[Y]**
(Einfacher CPU-Accounting-Controller)

Bietet einen einfachen Controller für die Überwachung des gesamten CPU-Verbrauchs der Tasks in einer cgroup an.

1.31.13 Perf controller

CONFIG_CGROUP_PERF [=y] **[Y]**
Diese Option erweitert den Modus perf per-cpu, um die Überwachung auf Threads zu beschränken, die zu der angegebenen cgroup gehören und auf der angegebenen CPU laufen. Sie kann auch verwendet werden, um die cgroup ID in Stichproben zu haben, so dass sie Leistungsereignisse zwischen cgroups überwachen kann.
Sagen Sie N, wenn Sie unsicher sind.

1.31.14 Support for eBPF programs attached to cgroups

CONFIG_CGROUP_BPF [=y] **[Y]**
Erlaubt das Anhängen von eBPF-Programmen an eine cgroup mit dem bpf(2)-Syscall-Befehl `BPF_PROG_ATTACH`.
In welchem Kontext auf diese Programme zugegriffen wird, hängt von der Art des Attachments ab. Zum Beispiel werden Programme, die mit BPF_CGROUP_INET_INGRESS angehängt werden, auf dem Ingress-Pfad von inet-Sockets ausgeführt.

1.31.15 Misc resource controller

CONFIG_CGROUP_MISC [=y] **[Y]**
Bietet einen Controller für verschiedene Ressourcen auf einem Host. Verschiedene skalare Ressourcen sind die Ressourcen auf dem Host-System, die nicht wie die anderen cgroups abstrahiert werden können. Dieser Controller verfolgt und begrenzt die verschiedenen Ressourcen, die von einem Prozess verwendet werden, der an eine cgroup-Hierarchie angeschlossen ist.
Weitere Informationen finden Sie im Abschnitt misc cgroup in /Documentation/admin-guide/cgroup-v2.rst.

1.31.16 Debug controller

CONFIG_CGROUP_DEBUG [=n] **[N]**
Diese Option aktiviert einen einfachen Controller, der Debugging-Informationen über das cgroups-Framework exportiert. Dieser Controller ist nur für das Debugging von Kontroll-C-Gruppen gedacht. Seine Schnittstellen sind nicht stabil.
Sagen Sie N.

1.32 Namespaces support →

CONFIG_NAMESPACES [=y] **[Y]**
(Unterstützung von Namensräumen, namespaces)
Bietet die Möglichkeit, Aufgaben mit verschiedenen Objekten unter Verwendung derselben Kennung arbeiten zu lassen. Zum Beispiel kann sich dieselbe IPC-ID auf verschiedene Objekte beziehen oder dieselbe Benutzer-ID oder pid kann sich auf verschiedene Aufgaben beziehen, wenn sie in verschiedenen Namensräumen verwendet werden.

1.32.1 UTS namespace

CONFIG_UTS_NS [=y] **[Y]**
In diesem Namensraum sehen Aufgaben verschiedene Informationen, die mit dem Systemaufruf uname() bereitgestellt werden

1.32.2 TIME namespace

CONFIG_TIME_NS [=y] **[Y]**
In diesem Namespace können boottime und monotone Uhren eingestellt werden. Die Zeit läuft dann mit der gleichen Geschwindigkeit weiter.

1.32.3 IPC namespace

CONFIG_IPC_NS [=y] **[Y]**
In diesem Namensraum arbeiten Aufgaben mit IPC-IDs (Interprozess-IDs), die jeweils verschiedenen IPC-Objekten in verschiedenen Namensräumen entsprechen.

1.32.4 User namespace

CONFIG_USER_NS [=y] **[Y]**
Dies ermöglicht es Containern, d.h. V-Servern, Benutzernamensräume zu verwenden, um verschiedene Benutzerinformationen für verschiedene Server bereitzustellen. Wenn Benutzernamensräume im Kernel aktiviert sind, wird empfohlen, dass die Option `MEMCG` ebenfalls aktiviert wird und dass der Benutzerbereich die Speicherkontrollgruppen verwendet, um die Speichermenge zu begrenzen, die nicht privilegierte Benutzer verwenden können.

1.32.4.1 Allow unprivileged users to create namespaces

CONFIG_USERS_NS_UNPRIVILEGED [=y] **[Y]**
Wenn diese Funktion deaktiviert ist, können unprivilegierte Benutzer keine neuen Namensräume erstellen. Die Möglichkeit, dass Benutzer ihre eigenen Namespaces erstellen können, war Teil mehrerer kürzlich erfolgter lokaler Privilegienerweiterungen. Wenn Sie also Benutzernamespaces benötigen, aber paranoid bzw. sicherheitsbewusst sind, sollten Sie diese Funktion deaktivieren. Diese Einstellung kann zur Laufzeit mit dem `kernel.unprivileged_userns_clone sysctl` außer Kraft gesetzt werden.
Wenn Sie unsicher sind, sagen Sie Y.

1.32.5 PID namespace

CONFIG_PID_NS [=y] **[Y]**
Unterstützung von Prozess-ID-Namensräumen. Dies ermöglicht es, mehrere Prozesse mit der gleichen pid zu haben, solange sie sich in verschiedenen pid-Namensräumen befinden. Dies ist ein Baustein von Containern.

1.32.6 Network namespace

CONFIG_NET_NS [=y] **[Y]**
Ermöglicht es dem Benutzer, scheinbar mehrere Instanzen des Netzwerkstapels zu erstellen.

1.33 Checkpoint/restore support

CONFIG_CHECKPOINT_RESTORE [=y] **[Y]**
Ermöglicht zusätzliche Kernel-Funktionen in einer Art Checkpoint/Restore. Insbesondere fügt es zusätzliche prctl-Codes zum Einrichten von Prozesstext, Daten- und Heap-Segmentgrößen sowie einige zusätzliche /proc-Dateisystemeinträge hinzu.
Wenn Sie unsicher sind, geben Sie hier N an.

1.34 Automatic process group scheduling

CONFIG_SCHED_AUTOGROUP [=y] **[Y]**
Mit dieser Option wird der Scheduler für gängige Desktop-Workloads optimiert, indem automatisch Aufgabengruppen erstellt und aufgefüllt werden. Diese Trennung von Arbeitslasten isoliert aggressive CPU-Brenner (wie Build-Jobs) von Desktop-Anwendungen. Die automatische Erstellung von Aufgabengruppen basiert derzeit auf der Aufgabensitzung.

1.35 Kernel→user space relay support (formerly relayfs)

CONFIG_RELAY [=y] **[Y]**
Diese Option aktiviert die Unterstützung für die Relaisschnittstelle in bestimmten Dateisystemen (wie debugfs). Sie wurde entwickelt, um einen effizienten Mechanismus für Werkzeuge und Einrichtungen zur Weiterleitung großer Datenmengen aus dem Kernelbereich in den Benutzerbereich bereitzustellen.
Wenn Sie unsicher sind, sagen Sie N.

1.36 Initial RAM filesystem and RAM disk (initramfs/initrd) support

CONFIG_BLK_DEV_INITRD [=y] **[Y]**
Das anfängliche RAM-Dateisystem ist ein ramfs, das vom Bootloader (loadlin oder lilo) geladen und vor dem normalen Bootvorgang als root eingehängt wird. Es wird typischerweise verwendet, um Module zu laden, die zum Einhängen des „echten“ Root-Dateisystems benötigt werden, usw.
Siehe <file:Documentation/admin-guide/initrd.rst> für Details. Wenn die RAM-Disk-Unterstützung (BLK_DEV_RAM) ebenfalls enthalten ist, aktiviert dies auch die anfängliche RAM-Disk-Unterstützung (initrd) und fügt 15 KByte (auf einigen anderen Architekturen mehr) zur Kernelgröße hinzu.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.1 Initramfs source file(s)

CONFIG_INITRAMFS_SOURCE [=] [ ]
Dies kann entweder ein einzelnes cpio-Archiv mit der Endung .cpio oder eine durch Leerzeichen getrennte Liste von Verzeichnissen und Dateien zur Erstellung des initramfs-Abbilds sein. Ein cpio-Archiv sollte ein Dateisystemarchiv enthalten, das als initramfs-Abbild verwendet werden soll. Verzeichnisse sollten ein Dateisystem-Layout enthalten, das in das initramfs-Abbild aufgenommen werden soll. Die Dateien sollten Einträge in dem Format enthalten, das vom Programm `usr/gen_init_cpio` im Kernelbaum beschrieben wird. Wenn mehrere Verzeichnisse und Dateien angegeben werden, wird das initramfs-Abbild die Summe aller dieser Verzeichnisse und Dateien sein.
Siehe <file:Documentation/driver-api/early-userspace/early_userspace_support.rst> für weitere Details.
Wenn Sie sich nicht sicher sind, lassen Sie das Feld leer.
Symbol: INITRAMFS_SOURCE [=]
Type : string (Zeichenkette)

1.36.2 Support initial ramdisk/ramfs compressed using gzip

CONFIG_RD_GZIP [=y] **[Y]**
Unterstützung des Ladens eines gzip-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.3 Support initial ramdisk/ramfs compressed using bzip2

CONFIG_RD_BZIP2 [=y] **[Y]**
Unterstützung des Ladens eines bzip2-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.4 Support initial ramdisk/ramfs compressed using LZMA

CONFIG_RD_LZMA [=y] **[Y]**
Unterstützung des Ladens eines LZMA-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.5 Support initial ramdisk/ramfs compressed using XZ

CONFIG_RD_XZ [=y] **[Y]**
Unterstützung des Ladens eines XZ-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.6 Support initial ramdisk/ramfs compressed using LZO

CONFIG_RD_LZO [=y] **[Y]**
Unterstützung des Ladens eines LZO-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.7 Support initial ramdisk/ramfs compressed using LZ4

CONFIG_RD_LZ4 [=y] **[Y]**
Unterstützung des Ladens eines LZ4-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.36.8 Support initial ramdisk/ramfs compressed using ZSTD

CONFIG_RD_ZSTD [=y] **[Y]**
Unterstützung des Ladens eines ZSTD-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.
Wenn Sie unsicher sind, sagen Sie Y.

1.37 Boot config support

CONFIG_BOOT_CONFIG [=y] **[Y]**
Extra boot config ermöglicht es dem Systemadministrator, eine Konfigurationsdatei als zusätzliche Erweiterung der Kernel-Cmdline beim Booten zu übergeben. Die Bootkonfigurationsdatei muss am Ende von initramfs mit Prüfsumme, Größe und magischem Wort angehängt werden.
Siehe <file:Documentation/admin-guide/bootconfig.rst> für Details.
Wenn Sie unsicher sind, sagen Sie Y.

1.37.1 Force unconditional bootconfig processing

CONFIG_BOOT_CONFIG_FORCE [=n] **[N]**
Wenn diese Kconfig-Option gesetzt ist, wird die BOOT_CONFIG-Verarbeitung auch dann durchgeführt, wenn der Kernel-Boot-Parameter "bootconfig"weggelassen wird. Tatsächlich gibt es mit dieser Kconfig-Option keine Möglichkeit, den Kernel dazu zu bringen, die von BOOT_CONFIG gelieferten Kernel-Boot-Parameter zu ignorieren.
Wenn Sie unsicher sind, sagen Sie N.

1.37.2 Embed bootconfig file in the kernel

CONFIG_BOOT_CONFIG_EMBED [=n] **[N]**
Eine mit BOOT_CONFIG_EMBED_FILE angegebene bootconfig-Datei in den Kernel einbetten. Normalerweise wird die bootconfig-Datei mit dem initrd-Image geladen. Wenn das System jedoch initrd nicht unterstützt, hilft Ihnen diese Option, indem sie eine bootconfig-Datei beim Erstellen des Kernels einbettet.
Wenn Sie unsicher sind, sagen Sie N.

1.38 Preserve cpio archive mtimes in initramfs

CONFIG_INITRAMFS_PRESERVE_MTIME [=y] **[Y]**
Jeder Eintrag in einem initramfs cpio-Archiv enthält einen mtime-Wert. Wenn diese Option aktiviert ist, übernehmen die extrahierten cpio-Einträge diese mtime, wobei die mtime-Einstellung des Verzeichnisses aufgeschoben wird, bis nach der Erstellung aller untergeordneten Einträge.
Wenn Sie unsicher sind, sagen Sie Y.

1.39 Compiler optimization level →

Optimierungsgrad des Compilers, Auswahl aus den folgenden zwei Punkten:

1.39.1 Optimize for performance (-O2)

CONFIG_CC_OPTIMIZE_FOR_Performance [=y] **[Y]**
Dies ist die Standardoptimierungsstufe für den Kernel, die mit dem Compiler-Flag `-O2` erstellt wird, um die beste Leistung und die hilfreichsten Warnungen bei der Kompilierung zu erhalten.

1.39.2 Optimize for size (-Os)

CONFIG_CC_OPTIMIZE_FOR_SIZE [=n] **[N]**
Wenn Sie diese Option wählen, wird `-Os` an Ihren Compiler übergeben, was zu einem kleineren Kernel führt.

1.40 Configure standard kernel features (expert users)

CONFIG_EXPERT [=n] [ ]
Mit dieser Option können bestimmte Basis-Kerneloptionen und -einstellungen deaktiviert oder optimiert werden. Dies ist für spezielle Umgebungen gedacht, die einen „Nicht-Standard“-Kernel tolerieren können. Verwenden Sie diese Option nur, wenn Sie wirklich wissen, was Sie tun.

1.40.1 Load all symbols for debugging/ksymoops

CONFIG_KALLSYMS [=y] **[Y]**
(sichtbar wenn EXPERT [=n])
Geben Sie hier Y ein, damit der Kernel symbolische Absturzinformationen und symbolische Stack-Backtraces ausgibt. Dies erhöht die Größe des Kernels etwas, da alle Symbole in das Kernel-Image geladen werden müssen.

1.40.1.1 Test the basic functions and performance of kallsyms

CONFIG_KALLSYMS_SELFTEST [=n] **[N]**
Testen Sie die Grundfunktionen und die Leistung einiger Schnittstellen, wie z. B. `kallsyms_lookup_name`. Außerdem wird die Kompressionsrate des kallsyms-Kompressionsalgorithmus für den aktuellen Symbolsatz berechnet. Starten Sie den Selbsttest automatisch nach dem Systemstart.
Es wird empfohlen, `dmesg | grep kallsyms_selftest` auszuführen, um die Testergebnisse zu sammeln. In der letzten Zeile wird `finish` angezeigt, was bedeutet, dass der Test abgeschlossen ist.

1.40.1.2 Include all symbols in kallsyms

CONFIG_KALLSYMS_ALL [=y] **[Y]**
Normalerweise enthält kallsyms nur die Symbole von Funktionen für schönere OOPS-Meldungen und Backtraces (d. h. Symbole aus den Abschnitten text und inittext). Dies ist für die meisten Fälle ausreichend. Nur wenn Sie Kernel-Live-Patching oder andere weniger häufige Anwendungsfälle (z. B. wenn ein Debugger verwendet wird) aktivieren wollen, sind alle Symbole erforderlich (d. h. die Namen von Variablen aus den Data-Abschnitten usw.).
Diese Option stellt sicher, dass alle Symbole in das Kernel-Image geladen werden (d.h. Symbole aus allen Sektionen), was die Kernelgröße erhöht (je nach Kernelkonfiguration kann sie 300KiB oder etwas Ähnliches betragen).
Sagen Sie N, es sei denn, Sie brauchen wirklich alle Symbole, oder Kernel-Live-Patching.

1.41 Kernel Performance Events And Counters →

Kernel-Leistungsereignisse und -Zähler

1.41.1 Kernel performance events and counters

CONFIG_PERF_EVENTS [=y] **[Y]**
Aktivieren Sie die Kernel-Unterstützung für verschiedene von Software und Hardware bereitgestellte Leistungsereignisse.

Software-Ereignisse werden entweder integriert oder über die Verwendung von generischen Tracepoints unterstützt.

Die meisten modernen CPUs unterstützen Leistungsereignisse über Leistungszählerregister. Diese Register zählen die Anzahl bestimmter Arten von hw-Ereignissen: z. B. ausgeführte Anweisungen, erlittene Cachemisses oder falsch vorhergesagte Verzweigungen – ohne den Kernel oder Anwendungen zu verlangsamen. Diese Register können auch Unterbrechungen auslösen, wenn eine bestimmte Anzahl von Ereignissen überschritten wird – und können so dazu verwendet werden, ein Profil des Codes zu erstellen, der auf dieser CPU läuft.

Das Linux-Performance-Event-Subsystem bietet eine Abstraktion dieser Software- und Hardware-Event-Fähigkeiten, die über einen Systemaufruf zugänglich sind und von dem Dienstprogramm `perf` in `tools/perf/` verwendet werden. Es stellt Zähler pro Task und pro CPU zur Verfügung und bietet darüber hinaus Ereignisfunktionen.
Sagen Sie Y, wenn Sie unsicher sind.

1.41.1.1 Debug: use vmalloc to back perf mmap() buffers

CONFIG_DEBUG_PERF_USE_VMALLOC [=n] **[N]**
Verwendung von vmalloc-Speicher zur Sicherung von mmap()-Puffern. Hauptsächlich nützlich zum Debuggen des vmalloc-Codes auf Plattformen, die dies nicht erfordern. Sagen Sie N, wenn Sie unsicher sind.

1.42 Profiling support

CONFIG_PROFILING [=y] **[Y]**
Sagen Sie hier Y, um die erweiterten Unterstützungsmechanismen für das Profiling zu aktivieren, die von Profilern verwendet werden.

1.43 Kexec and crash features →

Kexec und Absturzmerkmale

1.43.1 Enable kexec system call

CONFIG_KEXEC [=y] **[Y]**
`kexec` ist ein Systemaufruf, der die Fähigkeit implementiert, den aktuellen Kernel herunterzufahren und einen anderen Kernel zu starten. Es ist wie ein Neustart, aber er ist unabhängig von der System-Firmware. Und wie ein Neustart können Sie damit jeden Kernel starten, nicht nur Linux. Der Name kommt von der Anlehnung mit dem Systemaufruf `exec`. Es ist ein fortlaufender Prozess, um sicher zu sein, dass die Hardware eines Rechners ordnungsgemäß heruntergefahren wird, seien Sie also nicht überrascht, wenn dieser Code bei Ihnen zunächst nicht funktioniert. Zum Zeitpunkt des Verfassens dieses Artikels ist die genaue Hardwareschnittstelle noch stark im Wandel, so dass keine gute Empfehlung ausgesprochen werden kann.

1.43.2 Enable kexec file based system call

CONFIG_KEXEC_FILE [=y] **[Y]**
(Aktivieren des dateibasierten Systemaufrufs kexec)
Dies ist eine neue Version des Systemaufrufs `kexec`. Dieser Systemaufruf ist dateibasiert und nimmt Dateideskriptoren als Systemaufrufsargument für Kernel und initramfs anstelle einer Liste von Segmenten, wie sie vom kexec-Systemaufruf akzeptiert wird.

1.43.2.1 Verify kernel signature during kexec_file_load() syscall

CONFIG_KEXEC_SIG [=y] **[Y]**
Mit dieser Option wird der Syscall `kexec_file_load()` auf eine gültige Signatur des Kernel-Images geprüft. Das Image kann immer noch ohne gültige Signatur geladen werden, es sei denn, Sie aktivieren auch KEXEC_SIG_FORCE, aber wenn es eine Signatur gibt, die überprüft werden kann, dann muss sie auch gültig sein. Zusätzlich zu dieser Option müssen Sie die Signaturprüfung für den entsprechenden Kernel-Image-Typ, der geladen wird, aktivieren, damit dies funktioniert.

1.43.2.1.1 Require a valid signature in kexec_file_load() syscall

CONFIG_KEXEC_SIG [=n] **[N]**
Diese Option macht die Überprüfung der Kernelsignatur für den Syscall `kexec_file_load()` zwingend erforderlich.

1.43.2.1.2 Enable bzImage signature verification support

CONFIG_KEXEC_BZIMAGE_VERIFY_SIG [=n] **[N]**
Aktivierung der Unterstützung von bzImage für die Signaturprüfung.

1.43.3 kexec jump

CONFIG_KEXEC_JUMP [=y] **[Y]**
Sprung zwischen Original-Kernel und kexeced-Kernel und Aufruf von Code im physikalischen Adressmodus über KEXEC

1.43.4 kexec crash dumps

CONFIG_KEXEC_DUMP [=y] **[Y]**
Absturzdump (Speicherauszug) erzeugen, nachdem er von kexec gestartet wurde. Dies sollte normalerweise nur in speziellen Crash-Dump-Kerneln gesetzt werden, die im Hauptkernel mit kexec-tools in einen speziell reservierten Bereich geladen werden und dann später nach einem Absturz von kdump/kexec ausgeführt werden. Der Crash-Dump-Kernel muss mit PHYSICAL_START auf eine Speicheradresse kompiliert werden, die nicht vom Hauptkernel oder BIOS verwendet wird, oder er muss als relocatable image (CONFIG_RELOCATABLE=y) erstellt werden.
Für weitere Details siehe Documentation/admin-guide/kdump/kdump.rst

Für s390 aktiviert diese Option auch zfcpdump.
Siehe auch <file:Documentation/s390/zfcpdump.rst>

1.43.4.1 Update the crash elfcorehdr on system configuration changes

CONFIG_CRASH_HOTPLUG [=y] **[Y]**
Aktivierung der direkten Aktualisierung der Crash-Elfcorehdr (die die Liste der CPUs und Speicherbereiche enthält, die bei einem Absturz gelöscht werden sollen) als Reaktion auf Hot-Plug/Unplug oder Online/Offline von CPUs oder Speicher. Dies ist ein sehr viel fortschrittlicherer Ansatz als der Versuch dies im Userspace zu tun.
Wenn Sie unsicher sind, sagen Sie Y.

1.43.4.1.1 Specify the maximum number of memory regions for the elfcorehdr

CONFIG_CRASH_MAX_MEMORY_RANGES [=8192] **[8192]**
Für den Pfad des Systemaufrufs `kexec_file_load()` ist die maximale Anzahl der Speicherbereiche anzugeben, die der elfcorehdr-Puffer/das elfcorehdr-Segment aufnehmen kann. Diese Regionen werden über `walk_system_ram_res()` ermittelt, z. B. die 'System RAM'-Einträge in /proc/iomem. Dieser Wert wird mit NR_CPUS_DEFAULT kombiniert und mit `sizeof(Elf64_Phdr)` multipliziert, um die endgültige elfcorehdr-Speicherpuffer-/Segmentgröße zu bestimmen. Der Wert 8192 beispielsweise deckt ein (dünn besiedeltes) 1TiB-System ab, das aus 128MiB-Memblöcken besteht, und führt zu einer elfcorehdr-Speicherpuffer-/Segmentgröße von unter 1MiB. Dies ist eine vernünftige Wahl, um sowohl Baremetal- als auch virtuelle Maschinenkonfigurationen zu unterstützen.
Für den Syscall-Pfad `kexec_load()` ist CRASH_MAX_MEMORY_RANGES Teil der Berechnung hinter dem Wert, der über das Attribut /sys/kernel/crash_elfcorehdr_size bereitgestellt wird.

2 64-bit kernel

CONFIG_64BIT [=y] **[Y]**
Sagen Sie Y für ja, zur Erstellung eines 64-Bit-Kernels - früher bekannt als x86_64
Sagen Sie N für nein, um einen 32-Bit-Kernel zu erstellen - früher bekannt als i386

3 Processor type and features →

Prozessortyp und Eigenschaften

3.1 Symmetric multi-processing support

CONFIG_SMP [=y] **[Y]**
Dies ermöglicht die Unterstützung von Systemen mit mehr als einer CPU. Wenn Sie ein System mit nur einer CPU haben, sagen Sie N. Wenn Sie ein System mit mehr als einer CPU haben, sagen Sie Y. Wenn Sie hier N angeben, läuft der Kernel auf Uni- und Multiprozessor-Maschinen, verwendet aber nur eine CPU einer Multiprozessor-Maschine. Wenn Sie hier Y angeben, läuft der Kernel auf vielen, aber nicht auf allen Uniprozessor-Maschinen.

Auf einer Uniprozessor-Maschine läuft der Kernel schneller, wenn Sie hier N angeben. Beachten Sie, dass der Kernel nicht auf 486er-Architekturen läuft, wenn Sie hier Y angeben und unter „Prozessorfamilie“ die Architektur „586“ oder „Pentium“ auswählen.

Ebenso funktionieren Multiprozessor-Kernel für die „PPro“-Architektur möglicherweise nicht auf allen Pentium-basierten Boards.

Benutzer von Multiprozessor-Maschinen, die hier Y für „Ja“ angeben, sollten auch „Ja“ zu „Enhanced Real Time Clock Support“ (siehe unten) sagen. Der „Advanced Power Management“-Code wird deaktiviert, wenn Sie hier „Y“ angeben. Siehe auch <file:Documentation/arch/x86/i386/IO-APIC.rst>, <file:Documentation/admin-guide/lockup-watchdogs.rst> und das SMP-HOWTO, verfügbar unter: `http://www.tldp.org/docs.html#howto`.
Wenn Sie nicht wissen, was Sie hier tun sollen, sagen Sie N.

3.2 Support x2apic

CONFIG_X86_X2APIC [=y] **[Y]**
Dies ermöglicht die Unterstützung von x2apic auf CPUs, die über diese Funktion verfügen. Dies ermöglicht 32-Bit-Apic-IDs (so dass es sehr große Systeme unterstützen kann) und greift auf den lokalen apic über MSRs und nicht über mmio zu. Einige Intel-Systeme ab ca. 2022 sind in den x2APIC-Modus gesperrt und können nicht auf die alten APIC-Modi zurückgreifen, wenn SGX oder TDX im BIOS aktiviert sind. Ohne Aktivierung dieser Option booten sie mit stark eingeschränkter Funktionalität.
Wenn Sie nicht wissen, was Sie hier tun sollen, sagen Sie N.

3.3 Enable MPS table

CONFIG_X86_MPPARSE [=y] **[Y]**
Für alte smp-Systeme, die keine richtige acpi-Unterstützung haben. Neuere Systeme (insbesondere mit 64bit-CPUs) mit acpi-Unterstützung, werden von MADT und DSDT überschrieben.

3.4 x86 CPU resource control support

CONFIG_X86_CPU_RESCTRL [=y] **[Y]**
Aktivieren Sie die Unterstützung der x86-CPU-Ressourcensteuerung. Unterstützung für die Zuweisung und Überwachung der Nutzung von Systemressourcen durch die CPU. Intel nennt dies Intel Resource Director Technology (Intel(R) RDT). Weitere Informationen über RDT finden Sie im Intel x86 Architecture Software Developer Manual. AMD bezeichnet dies als AMD Platform Quality of Service (AMD QoS).
Weitere Informationen über AMD QoS finden Sie im Handbuch AMD64 Technology Platform Quality of Service Extensions.
Sagen Sie N, wenn Sie unsicher sind.

3.5 Support for extended (non-PC) x86 platforms

CONFIG_X86_EXTENDED_PLATFORM [=n] **[N]**
Wenn Sie diese Option deaktivieren, unterstützt der Kernel nur Standard-PC-Plattformen (was die große Mehrheit der Systeme da draußen abdeckt). Wenn Sie diese Option aktivieren, können Sie die Unterstützung für die folgenden (nicht-PC) 64-Bit-x86-Plattformen auswählen:

- Numascale NumaChip
- ScaleMP vSMP
- SGI Ultraviolet

Wenn Sie eines dieser Systeme haben, oder wenn Sie einen generischen Distributionskernel bauen wollen, geben Sie hier Y an – andernfalls sagen Sie N.

3.6 Intel Low Power Subsystem Support

CONFIG_X86_INTEL_LPSS [=y] **[Y]**
Wählen Sie diese Option, um Unterstützung für das Intel Low Power Subsystem zu erstellen, wie es auf dem Intel Lynxpoint PCH zu finden ist. Die Auswahl dieser Option ermöglicht Dinge wie Clock Tree (Common Clock Framework) und Pincontrol, die von den LPSS-Peripherietreibern benötigt werden.

3.7 AMD ACPI2Platform devices support

CONFIG_X86_AMD_PLATFORM_DEVICE [=y] **[Y]**
Wählen Sie diese Option, um AMD-spezifische ACPI-Geräte wie I2C, UART, GPIO, die auf AMD Carrizo und späteren Chipsätzen zu finden sind, als Plattformgeräte zu interpretieren. I2C und UART hängen von COMMON_CLK ab, um den Takt zu setzen. Der GPIO-Treiber ist im PINCTRL-Subsystem implementiert.

3.8 Intel SoC IOSF Sideband support for SoC platforms

CONFIG_IOSF_MBI [=y] **[Y]**
Diese Option aktiviert die Unterstützung des Seitenband-Registerzugriffs für Intel SoC-Plattformen. Auf diesen Plattformen wird das IOSF-Seitenband anstelle von MSRs für einige Registerzugriffe verwendet, vor allem, aber nicht ausschließlich, für thermische und Stromversorgungs-Register. Treiber können die Verfügbarkeit dieser Geräte abfragen, um festzustellen, ob sie das Seitenband benötigen, um auf diesen Plattformen zu funktionieren. Das Seitenband ist auf den folgenden SoC-Produkten verfügbar.

- BayTrail
- Braswell
- Quark

Sie sollten Y sagen, wenn Sie einen Kernel auf einem dieser SoCs ausführen.

3.8.1 Enable IOSF sideband access through debugfs

CONFIG_IOSF_MBI_DEBUG [=n] **[N]**
Wählen Sie diese Option, um die IOSF-Seitenband-Zugriffsregister (MCR, MDR, MCRX) über debugfs freizugeben, um Registerinformationen von verschiedenen Einheiten auf dem SoC zu schreiben und zu lesen. Dies ist sehr nützlich, um Informationen über den Gerätezustand für Debugging und Analyse zu erhalten. Da es sich um einen allgemeinen Zugriffsmechanismus handelt, müssen die Benutzer dieser Option das Gerät, auf das sie zugreifen wollen, genau kennen.
Wenn Sie die Option nicht benötigen oder im Zweifel sind, sagen Sie N.

3.9 Single-depth WCHAN output

CONFIG_SHED_OMIT_FRAME_POINTER [=y] **[Y]**
Berechne einfachere /proc/<PID>/wchan-Werte. Wenn diese Option deaktiviert ist, werden die wchan-Werte zur aufrufenden Funktion zurückgeführt. Dies liefert genauere wchan-Werte, allerdings auf Kosten eines etwas größeren Planungsaufwands (scheduling overhead).
Im Zweifelsfall sagen Sie "Y".

3.10 Linux guest support →

CONFIG_HYPERVISOR_GUEST [=y] **[Y]**
Geben Sie hier Y ein, um Optionen für die Ausführung von Linux unter verschiedenen Hypervisoren zu aktivieren. Diese Option aktiviert die grundlegende Hypervisor-Erkennung und die Einrichtung der Plattform. Wenn Sie N sagen, werden alle Optionen in diesem Untermenü übersprungen und deaktiviert, und die Linux-Gastunterstützung wird nicht eingebaut.

3.10.1 Enable paravirtualization code

CONFIG_PARAVIRT [=y] **[Y]**
Der Kernel wird so verändert, dass er sich selbst modifizieren kann, wenn er unter einem Hypervisor ausgeführt wird, was die Leistung gegenüber einer vollständigen Virtualisierung erheblich verbessern kann. Wenn der Kernel jedoch ohne Hypervisor ausgeführt wird, ist er theoretisch langsamer und etwas größer.

3.10.2 paravirt-ops debugging

CONFIG_PARAVIRT_DEBUG [=n] **[N]**
Ermöglicht das Debuggen von paravirt_ops Interna. Insbesondere BUG, wenn eine paravirt_op fehlt, wenn sie aufgerufen wird.

3.10.3 Paravirtualization layer for spinlocks

CONFIG_PARAVIRT_SPINLOCKS [=y] **[Y]**
Paravirtualisierte Spinlocks ermöglichen es einem pvops-Backend, die Spinlock-Implementierung durch etwas Virtualisierungsfreundliches zu ersetzen (z. B. Blockieren der virtuellen CPU anstelle von Spinning). Dies hat nur minimale Auswirkungen auf native Kernel und bringt einen deutlichen Leistungsvorteil für paravirtualisierte KVM/Xen-Kernel.
Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollen, antworten Sie mit Y.

3.10.4 Xen guest support

CONFIG_XEN [=y] **[Y]**
Dies ist der Linux-Xen-Port. Wenn Sie dies aktivieren, kann der Kernel in einer paravirtualisierten Umgebung unter dem Xen-Hypervisor booten.

3.10.4.1 Xen PV guest support

CONFIG_XEN_PV [=y] **[Y]**
Der Betrieb als Xen PV-Gast wird unterstützt.

3.10.4.1.1 Limit Xen pv-domain memory to 512GB

CONFIG_XEN_512GB [=y] **[Y]**
Begrenzen der paravirtualisierten Benutzerdomänen auf 512 GB RAM. Die Xen-Tools und die Tools zur Analyse von Crash-Dumps unterstützen möglicherweise keine pv-Domänen mit mehr als 512 GB RAM. Diese Option steuert die Standardeinstellung des Kernels, um nur bis zu 512 GB oder mehr zu verwenden. Es ist jederzeit möglich, die Standardeinstellung durch Angabe des Boot-Parameters `xen_512gb_limit` zu ändern.

3.10.4.2 Xen PVHVM guest support

CONFIG_XEN_PVHVM_GUEST [=y] **[Y]**
Der Betrieb als Xen PVHVM-Gast wird unterstützt.

3.10.4.3 Enable Xen debug and tuning parameters in debugfs

CONFIG_XEN_DEBUG_FS [=n] **[N]**
Der Betrieb als Xen PV-Gast wird unterstützt.

3.10.4.4 Xen PVH guest support

CONFIG_XEN_PVH [=y] **[Y]**
Der Betrieb als Xen PVH-Gast wird unterstützt.

3.10.5 Xen Dom0 support

CONFIG_XEN_DOM0 [=y] **[Y]**
Der Betrieb als Xen Dom0-Gast wird unterstützt.

3.10.6 Always use safe MSR accesses in PV guests

CONFIG_XEN_PV_MSR_SAFE [=y] **[Y]**
Verwenden Sie sichere (nicht fehlerhafte) MSR-Zugriffsfunktionen, auch wenn der MSR-Zugriff ohnehin nicht fehlerhaft sein sollte. Der Standardwert kann mit dem Boot-Parameter `xen_msr_safe` geändert werden.

3.10.7 KVM Guest support (including kvmclock)

CONFIG_KVM_GUEST [=y] **[Y]**
Diese Option ermöglicht verschiedene Optimierungen für die Ausführung unter dem KVM-Hypervisor. Sie beinhaltet eine paravirtualisierte Uhr, so dass der Host dem Gast eine Zeitinfrastruktur wie die Tageszeit und die Systemzeit zur Verfügung stellt, anstatt sich auf eine PIT-Emulation (oder wahrscheinlich eine andere) durch das zugrunde liegende Gerätemodell zu verlassen.

3.10.8 Disable host haltpoll when loading haltpoll driver

CONFIG_ARCH_CPUIDLE_HALTPOLL [=y] **[Y]**
(Haltpoll des Hosts beim Laden des Haltpoll-Treibers deaktivieren)
Wenn Sie unter KVM virtualisieren, deaktiviert den haltpoll des Hosts.

3.10.9 Support for running PVH guests

CONFIG_PVH [=y] **[Y]**
Diese Option aktiviert den PVH-Einstiegspunkt für virtuelle Gastmaschinen, wie in der x86/HVM Direct Boot ABI angegeben.

3.10.10 Paravirtual steal time accounting

CONFIG_PARAVIRT_TIME_ACCOUNTING [=y] **[Y]**
Wählen Sie diese Option aus, um die Berechnung der Zeit für das Stehlen von Aufgaben mit feiner Granularität zu aktivieren. Die Zeit, die für die Ausführung anderer Aufgaben parallel zur aktuellen vCPU aufgewendet wird, ist von der vCPU-Leistung abgezogen. Um dies zu berücksichtigen, kann es zu geringen Leistungseinbußen kommen.
Im Zweifelsfall geben Sie hier N an.

3.10.11 Jailhouse non-root cell support

CONFIG_JAILHOUSE_GUEST [=y] **[Y]**
Diese Option ermöglicht es, Linux als Gast in einer Jailhouse-Nicht-Root-Zelle auszuführen. Sie können diese Option deaktiviert lassen, wenn Sie Jailhouse nur starten und Linux anschließend in der Root-Zelle ausführen möchten.

3.10.12 ACRN Guest support

CONFIG_ACRN_GUEST [=y] **[Y]**
Diese Option ermöglicht die Ausführung von Linux als Gast im ACRN-Hypervisor. ACRN ist ein flexibler, leichtgewichtiger Referenz-Open-Source-Hypervisor, der mit Blick auf Echtzeit und Sicherheitskritik entwickelt wurde. Er wurde für eingebettete IOT mit kleinem Platzbedarf und Echtzeitfunktionen entwickelt. Weitere Einzelheiten finden Sie unter `https://projectacrn.org/`.

3.10.13 Intel TDX (Trust Domain Extensions) - Guest Support

CONFIG_INTEL_TDX_GUEST [=y] **[Y]**
Unterstützung der Ausführung als Gast unter Intel TDX.
Ohne diese Unterstützung kann der Gastkernel nicht booten oder unter TDX laufen. TDX umfasst Speicherverschlüsselungs- und Integritätsfunktionen, die die Vertraulichkeit und Integrität des Gastspeicherinhalts und des CPU-Status schützen. TDX-Gäste sind vor einigen Angriffen durch den VMM geschützt.

3.11 Processor family (Generic-x86-64) →

Dies ist der Prozessortyp Ihrer CPU. Diese Information wird für Optimierungszwecke verwendet. Um einen Kernel zu kompilieren, der auf allen unterstützten x86-CPU-Typen laufen kann (wenn auch nicht optimal schnell), können Sie hier „486“ angeben. Beachten Sie, dass der 386er nicht mehr unterstützt wird, dies schließt AMD/Cyrix/Intel 386DX/DXL/SL/SLC/SX, Cyrix/TI 486DLC/DLC2, UMC 486SX-S und den NexGen Nx586 ein. Der Kernel läuft nicht notwendigerweise auf älteren Architekturen als der von Ihnen gewählten, z. B. läuft ein Pentium-optimierter Kernel auf einem PPro, aber nicht unbedingt auf einem i486.

Hier sind die empfohlenen Einstellungen für höchste Geschwindigkeit:

- `486` für den AMD/Cyrix/IBM/Intel 486DX/DX2/DX4 oder SL/SLC/SLC2/SLC3/SX/SX2 und UMC U5D oder U5S.
- `586` für generische Pentium-CPUs, denen das TSC-Register (Zeitstempelzähler) fehlt.
- `Pentium-Classic` für den Intel Pentium.
- `Pentium-MMX` für den Intel Pentium MMX.
- `Pentium-Pro` für den Intel Pentium Pro.
- `Pentium-II` für den Intel Pentium II oder den Pre-Coppermine Celeron.
- `Pentium-III` für den Intel Pentium III oder Coppermine Celeron.
- `Pentium-4` für den Intel Pentium 4 oder den P4-basierten Celeron.
- `K6` für den AMD K6, K6-II und K6-III (auch bekannt als K6-3D).
- `Athlon` für die AMD K7-Familie (Athlon/Duron/Thunderbird).
- `Opteron/Athlon64/Hammer/K8` für alle K8 und neuere AMD-CPUs.
- `Crusoe` für die Transmeta Crusoe-Serie.
- `Efficeon` für die Transmeta Efficeon-Reihe.
- `Winchip-C6` für den ursprünglichen IDT-Winchip.
- `Winchip-2` für IDT-Winchips mit 3dNow! Fähigkeiten.
- `AMD Elan` für die 32-Bit AMD Elan Embedded CPU.
- `GeodeGX1` für Geode GX1 (Cyrix MediaGX).
- `Geode GX/LX` für AMD Geode GX und LX Prozessoren.
- `CyrixIII/VIA C3` für VIA Cyrix III oder VIA C3.
- `VIA C3-2` für VIA C3-2 "Nehemiah"(Modell 9 und höher).
- `VIA C7` für VIA C7.
- `Intel P4` für die Pentium 4/Netburst-Mikroarchitektur.
- `Core 2/newer Xeon` für alle Core2 und neueren Intel-CPUs.
- `Intel Atom` für die CPUs mit Atom-Mikroarchitektur.
- `Generic-x86-64` für einen Kernel, der auf jeder x86-64-CPU läuft.

Weitere Details finden Sie im Hilfetext der jeweiligen Option. Wenn Sie nicht wissen, was Sie tun sollen, wählen Sie `486`.

Derzeit (Kernelversion 6.6.x) können Sie nur aus fünf auswählen:

3.11.1 Opteron/Athlon64/Hammer/K8

CONFIG_MK8 [=n] **[N]**
Wählen Sie diese Option für einen Prozessor der AMD Opteron- oder Athlon64 Hammer-Familie. Ermöglicht die Verwendung einiger erweiterter Anweisungen und übergibt entsprechende Optimierungsflags an den GCC.

3.11.2 Intel P4 / older Netburst based Xeon

CONFIG_MPSC [=n] **[N]**
Optimiert für Intel Pentium 4, Pentium D und ältere Nocona/Dempsey Xeon CPUs mit Intel 64bit, die mit x86-64 kompatibel sind. Beachten Sie, dass die neuesten Xeons (Xeon 51xx und 53xx) nicht auf dem Netburst-Kern basieren und diese Option nicht verwenden sollten.
Sie können sie anhand des Feldes cpu family in /proc/cpuinfo unterscheiden. Familie 15 ist ein älterer Xeon, Familie 6 ein neuerer.

3.11.3 Intel P4 / older Netburst based Xeon

CONFIG_MCORE2 [=n] **[Y]**
Wählen Sie dies für Intel Core 2 und neuere Core 2 Xeons (Xeon 51xx und 53xx) CPUs.
Sie können neuere von älteren Xeons anhand der CPU-Familie in /proc/cpuinfo unterscheiden. Neuere haben 6 und ältere 15 (kein Tippfehler).

3.11.4 Intel Atom

CONFIG_MATOM [=n] **[N]**
Wählen Sie diese Option für die Intel Atom-Plattform. Intel Atom CPUs haben eine In-Order-Pipelining-Architektur und können daher von entsprechend optimiertem Code profitieren. Verwenden Sie einen aktuellen GCC mit spezieller Atom-Unterstützung, um die Vorteile dieser Option voll ausschöpfen zu können.

3.11.5 Generic-x86-64

CONFIG_GENERIC_CPU [=y] **[N]**
Allgemeine x86-64-CPU. Läuft gleich gut auf allen x86-64-CPUs.

3.12 Old AMD GART IOMMU support

CONFIG_GART_IOMMU [=n] **[N]**
Bietet einen Treiber für ältere AMD Athlon64/Opteron/Turion/Sempron GART basierte Hardware IOMMUs an. Der GART unterstützt vollen DMA-Zugriff für Geräte mit 32-Bit-Zugriffsbeschränkungen auf Systemen mit mehr als 3 GB. Dies wird normalerweise für USB, Sound, viele IDE/SATA-Chipsätze und einige andere Geräte benötigt. Neuere Systeme haben in der Regel eine moderne AMD IOMMU, die über die Konfigurationsoption CONFIG_AMD_IOMMU=y unterstützt wird. In normalen Konfigurationen ist dieser Treiber nur aktiv, wenn er benötigt wird: Es sind mehr als 3 GB Arbeitsspeicher vorhanden und das System enthält ein auf 32 Bit begrenztes Gerät.
Wenn Sie unsicher sind, sagen Sie Y.

3.13 Enable Maximum number of SMP Processors and NUMA Nodes

CONFIG_MAXSMP [=n] **[N]**
Aktivieren der maximalen Anzahl von CPUs- und NUMA-Knoten für diese Architektur.
Wenn Sie unsicher sind, sagen Sie N.

3.14 Maximum number of CPUs

CONFIG_NR_CPUS [=320] **[8]**
Hier können Sie die maximale Anzahl von CPUs angeben, die dieser Kernel unterstützen soll. Wenn CPUMASK_OFFSTACK aktiviert ist, ist der maximal unterstützte Wert 8192, andernfalls ist der maximale Wert 512. Der Mindestwert, der Sinn macht, ist 2.

Dies dient lediglich dazu, Speicher zu sparen: jede unterstützte CPU fügt dem Kernel-Image etwa 8 kB hinzu.

3.15 Cluster scheduler support

CONFIG_SCHED_CLUSTER [=y] **[N]**
Die Unterstützung des Cluster-Schedulers verbessert die Entscheidungsfindung des CPU-Schedulers beim Umgang mit Maschinen, die Cluster von CPUs haben. Mit Cluster sind in der Regel mehrere CPUs gemeint, die eng beieinander liegen und sich Mid-Level-Caches, Last-Level-Cache-Tags oder interne Busse teilen.

3.16 Multi-core scheduler support

CONFIG_SCHED_MC [=y] **[Y]**
Die Unterstützung des Multi-Core-Schedulers verbessert die Entscheidungsfindung des CPU-Schedulers beim Umgang mit Multi-Core-CPU-Chips auf Kosten eines leicht erhöhten Overheads an einigen Stellen. Wenn Sie unsicher sind, geben Sie hier N an.

3.16.1 CPU core priorities scheduler support

CONFIG_SCHED_MC_PRIO [=y] **[Y]**
Bei CPUs mit Intel Turbo-Boost-Max-Technik 3.0 wird die Reihenfolge der Kerne bei der Herstellung festgelegt, so dass bestimmte Kerne höhere Turbofrequenzen erreichen können (bei Single-Thread-Arbeitslasten) als andere. Durch die Aktivierung dieser Kernel-Funktion wird der Scheduler über die TBM3- (auch ITMT-) Prioritätsreihenfolge der CPU-Kerne informiert und passt die CPU-Auswahllogik des Schedulers entsprechend an, so dass eine höhere Gesamtsystemleistung erzielt werden kann. Diese Funktion hat keine Auswirkungen auf CPUs ohne diese Funktion.
Wenn Sie unsicher sind, geben Sie hier Y an.

3.17 Reroute for broken boot IRQs

CONFIG_X86_REROUTE_FOR_BROKEN_BOOT_IRQS [=y] **[Y]**
Diese Option ermöglicht eine Umgehung, die eine Quelle für unerwünschte Unterbrechungen behebt. Dies wird empfohlen, wenn die Thread-Interrupt-Behandlung auf Systemen verwendet wird, bei denen die Erzeugung von überflüssigen „Boot-Interrupts“ nicht deaktiviert werden kann. Einige Chipsätze erzeugen einen Legacy-INTx-„Boot-IRQ“, wenn der IRQ-Eintrag im IO-APIC des Chipsatzes maskiert ist (wie es z. B. der RT-Kernel während der Interruptbehandlung tut). Bei Chipsätzen, bei denen diese Boot-IRQ-Erzeugung nicht deaktiviert werden kann, wird durch diese Abhilfe die ursprüngliche IRQ-Leitung maskiert, so dass nur der entsprechende „Boot-IRQ“ an die CPUs geliefert wird. Die Problemumgehung weist den Kernel außerdem an, den IRQ-Handler auf der Boot-IRQ-Leitung einzurichten. Auf diese Weise wird nur ein Interrupt an den Kernel geliefert. Andernfalls kann der zweite Interrupt den Kernel dazu veranlassen, (lebenswichtige) Interrupt-Leitungen herunterzufahren. Betrifft nur „defekte“ Chipsätze. Die gemeinsame Nutzung von Interrupts kann auf diesen Systemen erhöht werden.

3.18 Machine Check / overheating reporting

CONFIG_X86_MCE [=y] **[Y]**
(Maschinenprüfung / Überhitzungsmeldung) Durch die Unterstützung von Machine Check kann der Prozessor den Kernel benachrichtigen, wenn er ein Problem feststellt (z. B. Überhitzung, Datenbeschädigung). Welche Maßnahmen der Kernel ergreift, hängt von der Schwere des Problems ab und reicht von Warnmeldungen bis zum Anhalten des Rechners.

3.18.1 Support for deprecated /dev/mcelog character device

CONFIG_X86_MCELOG_LEGACY [=n] **[N]**
Aktivierung der Unterstützung für /dev/mcelog, die vom alten mcelog-Benutzerraum-Logging-Daemon (mcelog userspace logging daemon) benötigt wird. Erwägen Sie den Umstieg auf die neue Generation des rasdaemon.

3.18.2 Intel MCE features

CONFIG_X86_MCE_INTEL [=y] **[Y]**
Zusätzliche Unterstützung für Intel-spezifische MCE-Funktionen wie den Temperaturmonitor (thermal monitor).

3.18.3 AMD MCE features

CONFIG_X86_MCE_AMD [=y] **[N]**
Zusätzliche Unterstützung für AMD-spezifische MCE-Funktionen wie den DRAM-Fehlerschwellenwert (DRAM Error Threshold).

3.19 Machine check injector support

CONFIG_X86_MCE_INJECT [=m] **[M]**
Unterstützung bei der Einspeisung von Maschinenprüfungen zu Testzwecken. Wenn Sie nicht wissen, was eine Maschinenprüfung ist und Sie keine Kernel-Qualitätssicherung durchführen, können Sie mit Sicherheit N sagen, also nein.

3.20 Performance monitoring →

Leistungsüberwachung

3.20.1 Intel uncore performance events

CONFIG_PERF_EVENTS_INTEL_UNCORE [=m] **[M]**
Unterstützung für Intel-Uncore-Leistungsereignisse. Diese sind auf NehalemEX und moderneren Prozessoren verfügbar.

3.20.2 Intel/AMD rapl performance events

CONFIG_PERF_EVENTS_INTEL_RAPL [=m] **[M]**
Unterstützung für Intel- und AMD-RAPL-Leistungsereignisse zur Leistungsüberwachung auf modernen Prozessoren.

3.20.3 Intel cstate performance events

CONFIG_PERF_EVENTS_INTEL_CSTATE [=m] **[M]**
Einbeziehung der Unterstützung für Intel cstate performance events für die Leistungsüberwachung auf modernen Prozessoren.

3.20.4 AMD Processor Power Reporting Mechanism

CONFIG_PERF_EVENTS_AMD_POWER [=m] **[M]**
Unterstützung von Stromversorgungsberichten für AMD-Prozessoren.
Derzeit wird die Schnittstelle X86_FEATURE_ACC_POWER (CPUID Fn8000_0007_EDX[12]) genutzt, um den durchschnittlichen Stromverbrauch von Prozessoren der Familie 15h zu berechnen.

3.20.5 AMD Uncore performance events

CONFIG_PERF_EVENTS_AMD_UNCORE [=m] **[M]**
Unterstützung für AMD-Uncore-Leistungsereignisse für die Verwendung mit z. B.
`perf stat -e amd_l3/.../,amd_df/.../`.
Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `amd-uncore` genannt.

3.20.6 AMD Zen3 Branch Sampling support

CONFIG_PERF_EVENTS_AMD_BRS [=y] **[Y]**
Aktivieren Sie die AMD Zen3 Branch Sampling-Unterstützung (BRS), die bis zu 16 aufeinanderfolgende Verzweigungen in Registern erfasst.

3.20.7 IOPERM and IOPL Emulation

CONFIG_X86_IOPL_IOPERM [=y] **[Y]**
Dies ermöglicht die ioperm()- und iopl()-Systemaufrufe, die für Legacy-Anwendungen erforderlich sind. Bei der Legacy-IOPL-Unterstützung handelt es sich um einen weitreichenden Mechanismus, der es dem Userspace ermöglicht, neben dem Zugriff auf alle 65536 E/A-Ports auch Interrupts zu deaktivieren. Um diesen Zugriff zu erhalten, benötigt der Aufrufer CAP_SYS_RAWIO-Fähigkeiten und die Erlaubnis von potenziell aktiven Sicherheitsmodulen. Die Emulation schränkt die Funktionalität des Syscalls auf den Zugriff auf alle E/A-Ports ein, verhindert aber die Möglichkeit, Interrupts aus dem Userspace zu deaktivieren, was bei Verwendung des Hardware-IOPL-Mechanismus möglich wäre.

3.20.8 Late microcode loading (DANGEROUS)

CONFIG_MICROCODE_LATE_LOADING [=n] **[N]**
Das späte Laden von Mikrocode, wenn das System bereits läuft und Befehle ausführt, ist eine heikle Angelegenheit und sollte nach Möglichkeit vermieden werden. Allein die Abfolge der Synchronisierung aller Kerne und SMT-Threads ist ein zerbrechlicher Tanz, der nicht garantiert, dass die Kerne nach dem Laden nicht softlocking werden. Daher sollten Sie dies auf eigenes Risiko tun. Das späte Laden färbt auch den Kernel.

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

CONFIG_X86_MSR [=y] **[Y]**
Dieses Gerät ermöglicht privilegierten Prozessen den Zugriff auf die modellspezifischen x86-Register (MSRs).
Es ist ein Zeichengerät mit Major 202 und Minors 0 bis 31 für `/dev/cpu/0/msr` bis `/dev/cpu/31/msr`. MSR-Zugriffe sind bei Multiprozessorsystemen an eine bestimmte CPU gerichtet.

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

CONFIG_X86_CPUID [=y] **[Y]**
Dieses Gerät ermöglicht Prozessen den Zugriff auf den x86 CPUID-Befehl, der auf einem bestimmten Prozessor ausgeführt werden soll. Es handelt sich um ein Zeichengerät mit Major 203 und Minors 0 bis 31 für `/dev/cpu/0/cpuid` bis `/dev/cpu/31/cpuid`.

3.20.11 Enable 5-level page tables support

CONFIG_X86_5LEVEL [=y] **[Y]**
5-Level-Paging ermöglicht den Zugriff auf einen größeren Adressraum: bis zu 128 PiB virtueller Adressraum und 4 PiB physikalischer Adressraum. Es wird von zukünftigen Intel-CPUs unterstützt werden. Ein Kernel mit aktivierter Option kann auf Rechnern gebootet werden, die 4- oder 5-Level-Paging unterstützen. Siehe Documentation/arch/x86/x86_64/5level-paging.rst für weitere Informationen.
Sagen Sie N, wenn Sie unsicher sind.

3.20.12 Enable statistic for Change Page Attribute

CONFIG_X86_CPA_STATISTICS [=y] **[Y]**
Statistiken über den Mechanismus zum Ändern von Seitenattributen offenlegen, der dabei hilft, die Wirksamkeit der Erhaltung großer und umfangreicher Seitenzuordnungen zu bestimmen, wenn Zuordnungsschutzmaßnahmen geändert werden.

3.20.13 AMD Secure Memory Encryption (SME) support

CONFIG_AMD_MEM_ENCRYPT [=y] **[N]**
Sagen Sie Ja, um die Unterstützung für die Verschlüsselung des Systemspeichers zu aktivieren. Dies erfordert einen AMD-Prozessor, der Secure Memory Encryption (SME) unterstützt.

3.20.13.1 Activate AMD Secure Memory Encryption (SME) by default

CONFIG_AMD_MEM_ENCRYPT_ACTIVE_BY_DEFAULT [=n] **[N]**
Sagen Sie Ja, damit der Systemspeicher standardmäßig verschlüsselt wird, wenn er auf einem AMD-Prozessor läuft, der Secure Memory Encryption (SME) unterstützt. Wenn Sie Y wählen, kann die Verschlüsselung des Systemspeichers mit der Befehlszeilenoption `mem_encrypt=off` deaktiviert werden. Ist der Wert auf N gesetzt, kann die Verschlüsselung des Systemspeichers mit der Befehlszeilenoption `mem_encrypt=on` aktiviert werden.

3.20.14 NUMA Memory Allocation and Scheduler Support

CONFIG_NUMA [=y] **[Y]**
Aktivieren Sie die NUMA-Unterstützung (Non-Uniform Memory Access). Der Kernel wird versuchen, den von einer CPU verwendeten Speicher dem lokalen Speicher-Controller der CPU zuzuweisen und dem Kernel mehr Kenntnis über NUMA zu geben.
Für 64-Bit wird dies empfohlen, wenn das System Intel Core i7 (oder höher), AMD Opteron oder EM64T NUMA ist.
Für 32-Bit ist dies nur erforderlich, wenn Sie einen 32-Bit-Kernel auf einer 64-Bit-NUMA-Plattform booten. Andernfalls sollten Sie N angeben.

3.20.14.1 Old style AMD Opteron NUMA detection

CONFIG_AMD_NUMA [=y] **[N]**
Aktivieren Sie die Erkennung der AMD NUMA-Knoten-Topologie. Wenn Sie ein AMD-Multiprozessorsystem haben, sollten Sie hier Y angeben. Dies verwendet eine alte Methode, um die NUMA-Konfiguration direkt von der eingebauten Northbridge des Opteron zu lesen.
Es wird empfohlen, stattdessen X86_64_ACPI_NUMA zu verwenden, das auch Priorität hat, wenn beide einkompiliert sind.

3.20.14.2 ACPI NUMA detection

CONFIG_X86_64_ACOU_NUMA [=y] **[Y]**
Aktivieren Sie die auf ACPI SRAT basierende Knoten-Topologie-Erkennung.

3.20.14.3 NUMA emulation

CONFIG_NUMA_EMU [=n] **[N]**
Aktivieren Sie die NUMA-Emulation. Eine flache Maschine wird in virtuelle Knoten aufgeteilt, wenn sie mit `numa=fake=N` gebootet wird, wobei N die Anzahl der Knoten ist. Dies ist nur für die Fehlersuche nützlich.

3.20.14.4 Maximum NUMA Nodes (as a power of 2)

CONFIG_NODES_SHIFT [=5] **[5]**
(Maximale NUMA-Knoten (als eine Potenz von 2))
Geben Sie die maximale Anzahl der auf dem Zielsystem verfügbaren NUMA-Knoten an. Erhöht den reservierten Speicherplatz für verschiedene Tabellen.

3.20.15 Enable sysfs memory/probe interface

CONFIG_ARCH_MEMORY_PROBE [=n] **[N]**
Diese Option aktiviert eine sysfs-Speicher/Probe-Schnittstelle für Tests.
Siehe Documentation/admin-guide/mm/memory-hotplug.rst für weitere Informationen. Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollen, antworten Sie mit N.

3.20.16 Support non-standard NVDIMMs and ADR protected memory

CONFIG_X86_PMEM_LEGACY [=m] **[M]**
Behandeln Sie Speicher, der mit dem nicht standardmäßigen e820-Typ von 12 markiert ist, wie er vom Intel Sandy Bridge-EP Referenz-BIOS verwendet wird, als geschützten Speicher. Der Kernel bietet diese Regionen dem `pmem`-Treiber an, so dass sie für persistenten Speicher verwendet werden können.
Sagen Sie Y, wenn Sie unsicher sind.

3.20.17 Check for low memory corruption

CONFIG_X86_CHECK_BIOS_CORRUPTION [=y] **[Y]**
Regelmäßige Überprüfung auf Speicherbeschädigung im niedrigen Speicher, die vermutlich durch das BIOS verursacht wird. Auch wenn dies in der Konfiguration aktiviert ist, zur Laufzeit ist es deaktiviert. Aktivieren Sie es, indem Sie `memory_corruption_check=1` in der Kernel-Befehlszeile eingeben. Standardmäßig werden die unteren 64 k des Speichers alle 60 Sekunden überprüft; siehe die Parameter `memory_corruption_check_size` und `memory_corruption_check_period` in Documentation/admin-guide/kernel-parameters.rst, um dies anzupassen. Wenn diese Option mit den Standardparametern aktiviert ist, hat sie so gut wie keinen Overhead, da sie eine relativ kleine Menge an Speicher reserviert und diesen nur selten durchsucht. Sie erkennt Korruption und verhindert, dass sie das laufende System beeinträchtigt. Sie ist jedoch als Diagnosewerkzeug gedacht; wenn eine wiederholte BIOS-verursachte Beschädigung stets denselben Speicher betrifft, können Sie `memmap=` verwenden, um zu verhindern, dass der Kernel diesen Speicher verwendet.

Hinweis: Kann ausgeschaltet werden, wenn im `journalctl` niemals „corrupted low memory“ erscheint.

3.20.17.1 Set the default setting of memory_corruption_check

CONFIG_X86_BOOTPARAM_MEMORY_CORRUPTION_CHECK [=y] **[Y]**
Legt fest, ob der Standardstatus von `memory_corruption_check` ein- oder ausgeschaltet ist.

3.20.18 MTRR (Memory Type Range Register) support

CONFIG_MTRR [=y] **[Y]**
Bei Prozessoren der Intel P6-Familie (Pentium Pro, Pentium II und später) können die Memory Type Range Register (MTRRs) verwendet werden, um den Zugriff des Prozessors auf Speicherbereiche zu steuern. Dies ist besonders nützlich, wenn Sie eine Videokarte (VGA) an einem PCI- oder AGP-Bus haben. Durch die Aktivierung von Write-Combining können Bus-Schreibübertragungen zu einer größeren Übertragung kombiniert werden, bevor sie über den PCI/AGP-Bus geleitet werden. Dies kann die Leistung von Bildschreiboperationen um das 2,5-fache oder mehr erhöhen. Wenn Sie hier Y angeben, wird eine /proc/mtrr-Datei erstellt, die zur Manipulation der MTRRs Ihres Prozessors verwendet werden kann. Normalerweise sollte der X-Server dies verwenden.
Dieser Code hat eine recht generische Schnittstelle, so dass ähnliche Steuerregister auf anderen Prozessoren ebenfalls leicht unterstützt werden können:
Die Prozessoren Cyrix 6x86, 6x86MX und M II verfügen über Address Range Registers (ARRs), die eine ähnliche Funktionalität wie MTRRs bieten. In diesen Fällen werden die ARRs zur Emulation der MTRRs verwendet. Die AMD-Prozessoren K6-2 (Stepping 8 und höher) und K6-3 haben zwei MTRRs. Der Centaur C6 (WinChip) hat 8 MCRs, die Schreibkombinationen ermöglichen. Alle diese Prozessoren werden von diesem Code unterstützt und es ist sinnvoll, hier Y anzugeben, wenn Sie einen dieser Prozessoren haben.
Die Angabe von Y an dieser Stelle behebt auch ein Problem mit fehlerhaften SMP-BIOSen, die die MTRRs nur für die Boot-CPU und nicht für die sekundären CPUs setzen. Das kann zu allen möglichen Problemen führen, also ist es gut, hier Y zu sagen.
Sie können sicher Y sagen, auch wenn Ihr Rechner keine MTRRs hat, Sie werden nur etwa 9 KB zu Ihrem Kernel hinzufügen. Siehe <file:Documentation/arch/x86/mtrr.rst> für weitere Informationen.

3.20.18.1 MTRR cleanup support

CONFIG_MTRR_SANITIZER [=y] **[Y]**
Umwandlung des MTRR-Layouts von kontinuierlich in diskret, damit X-Treiber Rückschreibeinträge hinzufügen können. Kann mit `disable_mtrr_cleanup` in der Kernel-Kommandozeile deaktiviert werden. Die größte MTRR-Eintragsgröße für einen kontinuierlichen Block kann mit `mtrr_chunk_size` festgelegt werden.
Wenn Sie unsicher sind, sagen Sie Y.

3.20.18.2 MTRR cleanup enable value (0-1)

CONFIG_MTRR_SANITIZER [=1] **[1]**
Aktivieren Sie den „mtrr cleanup“-Standardwert

3.20.18.3 MTRR cleanup spare reg num (0-7)

CONFIG_MTRR_SANITIZER_SPARE_REG_NR_DEFAULT [=0] **[0]**
MTRR cleanup spare entries Defaulteintrag, dies kann über `mtrr_spare_reg_nr=N` auf der Kernel-Befehlszeile geändert werden.

3.20.19 Indirect Branch Tracking

CONFIG_X86_KERNEL_IBT [=y] **[Y]**
Bauen Sie den Kernel mit Unterstützung für Indirect Branch Tracking auf, eine Hardware-Unterstützung, die die Integrität des Kontrollflusses an den Rändern schützt. Sie erzwingt, dass alle indirekten Aufrufe auf einer ENDBR-Anweisung landen müssen, und der Compiler wird den Code mit ihnen instrumentieren, damit dies geschieht.
Zusätzlich zur Erstellung des Kernels mit IBT werden alle Funktionen, die keine indirekten Aufrufziele sind, versiegelt, um zu verhindern, dass sie jemals zu solchen werden.
Dies erfordert LTO wie objtool-Läufe und verlangsamt den Bau. Es reduziert jedoch die Anzahl der ENDBR-Anweisungen im Kernel-Image erheblich.

3.20.20 Memory Protection Keys

CONFIG_X86_INTEL_MEMORY_PROTECTION_KEYS [=y] **[Y]**
Memory Protection Keys bietet einen Mechanismus zur Erzwingung seitenbasierter Schutzmaßnahmen, ohne dass die Seitentabellen geändert werden müssen, wenn eine Anwendung ihre Schutzdomänen ändert.
Einzelheiten siehe Documentation/core-api/protection-keys.rst
Wenn Sie unsicher sind, sagen Sie Y.

3.20.21 TSX enable mode () →

CONFIG_X86_INTEL_MEMORY_PROTECTION_KEYS [=y] **[Y]**
Intels TSX-Funktion (Transactional Synchronization Extensions) ermöglicht die Optimierung von Sperrprotokollen durch Lock Elision, was zu einer spürbaren Leistungssteigerung führen kann. Andererseits hat sich gezeigt, dass TSX für Seitenkanalangriffe (z. B. TAA) ausgenutzt werden kann, und es ist wahrscheinlich, dass in Zukunft weitere Angriffe dieser Art entdeckt werden. Daher ist TSX standardmäßig nicht aktiviert (aka `tsx=off`). Ein Administrator kann diese Entscheidung durch den Befehlszeilenparameter `tsx=on` außer Kraft setzen. Auch wenn TSX aktiviert ist, versucht der Kernel, die bestmögliche TAA-Abschwächung zu aktivieren, je nach dem für den jeweiligen Rechner verfügbaren Mikrocode. Mit dieser Option kann der Standard-Tsx-Modus zwischen `tsx=on`, `=off` und `=auto` eingestellt werden. Siehe Documentation/admin-guide/kernel-parameters.txt für weitere Details. Sagen Sie off, wenn Sie sich nicht sicher sind, auto, wenn TSX in Gebrauch ist, aber auf sicheren Plattformen verwendet werden sollte, oder on, wenn TSX in Gebrauch ist und der Sicherheitsaspekt von tsx nicht relevant ist.

3.20.21.1 off

CONFIG_X86_INTEL_TSX_MODE_OFF [=n] **[N]**
TSX ist, wenn möglich, deaktiviert – entspricht dem Befehlszeilenparameter `tsx=off`.

3.20.21.2 on

CONFIG_X86_INTEL_TSX_MODE_ON [=n] **[N]**
TSX ist auf TSX-fähiger Hardware immer aktiviert – gleichbedeutend mit dem Befehlszeilenparameter `tsx=on`

3.20.21.3 auto

CONFIG_X86_INTEL_TSX_MODE_AUTO [=y] **[Y]**
TSX wird auf TSX-fähiger Hardware aktiviert, die als sicher gegen Seitenkanalangriffe gilt – gleichbedeutend mit dem Befehlszeilenparameter `tsx=auto`.

3.20.22 Software Guard eXtensions (SGX)

CONFIG_X86_SGX [=y] **[Y]**
Intel(R) Software Guard eXtensions (SGX) ist eine Reihe von CPU-Befehlen, die von Anwendungen verwendet werden können, um private Code- und Datenbereiche, die so genannten Enklaven, zu reservieren.

Auf den privaten Speicher einer Enklave kann nur von Code zugegriffen werden, der innerhalb der Enklave läuft. Zugriffe von außerhalb der Enklave, einschließlich anderer Enklaven, werden von der Hardware nicht zugelassen.
Wenn Sie unsicher sind, sagen Sie N.

3.20.23 X86 userspace shadow stack

CONFIG_X86_USER_SHADOW_STACK [=y] **[Y]**
Der Schattenstapelschutz ist eine Hardwarefunktion, die eine Beschädigung der Rücksprungadresse einer Funktion erkennt. Dies hilft, ROP-Angriffe abzuschwächen. Anwendungen müssen aktiviert sein, um sie zu nutzen, und der alte Userspace erhält den Schutz nicht ümsonst". CPUs, die Shadow Stacks unterstützen, wurden erstmals im Jahr 2020 vorgestellt. Weitere Informationen finden Sie unter Documentation/arch/x86/shstk.rst.
Wenn Sie unsicher sind, sagen Sie N.

3.20.24 EFI runtime service support

CONFIG_EFI [=y] **[Y]**
Dies ermöglicht es dem Kernel, verfügbare EFI-Laufzeitdienste (wie die EFI-Variablendienste) zu nutzen. Diese Option ist nur auf Systemen mit EFI-Firmware sinnvoll. Außerdem sollten Sie den neuesten ELILO-Lader verwenden, der unter `http://elilo.sourceforge.net` verfügbar ist, um die Vorteile der EFI-Laufzeitdienste zu nutzen. Aber auch mit dieser Option sollte der resultierende Kernel weiterhin auf bestehenden Nicht-EFI-Plattformen booten.

3.20.24.1 EFI stub support

CONFIG_EFI_STUB [=y] **[Y]**
Mit dieser Kernel-Funktion kann ein bzImage direkt von der EFI-Firmware geladen werden, ohne dass ein Bootloader erforderlich ist.
Weitere Informationen finden Sie unter Documentation/admin-guide/efi-stub.rst.

3.20.24.1.1 EFI handover protocol (DEPRECATED)

CONFIG_EFI_STUB [=y] **[Y]**
(EFI-Übergabeprotokoll (VERALTET))
Wählen Sie dies, um Unterstützung für das veraltete EFI-Handover-Protokoll zu erhalten, das alternative Einstiegspunkte in den EFI-Stub definiert. Dies ist eine Praxis, die keine Grundlage in der UEFI-Spezifikation hat und ein Vorwissen seitens des Bootloaders über Linux/x86-spezifische Wege der Übergabe der Kommandozeile und initrd erfordert, und wo im Speicher diese Assets geladen werden können.
Im Zweifelsfall sagen Sie Y. Auch wenn die entsprechende Unterstützung im Upstream-GRUB oder anderen Bootloadern nicht vorhanden ist, bauen die meisten Distros GRUB mit zahlreichen Downstream-Patches und können sich daher auf das Handover-Protokoll verlassen.

3.20.24.1.2 EFI mixed-mode support

CONFIG_EFI_MIXED [=y] **[Y]**
Wenn Sie diese Funktion aktivieren, kann ein 64-Bit-Kernel auf einer 32-Bit-Firmware gebootet werden, vorausgesetzt, Ihre CPU unterstützt den 64-Bit-Modus.
Beachten Sie, dass es nicht möglich ist, einen Mixed-Mode-fähigen Kernel über den EFI-Boot-Stub zu booten – es muss ein Bootloader verwendet werden, der das EFI-Handover-Protokoll unterstützt.
Wenn Sie unsicher sind, sagen Sie N.

3.20.24.2 Enable EFI fake memory map

CONFIG_EFI_FAKE_MEMMAP [=n] **[N]**
Wenn Sie hier Y angeben, wird die Boot-Option `efi_fake_mem` aktiviert. Durch Angabe dieses Parameters können Sie einem bestimmten Speicherbereich beliebige Attribute hinzufügen, indem Sie die ursprüngliche (von der Firmware bereitgestellte) EFI-Memmap aktualisieren. Dies ist nützlich für das Debugging von EFI-Memmap-bezogenen Funktionen, z. B. Address Range Mirroring.

3.20.25 Timer frequency () →

Ermöglicht die Konfiguration der Timer-Frequenz. Es ist üblich, den Timer-Interrupt mit 1000 Hz laufen zu lassen, aber 100 Hz kann für Server und NUMA-Systeme vorteilhafter sein, die keine schnelle Reaktion für die Benutzerinteraktion benötigen und bei denen es zu Buskonflikten und Cacheline-Bounches als Folge von Timer-Interrupts kommen kann. Beachten Sie, dass der Timer-Interrupt in einer SMP-Umgebung auf jedem Prozessor auftritt, was zu NR_CPUS * HZ Anzahl der Timer-Interrupts pro Sekunde führt.

3.20.25.1 100 Hz

CONFIG_HZ_100 [=n] **[N]**
100 Hz ist eine typische Wahl für Server, SMP- und NUMA-Systeme mit vielen Prozessoren, die eine geringere Leistung aufweisen können, wenn zu viele Timer-Interrupts auftreten.

3.20.25.2 250 Hz

CONFIG_HZ_250 [=n] **[N]**
250 Hz ist ein guter Kompromiss, der eine gute Serverleistung ermöglicht und auch auf SMP- und NUMA-Systemen eine gute interaktive Reaktionsfähigkeit zeigt. Wenn Sie NTSC-Video oder Multimedia verwenden, wählen Sie stattdessen 300 Hz.

3.20.25.3 300 Hz

CONFIG_HZ_300 [=y] **[Y]**
300 Hz ist ein guter Kompromiss, der eine gute Serverleistung und gleichzeitig eine gute interaktive Reaktionsfähigkeit selbst auf SMP- und NUMA-Systemen ermöglicht und sowohl bei PAL- als auch bei NTSC-Bildraten für Video- und Multimedia-Arbeiten genau eingehalten wird.

3.20.25.4 1000 Hz

CONFIG_HZ_1000 [=n] **[N]**
1000 Hz ist die bevorzugte Wahl für Desktop-Systeme und andere Systeme, die schnelle interaktive Reaktionen auf Ereignisse erfordern.

3.20.26 Physical address where the kernel is loaded

CONFIG_PHYSICAL_START [=0x1000000] **[0x1000000]**
Dies gibt die physikalische Adresse an, unter der der Kernel geladen wird. Wenn der Kernel nicht verschiebbar ist (CONFIG_RELOCATABLE=n), dekomprimiert sich bzImage an die oben genannte physikalische Adresse und wird von dort aus gestartet. Andernfalls wird bzImage von der Adresse aus gestartet, an der es vom Bootloader geladen wurde, und ignoriert die obige physikalische Adresse. In normalen kdump-Fällen muss diese Option nicht gesetzt/geändert werden, da bzImage nun als vollständig relozierbares Image (CONFIG_RELOCATABLE=y) kompiliert und zum Laden und Ausführen von einer anderen Adresse verwendet werden kann. Diese Option ist vor allem für die Leute nützlich, die kein bzImage für die Erfassung des Crash-Dumps verwenden wollen und stattdessen vmlinux einsetzen wollen. vmlinux ist nicht relocatable, daher muss ein Kernel speziell kompiliert werden, um von einem bestimmten Speicherbereich (normalerweise ein reservierter Bereich) zu laufen, und diese Option ist sehr nützlich. Wenn Sie also bzImage zum Erfassen des Crash-Dumps verwenden, lassen Sie den Wert hier unverändert auf 0x1000000 und setzen Sie CONFIG_RELOCATABLE=y.
Andernfalls, wenn Sie vmlinux für die Aufzeichnung des Crash-Dumps verwenden wollen, ändern Sie diesen Wert auf den Beginn des reservierten Bereichs. Mit anderen Worten, er kann auf der Grundlage des "XWertes gesetzt werden, wie er im "crashkernel=YM@XMBefehlszeilen-Boot-Parameter angegeben ist, der an den panic-ed-Kernel übergeben wird. Weitere Details zu Crash Dumps finden Sie in Documentation/admin-guide/kdump/kdump.rst. Die Verwendung von bzImage für die Aufzeichnung des Crash-Dumps wird empfohlen, da man nicht zwei Kernel erstellen muss. Derselbe Kernel kann als Produktionskernel und als Erfassungskernel verwendet werden. Die obige Option sollte verschwinden, nachdem die Unterstützung von relocatable bzImage eingeführt wurde. Sie ist aber noch vorhanden, weil es Benutzer gibt, die weiterhin vmlinux für die Dump-Erfassung verwenden. Diese Option sollte im Laufe der Zeit verschwinden. Ändern Sie diese Option nicht, wenn Sie nicht wissen, was Sie tun.

3.20.27 Build a relocatable kernel

CONFIG_RELOCATABLE [=y] **[Y]**
Dadurch wird ein Kernel-Image erstellt, das die Informationen über den Standortwechsel beibehält, so dass es an einem anderen Ort als den standardmäßigen 1 MB geladen werden kann. Die Verschiebungen machen das Kernel-Binary etwa 10 % größer, werden aber zur Laufzeit verworfen. Eine Anwendung ist der kexec on panic-Fall, bei dem der Wiederherstellungs-Kernel an einer anderen physikalischen Adresse liegen muss als der primäre Kernel. Anmerkung: Wenn CONFIG_RELOCATABLE=y ist, dann läuft der Kernel von der Adresse aus, an der er geladen wurde, und von der zur Kompilierzeit physische Adresse (CONFIG_PHYSICAL_START) als Mindeststandort verwendet.

3.20.27.1 Randomize the address of the kernel image (KASLR)

CONFIG_RANDOMIZE_BASE [=y] **[Y]**
Zur Unterstützung der Kernel Address Space Layout Randomization (KASLR) werden die physische Adresse, an der das Kernel-Image dekomprimiert wird, und die virtuelle Adresse, auf die das Kernel-Image abgebildet wird, randomisiert. Dies ist ein Sicherheitsmerkmal, das Exploit-Versuche verhindert, die auf der Kenntnis des Speicherorts von Kernel-Code-Interna beruhen. Bei 64-Bit werden die physische und die virtuelle Adresse des Kernels getrennt randomisiert. Die physische Adresse liegt irgendwo zwischen 16 MB und dem Anfang des physischen Speichers (bis zu 64 TB). Die virtuelle Adresse wird von 16 MB bis zu 1 GB randomisiert (9 Bits Entropie). Beachten Sie, dass dadurch auch der für Kernel-Module verfügbare Speicherplatz von 1,5 GB auf 1 GB reduziert wird. Bei 32-Bit werden die physischen und virtuellen Adressen des Kernels zusammen randomisiert. Sie werden von 16 MB bis zu 512 MB randomisiert (8 Bits Entropie).
Die Entropie wird mit dem RDRAND-Befehl erzeugt, sofern er unterstützt wird. Wenn RDTSC unterstützt wird, wird sein Wert ebenfalls in den Entropie-Pool gemischt. Wenn weder RDRAND noch RDTSC unterstützt werden, wird die Entropie aus dem i8254-Zeitgeber gelesen. Die nutzbare Entropie ist dadurch begrenzt, dass der Kernel mit 2 GB-Adressierung aufgebaut ist und dass PHYSICAL_ALIGN mindestens 2 MB betragen muss. Infolgedessen sind theoretisch nur 10 Bits Entropie möglich, aber die Implementierungen sind aufgrund des Speicherlayouts noch weiter eingeschränkt.
Wenn Sie unsicher sind, sagen Sie Y.

3.20.28 Alignment value to which kernel should be aligned

CONFIG_PHYSICAL_ALIGN [=0x200000] **[0x200000]**
Dieser Wert legt die Ausrichtungsbeschränkungen für die physikalische Adresse fest, von der der Kernel geladen und ausgeführt wird. Der Kernel wird für eine Adresse kompiliert, die den obigen Ausrichtungsbeschränkungen entspricht. Wenn der Bootloader den Kernel an einer nicht ausgerichteten Adresse lädt und CONFIG_RELOCATABLE gesetzt ist, verschiebt sich der Kernel an die nächstgelegene Adresse, die auf den obigen Wert ausgerichtet ist, und wird von dort aus gestartet. Wenn der Bootloader den Kernel an einer nicht ausgerichteten Adresse lädt und CONFIG_RELOCATABLE nicht gesetzt ist, ignoriert der Kernel die Ladeadresse zur Laufzeit und dekomprimiert sich an die Adresse, für die er kompiliert wurde, und läuft von dort aus. Die Adresse, für die der Kernel kompiliert wurde, erfüllt bereits die oben genannten Ausrichtungsbeschränkungen. Das Endergebnis ist also, dass der Kernel von einer physikalischen Adresse aus läuft, die die oben genannten Ausrichtungsbeschränkungen erfüllt. Bei 32-Bit muss dieser Wert ein Vielfaches von 0x2000 sein. Bei 64-Bit muss dieser Wert ein Vielfaches von 0x200000 sein. Ändern Sie dies nicht, wenn Sie nicht wissen, was Sie tun.

3.20.29 Randomize the kernel memory sections

CONFIG_RANDOMIZE_MEMORY [=y] **[Y]**
Randomisiert die virtuelle Basisadresse von Kernel-Speicherabschnitten (physische Speicherzuordnung, vmalloc & vmemmap). Dieses Sicherheitsmerkmal macht Exploits, die sich auf vorhersehbare Speicherplätze verlassen, weniger zuverlässig. Die Reihenfolge der Zuweisungen bleibt unverändert. Entropie wird auf die gleiche Weise wie bei RANDOMIZE_BASE erzeugt. Aktuelle Implementierung in der optimalen Konfiguration haben im Durchschnitt 30.000 verschiedene mögliche virtuelle Adressen für jeden Speicherabschnitt.
Wenn Sie unsicher sind, sagen Sie Y.

3.20.30 Linear Address Masking support

CONFIG_ADDRESS_MASKING [=y] **[Y]**
Linear Address Masking (LAM) ändert die Prüfung, die auf lineare 64-Bit-Adressen angewandt wird, und ermöglicht der Software die nicht übersetzten Adressbits für Metadaten zu verwenden.
Diese Fähigkeit kann für die effiziente Implementierung von Adress-Sanitizern (ASAN) und für Optimierungen in JITs genutzt werden.

3.20.31 Disable the 32-bit vDSO (needed for glibc 2.3.3)

CONFIG_COMPAT_VDSO [=n] **[N]**
Bestimmte fehlerhafte Versionen der glibc stürzen ab, wenn sie mit einem 32-Bit vDSO konfrontiert werden, das nicht auf die in der Segmenttabelle angegebene Adresse abgebildet ist. Adresse zugeordnet ist, die in der Segmenttabelle angegeben ist.
Der Fehler wurde eingeführt durch f866314b89d56845f55e6f365e18b31ec978ec3a und behoben durch 3b3ddb4f7db98ec9e912ccdf54d35df4aa30e04a und 49ad572a70b8aeb91e57483a11dd1b77e31c4468.
Glibc 2.3.3 ist die einzige veröffentlichte Version mit dem Fehler, aber OpenSUSE 9 enthält eine fehlerhafte „glibc 2.3.2“. Das Symptom des Fehlers ist, dass alles beim Start abstürzt und sagt:
`dl_main: Assertion ` (void) ph->p_vaddr == _rtld_local._dl_sysinfo_dso`
Wenn Sie hier Y sagen, wird der Standardwert der Bootoption vdso32 von 1 auf 0 geändert, wodurch die 32-Bit vDSO vollständig deaktiviert wird. Dies umgeht zwar den Glibc-Bug, beeinträchtigt aber die Leistung.
Wenn Sie unsicher sind, sagen Sie N: Wenn Sie Ihren eigenen Kernel kompilieren, ist es unwahrscheinlich, dass Sie eine fehlerhafte Version der glibc verwenden.

3.20.32 vsyscall table for legacy applications () →

Legacy-Benutzercode, der nicht weiß, wie er den vDSO finden kann, erwartet, dass er drei Syscalls ausgeben kann, indem er feste Adressen im Kernel-Bereich aufruft. Da dieser Ort nicht mit ASLR randomisiert wird, kann er dazu verwendet werden, die Ausnutzung von Sicherheitslücken zu unterstützen. Diese Einstellung kann zur Boot-Zeit über den Kernel-Befehlszeilenparameter `vsyscall=[emulate|xonly|none]` geändert werden.
Der Emulationsmodus ist veraltet und kann nur noch über die Kernel-Befehlszeile aktiviert werden. Auf einem System mit ausreichend aktueller glibc (2.14 oder neuer) und ohne statische Binärdateien können Sie „None“ ohne Leistungseinbußen verwenden um die Sicherheit zu verbessern.
Wenn Sie unsicher sind, wählen Sie „Nur Ausführung emulieren„.

3.20.32.1 Emulate execution only

CONFIG_LEGACY_VSYSCALL_XONLY [=y] **[Y]**
Der Kernel fängt und emuliert Aufrufe in die feste vsyscall-Adresszuordnung und lässt keine Lesezugriffe zu. Diese Konfiguration wird empfohlen, wenn der Userspace den Legacy-Vsyscall-Bereich verwenden könnte, aber keine Unterstützung für die binäre Instrumentierung von Legacy-Code benötigt wird. Sie entschärft bestimmte Verwendungen des vsyscall-Bereichs als Puffer zur Umgehung von ASLR.

3.20.32.2 None

CONFIG_LEGACY_VSYSCALL_NONE [=n] **[N]**
Es wird überhaupt keine vsyscall-Zuordnung geben. Dies eliminiert jegliches Risiko einer ASLR-Umgehung aufgrund der festen vsyscall-Adressen-Zuordnung. Versuche, die vsyscalls zu verwenden, werden an dmesg gemeldet, so dass entweder alte oder bösartige Userspace-Programme identifiziert werden können.

3.20.33 Built-in kernel command line

CONFIG_CMDLINE_BOOL [=n] **[N]**
Ermöglicht die Angabe von Boot-Argumenten für den Kernel zur Erstellungszeit. Auf einigen Systemen (z. B. eingebetteten [embedded]) ist es notwendig oder praktisch, einige oder alle Kernel-Boot-Argumente mit dem Kernel selbst bereitzustellen (d.h. sich nicht darauf zu verlassen, dass der Bootloader sie bereitstellt). Um Kommandozeilenargumente in den Kernel zu kompilieren, setzen Sie diese Option auf Y und geben Sie dann die Boot-Argumente in CONFIG_CMDLINE ein. Bei Systemen mit voll funktionsfähigen Bootloadern (d.h. nicht eingebetteten) sollte diese Option auf N gesetzt bleiben.

3.20.34 Enforce strict size checking for sigaltstack

CONFIG_STRICT_SIGALTSTACK_SIZE [=n] **[N]**
Aus historischen Gründen ist MINSIGSTKSZ eine Konstante, die mit der AVX512-Unterstützung bereits zu klein wurde. Fügen Sie einen Mechanismus hinzu, um die strenge Überprüfung der Sigaltstack-Größe gegen die tatsächliche Größe des FPU-Rahmens zu erzwingen. Diese Option aktiviert die Überprüfung standardmäßig. Sie kann auch über die Kernel-Kommandozeilenoption `strict_sas_size` unabhängig von diesem Konfigurationsschalter gesteuert werden. Das Aktivieren dieser Option könnte bestehende Anwendungen zerstören, die einen zu kleinen Sigaltstack zuweisen, aber ‚funktionieren', weil sie nie ein Signal geliefert bekommen.
Sagen Sie N, wenn Sie diese Prüfung nicht wirklich erzwingen wollen.

3.20.35 Kernel Live Patching

CONFIG_LIVEPATCH [=n] **[N]**
Geben Sie hier Y an, wenn Sie Kernel-Live-Patching unterstützen wollen. Diese Option hat keine Auswirkungen auf die Laufzeit, bis ein Kernel-„Patch"-Modul die von dieser Option bereitgestellte Schnittstelle verwendet, um einen Patch zu registrieren, was dazu führt, dass Aufrufe der gepatchten Funktionen auf den neuen Funktionscode im Patch-Modul umgeleitet werden.

4 Mitigations for speculative execution vulnerabilities →

CONFIG_SPECULATION_MITIGATIONS [=y] **[Y]**
Sagen Sie hier Y, um Optionen zu aktivieren, die Abhilfemaßnahmen für Hardware-Schwachstellen durch spekulative Ausführung ermöglichen. Wenn Sie N sagen, werden alle Abhilfemaßnahmen deaktiviert. Sie sollten wirklich wissen, was Sie tun, um dies anzugeben.

4.1 Remove the kernel mapping in user mode

CONFIG_PAGE_TABLE_ISOLATION [=y] **[Y]**
Diese Funktion reduziert die Anzahl der Hardware-Seitenkanäle, indem sie sicherstellt, dass die meisten Kernel-Adressen nicht in den Benutzerraum abgebildet werden. Siehe Documentation/arch/x86/pti.rst für weitere Details.

4.2 Avoid speculative indirect branches in kernel

CONFIG_RETPOLINE [=y] **[Y]**
Kompilieren Sie den Kernel mit den retpoline Compiler-Optionen, um Datenlecks zwischen Kernel und Benutzer zu verhindern, indem spekulative indirekte Verzweigungen vermieden werden. Erfordert einen Compiler mit `-mindirect-branch=thunk-extern` Unterstützung für vollen Schutz. Der Kernel kann langsamer laufen.

4.2.1 Enable return-thunks

CONFIG_RETHUNK [=y] **[Y]**
Kompiliere den Kernel mit der Compileroption return-thunks, um Datenlecks zwischen Kernel und Benutzer zu verhindern, indem Rückgabespekulationen vermieden werden. Erfordert einen Compiler mit `-mfunction-return=thunk-extern` Unterstützung für vollen Schutz. Der Kernel kann langsamer laufen.

4.2.1.1 Enable UNRET on kernel entry

CONFIG_CPU_UNRET_ENTRY [=y] **[Y]**
Kompiliere den Kernel mit Unterstützung für die `retbleed=unret`-Abschwächung.

4.3 Mitigate RSB underflow with call depth tracking

CONFIG_CALL_DEPTH_TRACKING [=y] **[Y]**
Kompiliere den Kernel mit Call-Depth-Tracking, um das Intel SKL Return-Speculation-Buffer (RSB) Underflow-Problem zu entschärfen. Die Entschärfung ist standardmäßig ausgeschaltet und muss in der Kernel-Befehlszeile über die Option `retbleed=stuff` aktiviert werden. Für nicht betroffene Systeme ist

der Overhead dieser Option marginal, da die Verfolgung der Aufruftiefe zur Laufzeit generierte Call Thunks in einem vom Compiler generierten Padding-Bereich und Call Patching verwendet. Dies erhöht die Textgröße um $\sim$ 5%. Bei nicht betroffenen Systemen ist dieser Platz ungenutzt. Auf betroffenen SKL-Systemen führt dies zu einem erheblichen Leistungsgewinn gegenüber der IBRS-Abschwächung.

4.3.1 Enable call thunks and call depth tracking debugging

CONFIG_CALL_THUNKS_DEBUG [=n] **[N]**
Aktiviere Call/Ret-Zähler zur Erkennung von Ungleichgewichten und baue ein lautes dmesg über die Erzeugung von Callthunks und Call-Patching zur Fehlersuche ein. Die Debug-Ausdrucke müssen in der Kernel-Befehlszeile mit `debug-callthunks` aktiviert werden. Aktivieren Sie dies nur, wenn Sie Call Thunks debuggen wollen, da dies einen spürbaren Laufzeit-Overhead erzeugt. Wenn Sie unsicher sind, sagen Sie N.

4.4 Enable IBPB on kernel entry

CONFIG_CPU_IBPB_ENTRY [=y] **[Y]**
Kompiliere den Kernel mit Unterstützung für die `retbleed=ibpb`-Abschwächung.

4.5 Enable IBRS on kernel entry

CONFIG_CPU_IBRS_ENTRY [=y] **[Y]**
Kompiliere den Kernel mit Unterstützung für die `spectre_v2=ibrs`-Abschwächung. Dadurch werden sowohl spectre_v2 als auch retbleed auf Kosten der Leistung abgeschwächt.

4.6 Mitigate speculative RAS overflow on AMD

CONFIG_CPU_SRSO [=y] **[N]**
Aktiviert die SRSO-Abschwächung, die auf AMD Zen1-4-Maschinen benötigt wird.

4.7 Mitigate Straight-Line-Speculation

CONFIG_SLS [=y] **[Y]**
Kompiliere den Kernel mit Straight-Line-Speculation-Optionen, um ihn vor Straight-Line-Speculation zu schützen. Das Kernel-Image könnte etwas größer sein.

4.8 Force GDS Mitigation

CONFIG_GDS_FORCE_MITIGATION [=n] **[N]**
Gather Data Sampling (GDS) ist eine Hardware-Schwachstelle, die unberechtigten spekulativen Zugriff auf Daten ermöglicht, die zuvor in Vektorregistern gespeichert wurden. Diese Option ist gleichbedeutend mit der Einstellung `gather_data_sampling=force` in der Befehlszeile. Die Mikrocode-Abschwächung wird verwendet, falls vorhanden, andernfalls wird AVX als Abschwächung deaktiviert. Auf betroffenen Systemen, denen der Microcode fehlt, wird jeder Userspace-Code, der AVX bedingungslos verwendet, bei gesetzter Option abbrechen. Das Setzen dieser Option auf Systemen, die nicht für GDS anfällig sind, hat keine Auswirkungen.
Im Zweifelsfall sagen Sie N.

5 Power management and ACPI options →

Energieverwaltung und ACPI-Optionen

5.1 Suspend to RAM and standby

CONFIG_SUSPEND [=y] **[Y]**
Ermöglicht dem System, in Ruhezustände einzutreten, in denen der Hauptspeicher mit Strom versorgt wird und somit sein Inhalt erhalten bleibt, wie z. B. der Suspend-to-RAM-Zustand (z. B. der ACPI S3-Zustand).

5.2 Hibernation (aka 'suspend to disk')

CONFIG_HIBERNATION [=y] **[Y]**
Aktiviert die Funktion „Suspend to Disk“ (STD), die in den Benutzeroberflächen gewöhnlich als „Ruhezustand“ bezeichnet wird. STD setzt das System an einen Haltepunkt und schaltet es aus; beim Neustart wird dieser Haltepunkt wiederhergestellt. Sie können Ihren Rechner mit `echo disk > /sys/power/state` in den Ruhezustand versetzen, nachdem Sie `resume=/dev/swappartition` in der Kernel-Befehlszeile in der Konfigurationsdatei Ihres Bootloaders angegeben haben. Alternativ können Sie auch die zusätzlichen Userland-Tools verwenden, die unter `http://suspend.sf.net` verfügbar sind. Im Prinzip sind weder ACPI noch APM erforderlich, obwohl beispielsweise ACPI für die letzten Schritte verwendet wird, wenn es verfügbar ist. Einer der Gründe für die Verwendung von Software-Suspend ist, dass die Firmware-Hooks für Suspend-Zustände wie Suspend-to-RAM (STR) oft nicht sehr gut mit Linux funktionieren. Es wird ein Abbild erstellt, das in der aktiven Auslagerungsdatei gespeichert wird. Beim nächsten Start übergeben Sie dem Kernel das Argument `resume=/dev/swappartition`, damit er das gespeicherte Abbild erkennt, den Speicherstatus daraus wiederherstellt und wie zuvor weiterarbeitet. Wenn Sie nicht wollen, dass der vorherige Zustand wiederhergestellt wird, verwenden Sie das Kernel-Befehlszeilenargument `noresume`. Beachten Sie jedoch, dass fsck auf Ihren Dateisystemen ausgeführt wird und Sie mkswap auf der Swap-Partition ausführen müssen, die für den Suspend verwendet wird. In begrenztem Umfang funktioniert es auch mit Swap-Dateien (für Details siehe <file:Documentation/power/swsusp-and-swap-files.rst>). Sie können jetzt booten, ohne den Vorgang fortzusetzen, und ihn später fortsetzen, aber in der Zwischenzeit können Sie die Swap-Partition(en)/Datei(en), die am Suspendieren beteiligt waren, nicht verwenden. In diesem Fall dürfen Sie auch nicht die Dateisysteme verwenden, die vor dem Suspendieren gemountet waren. Insbesondere dürfen Sie keine journalisierten Dateisysteme mounten, die vor dem Suspending gemountet wurden, da diese sonst auf unschöne Weise beschädigt werden. Weitere Informationen finden Sie in <file:Documentation/power/swsusp.rst>.

5.2.1 Userspace snapshot device

CONFIG_HIBERNATION_SNAPSHOT_DEV [=y] **[Y]**
Gerät, das von den uswsusp-Werkzeugen verwendet wird. Sagen Sie N, wenn kein Snapshotting aus dem Userspace benötigt wird, dies reduziert auch die Angriffsfläche des Kernels. Im Zweifelsfall sagen Sie Y.

5.2.2 Default resume partition

CONFIG_PM_STD_PARTITION [=] [ ]
Die Standard-Wiederaufnahmepartition ist die Partition, auf der die Suspend-to-Disk-Implementierung nach einem Suspend-Disk-Image suchen wird. Die hier angegebene Partition wird für fast jeden Benutzer anders sein. Es sollte eine gültige Swap-Partition sein (zumindest im Moment), die vor dem Suspendieren eingeschaltet wird. Die angegebene Partition kann durch die Angabe von:
`resume=/dev/<anderes Gerät>`
überschrieben werden, wodurch die Partition für die Wiederaufnahme auf das angegebene Gerät gesetzt wird. Beachten Sie, dass es derzeit keine Möglichkeit gibt, das Gerät anzugeben, auf dem das suspendierte Image gespeichert werden soll. Es wird einfach das erste verfügbare Swap-Gerät ausgewählt.

5.3 Opportunistic sleep

CONFIG_PM_AUTOSLEEP [=n] **[N]**
Ermöglicht es dem Kernel, automatisch einen Systemübergang in einen globalen Ruhezustand auszulösen, wenn es keine aktiven Weckquellen gibt.

5.4 Userspace opportunistic sleep

CONFIG_PM_USERSPACE_AUTOSLEEP [=n] **[N]**
Benachrichtigt den Kernel über eine aggressive Benutzerraum-Energieverwaltungspolitik für den automatischen Schlaf. Diese Option ändert das Verhalten verschiedener schlafempfindlicher Codes, um mit häufigen, vom Benutzer initiierten Übergängen in einen globalen Schlafzustand umzugehen. Wenn Sie hier Y sagen, werden Codepfade deaktiviert, die die meisten Benutzer wirklich aktiviert lassen sollten. Aktivieren Sie dies nur, wenn es sehr häufig vorkommt, dass man für sehr kurze Zeiträume (<= 2 Sekunden) schläft/wach ist. Nur Plattformen, wie z. B. Android, die opportunistischen Ruhezustand von einem Userspace-Energieverwaltungsdienst implementieren, sollten diese Option aktivieren, nicht aber andere

Maschinen. Daher sollten Sie hier N sagen, es sei denn, Sie sind sich sehr sicher, dass Sie dies wollen. Die Option hat andernfalls schlechte, unerwünschte Auswirkungen und sollte nicht nur zum Spaß aktiviert werden.

5.5 User space wakeup sources interface

CONFIG_PM_WAKELOCKS [=n] **[N]**
Ermöglicht es dem Benutzer, Wakeup-Quellobjekte mit Hilfe einer sysfs-basierten Schnittstelle zu erstellen, zu aktivieren und zu deaktivieren.

5.6 Device power management core functionality

CONFIG_PM_WAKELOCKS [=y] **[Y]**
Aktivierung von Funktionen, die es ermöglichen, E/A-Geräte in einen energiesparenden (stromsparenden) Zustand zu versetzen, z. B. nach einer bestimmten Zeit der Inaktivität (autosuspended), und sie als Reaktion auf ein von der Hardware erzeugtes Wake-up-Ereignis oder eine Anforderung des Treibers aufzuwecken. Damit diese Funktion funktioniert, ist in der Regel eine Hardwareunterstützung erforderlich, und die Bustreiber der Busse, an denen die Geräte angeschlossen sind, sind für die tatsächliche Handhabung von Suspendierungsanforderungen und Weckereignissen zuständig.

5.6.1 Power Management Debug Support

CONFIG_PM_DEBUG [=y] **[Y]**
Diese Option aktiviert verschiedene Debugging-Funktionen im Power-Management-Code. Dies ist hilfreich bei der Fehlersuche und der Meldung von PM-Fehlern, wie z. B. der Suspend-Unterstützung.

5.6.1.1 Extra PM attributes in sysfs for low-level debugging/testing

CONFIG_PM_ADVANCED_DEBUG [=n] **[N]**
Hinzufügen zusätzlicher sysfs-Attribute, die den Zugriff auf einige Power-Management-Felder von Geräteobjekten aus dem Userspace ermöglichen. Wenn Sie kein Kernel-Entwickler sind, der am Debuggen/Testen von Power Management interessiert ist, sagen Sie N für nein.

5.6.1.2 Test suspend/resume and wakealarm during bootup

CONFIG_PM_TEST_SUSPEND [=n] **[N]**
Mit dieser Option können Sie Ihren Rechner während des Bootvorgangs in den Ruhezustand versetzen und ihn einige Sekunden später mit einem RTC-Weckalarm aufwecken. Aktivieren Sie dies mit einem Kernelparameter wie `test_suspend=mem`. Wahrscheinlich sollten Sie den RTC-Treiber Ihres Systems statisch einbinden, um sicherzustellen, dass er verfügbar ist, wenn dieser Test läuft.

5.7 Suspend/resume event tracing

CONFIG_PM_TRACE_RTC [=y] **[Y]**
Dies ermöglicht es, den letzten PM-Ereignispunkt in der RTC über Neustarts hinweg zu speichern, so dass Sie einen Rechner, der während des Suspendierens (oder häufiger während des Wiederaufnehmens) einfach hängen bleibt, debuggen können. Um diese Debugging-Funktion zu nutzen, sollten Sie versuchen, den Rechner in den Suspend-Modus zu versetzen, ihn neu zu starten und dann Folgendes auszuführen

```
dmesg -s 1000000 | grep 'hash matches'
```

ACHTUNG: Diese Option führt dazu, dass die Echtzeituhr Ihres Rechners nach einem Neustart auf eine ungültige Zeit gesetzt wird.

5.8 Enable workqueue power-efficient mode by default

CONFIG_WQ_POWER_EFFICIENT_DEFAULT [=y] **[Y]**
Pro-CPU-Workqueues werden im Allgemeinen bevorzugt, da sie dank der Cache-Lokalität eine bessere Leistung aufweisen; leider neigen Pro-CPU-Workqueues dazu, mehr Strom zu verbrauchen als ungebundene Workqueues. Durch die Aktivierung des Kernelparameters `workqueue.power_efficient` werden die Pro-CPU-Workqueues, die nachweislich erheblich zum Stromverbrauch beitragen, ungebunden, was zu einem messbar geringeren Stromverbrauch auf Kosten eines geringen Leistungsoverheads führt. Diese

Konfigurationsoption legt fest, ob `workqueue.power_efficient` standardmäßig aktiviert ist.
Im Zweifelsfall sagen Sie N.

5.9 Energy Model for devices with DVFS (CPUs, GPUs, etc)

CONFIG_ENERGY_MODEL [=y] **[Y]**
Mehrere Teilsysteme (z. B. das thermische System und/oder der Aufgabenplaner) können Informationen über den Energieverbrauch von Geräten nutzen, um intelligentere Entscheidungen zu treffen. Diese Konfigurationsoption aktiviert den Rahmen, von dem aus die Subsysteme auf die Energiemodelle zugreifen können. Die genaue Verwendung des Energiemodells ist subsystemabhängig.
Im Zweifelsfall sagen Sie N.

5.10 ACPI (Advanced Configuration and Power Interface) Support →

CONFIG_ACPI [=y] **[Y]**
Die Unterstützung von ACPI (Advanced Configuration and Power Interface) für Linux erfordert eine ACPI-kompatible Plattform (Hardware/Firmware) und setzt das Vorhandensein von OS-directed configuration and power management (OSPM) Software voraus. Mit dieser Option wird Ihr Kernel um etwa 70K erweitert. Linux ACPI bietet einen robusten funktionalen Ersatz für mehrere ältere Konfigurations- und Energieverwaltungsschnittstellen, einschließlich der Plug-and-Play-BIOS-Spezifikation (PnP BIOS), der MultiProzessor-Spezifikation (MPS) und der Advanced Power Management (APM)-Spezifikation. Wenn sowohl ACPI- als auch APM-Unterstützung konfiguriert sind, wird ACPI verwendet. Die Linux-Unterstützung für ACPI basiert auf der ACPI Component Architecture (ACPI CA) der Intel Corporation. Weitere Informationen über die ACPI CA finden Sie unter: `https://acpica.org/` ACPI ist eine offene Industriespezifikation, die ursprünglich von Hewlett-Packard, Intel, Microsoft, Phoenix und Toshiba mitentwickelt wurde. Derzeit wird sie von der ACPI Specification Working Group (ASWG) im Rahmen des UEFI-Forums entwickelt, und jedes UEFI-Mitglied kann der ASWG beitreten und zur ACPI-Spezifikation beitragen.
Die Spezifikation ist verfügbar unter: `https://uefi.org/specifications`.

5.10.1 AML debugger interface

CONFIG_ACPI_DEBUGGER [=n] **[N]**
Aktiviert das In-Kernel-Debugging von AML-Funktionen: Statistiken, interner Objekt-Dump, Ausführung von Einzelschritt-Kontrollmethoden. Dies befindet sich noch in der Entwicklung, derzeit führt die Aktivierung nur zur Kompilierung der ACPICA-Debugger-Dateien.

5.10.2 ACPI Serial Port Console Redirection Support

CONFIG_ACPI_SPCR_TABLE [=y] **[Y]**
Aktiviert die Unterstützung für die Serial Port Console Redirection (SPCR) Tabelle. Diese Tabelle enthält Informationen über die Konfiguration der earlycon-Konsole.

5.10.3 ACPI Firmware Performance Data Table (FPDT] support

CONFIG_ACPI_FPDT [=y] **[Y]**
Aktiviert die Unterstützung für die Firmware Performance Data Table (FPDT). Diese Tabelle enthält Informationen über das Timing des Systemstarts, der S3-Suspend- und S3-Resume-Firmware-Codepfade.

5.10.4 Allow supported ACPI revision to be overridden

CONFIG_ACPI_FPDT [=y] **[Y]**
(Erlaubt das Überschreiben der unterstützten ACPI-Revision)
Die Plattform-Firmware auf einigen Systemen erwartet, dass Linux „5“ als unterstützte ACPI-Revision zurückgibt, was dazu führt, dass sie Systemkonfigurationsinformationen auf eine besondere Weise offenlegt. Basierend darauf, was ACPI als unterstützte Revision exportiert, konfiguriert beispielsweise das Dell XPS 13 (2015) sein Audiogerät so, dass es entweder im HDA-Modus oder im I2S-Modus arbeitet, wobei ersterer unter Linux verwendet werden soll, bis letzterer vollständig unterstützt wird (sowohl im Kernel als auch im Userspace). Diese Option ermöglicht eine DMI-basierte Besonderheit für den oben genannten Dell-Rechner (so dass HDA-Audio von der Plattform-Firmware dem Kernel offengelegt wird)

und macht es möglich, den Kernel zu zwingen, „5“ als unterstützte ACPI-Revision über den Befehlszeilenschalter `acpi_rev_override` zurückzugeben.

5.10.5 EC read/write access through /sys/kernel/debug/ec

CONFIG_ACPI_EC_DEBUGFS [=m] **[M]**
Sagen Sie N, um die Schnittstelle Embedded Controller /sys/kernel/debug zu deaktivieren. Beachten Sie, dass die Verwendung dieser Schnittstelle Ihren Embedded Controller so verwirren kann, dass ein normaler Neustart nicht ausreicht. Sie müssen dann Ihr System ausschalten und den Akku des Laptops für einige Sekunden entfernen. Ein Embedded Controller ist in der Regel auf Laptops vorhanden und liest Sensorwerte wie Batteriestatus und Temperatur aus. Der Kernel greift auf den EC über ACPI-geparsten Code zu, der von BIOS-Tabellen bereitgestellt wird. Diese Option ermöglicht den direkten Zugriff auf den EC, ohne dass ACPI-Code involviert ist.
Somit ist diese Option eine Debug-Option, die beim Schreiben von ACPI-Treibern hilft und zur Identifizierung von ACPI-Code oder EC-Firmware-Fehlern verwendet werden kann.

5.10.6 AC Adapter

CONFIG_ACPI_AC [=y] **[Y]**
Dieser Treiber unterstützt das AC-Adapter-Objekt, das anzeigt, ob ein System mit Wechselstrom betrieben wird oder nicht. Wenn Sie ein System haben, das zwischen Wechselstrom und Batterie umschalten kann, sagen Sie Y. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `ac` heißen.

5.10.7 Battery

CONFIG_ACPI_BATTERY [=y] **[Y]**
Dieser Treiber bietet Unterstützung für Batterieinformationen über /proc/acpi/battery. Wenn Sie ein mobiles System mit einer Batterie haben, sagen Sie Y. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `battery` genannt.

5.10.8 Button

CONFIG_ACPI_BUTTON [=y] **[Y]**
Dieser Treiber verarbeitet Ereignisse für die Tasten Power, Sleep und Deckel. Ein Daemon liest Ereignisse von Eingabegeräten oder über Netlink und führt benutzerdefinierte Aktionen wie das Herunterfahren des Systems aus. Dies ist für die softwaregesteuerte Abschaltung erforderlich. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `button` genannt.

5.10.9 Video

CONFIG_ACPI_VIDEO [=m] **[M]**
Dieser Treiber implementiert die ACPI-Erweiterungen für Display-Adapter für integrierte Grafikgeräte auf dem Motherboard, wie in der ACPI 2.0-Spezifikation, Anhang B, angegeben. Er unterstützt grundlegende Vorgänge wie das Definieren des Video-POST-Geräts, das Abrufen von EDID-Informationen und das Einrichten eines Videoausgangs. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `video` genannt.

5.10.10 Fan

CONFIG_ACPI_FAN [=y] **[Y]**
Dieser Treiber unterstützt ACPI-Lüftergeräte und ermöglicht es Anwendungen im Benutzermodus, grundlegende Lüftersteuerungen (Ein, Aus, Status) durchzuführen. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `fan` genannt.

5.10.11 ACPI Time and Alarm (TAD) Device Support

CONFIG_ACPI_TAD [=m] **[N]**
Das ACPI Time and Alarm (TAD) Gerät ist eine Alternative zur Real Time Clock (RTC). Seine Weckzeitgeber ermöglichen es dem System, nach Ablauf einer bestimmten Zeitspanne vom Zustand S3 (oder

optional S4/S5) in den Zustand S0 überzugehen. Im Vergleich zum RTC-Alarm bietet der TAD eine größere Flexibilität bei den Wake-Timern. Die Zeitfunktionen des TAD behalten die Tageszeitinformationen bei, auch wenn die Plattform ausgeschaltet ist.

5.10.12 Dock

CONFIG_ACPI_DOCK [=y] **[Y]**
Dieser Treiber unterstützt ACPI-gesteuerte Dockingstationen und Wechsellaufwerkseinschübe wie den IBM Ultrabay und den Dell Module Bay.

5.10.13 Processor

CONFIG_ACPI_PROCESSOR [=y] **[Y]**
Dieser Treiber bietet Unterstützung für das ACPI-Prozessor-Paket. Er wird von mehreren Varianten der cpufreq-Treiber für den Leistungszustand, die Wärmeentwicklung, die Drosselung und den Leerlauf benötigt. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul heißt dann `processor`.

5.10.14 IPMI

CONFIG_ACPI_IPMI [=m] **[M]**
Dieser Treiber ermöglicht dem ACPI den Zugriff auf den BMC-Controller. Und er verwendet die IPMI-Anfrage/Antwort-Nachricht zur Kommunikation mit dem BMC-Controller, der sich auf dem Server befindet. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird als `acpi_ipmi` aufgerufen.

5.10.15 Processor Aggregator

CONFIG_ACPI_PROCESSOR_AGGREGATOR [=m] **[M]**
ACPI 4.0 definiert einen Prozessor-Aggregator, der es dem Betriebssystem ermöglicht, eine spezifische Prozessorkonfiguration und -steuerung durchzuführen, die für alle Prozessoren der Plattform gilt. Derzeit ist nur der logische Leerlauf des Prozessors definiert, der den Stromverbrauch senken soll. Dieser Treiber unterstützt das neue Gerät.

5.10.16 Thermal Zone

CONFIG_ACPI_THERMAL [=y] **[Y]**
Dieser Treiber unterstützt ACPI-Thermozonen. Die meisten mobilen und einige Desktop-Systeme unterstützen ACPI-Wärmezonen. Es wird DRINGEND empfohlen, diese Option zu aktivieren, da Ihr(e) Prozessor(en) sonst beschädigt werden können. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `thermal` genannt.

5.10.17 Allow upgrading ACPI tables via initrd

CONFIG_ACPI_TABLE_UPGRADE [=y] **[Y]**
Diese Option bietet die Möglichkeit, beliebige ACPI-Tabellen über initrd zu aktualisieren. Keine funktionale Änderung, wenn keine ACPI-Tabellen über initrd übergeben werden, daher ist es sicher, Y zu sagen. Siehe Documentation/admin-guide/acpi/initrd_table_override.rst für Details

5.10.18 Debug Statements

CONFIG_ACPI_DEBUG [=y] **[Y]**
Das ACPI-Subsystem kann Debug-Ausgaben erzeugen. Die Angabe von Y aktiviert diese Ausgabe und erhöht die Kernelgröße um etwa 50K.
Verwenden Sie die Kernel-Befehlszeilenparameter `acpi.debug_layer` und `acpi.debug_level`, die in Documentation/firmware-guide/acpi/debug.rst und Documentation/admin-guide/kernel-parameters.rst dokumentiert sind, um die Art und Menge der Debug-Ausgabe zu steuern.

5.10.19 PCI slot detection driver

CONFIG_ACPI_PCI_SLOT [=y] **[Y]**
Dieser Treiber erstellt Einträge in `/sys/bus/pci/slots/` für alle PCI-Steckplätze im System. Dies kann helfen, PCI-Bus-Adressen, d.h. Segment/Bus/Gerät/Funktions-Tupel, mit physischen Steckplätzen im System zu korrelieren.
Wenn Sie unsicher sind, sagen Sie N.

5.10.20 Container and Module Devices

CONFIG_ACPI_CONTAINER [=y] **[Y]**
Dieser Treiber unterstützt ACPI-Container- und Modulgeräte (IDs ACPI0004, PNP0A05 und PNP0A06). Dies hilft, Hotplug von Knoten, CPUs und Speicher zu unterstützen.

5.10.21 Memory Hotplug

CONFIG_ACPI_HOTPLUG_MEMORY [=y] **[Y]**
Dieser Treiber unterstützt ACPI-Speicher-Hotplug. Die Treiberfelder enthalten Benachrichtigungen über ACPI-Speichergeräte (PNP0C80), die Speicherbereiche darstellen, die zur Laufzeit ein- oder ausgeschaltet werden können. Wenn Ihre Hardware und Firmware das Hinzufügen oder Entfernen von Speichergeräten zur Laufzeit nicht unterstützen, müssen Sie diesen Treiber nicht aktivieren.

5.10.22 Smart Battery System

CONFIG_ACPI_SBS [=m] **[N]**
Dieser Treiber unterstützt das Smart Battery System, eine andere Art des Zugriffs auf Batterieinformationen, die bei einigen Laptops zu finden ist. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Die Module heißen dann sbs und sbshc.

5.10.23 Hardware Error Device

CONFIG_ACPI_HED [=y] **[Y]**
Dieser Treiber unterstützt das Hardware Error Device (PNP0C33), das dazu dient, einige über SCI gemeldete Hardwarefehler zu melden, hauptsächlich die korrigierten Fehler.

5.10.24 Allow ACPI methods to be inserted/replaced at run time

CONFIG_ACPI_CUSTOM_METHOD [=m] **[M]**
Mit dieser Debug-Funktion können ACPI-AML-Methoden eingefügt und/oder ersetzt werden, ohne dass das System neu gestartet werden muss.
Für Details siehe: Documentation/firmware-guide/acpi/method-customizing.rst.
HINWEIS: Diese Option ist sicherheitsrelevant, da sie es erlaubt, dass root (uid=0) Benutzer in beliebigen Kernelspeicher schreiben können und so bestimmte Sicherheitsmaßnahmen umgehen können (z. B. wenn es root nicht erlaubt ist, zusätzliche Kernelmodule nach dem Booten zu laden, kann diese Funktion verwendet werden, um diese Einschränkung zu umgehen).

5.10.25 Boottime Graphics Resource Table support

CONFIG_ACPI_BGRT [=y] **[Y]**
Dieser Treiber bietet Unterstützung für die ACPI Boottime Graphics Resource Table, die es dem Betriebssystem ermöglicht, Daten aus dem Firmware-Boot-Splash zu beziehen.
Er erscheint unter `/sys/firmware/acpi/bgrt/` .

5.10.26 ACPI NVDIMM Firmware Interface Table (NFIT)

CONFIG_ACPI_NFIT [=m] **[M]**
Infrastruktur, um ACPI 6-konforme Plattformen auf NVDIMMs zu untersuchen (NFIT) und einen libnvdimm-Gerätebaum zu registrieren. Zusätzlich zu den Speichergeräten ermöglicht dies libnvdimm auch die Weitergabe von ACPI._DSM-Nachrichten für die Plattform/Dimm-Konfiguration. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `nfit` genannt.

5.10.26.1 Enable debug for NVDIMM security commands

CONFIG_NFIT_SECURITY_DEBUG [=n] **[N]**
Einige NVDIMM-Geräte und -Controller unterstützen Verschlüsselung und andere Sicherheitsfunktionen. Die Nutzdaten für die Befehle, die diese Funktionen aktivieren, können sensibles Sicherheitsmaterial im Klartext enthalten. Deaktivieren Sie das Debuggen dieser Befehls-Payloads standardmäßig. Wenn Sie ein Kernel-Entwickler sind, der aktiv an der Aktivierung der NVDIMM-Sicherheit arbeitet, sagen Sie Y, andernfalls sagen Sie N.

5.10.27 NUMA support

CONFIG_ACPI_NUMA [=y] **[Y]**
Für diese Option ist keine Hilfe verfügbar.

5.10.27.1 ACPI Heterogeneous Memory Attribute Table Support

CONFIG_ACPI_HMAT [=y] **[Y]**
Wenn diese Option gesetzt ist, lässt der Kernel die ACPI HMAT (Heterogeneous Memory Attributes Table) der Plattform auslesen und melden, Speicherinitiatoren mit ihren Zielen registrieren und Leistungsattribute über das sysfs-Gerät des Knotens exportieren, falls vorhanden.

5.10.28 ACPI Platform Error Interface (APEI)

CONFIG_ACPI_APEI [=y] **[Y]**
APEI ermöglicht es, Fehler (z. B. vom Chipsatz) an das Betriebssystem zu melden. Dies verbessert insbesondere die NMI-Behandlung. Darüber hinaus unterstützt es Fehlerserialisierung und Fehlerinjektion.

5.10.28.1 ACPI Generic Hardware Error Source

CONFIG_ACPI_APEI_GHES [=y] **[Y]**
Generic Hardware Error Source bietet eine Möglichkeit, Plattform-Hardware-Fehler (z. B. vom Chipsatz) zu melden. Sie arbeitet im so genannten „Firmware First“-Modus, d. h. Hardwarefehler werden zunächst an die Firmware gemeldet und dann von der Firmware an Linux weitergeleitet. Auf diese Weise können einige Nicht-Standard-Hardware-Fehlerregister oder Nicht-Standard-Hardware-Verbindungen von der Firmware überprüft werden, um wertvollere Hardware-Fehlerinformationen für Linux zu erhalten.

5.10.28.2 ACPI PCIe AER logging/recovering support

CONFIG_ACPI_APEI_PCIEAER [=y] **[Y]**
PCIe-AER-Fehler können über den APEI-Firmware-First-Modus gemeldet werden. Aktivieren Sie diese Option, um die entsprechende Unterstützung zu aktivieren.

5.10.29 ACPI memory error recovering support

CONFIG_ACPI_APEI_MEMORY_FAILURE [=y] **[Y]**
Speicherfehler können über den APEI-Firmware-First-Modus gemeldet werden. Aktivieren Sie diese Option, um die Unterstützung für die Speicherwiederherstellung zu aktivieren.

5.10.30 APEI Error INJection (EINJ)

CONFIG_ACPI_APEI_EINJ [=m] **[M]**
EINJ bietet einen Hardware-Fehlerinjektionsmechanismus, der hauptsächlich zur Fehlersuche und zum Testen der anderen Teile von APEI und einiger anderer RAS-Funktionen verwendet wird.

5.10.31 APEI Error Record Serialization Table (ERST) Debug Support

CONFIG_ACPI_APEI_ERST_DEBUG [=m] **[M]**
ERST ist eine von APEI bereitgestellte Möglichkeit, Hardware-Fehlerinformationen in einem dauerhaften Speicher zu speichern und von dort abzurufen. Aktivieren Sie dies, wenn Sie die ERST-Kernelunterstützung und Firmware-Implementierung debuggen und testen wollen.

5.10.32 Intel DPTF (Dymnamic Platform and Thermal Framework) Support →

CONFIG_ACPI_DPTF [=y] [**Y**]
Intel Dynamic Platform and Thermal Framework (DPTF) ist eine Hardware-/Softwarelösung auf Plattformebene für das Energie- und Wärmemanagement. Als Container für mehrere Energie-/Thermotechnologien bietet DPTF einen koordinierten Ansatz für verschiedene Richtlinien, die den Hardwarezustand eines Systems beeinflussen.

5.10.32.1 Platform Power DPTF Participant

CONFIG_DPTF_POWER [=m] [**M**]
Dieser Treiber bietet Unterstützung für das Dynamic Platform and Thermal Framework (DPTF) Platform Power Participant Device (INT3407). Dieser Teilnehmer ist für die Offenlegung der Plattformtelemetrie verantwortlich:

- max_platform_power (max. Plattformleistung)
- platform_power_source (Plattformstromquelle)
- adapter_rating (Leistung des Netzteils)
- battery_steady_power (Dauerleistung der Batterie)
- ladegerät_typ (Ladegerättyp)

Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul heißt dann `dptf_power`.

5.10.32.2 PCH FIVR DPTF Participant

CONFIG_DPTF_PCH_FIVR [=m] [**M**]
Dieser Treiber fügt Unterstützung für Dynamic Platform and Thermal Framework (DPTF) PCH FIVR Participant Device Support hinzu. Dieser Treiber ermöglicht es, die Frequenz des PCH FIVR (Fully Integrated Voltage Regulator) zu schalten. Dieser Teilnehmer ist für die Bereitstellung verantwortlich:

- freq_mhz_low_clock
- freq_mhz_high_clock

Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `dptf_pch_fivr` heißen.

5.10.33 Extended Error Log support

CONFIG_ACPI_EXTLOG [=m] [**M**]
Bestimmte Anwendungen wie die vorausschauende Fehleranalyse (Predictive Failure Analysis, PFA) erfordern mehr Informationen über den Fehler, als in den Prüfbänken der Prozessormaschine beschrieben werden können. Die meisten Server-Prozessoren protokollieren zusätzliche Informationen über den Fehler in Prozessor-Uncore-Registern. Da die Adressen und das Layout dieser Register von einem Prozessor zum anderen sehr unterschiedlich sind, kann die Systemsoftware sie nicht ohne weiteres nutzen. Erschwerend kommt hinzu, dass einige der zusätzlichen Fehlerinformationen nicht ohne detaillierte Kenntnisse der Plattformtopologie erstellt werden können. Die erweiterte MCA-Protokollierung ermöglicht es der Firmware, der Systemsoftware synchron mit MCE oder CMCI zusätzliche Fehlerinformationen zu liefern. Dieser Treiber unterstützt diese Funktionalität mit einem entsprechenden Tracepoint, der diese Informationen an den Userspace weiterleitet.

5.10.34 ACPI configfs support

CONFIG_ACPI_CONFIGFS [=m] [**M**]
Wählen Sie diese Option, um die Unterstützung für die ACPI-Konfiguration aus dem Userspace zu aktivieren. Die konfigurierbaren ACPI-Gruppen sind dann unter /config/acpi sichtbar, vorausgesetzt, configfs ist unter /config eingebunden.

5.10.35 ACPI Platform Firmware Runtime Update and Telemetry

CONFIG_ACPI_PFRUT [=m] **[M]**
Dieser Mechanismus ermöglicht es, bestimmte Teile der Plattform-Firmware während des laufenden Betriebs (Laufzeit) zu aktualisieren, ohne dass ein Neustart erforderlich ist. Dies ist von entscheidender Bedeutung, wenn das System zu 100 % verfügbar sein muss und sich die mit einem Neustart verbundene Ausfallzeit nicht leisten kann, oder wenn die vom System ausgeführte Arbeit besonders wichtig ist, so dass sie nicht unterbrochen werden kann und es nicht sinnvoll ist, zu warten, bis sie abgeschlossen ist. Der bestehende Firmware-Code kann geändert (Treiber-Update) oder durch Hinzufügen neuen Codes zur Firmware erweitert werden (Code-Injektion). Außerdem ermöglicht der Telemetrietreiber dem Benutzer, mit Hilfe der Plattform-Firmware-Laufzeit-Telemetrieschnittstelle Telemetriedaten aus der Firmware abzurufen. Um die Treiber als Module zu kompilieren, wählen Sie hier M: die Module heißen dann `pfr_update` und `pfr_telemetry`.

5.10.36 ACPI PCC Address Space

CONFIG_ACPI_PCC [=y] **[Y]**
Der PCC-Adressraum, der auch als PCC-Operationsbereich bezeichnet wird, bezieht sich auf den Bereich des PCC-Unterraums, der auf die PCC-Signatur folgt. Die PCC Operation Region arbeitet mit der PCC Table (Platform Communications Channel Table) zusammen. PCC-Unterräume, die für die Verwendung als PCC Operation Region markiert sind, dürfen nicht als PCC-Unterräume für die Standard-ACPI-Funktionen wie CPPC, RASF, PDTT und MPST verwendet werden. Diese Standardfunktionen müssen stattdessen immer die PCC-Tabelle verwenden. Aktivieren Sie diese Funktion, wenn Sie den PCC Address Space Handler einrichten und installieren möchten, um PCC OpRegion in der Firmware zu behandeln.

5.10.37 ACPI FFH Address Space

CONFIG_ACPI_FFH [=y] **[Y]**
Der FFH (Fixed Function Hardware) Adressraum, auch FFH Operation Region genannt, erlaubt es, plattformspezifische OpRegions zu definieren. Aktivieren Sie diese Funktion, wenn Sie den FFH-Adressraum-Handler einrichten und installieren möchten, um die FFH-OpRegion in der Firmware zu behandeln.

5.10.38 PMIC (Power Management Integrated Circuit) operation region support

CONFIG_PMIC_OPREGION [=y] **[Y]**
Wählen Sie diese Option, um die Unterstützung für den ACPI-Betriebsbereich des PMIC-Chips zu aktivieren. Der Betriebsbereich kann zur Steuerung von Stromschienen und zum Lesen/Schreiben von Sensoren auf dem PMIC-Chip verwendet werden.

5.10.39 ACPI operation region support for TPS68470 PMIC

CONFIG_TPS68470_PMIC_OPREGION [=y] **[Y]**
Diese Konfiguration fügt ACPI-Betriebsbereich-Unterstützung für TI TPS68470 PMIC hinzu. Der Baustein TPS68470 ist eine fortschrittliche Energieverwaltungseinheit, die ein Kompaktkameramodul (CCM) mit Strom versorgt, Takte für Bildsensoren erzeugt, eine Dual-LED für den Blitz ansteuert und zwei LED-Treiber für allgemeine Anzeigen enthält. Dieser Treiber ermöglicht die Unterstützung der ACPI-Betriebsregion für die Steuerung von Spannungsreglern und Taktgebern. Bei dieser Option handelt es sich um ein bool, da sie eine ACPI-Betriebsregion bereitstellt, die verfügbar sein muss, bevor eines der Geräte, die diese Option verwenden, getestet wird.

5.10.40 Platform Runtime Mechanism Support

CONFIG_ACPI_PRMT [=y] **[Y]**
Der Plattform-Laufzeit-Mechanismus (Platform Runtime Mechanism, PRM) ist eine Firmware-Schnittstelle, die eine Reihe von ausführbaren Binärdateien bereitstellt, die vom AML-Interpreter oder direkt von Gerätetreibern aufgerufen werden können. Sagen Sie Y, um den AML-Interpreter für die Ausführung des PRM-Codes zu aktivieren. Während diese Funktion im Prinzip optional ist, kann das Weglassen dieser Funktion den Rechenaufwand für die Initialisierung einiger Serversysteme erheblich erhöhen.

5.11 CPU Frequency scaling →

CONFIG_CPU_FREQ [=y] **[Y]**
Mit der CPU-Frequenzskalierung können Sie die Taktfrequenz von CPUs im laufenden Betrieb ändern. Dies ist eine gute Methode, um Strom zu sparen, denn je niedriger die CPU-Taktfrequenz, desto weniger Strom verbraucht die CPU. Beachten Sie, dass dieser Treiber die CPU-Taktfrequenz nicht automatisch ändert. Sie müssen entweder einen dynamischen cpufreq-Governor (siehe unten) nach dem Booten aktivieren oder ein Userspace-Tool verwenden.
Details finden Sie in <file:Documentation/admin-guide/pm/cpufreq.rst>. Im Zweifelsfall sagen Sie N.

5.11.1 CPU frequency transition statistics

CONFIG_CPU_FREQ_STAT [=y] **[Y]**
Exportieren Sie CPU-Häufigkeitsstatistiken über sysfs. Im Zweifelsfall sagen Sie N.

5.11.2 Default CPUFreq governor () →

Diese Option legt fest, welcher CPUFreq-Governor beim Start geladen werden soll. Im Zweifelsfall ist die Standardeinstellung zu verwenden.

5.11.2.1 performance
CONFIG_CPU_FREQ_DEFAULT_GOV_PERFORMANCE [=n] **[N]**
Verwenden Sie den CPUFreq-Governor ‚performance' als Standard. Damit wird die Frequenz statisch auf die höchste von der CPU unterstützte Frequenz eingestellt.

5.11.2.2 powersave
CONFIG_CPU_FREQ_DEFAULT_GOV_POWERSAVE [=n] **[N]**
Verwenden Sie den CPUFreq-Governor ‚powersave' als Standard. Damit wird die Frequenz statisch auf die niedrigste von der CPU unterstützte Frequenz eingestellt.

5.11.2.3 userspace
CONFIG_CPU_FREQ_DEFAULT_GOV_USERSPACE [=n] **[N]**
Verwenden Sie den CPUFreq-Governor ‚userspace' als Standard. Damit können Sie die CPU-Frequenz manuell einstellen oder ein Userspace-Programm soll die CPU dynamisch einstellen können, ohne den Userspace-Governor manuell aktivieren zu müssen.

5.11.2.4 schedutil
CONFIG_CPU_FREQ_DEFAULT_GOV_SCHEDUTIL [=y] **[Y]**
Verwenden Sie standardmäßig den CPUFreq-Governor ‚schedutil'. Wenn Sie sich nicht sicher sind, sehen Sie in der Hilfe zu diesem Gouverneur nach. Der Fallback-Regler ist „performance".

5.11.3 'performance' governor

CONFIG_CPU_FREQ_GOV_PERFORMANCE [=y] **[Y]**
Dieser cpufreq-Regler setzt die Frequenz statisch auf die höchste verfügbare CPU-Frequenz. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `cpufreq_performance` heißen. Im Zweifelsfall sagen Sie Y.

5.11.4 'powersave' governor

CONFIG_CPU_FREQ_GOV_POWERSAVE [=y] **[Y]**
Dieser cpufreq-Regler setzt die Frequenz statisch auf die niedrigste verfügbare CPU-Frequenz. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `cpufreq_powersave` heißen. Im Zweifelsfall wählen Sie Y.

5.11.5 'userspace' governor for userspace frequency scaling

CONFIG_CPU_FREQ_GOV_USERSPACE [=y] **[Y]**
Aktivieren Sie diesen cpufreq-Governor, wenn Sie die CPU-Frequenz entweder manuell einstellen wollen oder wenn ein Userspace-Programm in der Lage sein soll, die CPU dynamisch einzustellen, wie bei LART `http://www.lartmaker.nl/`. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird cpufreq_userspace heißen. Im Zweifelsfall sagen Sie Y.

5.11.6 'ondemand' cpufreq policy governor

CONFIG_CPU_FREQ_GOV_ONDEMAND [=y] **[Y]**
'ondemand' – Dieser Treiber fügt einen dynamischen cpufreq policy governor hinzu. Der Gouverneur führt eine periodische Abfrage durch und ändert die Frequenz auf der Grundlage der CPU-Auslastung. Die Unterstützung für diesen Gouverneur hängt von der Fähigkeit der CPU ab, schnelle Frequenzwechsel durchzuführen (d.h. Frequenzübergänge mit sehr geringer Latenzzeit). Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `cpufreq_ondemand` heißen. Details finden Sie in <file:Documentation/admin-guide/pm/cpufreq.rst>. Im Zweifelsfall sagen Sie N.

5.11.7 'conservative' cpufreq governor

CONFIG_CPU_FREQ_GOV_CONSERVATIVE [=y] **[Y]**
'konservativ' – dieser Treiber ähnelt dem „On-Demand"-Regler sowohl in seinem Quellcode als auch in seinem Zweck, der Unterschied besteht in seiner Optimierung für eine bessere Eignung in einer batteriebetriebenen Umgebung. Die Frequenz wird sanft erhöht und gesenkt, anstatt auf 100 % zu springen, wenn die Geschwindigkeit erforderlich ist. Wenn Sie einen Desktop-Rechner haben, sollten Sie stattdessen den „On-Demand"-Regler in Betracht ziehen. Wenn Sie jedoch einen Laptop, einen PDA oder sogar einen AMD64-basierten Computer verwenden (wegen der inakzeptablen schrittweisen Latenzprobleme zwischen den minimalen und maximalen Frequenzübergängen in der CPU), werden Sie wahrscheinlich diesen Regler verwenden wollen. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `cpufreq_conservative` heißen.
Einzelheiten finden Sie in <file:Documentation/admin-guide/pm/cpufreq.rst>.
Im Zweifelsfall sagen Sie N.

5.11.8 'schedutil' cpufreq policy governor

CONFIG_CPU_FREQ_GOV_SCHEDUTIL [=y] **[Y]**
Dieser Gouverneur trifft seine Entscheidungen auf der Grundlage der vom Scheduler bereitgestellten Nutzungsdaten. Er stellt die CPU-Frequenz so ein, dass sie proportional zu dem vom Scheduler gelieferten Verhältnis zwischen Auslastung und Kapazität ist. Wenn die Auslastung frequenzinvariant ist, ist die neue Frequenz auch proportional zur maximal verfügbaren Frequenz. Wenn dies nicht der Fall ist, ist sie proportional zur aktuellen Frequenz der CPU. Der Kipppunkt der Frequenz liegt in beiden Fällen bei einer Auslastung/Kapazität von 80 %.
Im Zweifelsfall sagen Sie N.

***** CPU frequency scaling drivers *****

(Treiber zur Skalierung der CPU-Frequenz)

5.11.9 Intel P state control

CONFIG_X86_INTEL_PSTATE [=y] **[Y]**
Dieser Treiber bietet einen P-Status für Intel-Core-Prozessoren. Der Treiber implementiert einen internen Gouverneur und wird der Skalierungstreiber und Gouverneur für Sandy-Bridge-Prozessoren werden. Wenn dieser Treiber aktiviert ist, wird er der bevorzugte Skalierungstreiber für Sandy-Bridge-Prozessoren.
Im Zweifelsfall sagen Sie N.

5.11.10 Processor Clocking Control interface driver

CONFIG_X86_PCC_CPUFREQ [=y] **[Y]**
Dieser Treiber bietet Unterstützung für die PCC-Schnittstelle. Einzelheiten finden Sie unter:

<file:Documentation/admin-guide/pm/cpufreq_drivers.rst>. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `pcc-cpufreq` heißen. Im Zweifelsfall sagen Sie N.

5.11.11 AMD Processor P-State driver

CONFIG_X86_AMD_PSTATE [=y] **[Y]**
Dieser Treiber fügt einen CPUFreq-Treiber hinzu, der einen feinkörnigen Frequenzsteuerungsbereich für die Prozessorleistung anstelle der alten Leistungsstufen verwendet. In den ACPI-Tabellen des Systems muss _CPC vorhanden sein.
Details finden Sie unter: <file:Documentation/admin-guide/pm/amd-pstate.rst>. Im Zweifelsfall sagen Sie N.

5.11.11.1 AMD Processor P-State default mode

CONFIG_X86_AMD_PSTATE_DEFAULT_MODE [=3] **[3]**
Wählen Sie den Standardmodus, den der amd-pstate-Treiber auf unterstützter Hardware verwenden soll. Der eingestellte Wert hat die folgenden Bedeutungen:

1 → Deaktiviert

2 → Passiv

3 → Aktiv (EPP)

4 → Geführt

Für Details, siehe: <file:Documentation/admin-guide/pm/amd-pstate.rst>.
Symbol: X86_AMD_PSTATE_DEFAULT_MODE [=3]
Type : Ganzzahl (integer)
Bereich (range): [1 4]

5.11.12 selftest for AMD Processor P-State driver

CONFIG_X86_AMD_PSTATE_UT [=m] **[M]**
Dieses Kernelmodul wird für Tests verwendet. Hier kann man mit Sicherheit M sagen. Es kann auch ohne aktiviertes X86_AMD_PSTATE eingebaut werden. Derzeit werden nur Tests für amd-pstate unterstützt. Wenn X86_AMD_PSTATE deaktiviert ist, kann es den Benutzern sagen, dass der Test nur auf dem amd-pstate Treiber laufen kann, bitte setzen Sie X86_AMD_PSTATE aktiviert. In der Zukunft werden Vergleichstests hinzugefügt werden. Es kann amd-pstate deaktiviert und acpi-cpufreq aktiviert werden, um Testfälle auszuführen und dann die Testergebnisse zu vergleichen.

5.11.13 ACPI Processor P-State driver

CONFIG_X86_ACPI_CPUFREQ [=m] **[M]**
Dieser Treiber fügt einen CPUFreq-Treiber hinzu, der die ACPI Processor Performance States nutzt. Dieser Treiber unterstützt auch Intel Enhanced Speedstep und neuere AMD-CPUs. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `acpi-cpufreq` heißen.
Details finden Sie unter <file:Documentation/cpu-freq/>. Im Zweifelsfall sagen Sie N.

5.11.13.1 Legacy cpb sysfs knob support for AMD CPUs

CONFIG_X86_ACPI_CPUFREQ_CPB [=y] **[Y]**
Der powernow-k8-Treiber stellte früher einen sysfs-Regler namens `cpb` zur Verfügung, um die Core Performance Boosting-Funktion von AMD-CPUs zu deaktivieren. Diese Datei wurde nun durch den allgemeineren „boost“-Eintrag abgelöst. Wenn Sie diese Option aktivieren, stellt der acpi_cpufreq-Treiber aus Kompatibilitätsgründen den alten Eintrag zusätzlich zum neuen „boost“-Eintrag bereit.

5.11.14 AMD Opteron/Athlon64 PowerNow!

CONFIG_X86_POWERNOW_K8 [=m] **[M]**
Dies fügt den CPUFreq-Treiber für K8/frühe Opteron/Athlon64-Prozessoren hinzu. Unterstützung für K10 und neuere Prozessoren ist jetzt in acpi-cpufreq enthalten. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `powernow-k8` heißen.
Details finden Sie in <file:Documentation/cpu-freq/>.

5.11.15 AMD frequency sensitivity feedback powersave bias

CONFIG_X86_AMD_FREQ_SENSITIVITY [=m] **[M]**
Dies fügt dem On-Demand-Governor eine AMD-spezifische Powersave-Bias-Funktion hinzu, die es ihm ermöglicht, auf der Grundlage von Rückmeldungen der Hardware energiebewusstere Entscheidungen über Frequenzänderungen zu treffen (verfügbar ab AMD-Familie 16h). Durch das Hardware-Feedback erfährt die Software, wie „empfindlich“ die Arbeitslasten der CPUs gegenüber Frequenzänderungen sind. CPU-gebundene Arbeitslasten sind empfindlicher, d. h. sie werden bei einer Frequenzerhöhung besser funktionieren. Speicher-/IO-gebundene Arbeitslasten reagieren weniger empfindlich, d. h. sie werden nicht unbedingt besser, wenn die Frequenz erhöht wird.
Im Zweifelsfall sagen Sie N.

5.11.16 Intel Enhanced SpeedStep (deprecated)

CONFIG_X86_SPEEDSTEP_CENTRINO [=n] **[N]**
Dies ist veraltet und diese Funktionalität ist nun in acpi_cpufreq (X86_ACPI_CPUFREQ) integriert. Verwenden Sie diesen Treiber anstelle von speedstep_centrino. Dies fügt den CPUFreq-Treiber für Enhanced SpeedStep-fähige mobile CPUs hinzu. Dies bedeutet Intel Pentium M (Centrino) CPUs oder 64bit-fähige Intel Xeons. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `speedstep-centrino` heißen.
Details finden Sie unter <file:Documentation/cpu-freq/>. Im Zweifelsfall wählen Sie N.

5.11.17 Intel Pentium 4 clock modulation

CONFIG_X86_P4_CLOCKMOD [=m] **[N]**
Dies fügt den CPUFreq-Treiber für Intel Pentium 4 / XEON Prozessoren hinzu. Wenn er aktiviert ist, senkt er die CPU-Temperatur durch Überspringen von Takten. Dieser Treiber sollte nur in Ausnahmefällen verwendet werden, wenn eine sehr niedrige Leistung benötigt wird, da er starke Verlangsamungen und spürbare Latenzen verursacht. Normalerweise sollte stattdessen Speedstep verwendet werden. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: das Modul wird `p4-clockmod` genannt.
Für Details werfen Sie einen Blick auf <file:Documentation/cpu-freq/>. Wenn Sie sich nicht absolut sicher sind, wählen Sie N.

***** shared options *****

(gemeinsame Optionen)

5.12 CPU Idle →

(CPU im Leerlauf)

5.12.1 CPU idle PM support

CONFIG_CPU_IDLE [=y] **[Y]**
CPU idle ist ein allgemeiner Mechanismus zur Unterstützung der softwaregesteuerten Verwaltung der Prozessorleistung im Leerlauf. Es umfasst modulare plattformübergreifende Regler, die während der Laufzeit ausgetauscht werden können. Wenn Sie eine ACPI-aktivierte Plattform verwenden, sollten Sie hier Y angeben. PM steht für „power management“ – Verwaltung der Prozessorleistung.

5.12.1.1 Ladder governor (for periodic timer tick)

CONFIG_CPU_IDLE_GOV_LADDER [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

5.12.1.2 Menu governor (for tickless system)

CONFIG_CPU_IDLE_GOV_MENU [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

5.12.1.3 Timer events oriented (TEO) governor (for tickless systems)

CONFIG_CPU_IDLE_GOV_TEO [=y] **[Y]**
Dieser Gouverneur implementiert eine vereinfachte Methode zur Auswahl des Ruhezustands, die sich auf Timer-Ereignisse konzentriert und keine Steigerung der Interaktivität bewirkt. Einige Arbeitslasten profitieren davon, und es sollte im Allgemeinen sicher zu verwenden sein.
Sagen Sie hier Y, wenn Sie mit den Alternativen nicht zufrieden sind.

5.12.1.4 Haltpoll governor (for virtualized systems)

CONFIG_CPU_IDLE_GOV_HALTPOLL [=y] **[Y]**
Dieser Gouverneur implementiert die Auswahl des Leerlaufzustands von haltpoll, der in Verbindung mit dem haltpoll cpuidle-Treiber verwendet wird und es ermöglicht, eine bestimmte Zeit lang zu pollen, bevor der Leerlaufzustand erreicht wird. Einige virtualisierte Arbeitslasten profitieren von dieser Funktion.

5.12.1.5 Halt poll cpuidle driver

CONFIG_HALTPOLL_CPUIDLE [=m] **[N]**
Diese Option aktiviert den „halt poll cpuidle“-Treiber, der eine Abfrage vor dem Anhalten im Gast ermöglicht (effizienter als die Abfrage im Host über halt_poll_ns für einige Szenarien).

5.13 CPUidle Driver for Intel Processors

CONFIG_INTEL_IDLE [=y] **[Y]**
Aktivieren Sie intel_idle, einen cpuidle-Treiber, der das Wissen über native Intel-Hardware-Idle-Funktionen enthält. Der acpi_idle-Treiber kann zur gleichen Zeit konfiguriert werden, um Prozessoren zu behandeln, die intel_idle nicht unterstützt.

6 Bus options (PCI etc.) →

Bus-Optionen (PCI usw.)

6.1 Support mmconfig PCI config space access

CONFIG_PCI_MMCONFIG [=y] **[Y]**
(Unterstützung des mmconfig PCI-Konfigurationsraumzugriffs)
Für diese Option gibt es keine Hilfe.

7 Binary Emulations →

Binäre Emulationen

7.1 IA32 Emulation

CONFIG_IA32_EMULATION [=y] **[N]**
Code einbinden, um ältere 32-Bit-Programme unter einem 64-Bit-Kernel auszuführen. Sie sollten dies wahrscheinlich aktivieren, es sei denn, Sie sind sich zu 100 % sicher, dass Sie keine 32-Bit-Programme mehr haben.

7.2 x32 ABI for 64-bit mode

CONFIG_X86_X32_ABI [=n] **[N]**
Fügen Sie Code ein, um Binärdateien für die native 32-Bit-ABI x32 für 64-Bit-Prozessoren auszuführen. Ein x32-Prozess erhält Zugriff auf die vollständige 64-Bit-Registerdatei und den breiten Datenpfad, während Zeiger auf 32 Bit belassen werden, um den Speicherbedarf zu verringern.

8 Virtualization →

CONFIG_VIRTUALIZATION [=y] **[Y]**
Sagen Sie hier Y, um Optionen für die Verwendung Ihres Linux-Hosts zur Ausführung anderer Betriebssysteme in virtuellen Maschinen (Gäste) zu erhalten. Diese Option allein fügt keinen Kernel-Code hinzu. Wenn Sie N sagen, werden alle Optionen in diesem Untermenü übersprungen und deaktiviert.

8.1 Kernel-based Virtual Machine (KVM) support

CONFIG_KVM [=m] **[M]**
Unterstützung für das Hosten vollständig virtualisierter Gastmaschinen mit Hardware-Virtualisierungserweiterungen. Sie benötigen einen relativ aktuellen Prozessor mit Virtualisierungserweiterungen. Außerdem müssen Sie eines oder mehrere der unten aufgeführten Prozessormodule auswählen. Dieses Modul ermöglicht den Zugriff auf die Hardware-Funktionen über einen Geräteknoten namens /dev/kvm. Um dies als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `kvm` heißen.
Wenn Sie unsicher sind, sagen Sie N.

8.1.1 KVM for Intel (and compatible) processors support

CONFIG_KVM_INTEL [=m] **[M]**
Bietet Unterstützung für KVM auf Prozessoren, die mit Intels VT-Erweiterungen, auch bekannt als Virtual Machine Extensions (VMX), ausgestattet sind. Um dies als Modul zu kompilieren, wählen Sie hier M: das Modul wird `kvm-intel` genannt.

8.1.1.1 Software Guard eXtension (SGX) Virtualization

CONFIG_X86_SGX_KVM [=y] **[Y]**
Ermöglicht KVM-Gästen, SGX-Enklaven zu erstellen. Dies schließt die Unterstützung ein, „rohen“, nicht wiederverwendbaren Enklavenspeicher für Gäste über einen Geräteknoten, z. B. /dev/sgx_vepc, freizugeben. Wenn Sie unsicher sind, sagen Sie N.

8.1.2 KVM for AMD processors support

CONFIG_KVM_AMD [=m] **[M]**
Bietet Unterstützung für KVM auf Prozessoren, die mit Intels VT-Erweiterungen, auch bekannt Bietet Unterstützung für KVM auf AMD-Prozessoren, die mit den AMD-V (SVM)-Erweiterungen ausgestattet sind. Um dies als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `kvm-amd` genannt.

8.1.2.1 AMD Secure Encrypted Virtualization (SEV) support

CONFIG_KVM_AMD_SEV [=y] **[N]**
Bietet Unterstützung für den Start von verschlüsselten VMs (SEV) und verschlüsselten VMs mit verschlüsseltem Status (SEV-ES) auf AMD-Prozessoren.

8.1.3 System Management Mode emulation

CONFIG_KVM_SMM [=y] **[Y]**
Bietet Unterstützung für KVM zur Emulation des Systemverwaltungsmodus (SMM) in virtuellen Maschinen. Dies kann von der Firmware der virtuellen Maschine verwendet werden, um UEFI Secure Boot zu implementieren.

8.1.4 Support for Xen hypercall interface

CONFIG_KVM_XEN [=y] **[Y]**
Bietet KVM-Unterstützung für das Hosten von Xen HVM-Gästen und die Weitergabe von Xen-Hyperaufrufen an den Userspace. Im Zweifelsfall sagen Sie N.

9 General architecture-dependent options →

(Allgemeine architekturabhängige Optionen)

9.1 Kprobes

CONFIG_KPROBES [=y] **[Y]**
Mit Kprobes können Sie an fast jeder Kerneladresse trappen und eine Callback-Funktion ausführen. register_kprobe() legt einen Probepoint fest und spezifiziert den Callback. Kprobes ist nützlich für Kernel-Debugging, nicht-intrusive Instrumentierung und Tests.
Im Zweifelsfall sagen Sie "N".

9.2 Optimize very unlikely/likely branches

CONFIG_JUMP_LABEL [=y] **[Y]**
Diese Option ermöglicht eine transparente Verzweigungsoptimierung, die die Ausführung bestimmter fast-immer-wahrer oder fast-immer-falscher Verzweigungsbedingungen im Kernel noch billiger macht. Bestimmte leistungsempfindliche Kernel-Codes wie Trace-Points, Scheduler-Funktionen, Netzwerk-Code und KVM haben solche Verzweigungen und bieten Unterstützung für diese Optimierungstechnik. Wenn festgestellt wird, dass der Compiler „asm goto“ unterstützt, kompiliert der Kernel solche Verzweigungen mit einer einfachen nop-Anweisung. Wenn das Bedingungsflag auf true gesetzt wird, wird der nop-Befehl in einen Sprungbefehl umgewandelt, um den bedingten Befehlsblock auszuführen. Diese Technik senkt den Overhead und die Belastung der Verzweigungsvorhersage des Prozessors und macht den Kernel im Allgemeinen schneller. Die Aktualisierung der Bedingung ist zwar langsamer, aber das kommt immer sehr selten vor. (Bei 32-Bit-x86 können die erforderlichen Optionen, die zu den Compiler-Flags hinzugefügt werden, die Größe des Kernels leicht erhöhen).

9.2.1 Static key selftest

CONFIG_STATIC_KEYS_SELFTEST [=n] **[N]**
Bootzeit-Selbsttest des Branch-Patching-Codes.

9.3 Static call selftest

CONFIG_STATIC_CALL_SELFTEST [=n] **[N]**
Bootzeit-Selbsttest des Call-Patching-Codes.

9.4 Enable seccomp to safely execute untrusted bytecode

CONFIG_SECCOMP [=n] **[N]**
Diese Kernel-Funktion ist nützlich für numerische Anwendungen, die während ihrer Ausführung mit nicht vertrauenswürdigem Bytecode umgehen müssen. Durch die Verwendung von Pipes oder anderen Transporten, die dem Prozess als Dateideskriptoren zur Verfügung gestellt werden und die Lese-/Schreib-Syscalls unterstützen, ist es möglich, diese Anwendungen mit seccomp in ihrem eigenen Adressraum zu isolieren. Sobald seccomp über prctl(PR_SET_SECCOMP) oder den seccomp()-Syscall aktiviert ist, kann es nicht mehr deaktiviert werden, und die Task darf nur einige wenige sichere Syscalls ausführen, die für jeden seccomp-Modus definiert sind. Wenn Sie unsicher sind, sagen Sie Y.

9.4.1 Show seccomp filter cache status in /proc/pid/seccomp_cache

CONFIG_SECCOMP_CACHE_DEBUG [=n] **[N]**
Dies ermöglicht der Schnittstelle /proc/pid/seccomp_cache die Überwachung der seccomp-Cache-Daten. Das Dateiformat kann sich ändern. Zum Lesen der Datei ist CAP_SYS_ADMIN erforderlich. Diese Option ist nur zur Fehlersuche gedacht. Die Aktivierung birgt das Risiko, dass ein Angreifer die seccomp-Filterlogik ableiten kann.
Wenn Sie unsicher sind, sagen Sie N.

9.5 Stack Protector buffer overflow detection

CONFIG_STACKPROTECTOR [=y] **[Y]**
Diese Option schaltet die GCC-Funktion „stack-protector“ ein. Diese Funktion legt am Anfang von Funktionen einen Canary-Wert (Kanarienvogelwert) auf den Stack kurz vor der Rücksprungadresse und überprüft den Wert kurz vor der eigentlichen Rückkehr. Stack-basierte Pufferüberläufe (die diese Rücksprungadresse überschreiben müssen) überschreiben nun auch den Canary-Wert, was erkannt

wird und der Angriff wird dann durch eine Kernel-Panik neutralisiert. Bei Funktionen, die ein 8-Byte- oder größeres Zeichenarray auf dem Stack haben, wird die Logik des Stack-Protector-Canarys hinzugefügt. Diese Funktion erfordert gcc Version 4.2 oder höher, oder eine gcc-Distribution, die die Funktion zurückportiert hat (`-fstack-protector`). Auf einem x86-„defconfig"-Build fügt diese Funktion Canary-Prüfungen zu etwa 3% aller Kernel-Funktionen hinzu, was die Kernel-Codegröße um etwa 0,3% erhöht.

9.5.1 Strong Stack Protector

CONFIG_STACKPROTECTOR_STRONG [=y] **[Y]**
Bei Funktionen wird die Stack-Protector-Canary-Logik unter einer der folgenden Bedingungen hinzugefügt:

- die Adresse einer lokalen Variablen wird als Teil der rechten Seite einer Zuweisung oder eines Funktionsarguments verwendet
- die lokale Variable ist ein Array (oder eine Union, die ein Array enthält), unabhängig von Typ oder Länge des Arrays
- Lokale Variablen werden als Register verwendet

Diese Funktion erfordert gcc Version 4.9 oder höher, oder eine gcc-Distribution, die die Funktion zurückportiert hat (`-fstack-protector-strong`). Auf einem x86-„defconfig"-Build fügt diese Funktion Canary-Prüfungen zu etwa 20% aller Kernel-Funktionen hinzu, was die Größe des Kernel-Codes um etwa 2% erhöht.

9.6 Link Time Optimization (LTO) () →

9.6.1 None

CONFIG_LTO_NONE [=y] **[Y]**
Erstellen Sie den Kernel normal, ohne Link Time Optimization (LTO).

9.7 Provide system calls for 32-bit time_t

CONFIG_COMPAT_32BIT_TIME [=y] **[Y]**
Dies ermöglicht die Unterstützung von 32 Bit time_t zusätzlich zur Unterstützung von 64 Bit time_t. Dies ist auf allen 32-Bit-Architekturen und 64-Bit-Architekturen als Teil der Kompatibilitäts-Syscall-Behandlung relevant.

9.8 Use a virtually-mapped stack

CONFIG_VMAP_STACK [=y] **[Y]**
Aktivieren Sie dies, wenn Sie virtuell gemappte Kernel-Stacks mit Guard Pages verwenden wollen. Dies führt dazu, dass Kernel-Stack-Überläufe sofort abgefangen werden und keine schwer zu diagnostizierende Korruption verursachen. Um dies mit Software-KASAN-Modi zu verwenden, muss die Architektur die Unterstützung von virtuellen Mappings mit echtem Schattenspeicher unterstützen und KASAN_VMALLOC muss aktiviert sein.

9.9 Support for randomizing kernel stack offset on syscall entry

CONFIG_RANDOMIZE_KSTACK_OFFSET [=y] **[Y]**
Der Kernel-Stack-Offset kann (nach pt_regs) mit etwa 5 Bits Entropie randomisiert werden, wodurch Angriffe auf Speicherbeschädigung vereitelt werden, die auf Stack-Adress-Determinismus oder auf die Offenlegung der Adressen von Cross-Syscalls angewiesen sind. Die Funktion wird über den Kernel-Boot-Parameter `randomize_kstack_offset=on/off` gesteuert und hat, wenn sie ausgeschaltet ist, aufgrund der Verwendung von statischen Verzweigungen (siehe JUMP_LABEL) keinen Overhead.
Wenn Sie unsicher sind, sagen Sie Y.

9.9.1 Default state of kernel stack offset randomization

CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT [=y] **[Y]**
Die Randomisierung des Kernel-Stack-Offsets wird durch den Kernel-Boot-Parameter `randomize_kstack_offset=on/off` gesteuert, und diese Konfiguration wählt den Standard-Boot-Status.

9.10 Locking event counts collection

CONFIG_LOCK_EVENT_COUNTS [=y] **[Y]**
Ermöglicht eine leichtgewichtige Zählung verschiedener sperrungsbezogener Ereignisse im System mit minimalen Auswirkungen auf die Leistung. Dies verringert die Wahrscheinlichkeit, dass sich das Anwendungsverhalten aufgrund von Zeitunterschieden ändert. Die Zählungen werden über debugfs gemeldet.

9.11 GCOV-based kernel profiling →

(GCOV-basierte Kernel-Profilierung)

9.11.1 Enable gcov-based kernel profiling

CONFIG_GCOV_KERNEL [=n] **[N]**
Diese Option aktiviert die gcov-basierte Code-Profilierung (z. B. für Code-Abdeckungsmessungen). Wenn Sie unsicher sind, sagen Sie N.

Geben Sie zusätzlich CONFIG_GCOV_PROFILE_ALL=y an, um Profilerstellungsdaten für den gesamten Kernel zu erhalten. Um die Profilerstellung für bestimmte Dateien oder Verzeichnisse zu aktivieren, fügen Sie eine Zeile ähnlich der folgenden in das jeweilige Makefile ein:

Für eine einzelne Datei (z. B. main.o):

```
GCOV_PROFILE_main.o := y
```

Für alle Dateien in einem Verzeichnis:

```
GCOV_PROFILE := y
```

Um Dateien von der Profilerstellung auszuschließen, auch wenn CONFIG_GCOV_PROFILE_ALL angegeben ist, verwenden Sie:

```
GCOV_PROFILE_main.o := n
```

und:

```
GCOV_PROFILE := n
```

Beachten Sie, dass das debugfs-Dateisystem gemountet sein muss, um auf die Profilerstellungsdaten zugreifen zu können.

9.12 GCC plugins →

CONFIG_GCC_PLUGINS [=y] **[Y]**
GCC-Plugins sind ladbare Module, die zusätzliche Funktionen für den Compiler bereitstellen. Sie sind nützlich für die Laufzeitinstrumentierung und die statische Analyse.
Siehe Documentation/kbuild/gcc-plugins.rst für Details.

9.12.1 Generate some entropy during boot and runtime

CONFIG_GCC_PLUGIN_LATENT_ENTROPY [=n] **[N]**
Mit der Eingabe von Y wird der Kernel einen Teil des Kernel-Codes instrumentieren, um sowohl aus dem ursprünglichen als auch aus dem künstlich erzeugten Programmzustand etwas Entropie zu gewinnen. Dies ist insbesondere bei eingebetteten Systemen hilfreich, bei denen es normalerweise wenig „natürliche“ Entropiequellen gibt. Der Preis ist eine gewisse Verlangsamung des Boot-Prozesses (etwa 0,5 %) und der fork- und irq-Verarbeitung. Beachten Sie, dass die auf diese Weise extrahierte Entropie nicht kryptografisch sicher ist!
Dieses Plugin wurde von grsecurity/PaX portiert. Mehr Informationen unter:

```
https://grsecurity.net/

https://pax.grsecurity.net/
```

10 Enable loadable module support →

CONFIG_MODULES [=y] **[Y]**
Kernel-Module sind kleine Stücke kompilierten Codes, die in den laufenden Kernel eingefügt werden

können, anstatt dauerhaft in den Kernel eingebaut zu werden. Sie verwenden das Werkzeug `modprobe`, um sie hinzuzufügen (und manchmal zu entfernen).
Wenn Sie hier Y angeben, können viele Teile des Kernels als Module gebaut werden (indem Sie M anstelle von Y antworten, wo dies angegeben ist):
Dies ist besonders nützlich für selten verwendete Optionen, die zum Booten nicht benötigt werden. Weitere Informationen finden Sie in den Man Pages für `modprobe`, `lsmod`, `modinfo`, `insmod` und `rmmod`.
Wenn Sie hier Y angeben, müssen Sie `make modules_install` ausführen, um die Module unter /lib/modules/ abzulegen, wo sie von modprobe gefunden werden können (möglicherweise müssen Sie dazu root sein). Wenn Sie unsicher sind, sagen Sie Y.

10.1 Module debugging

CONFIG_MODULE_DEBUG [=n] [**N**]
Ermöglicht das Aktivieren/Deaktivieren von Funktionen, die Ihnen beim Debuggen von Modulen helfen können. Auf Produktionssystemen benötigen Sie diese Optionen nicht.

10.2 Forced module loading

CONFIG_MODULE_FORCE_LOAD [=y] [**Y**]
Erlaubt das Laden von Modulen ohne Versionsinformationen (z. B. `modprobe --force`). Erzwungenes Laden von Modulen setzt das 'F' (forced) taint Flag und ist normalerweise eine wirklich schlechte Idee.

10.3 Module unloading

CONFIG_MODULE_UNLOAD [=y] [**Y**]
Ohne diese Option können Sie keine Module entladen (beachten Sie, dass einige Module möglicherweise ohnehin nicht entladbar sind), was Ihren Kernel kleiner, schneller und einfacher macht. Wenn Sie unsicher sind, sagen Sie Y.

10.3.1 Forced module unloading

CONFIG_MODULE_FORCE_UNLOAD [=y] [**Y**]
Mit dieser Option können Sie das Entladen eines Moduls erzwingen, auch wenn der Kernel es für unsicher hält: Der Kernel wird das Modul entfernen, ohne darauf zu warten, dass jemand die Verwendung des Moduls beendet (mit der Option `-f` von `rmmod`). Dies ist hauptsächlich für Kernel-Entwickler und verzweifelte Benutzer gedacht. Wenn Sie unsicher sind, sagen Sie N.

10.3.2 Tainted module unload tracking

CONFIG_MODULE_UNLOAD_TAINT_TRACKING [=y] [**Y**]
Mit dieser Option können Sie eine Aufzeichnung über jedes entladene Modul führen, das den Kernel beschädigt hat. Zusätzlich zur Anzeige einer Liste der verknüpften (oder geladenen) Module, z. B. bei der Erkennung einer schlechten Seite (siehe bad_page()), werden auch die oben genannten Details angezeigt. Wenn Sie unsicher sind, sagen Sie N.

10.4 Module versioning support

CONFIG_MODVERSIONS [=n] [**N**]
Normalerweise müssen Sie Module verwenden, die mit Ihrem Kernel kompiliert wurden. Wenn Sie hier Y angeben, ist es manchmal möglich, Module zu verwenden, die für andere Kernel kompiliert wurden, indem Sie genügend Informationen zu den Modulen hinzufügen, um (hoffentlich) alle Änderungen zu erkennen, die sie mit dem von Ihnen verwendeten Kernel inkompatibel machen würden.
Wenn Sie unsicher sind, sagen Sie N.

10.5 Source checksum for all modules

CONFIG_MODULE_SRCVERSION_ALL [=y] [**Y**]
Module, die eine MODULE_VERSION enthalten, bekommen ein zusätzliches „srcversion"-Feld in ihre modinfo-Sektion eingefügt, das eine Summe der Quelldateien enthält, aus denen sie entstanden sind. Dies hilft den Betreuern, genau zu sehen, welche Quelle verwendet wurde, um ein Modul zu bauen (da andere

manchmal die Modulquelle ändern, ohne die Version zu aktualisieren). Mit dieser Option wird ein solches „srcversion“-Feld für alle Module erstellt. Wenn Sie unsicher sind, sagen Sie N.

10.6 Module signature verification

CONFIG_MODULE_SIG [=y] **[Y]**
Überprüfung von Modulen auf gültige Signaturen beim Laden: Die Signatur wird einfach an das Modul angehängt. Für weitere Informationen siehe <file:Documentation/admin-guide/module-signing.rst>. Beachten Sie, dass diese Option die OpenSSL-Entwicklungspakete als Kernel-Build-Abhängigkeit hinzufügt, so dass das Signierwerkzeug seine Krypto-Bibliothek verwenden kann. Sie sollten diese Option aktivieren, wenn Sie entweder CONFIG_SECURITY_LOCKDOWN_LSM oder eine durch eine andere LSM auferlegte Lockdown-Funktionalität verwenden wollen – andernfalls werden unsignierte Module unabhängig von der Lockdown-Policy ladbar sein. !!!WARNUNG!!! Wenn Sie diese Option aktivieren, MÜSSEN Sie sicherstellen, dass das Modul nach dem Signieren NICHT gestrippt wird. Dies schließt den Debuginfo-Strip ein, der von einigen Paketierern (wie z. B. rpmbuild) durchgeführt wird, sowie die Einbindung in ein initramfs, das die Modulgröße reduzieren möchte.

10.6.1 Require modules to be validly signed

CONFIG_MODULE_SIG_FORCE [=n] **[N]**
Ablehnung von unsignierten Modulen oder signierten Modulen, für die wir keinen Schlüssel haben. Ohne diesen Schlüssel werden solche Module den Kernel einfach verunreinigen.

10.6.2 Automatically sign all modules

CONFIG_MODULE_SIG_ALL [=y] **[Y]**
Signiere alle Module während make modules_install. Ohne diese Option müssen die Module manuell signiert werden, und zwar mit dem Werkzeug scripts/sign-file.

10.7 Which hash algorithm should modules be signed with? () →

Damit wird festgelegt, welche Art von Hashing-Algorithmus bei der Signaturerstellung verwendet wird. Dieser Algorithmus **muss** direkt in den Kernel eingebaut werden, damit eine Signaturprüfung stattfinden kann. Es ist nicht möglich, ein signiertes Modul zu laden, das den Algorithmus enthält, um die Signatur dieses Moduls zu überprüfen.

10.7.1 Sign modules with SHA-1

CONFIG_MODULE_SIG_SHA1 [=n] **[N]**
Für diese Option gibt es keine Hilfe.

10.7.2 Sign modules with SHA-224

CONFIG_MODULE_SIG_SHA224 [=n] **[N]**
Für diese Option gibt es keine Hilfe.

10.7.3 Sign modules with SHA-256

CONFIG_MODULE_SIG_SHA256 [=n] **[N]**
Für diese Option gibt es keine Hilfe.

10.7.4 Sign modules with SHA-384

CONFIG_MODULE_SIG_SHA384 [=n] **[N]**
Für diese Option gibt es keine Hilfe.

10.7.5 Sign modules with SHA-512

CONFIG_MODULE_SIG_SHA512 [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

10.8 Module compression mode

Mit dieser Option können Sie den Algorithmus auswählen, der zur Komprimierung von Modulen verwendet wird, wenn `make modules_install` ausgeführt wird. (oder Sie können wählen, dass Module überhaupt nicht komprimiert werden.) Externe Module werden während der Installation ebenfalls auf die gleiche Weise komprimiert. Für Module innerhalb einer initrd oder initramfs ist es effizienter, stattdessen die gesamte initrd oder initramfs zu komprimieren. Dies ist vollständig kompatibel mit signierten Modulen. Bitte beachten Sie, dass das zum Laden von Modulen verwendete Werkzeug den entsprechenden Algorithmus unterstützen muss. module-init-tools KANN gzip unterstützen, und kmod KANN gzip, xz und zstd unterstützen. Ihr Build-System muss das entsprechende Komprimierungswerkzeug bereitstellen, um die Module zu komprimieren. Im Zweifelsfall wählen Sie 'None'.

10.8.1 None

CONFIG_MODULE_COMPRESSION_NONE [=n] **[N]**
Komprimieren Sie die Module nicht. Die installierten Module sind mit der Endung .ko versehen.

10.8.2 GZIP

CONFIG_MODULE_COMPRESSION_GZIP [=n] **[N]**
Komprimieren Sie Module mit GZIP. Die installierten Module sind mit der Endung .ko.gz versehen.

10.8.3 XZ

CONFIG_MODULE_COMPRESSION_XZ [=n] **[N]**
Komprimieren Sie Module mit XZ. Die installierten Module sind mit der Endung .ko.xz versehen.

10.8.4 ZSTD

CONFIG_MODULE_COMPRESS_ZSTD [=y] **[Y]**
Komprimieren Sie Module mit ZSTD. Die installierten Module sind mit der Endung .ko.zst versehen.

10.9 Support in-kernel module decompression

CONFIG_MODULE_DECOMPRESS [=y] **[Y]**
Unterstützung für die Dekomprimierung von Kernelmodulen durch den Kernel selbst, anstatt sich auf den Userspace zu verlassen, um diese Aufgabe zu erledigen. Nützlich, wenn die Sicherheitsrichtlinie für das Load Pinning aktiviert ist. Wenn Sie unsicher sind, sagen Sie N.

10.10 Allow loading of modules with missing namespace imports

CONFIG_MODULE_ALLOW_MISSING_NAMESPACE_IMPORTS [=y] **[Y]**
Symbole, die mit EXPORT_SYMBOL_NS*() exportiert werden, gelten als in einem Namespace exportiert. Ein Modul, das ein Symbol verwendet, das mit einem solchen Namespace exportiert wurde, muss den Namespace über MODULE_IMPORT_NS() importieren. Es gibt keinen technischen Grund, korrekte Namespace-Importe zu erzwingen, aber es schafft Konsistenz zwischen Symbolen, die Namespaces definieren und Benutzern, die Namespaces importieren, die sie verwenden. Diese Option lockert diese Anforderung und hebt die Durchsetzung beim Laden eines Moduls auf. Wenn Sie unsicher sind, sagen Sie N.

10.11 Path to modprobe binary

CONFIG_MODPROBE_PATH [=/sbin/modprobe] **[/sbin/modprobe]**
Wenn der Kernel-Code ein Modul anfordert, geschieht dies durch den Aufruf des Userspace-Dienstprogramms `modprobe`. Mit dieser Option können Sie den Pfad festlegen, in dem diese Binärdatei zu finden ist. Dies kann zur Laufzeit über die sysctl-Datei /proc/sys/kernel/modprobe geändert werden. Wenn Sie diese Option auf eine leere Zeichenkette setzen, wird die Fähigkeit des Kernels, Module anzufordern, ausgeschaltet (der Userspace kann jedoch weiterhin explizit Module laden).

11 Enable the block layer →

CONFIG_BLOCK [=y] [**Y**]
Bietet dem Kernel Unterstützung für die Blockschicht.
Deaktivieren Sie diese Option, um die Unterstützung für die Blockschicht aus dem Kernel zu entfernen. Dies kann für eingebettete Geräte nützlich sein.
Wenn diese Option deaktiviert ist:

- werden Blockgerätedateien unbrauchbar,

- werden einige Dateisysteme (wie ext3) nicht mehr verfügbar.

Außerdem werden SCSI-Zeichengeräte und USB-Speicher deaktiviert, da sie verschiedene Definitionen und Möglichkeiten der Blockschicht nutzen.
Sagen Sie hier "Y", es sei denn, Sie wissen, dass Sie wirklich keine Festplatten und dergleichen einbinden wollen.

11.1 Legacy autoloading support

CONFIG_BLOCK_LEGACY_AUTOLOAD [=y] [**Y**]
Ermöglicht das Laden von Modulen und das Erstellen von Block-Geräteinstanzen auf der Grundlage von Zugriffen durch ihre spezielle Gerätedatei. Dies ist ein historisches Linux-Feature und ergibt in einer udev-Welt, in der Gerätedateien bei Bedarf erstellt werden, keinen Sinn, aber Skripte, die manuell Geräteknoten erstellen und dann losetup aufrufen, könnten sich auf dieses Verhalten verlassen.

11.2 Block layer SG support v4 helper lib

CONFIG_BLK_DEV_BSGLIB [=y] [**Y**]
Die Teilsysteme werden dies normalerweise bei Bedarf aktivieren. Die Benutzer müssen dies normalerweise nicht manuell aktivieren. Wenn Sie unsicher sind, sagen Sie N.

11.3 Block layer data integrity support

CONFIG_BLK_DEV_INTEGRITY [=y] [**Y**]
Einige Speichermedien erlauben die Speicherung/Abrufung zusätzlicher Informationen, um die Daten zu schützen. Die Option für die Datenintegrität auf Blockebene bietet Hooks, die von Dateisystemen verwendet werden können, um eine bessere Datenintegrität zu gewährleisten. Sagen Sie hier Ja, wenn Sie ein Speichergerät haben, das das T10/SCSI Data Integrity Field oder den T13/ATA External Path Protection bietet. Im Zweifelsfall sagen Sie N.

11.4 Zoned block device support

CONFIG_BLK_DEV_ZONED [=y] [**Y**]
Unterstützung von Blockebenen mit Zonen für Blockgeräte. Diese Option aktiviert die Unterstützung für ZAC/ZBC/ZNS Host-verwaltete und Host-bewusste Zoned-Block-Geräte. Sagen Sie hier Ja, wenn Sie ein ZAC-, ZBC- oder ZNS-Speichergerät haben.

11.5 Block layer bio throttling support

CONFIG_BLK_DEV_THROTTLING [=y] [**Y**]
Unterstützung der Bio-Drosselung auf der Blockschicht. Sie kann verwendet werden, um die IO-Rate für ein Gerät zu begrenzen. IO-Rate-Policies sind pro cgroup und man muss blkio cgroup controller mounten und verwenden, um cgroups zu erstellen und IO-Rate-Policies pro Gerät festzulegen.
Siehe Documentation/admin-guide/cgroup-v1/blkio-controller.rst für weitere Informationen.

11.5.1 Block throttling .low limit interface support (EXPERIMENTAL)

CONFIG_BLK_DEV_THROTTLING_LOW [=y] [**Y**]
Hinzufügen der Schnittstelle .low limit für die Blockdrosselung. Das niedrige Limit ist ein Best-Effort-Limit zur Priorisierung von C-Gruppen. Je nach Einstellung kann das Limit verwendet werden, um C-Gruppen in Bezug auf Bandbreite/iops zu schützen und die Festplattenressourcen besser zu nutzen.

Beachten Sie, dass es sich hierbei um eine experimentelle Schnittstelle handelt, die eines Tages geändert werden könnte.

11.6 Enable support for block device writeback throttling

CONFIG_BLK_WBT [=y] **[Y]**
Die Aktivierung dieser Option ermöglicht es der Blockschicht, gepufferte Hintergrund-Schreibvorgänge der VM zu drosseln, so dass diese reibungsloser ablaufen und weniger Auswirkungen auf die Vordergrundvorgänge haben. Die Drosselung erfolgt dynamisch nach einem Algorithmus, der lose auf CoDel basiert und die Echtzeitleistung der Festplatte berücksichtigt.

11.6.1 Enable writeback throttling by default

CONFIG_BLK_WBT_MQ [=y] **[Y]**
Aktivieren Sie die Rückschreibdrosselung standardmäßig für anforderungsbasierte Blockgeräte.

11.7 Enable support for latency based cgroup IO protection

CONFIG_BLK_CGROUP_IOLATENCY [=y] **[Y]**
Durch die Aktivierung dieser Option wird die .latency-Schnittstelle für die IO-Drosselung aktiviert. Der IO-Controller versucht, die durchschnittlichen IO-Latenzen unter dem konfigurierten Latenzziel zu halten und drosselt jeden mit einem höheren Latenzziel als die betroffene Gruppe.
Beachten Sie, dass es sich hierbei um eine experimentelle Schnittstelle handelt, die eines Tages geändert werden könnte.

11.8 Enable support to track FC I/O Traffic across cgroup applications

CONFIG_BLK_CGROUP_FC_APPID [=y] **[Y]**
Die Aktivierung dieser Option ermöglicht die Verfolgung des FC-I/O-Verkehrs über cgroup-Anwendungen hinweg. Sie ermöglicht es der Fabric und den Speicherzielen, den FC-Verkehr auf der Grundlage von VM-Tags zu identifizieren, zu überwachen und zu verarbeiten, indem eine anwendungsspezifische Identifikation in den FC-Frame eingefügt wird.

11.9 Enable support for cost model based cgroup IO controller

CONFIG_BLK_CGROUP_IOCOST [=y] **[Y]**
Durch Aktivieren dieser Option wird die .weight-Schnittstelle für die kostenmodellbasierte proportionale IO-Steuerung aktiviert. Der IO-Controller verteilt die IO-Kapazität zwischen verschiedenen Gruppen auf der Grundlage ihres Anteils an der Gesamtgewichtsverteilung.

11.10 Cgroup I/O controller for assigning an I/O priority class

CONFIG_BLK_CGROUP_IOPRIO [=y] **[Y]**
Aktivieren Sie die Schnittstelle .prio, um Anfragen eine E/A-Prioritätsklasse zuzuweisen. Die E/A-Prioritätsklasse beeinflusst die Reihenfolge, in der ein E/A-Scheduler und Blockgeräte Anforderungen verarbeiten. Nur einige E/A-Scheduler und einige Blockgeräte unterstützen E/A-Prioritäten.

11.11 Block layer debugging information in debugfs

CONFIG_BLK_DEBUG_FS [=y] **[Y]**
Aufnahme von Debugging-Informationen der Blockschicht in debugfs. Diese Informationen sind vor allem für Kernel-Entwickler nützlich, aber sie verursachen keine Kosten zur Laufzeit. Wenn Sie nicht gerade einen Kernel für ein winziges System bauen, sollten Sie hier Y für Ja sagen.

11.12 Logic for interfacing with Opal enabled SEDs

CONFIG_BLK_SED_OPAL [=y] **[Y]**
Erstellt die Logik für die Verbindung mit Opal-fähigen Steuergeräten. Die Aktivierung dieser Option ermöglicht es Benutzern, Sperrbereiche für SED-Geräte mit dem Opal-Protokoll einzurichten/zu entsperren/zu sperren.

11.13 Enable inline encryption support in block layer

CONFIG_BLK_INLINE_ENCRYPTION [=y] **[Y]**
Bauen Sie das blk-crypto-Subsystem auf. Wenn Sie dies aktivieren, kann die Blockschicht die Verschlüsselung handhaben, so dass Benutzer die Vorteile der Inline-Verschlüsselungshardware nutzen können, falls vorhanden.

11.13.1 Enable crypto API fallback for blk-crypto

CONFIG_BLK_INLINE_ENCRYPTION_FALLBACK [=y] **[Y]**
Wenn dies aktiviert ist, kann die Blockschicht die Inline-Verschlüsselung handhaben, indem sie auf die Kernel-Krypto-API zurückgreift, wenn keine Inline-Verschlüsselungshardware vorhanden ist.

11.14 Partition Types →

(Partitionstypen)

11.14.1 Advanced partition selection

CONFIG_PARTITION_ADVANCED [=y] **[Y]**
Geben Sie hier Y ein, wenn Sie unter Linux Festplatten verwenden möchten, die unter einem Betriebssystem partitioniert wurden, das auf einer anderen Architektur als Ihr Linux-System läuft.
Beachten Sie, dass sich die Antwort auf diese Frage nicht direkt auf den Kernel auswirkt: Wenn Sie N angeben, überspringt der Konfigurator lediglich alle Fragen zu fremden Partitionierungsschemata. Wenn Sie unsicher sind, sagen Sie N.

11.14.1.1 Acorn partition support

CONFIG_ACORN_PARTITION [=n] **[N]**
Unterstützung von Festplatten, die unter Acorn-Betriebssystemen partitioniert sind.

11.14.1.2 AIX basic partition table support

CONFIG_AIX_PARTITION [=y] **[Y]**
Geben Sie hier Y ein, wenn Sie das Format der Festplattenpartitionstabelle lesen möchten, das von IBM- oder Motorola-PowerPC-Maschinen unter AIX verwendet wird. AIX verwendet eigentlich einen Logical Volume Manager, bei dem „logische Volumes“ über eine oder mehrere Festplatten verteilt sein können, aber dieser Treiber funktioniert nur für den einfachen Fall von zusammenhängenden Partitionen. Andernfalls, sagen wir N.

11.14.1.3 Alpha OSF partition support

CONFIG_OSF_PARTITION [=n] **[N]**
Geben Sie hier Y an, wenn Sie unter Linux Festplatten verwenden möchten, die auf einer Alpha-Maschine partitioniert wurden.

11.14.1.4 Amiga partition table support

CONFIG_AMIGA_PARTITION [=n] **[N]**
Sagen Sie hier Y, wenn Sie unter Linux Festplatten verwenden möchten, die unter AmigaOS partitioniert wurden.

11.14.1.5 Atari partition table support

CONFIG_ATARI_PARTITION [=n] **[N]**
Sagen Sie hier Y, wenn Sie unter Linux Festplatten verwenden möchten, die unter dem Atari-Betriebssystem partitioniert wurden.

11.14.1.6 Macintosh partition map support

CONFIG_MAC_PARTITION [=y] **[Y]**
Sagen Sie hier Y, wenn Sie unter Linux Festplatten verwenden möchten, die auf einem Macintosh partitioniert wurden.

11.14.1.7 PC BIOS (MSDOS partition tables) support

CONFIG_MSDOS_PARTITION [=y] **[Y]**
Sagen Sie hier Y.

11.14.1.7.1 BSD disklabel (FreeBSD partition tables) support

CONFIG_BSD_DISKLABEL [=y] **[Y]**
FreeBSD verwendet ein eigenes Partitionsschema für die Festplatten Ihres PCs. Es benötigt nur einen Eintrag in der primären Partitionstabelle Ihrer Festplatte und verwaltet diese ähnlich wie erweiterte DOS-Partitionen, indem es im ersten Sektor eine neue Partitionstabelle im BSD-Disklabel-Format anlegt. Wenn Sie hier Y angeben, können Sie diese Disklabels lesen und FreeBSD-Partitionen von Linux aus einbinden, wenn Sie oben bei „UFS file system support“ ebenfalls Y angegeben haben. Wenn Sie nicht wissen, was das alles soll, sagen Sie N.

11.14.1.7.2 Minix subpartition support

CONFIG_MINIX_SUBPARTITION [=y] **[Y]**
Unterstützung von Minix 2.0.0/2.0.2 Subpartitionstabellen für Linux. Sagen Sie hier Y, wenn Sie Minix 2.0.0/2.0.2 Subpartitionen mounten und verwenden wollen.

11.14.1.7.3 Solaris (x86) partition table support

CONFIG_SOLARIS_X86_PARTITION [=y] **[Y]**
Wie die meisten Systeme verwendet Solaris x86 ein eigenes Festplattenpartitionstabellenformat, das mit allen anderen nicht kompatibel ist. Wenn Sie hier Y angeben, können Sie diese Partitionstabellen lesen und Solaris x86-Partitionen von Linux aus einbinden, wenn Sie oben bei „UFS-Dateisystemunterstützung“ ebenfalls Y angegeben haben.

11.14.1.7.4 Unixware slices support

CONFIG_UNIXWARE_DISKLABEL [=n] **[N]**
Wie einige Systeme verwendet auch UnixWare eine eigene Slice-Tabelle innerhalb einer Partition (VTOC – Virtual Table of Contents). Ihr Format ist mit allen anderen Betriebssystemen nicht kompatibel. Wenn Sie hier Y angeben, können Sie VTOC lesen und UnixWare-Partitionen von Linux aus schreibgeschützt einbinden, wenn Sie oben auch Y zu „UFS-Dateisystemunterstützung“ oder „System V und Coherent-Dateisystemunterstützung“ angegeben haben. Dies wird hauptsächlich verwendet, um Daten von einem UnixWare-Rechner auf Ihren Linux-Rechner zu übertragen, und zwar über ein Wechselmedium wie magneto-optische, ZIP- oder IDE-Wechselplatten. Beachten Sie jedoch, dass das Programm `tar` (`man tar` oder vorzugsweise `info tar`) eine gute Möglichkeit bietet, Dateien und Verzeichnisse zwischen Unixen (und sogar anderen Betriebssystemen) zu transportieren. Wenn Sie nicht wissen, was das alles soll, sagen Sie N.

11.14.1.8 Windows Logical Disk Manager (Dynamic Disk) support

CONFIG_LDM_PARTITION [=y] **[Y]**
Sagen Sie hier Y, wenn Sie unter Linux Festplatten verwenden möchten, die mit dem Logical Disk Manager von Windows 2000/XP oder Vista partitioniert wurden. Sie werden auch als „dynamische Festplatten“ bezeichnet.
Beachten Sie, dass dieser Treiber nur dynamische Festplatten mit einem schützenden MBR-Label, d.h. einer DOS-Partitionstabelle, unterstützt. Dynamische Festplatten mit GPT-Label, wie sie mit Vista erstellt werden können, werden noch nicht unterstützt. Windows 2000 führte das Konzept der dynamischen Festplatten ein, um die Einschränkungen des PC-Partitionierungsschemata zu umgehen. Der Logical Disk Manager ermöglicht es dem Benutzer, eine Festplatte neu zu partitionieren und übergreifende, gespiegelte, striped oder RAID-Volumes zu erstellen, ohne dass ein Neustart erforderlich ist. Normale Partitionen werden nun unter Windows 2000, XP und Vista als Basisfestplatten bezeichnet.
Für eine ausführlichere Beschreibung lesen Sie <file:Documentation/admin-guide/ldm.rst>.
Wenn Sie unsicher sind, sagen Sie N.

11.14.1.8.1 Windows LDM extra logging

CONFIG_LDM_DEBUG [=n] **[N]**
Geben Sie hier Y an, wenn Sie möchten, dass LDM ausführlich protokolliert. Dies könnte hilfreich sein, wenn der Treiber nicht wie erwartet funktioniert und Sie einen Fehler melden möchten. Wenn Sie unsicher sind, sagen Sie N.

11.14.1.9 SGI partition support

CONFIG_SGI_PARTITION [=n] **[N]**
Wählen Sie hier Y, wenn Sie das von SGI-Maschinen verwendete Format der Festplattenpartitionstabelle lesen möchten.

11.14.1.10 Ultrix partition table support

CONFIG_ULTRIX_PARTITION [=n] **[N]**
Sagen Sie hier Y, wenn Sie das von DEC (jetzt Compaq) Ultrix-Maschinen verwendete Format der Festplattenpartitionstabelle lesen möchten. Andernfalls sagen Sie N.

11.14.1.11 Sun partition tables support

CONFIG_SUN_PARTITION [=n] **[N]**
Wie die meisten Systeme verwendet SunOS ein eigenes Format für Festplattenpartitionstabellen, das mit allen anderen nicht kompatibel ist. Wenn Sie hier Y angeben, können Sie diese Partitionstabellen lesen und SunOS-Partitionen von Linux aus einbinden, wenn Sie oben bei „UFS-Dateisystemunterstützung“ ebenfalls Y angegeben haben. Dies wird hauptsächlich benutzt, um Daten von einem SPARC unter SunOS zu Ihrem Linux-Rechner über ein Wechselmedium wie magneto-optische oder ZIP-Laufwerke zu transportieren; beachten Sie jedoch, dass ein guter portabler Weg, Dateien und Verzeichnisse zwischen Unixen (und sogar anderen Betriebssystemen) zu transportieren, durch das `tar`-Programm (`man tar` oder vorzugsweise `info tar`) gegeben ist. Wenn Sie nicht wissen, was das alles soll, sagen Sie N.

11.14.1.12 Karma Partition support

CONFIG_KARMA_PARTITION [=y] **[Y]**
Sagen Sie hier Y, wenn Sie den Rio Karma MP3-Player mounten möchten, da dieser eine proprietäre Partitionstabelle verwendet.

11.14.1.13 EFI GUID Partition support

CONFIG_EFI_PARTITION [=y] **[Y]**
Geben Sie hier Y an, wenn Sie unter Linux Festplatten verwenden möchten, die mit EFI GPT partitioniert wurden.

11.14.1.14 SYSV68 partition table support

CONFIG_SYSV68_PARTITION [=n] **[N]**
Geben Sie hier Y ein, wenn Sie das von Motorola-Delta-Maschinen verwendete Format der Festplattenpartitionstabelle lesen möchten (unter Verwendung von sysv68). Andernfalls sagen Sie N.

11.14.1.15 Command line partition support

CONFIG_CMDLINE_PARTITION [=n] **[N]**
Sagen Sie hier Y, wenn Sie die Partitionstabelle aus bootargs lesen wollen. Das Format für die Kommandozeile ist genau wie bei mtdparts.

11.15 IO Schedulers →

(E/A-Zeitplaner)

11.15.1 MQ deadline I/O scheduler

CONFIG_MQ_IOSCHED_DEADLINE [=y] **[Y]**
MQ-Version des Deadline-IO-Schedulers.

11.15.2 Kyber I/O scheduler

CONFIG_MQ_IOSCHED_KYBER [=m] **[M]**
Der Kyber E/A-Scheduler ist ein Scheduler mit geringem Aufwand, der sich für Multiqueue- und andere schnelle Geräte eignet. Bei vorgegebenen Ziellatenzen für Lese- und synchrone Schreibvorgänge passt er die Tiefe der Warteschlangen selbst an, um dieses Ziel zu erreichen.

11.15.3 BFQ I/O scheduler

CONFIG_IOSCHED_BFQ [=y] **[Y]**
BFQ E/A-Scheduler für BLK-MQ. BFQ verteilt die Bandbreite des Geräts auf alle Prozesse entsprechend ihrer Gewichtung, unabhängig von den Geräteparametern und bei jeder Arbeitslast. Es garantiert auch eine niedrige Latenzzeit für interaktive und weiche Echtzeitanwendungen.
Weitere Details in Dokumentation/block/bfq-iosched.rst

11.15.3.1 BFQ hierarchical scheduling support

CONFIG_BFQ_GROUP_IOSCHED [=y] **[Y]**
Aktivierung der hierarchischen Zeitplanung in BFQ unter Verwendung des blkio (cgroupss-v1) oder io (cgroupss-v2) Controllers.

11.15.3.1.1 BFQ IO controller debugging

CONFIG_BFQ_CGROUP_DEBUG [=n] **[N]**
Aktivierung einer Hilfe zur Fehlersuche. Derzeit werden zusätzliche Statistikdateien in einer cgroup exportiert, die für die Fehlersuche nützlich sein können.

12 Executable file formats →

(Ausführbare Dateiformate)

12.1 Kernel support for ELF binaries

CONFIG_BINFMT_ELF [=y] **[Y]**
ELF (Executable and Linkable Format) ist ein Format für Bibliotheken und ausführbare Dateien, das auf verschiedenen Architekturen und Betriebssystemen verwendet wird. Wenn Sie hier Y angeben, kann Ihr Kernel ELF-Binärdateien ausführen und wird um etwa 13 KB vergrößert. Die ELF-Unterstützung unter Linux hat inzwischen die traditionellen Linux a.out-Formate (QMAGIC und ZMAGIC) fast vollständig ersetzt, da es portabel ist (was jedoch *nicht* bedeutet, dass Sie ausführbare Dateien von verschiedenen Architekturen oder Betriebssystemen ausführen können) und die Erstellung von Laufzeitbibliotheken sehr einfach macht. Viele neue ausführbare Dateien werden ausschließlich im ELF-Format vertrieben. Hier sollten Sie unbedingt Y sagen. Informationen über ELF sind im ELF HOWTO enthalten, das unter `http://www.tldp.org/docs.html#howto` verfügbar ist. Wenn Sie feststellen, dass Sie nach dem Upgrade von Linux-Kernel 1.2 und der Angabe von Y hier immer noch keine ELF-Binärdateien ausführen können (sie stürzen einfach ab), dann müssen Sie die neuesten ELF-Laufzeitbibliotheken installieren, einschließlich `ld.so` (überprüfen Sie die Datei <file:Documentation/Changes> für den Ort und die neueste Version).

12.2 Write ELF core dumps with partial segments

CONFIG_CORE_DUMP_DEFAULT_ELF_HEADERS [=y] **[Y]**
ELF-Core-Dump-Dateien beschreiben jede Speicherabbildung des abgestürzten Prozesses und können den Speicherinhalt jedes einzelnen Prozesses enthalten oder auslassen. Der Inhalt eines unveränderten Text-Mappings wird standardmäßig weggelassen. Bei einem unveränderten Text-Mapping eines ELF-Objekts ermöglicht die Aufnahme nur der ersten Seite der Datei in einen Core-Dump die Identifizierung der Build-ID-Bits in der Datei, ohne dass die E/A-Kosten und der Plattenplatz für das Dump des gesamten Textes anfallen. Versionen von GDB vor 6.7 werden jedoch von ELF-Core-Dump-Dateien in diesem Format verwirrt. Das Verhalten des Kerndumps kann pro Prozess mit der Pseudodatei /proc/PID/coredump_filter gesteuert werden; diese Einstellung wird vererbt. Siehe Dokumentation/filesystems/proc.rst für Details. Diese Konfigurationsoption ändert die Standardeinstellung von coredump_filter, die beim Booten zu sehen ist. Wenn Sie unsicher sind, sagen Sie Y.

12.3 Kernel support for scripts starting with #!

CONFIG_BINFMT_SCRIPT [=y] **[Y]**
(Kernel-Unterstützung für Skripte, die mit #!, dem Shebang, anfangen) Geben Sie hier Y an, wenn Sie interpretierte Skripte ausführen wollen, die mit #! beginnen, gefolgt von dem Pfad zu einem Interpreter. Sie können diese Unterstützung als Modul bauen; bis dieses Modul jedoch geladen ist, können Sie keine

Skripte ausführen. Wenn Sie also dieses Modul aus einem initramfs laden wollen, darf der Teil des initramfs vor dem Laden dieses Moduls nur aus kompilierten Binärdateien bestehen. Die meisten Systeme werden nicht booten, wenn Sie hier M oder N angeben. Wenn Sie unsicher sind, sagen Sie Y.

12.4 Kernel support for MISC binaries

CONFIG_BINFMT_MISC [=y] **[Y]**
Wenn Sie hier Y sagen, wird es möglich sein, Wrapper-gesteuerte Binärformate in den Kernel einzubinden. Dies ist vor allem dann sinnvoll, wenn Sie Programme verwenden, die einen Interpreter benötigen, wie Java, Python, .NET oder Emacs-Lisp. Es ist auch nützlich, wenn Sie häufig DOS-Programme unter dem Linux-DOS-Emulator DOSEMU ausführen (lesen Sie das DOSEMU-HOWTO, verfügbar unter `http://www.tldp.org/docs.html#howto`). Sobald Sie eine solche Binärklasse beim Kernel registriert haben, können Sie eines dieser Programme einfach starten, indem Sie seinen Namen an einer Shell-Eingabeaufforderung eingeben; Linux wird es automatisch an den richtigen Interpreter weiterleiten. Sie können auch andere nette Dinge tun.
Lesen Sie die Datei <file:Documentation/admin-guide/binfmt-misc.rst>, um zu erfahren, wie Sie diese Funktion nutzen können, <file:Documentation/admin-guide/java.rst>, um zu erfahren, wie Sie Java-Unterstützung einbinden können, und <file:Documentation/admin-guide/mono.rst>, um zu erfahren, wie Sie Mono-basierte .NET-Unterstützung einbinden können. Um binfmt_misc zu verwenden, müssen Sie es mounten: `mount binfmt_misc -t binfmt_misc /proc/sys/fs/binfmt_misc` Sie können hier M für Modulunterstützung sagen und später das Modul laden, wenn Sie es brauchen; das Modul heißt `binfmt_misc`. Wenn Sie nicht wissen, was Sie an dieser Stelle antworten sollen, sagen Sie Y.

13 Memory Management options →

(Speicherverwaltungsoptionen)

13.1 Support for paging of anonymous memory (swap) →

CONFIG_SWAP [=y] **[Y]**
Mit dieser Option können Sie wählen, ob Sie Unterstützung für so genannte Swap-Geräte oder Swap-Dateien in Ihrem Kernel haben möchten, die dazu dienen, mehr virtuellen Speicher als den tatsächlichen Arbeitsspeicher in Ihrem Computer bereitzustellen. Wenn Sie unsicher sind, sagen Sie Y.

13.1.1 Compressed cache for swap pages

CONFIG_ZSWAP [=y] **[Y]**
Ein leichtgewichtiger komprimierter Cache für Auslagerungsseiten. Er nimmt Seiten, die gerade ausgelagert werden, und versucht, sie in einem dynamisch zugewiesenen RAM-basierten Speicherpool zu komprimieren. Dies kann zu einer erheblichen E/A-Reduzierung auf dem Swap-Gerät führen und in dem Fall, in dem die Dekomprimierung aus dem RAM schneller ist als das Lesen aus dem Swap-Gerät, auch die Arbeitslastleistung verbessern.

13.1.1.1 Enable the compressed cache for swap pages by default

CONFIG_ZSWAP_DEFAULT_ON [=y] **[Y]**
Wenn diese Option ausgewählt ist, wird der komprimierte Cache für Auslagerungsseiten beim Booten aktiviert, andernfalls wird er deaktiviert. Die hier getroffene Auswahl kann mit der Kernel-Kommandozeilenoption `zswap.enabled=` überschrieben werden.

13.1.1.2 Invalidate zswap entries when pages are loaded

CONFIG_ZSWAP_EXCLUSIVE_LOADS_DEFAULT_ON [=n] **[N]**
Wenn diese Option ausgewählt ist, werden exklusive Lasten für zswap beim Booten aktiviert, andernfalls wird sie deaktiviert. Wenn exklusive Ladungen aktiviert sind, wird beim Laden einer Seite aus zswap der zswap-Eintrag sofort ungültig gemacht, anstatt ihn in zswap zu belassen, bis der Swap-Eintrag freigegeben wird. Dadurch wird vermieden, dass sich zwei Kopien derselben Seite im Speicher befinden (komprimiert und unkomprimiert), nachdem eine Seite aus zswap geladen wurde. Der Preis dafür ist, dass die Seite neu komprimiert wird, wenn sie nie verschmutzt wurde und erneut ausgelagert werden muss.

13.1.1.3 Default compressor () → Wählt den Standardkomprimierungsalgorithmus für den komprimierten Cache für Auslagerungsseiten aus. Einen Überblick darüber, welche Leistung von einem bestimmten Kompressionsalgorithmus erwartet werden kann, finden Sie in den Benchmarks auf der folgenden LWN-Seite: `https://lwn.net/Articles/751795/`
Im Zweifelsfall wählen Sie `LZO`. Die hier getroffene Auswahl kann durch Verwendung der Kernel-Befehlszeilenoption `zswap.compressor=` überschrieben werden.

13.1.1.3.1 Deflate
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_DEFLATE [=n] **[N]**
Verwenden Sie den Deflate-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.3.2 LZO
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_LZO [=n] **[N]**
Verwenden Sie den LZO-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.3.3 842
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_842 [=n] **[N]**
Verwenden Sie den 842-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.3.4 LZ4
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_LZ4 [=n] **[N]**
Verwenden Sie den LZ4-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.3.5 LZ4HC
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_LZ4HC [=n] **[N]**
Verwenden Sie den LZ4HC-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.3.6 zstd
CONFIG_ZSWAP_COMPRESSOR_DEFAULT_ZSTD [=y] **[Y]**
Verwenden Sie den zstd-Algorithmus als Standard-Komprimierungsalgorithmus.

13.1.1.4 Default allocator () →
Wählt den Standardzuweiser für den komprimierten Cache für Auslagerungsseiten aus. Die Voreinstellung ist aus Kompatibilitätsgründen „zbud“, aber lesen Sie bitte die Beschreibung der einzelnen Zuweiser unten, bevor Sie die richtige Wahl treffen. Die hier getroffene Auswahl kann mit der Kernel-Kommandozeilenoption `zswap.zpool=` überschrieben werden.

13.1.1.4.1 zbud
CONFIG_ZSWAP_ZPOOL_DEFAULT_ZBUD [=n] **[N]**
Verwendung des zbud-Allokators als Standard-Allokator.

13.1.1.4.2 z3fold
CONFIG_ZSWAP_ZPOOL_DEFAULT_Z3FOLD [=n] **[N]**
Verwendung des z3fold-Allokators als Standard-Allokator.

13.1.1.4.3 zsmalloc
CONFIG_ZSWAP_ZPOOL_DEFAULT_ZSMALLOC [=n] **[N]**
Verwendung des zsmalloc-Allokators als Standard-Allokator.

13.1.1.5 2:1 compression allocator (zbud) →
CONFIG_ZBUD [=y] **[Y]**
Ein spezieller Allokator für die Speicherung komprimierter Seiten. Er ist für die Speicherung von bis zu zwei komprimierten Seiten pro physischer Seite ausgelegt. Dieses Design schränkt zwar die Speicherdichte ein, hat aber einfache und deterministische Rückgewinnungseigenschaften, die es einem Ansatz mit höherer Dichte vorziehen, wenn die Rückgewinnung verwendet wird.

13.1.1.6 3:1 compression allocator (z3fold) →

CONFIG_Z3FOLD [=y] **[Y]**
Ein spezieller Allokator für die Speicherung komprimierter Seiten. Er ist für die Speicherung von bis zu drei komprimierten Seiten pro physischer Seite ausgelegt. Es handelt sich um ein ZBUD-Derivat, so dass die Einfachheit und der Determinismus weiterhin gegeben sind.

13.1.1.7 N:1 compression allocator (zsmalloc) →

CONFIG_ZSMALLOC [=y] **[Y]**
zsmalloc ist ein Slab-basierter Speicherallokator, der für die effiziente Speicherung von Seiten verschiedener Komprimierungsstufen entwickelt wurde. Er erreicht die höchste Speicherdichte mit der geringsten Fragmentierung.

13.1.1.7.1 Export zsmalloc statistics

CONFIG_ZSMALLOC_STAT [=n] **[N]**
Diese Option ermöglicht es dem Code in zsmalloc, verschiedene Statistiken über die Vorgänge in zsmalloc zu sammeln und diese Informationen über debugfs in den Userspace zu exportieren. Wenn Sie unsicher sind, sagen Sie N.

13.1.1.7.2 Maximum number of physical pages per-zspage

CONFIG_ZSMALLOC_CHAIN_SIZE [=8] **[8]**
Diese Option legt die Obergrenze für die Anzahl der physischen Seiten fest, aus denen eine zmalloc-Seite (zspage) bestehen kann. Die optimale zspage-Kettengröße wird für jede Größenklasse während der Initialisierung des Pools berechnet.
Eine Änderung dieser Option kann die Eigenschaften der Größenklassen verändern, z. B. die Anzahl der Seiten pro zspage und die Anzahl der Objekte pro zspage. Dies kann auch zu unterschiedlichen Konfigurationen des Pools führen, da zsmalloc Größenklassen mit ähnlichen Eigenschaften zusammenführt.
Weitere Informationen finden Sie in der Dokumentation zu zsmalloc.

13.2 SLAB allocator options →

13.2.1 Choose SLAB allocator (SLUB (Unqueued Allocator)) →

Diese Option ermöglicht die Auswahl eines Slab-Allokators.

13.2.1.1 SLAB (DEPRECATEDUnqueued Allocator)

CONFIG_SLAB_DEPRECATED [=n] **[N]**
Veraltet und soll in ein paar Zyklen entfernt werden. Ersetzt durch SLUB. Wenn Sie nicht auf SLUB umsteigen können, wenden Sie sich bitte an linux-mm@kvack.org und an die Personen, die im Abschnitt SLAB ALLOCATOR der MAINTAINERS-Datei aufgeführt sind, und erläutern Sie die Gründe. Der reguläre Slab-Allokator, der sich bewährt hat und bekanntermaßen in allen Umgebungen gut funktioniert. Er organisiert Cache-Hot-Objekte in Warteschlangen pro CPU und pro Knoten.

13.2.1.2 SLUB (Unqueued Allocator)

CONFIG_SLUB [=y] **[Y]**
SLUB ist ein Slab-Allokator, der die Nutzung von Cache-Zeilen minimiert, anstatt Warteschlangen von gecachten Objekten zu verwalten (SLAB-Ansatz). Die Zwischenspeicherung pro CPU wird durch Slabs von Objekten anstelle von Objekt-Warteschlangen realisiert. SLUB kann den Speicher effizient nutzen und verfügt über verbesserte Diagnosefunktionen. SLUB ist die Standardwahl für einen Slab-Allokator.

13.2.2 Allow slab caches to be merged

CONFIG_SLAB_MERGE_DEFAULT [=y] **[Y]**
Um die Fragmentierung des Kernspeichers zu verringern, können Slab-Caches zusammengelegt werden, wenn sie die gleiche Größe und andere Merkmale aufweisen. Dies birgt das Risiko, dass Kernel-Heap-Überläufe Objekte aus zusammengeführten Caches überschreiben können (und das Cache-Layout leichter zu kontrollieren ist), wodurch solche Heap-Angriffe von Angreifern leichter ausgenutzt werden können. Wenn die Caches nicht gemischt werden, können diese Arten von Angriffen normalerweise nur Objekte im selben Cache beschädigen. Um die Zusammenführung zur Laufzeit zu deaktivieren, kann `slab_nomerge` in der Kernel-Befehlszeile übergeben werden.

13.2.3 Randomize slab freelist

CONFIG_SLAB_FREELIST_RANDOM [=y] **[Y]**
Die Reihenfolge der Freelist bei der Erstellung neuer Seiten wird zufällig festgelegt. Dieses Sicherheitsmerkmal verringert die Vorhersagbarkeit der Kernel-Slab-Zuweisung gegen Heap-Überläufe.

13.2.4 Harden slab freelist metadata

CONFIG_SLAB_FREELIST_HARDENED [=y] **[Y]**
Viele Kernel-Heap-Angriffe zielen auf Slab-Cache-Metadaten und andere Infrastrukturen ab. Diese Optionen bringen geringfügige Leistungseinbußen mit sich, um den Kernel-Slab-Allokator gegen gängige Freelist-Angriffsmethoden zu härten. Einige Slab-Implementierungen haben mehr Sanity-Checking als andere. Diese Option ist am effektivsten mit CONFIG_SLUB.

13.2.5 Enable SLUB performance statistics

CONFIG_SLUB_STATS [=n] **[N]**
SLUB-Statistiken sind nützlich, um das Zuweisungsverhalten von SLUBs zu debuggen und Wege zur Optimierung der Zuweisungsfunktion zu finden. Diese Funktion sollte niemals für den produktiven Einsatz aktiviert werden, da die Führung von Statistiken die Zuweisungsfunktion um einige Prozentpunkte verlangsamt. Der Befehl `slabinfo` unterstützt die Ermittlung der aktivsten Slabs, um herauszufinden, welche Slabs für eine bestimmte Last relevant sind. Versuchen Sie Folgendes: `slabinfo -DA`

13.2.6 SLUB per cpu partial cache

CONFIG_SLUB_CPU_PARTIAL [=y] **[Y]**
Partielle Zwischenspeicher pro CPU beschleunigen die Zuweisung und Freigabe von Objekten, die lokal auf einem Prozessor liegen, zum Preis einer größeren Unbestimmtheit bei der Latenzzeit der Freigabe. Bei Überlauf werden diese Caches geleert, was das Einnehmen von Sperren erfordert, die Latenzspitzen verursachen können. Normalerweise würde man sich bei einem Echtzeitsystem für nein entscheiden.

13.2.7 Randomize slab caches for normal kmalloc

CONFIG_RANDOM_KMALLOC_CACHES [=n] **[N]**
Eine Härtungsfunktion, die mehrere Kopien von Slab-Caches für die normale kmalloc-Allokation erstellt und kmalloc veranlasst, eine zufällig auf der Grundlage der Code-Adresse auszuwählen, was es Angreifern erschwert, verwundbare Speicherobjekte auf den Heap zu sprühen, um Speicherschwachstellen auszunutzen. Gegenwärtig ist die Anzahl der Kopien auf 16 festgelegt, ein angemessen großer Wert, der die für verschiedene Subsysteme oder Module zugewiesenen Speicherobjekte effektiv in verschiedene Caches aufteilt, und zwar auf Kosten eines begrenzten Grades an Speicher- und CPU-Overhead, der mit der Hardware und der Systemauslastung zusammenhängt.

13.3 Page allocator randomization

CONFIG_SHUFFLE_PAGE_ALLOCATOR [=y] **[Y]**
Die Randomisierung der Seitenzuweisung verbessert die durchschnittliche Auslastung eines direkt abgebildeten Memory-Side-Cache. In Abschnitt 5.2.27 Heterogeneous Memory Attribute Table (HMAT) der ACPI 6.2a-Spezifikation finden Sie ein Beispiel dafür, wie eine Plattform das Vorhandensein eines speicherseitigen Cache anzeigt. Es gibt auch zufällige Sicherheitsvorteile, da es die Vorhersagbarkeit von Seitenzuweisungen reduziert, um SLAB_FREELIST_RANDOM zu ergänzen, aber die Standardgranularität des Shufflings auf MAX_ORDER, d.h. die 10. Reihenfolge der Seiten wird auf der Grundlage der Cache-Nutzung auf x86 ausgewählt. Die Randomisierung verbessert zwar die Cache-Nutzung, kann sich aber auf Plattformen ohne Cache negativ auf die Arbeitslast auswirken. Aus diesem Grund wird die Randomisierung standardmäßig nur aktiviert, wenn zur Laufzeit ein direkt zugeordneter Memory-Side-Cache erkannt wird. Andernfalls kann die Randomisierung mit dem Kernel-Befehlszeilenparameter `page_alloc.shuffle` zwangsweise aktiviert werden. Sagen Sie Y, wenn Sie unsicher sind.

13.4 Disable heap randomization

CONFIG_COMPAT_BRK [=n] **[N]**
Die Randomisierung der Heap-Platzierung macht Heap-Exploits schwieriger, aber sie macht auch alte Binärdateien (einschließlich aller libc5-basierten) kaputt. Diese Option ändert die Standardeinstellung beim Booten auf Heap-Randomisierung deaktiviert und kann zur Laufzeit überschrieben werden, indem /proc/sys/kernel/randomize_va_space auf 2 gesetzt wird. Auf nicht-alten Distributionen (nach 2000) ist N normalerweise eine sichere Wahl.

13.5 Sparse Memory virtual memmap

CONFIG_SPARSEMEM_VMEMMAP [=y] **[Y]**
SPARSEMEM_VMEMMAP verwendet eine virtuell gemappte Memmap, um textttpfn_to_page und `page_to_pfn` Operationen zu optimieren. Dies ist die effizienteste Option, wenn genügend Kernel-Ressourcen verfügbar sind.

13.6 Memory hotplug →

CONFIG_HOTPLUG [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

13.6.1 Online the newly added memory blocks by default

CONFIG_MEMORY_HOTPLUG_DEFAULT_ONLINE [=y] **[Y]**
Diese Option legt die Standardeinstellung für die Hotplug-Onlining-Richtlinie für Speicher fest (/sys/devices/system/memory/auto_online_blocks), die bestimmt, was mit neu hinzugefügten Speicherbereichen geschieht. Die Richtlinieneinstellung kann jederzeit zur Laufzeit geändert werden.
Siehe Documentation/admin-guide/mm/memory-hotplug.rst für weitere Informationen. Geben Sie hier Y an, wenn Sie möchten, dass alle Hotplug-Speicherblöcke standardmäßig im „Online“-Zustand erscheinen. Geben Sie hier N an, wenn Sie möchten, dass die Standardrichtlinie alle Hot-Plugged-Speicherblöcke im „Offline“-Zustand hält.

13.6.2 Allow for memory hot remove

CONFIG_MEMORY_HOTREMOVE [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

13.7 Allow for balloon memory compaction/migration

CONFIG_BALLOON_COMPACTION [=y] **[Y]**
Die durch das Ballooning verursachte Speicherfragmentierung kann die Anzahl der zusammenhängenden 2-MB-Speicherblöcke, die in einem Gastsystem verwendet werden können, erheblich verringern, was zu Leistungseinbußen aufgrund der geringeren Anzahl transparenter großer Seiten führt, die vom Gastsystem verwendet werden können. Das Zulassen der Verdichtung und Migration für Speicherseiten, die als Teil von Speicher-Ballon-Geräten eingetragen sind, vermeidet das oben beschriebene Szenario und trägt zur Verbesserung der Speicherdefragmentierung bei.

13.8 Allow for memory compaction

CONFIG_COMPACTION [=y] **[Y]**
Die Verdichtung ist die einzige Speicherverwaltungskomponente, die zuverlässig Speicherblöcke hoher Ordnung (größere, physisch zusammenhängende Blöcke) bildet. Die Seitenzuweisung ist in hohem Maße auf die Verdichtung angewiesen, und das Fehlen dieser Funktion kann bei Speicheranforderungen hoher Ordnung zu unerwarteten OOM-Killer-Aufrufen führen. Sie sollten diese Option nicht deaktivieren, es sei denn, es gibt wirklich einen triftigen Grund dafür, und dann wären wir sehr daran interessiert, diesen unter linux-mm@kvack.org zu erfahren.

13.9 Free page reporting

CONFIG_PAGE_REPORTING [=y] **[Y]**
Die Meldung freier Seiten ermöglicht die inkrementelle Erfassung freier Seiten vom Buddy-Allokator mit dem Ziel, diese Seiten einer anderen Einheit, z. B. einem Hypervisor, zu melden, damit der Speicher innerhalb des Hosts für andere Zwecke freigegeben werden kann.

13.10 Page migration

CONFIG_MIGRATION [=y] **[Y]**
Ermöglicht die Migration des physischen Standorts von Seiten von Prozessen, während die virtuellen Adressen nicht geändert werden. Dies ist in zwei Situationen nützlich. Erstens auf NUMA-Systemen, um Seiten näher an die zugreifenden Prozessoren zu bringen. Zweitens bei der Zuweisung großer Seiten, da durch die Migration Seiten verlagert werden können, um eine große Seitenzuweisung zu erfüllen, anstatt sie zurückzufordern.

13.11 Enable KSM for page merging

CONFIG_KSM [=y] **[Y]**
Aktivieren Sie Kernel Samepage Merging: KSM scannt in regelmäßigen Abständen die Bereiche des Adressraums einer Anwendung, die laut einer Anwendung zusammengeführt werden können. Wenn er Seiten mit identischem Inhalt findet, ersetzt er die vielen Instanzen durch eine einzige Seite mit diesem Inhalt und spart so Speicher, bis eine oder eine andere Anwendung den Inhalt ändern muss. Empfohlen für die Verwendung mit KVM oder mit anderen doppelten Anwendungen.
Siehe Documentation/mm/ksm.rst für weitere Informationen: KSM ist inaktiv, bis ein Programm festgestellt hat, dass ein Bereich MADV_MERGEABLE ist, und root /sys/kernel/mm/ksm/run auf 1 gesetzt hat (wenn CONFIG_SYSFS gesetzt ist).

13.12 Low address space to protect from user allocation

CONFIG_DEFAULT_MMAP_MIN_ADDR [=65536] **[65536]**
Dies ist der Teil des niedrigen virtuellen Speichers, der vor der Zuweisung an den Benutzerraum geschützt werden sollte. Wenn ein Benutzer davon abgehalten wird, auf niedrige Seiten zu schreiben, kann dies dazu beitragen, die Auswirkungen von NULL-Zeiger-Fehlern im Kernel zu verringern. Für die meisten ia64-, ppc64- und x86-Benutzer mit viel Adressraum ist ein Wert von 65536 angemessen und sollte keine Probleme verursachen. Auf Arm und anderen Architekturen sollte er nicht höher als 32768 sein. Programme, die die vm86-Funktionalität nutzen oder diesen niedrigen Adressraum abbilden müssen, benötigen CAP_SYS_RAWIO oder deaktivieren diesen Schutz, indem sie den Wert auf 0 setzen. Dieser Wert kann nach dem Booten mit dem Parameter /proc/sys/vm/mmap_min_addr geändert werden.

13.13 Enable recovery from hardware memory errors

CONFIG_MEMORY_FAILURE [=y] **[Y]**
Ermöglicht die Wiederherstellung von Code nach einigen Speicherfehlern auf Systemen mit MCA-Wiederherstellung. Dadurch kann ein System auch dann weiterlaufen, wenn ein Teil des Speichers unkorrigierte Fehler aufweist. Dies erfordert spezielle Hardwareunterstützung und in der Regel ECC-Speicher.

13.13.1 HWPoison pages injector

CONFIG_HWPOISON_INJECT [=m] **[M]**
Für diese Option gibt es keine Hilfe.

13.14 Transparent Hugepage Support →

CONFIG_TRANSPARENT_HUGEPAGE [=y] **[Y]**
Transparent Hugepages erlaubt es dem Kernel, große Seiten und große tlb transparent für die Anwendungen zu verwenden, wann immer dies möglich ist. Diese Funktion kann die Rechenleistung bestimmter Anwendungen verbessern, indem sie Seitenfehler bei der Speicherzuweisung beschleunigt, die Anzahl der tlb-Misses verringert und das Durchlaufen der Seitentabelle beschleunigt. Wenn der Speicher bei eingebetteten Systemen begrenzt ist, können Sie N angeben.

13.14.1 Transparent Hugepage Support sysfs defaults () →

Wählt die sysfs-Vorgaben für die transparente Hugepage-Unterstützung aus.

13.14.1.1 always

CONFIG_TRANSPARENT_HUGEPAGE_ALWAYS [=y] **[Y]**
Die ständige Aktivierung von Transparent Hugepage kann den Speicherbedarf von Anwendungen erhöhen, ohne dass dies einen garantierten Nutzen hat, aber es funktioniert automatisch für alle Anwendungen.

13.14.1.2 madvise

CONFIG_TRANSPARENT_HUGEPAGE_MADVISE [=n] **[N]**
Die Aktivierung von Transparent Hugepage madvise bringt nur eine Leistungsverbesserung für die Anwendungen, die madvise(MADV_HUGEPAGE) verwenden, aber es besteht nicht die Gefahr, dass der Speicherbedarf von Anwendungen ohne garantierten Nutzen erhöht wird.

13.14.2 Read-only THP for filesystems (EXPERIMENTAL)

CONFIG_READ_ONLY_THP_FOR_FS [=y] **[Y]**
Erlaubt khugepaged, schreibgeschützte Seiten in THP zu speichern. Dies ist als experimentell gekennzeichnet, da es sich um eine neue Funktion handelt. Schreibunterstützung für Datei-THPs wird in den nächsten Release-Zyklen entwickelt werden.

13.15 Contiguous Memory Allocator

CONFIG_CMA [=y] **[Y]**
Dadurch wird der Contiguous Memory Allocator aktiviert, der es anderen Subsystemen ermöglicht, große, physisch zusammenhängende Speicherblöcke zuzuweisen. CMA reserviert einen Speicherbereich und erlaubt nur die Zuweisung beweglicher Seiten aus diesem Bereich. Auf diese Weise kann der Kernel den Speicher als Pagecache verwenden, und wenn ein Subsystem einen zusammenhängenden Bereich anfordert, werden die zugewiesenen Seiten verschoben, um die zusammenhängende Anforderung zu bedienen. Wenn Sie unsicher sind, sagen Sie N für nein.

13.15.1 CMA debug messages (DEVELOPMENT)

CONFIG_CMA_DEBUG [=n] **[N]**
Schaltet Debug-Meldungen in CMA ein. Dies erzeugt KERN_DEBUG-Meldungen für jeden CMA-Aufruf sowie verschiedene Meldungen während der Verarbeitung von Aufrufen wie dma_alloc_from_contiguous(). Diese Option hat keinen Einfluss auf Warn- und Fehlermeldungen.

13.15.2 CMA debugfs interface

CONFIG_CMA_DEBUGFS [=y] **[Y]**
Schaltet die DebugFS-Schnittstelle für CMA ein.

13.15.3 CMA information through sysfs interface

CONFIG_CMA_SYSFS [=y] **[Y]**
Diese Option legt einige sysfs-Attribute offen, um Informationen von CMA zu erhalten.

13.15.4 Maximum count of the CMA areas

CONFIG_CMA_AREAS [=7] **[7]**
CMA ermöglicht es, CMA-Bereiche für bestimmte Zwecke zu erstellen, die hauptsächlich als privater Bereich des Geräts verwendet werden. Mit diesem Parameter wird die maximale Anzahl von CMA-Bereichen im System festgelegt. Wenn Sie unsicher sind, belassen Sie den Standardwert „7“ bei UMA und „19“ bei NUMA.

13.16 Track memory changes

CONFIG_MEM_SOFT_DIRTY [=y] **[Y]**
Diese Option ermöglicht die Verfolgung von Speicheränderungen durch Einführung eines Soft-Dirty-Bits auf pte-s. Dieses Bit wird gesetzt, wenn jemand in eine Seite schreibt, genau wie das reguläre Dirty Bit, aber im Gegensatz zu letzterem kann es von Hand gelöscht werden. Siehe Documentation/admin-guide/mm/soft-dirty.rst für weitere Details.

13.17 Defer initialisation of struct pages to kthreads

CONFIG_DEFERRED_STRUCT_PAGE_INIT [=n] **[N]**
Normalerweise werden alle Strukturseiten beim Frühstart in einem einzigen Thread initialisiert. Auf sehr großen Rechnern kann dies sehr viel Zeit in Anspruch nehmen. Wenn diese Option gesetzt ist, wird bei großen Maschinen eine Teilmenge der memmap beim Booten aufgerufen und der Rest parallel initialisiert. Dies kann sich auf die Leistung von Aufgaben auswirken, die zu Beginn der Lebensdauer des Systems ausgeführt werden, bis diese kthreads die Initialisierung abgeschlossen haben.

13.18 Enable idle page tracking

CONFIG_IDLE_PAGE_TRACKING [=y] **[Y]**
Diese Funktion ermöglicht es, die Anzahl der Benutzerseiten zu schätzen, die in einem bestimmten Zeitraum nicht berührt wurden. Diese Information kann nützlich sein, um die Grenzen der Speichergruppen und/oder die Platzierung von Aufträgen innerhalb eines Rechenclusters zu optimieren.
Siehe Documentation/admin-guide/mm/idle_page_tracking.rst für weitere Einzelheiten.

13.19 Device memory (pmem, HMM, etc...) hotplug support

CONFIG_ZONE_DEVICE [=y] **[Y]**
Die Hotplug-Unterstützung für Gerätespeicher ermöglicht es, pmem oder andere vom Gerätetreiber entdeckte Speicherregionen in der Memmap zu etablieren. Dies ermöglicht pfn_to_page()-Lookups von ansonsten „gerätephysikalischen“ Adressen, was unter anderem für die Verwendung einer DAX-Zuordnung in einer O_DIRECT-Operation erforderlich ist. Wenn FS_DAX aktiviert ist, dann sagen Sie Y.

13.20 Unaddressable device memory (GPU memory, ...)

CONFIG_DEVICE_PRIVATE [=y] **[Y]**
Ermöglicht die Erstellung von Strukturseiten zur Darstellung von nicht adressierbarem Gerätespeicher, d. h. Speicher, auf den nur vom Gerät (oder einer Gruppe von Geräten) aus zugegriffen werden kann. Wahrscheinlich sollten Sie auch HMM_MIRROR auswählen.

13.21 Collect percpu memory statistics

CONFIG_PERCPU_STATS [=n] **[N]**
Diese Funktion sammelt Statistiken und stellt sie über debugfs zur Verfügung. Die Informationen umfassen globale und pro Chunk-Statistiken, die dazu beitragen können, die Speichernutzung der CPU zu verstehen.

13.22 Enable infrastructure for get_user_pages()-related unit tests

CONFIG_GUP_TEST [=n] **[N]**
Stellt /sys/kernel/debug/gup_test zur Verfügung, das wiederum eine Möglichkeit bietet, ioctl-Aufrufe zu machen, die kernelbasierte Unit-Tests für die get_user_pages*()- und pin_user_pages*()-Familie von API-Aufrufen starten können. Diese Tests umfassen Benchmark-Tests für die schnellen Varianten von get_user_pages*() und pin_user_pages*() sowie Smoke-Tests für die nicht schnellen Varianten. Es gibt auch einen Untertest, der die Ausführung von dump_page() auf bis zu acht Seiten (ausgewählt durch Befehlszeilen-Args) innerhalb des Bereichs der User-Space-Adressen ermöglicht. Diese Seiten werden entweder über pin_user_pages*() oder über get_user_pages*() angeheftet, wie durch andere Befehlszeilenargumente angegeben.
Siehe tools/testing/selftests/mm/gup_test.c

13.23 Enable a module to run time tests on dma_pool

CONFIG_DMAPOOL_TEST [=n] **[N]**
Stellt ein Testmodul zur Verfügung, das viele Blöcke unterschiedlicher Größe alloziert und freigibt und berichtet, wie lange es dauert. Damit soll ein konsistenter Weg gefunden werden, um zu messen, wie sich Änderungen an den dma_pool_alloc/free-Routinen auf die Leistung auswirken.

13.24 Anonymous VMS name support

CONFIG_ANON_VMA_NAME [=y] **[Y]**
Erlaubt die Benennung anonymer virtueller Speicherbereiche. Mit dieser Funktion können virtuellen Speicherbereichen Namen zugewiesen werden.
Die zugewiesenen Namen können später aus /proc/pid/maps und /proc/pid/smaps abgerufen werden und helfen bei der Identifizierung einzelner anonymer Speicherbereiche. Die Zuweisung eines Namens für einen anonymen virtuellen Speicherbereich kann verhindern, dass dieser Bereich aufgrund des unterschiedlichen Namens mit benachbarten virtuellen Speicherbereichen zusammengelegt wird.

13.25 Enable userfaultfd() system call

CONFIG_USERFAULTFD [=y] **[Y]**
Aktivieren Sie den Systemaufruf userfaultfd(), der das Abfangen und Behandeln von Seitenfehlern im Userland ermöglicht.

13.26 Userfaultfd write protection support for shmem/hugetlbfs

CONFIG_PTE_MARKER_UFFD_WP [=y] **[Y]**
Ermöglicht die Erstellung von Marker-PTEs für den Userfaultfd-Schreibschutz. Sie ist erforderlich, um den userfaultfd-Schreibschutz für dateigebundene Speichertypen wie shmem und hugetlbfs zu aktivieren.

13.27 Multi-Gen LRU

CONFIG_LRU_GEN [=y] **[Y]**
Eine hochleistungsfähige LRU-Implementierung zur Überbelegung von Speicher.
Siehe Documentation/admin-guide/mm/multigen_lru.rst für Details.

13.27.1 Enable by default

CONFIG_LRU_GEN_ENABLED [=y] **[Y]**
Mit dieser Option wird das Multi-Gen-LRU standardmäßig aktiviert.

13.27.2 Full stats for debugging

CONFIG_LRU_GEN_STATS [=n] **[N]**
Aktivieren Sie diese Option nicht, es sei denn, Sie möchten sich die historischen Statistiken der ausgeschiedenen Generationen zu Fehlersuchzwecken ansehen. Diese Option hat einen Speicher-Overhead pro memcg und pro Knoten.

13.28 Data Access Monitoring →

(Überwachung des Datenzugriffs)

13.28.1 DAMON: Data Access Monitoring Framework

CONFIG_DAMON [=y] **[Y]**
Damit wird ein Rahmen geschaffen, der es den Kernel-Subsystemen ermöglicht, die Zugriffshäufigkeit der einzelnen Speicherbereiche zu überwachen. Diese Informationen können für eine leistungsorientierte Speicherverwaltung auf DRAM-Ebene nützlich sein. Weitere Informationen finden Sie unter
`https://damonitor.github.io/doc/html/latest-damon/index.html`.

13.28.1.1 Data access monitoring operations for virtual address spaces

CONFIG_DAMON_VADDR [=y] **[Y]**
Damit werden die Standardoperationen zur Überwachung des Datenzugriffs für DAMON erstellt, die für virtuelle Adressräume funktionieren.

13.28.1.2 Data access monitoring operations for the physical address space

CONFIG_DAMON_PADDR [=y] **[Y]**
Damit werden die Standardvorgänge zur Datenzugriffsüberwachung für DAMON erstellt, die für den physischen Adressraum funktionieren.

13.28.2 DAMON sysfs interface

CONFIG_DAMON_SYSFS [=y] **[Y]**
Dies bildet die sysfs-Schnittstelle für DAMON. Der Benutzerbereich kann die Schnittstelle für die Überwachung beliebiger Datenzugriffe verwenden.

13.28.3 DAMON debugfs interface (DEPRECATED!)

CONFIG_DAMON_DBGFS [=y] **[N]**
Damit wird die debugfs-Schnittstelle für DAMON erstellt. Die Benutzerraum-Administratoren können die Schnittstelle für die Überwachung beliebiger Datenzugriffe verwenden. Wenn Sie unsicher sind, sagen Sie N.
Dies ist veraltet, daher sollten Benutzer auf die sysfs-Schnittstelle (DAMON_SYSFS) umsteigen. Wenn Sie auf diese Schnittstelle angewiesen sind und nicht umsteigen können, melden Sie bitte Ihren Anwendungsfall an damon@lists.linux.dev und linux-mm@kvack.org.

13.28.4 Build DAMON-based reclaim (DAMON_RECLAIM)

CONFIG_DAMON_RECLAIM [=y] **[Y]**
Damit wird das DAMON-basierte Reklamationssubsystem aufgebaut. Es findet Seiten, auf die lange Zeit nicht mehr mit DAMON zugegriffen wurde (Cold) und fordert diese zurück. Dies wird als proaktive und leichtgewichtige Rückgewinnung bei geringem Speicherdruck vorgeschlagen, während die traditionelle, auf Seitenscans basierende Rückgewinnung bei hohem Druck verwendet wird.

13.28.5 Build DAMON-based LRU-lists sorting (DAMON_LRU_SORT)

CONFIG_DAMON_LRU_SORT [=y] **[Y]**
Damit wird das DAMON-basierte LRU-Listensortier-Subsystem aufgebaut. Es versucht, häufig zugegriffene (heiße) Seiten zu schützen, während selten zugegriffene (kalte) Seiten unter Speicherdruck zuerst zurückgefordert werden.

14 Networking support →

CONFIG_NET [=y] **[Y]**
Wenn Sie nicht wirklich wissen, was Sie tun, sollten Sie hier Y sagen. Der Grund dafür ist, dass einige Programme die Netzwerkunterstützung des Kernels benötigen, auch wenn sie auf einem eigenständigen Rechner laufen, der nicht mit einem anderen Computer verbunden ist. Wenn Sie von einem älteren Kernel aufrüsten, sollten Sie auch Ihre Netzwerkwerkzeuge aktualisieren, da Änderungen am Kernel und an den Werkzeugen oft Hand in Hand gehen. Die Werkzeuge sind in dem Paket `net-tools` enthalten, dessen Standort und Versionsnummer in <file:Documentation/Changes> angegeben sind.
Für eine allgemeine Einführung in Linux-Netzwerke ist es sehr empfehlenswert, das NET-HOWTO zu lesen, das unter `http://www.tldp.org/docs.html#howto` verfügbar ist.

14.1 Networking options →

(Vernetzungsoptionen)

14.1.1 Packet socket

CONFIG_PACKET [=y] **[Y]**
Das Packet-Protokoll wird von Anwendungen verwendet, die direkt mit Netzwerkgeräten kommunizieren, ohne dass ein dazwischenliegendes Netzwerkprotokoll im Kernel implementiert ist, z. B. tcpdump. Wenn Sie wollen, dass diese Anwendungen funktionieren, wählen Sie Y. Um diesen Treiber als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `af_packet` heißen.
Wenn Sie unsicher sind, wählen Sie Y.

14.1.1.1 Packet: sockets monitoring interface

CONFIG_PACKET_DIAG [=m] **[M]**
Unterstützung für die PF_PACKET-Sockel-Überwachungsschnittstelle, die vom Werkzeug `ss` verwendet wird. Wenn Sie unsicher sind, sagen Sie Y.

14.1.2 Unix domain sockets

CONFIG_UNIX [=y] **[Y]**
Wenn Sie hier Y angeben, wird die Unterstützung für Unix-Domain-Sockets einbezogen; Sockets sind der Standard-Unix-Mechanismus für den Aufbau von und den Zugriff auf Netzwerkverbindungen. Viele häufig verwendete Programme wie das X-Window-System und syslog verwenden diese Sockets, auch wenn Ihr Rechner nicht an ein Netzwerk angeschlossen ist. Wenn Sie nicht gerade an einem eingebetteten System oder etwas Ähnlichem arbeiten, sollten Sie hier also unbedingt Y sagen. Sagen Sie Y, wenn Sie nicht genau wissen, was Sie tun.

14.1.2.1 UNIX: socket monitoring interface

CONFIG_UNIX_DIAG [=m] **[M]**
Unterstützung für die vom Tool `ss` verwendete UNIX-Socket-Überwachungsschnittstelle. Wenn Sie unsicher sind, sagen Sie Y.

14.1.3 Transport Layer Security support

CONFIG_TLS [=m] **[M]**
Aktivierung der Kernel-Unterstützung für das TLS-Protokoll. Dadurch kann die symmetrische Verschlüsselung des TLS-Protokolls im Kernel durchgeführt werden. Wenn Sie unsicher sind, sagen Sie N.

14.1.3.1 Transport Layer Security HW offload

CONFIG_TLS_DEVICE [=y] **[Y]**
Aktivierung der Kernel-Unterstützung für die HW-Auslagerung des TLS-Protokolls. Wenn Sie unsicher sind, sagen Sie N.

14.1.3.2 Transport Layer Security TCP stack bypass

CONFIG_TLS_TOE [=n] **[N]**
Aktivierung der Kernel-Unterstützung für das Legacy-HW-Offload des TLS-Protokolls, das mit der Semantik des Linux-Netzwerkstacks inkompatibel ist. Wenn Sie unsicher sind, sagen Sie N.

14.1.4 Transformation user configuration interface

CONFIG_XFRM_USER [=y] **[Y]**
Unterstützung für Transformation(XFRM)-Benutzerkonfigurationsschnittstelle wie IPsec, die von nativen Linux-Tools verwendet wird. Wenn Sie unsicher sind, sagen Sie Y.

14.1.4.1 Compatible ABI support

CONFIG_XFRM_USER_COMPAT [=n] **[N]**
Transformation(XFRM)-Benutzerkonfigurationsschnittstelle wie IPsec, die von kompatiblen Linux-Anwendungen verwendet wird. Wenn Sie unsicher sind, sagen Sie N.

14.1.5 Transformation virtual interface

CONFIG_XFRM_Interface [=m] **[M]**
Damit wird eine virtuelle Schnittstelle zum Routen des IPsec-Verkehrs bereitgestellt. Wenn Sie unsicher sind, sagen Sie N.

14.1.6 Transformation sub policy support

CONFIG_XFRM_SUB_POLICY [=y] **[Y]**
Unterstützung von Unterrichtsrichtlinien für Entwickler. Durch die Verwendung der Unterrichtlinie mit der Hauptrichtlinie können zwei Richtlinien gleichzeitig auf dasselbe Paket angewendet werden. Eine Richtlinie, die kürzer im Kernel lebt, sollte eine Unterrichtlinie sein. Wenn Sie unsicher sind, sagen Sie N.

14.1.7 Transformation migrate database

CONFIG_XFRM_MIGRATE [=y] **[Y]**
Eine Funktion zur dynamischen Aktualisierung von Locator(s) einer bestimmten IPsec-Sicherheitsassoziation. Diese Funktion ist z. B. in einer mobilen IPv6-Umgebung mit IPsec-Konfiguration erforderlich, in der mobile Knoten ihren Verbindungspunkt zum Internet ändern. Wenn Sie unsicher sind, sagen Sie N.

14.1.8 Transformation statistics

CONFIG_XFRM_STATISTICS [=y] **[Y]**
Diese Statistik ist keine SNMP/MIB-Spezifikation, sondern zeigt Statistiken über Transformationsfehler (oder Fast-Fehler) bei der Paketverarbeitung für Entwickler. Wenn Sie unsicher sind, sagen Sie N.

14.1.9 PF_KEY sockets

CONFIG_NET_KEY [=m] **[M]**
PF_KEYv2-Buchsenfamilie, kompatibel zu KAME-Buchsen. Sie sind erforderlich, wenn Sie die von KAME portierten IPsec-Tools verwenden wollen. Sagen Sie Y, wenn Sie nicht wissen, was Sie tun.

14.1.9.1 PF_KEY MIGRATE

CONFIG_NET_KEY_MIGRATE [=y] **[Y]**
Hinzufügen einer PF_KEY MIGRATE Nachricht zur PF_KEYv2 Socket Familie. Die PF_KEY MIGRATE-Nachricht wird zur dynamischen Aktualisierung von Locator(s) einer bestimmten IPsec-Sicherheitsassoziation verwendet. Diese Funktion ist z. B. in einer mobilen IPv6-Umgebung mit IPsec-Konfiguration erforderlich, in der mobile Knoten ihren Verbindungspunkt zum Internet ändern. Detaillierte Informationen sind im Internet-Entwurf <draft-sugimoto-mip6-pfkey-migrate> zu finden. Wenn Sie unsicher sind, sagen Sie N.

14.1.10 SMC socket protocol family

CONFIG_SMC [=m]] **[M]**
SMC-R bietet eine „Sockets over RDMA“-Lösung, die die RDMA over Converged Ethernet (RoCE)-Technologie nutzt, um AF_INET-TCP-Verbindungen transparent zu aktualisieren. Die Linux-Implementierung der SMC-R-Lösung ist als separate Socket-Familie SMC konzipiert. Wählen Sie diese Option, wenn Sie SMC-Socket-Anwendungen ausführen möchten.

14.1.10.1 SMC socket protocol family

CONFIG_SMC_DIAG [=m]] **[M]**
Unterstützung für die SMC-Socket-Überwachungsschnittstelle, die von Tools wie `smcss` verwendet wird. Wenn Sie unsicher sind, sagen Sie Y.

14.1.11 XDP sockets

CONFIG_XDP_SOCKETS [=y]] **[Y]**
XDP-Sockets ermöglichen einen Kanal zwischen XDP-Programmen und Userspace-Anwendungen.

14.1.11.1 XDP sockets: monitoring interface

CONFIG_XDP_SOCKETS_DIAG [=m]] **[M]**
Unterstützung für die vom `ss`-Tool verwendete PF_XDP-Socket-Überwachungsschnittstelle. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12 TCP/IP networking

CONFIG_INET [=y]] **[Y]**
Dies sind die Protokolle, die im Internet und in den meisten lokalen Ethernets verwendet werden. Es wird dringend empfohlen hier Y anzugeben (dadurch wird Ihr Kernel um etwa 400 KB vergrößert), da einige Programme (z. B. das X-Window-System) TCP/IP verwenden, auch wenn Ihr Rechner nicht mit einem anderen Computer verbunden ist. Sie erhalten das sogenannte Loopback-Gerät, mit dem Sie sich selbst anpingen können (was ein großer Spaß ist!). Eine ausgezeichnete Einführung in die Linux-Netzwerktechnik finden Sie im Linux Networking HOWTO, erhältlich bei `http://www.tldp.org/docs.html#howto`. Wenn Sie hier Y sagen und auch zu „/proc file system support“ und „Sysctl support“ unten, können Sie verschiedene Aspekte des Verhaltens des TCP/IP-Codes ändern, indem Sie in die (virtuellen) Dateien in /proc/sys/net/ipv4/* schreiben; die Optionen werden in der Datei <file:Documentation/networking/ip-sysctl.rst> erläutert. Kurze Antwort: Sagen Sie Y.

14.1.12.1 IP: multicasting

CONFIG_IP_MULTICAST [=y] **[Y]**
Dabei handelt es sich um einen Code zur gleichzeitigen Adressierung mehrerer vernetzter Computer, der Ihren Kernel um etwa 2 KB vergrößert. Sie brauchen Multicasting, wenn Sie am MBONE teilnehmen wollen, einem Netz mit hoher Bandbreite über dem Internet, das Audio- und Videoübertragungen überträgt. Weitere Informationen über MBONE finden Sie im WWW unter `https://www.savetz.com/mbone/`. Für die meisten Leute ist es sicher, N zu sagen.

14.1.12.2 IP: advanced router

CONFIG_IP_ADVANCED_ROUTER [=y] **[Y]**
Wenn Sie beabsichtigen, Ihren Linux-Rechner hauptsächlich als Router zu betreiben, d.h. als Computer, der Netzwerkpakete weiterleitet und umverteilt, sagen Sie Y; Ihnen werden dann mehrere Optionen angezeigt, die eine genauere Kontrolle über den Routing-Prozess ermöglichen.
Die Antwort auf diese Frage wirkt sich nicht direkt auf den Kernel aus: Wenn Sie mit N antworten, überspringt der Konfigurator einfach alle Fragen zum erweiterten Routing. Beachten Sie, dass Ihr Rechner nur dann als Router fungieren kann, wenn Sie die IP-Weiterleitung in Ihrem Kernel aktivieren; dies können Sie tun, indem Sie „/proc file system support“ und „Sysctl support“ mit Y beantworten und die folgende Zeile

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

beim Booten ausführen, nachdem das Dateisystem /proc eingehängt wurde.
Wenn Sie die IP-Weiterleitung einschalten, sollten Sie den rp_filter in Betracht ziehen, der eingehende Pakete automatisch zurückweist, wenn der Routing-Tabelleneintrag für ihre Quelladresse nicht mit der Netzwerkschnittstelle übereinstimmt, an der sie ankommen. Dies hat Sicherheitsvorteile, weil es das so genannte IP-Spoofing verhindert, kann aber Probleme bereiten, wenn Sie asymmetrisches Routing verwenden (Pakete von Ihnen zu einem Host nehmen einen anderen Weg als Pakete von diesem Host zu Ihnen) oder wenn Sie einen nicht routingfähigen Host betreiben, der mehrere IP-Adressen auf verschiedenen Schnittstellen hat. Um rp_filter einzuschalten, verwenden Sie:

```
echo 1 > /proc/sys/net/ipv4/conf/<Gerät>/rp_filter
```

oder

```
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Beachten Sie, dass einige Distributionen dies in Startskripten aktivieren. Für Details über rp_filter strict und loose mode lesen Sie <file:Documentation/networking/ip-sysctl.rst>. Wenn Sie unsicher sind, geben Sie hier N an.

14.1.12.2.1 FIB TRIE statistics

CONFIG_IP_FIB_TRIE_STATS [=y] **[Y]**
Behalten Sie die Statistiken über die Struktur der FIB TRIE-Tabelle im Auge. Nützlich zum Testen und Messen der TRIE-Leistung.

14.1.12.2.2 IP: policy routing

CONFIG_IP_MULTIPLE_TABLES [=y] **[Y]**
Normalerweise entscheidet ein Router, was mit einem empfangenen Paket zu tun ist, und zwar ausschließlich auf der Grundlage der endgültigen Zieladresse des Pakets. Wenn Sie hier Y angeben, kann der Linux-Router auch die Quelladresse des Pakets berücksichtigen. Darüber hinaus kann auch das TOS-Feld (Type-Of-Service) des Pakets für Routing-Entscheidungen verwendet werden.
Weitere Informationen finden Sie in der Linux-Dokumentation Advanced Routing and Traffic Control unter `https://lartc.org/howto/lartc.rpdb.html`. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.2.3 IP: equal cost multipath

CONFIG_IP_ROUTE_MULTIPATH [=y] **[Y]**
Normalerweise geben die Routing-Tabellen eine einzige Aktion an, die für ein bestimmtes Paket auf deterministische Weise durchgeführt wird. Wenn Sie hier jedoch Y sagen, ist es möglich, mehrere Aktionen an ein Paketmuster zu knüpfen und damit mehrere alternative Wege für diese Pakete festzulegen. Der Router betrachtet alle diese Pfade als gleich teuer und wählt einen von ihnen auf nicht-deterministische Weise aus, wenn ein passendes Paket eintrifft.

14.1.12.2.4 IP: verbose route monitoring

CONFIG_IP_ROUTE_VERBOSE [=y] **[Y]**
Wenn Sie hier Y angeben, was empfohlen wird, gibt der Kernel ausführliche Meldungen über das Routing aus, zum Beispiel Warnungen über empfangene Pakete, die seltsam aussehen und auf einen Angriff oder ein falsch konfiguriertes System hindeuten könnten. Die Informationen werden vom klogd-Daemon verarbeitet, der für die Kernelmeldungen zuständig ist („man klogd“).

14.1.12.3 IP: kernel level autoconfiguration

CONFIG_IP_ADVANCED_ROUTER [=n] **[N]**
Dies ermöglicht die automatische Konfiguration der IP-Adressen von Geräten und der Routing-Tabelle beim Booten des Kernels auf der Grundlage von Informationen, die entweder über die Kernel-Befehlszeile oder über BOOTP- oder RARP-Protokolle bereitgestellt werden. Sie müssen Y nur für plattenlose Maschinen angeben, die zum Booten Netzwerkzugriff benötigen (in diesem Fall sollten Sie auch Y für „Root file system on NFS“ angeben), da alle anderen Maschinen das Netzwerk in ihren Startskripten konfigurieren.

14.1.12.4 IP: tunneling

CONFIG_NET_IPIP [=m] **[M]**
Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dieser spezielle Tunneling-Treiber implementiert die Verkapselung von IP innerhalb von IP, was sich zwar ziemlich sinnlos anhört, aber nützlich sein kann, wenn Sie Ihren (oder einen anderen) Rechner in einem anderen Netz erscheinen lassen wollen, als er tatsächlich ist, oder wenn Sie die Möglichkeiten von Mobile-IP nutzen wollen (wodurch Laptops nahtlos zwischen Netzen wechseln können, ohne ihre IP-Adressen zu ändern). Wenn Sie diese Option mit Y bestätigen, werden zwei Module (= Code, der in den laufenden Kernel eingefügt und aus ihm entfernt werden kann, wann immer Sie wollen) erzeugt. Die meisten Leute werden das nicht brauchen und können N sagen.

14.1.12.5 IP: GRE demultiplexer

CONFIG_NET_IPGRE_DEMUX [=m] **[M]**
Dies ist ein Hilfsmodul zum Demultiplexen von GRE-Paketen anhand von GRE-Versionsfeldkriterien. Erforderlich für die Module `ip_gre` und `pptp`.

14.1.12.6 IP: GRE tunnels over IP

CONFIG_NET_IPGRE [=m] **[M]**
Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dieser spezielle Tunneling-Treiber implementiert GRE (Generic Routing Encapsulation) und ermöglicht derzeit die Verkapselung von IPv4 oder IPv6 über eine bestehende IPv4-Infrastruktur. Dieser Treiber ist nützlich, wenn der andere Endpunkt ein Cisco-Router ist: Cisco mag GRE viel lieber als den anderen Linux-Tunneltreiber („IP-Tunneling“ oben). Außerdem erlaubt GRE die Weiterverteilung von Multicast durch den Tunnel.

14.1.12.6.1 IP: broadcast GRE over IP

CONFIG_NET_IPGRE_BROADCAST [=y] **[Y]**
Eine Anwendung von GRE/IP ist der Aufbau eines Broadcast-WAN (Wide Area Network), das wie ein normales Ethernet-LAN (Local Area Network) aussieht, aber über das gesamte Internet verteilt werden kann. Wenn Sie das tun wollen, sagen Sie hier und bei „IP-Multicast-Routing“ unten Y.

14.1.12.7 IP: multicast routing

CONFIG_IP_MROUTE [=y] **[Y]**
Dies wird verwendet, wenn Ihr Rechner als Router für IP-Pakete mit mehreren Zieladressen fungieren soll. Er wird für das MBONE benötigt, ein Netzwerk mit hoher Bandbreite über dem Internet, das Audio- und Videoübertragungen überträgt. Um dies zu tun, würden Sie wahrscheinlich das Programm `mrouted` ausführen. Wenn Sie davon noch nichts gehört haben, brauchen Sie es nicht.

14.1.12.7.1 IP: multicast policy routing

CONFIG_IP_MROUTE_MULTIPLE_TABLES [=y] **[Y]**
Normalerweise führt ein Multicast-Router einen Userspace-Daemon aus und entscheidet auf der Grundlage der Quell- und Zieladressen, was mit einem Multicast-Paket geschehen soll. Wenn Sie hier Y angeben, kann der Multicast-Router auch Schnittstellen und Paketmarkierungen berücksichtigen und mehrere Instanzen von Userspace-Dämonen gleichzeitig laufen lassen, von denen jeder eine einzelne Tabelle bearbeitet. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.7.2 IP: PIM-SM version 1 support

CONFIG_IP_PIMSM_V1 [=y] **[Y]**
Kernelseitige Unterstützung für Sparse Mode PIM (Protocol Independent Multicast) Version 1. Dieses Multicast-Routing-Protokoll ist weit verbreitet, da Cisco es unterstützt. Sie benötigen eine spezielle Software, um es zu verwenden (pimd-v1).
Weitere Informationen über PIM finden Sie unter `http://netweb.usc.edu/pim/`. Sagen Sie Y, wenn Sie PIM-SM v1 verwenden wollen. Beachten Sie, dass Sie hier N sagen können, wenn Sie nur Dense Mode PIM verwenden wollen.

14.1.12.7.3 IP: PIM-SM version 2 support

CONFIG_IP_PIMSM_V2 [=y] **[Y]**
Kernelseitige Unterstützung für Sparse Mode PIM Version 2. Um dies nutzen zu können, benötigen Sie einen experimentellen Routing-Daemon, der dies unterstützt (pimd oder gated-5). Dieses Routing-Protokoll ist nicht weit verbreitet, also sagen Sie N, es sei denn, Sie wollen damit spielen.

14.1.12.8 IP: TCP syncookie support

CONFIG_SYN_COOKIES [=y] **[Y]**
Normale TCP/IP-Netzwerke sind anfällig für einen Angriff, der als SSYN-Flooding”bekannt ist. Dieser Denial-of-Service-Angriff verhindert, dass legitime Remote-Benutzer während eines laufenden Angriffs eine Verbindung zu Ihrem Computer herstellen können, und erfordert vom Angreifer, der von einem beliebigen Ort im Internet aus operieren kann, nur sehr wenig Arbeit. SYN-Cookies bieten Schutz gegen diese Art von Angriffen. Wenn Sie hier ”Yëingeben, verwendet der TCP/IP-Stack ein kryptografisches Herausforderungsprotokoll, das als SSYN-Cookies”bekannt ist, um legitime Benutzer in die Lage zu versetzen, weiterhin eine Verbindung herzustellen, selbst wenn Ihr Rechner angegriffen wird. Die rechtmäßigen Benutzer brauchen ihre TCP/IP-Software nicht zu ändern; SYN-Cookies arbeiten für sie transparent. Technische Informationen über SYN-Cookies finden Sie unter `https://cr.yp.to/syncookies.html`. Wenn Sie SYN-geflutet werden, ist die vom Kernel gemeldete Quelladresse wahrscheinlich vom Angreifer gefälscht worden; sie wird nur als Hilfe bei der Rückverfolgung der Pakete zu ihrer tatsächlichen Quelle gemeldet und sollte nicht als absolute Wahrheit angesehen werden. SYN-Cookies können eine korrekte Fehlermeldung auf Clients verhindern, wenn der Server wirklich überlastet ist. Wenn dies häufig vorkommt, schalten Sie sie besser aus. Wenn Sie hier Y angeben, können Sie SYN-Cookies zur Laufzeit deaktivieren, indem Sie Y zu „/proc file system support“ und „Sysctl support“ unten angeben und den Befehl `echo 0 > /proc/sys/net/ipv4/tcp_syncookies` ausführen nachdem das /proc-Dateisystem eingehängt wurde. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.9 Virtual (secure) IP: tunneling

CONFIG_NET_IPVTI [=m] **[M]**

Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dies kann mit xfrm mode tunnel verwendet werden, um die Vorstellung eines sicheren Tunnels für IPSEC zu vermitteln und dann ein Routing-Protokoll darüber zu legen.

14.1.12.10 IP: Foo (IP protocols) over UDP

CONFIG_NET_FOU [=m] **[M]**

Mit Foo over UDP kann jedes IP-Protokoll direkt über UDP gekapselt werden, einschließlich Tunneln (IPIP, GRE, SIT). Durch die Verkapselung in UDP können Netzwerkmechanismen und Optimierungen für UDP (wie ECMP und RSS) genutzt werden, um einen besseren Service zu bieten.

14.1.12.11 IP: FOU encapsulation of IP tunnels

CONFIG_NET_FOU_IP_TUNNELS [=y] **[Y]**

Ermöglicht die Konfiguration von FOU- oder GUE-Kapselung für IP-Tunnel. Wenn diese Option aktiviert ist, können IP-Tunnel für die Verwendung von FOU- oder GUE-Kapselung konfiguriert werden.

14.1.12.12 IP: AH transformation

CONFIG_INET_AH [=m] **[M]**

Unterstützung für IPsec AH (Authentication Header). AH kann mit verschiedenen Authentifizierungsalgorithmen verwendet werden. Diese Option aktiviert nicht nur die AH-Unterstützung selbst, sondern auch die generischen Implementierungen der Algorithmen, die nach RFC 8221 implementiert werden MÜSSEN. Wenn Sie andere Algorithmen benötigen, müssen Sie diese in der Krypto-API aktivieren. Sie sollten auch beschleunigte Implementierungen aller benötigten Algorithmen aktivieren, sofern verfügbar. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.13 IP: ESP transformation

CONFIG_INET_ESP [=m] **[M]**

Unterstützung für IPsec ESP (Encapsulating Security Payload). ESP kann mit verschiedenen Verschlüsselungs- und Authentifizierungsalgorithmen verwendet werden. Diese Option aktiviert nicht nur die ESP-Unterstützung selbst, sondern auch die generischen Implementierungen der Algorithmen, die nach RFC 8221 implementiert werden MÜSSEN. Wenn Sie andere Algorithmen benötigen, müssen Sie diese in der Krypto-API aktivieren. Sie sollten auch beschleunigte Implementierungen aller benötigten Algorithmen aktivieren, sofern verfügbar. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.13.1 IP: ESP transformation offload

CONFIG_INET_ESP_OFFLOAD [=m] **[M]**

Unterstützung für ESP-Transformationsoffload. Dies ist nur dann sinnvoll, wenn das System wirklich IPsec verwendet und einen hohen Durchsatz erzielen möchte. Ein typisches Desktop-System braucht dies nicht, selbst wenn es IPsec verwendet. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.13.2 IP: ESP in TCP encapsulation (RFC 8229)

CONFIG_INET_ESPINTCP [=y] **[Y]**

Unterstützung für die RFC 8229-Kapselung von ESP und IKE über TCP/IPv4-Sockets. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.14 IP: IPComp transformation

CONFIG_INET_IPCOMP [=m] **[M]**

Unterstützung für das IP Payload Compression Protocol (IPComp) (RFC 3173), das normalerweise für IPsec benötigt wird.
Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.15 INET: socket monitoring interface

CONFIG_INET_DIAG [=m] **[M]**

Unterstützung für die INET (TCP, DCCP usw.) Socket-Überwachungsschnittstelle, die von nativen Linux-Tools wie `ss` verwendet wird. `ss` ist in iproute2 enthalten und kann derzeit heruntergeladen werden

unter: `http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2`
Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.15.1 UDP: socket monitoring interface

CONFIG_INET_UDP_DIAG [=m] **[M]**
Unterstützung für die UDP-Socket-Überwachungsschnittstelle, die vom Tool `ss` verwendet wird. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.15.2 RAW: socket monitoring interface

CONFIG_INET_RAW_DIAG [=m] **[M]**
Unterstützung für die vom `ss`-Tool verwendete RAW-Socket-Überwachungsschnittstelle. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.15.3 INET: allow privileged process to administratively close sockets

CONFIG_INET_DIAG_DESTROY [=y] **[Y]**
Stellt eine SOCK_DESTROY-Operation zur Verfügung, die es privilegierten Prozessen (z. B. einem Verbindungsmanager oder einem Netzwerkverwaltungsprogramm wie `ss`) ermöglicht, von anderen Prozessen geöffnete Sockets zu schließen. Das Schließen eines Sockets auf diese Weise unterbricht alle blockierenden Lese-/Schreib-/Verbindungsoperationen auf dem Socket und bewirkt, dass sich zukünftige Socket-Aufrufe so verhalten, als ob der Socket getrennt worden wäre. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.16 TCP: advanced congestion control →

CONFIG_TCP_CONG_ADVANCED [=y] **[Y]**
Unterstützung für die Auswahl verschiedener TCP-Staukontrollmodule. Fast alle Benutzer können hier sicher nein sagen, und es wird eine sichere Standardauswahl getroffen (CUBIC mit neuem Reno als Fallback). Wenn Sie unsicher sind, sagen Sie N.

14.1.12.16.1 Binary Increase Congestion (BIC) control

CONFIG_TCP_CONG_BIC [=m] **[M]**
BIC-TCP ist eine rein senderseitige Änderung, die eine lineare RTT-Fairness bei großen Fenstern gewährleistet und gleichzeitig Skalierbarkeit und begrenzte TCP-Freundlichkeit bietet. Das Protokoll kombiniert zwei Verfahren, die additive Erhöhung und die binäre Sucherhöhung. Bei großen Überlastungsfenstern gewährleistet die additive Erhöhung mit einem großen Inkrement eine lineare RTT-Fairness sowie eine gute Skalierbarkeit. Bei kleinen Überlastungsfenstern sorgt die binäre Sucherhöhung für TCP-Freundlichkeit.
Siehe `http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/`

14.1.12.16.2 CUBIC TCP

CONFIG_TCP_CONG_CUBIC [=y] **[Y]**
Dies ist die Version 2.0 von BIC-TCP, die neben anderen Techniken eine kubische Wachstumsfunktion verwendet.
Siehe `http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/cubic-paper.pdf`

14.1.12.16.3 TCP Westwood+

CONFIG_TCP_CONG_WESTWOOD [=m] **[M]**
TCP Westwood+ ist eine absenderseitige Modifikation des TCP-Reno-Protokollstapels, die die Leistung der TCP-Überlastungssteuerung optimiert. Es basiert auf einer Ende-zu-Ende-Bandbreitenschätzung, um das Überlastungsfenster und den Schwellenwert für den langsamen Start nach einer Überlastungsepisode festzulegen. Auf der Grundlage dieser Schätzung legt TCP Westwood+ adaptiv einen Schwellenwert für den langsamen Start und ein Überlastungsfenster fest, das die zum Zeitpunkt des Auftretens der Überlastung genutzte Bandbreite berücksichtigt. TCP Westwood+ erhöht die Fairness gegenüber TCP Reno in kabelgebundenen Netzen und den Durchsatz über drahtlose Verbindungen erheblich.

14.1.12.16.4 H-TCP

CONFIG_TCP_CONG_HTCP [=m] **[M]**
H-TCP ist eine nur sendeseitige Modifikation des TCP-Reno-Protokollstapels, die die Leistung der TCP-Überlastungssteuerung für Hochgeschwindigkeitsnetzverbindungen optimiert. Es verwendet einen Mode-

switch, um die Alpha- und Beta-Parameter von TCP Reno auf der Grundlage der Netzbedingungen und in einer Weise zu ändern, die gegenüber anderen Reno- und H-TCP-Datenströmen fair ist.

14.1.12.16.5 High Speed TCP

CONFIG_TCP_CONG_HSTCP [=m] **[M]**
Sally Floyds High Speed TCP (RFC 3649) Staukontrolle. Eine Modifikation des TCP-Überlastungssteuerungsmechanismus zur Verwendung mit großen Überlastungsfenstern. In einer Tabelle wird angegeben, um wie viel das Überlastungsfenster vergrößert werden soll, wenn eine ACK empfangen wird. Für weitere Einzelheiten siehe `https://www.icir.org/floyd/hstcp.html`

14.1.12.16.6 TCP-Hybla congestion control algorithm

CONFIG_TCP_CONG_HYBLA [=m] **[M]**
TCP-Hybla ist eine Änderung, die nur auf der Absenderseite vorgenommen wird, um die Benachteiligung von Verbindungen mit langen Übertragungszeiten und großen Bandbreiten zu beseitigen, z. B. wenn Satellitenverbindungen beteiligt sind, insbesondere wenn sie einen gemeinsamen Engpass mit normalen terrestrischen Verbindungen teilen.

14.1.12.16.7 TCP Vegas

CONFIG_TCP_CONG_VEGAS [=m] **[M]**
TCP Vegas ist eine absenderseitige Änderung von TCP, die den Beginn einer Überlastung durch Schätzung der Bandbreite vorwegnimmt. TCP Vegas passt die Übertragungsrate durch Änderung des Überlastungsfensters an. TCP Vegas sollte weniger Paketverluste verursachen, ist aber nicht so aggressiv wie TCP Reno.

14.1.12.16.8 TCP NV

CONFIG_TCP_CONG_NV [=m] **[M]**
TCP NV ist ein Nachfolger von TCP Vegas. Es wurde geändert, um mit 10G-Netzen, Messrauschen durch LRO, GRO und Unterbrechungskoaleszenz fertig zu werden. Außerdem wird der cwnd-Wert nicht mehr linear, sondern multiplikativ verringert.
Es ist zu beachten, dass die Stauvermeidung (cwnd wird verringert, wenn die Anzahl der Pakete in der Warteschlange steigt) im Allgemeinen nicht mit der Staukontrolle (cwnd wird nur verringert, wenn es zu Paketverlusten kommt) koexistieren kann, da die Fairness nicht gewährleistet ist. Ein Szenario, in dem sie sicher koexistieren können, ist, wenn die CA-Flüsse RTTs $\ll$ CC-Flüsse RTTs haben. Für weitere Einzelheiten siehe `http://www.brakmo.org/networking/tcp-nv/`

14.1.12.16.9 Scalable TCP

CONFIG_TCP_CONG_SCALABLE [=m] **[M]**
Scalable TCP ist eine Änderung von TCP nur auf der Absenderseite, die einen MIMD-Algorithmus zur Staukontrolle verwendet, der einige nette Skalierungseigenschaften hat, obwohl er bekanntermaßen Probleme mit der Fairness hat. Siehe `http://www.deneholme.net/tom/scalable/`

14.1.12.16.10 TCP Low Priority

CONFIG_TCP_CONG_LP [=m] **[M]**
TCP Low Priority (TCP-LP), ein verteilter Algorithmus, dessen Ziel es ist, nur die überschüssige Bandbreite des Netzes im Vergleich zum „fairen Anteil“ der Bandbreite, wie er von TCP angestrebt wird, zu nutzen. Siehe `http://www-ece.rice.edu/networks/TCP-LP/`

14.1.12.16.11 TCP Veno

CONFIG_TCP_CONG_VENO [=m] **[M]**
TCP Veno ist eine rein senderseitige Erweiterung von TCP, um einen besseren Durchsatz in drahtlosen Netzen zu erzielen. TCP Veno nutzt die Zustandsunterscheidung, um die schwierige Beurteilung der Paketverlustart zu umgehen. TCP Veno verkleinert das Überlastungsfenster als Reaktion auf zufällige Paketverluste. Siehe `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1177186`

14.1.12.16.12 YeAH TCP

CONFIG_TCP_CONG_YEAH [=m] [**M**]
YeAH-TCP ist ein absenderseitiger Hochgeschwindigkeits-TCP-Überlastungskontrollalgorithmus, der einen gemischten Verlust-/Verzögerungsansatz zur Berechnung des Überlastungsfensters verwendet. Seine Entwurfsziele sind hohe Effizienz, interne, RTT- und Reno-Fairness, Widerstandsfähigkeit gegenüber Verbindungsverlusten und eine möglichst geringe Belastung der Netzelemente.
Weitere Einzelheiten finden Sie hier: `http://wil.cs.caltech.edu/pfldnet2007/paper/YeAH_TCP.pdf` or Link via www.gdt.id.au

14.1.12.16.13 TCP Illinois

CONFIG_TCP_CONG_ILLINOIS [=m] [**M**]
TCP-Illinois ist eine absenderseitige Modifikation von TCP Reno für Hochgeschwindigkeitsverbindungen mit langer Verzögerung. Es nutzt die Round-Trip-Zeit, um die Alpha- und Beta-Parameter anzupassen, um einen höheren durchschnittlichen Durchsatz zu erreichen und Fairness zu wahren. Für weitere Einzelheiten siehe: `http://www.ews.uiuc.edu/~shaoliu/tcpillinois/index.html`

14.1.12.16.14 DataCenter TCP (DCTCP)

CONFIG_TCP_CONG_DCTCP [=m] [**M**]
DCTCP nutzt die explizite Überlastungsanzeige (Explicit Congestion Notification, ECN) im Netz, um den Endhosts ein Multi-Bit-Feedback zu geben. Es wurde entwickelt, um Folgendes zu bieten:

- Hohe Burst-Toleranz (Incast aufgrund von Partition/Aggregat),
- Geringe Latenz (kurze Flüsse, Abfragen),
- hohen Durchsatz (kontinuierliche Datenaktualisierungen, große Dateiübertragungen) mit handelsüblichen, flach gepufferten Switches.

Alle Switches im Rechenzentrumsnetz, auf denen DCTCP läuft, müssen die ECN-Kennzeichnung unterstützen und so konfiguriert sein, dass sie bei Erreichen bestimmter Switch-Pufferschwellenwerte gekennzeichnet werden. Die Standardheuristik für die ECN-Markierungsschwelle für DCTCP auf Switches liegt bei 20 Paketen (30 KB) bei 1 Gbps und 65 Paketen (≈ 100 KB) bei 10 Gbps, muss aber möglicherweise noch weiter optimiert werden.
Weitere Einzelheiten siehe:
`http://simula.stanford.edu/~alizade/Site/DCTCP_files/dctcp-final.pdf`

14.1.12.16.15 CAIA Delay-Gradient (CDG)

CONFIG_TCP_CONG_CDG [=m] [**M**]
CAIA Delay-Gradient (CDG) ist eine TCP-Überlastungskontrolle, die den TCP-Sender modifiziert, um:

- o Verwendung des Verzögerungsgradienten als Überlastungssignal.
- o mit einer durchschnittlichen Wahrscheinlichkeit, die unabhängig von der RTT ist, zurückzufahren.
- o mit Datenströmen zu koexistieren, die eine verlustbasierte Staukontrolle verwenden.
- o Paketverluste zu tolerieren, die nicht mit der Überlastung zusammenhängen.

Für weitere Einzelheiten siehe:
D.A. Hayes und G. Armitage. "Revisiting TCP congestion control using delay gradients".
In Networking 2011. Preprint: `http://goo.gl/No3vdg`

14.1.12.16.16 BBR TCP

CONFIG_TCP_CONG_BBR [=m] [**M**]
BBR (Bottleneck Bandwidth and RTT) Die TCP-Überlastungssteuerung zielt darauf ab, die Netzauslastung zu maximieren und Warteschlangen zu minimieren. Sie erstellt ein explizites Modell der Bottleneck-Zustellrate und der Umlaufverzögerung des Pfades. Sie toleriert Paketverluste und Verzögerungen, die nicht mit der Überlastung zusammenhängen. Es kann über LAN-, WAN-, Mobilfunk-, WLAN- oder Kabelmodem-Verbindungen arbeiten. Es kann mit Datenströmen koexistieren, die eine verlustbasierte Staukontrolle verwenden, und es kann mit flachen Puffern, tiefen Puffern, Bufferbloat, Policern oder AQM-Schemata arbeiten, die kein Verzögerungssignal liefern. Es erfordert den `fq` („Fair Queue“) Pacing Packet Scheduler.

14.1.12.16.17 Default TCP congestion control () →
Wählen Sie die TCP-Überlastungssteuerung aus, die standardmäßig für alle Verbindungen verwendet werden soll.

14.1.12.16.17.1 Cubic
CONFIG_DEFAULT_CUBIC [=y] **[Y]**
Für diese Option ist keine Hilfe vorhanden.

14.1.12.16.17.2 Reno
CONFIG_DEFAULT_RENO [=n] **[N]**
Für diese Option ist keine Hilfe vorhanden.

14.1.12.17 TCP: MD5 Signature Option support (RFC 2385)
CONFIG_TCP_MD5SIG [=y] **[Y]**
RFC2385 spezifiziert eine Methode zum MD5-Schutz von TCP-Sitzungen. Die wichtigste (einzige?) Anwendung ist der Schutz von BGP-Sitzungen zwischen Core-Routern im Internet. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18 The IPv6 protocol
CONFIG_IPV6 [=y] **[Y]**
Unterstützung für die IP Version 6 (IPv6).
Allgemeine Informationen über IPv6 finden Sie unter `https://en.wikipedia.org/wiki/IPv6`. Spezielle Informationen über IPv6 unter Linux finden Sie unter Documentation/networking/ipv6.rst und lesen Sie das HOWTO unter `https://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/`
Um diese Protokollunterstützung als Modul zu kompilieren, wählen Sie hier M: Das Modul wird `ipv6` heißen.

14.1.12.18.1 IPv6: Router Preference (RFC 4191) support
CONFIG_IPV6_IOAM6_LWTUNNEL [=y] **[Y]**
Die Router-Präferenz ist eine optionale Erweiterung der Router-Advertisement-Nachricht, die die Fähigkeit der Hosts verbessert, einen geeigneten Router auszuwählen, insbesondere wenn die Hosts in einem Netz mit mehreren Hosts untergebracht sind. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.1.1 IPv6: Router Information (RFC 4191) support
CONFIG_IPV6_ROUTE_INFO [=y] **[Y]**
Unterstützung von Routeninformationen. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.2 IPv6: Enable RC 4429 Optimistic DAD
CONFIG_IPV6_OPTIMISTIC_DAD [=y] **[Y]**
Unterstützung für die optimistische Erkennung von doppelten Adressen. Dadurch können automatisch konfigurierte Adressen schneller verwendet werden. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.3 IPv6: AH transformation
CONFIG_INET6_AH [=m] **[M]**
Unterstützung für IPsec AH (Authentication Header). AH kann mit verschiedenen Authentifizierungsalgorithmen verwendet werden. Diese Option aktiviert nicht nur die AH-Unterstützung selbst, sondern auch die generischen Implementierungen der Algorithmen, die nach RFC 8221 implementiert werden MÜSSEN. Wenn Sie andere Algorithmen benötigen, müssen Sie diese in der Krypto-API aktivieren. Sie sollten auch beschleunigte Implementierungen aller benötigten Algorithmen aktivieren, sofern verfügbar. Wenn Sie unsicher sind, sagen Sie Y für Ja.

14.1.12.18.4 IPv6: ESP transformation
CONFIG_INET6_ESP [=m] **[M]**
Unterstützung für IPsec ESP (Encapsulating Security Payload). ESP kann mit verschiedenen Verschlüsselungs- und Authentifizierungsalgorithmen verwendet werden. Diese Option aktiviert nicht nur die ESP-Unterstützung selbst, sondern auch die generischen Implementierungen der Algorithmen, die nach RFC 8221 implementiert werden MÜSSEN. Wenn Sie andere Algorithmen benötigen, müssen Sie diese

in der Krypto-API aktivieren. Sie sollten auch beschleunigte Implementierungen aller benötigten Algorithmen aktivieren, sofern verfügbar. Wenn Sie unsicher sind, sagen Sie Y für Ja.

14.1.12.18.4.1 IPv6: ESP transformation offload

CONFIG_INET6_ESP [=m] **[M]**
Unterstützung für ESP-Transformationsoffload. Dies ist nur dann sinnvoll, wenn das System wirklich IPsec verwendet und einen hohen Durchsatz erzielen möchte. Ein typisches Desktop-System braucht dies nicht, selbst wenn es IPsec verwendet. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.4.2 IPv6: ESP in TCP encapsulation (RFC 8229)

CONFIG_INET6_ESPINTCP [=y] **[Y]**
Unterstützung für die RFC 8229-Kapselung von ESP und IKE über TCP/IPv6-Sockets. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.5 IPv6: IPComp transformation

CONFIG_INET6_IPCOMP [=m] **[M]**
Unterstützung für IP Payload Compression Protocol (IPComp) (RFC 3173), typischerweise erforderlich für IPsec. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.18.6 IPv6: Mobility

CONFIG_IPV6_MIP6 [=m] **[M]**
Unterstützung für IPv6-Mobilität, beschrieben in RFC 3775.
Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.7 IPv6: Identifier Locator Addressing (ILA)

CONFIG_IPV6_ILA [=m] **[M]**
Unterstützung für IPv6 Identifier Locator Addressing (ILA). ILA ist ein Mechanismus zur Netzwerkvirtualisierung ohne Verkapselung. Das Grundkonzept von ILA besteht darin, dass wir eine IPv6-Adresse in einen 64-Bit-Locator und einen 64-Bit-Identifier aufteilen. Der Bezeichner ist die Identität einer Entität in der Kommunikation („who“) und der Locator drückt den Standort der Entität („where“) aus. ILA kann unter Verwendung der Option `encap ila` mit dem Befehl `ip -6 route` konfiguriert werden.
ILA wird in `https://tools.ietf.org/html/draft-herbert-nvo3-ila-00` beschrieben. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.8 Virtual (secure) IPv6: tunneling

CONFIG_IPV6_VTI [=m] **[M]**
Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dies kann mit xfrm mode tunnel verwendet werden, um die Vorstellung eines sicheren Tunnels für IPSEC zu vermitteln und dann ein Routing-Protokoll darüber zu legen.

14.1.12.18.9 IPv6: IPv6-in-IPv4 tunnel (SIT driver)

CONFIG_IPV6_SIT [=m] **[M]**
Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dieser Treiber implementiert die Einkapselung von IPv6 in IPv4-Pakete. Dies ist nützlich, wenn Sie zwei IPv6-Netzwerke über einen reinen IPv4-Pfad verbinden wollen. Wenn Sie hier M sagen, wird ein Modul namens `sit` erzeugt. Wenn Sie unsicher sind, sagen Sie Y.

14.1.12.18.9.1 IPv6: IPv6 Rapid Deployment (6RD)

CONFIG_IPV6_SIT_6RD [=y] **[Y]**
IPv6 Rapid Deployment (6rd; draft-ietf-softwire-ipv6-6rd) baut auf Mechanismen von 6to4 (RFC 3056) auf, um einen Dienstanbieter in die Lage zu versetzen, IPv6-Unicast-Dienste schnell an IPv4-Standorten einzurichten, für die er Kundengeräte bereitstellt. Wie 6to4 verwendet es zustandsloses IPv6 in einer IPv4-Kapselung, um eine reine IPv4-Netzinfrastruktur zu durchqueren. Im Gegensatz zu 6to4 verwendet ein 6rd-Dienstanbieter ein eigenes IPv6-Präfix anstelle des festen 6to4-Präfixes. Wenn diese Option aktiviert ist, bietet der SIT-Treiber 6rd-Funktionalität, indem er eine zusätzliche ioctl-API zur Konfiguration des

IPv6-Präfixes anstelle des statischen 2002::/16 für 6to4 bereitstellt.
Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.10 IPv6: IP-in-IPv6 tunnel (RFC 2473)

CONFIG_IPV6_TUNNEL [=m] **[M]**
Unterstützung für IPv6-in-IPv6- und IPv4-in-IPv6-Tunnel, beschrieben in RFC 2473. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.11 IPv6: GRE tunnel (RFC 2473)

CONFIG_IPV6_GRE [=m] **[M]**
Tunneling bedeutet, dass Daten eines Protokolltyps in ein anderes Protokoll eingekapselt und über einen Kanal gesendet werden, der das einkapselnde Protokoll versteht. Dieser spezielle Tunneling-Treiber implementiert GRE (Generic Routing Encapsulation) und ermöglicht derzeit die Verkapselung von IPv4 oder IPv6 über eine bestehende IPv6-Infrastruktur. Dieser Treiber ist nützlich, wenn der andere Endpunkt ein Cisco-Router ist: Cisco mag GRE viel lieber als den anderen Linux-Tunneltreiber („IP-Tunneling“ oben). Außerdem erlaubt GRE die Umverteilung von Multicast durch den Tunnel.
Wenn Sie hier M sagen, wird ein Modul namens `ip6_gre` erzeugt. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.12 IPv6: Multiple Routing Tables

CONFIG_IPV6_MULTIPLE_TABLES [=y] **[Y]**
Unterstützung mehrerer Routing-Tabellen.

14.1.12.18.12.1 IPv6: source address based routing

CONFIG_IPV6_SUBTREES [=y] **[Y]**
Aktivieren Sie das Routing nach Quelladresse oder Präfix.
Die Zieladresse ist immer noch der primäre Routing-Schlüssel, so dass das Mischen von normalen und quellpräfixspezifischen Routen in derselben Routing-Tabelle manchmal zu einem unbeabsichtigten Routing-Verhalten führen kann. Dies kann vermieden werden, indem unterschiedliche Routing-Tabellen für die normalen und die quellpräfixspezifischen Routen definiert werden.
Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.13 IPv6: multicast routing

CONFIG_IPV6_MROUTE [=y] **[Y]**
Unterstützung der IPv6-Multicast-Weiterleitung. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.13.1 IPv6: multicast policy routing S S

CONFIG_IPV6_MROUTE_MULTIPLE_TABLES [=y] **[Y]**
Normalerweise führt ein Multicast-Router einen Userspace-Daemon aus und entscheidet auf der Grundlage der Quell- und Zieladressen, was mit einem Multicast-Paket geschehen soll. Wenn Sie hier Y angeben, kann der Multicast-Router auch Schnittstellen und Paketmarkierungen berücksichtigen und mehrere Instanzen von Userspace-Dämonen gleichzeitig laufen lassen, von denen jeder eine einzelne Tabelle bearbeitet. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.13.2 IPv6: multicast policy routing S S

CONFIG_IPV6_MROUTE_MULTIPLE_TABLES [=y] **[Y]**
Unterstützung für das IPv6-PIM-Multicast-Routing-Protokoll PIM-SMv2. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.14 IPv6: Segment Routing Header encapsulation support

CONFIG_IPV6_SEG6_LWTUNNEL [=y] **[Y]**
Unterstützung für die Einkapselung von Paketen in einen äußeren IPv6-Header und einen Segment-Routing-Header unter Verwendung des leichtgewichtigen Tunnelmechanismus. Aktivieren Sie auch die Unterstützung für die erweiterte lokale Verarbeitung von SRv6-Paketen auf der Grundlage ihres aktiven Segments. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.15 IPv6: Segment Routing HMAC support

CONFIG_IPV6_SEG6_HMAC [=y] **[Y]**
Unterstützung für die Erzeugung von HMAC-Signaturen und die Überprüfung von SR-aktivierten Paketen. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.16 IPv6: RPL Source Routing Header support

CONFIG_IPV6_RPL_LWTUNNEL [=y] **[Y]**
Unterstützung für RFC 6554 RPL Source Routing Header unter Verwendung des Lightweight-Tunnel-Mechanismus. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.18.17 IPv6: IOAM Pre-allocated Trace insertion support

CONFIG_IPV6_IOAM6_LWTUNNEL [=y] **[Y]**
Unterstützung für das Einfügen von IOAM Pre-allocated Trace Header unter Verwendung des Lightweight-Tunnel-Mechanismus. Wenn Sie unsicher sind, sagen Sie N.

14.1.12.19 NetLabel subsystem support

CONFIG_NETLABEL [=y] **Y**
NetLabel bietet Unterstützung für explizite Netzwerk-Paketkennzeichnungsprotokolle wie CIPSO und RIPSO. Weitere Informationen finden Sie unter Documentation/netlabel sowie im NetLabel SourceForge-Projekt für Konfigurationswerkzeuge und zusätzliche Dokumentation.
∗ `https://github.com/netlabel/netlabel_tools` Wenn Sie unsicher sind, sagen Sie N.

14.1.12.20 MPTCP: Multipath TCP

CONFIG_MPTCP [=y] **Y**
Multipath TCP (MPTCP)-Verbindungen senden und empfangen Daten über mehrere Subflows, um mehrere Netzwerkpfade zu nutzen. Jeder Subflow verwendet das TCP-Protokoll, und die TCP-Optionen enthalten Header-Informationen für MPTCP.

14.1.12.20.1 MPTCP: IPv6 support for Multipath TCP

CONFIG_MPTCP_IPV6 [=y] **[Y]**
Für diese Option gibt es keine Hilfe.

14.1.13 Security Marking

CONFIG_NETWORK_SECMARK [=y] **[Y]**
Dies ermöglicht die Sicherheitsmarkierung von Netzwerkpaketen, ähnlich wie bei nfmark, aber für Sicherheitszwecke. Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollen, antworten Sie mit N.

14.1.14 Timestamping in PHY devices

CONFIG_NETWORK_PHY_TIMESTAMPING [=y] **[Y]**
Dies ermöglicht die Zeitstempelung von Netzwerkpaketen durch PHYs (oder andere MII-Bus-Snooping-Geräte) mit Hardware-Zeitstempelfunktionen. Diese Option fügt einen gewissen Overhead in den Sende- und Empfangswegen hinzu. Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollen, antworten Sie mit N.

14.1.15 Network packet filtering framework (Netfilter) →

CONFIG_NETFILTER [=y] **[Y]**
Netfilter ist ein Framework zum Filtern und Verarbeiten von Netzwerkpaketen, die Ihren Linux-Rechner durchlaufen. Die häufigste Anwendung der Paketfilterung ist der Einsatz Ihres Linux-Rechners als Firewall zum Schutz eines lokalen Netzwerks vor dem Internet. Die Art von Firewall, die durch diese Kernelunterstützung bereitgestellt wird, wird als „Paketfilter“ bezeichnet, was bedeutet, dass sie einzelne Netzwerkpakete auf der Grundlage von Typ, Quelle, Ziel usw. zurückweisen kann. Die andere Art von Firewall, eine „proxy-basierte“ Firewall, ist sicherer, aber aufdringlicher und mühsamer einzurichten; sie untersucht den Netzwerkverkehr viel genauer, verändert ihn und hat Kenntnisse über die höheren Protokolle, die ein Paketfilter nicht hat. Außerdem erfordern proxy-basierte Firewalls oft Änderungen an den Programmen, die auf den lokalen Clients laufen. Proxy-basierte Firewalls brauchen keine Unterstützung

durch den Kernel, aber sie werden oft mit einem Paketfilter kombiniert, der nur funktioniert, wenn man hier Y sagt.
Sie sollten hier auch Y angeben, wenn Sie Ihren Linux-Rechner als Gateway zum Internet für ein lokales Netzwerk von Rechnern ohne global gültige IP-Adresse verwenden wollen. Dies nennt man „masquerading“: Wenn einer der Computer in Ihrem lokalen Netzwerk etwas nach außen senden möchte, kann sich Ihre Box als dieser Computer „maskieren“, d. h. sie leitet den Datenverkehr an das vorgesehene Ziel nach außen weiter, verändert aber die Pakete so, dass es so aussieht, als kämen sie von der Firewall-Box selbst. Das funktioniert in beide Richtungen: Wenn der externe Rechner antwortet, leitet die Linux-Box den Datenverkehr stillschweigend an den richtigen lokalen Rechner weiter.
Auf diese Weise sind die Computer in Ihrem lokalen Netz für die Außenwelt völlig unsichtbar, obwohl sie die Außenwelt erreichen und Antworten empfangen können. Es ist sogar möglich, global sichtbare Server von einem maskierten lokalen Netzwerk aus zu betreiben, indem man einen Mechanismus namens Portforwarding verwendet. Masquerading wird oft auch als NAT (Network Address Translation) bezeichnet.
Eine weitere Anwendung von Netfilter ist das transparente Proxying: Wenn ein Rechner im lokalen Netzwerk versucht, eine Verbindung zu einem externen Host herzustellen, kann Ihr Linux-System den Datenverkehr transparent an einen lokalen Server weiterleiten, in der Regel einen Caching-Proxy-Server.
Eine weitere Verwendung von Netfilter ist der Aufbau einer Bridging-Firewall. Wenn Sie eine Bridge mit aktivierter Netzwerk-Paketfilterung verwenden, kann iptables den überbrückten Verkehr „sehen“. Für die Filterung des unteren Netzwerks und der Ethernet-Protokolle über die Brücke, verwenden Sie ebtables (unter bridge netfilter configuration). Für netfilter gibt es verschiedene Module, die die bisherigen Mechanismen Masquerading (ipmasqadm), Paketfilterung (ipchains), transparentes Proxying und Portforwarding ersetzen. Bitte sehen Sie <file:Documentation/Changes> unter „iptables“ nach, wo diese Pakete zu finden sind.

14.1.15.1 Advanced netfilter configuration
CONFIG_NETFILTER_ADVANCED [=y] **[Y]**
Wenn Sie hier Y angeben, können Sie zwischen allen Netzfiltermodulen wählen. Wenn Sie N sagen, werden die ungewöhnlicheren nicht angezeigt, und die grundlegenden Module, die von den meisten Benutzern benötigt werden, werden standardmäßig mit „M“ angezeigt. Wenn Sie unsicher sind, sagen Sie Y.

14.1.15.1.1 Bridged IP/ARP packets filtering
CONFIG_BRIDGE_NETFILTER [=m] **[M]**
Wenn Sie diese Option aktivieren, kann arptables bzw. iptables überbrückten ARP- bzw. IP-Verkehr sehen. Wenn Sie eine Bridging-Firewall wollen, sollten Sie diese Option aktivieren. Durch das Aktivieren oder Deaktivieren dieser Option wird ebtables nicht aktiviert oder deaktiviert.
Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2 Core Netfilter Configuration →
(Kern-Netfilter-Konfiguration)

14.1.15.2.1 Netfilter ingress support
CONFIG_NETFILTER_INGRESS [=y] **[Y]**
Damit können Sie Pakete bereits am Eingang über die Netfilter-Infrastruktur klassifizieren.

14.1.15.2.2 Netfilter egress support
CONFIG_NETFILTER_EGRESS [=y] **[Y]**
Damit können Sie Pakete vor der Übertragung über die Netfilter-Infrastruktur klassifizieren.

14.1.15.2.3 Netfilter base hook dump support
CONFIG_NETFILTER_NETLINK_HOOK [=m] **[M]**
Wenn diese Option aktiviert ist, unterstützt der Kernel die Auflistung der Basis-Netzfilter-Hooks über NFNETLINK. Dies ist hilfreich für die Fehlersuche.

14.1.15.2.4 Netfilter NFACCT over NFNETLINK interface
CONFIG_NETFILTER_NETLINK_ACCT [=m] **[M]**
Wenn diese Option aktiviert ist, unterstützt der Kernel die erweiterte Abrechnung über NFNETLINK.

14.1.15.2.5 Netfilter NFQUEUE over NFNETLINK interface

CONFIG_NETFILTER_NETLINK_QUEUE [=m] **[M]**
Wenn diese Option aktiviert ist, unterstützt der Kernel die Warteschlangenbildung für Pakete über NFNETLINK.

14.1.15.2.6 Netfilter LOG over NFNETLINK interface

CONFIG_NETFILTER_NETLINK_LOG [=m] **[M]**
Wenn diese Option aktiviert ist, bietet der Kernel Unterstützung für die Protokollierung von Paketen über NFNETLINK. Dadurch werden die bestehenden ipt_ULOG- und ebg_ulog-Mechanismen überflüssig und es ist auch geplant, die alten syslog-basierten ipt_LOG- und ip6t_LOG-Module zu ersetzen.

14.1.15.2.7 Netfilter OSF over NFNETLINK interface

CONFIG_NETFILTER_NETLINK_OSF [=m] **[M]**
Wenn diese Option aktiviert ist, unterstützt der Kernel den passiven OS-Fingerprint über NFNETLINK.

14.1.15.2.8 Netfilter connection tracking support

CONFIG_NF_CONNTRACK [=m] **[M]**
Die Verbindungsverfolgung zeichnet auf, welche Pakete Ihren Rechner durchlaufen haben, um herauszufinden, wie sie zu Verbindungen zusammenhängen. Dies ist erforderlich, um Masquerading oder andere Arten der Netzwerkadressübersetzung durchzuführen. Es kann auch verwendet werden, um die Paketfilterung zu verbessern (siehe „Unterstützung von Verbindungsstatusübereinstimmungen“ unten). Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.9 Syslog packet logging

CONFIG_NF_LOG_SYSLOG [=m] **[M]**
Diese Option aktiviert die Unterstützung für die Paketprotokollierung über Syslog. Sie unterstützt IPv4, IPV6, ARP und gängige Transportprotokolle wie TCP und UDP. Dies ist eine einfachere, aber weniger flexible Protokollierungsmethode im Vergleich zu CONFIG_NETFILTER_NETLINK_LOG. Wenn beide aktiviert sind, kann das zu verwendende Backend zur Laufzeit mit Hilfe von sysctl-Tunables pro Adressfamilie konfiguriert werden.

14.1.15.2.10 Connection mark tracking support

CONFIG_NF_CONNTRACK_MARK [=y] **[Y]**
Diese Option aktiviert die Unterstützung für Verbindungsmarkierungen, die vom Ziel „CONNMARK“ und der Übereinstimmung „connmark“ verwendet werden. Ähnlich wie der Markierungswert von Paketen, aber dieser Markierungswert wird in der conntrack-Sitzung statt in den einzelnen Paketen gespeichert.

14.1.15.2.11 Connection tracking security mark support

CONFIG_NF_CONNTRACK_SECMARK [=y] **[Y]**
Mit dieser Option können Sicherheitsmarkierungen auf Verbindungen angewendet werden. Normalerweise werden sie von Paketen, die das CONNSECMARK-Ziel verwenden, auf Verbindungen kopiert und von Verbindungen auf Pakete mit demselben Ziel zurückkopiert, wobei die Pakete ursprünglich über SECMARK gekennzeichnet wurden.
Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.12 Connection tracking zones

CONFIG_NF_CONNTRACK_ZONES [=y] **[Y]**
Mit dieser Option wird die Unterstützung für Zonen zur Verfolgung von Verbindungen aktiviert. Normalerweise muss jede Verbindung eine eindeutige systemweite Identität haben. Zonen für die Verbindungsverfolgung ermöglichen es, dass mehrere Verbindungen dieselbe Identität verwenden, solange sie in verschiedenen Zonen enthalten sind. Wenn Sie unsicher sind, sagen Sie ‘N’.

14.1.15.2.13 Supply CT list in procfs (OBSOLETE)

CONFIG_NF_CONNTRACK_PROCFS [=y] **[Y]**
Mit dieser Option kann die Liste der bekannten Conntrack-Einträge in procfs unter net/netfilter/nf_conntrack angezeigt werden. Diese Option wird als veraltet betrachtet, da das Werkzeug conntrack(8), das Netlink benutzt, verwendet wird.

14.1.15.2.14 Connection tracking events

CONFIG_NF_CONNTRACK_EVENTS [=y] **[Y]**
Wenn diese Option aktiviert ist, stellt der Code für die Verbindungsüberwachung eine Benachrichtigungskette zur Verfügung, die von anderem Kernel-Code verwendet werden kann, um über Änderungen im Status der Verbindungsüberwachung informiert zu werden.
Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.15 Connection tracking timeout

CONFIG_NF_CONNTRACK_TIMEOUT [=y] **[Y]**
Diese Option aktiviert die Unterstützung für die Timeout-Erweiterung der Verbindungsverfolgung. Damit können Sie Zeitüberschreitungsrichtlinien an den Datenfluss über das CT-Ziel anhängen.
Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.16 Connection tracking timestamping

CONFIG_NF_CONNTRACK_TIMESTAMP [=y] **[Y]**
Diese Option aktiviert die Unterstützung für die Zeitstempelung der Verbindungsverfolgung. Dadurch können Sie die Startzeit des Datenflusses speichern und die Zeit für die Beendigung des Datenflusses (nach dessen Zerstörung) über Ereignisse der Verbindungsverfolgung abrufen. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.17 Connection tracking labels

CONFIG_NF_CONNTRACK_LABELS [=y] **[Y]**
Diese Option ermöglicht die Zuweisung von benutzerdefinierten Flaggenbits zu Einträgen der Verbindungsverfolgung. Sie kann mit xtables connlabel match und dem nftables ct Ausdruck verwendet werden.

14.1.15.2.18 DCCP protocol connection tracking support

CONFIG_NF_CT_PROTO_DCCP [=y] **[Y]**
Wenn diese Option aktiviert ist, kann der Layer-3-unabhängige Code für die Verbindungsverfolgung den Zustand von DCCP-Verbindungen verfolgen. Wenn Sie unsicher sind, sagen Sie Y.

14.1.15.2.19 SCTP protocol connection tracking support

CONFIG_NF_CT_PROTO_SCTP [=y] **[Y]**
Wenn diese Option aktiviert ist, kann der Layer-3-unabhängige Verbindungsverfolgungscode den Status von SCTP-Verbindungen verfolgen. Wenn Sie unsicher sind, sagen Sie Y.

14.1.15.2.20 UDP-Lite protocol connection tracking support

CONFIG_NF_CT_PROTO_UPDLITE [=y] **[Y]**
Wenn diese Option aktiviert ist, kann der Layer-3-unabhängige Verbindungsverfolgungscode eine Zustandsverfolgung bei UDP-Lite-Verbindungen durchführen. Wenn Sie unsicher sind, sagen Sie Y.

14.1.15.2.21 Amanda backup protocol support

CONFIG_NF_CONNTRACK_AMANDA [=m] **[M]**
Wenn Sie das Amanda-Backup-Paket `http://www.amanda.org/` auf diesem Rechner oder auf Rechnern, die über diesen Rechner MASQUERADED werden, ausführen, sollten Sie diese Funktion aktivieren. Dies ermöglicht es dem Verbindungsverfolgungs- und Natting-Code, die Unterkanäle zuzulassen, die Amanda für die Kommunikation der Sicherungsdaten, Nachrichten und des Index benötigt. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.22 FTP protocol support

CONFIG_NF_CONNTRACK_FTP [=m] **[M]**
Die Verfolgung von FTP-Verbindungen ist problematisch: Für die Verfolgung dieser Verbindungen und die Durchführung von Masquerading und anderen Formen der Network Address Translation sind spezielle Hilfsmittel erforderlich. Dies ist FTP-Unterstützung auf Layer 3 mit unabhängiger Verbindungsverfolgung. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.23 H.323 protocol support

CONFIG_NF_CONNTRACK_H323 [=m] [**M**]
H.323 ist ein VoIP-Signalisierungsprotokoll der ITU-T. Als eines der wichtigsten VoIP-Protokolle wird es weithin von Sprach-Hardware und -Software verwendet, darunter Sprach-Gateways, IP-Telefone, Netmeeting, OpenPhone, Gnomemeeting usw. Mit diesem Modul können Sie H.323 auf einer Verbindungsverfolgung/NAT-Firewall unterstützen. Dieses Modul unterstützt RAS, Fast Start, H.245 Tunnelling, Call Forwarding, RTP/RTCP und T.120 basierte Audio-, Video-, Fax-, Chat-, Whiteboard-, Dateiübertragung, etc. Für weitere Informationen besuchen Sie bitte `http://nath323.sourceforge.net/`.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.24 IRC protocol support

CONFIG_NF_CONNTRACK_IRC [=m] [**M**]
Es gibt eine weit verbreitete Erweiterung des IRC, das Direct Client-to-Client Protocol (DCC). Damit können Benutzer Dateien aneinander senden und auch miteinander chatten, ohne dass ein Server erforderlich ist. DCC Sending wird überall dort verwendet, wo Sie Dateien über IRC senden, und DCC Chat wird am häufigsten von Eggdrop-Bots verwendet. Wenn Sie NAT verwenden, ermöglicht Ihnen diese Erweiterung, Dateien zu senden und Chats zu initiieren. Beachten Sie, dass Sie diese Erweiterung NICHT benötigen, um Dateien abzurufen oder Chats zu initiieren, oder alles andere im IRC.
Um sie als Modul zu kompilieren, wähle hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.25 NetBIOS name service protocol support

CONFIG_NF_CONNTRACK_NETBIOS_NS [=m] [**M**]
NetBIOS-Namensdienstanfragen werden als Broadcast-Nachrichten von einem unprivilegierten Port gesendet und mit Unicast-Nachrichten an denselben Port beantwortet. Das macht es schwierig, sie mit einer Firewall zu schützen, da die Verbindungsverfolgung nicht mit Broadcasts umgehen kann. Dieses Hilfsprogramm verfolgt die lokalen NetBIOS-Namensdienstanfragen und die entsprechenden Antworten. Er ist auf eine korrekte IP-Adresskonfiguration angewiesen, insbesondere auf die Netzmaske und die Broadcast-Adresse. Wenn sie richtig konfiguriert sind, sollte die Ausgabe von `ip address show` etwa so aussehen:

```
$ ip -4 address show eth0
4: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
         inet 172.16.2.252/24 brd 172.16.2.255 Bereich global eth0
```

Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.26 SNMP service protocol support

CONFIG_NF_CONNTRACK_SNMP [=m] [**M**]
SNMP-Dienstanforderungen werden als Broadcast-Nachrichten von einem unprivilegierten Port gesendet und mit Unicast-Nachrichten an denselben Port beantwortet. Das macht es schwierig, sie mit einer Firewall zu schützen, da die Verbindungsverfolgung nicht mit Broadcasts umgehen kann. Dieses Hilfsprogramm verfolgt die lokalen SNMP-Dienstanfragen und die entsprechenden Antworten. Es verlässt sich auf die korrekte Konfiguration der IP-Adresse, insbesondere der Netzmaske und der Broadcast-Adresse.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.27 PPtP protocol support

CONFIG_NF_CONNTRACK_PPTP [=m] [**M**]
Dieses Modul fügt Unterstützung für PPTP (Point to Point Tunnelling Protocol, RFC 2637) Verbindungsverfolgung und NAT hinzu. Wenn Sie PPTP-Sitzungen über eine Stateful-Firewall oder NAT-Box laufen lassen, sollten Sie diese Funktion aktivieren. Bitte beachten Sie, dass noch nicht alle PPTP-Betriebsmodi unterstützt werden. Insbesondere bestehen diese Einschränkungen:

- Es wird blind davon ausgegangen, dass Kontrollverbindungen immer in Richtung PNS→PAC aufgebaut werden. Dies ist eine Verletzung von RFC 2637.

- Unterstützt nur einen einzigen Aufruf innerhalb jeder Sitzung

Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.28 SANE protocol support

CONFIG_NF_CONNTRACK_SANE [=m] **[M]**
SANE ist ein Protokoll für den Fernzugriff auf Scanner, das durch den Daemon `saned` implementiert wird. Wie FTP verwendet es getrennte Kontroll- und Datenverbindungen. Mit diesem Modul können Sie SANE auf einer Firewall mit Verbindungsverfolgung unterstützen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.29 SIP protocol support

CONFIG_NF_CONNTRACK_SANE [=m] **[M]**
SIP ist ein Kontrollprotokoll der Anwendungsschicht, mit dem Multimedia-Sitzungen (Konferenzen) wie Internet-Telefonate aufgebaut, geändert und beendet werden können.
Mit den Modulen `nf_conntrack_sip` und `nf_nat_sip` können Sie das Protokoll auf einer Verbindungsverfolgung/NATing-Firewall unterstützen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.30 TFTP protocol support

CONFIG_NF_CONNTRACK_TFTP [=m] **[M]**
TFTP-Verbindungsnachverfolgungshilfe; dies ist erforderlich, je nachdem, wie restriktiv Ihr Regelwerk ist. Wenn Sie einen tftp-Client hinter -j SNAT oder -j MASQUERADING verwenden, benötigen Sie dies. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.31 Connection tracking netlink interface

CONFIG_NF_CT_NETLINK [=m] **[M]**
Diese Option aktiviert die Unterstützung für eine netlink-basierte Benutzerschnittstelle.

14.1.15.2.32 Connection tracking timeout tuning via Netlink

CONFIG_NF_CT_NETLINK_TIMEOUT [=m] **[M]**
Mit dieser Option wird die Unterstützung für die Feinabstimmung des Zeitlimits für die Verbindungsverfolgung aktiviert. Dies ermöglicht es Ihnen, spezifische Zeitüberschreitungsrichtlinien an Abläufe anzuhängen, anstatt die globale Zeitüberschreitungsrichtlinie zu verwenden. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.33 Connection tracking helpers in user-space via Netlink

CONFIG_NF_CT_NETLINK_HELPER [=m] **[M]**
Mit dieser Option wird die Infrastruktur für die Verbindungsnachverfolgung im Benutzerbereich aktiviert. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.34 NFQUEUE and NFLOG integration with Connection Tracking

CONFIG_NETFILTER_NETLINK_GLUE_CT [=y] **[Y]**
Wenn diese Option aktiviert ist, können NFQUEUE und NFLOG zusammen mit dem Paket, das über NFNETLINK in die Warteschlange gestellt wurde, Informationen zur Verbindungsverfolgung enthalten.

14.1.15.2.35 Network Address Translation support

CONFIG_NF_NAT [=m] **[M]**
Die NAT-Option ermöglicht Masquerading, Portweiterleitung und andere Formen der vollständigen Network Address Port Translation. Dies kann durch iptables, ip6tables oder nft kontrolliert werden.

14.1.15.2.36 Netfilter nf_tables support

CONFIG_NF_TABLES [=m] **[M]**
nftables ist das neue Rahmenwerk zur Paketklassifizierung, das die bestehende {ip,ip6,arp,eb}_tables-Infrastruktur ersetzen soll. Es bietet eine Pseudo-Zustandsmaschine mit einem erweiterbaren Befehlssatz (auch als Ausdrücke bekannt), den das Userspace-Dienstprogramm `nft` (`https://www.netfilter.org/projects/nftables`) zum Aufbau des Regelsatzes verwendet. Außerdem enthält es die generische Set-Infrastruktur, die es Ihnen ermöglicht, Zuordnungen zwischen Übereinstimmungen und Aktionen zu konstruieren, um die Leistung zu verbessern. Um es als Modul zu kompilieren, wählen Sie hier M.

14.1.15.2.36.1 Netfilter nf_tables mixed IPv4/IPv6 tables support

CONFIG_NF_TABLES_INET [=y] **[Y]**
Diese Option aktiviert die Unterstützung für eine gemischte IPv4/IPv6-„inet“-Tabelle.

14.1.15.2.36.2 Netfilter nf_tables netdev tables support

CONFIG_NF_TABLES_NETDEV [=y] **[Y]**
Diese Option aktiviert die Unterstützung für die Tabelle „netdev“.

14.1.15.2.36.3 Netfilter nf_tables number generator module

CONFIG_NFT_NUMGEN [=m] **[M]**
Diese Option fügt den Ausdruck für den Zahlengenerator hinzu, der zur Durchführung der inkrementellen Zählung und der an eine Obergrenze gebundenen Zufallszahlen verwendet wird.

14.1.15.2.36.4 Netfilter nf_tables conntrack module

CONFIG_NFT_CT [=m] **[M]**
Diese Option fügt den Ausdruck "ct"hinzu, den Sie verwenden können, um Informationen zur Verbindungsverfolgung, wie z. B. den Status des Datenflusses, abzugleichen.

14.1.15.2.36.5 Netfilter nf_tables hardware flow offload module

CONFIG_NFT_FLOW_OFFLOAD [=m] **[M]**
Diese Option fügt den Ausdruck „flow_offload“ hinzu, mit dem Sie festlegen können, welche Datenströme in die Hardware eingespeist werden.

14.1.15.2.36.6 Netfilter nf_tables connlimit module

CONFIG_NFT_CONNLIMIT [=m] **[M]**
Diese Option fügt den Ausdruck „connlimit“ hinzu, den Sie verwenden können, um die Übereinstimmung von Regeln pro Verbindung zu begrenzen.

14.1.15.2.36.7 Netfilter nf_tables log module

CONFIG_NFT_LOG [=m] **[M]**
Diese Option fügt den Ausdruck „log“ hinzu, den Sie verwenden können, um Pakete zu protokollieren, die bestimmten Kriterien entsprechen.

14.1.15.2.36.8 Netfilter nf_tables limit module

CONFIG_NFT_LOG [=m] **[M]**
Diese Option fügt den Ausdruck „limit“ hinzu, den Sie verwenden können, um die Übereinstimmung von Regeln zu begrenzen.

14.1.15.2.36.9 Netfilter nf_tables masquerade support

CONFIG_NFT_MASQ [=m] **[M]**
Diese Option fügt den „masquerade“-Ausdruck hinzu, den Sie verwenden können, um NAT im Masquerade-Flavour durchzuführen.

14.1.15.2.36.10 Netfilter nf_tables redirect support

CONFIG_NFT_REDIR [=m] **[M]**
Diese Option fügt den Ausdruck „redirect“ hinzu, mit dem Sie NAT im Redirect-Flavour durchführen können.

14.1.15.2.36.11 Netfilter nf_tables nat module

CONFIG_NFT_NAT [=m] **[M]**
Diese Option fügt den Ausdruck „nat“ hinzu, mit dem Sie typische NAT-Paketumwandlungen (Network Address Translation) durchführen können.

14.1.15.2.36.12 Netfilter nf_tables tunnel module

CONFIG_NFT_TUNNEL [=m] **[M]**
Diese Option fügt den Ausdruck „tunnel“ hinzu, den Sie zum Festlegen von Tunneling-Richtlinien verwenden können.

14.1.15.2.36.13 Netfilter nf_tables queue module

CONFIG_NFT_QUEUE [=m] [**M**]
Dies ist erforderlich, wenn Sie die Userspace-Warteschlangen-Infrastruktur (auch bekannt als NFQUEUE) von nftables verwenden wollen.

14.1.15.2.36.14 Netfilter nf_tables quota module

CONFIG_NFT_QUEUE [=m] [**M**]
Diese Option fügt den Ausdruck „quota“ hinzu, den Sie verwenden können, um Byte-Quoten zu erzwingen.

14.1.15.2.36.15 Netfilter nf_tables reject support

CONFIG_NFT_REJECT [=m] [**M**]
Diese Option fügt den Ausdruck „reject“ hinzu, den Sie verwenden können, um nicht zugelassenen Datenverkehr explizit abzulehnen und über TCP-Reset/ICMP-Informationsfehler zu benachrichtigen.

14.1.15.2.36.16 Netfilter x_tables over nf_tables module

CONFIG_NFT_COMPAT [=m] [**M**]
Dies ist erforderlich, wenn Sie beabsichtigen, eine der vorhandenen x_tables match/target-Erweiterungen über das nf_tables-Framework zu verwenden.

14.1.15.2.36.17 Netfilter nf_tables hash module

CONFIG_NFT_HASH [=m] [**M**]
Diese Option fügt den Ausdruck glqq hash“ hinzu, mit dem Sie eine Hash-Operation für Register durchführen können.

14.1.15.2.36.18 Netfilter nf_tables fib inet support

CONFIG_NFT_FIB_INET [=m] [**M**]
Diese Option ermöglicht die Verwendung des FIB-Ausdrucks aus der Inet-Tabelle. Die Suche wird an die IPv4- oder IPv6-FIB delegiert, je nach dem Protokoll des Pakets.

14.1.15.2.36.19 Netfilter nf_tables xfrm/IPSec security association matching

CONFIG_NFT_XFRM [=m] [**M**]
Diese Option fügt einen Ausdruck hinzu, den Sie verwenden können, um Eigenschaften einer Paketsicherheitszuordnung zu extrahieren.

14.1.15.2.36.20 Netfilter nf_tables socket match support

CONFIG_NFT_SOCKET [=m] [**M**]
Diese Option ermöglicht den Abgleich auf das Vorhandensein oder Nichtvorhandensein eines entsprechenden Sockets und seiner Attribute.

14.1.15.2.36.21 Netfilter nf_tables passive OS fingerprint support

CONFIG_NFT_OSF [=m] [**M**]
Mit dieser Option können Pakete von einem bestimmten Betriebssystem abgeglichen werden.

14.1.15.2.36.22 Netfilter nf_tables tproxy support

CONFIG_NFT_TPROXY [=m] [**M**]
Dadurch wird die Unterstützung für transparente Proxys in nftables verfügbar.

14.1.15.2.36.23 Netfilter nf_tables SYNPROXY expression support

CONFIG_NFT_SYNPROXY [=m] [**M**]
Mit dem SYNPROXY-Ausdruck können Sie TCP-Verbindungen abfangen und mit Syncookies aufbauen, bevor sie an den Server weitergeleitet werden. Auf diese Weise können Sie die Nutzung von Verbindungen und Serverressourcen bei SYN-Flood-Angriffen vermeiden.

14.1.15.2.36.24 Netfilter packet duplication support

CONFIG_NF_DUP_NETDEV [=m] [**M**]
Diese Option aktiviert die generische Infrastruktur zur Paketvervielfältigung für Netfilter.

14.1.15.2.36.25 Netfilter nf_tables netdev packet duplication support

CONFIG_NFT_DUP_NETDEV [=m] **[M]**
Mit dieser Option wird die Paketverdopplung für die „netdev“-Familie aktiviert.

14.1.15.2.36.26 Netfilter nf_tables netdev packet forwarding support

CONFIG_NFT_FWD_NETDEV [=m] **[M]**
Diese Option aktiviert die Paketweiterleitung für die Familie „netdev“.

14.1.15.2.36.27 Netfilter nf_tables netdev fib lookups support

CONFIG_NFT_FIB_NETDEV [=m] **[M]**
Diese Option ermöglicht die Verwendung des FIB-Ausdrucks aus der netdev-Tabelle. Die Suche wird an die IPv4- oder IPv6-FIB delegiert, je nach dem Protokoll des Pakets.

14.1.15.2.36.28 Netfilter nf_tables netdev fib REJECT support

CONFIG_NFT_REJECT_NETDEV [=m] **[M]**
Diese Option aktiviert die REJECT-Unterstützung in der netdev-Tabelle. Die Erzeugung von Rücksendepaketen wird an die IPv4- oder IPv6-ICMP- oder TCP-RST-Implementierung delegiert, je nach dem Protokoll des Pakets.

14.1.15.2.37 Netfilter flow table mixed IPv4/IPv6 module

CONFIG_NF_FLOW_TABLE_INET [=m] **[M]**
Diese Option fügt die gemischte IPv4/IPv6-Unterstützung der Flow Table hinzu. Um sie als Modul zu kompilieren, wählen Sie hier M.

14.1.15.2.38 Netfilter flow table module

CONFIG_NF_FLOW_TABLE [=m] **[M]**
Diese Option fügt die Kerninfrastruktur der Ablauftabelle hinzu. Um sie als Modul zu kompilieren, wählen Sie hier M.

14.1.15.2.38.1 Supply flow table statistics in procfs

CONFIG_NF_FLOW_TABLE_PROCFS [=y] **[Y]**
Diese Option ermöglicht die Anzeige der Flow-Table-Offload-Statistiken in procfs unter net/netfilter/nf_flowtable.

14.1.15.2.39 Netfilter Xtables support (required for ip_tables)

CONFIG_NETFILTER_XTABLES [=m] **[M]**
Dies ist erforderlich, wenn Sie eine der Tabellen ip_tables, ip6_tables oder arp_tables verwenden wollen.

14.1.15.2.39.1 Netfilter Xtables 32bit support

CONFIG_NETFILTER_XTABLES_COMPAT [=y] **[Y]**
Diese Option bietet eine Übersetzungsschicht, um 32bit arp,ip(6),ebtables-Binärdateien auf 64bit-Kerneln laufen zu lassen. Wenn Sie unsicher sind, sagen Sie N.

***** Xtables combined modules *****
(Xtables kombinierte Module)

14.1.15.2.39.2 nfmark target and match support

CONFIG_NETFILTER_XT_MARK [=m] **[M]**
Diese Option fügt das „MARK“-Ziel und die „mark“-Übereinstimmung hinzu. Mit dem Netfilter-Mark-Matching können Sie Pakete auf der Grundlage des „nfmark“-Werts im Paket abgleichen. Mit dem Ziel können Sie in der „mangle“-Tabelle Regeln erstellen, die das mit dem Paket verbundene Feld „netfilter mark“ (nfmark) ändern. Vor dem Routing kann die nfmark die Routing-Methode beeinflussen und kann auch von anderen Subsystemen verwendet werden, um ihr Verhalten zu ändern.

14.1.15.2.39.3 ctmark target and match support

CONFIG_NETFILTER_XT_CONNMARK [=m] [**M**]
Diese Option fügt das Ziel „CONNMARK“ und die Übereinstimmung „connmark“ hinzu. Netfilter ermöglicht es Ihnen, einen Markierungswert pro Verbindung (auch bekannt als ctmark) zu speichern, ähnlich wie bei der Paketmarkierung (nfmark). Mit Hilfe dieses Ziels und der Übereinstimmung können Sie diese Markierung setzen und abgleichen.

14.1.15.2.39.4 set target and match support

CONFIG_NETFILTER_XT_SET [=m] [**M**]
Diese Option fügt das „SET“-Ziel und die „set“-Übereinstimmung hinzu. Mit diesem Ziel und dieser Übereinstimmung können Sie Elemente in den von ipset(8) erstellten Sets hinzufügen/löschen und abgleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

***** Xtables combined modules *****
(Xtables kombinierte Module)

14.1.15.2.39.5 AUDIT target support

CONFIG_NETFILTER_XT_TARGET_AUDIT [=m] [**M**]
Diese Option fügt ein 'AUDIT'-Ziel hinzu, das verwendet werden kann, um Audit-Aufzeichnungen für verworfene/akzeptierte Pakete zu erstellen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.6 CHECKSUM target support

CONFIG_NETFILTER_XT_TARGET_CHECKSUM [=m] [**M**]
Diese Option fuegt ein ‚CHECKSUM‘-Ziel hinzu, das in der iptables Mangle-Tabelle verwendet werden kann, um fehlerhafte DHCP-Clients in virtualisierten Umgebungen zu umgehen. Einige alte DHCP-Clients lassen Pakete fallen, weil sie nicht wissen, dass die Prüfsumme normalerweise auf die Hardware ausgelagert wird und daher als gültig angesehen werden sollte. Dieses Ziel kann verwendet werden, um die Prüfsumme mit iptables auszufüllen, wenn solche Pakete über ein virtuelles Netzwerkgerät gesendet werden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.7 “CLASSIFY” target support

CONFIG_NETFILTER_XT_TARGET_CLASSIFY [=m] [**M**]
Diese Option fügt ein ‚CLASSIFY‘-Ziel hinzu, das es dem Benutzer ermöglicht, die Priorität eines Pakets festzulegen. Einige qdiscs können diesen Wert zur Klassifizierung verwenden, darunter sind:

atm, cbq, dsmark, pfifo_fast, htb, prio

Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.8 “CONNMARK” target support

CONFIG_NETFILTER_XT_TARGET_CONNMARK [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig).
Mit ihr wird CONFIG_NETFILTER_XT_CONNMARK (kombiniertes connmark/CONNMARK-Modul) ausgewählt.

14.1.15.2.39.9 “CONNSECMARK” target support

CONFIG_NETFILTER_XT_TARGET_CONNSECMARK [=m] [**M**]
Die Zielvorgabe CONNSECMARK kopiert Sicherheitsmarkierungen von Paketen auf Verbindungen und stellt Sicherheitsmarkierungen von Verbindungen auf Pakete wieder her (wenn die Pakete nicht bereits markiert sind). Er wird normalerweise in Verbindung mit dem SECMARK-Ziel verwendet.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.10 “CT” target support

CONFIG_NETFILTER_XT_TARGET_CT [=m] [**M**]
Diese Option fügt ein ‚CT‘-Ziel hinzu, das es ermöglicht, anfängliche Parameter für die Verbindungsverfolgung wie zu übermittelnde Ereignisse und den zu verwendenden Helfer anzugeben.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.11 “DSCP” and “TOS” target support

CONFIG_NETFILTER_XT_TARGET_DSCP [=m] **[M]**

Diese Option fügt ein ‚DSCP‘-Ziel hinzu, mit dem Sie das DSCP-Feld (Differentiated Services Codepoint) des IPv4/IPv6-Headers manipulieren können. Das DSCP-Feld kann einen beliebigen Wert zwischen 0x0 und einschließlich 0x3f haben. Es fügt auch das „TOS“-Ziel hinzu, mit dem Sie Regeln in der „Mangle“-Tabelle erstellen können, die das „Type Of Service“-Feld eines IPv4- oder das Prioritätsfeld eines IPv6-Pakets vor dem Routing ändern.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, wählen Sie N.

14.1.15.2.39.12 “HL” hoplimit target support

CONFIG_NETFILTER_XT_TARGET_HL [=m] **[M]**

Diese Option fügt die Ziele „HL“ (für IPv6) und „TTL“ (für IPv4) hinzu, die es dem Benutzer ermöglichen, den Hoplimit-/Time-to-live-Wert des IP-Headers zu ändern. Während es sicher ist, den Hoplimit/TTL-Wert zu dekrementieren, erlauben die Module auch, den Hoplimit-Wert des Headers zu erhöhen und auf beliebige Werte zu setzen. Dies ist EXTREM GEFÄHRLICH, da man leicht unsterbliche Pakete erzeugen kann, die sich ewig im Netz drehen.

14.1.15.2.39.13 “HMARK” target support

CONFIG_NETFILTER_XT_TARGET_HMARK [=m] **[M]**

Diese Option fügt das Ziel „HMARK“ hinzu. Mit diesem Ziel können Sie in den Tabellen „raw“ und „mangle“ Regeln erstellen, die die skbuff-Marke mittels Hash-Berechnung innerhalb eines bestimmten Bereichs setzen. Die nfmark kann die Routing-Methode beeinflussen und kann auch von anderen Teilsystemen verwendet werden, um deren Verhalten zu ändern. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.14 IDLETIMER target support

CONFIG_NETFILTER_XT_TARGET_IDLETIMER [=m] **[M]**

Diese Option fügt das Ziel „IDLETIMER“ hinzu. Jedes übereinstimmende Paket setzt den Timer zurück, der mit dem Label verbunden ist, das beim Hinzufügen der Regel angegeben wurde. Wenn der Timer abläuft, löst er eine sysfs-Benachrichtigung aus. Die verbleibende Zeit bis zum Ablauf kann über sysfs ausgelesen werden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.15 “LED” target support

CONFIG_NETFILTER_XT_TARGET_LED [=m] **[M]**

Diese Option fügt ein ‘LED’-Ziel hinzu, mit dem Sie LEDs als Reaktion auf bestimmte Pakete, die Ihren Rechner passieren, blinken lassen können. Dies kann dazu verwendet werden, eine freie LED in eine Netzwerkaktivitäts-LED zu verwandeln, die z. B. nur bei FTP-Übertragungen blinkt. Oder Sie könnten eine LED haben, die jedes Mal für ein oder zwei Minuten aufleuchtet, wenn sich jemand über SSH mit Ihrem Rechner verbindet. Damit dies funktioniert, benötigen Sie Unterstützung für die Klasse „led“. So erstellen Sie einen LED-Auslöser für eingehenden SSH-Verkehr:

```
iptables -A INPUT -p tcp --dport 22 -j LED --led-trigger-id ssh --led-delay 1000
```

Verbinden Sie dann den neuen Auslöser mit einer LED auf Ihrem System:

```
echo netfilter-ssh > /sys/class/leds/<ledname>/trigger
```

Weitere Informationen zu den auf Ihrem System verfügbaren LEDs finden Sie unter Documentation/leds/leds-class.rst

14.1.15.2.39.16 LOG target support

CONFIG_NETFILTER_XT_TARGET_LOG [=m] **[M]**

Diese Option fügt ein ‚LOG‘-Ziel hinzu, das es Ihnen erlaubt, Regeln in jeder iptables-Tabelle zu erstellen, die den Paket-Header im Syslog aufzeichnen. Um es als Modul zu kompilieren, wähle hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.17 “MARK” target support

CONFIG_NETFILTER_XT_TARGET_MARK [=m] **[M]**

Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von

oldconfig). Mit ihr wird CONFIG_NETFILTER_XT_MARK (kombiniertes Mark/MARK-Modul) ausgewählt.

14.1.15.2.39.18 "SNAT and DNAT" targets support

CONFIG_NETFILTER_XT_NAT [=m] [**M**]
Mit dieser Option werden die Ziele SNAT und DNAT aktiviert. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.19 "NETMAP" target support

CONFIG_NETFILTER_XT_TARGET_NETMAP [=m] [**M**]
NETMAP ist eine Implementierung der statischen 1:1-NAT-Zuordnung von Netzwerkadressen. Sie bildet den Teil der Netzwerkadresse ab, während der Teil der Hostadresse intakt bleibt.

14.1.15.2.39.20 "NFLOG" target support

CONFIG_NETFILTER_XT_TARGET_NFLOG [=m] [**M**]
Diese Option aktiviert das NFLOG-Ziel, das es ermöglicht, Nachrichten über nfnetlink_log zu protokollieren. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.21 "NFQUEUE" target support

CONFIG_NETFILTER_XT_TARGET_NFQUEUE [=m] [**M**]
Dieses Ziel hat das alte, veraltete QUEUE-Ziel ersetzt. Im Gegensatz zu QUEUE unterstützt es 65535 verschiedene Warteschlangen, nicht nur eine. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.22 "NOTRACK" target support (DEPRECATED)

CONFIG_NETFILTER_XT_TARGET_NFQUEUE [=m] [**M**]
Für diese Option gibt es keine Hilfe.

14.1.15.2.39.23 "RATEEST" target support

CONFIG_NETFILTER_XT_TARGET_RATEEST [=m] [**M**]
Diese Option fügt ein ‚RATEEST'-Ziel hinzu, das es ermöglicht, Raten ähnlich wie bei TC-Schätzern zu messen. Die ‚Rateest'-Übereinstimmung kann zum Abgleich mit den gemessenen Raten verwendet werden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.24 REDIRECT target support

CONFIG_NETFILTER_XT_TARGET_REDIRECT [=m] [**M**]
REDIRECT ist ein Spezialfall von NAT: Alle eingehenden Verbindungen werden auf die Adresse der eingehenden Schnittstelle abgebildet, so dass die Pakete zum lokalen Rechner gelangen, anstatt durchzugehen. Dies ist nützlich für transparente Proxies. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.25 MASQUERADE target support

CONFIG_NETFILTER_XT_TARGET_MASQUERADE [=m] [**M**]
Masquerading ist ein Spezialfall von NAT: Alle ausgehenden Verbindungen werden so verändert, dass sie von einer bestimmten Schnittstellenadresse zu kommen scheinen, und wenn die Schnittstelle ausfällt, gehen diese Verbindungen verloren. Dies ist nur für Einwahlkonten mit dynamischer IP-Adresse nützlich (d. h. Ihre IP-Adresse wird bei der nächsten Einwahl eine andere sein). Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.26 "TEE" – packet cloning to alternate destination

CONFIG_NETFILTER_XT_TARGET_TEE [=m] [**M**]
Diese Option fügt ein „TEE"-Ziel hinzu, mit dem ein Paket geklont werden kann und dieser Klon zu einem anderen Nexthop umgeleitet wird.

14.1.15.2.39.27 “TPROXY” target transparent proxying support

CONFIG_NETFILTER_XT_TARGET_TPROXY [=m] **[M]**
Diese Option fügt ein ‚TPROXY‘-Ziel hinzu, das dem REDIRECT-Ziel ähnlich ist. Es kann nur in der Mangle-Tabelle verwendet werden und ist nützlich, um den Verkehr an einen transparenten Proxy umzuleiten. Im Gegensatz zu REDIRECT ist sie **nicht** von der Netfilter-Verbindungsverfolgung und NAT abhängig. Damit es funktioniert, müssen Sie bestimmte iptables-Regeln konfigurieren und Policy-Routing verwenden. Weitere Informationen zur Einrichtung finden Sie unter
Documentation/networking/tproxy.rst.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.28 “TRACE” target support

CONFIG_NETFILTER_XT_TARGET_TRACE [=m] **[M]**
Mit dem TRACE-Ziel können Sie Pakete markieren, so dass der Kernel jede Regel protokolliert, die mit den Paketen übereinstimmt, während diese die Tabellen, Ketten und Regeln durchlaufen. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.29 “SECMARK” target support

CONFIG_NETFILTER_XT_TARGET_SECMARK [=m] **[M]**
Die SECMARK-Zielvorgabe ermöglicht die Sicherheitsmarkierung von Netzpaketen zur Verwendung mit Sicherheits-Subsystemen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.30 “TCPMSS” target support

CONFIG_NETFILTER_XT_TARGET_TCPMSS [=m] **[M]**
Diese Option fügt ein ‚TCPMSS‘-Ziel hinzu, das es Ihnen erlaubt, den MSS-Wert von TCP SYN-Paketen zu ändern, um die maximale Größe für diese Verbindung zu kontrollieren (normalerweise wird sie auf die MTU Ihrer ausgehenden Schnittstelle minus 40 begrenzt). Dies wird verwendet, um kriminell hirnlose ISPs oder Server zu überwinden, die ICMP Fragmentation Needed Pakete blockieren. Dieses Problem äußert sich darin, dass von der Linux-Firewall/vom Router aus alles gut funktioniert, aber die Rechner dahinter nie große Pakete austauschen können:

- Webbrowser stellen eine Verbindung her und bleiben dann hängen, ohne dass Daten empfangen werden.
- Kleine E-Mails funktionieren gut, aber große E-Mails bleiben hängen.
- ssh funktioniert gut, aber scp hängt sich nach dem ersten Handshaking auf.

Abhilfe: Aktivieren Sie diese Option und fügen Sie eine Regel in Ihre Firewall-Konfiguration ein, z. B:

```
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN \
             -j TCPMSS --clamp-mss-to-pmtu
```

Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.31 “TCPOPTSTRIP” target support

CONFIG_NETFILTER_XT_TARGET_TCPOPTSTRIP [=m] **[M]**
Diese Option fügt ein „TCPOPTSTRIP“-Ziel hinzu, mit dem Sie TCP-Optionen aus TCP-Paketen entfernen können.

***** Xtables matches *****
(Xtables Übereinstimmungen)

14.1.15.2.39.32 “addrtype” address type match support

CONFIG_NETFILTER_XT_MATCH_ADDRTYPE [=m] **[M]**
Mit dieser Option können Sie festlegen, was das Routing von einer Adresse hält, z. B. UNICAST, LOCAL, BROADCAST, ... Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.33 “bpf” match support

CONFIG_NETFILTER_XT_MATCH_BPF [=m] [**M**]
Der BPF-Abgleich wendet einen Linux-Socket-Filter auf jedes Paket an und akzeptiert diejenigen, für die der Filter einen Wert ungleich Null liefert. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.34 “control group” match support

CONFIG_NETFILTER_XT_MATCH_CGROUP [=m] [**M**]
Mit dem Socket/Prozess-Kontrollgruppenabgleich können Sie lokal erzeugte Pakete anhand der Zugehörigkeit von Prozessen zur net_cls-Kontrollgruppe abgleichen.

14.1.15.2.39.35 “cluster” match support

CONFIG_NETFILTER_XT_MATCH_CLUSTER [=m] [**M**]
Mit dieser Option können Sie arbeitsteilige Cluster von Netzwerkservern/zustandsfähigen Firewalls aufbauen, ohne einen dedizierten Router/Server/Switch für den Lastausgleich zu haben. Grundsätzlich gibt diese Übereinstimmung wahr zurück, wenn das Paket von diesem Clusterknoten verarbeitet werden muss. Somit sehen alle Knoten alle Pakete und diese Übereinstimmung entscheidet, welcher Knoten welche Pakete bearbeitet. Der Algorithmus für die Arbeitsteilung basiert auf dem Hashing der Quelladressen. Wenn Sie hier Y oder M sagen, versuchen Sie `iptables -m cluster --help` für weitere Informationen.

14.1.15.2.39.36 “comment” match support

CONFIG_NETFILTER_XT_MATCH_COMMENT [=m] [**M**]
Diese Option fügt einen „Kommentar“-Dummy-Match hinzu, der es Ihnen erlaubt, Kommentare in Ihren iptables-Regelsatz einzufügen. Wenn Du es als Modul kompilieren willst, sag hier M und lies <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.37 “connbytes” per-connection counter match support

CONFIG_NETFILTER_XT_MATCH_CONNBYTES [=m] [**M**]
Diese Option fügt eine „Connbytes“-Übereinstimmung hinzu, mit der Sie die Anzahl der Bytes und/oder Pakete für jede Richtung innerhalb einer Verbindung abgleichen können. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.38 “connlabel” match support

CONFIG_NETFILTER_XT_MATCH_CONNLABEL [=m] [**M**]
Mit dieser Zuordnung können Sie benutzerdefinierte Bezeichnungen testen und einer Verbindung zuweisen. Der Kernel speichert nur Bit-Werte – die Zuordnung von Namen zu Bits wird vom Userspace vorgenommen. Anders als bei connmark können einer Verbindung mehr als 32 Flaggenbits gleichzeitig zugewiesen werden.

14.1.15.2.39.39 “connlimit” match support

CONFIG_NETFILTER_XT_MATCH_CONNLIMIT [=m] [**M**]
Mit diesem Abgleich können Sie die Anzahl der parallelen Verbindungen zu einem Server pro Client-IP-Adresse (oder Adressblock) abgleichen.

14.1.15.2.39.40 “connmark” connection mark match support

CONFIG_NETFILTER_XT_MATCH_CONNMARK [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_CONNMARK (kombiniertes connmark/CONNMARK-Modul).

14.1.15.2.39.41 “conntrack” connection tracking match support

CONFIG_NETFILTER_XT_MATCH_CONNTRACK [=m] [**M**]
Dies ist ein allgemeines conntrack Abgleichsmodul, eine Obermenge des state match. Es ermöglicht den Abgleich zusätzlicher Conntrack-Informationen, was in komplexen Konfigurationen wie NAT-Gateways mit mehreren Internetverbindungen oder Tunneln nützlich ist. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.42 “cpu” match support

CONFIG_NETFILTER_XT_MATCH_CPU [=m] **[M]**
Mit dem CPU-Abgleich können Sie Pakete auf der Grundlage der CPU abgleichen, die das Paket gerade bearbeitet. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.43 “dccp” protocol match support

CONFIG_NETFILTER_XT_MATCH_DCCP [=m] **[M]**
Wenn diese Option aktiviert ist, können Sie die iptables-Übereinstimmung „dccp“ verwenden, um auf DCCP-Quell-/Zielports und DCCP-Flags zu reagieren. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>.
Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.44 “devgroup” match support

CONFIG_NETFILTER_XT_MATCH_DEVGROUP [=m] **[M]**
Diese Option fügt eine „Gerätegruppe“-Übereinstimmung hinzu, die eine Übereinstimmung mit der Gerätegruppe ermöglicht, der ein Netzwerkgerät zugeordnet ist. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.45 “dscp” and “tos” match support

CONFIG_NETFILTER_XT_MATCH_DSCP [=m] **[M]**
Diese Option fügt eine „DSCP“-Übereinstimmung hinzu, die eine Übereinstimmung mit dem DSCP-Feld des IPv4/IPv6-Headers (Differentiated Services Codepoint) ermöglicht. Das DSCP-Feld kann einen beliebigen Wert zwischen 0x0 und 0x3f einschließlich haben. Außerdem wird eine „tos“-Übereinstimmung hinzugefügt, die es Ihnen ermöglicht, Pakete auf der Grundlage der „Type Of Service“-Felder des IPv4-Pakets abzugleichen (die dieselben Bits wie DSCP haben). Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.46 “ecn” match support

CONFIG_NETFILTER_XT_MATCH_ECN [=m] **[M]**
Diese Option fügt eine „ECN“-Übereinstimmung hinzu, mit der Sie die ECN-Felder des IPv4- und TCP-Headers abgleichen können. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.47 “esp” match support

CONFIG_NETFILTER_XT_MATCH_ESP [=m] **[M]**
Mit dieser Abgleichserweiterung können Sie einen Bereich von SPIs im ESP-Header von IPSec-Paketen abgleichen. Um sie als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.48 “hashlimit” match support

CONFIG_NETFILTER_XT_MATCH_HASHLIMIT [=m] **[M]**
Diese Option fügt eine „hashlimit“-Übereinstimmung hinzu. Im Gegensatz zu „limit“ erstellt diese Übereinstimmung dynamisch eine Hash-Tabelle von Limit-Buckets, die auf Ihrer Auswahl von Quell-/Zieladressen und/oder Ports basiert. Sie ermöglicht es Ihnen, Richtlinien wie „10kpps für eine bestimmte Zieladresse“ oder „500pps von einer bestimmten Quelladresse“ mit einer einzigen Regel auszudrücken.

14.1.15.2.39.49 “helper” match support

CONFIG_NETFILTER_XT_MATCH_HASHLIMIT [=m] **[M]**
Helper Matching ermöglicht es Ihnen, Pakete in dynamischen Verbindungen, die von einem conntrack-Helper verfolgt werden, anzupassen, z. B. nf_conntrack_ftp
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie Y.

14.1.15.2.39.50 “hl” hoplimit/TTL match support

CONFIG_NETFILTER_XT_MATCH_HL [=m] **[M]**
Mit dem HL-Matching können Sie Pakete basierend auf dem Hoplimit im IPv6-Header oder dem Time-to-Live-Feld im IPv4-Header des Pakets abgleichen.

14.1.15.2.39.51 “ipcomp” match support

CONFIG_NETFILTER_XT_MATCH_IPCOMP [=m] [**M**]
Mit dieser Match-Erweiterung können Sie einen Bereich von CPIs (16 Bits) im IPComp-Header von IPSec-Paketen abgleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.52 “iprange” address range match support

CONFIG_NETFILTER_XT_MATCH_IPRANGE [=m] [**M**]
Diese Option fügt eine „iprange“-Übereinstimmung hinzu, die es Ihnen ermöglicht, eine Übereinstimmung auf der Grundlage eines IP-Adressbereichs zu erzielen. (Normalerweise passt iptables nur auf einzelne Adressen mit einer optionalen Maske.) Wenn Sie unsicher sind, sagen Sie M.

14.1.15.2.39.53 “ipvs” match support

CONFIG_NETFILTER_XT_MATCH_IPVS [=m] [**M**]
Mit dieser Option können Sie die IPVS-Eigenschaften eines Pakets abgleichen. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.54 “l2tp” match support

CONFIG_NETFILTER_XT_MATCH_L2TP [=m] [**M**]
Diese Option fügt eine „L2TP“-Übereinstimmung hinzu, die es Ihnen ermöglicht, die Header-Felder des L2TP-Protokolls abzugleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.55 “length” match support

CONFIG_NETFILTER_XT_MATCH_LENGTH [=m] [**M**]
Mit dieser Option können Sie die Länge eines Pakets mit einem bestimmten Wert oder einer Reihe von Werten vergleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.56 “limit” match support

CONFIG_NETFILTER_XT_MATCH_LIMIT [=m] [**M**]
Mit dem Limit-Matching können Sie die Rate kontrollieren, mit der eine Regel abgeglichen werden kann: Dies ist vor allem in Kombination mit dem LOG-Target („LOG-Target-Unterstützung“, unten) und zur Vermeidung einiger Denial-of-Service-Angriffe nützlich. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.57 “mac” address match support

CONFIG_NETFILTER_XT_MATCH_MAC [=m] [**M**]
Mit dem MAC-Abgleich können Sie Pakete auf der Grundlage der Ethernet-Quelladresse des Pakets abgleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.58 “mark” match support

CONFIG_NETFILTER_XT_MATCH_MARK [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_MARK (kombiniertes Mark/MARK-Modul).

14.1.15.2.39.59 “multiport” Multiple port match support

CONFIG_NETFILTER_XT_MATCH_MULTIPORT [=m] [**M**]
Mit dem Multiport-Matching können Sie TCP- oder UDP-Pakete auf der Grundlage einer Reihe von Quell- oder Zielports abgleichen: Normalerweise kann eine Regel nur einen einzigen Bereich von Ports abgleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.60 “nfacct” match support

CONFIG_NETFILTER_XT_MATCH_NFACCT [=m] [**M**]
Mit dieser Option können Sie die erweiterte Buchhaltung über nfnetlink_acct verwenden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.61 “osf” Passive OS fingerprint match

CONFIG_NETFILTER_XT_MATCH_OSF [=m] [**M**]
Mit dieser Option wird das Modul Passive OS Fingerprinting ausgewählt, das einen passiven Abgleich des entfernten Betriebssystems durch die Analyse eingehender TCP SYN-Pakete ermöglicht. Die Regeln und die Ladesoftware können von der Website `http://www.ioremap.net/projects/osf` heruntergeladen werden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.62 “owner” match support

CONFIG_NETFILTER_XT_MATCH_OWNER [=m] [**M**]
Mit dem Socket-Eigentümer-Abgleich können Sie lokal erzeugte Pakete danach abgleichen, wer den Socket erstellt hat: der Benutzer oder die Gruppe. Es ist auch möglich, zu prüfen, ob ein Socket tatsächlich existiert.

14.1.15.2.39.63 IPSEC “policy” match support

CONFIG_NETFILTER_XT_MATCH_POLICY [=m] [**M**]
Der Richtlinienabgleich ermöglicht es Ihnen, Pakete auf der Grundlage der IPsec-Richtlinie abzugleichen, die bei der Entkapselung verwendet wurde bzw. bei der Einkapselung verwendet werden wird. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.64 “physdev” match support

CONFIG_NETFILTER_XT_MATCH_PHYSDEV [=m] [**M**]
Der Physdev-Paketabgleich gleicht die physischen Bridge-Ports ab, an denen das IP-Paket angekommen ist oder die es verlassen wird. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.65 “pkttype” packet type match support

CONFIG_NETFILTER_XT_MATCH_PKTTYPE [=m] [**M**]
Der Pakettyp-Abgleich ermöglicht es Ihnen, ein Paket anhand seiner „Klasse“ abzugleichen, z. B. BROADCAST, MULTICAST, ...
Typische Verwendung:

```
iptables -A INPUT -m pkttype --pkt-type broadcast -j LOG
```

Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.66 “quota” match support

CONFIG_NETFILTER_XT_MATCH_QUOTA [=m] [**M**]
Diese Option fügt eine „quota“-Übereinstimmung hinzu, die eine Übereinstimmung mit einem Byte-Zähler ermöglicht. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.67 “rateest” match support

CONFIG_NETFILTER_XT_MATCH_RATEEST [=m] [**M**]
Diese Option fügt eine „Rateest“-Übereinstimmung hinzu, die eine Übereinstimmung mit der durch das RATEEST-Ziel geschätzten Rate ermöglicht. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.68 “realm” match support

CONFIG_NETFILTER_XT_MATCH_REALM [=m] [**M**]
Diese Option fügt eine „realm“-Übereinstimmung hinzu, die es Ihnen erlaubt, den Realm-Schlüssel aus dem Routing-Subsystem innerhalb von iptables zu verwenden. Diese Übereinstimmung ähnelt ziemlich genau der Option CONFIG_NET_CLS_ROUTE4 in tc world. Wenn Du es als Modul kompilieren willst, sage hier M und lies <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.69 “recent” match support

CONFIG_NETFILTER_XT_MATCH_RECENT [=m] [**M**]
Dieser Abgleich wird verwendet, um eine oder mehrere Listen mit kürzlich verwendeten Adressen zu erstellen und dann einen Abgleich mit dieser Liste bzw. diesen Listen durchzuführen. Kurze Optionen

sind verfügbar, indem man `iptables -m recent -h` verwendet. Offizielle Website: `http://snowman.net/projects/ipt_recent/`

14.1.15.2.39.70 "sctp" match support

CONFIG_NETFILTER_XT_MATCH_SCTP [=m] **[M]**
Wenn diese Option aktiviert ist, können Sie die „sctp"-Übereinstimmung verwenden, um auf SCTP-Quell-/Zielports und SCTP-Chunk-Typen abzustimmen. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.71 "socket" match support

CONFIG_NETFILTER_XT_MATCH_SOCKET [=m] **[M]**
Diese Option fügt eine „Socket"-Übereinstimmung hinzu, die verwendet werden kann, um Pakete zu finden, für die ein TCP- oder UDP-Socket-Lookup einen gültigen Socket findet. Sie kann in Kombination mit dem MARK-Ziel und dem Policy-Routing verwendet werden, um voll funktionsfähige, nicht ortsgebundene Sockets zu implementieren. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.72 "state" match support

CONFIG_NETFILTER_XT_MATCH_STATE [=m] **[M]**
Mit dem Verbindungsstatusabgleich können Sie Pakete auf der Grundlage ihrer Beziehung zu einer verfolgten Verbindung (d. h. früheren Paketen) abgleichen. Dies ist ein leistungsfähiges Werkzeug zur Klassifizierung von Paketen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.73 "statistic" match support

CONFIG_NETFILTER_XT_MATCH_STATISTIC [=m] **[M]**
Diese Option fügt einen „statistischen" Abgleich hinzu, der es ermöglicht, Pakete periodisch oder zufällig mit einem bestimmten Prozentsatz abzugleichen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.74 "string" match support

CONFIG_NETFILTER_XT_MATCH_STRING [=m] **[M]**
Diese Option fügt eine „String"-Übereinstimmung hinzu, die es Ihnen ermöglicht, nach Musterübereinstimmungen in Paketen zu suchen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.75 "tcpmss" match support

CONFIG_NETFILTER_XT_MATCH_TCPMSS [=m] **[M]**
Diese Option fügt eine „tcpmss"-Übereinstimmung hinzu, die es Ihnen ermöglicht, den MSS-Wert von TCP SYN-Paketen zu untersuchen, der die maximale Paketgröße für diese Verbindung kontrolliert. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.76 "time" match support

CONFIG_NETFILTER_XT_MATCH_TIME [=m] **[M]**
Diese Option fügt eine „Zeit"-Übereinstimmung hinzu, die es Ihnen ermöglicht, eine Übereinstimmung auf der Grundlage der Ankunftszeit des Pakets (auf dem Rechner, auf dem der Netfilter läuft) oder der Abfahrtszeit/des Abfahrtsdatums (für lokal erzeugte Pakete) zu erzielen. Wenn Sie hier Y für Ja sagen, versuchen Sie `iptables -m time --help` um weitere Informationen zu erhalten. Wenn Sie es als Modul kompilieren wollen, sagen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.2.39.77 "u32" match support

CONFIG_NETFILTER_XT_MATCH_U32 [=m] **[M]**
u32 ermöglicht es Ihnen, Mengen von bis zu 4 Bytes aus einem Paket zu extrahieren, sie mit bestimmten Masken mit UND zu verknüpfen, sie um bestimmte Beträge zu verschieben und zu prüfen, ob die

Ergebnisse in einem der angegebenen Bereiche liegen. Die Angabe, was extrahiert werden soll, ist allgemein genug, um Header mit im Paket gespeicherten Längen, wie z. B. IP- oder TCP-Header-Längen, zu überspringen. Details und Beispiele sind im Quelltext des Kernelmoduls zu finden.

14.1.15.3 IP set support →

CONFIG_IP_SET [=m] **[M]**
Diese Option erweitert den Kernel um die Unterstützung von IP-Sets. Um die Sets zu definieren und zu verwenden, benötigen Sie das Userspace-Dienstprogramm ipset(8). Sie können die Sets in netfilter über die „set“-Übereinstimmung und das „SET“-Ziel verwenden. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.1 Maximum number of IP sets

CONFIG_IP_SET_MAX [=256] **[256]**
Sie können hier den Standardwert für die maximale Anzahl von IP-Sets für den Kernel festlegen. Der Wert kann durch den Modulparameter „max_sets“ des Moduls „ip_set“ überschrieben werden.

Symbol: IP_SET_MAX [=256]
Type : Ganzzahl (integer)
Bereich: [2 65534]

14.1.15.3.2 bitmap:ip set support

CONFIG_IP_SET_BITMAP_IP [=m] **[M]**
Diese Option fügt die Unterstützung des Typs bitmap:ip set hinzu, mit dem man IPv4-Adressen (oder Netzwerkadressen) aus einem Bereich speichern kann.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.3 bitmap:ip,mac set support

CONFIG_IP_SET_BITMAP_IPMAC [=m] **[M]**
Diese Option fügt die Unterstützung des Typs bitmap:ip,mac set hinzu, mit dem man Paare von IPv4-Adressen und (Quell-)MAC-Adressen aus einem Bereich speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.4 bitmap:port set support

CONFIG_IP_SET_BITMAP_PORT [=m] **[M]**
Diese Option fügt die Unterstützung des Typs bitmap:port set hinzu, mit dem man TCP/UDP-Portnummern aus einem Bereich speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.5 hash:ip set support

CONFIG_IP_SET_HASH_IP [=m] **[M]**
Diese Option fügt die Unterstützung des Typs hash:ip set hinzu, mit dem man beliebige IPv4- oder IPv6-Adressen (oder Netzwerkadressen) in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.6 hash:ip,mark set support

CONFIG_IP_SET_HASH_IPMARK [=m] **[M]**
Diese Option fügt die Unterstützung des Typs hash:ip,mark set hinzu, mit dem man IPv4/IPv6-Adress- und Markenpaare speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.7 hash:ip,port set support

CONFIG_IP_SET_HASH_IPPORT [=m] **[M]**
Diese Option fügt die Unterstützung des Typs hash:ip,port set hinzu, mit dem man IPv4/IPv6-Adressen und Protokoll/Port-Paare speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.8 hash:ip,port,ip set support

CONFIG_IP_SET_HASH_IPPORTIP [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:ip,port,ip set hinzu, mit dem man IPv4/IPv6-Adress-, Protokoll/Port- und IPv4/IPv6-Adress-Tripel in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.9 hash:ip,port,net set support

CONFIG_IP_SET_HASH_IPPORTNET [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:ip,port,net hinzu, mit dem man IPv4/IPv6-Adress-, Protokoll/Port- und IPv4/IPv6-Netzwerkadressen/Präfix-Tripel in einem Set speichern kann.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.10 hash:ip,mac set support

CONFIG_IP_SET_HASH_IPMAC [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:ip,mac set hinzu, mit dem man Paare von IPv4/IPv6-Adressen und MAC (Ethernet-Adresse) in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.11 hash:mac set support

CONFIG_IP_SET_HASH_MAC [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:mac set hinzu, mit dem man MAC-Elemente (Ethernet-Adressen) in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.12 hash:net,port,net set support

CONFIG_IP_SET_HASH_NETPORTNET [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:net,port,net set hinzu, mit dem man zwei IPv4/IPv6-Subnetze und ein Protokoll/Port in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.13 hash:net set support

CONFIG_IP_SET_HASH_NET [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:net set hinzu, mit dem man IPv4/IPv6-Netzwerkadressen/Präfixe in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.14 hash:net,net set support

CONFIG_IP_SET_HASH_NETNET [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:net,net set hinzu, mit dem man IPv4/IPv6-Netzwerkadressen/Präfixpaare in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.15 hash:net,port set support

CONFIG_IP_SET_HASH_NETPORT [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:net,port set hinzu, mit dem man IPv4/IPv6-Netzwerkadressen/Präfix und Protokoll/Port-Paare als Elemente in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.16 hash:net,iface set support

CONFIG_IP_SET_HASH_NETIFACE [=m] [**M**]
Diese Option fügt die Unterstützung des Typs hash:net,port set hinzu, mit dem man IPv4/IPv6-Netzwerkadressen/Präfix und Schnittstellennamenpaare als Elemente in einem Set speichern kann. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.3.17 list:set set support

CONFIG_IP_SET_LIST_SET [=m] **[M]**
Mit dieser Option wird die Unterstützung des Typs list:set hinzugefügt. In dieser Art von Set kann man den Namen anderer Sets speichern und es bildet eine geordnete Vereinigung der Mitglieds-Sets. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4 IP virtual server support →

CONFIG_IP_VS [=m] **[M]**
Mit der Unterstützung von IP Virtual Server können Sie einen hochleistungsfähigen virtuellen Server auf der Grundlage eines Clusters von zwei oder mehr realen Servern erstellen. Diese Option muss für mindestens einen der Computer im Cluster aktiviert werden, der die eingehenden Verbindungen zu einer einzelnen IP-Adresse abfängt und sie an reale Server weiterleitet.
Es sind drei Techniken zur Verteilung von Anfragen implementiert: virtueller Server über NAT, virtueller Server über Tunneling und virtueller Server über direktes Routing. Mit Hilfe der verschiedenen Planungsalgorithmen kann ausgewählt werden, zu welchem Server die Verbindung geleitet wird, so dass ein Lastausgleich zwischen den Servern erreicht werden kann. Weitere Informationen und das Verwaltungsprogramm finden Sie unter der folgenden URL: `http://www.linuxvirtualserver.org/`.
Wenn Sie es im Kernel kompilieren wollen, geben Sie Y an. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.1 IPv6 support for IPVS

CONFIG_IP_VS_IPV6 [=y] **[Y]**
Hinzufügen von IPv6-Unterstützung zu IPVS. Sagen Sie Y, wenn Sie unsicher sind.

14.1.15.4.2 IP virtual server debugging

CONFIG_IP_VS_DEBUG [=n] **[N]**
Geben Sie hier Y ein, wenn Sie zusätzliche Meldungen erhalten möchten, die bei der Fehlersuche im Code des virtuellen IP-Servers nützlich sind. Sie können die Debug-Ebene in /proc/sys/net/ipv4/vs/debug_level ändern.

14.1.15.4.3 IPVS connection table size (the Nth power of 2)

CONFIG_IP_VS_TAB_BITS [=15] **[15]**
Größe der IPVS-Verbindungstabelle (die n-te Potenz von 2)
Die IPVS-Verbindungs-Hashtabelle verwendet das Verkettungsschema, um Hash-Kollisionen zu behandeln. Durch die Verwendung einer großen IPVS-Verbindungs-Hashtabelle werden Konflikte bei Hunderttausenden von Verbindungen in der Hashtabelle erheblich reduziert.
Beachten Sie, dass die Tabellengröße eine Potenz von 2 sein muss. Die Tabellengröße ist der Wert von 2 hoch der von Ihnen eingegebenen Zahl. Die zu wählende Zahl liegt zwischen 8 und 27 für 64BIT (sonst 20), die Standardzahl ist 12, was eine Tabellengröße von 4096 bedeutet. Geben Sie die Zahl nicht zu klein ein, sonst verlieren Sie Leistung. Sie können die Tabellengröße selbst anpassen, je nach Ihrer virtuellen Serveranwendung. Es ist gut, die Tabellengröße nicht viel kleiner als die Anzahl der Verbindungen pro Sekunde, multipliziert mit der durchschnittlichen Dauer der Verbindung in der Tabelle, festzulegen. Zum Beispiel, Ihr virtueller Server bekommt 200 Verbindungen pro Sekunde, die Verbindung dauert im Durchschnitt 200 Sekunden in der Verbindungstabelle, die Tabellengröße sollte nicht viel kleiner als 200x200 sein, es ist gut, die Tabellengröße 32768 (2**15) zu setzen. Ein weiterer Hinweis: Jede Verbindung belegt effektiv 128 Bytes und jeder Hash-Eintrag 8 Bytes, so dass Sie abschätzen können, wie viel Speicher für Ihre Box benötigt wird.
Sie können diese Zahl überschreiben, indem Sie den Modulparameter conn_tab_bits setzen oder indem Sie `ip_vs.conn_tab_bits=?` an die Kernel-Befehlszeile anhängen, wenn IP VS integriert kompiliert wurde.
Symbol: IP_VS_TAB_BITS [=15]
Typ: Ganzzahl (integer)
Bereich (range): [8 27]

***** IPVS transport protocol load balancing support *****
*(*** Unterstützung des IPVS-Transportprotokolls für den Lastausgleich ***)*

14.1.15.4.4 TCP load balancing support

CONFIG_IP_VS_PROTO_TCP [=y] **[Y]**
Diese Option aktiviert die Unterstützung des TCP-Transportprotokolls für den Lastausgleich. Sagen Sie Y, wenn Sie unsicher sind.

14.1.15.4.5 UDP load balancing support

CONFIG_IP_VS_PROTO_UDP [=y] **[Y]**
Diese Option aktiviert die Unterstützung des UDP-Transportprotokolls für den Lastausgleich. Sagen Sie Y, wenn Sie unsicher sind.

14.1.15.4.6 ESP load balancing support

CONFIG_IP_VS_PROTO_ESP [=y] **[Y]**
Diese Option aktiviert die Unterstützung des Transportprotokolls ESP (Encapsulation Security Payload) für den Lastausgleich. Sagen Sie Y, wenn Sie unsicher sind.

14.1.15.4.7 AH load balancing support

CONFIG_IP_VS_PROTO_AH [=y] **[Y]**
Diese Option aktiviert die Unterstützung für den Lastausgleich des AH (Authentication Header)-Transportprotokolls. Sagen Sie Y, wenn Sie unsicher sind.

14.1.15.4.8 SCTP load balancing support

CONFIG_IP_VS_PROTO_SCTP [=y] **[Y]**
Diese Option aktiviert die Unterstützung des SCTP-Transportprotokolls für den Lastausgleich. Sagen Sie Y, wenn Sie unsicher sind.

***** IPVS scheduler *****
*(*** IPVS-Scheduler/Zeitplaner ***)*

14.1.15.4.9 round-robin scheduling

CONFIG_IP_VS_RR [=m] **[M]**
Der Round-Robin-Scheduling-Algorithmus leitet die Netzverbindungen einfach nach dem Rotationsprinzip an verschiedene reale Server weiter. Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.10 weighted round-robin scheduling

CONFIG_IP_VS_WRR [=m] **[M]**
Der gewichtete Round-Robin-Planungsalgorithmus leitet Netzverbindungen auf der Grundlage von Servergewichten in einem Round-Robin-Verfahren an verschiedene reale Server weiter. Server mit höherer Gewichtung erhalten neue Verbindungen zuerst als solche mit geringerer Gewichtung, und Server mit höherer Gewichtung erhalten mehr Verbindungen als solche mit geringerer Gewichtung und Server mit gleicher Gewichtung erhalten gleiche Verbindungen. Wenn Sie es im Kernel kompilieren wollen, sagen Sie Y. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.11 least-connection scheduling

CONFIG_IP_VS_LC [=m] **[M]**
Der Least-Connection-Scheduling-Algorithmus leitet Netzwerkverbindungen an den Server mit der geringsten Anzahl aktiver Verbindungen weiter. Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.12 weighted least-connection scheduling

CONFIG_IP_VS_WLC [=m] **[M]**
Der gewichtete Least-Connection-Scheduling-Algorithmus leitet die Netzwerkverbindungen zu dem Server mit den wenigsten aktiven Verbindungen, normalisiert durch das Servergewicht. Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.13 weighted failover scheduling

CONFIG_IP_VS_FO [=m] [**M**]
Der gewichtete Failover-Planungsalgorithmus leitet die Netzwerkverbindungen an den Server mit der höchsten Gewichtung, der gerade verfügbar ist. Wenn Sie ihn im Kernel kompilieren wollen, geben Sie Y an. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.14 weighted overflow scheduling

CONFIG_IP_VS_OVF [=m] [**M**]
Der gewichtete Überlaufplanungsalgorithmus leitet die Netzwerkverbindungen zu dem Server mit dem höchsten Gewicht, der gerade verfügbar ist, und geht zum nächsten über, wenn die aktiven Verbindungen das Gewicht des Knotens überschreiten. Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.15 locality-based least-connection scheduling

CONFIG_IP_VS_LBLC [=m] [**M**]
Der ortsbezogene Planungsalgorithmus für die kleinste Verbindung ist für den IP-Lastausgleich bestimmt. Er wird normalerweise in Cache-Clustern verwendet. Dieser Algorithmus leitet Pakete, die für eine IP-Adresse bestimmt sind, in der Regel an ihren Server weiter, wenn der Server aktiv und ausgelastet ist. Wenn der Server überlastet ist (die Anzahl seiner aktiven Verbindungen ist größer als sein Gewicht) und es einen Server mit halber Auslastung gibt, wird dieser IP-Adresse der gewichtete Server mit der geringsten Verbindung zugewiesen. Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.16 locality-based least-connection with replication scheduling

CONFIG_IP_VS_LBLCR [=m] [**M**]
Der ortsbezogene Algorithmus zur Planung der kleinsten Verbindung mit Replikation ist ebenfalls für den IP-Lastausgleich bestimmt. Er wird normalerweise in Cache-Clustern verwendet. Er unterscheidet sich von der LBLC-Planung wie folgt: Der Lastverteiler unterhält Zuordnungen von einem Ziel zu einer Gruppe von Serverknoten, die das Ziel bedienen können. Anfragen für ein Ziel werden dem Knoten mit der geringsten Verbindung in der Servergruppe des Ziels zugewiesen. Wenn alle Knoten in der Servergruppe überlastet sind, wird ein Knoten mit der geringsten Verbindung im Cluster ausgewählt und der Servergruppe für das Ziel hinzugefügt. Wenn der Serversatz für die angegebene Zeit nicht geändert wurde, wird der am stärksten belastete Knoten aus dem Serversatz entfernt, um ein hohes Maß an Replikation zu vermeiden. Wenn Sie es im Kernel kompilieren wollen, geben Sie Y an. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.17 destination hashing scheduling

CONFIG_IP_VS_DH [=m] [**M**]
Der Ziel-Hash-Scheduling-Algorithmus weist den Servern Netzwerkverbindungen zu, indem er eine statisch zugewiesene Hash-Tabelle nach ihren Ziel-IP-Adressen durchsucht. Wenn Sie ihn im Kernel kompilieren wollen, geben Sie Y an. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.18 source hashing scheduling

CONFIG_IP_VS_SH [=m] [**M**]
Der Source-Hashing-Scheduling-Algorithmus weist den Servern Netzwerkverbindungen zu, indem er eine statisch zugewiesene Hash-Tabelle nach ihren Quell-IP-Adressen durchsucht. Wenn Sie ihn im Kernel kompilieren wollen, geben Sie Y an. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.19 maglev hashing scheduling

CONFIG_IP_VS_MH [=m] [**M**]
Der Maglev Consistent Hashing Scheduling Algorithmus stellt den Maglev Hashing Algorithmus von Google als IPVS Scheduler zur Verfügung. Er weist den Servern Netzwerkverbindungen zu, indem er eine statisch zugewiesene spezielle Hash-Tabelle, die so genannte Lookup-Tabelle, nachschlägt. Der Maglev-Hash-Algorithmus weist jedem Ziel eine Präferenzliste aller Positionen der Nachschlagetabelle zu.
Durch diesen Vorgang gibt das Maglev-Hashing jedem der Ziele einen nahezu gleichen Anteil an der

Nachschlagetabelle und sorgt für eine minimale Störung durch die Verwendung der Nachschlagetabelle. Wenn sich die Menge der Ziele ändert, wird eine Verbindung wahrscheinlich an dasselbe Ziel wie zuvor gesendet.
Wenn Sie es im Kernel kompilieren wollen, sagen Sie Y. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.20 shortest expected delay scheduling

CONFIG_IP_VS_SED [=m] **[M]**
Der Scheduling-Algorithmus mit der kürzesten erwarteten Verzögerung weist die Netzverbindungen dem Server mit der kürzesten erwarteten Verzögerung zu. Die erwartete Verzögerung, die der Auftrag erfährt, ist $(C_i + 1)/U_i$, wenn er an den i-ten Server gesendet wird, wobei C_i die Anzahl der Verbindungen auf dem i-ten Server und U_i die feste Dienstrate (Gewicht) des i-ten Servers ist.
Wenn Sie es im Kernel kompilieren wollen, sagen Sie Y. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.21 never queue scheduling

CONFIG_IP_VS_NQ [=m] **[M]**
Der Algorithmus für die Planung der Warteschlange „Never Queue“ basiert auf einem Modell mit zwei Geschwindigkeiten. Wenn ein ungenutzter Server verfügbar ist, wird der Auftrag an den ungenutzten Server geschickt, anstatt auf einen schnellen Server zu warten. Wenn kein freier Server verfügbar ist, wird der Auftrag an den Server geschickt, bei dem die erwartete Verzögerung am geringsten ist (Scheduling-Algorithmus mit der kürzesten erwarteten Verzögerung).
Wenn Sie ihn im Kernel kompilieren wollen, geben Sie Y an. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.22 weighted random twos choice least-connection scheduling

CONFIG_IP_VS_TWOS [=m] **[M]**
Der Algorithmus für die gewichtete zufällige Zweierauswahl der geringsten Verbindungen wählt zwei zufällige reale Server aus und leitet die Netzverbindungen zu dem Server mit den wenigsten aktiven Verbindungen, normiert durch das Servergewicht.
Wenn Sie ihn im Kernel kompilieren wollen, sagen Sie Y. Um ihn als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

***** IPVS SH scheduler *****
*(*** IPVS-SH-Scheduler/Zeitplaner ***)*

14.1.15.4.23 IPVS source hashing table size (the Nth power of 2)

CONFIG_IP_VS_SH_TAB_BITS [=8] **[8]**
Der Quell-Hashing-Scheduler ordnet Quell-IPs den in einer Hash-Tabelle gespeicherten Zielen zu. Diese Tabelle wird für jedes Ziel so lange abgearbeitet, bis alle Plätze in der Tabelle gefüllt sind. Wenn Gewichte verwendet werden, damit die Ziele mehr Verbindungen erhalten können, wird die Tabelle proportional zu den angegebenen Gewichten gekachelt. Die Tabelle muss groß genug sein, um alle Ziele, multipliziert mit ihren jeweiligen Gewichten, effektiv aufzunehmen. Symbol: IP_VS_SH_TAB_BITS [=8]
Typ: Ganzzahl (integer)
Bereich: [4 20]

***** IPVS MH scheduler *****
*(*** IPVS-MH-Scheduler/Zeitplaner ***)*

14.1.15.4.24 IPVS maglev hashing table index of size (the prime numbers)

CONFIG_IP_VS_MH_TAB_INDEX [=12] **[12]**
Der Maglev-Hashing-Scheduler ordnet Quell-IPs Zielen zu, die in einer Hash-Tabelle gespeichert sind. Diese Tabelle wird durch eine Präferenzliste der Positionen jedem Ziel zugewiesen, bis alle Slots in der Tabelle gefüllt sind. Der Index bestimmt die Primzahl für die Größe der Tabelle: 251, 509, 1021, 2039, 4093, 8191, 16381, 32749, 65521 oder 131071. Bei der Verwendung von Gewichtungen, die es den Zielen ermöglichen, mehr Verbindungen zu erhalten, wird der Tabelle ein Betrag proportional zu den angegebenen Gewichtungen zugewiesen. Die Tabelle muss groß genug sein, um alle Ziele, multipliziert mit ihren

jeweiligen Gewichtungen, effektiv aufzunehmen.
Symbol: IP_VS_MH_TAB_INDEX [=12]
Typ : Ganzzahl (integer)
Bereich : [8 17]

***** IPVS application helper *****
*(*** IPVS-Anwendungshilfe ***)*

14.1.15.4.25 FTP protocol helper
CONFIG_IP_VS_FTP [=m] **[M]**
FTP ist ein Protokoll, das IP-Adressen und/oder Portnummern in der Nutzlast überträgt. Im virtuellen Server über Network Address Translation können die IP-Adresse und die Portnummer des realen Servers nicht direkt an die Clients in FTP-Verbindungen gesendet werden, so dass ein FTP-Protokollhelfer erforderlich ist, um die Verbindung zu verfolgen und sie in die des virtuellen Dienstes zurückzuverwandeln.
Wenn Sie es im Kernel kompilieren wollen, sagen Sie Y. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.4.26 Netfilter connection tracking
CONFIG_IP_VS_NFCT [=y] **[Y]**
Durch die Unterstützung der Netfilter-Verbindungsverfolgung kann der IPVS-Verbindungsstatus zu Filterzwecken in das Netfilter-Framework exportiert werden.

14.1.15.4.27 SIP persistence engine
CONFIG_IP_VS_PE_SIP [=m] **[M]**
Persistenz auf Basis der SIP Call-ID zulassen

14.1.15.5 IP: Netfilter Configuration →
IP: Netzfilter-Konfiguration

14.1.15.5.1 IPv4 socket lookup support
CONFIG_NF_SOCKET_IPV4 [=m] **[M]**
Diese Option aktiviert die IPv4-Socket-Lookup-Infrastruktur. Dies ist für die Socket-Übereinstimmung {ip,nf}tables erforderlich.

14.1.15.5.2 IPv4 tproxy support
CONFIG_NF_TPROXY_IPV4 [=m] **[M]**
Für diese Option gibt es keine Hilfe.

14.1.15.5.3 IPv4 nf_tables support
CONFIG_NF_TABLES_IPV4 [=y] **[Y]**
Diese Option aktiviert die IPv4-Unterstützung für nf_tables.

14.1.15.5.3.1 IPv4 nf_tables packet duplication support
CONFIG_NFT_DUP_IPV4 [=y] **[Y]**
Dieses Modul ermöglicht die Unterstützung der IPv4-Paketduplikation für nf_tables.

14.1.15.5.3.2 nf_tables fib / ip route lookup support
CONFIG_NFT_FIB_IPV4 [=m] **[M]**
Dieses Modul ermöglicht IPv4-FIB-Lookups, z. B. für Reverse Path Filtering. Es ermöglicht auch die Abfrage der FIB nach dem Routentyp, z. B. lokal, Unicast, Multicast oder Blackhole.

14.1.15.5.4 ARP nf_tables support
CONFIG_NF_TABLES_ARP [=y] **[Y]**
Diese Option aktiviert die ARP-Unterstützung für nf_tables.

14.1.15.5.5 Netfilter IPv4 packet duplication to alternate destination

CONFIG_NF_DUP_IPV4 [=m] **[M]**
Diese Option aktiviert den nf_dup_ipv4-Kern, der ein IPv4-Paket dupliziert, um es an ein anderes Ziel umzuleiten.

14.1.15.5.6 ARP packet logging

CONFIG_NF_LOG_ARP [=m] **[M]**
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NF_LOG_SYSLOG aus.

14.1.15.5.7 IPv4 packet logging

CONFIG_NF_LOG_IPV4 [=m] **[M]**
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NF_LOG_SYSLOG aus.

14.1.15.5.8 IPv4 packet rejection

CONFIG_NF_REJECT_IPV4 [=m] **[M]**
Für diese Option gibt es keine Hilfe.

14.1.15.5.9 Basic SNMP-ALG support

CONFIG_NF_NAT_SNMP_BASIC [=m] **[M]**
Dieses Modul implementiert ein Application Layer Gateway (ALG) für SNMP-Payloads. In Verbindung mit NAT ermöglicht es einem Netzwerkmanagementsystem den Zugang zu mehreren privaten Netzwerken mit widersprüchlichen Adressen. Dabei werden die IP-Adressen in den SNMP-Payloads so geändert, dass sie mit der IP-Layer-NAT-Zuordnung übereinstimmen. Dies ist die „Grundform“ von SNMP-ALG, wie in RFC 2962 beschrieben.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10 IP tables support (required for filtering/masq/NAT)

CONFIG_IP_NF_IPTABLES [=m] **[M]**
iptables ist ein allgemeines, erweiterbares Framework zur Paketidentifizierung. Die Subsysteme fuer Paketfilterung und vollstaendiges NAT (Masquerading, Portweiterleitung, etc.) benutzen dies nun: sage hier Y oder M, wenn Du eines davon benutzen willst. Um es als Modul zu kompilieren, wähle hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.1 “ah” match support

CONFIG_IP_NF_MATCH_AH [=m] **[M]**
Mit dieser Match-Erweiterung können Sie einen Bereich von SPIs im AH-Header von IPSec-Paketen abgleichen. Um sie als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.2 “ecn” match support

CONFIG_IP_NF_MATCH_AH [=m] **[M]**
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_MATCH_ECN aus.

14.1.15.5.10.3 “rpfilter” reverse path filter match support

CONFIG_IP_NF_MATCH_RPFILTER [=m] **[M]**
Mit dieser Option können Sie Pakete abgleichen, deren Antworten über die Schnittstelle hinausgehen würden, über die das Paket eingegangen ist. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N. Das Modul wird `ipt_rpfilter` heißen.

14.1.15.5.10.4 “ttl” match support

CONFIG_IP_NF_MATCH_TTL [=m] **[M]**
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_MATCH_HL aus.

14.1.15.5.10.5 Packet filtering

CONFIG_IP_NF_FILTER [=m] [**M**]
Paketfilterung definiert eine Tabelle `filter`, die eine Reihe von Regeln für einfache Paketfilterung bei der lokalen Eingabe, Weiterleitung und lokalen Ausgabe enthält. Siehe die Manpage für iptables(8).
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.5.1 REJECT target support

CONFIG_IP_NF_TARGET_REJECT [=m] [**M**]
Mit dem REJECT-Ziel kann eine Filterregel angeben, dass als Antwort auf ein eingehendes Paket ein ICMP-Fehler ausgegeben werden soll, anstatt es stillschweigend zu verwerfen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.6 SYNPROXY target support

CONFIG_IP_NF_TARGET_SYNPROXY [=m] [**M**]
Das SYNPROXY-Ziel ermöglicht es Ihnen, TCP-Verbindungen abzufangen und sie unter Verwendung von Syncookies aufzubauen, bevor sie an den Server weitergeleitet werden. Auf diese Weise können Sie die Verfolgung von Verbindungen und die Nutzung von Serverressourcen bei SYN-Flood-Angriffen vermeiden.
Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.7 iptables NAT support

CONFIG_IP_NF_NAT [=m] [**M**]
Dies aktiviert die `nat`-Tabelle in iptables. Dies erlaubt Masquerading, Portweiterleitung und andere Formen der vollständigen Network Address Port Translation. Um es als Modul zu kompilieren, wähle hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.7.1 MASQUERADE target support

CONFIG_IP_NF_TARGET_MASQUERADE [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt NETFILTER_XT_TARGET_MASQUERADE aus.

14.1.15.5.10.7.2 NETMAP target support

CONFIG_IP_NF_TARGET_NETMAP [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_TARGET_NETMAP aus.

14.1.15.5.10.7.3 REDIRECT target support

CONFIG_IP_NF_TARGET_REDIRECT [=m] [**M**]
Dies ist eine rückwärtskompatible Option zur Bequemlichkeit des Benutzers (z. B. bei der Ausführung von oldconfig). Sie wählt CONFIG_NETFILTER_XT_TARGET_REDIRECT aus.

14.1.15.5.10.8 Packet mangling

CONFIG_IP_NF_MANGLE [=m] [**M**]
Diese Option fügt eine „mangle“-Tabelle zu iptables hinzu: siehe die Manpage für iptables(8). Diese Tabelle wird fuer verschiedene Paketveraenderungen benutzt, die beeinflussen koennen, wie das Paket weitergeleitet wird. Um sie als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.8.1 ECN target support

CONFIG_IP_NF_TARGET_ECN [=m] [**M**]
Diese Option fügt ein „ECN“-Ziel hinzu, das in der iptables-Mangeltabelle verwendet werden kann. Sie können dieses Ziel verwenden, um die ECN-Bits aus dem IPv4-Header eines IP-Pakets zu entfernen. Dies ist besonders nützlich, wenn Sie bestehende ECN-Blackholes im Internet umgehen müssen, aber die ECN-Unterstützung nicht generell abschalten wollen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.8.2 “TTL” target support

CONFIG_IP_NF_TARGET_TTL [=m] [**M**]
Dies ist eine rückwärtskompatible Option, die dem Benutzer die Arbeit erleichtert (z. B. wenn er oldconfig verwendet). Sie wählt CONFIG_NETFILTER_XT_TARGET_HL aus.

14.1.15.5.10.9 raw table support (required for NOTRACK/TRACE)

CONFIG_IP_NF_RAW [=m] [**M**]
Diese Option fügt eine "rohe"Tabelle zu iptables hinzu. Diese Tabelle ist die allererste im Netfilter-Framework und hakt sich bei den PREROUTING- und OUTPUT-Ketten ein. Wenn Sie sie als Modul kompilieren wollen, sagen Sie hier M und lesen Sie <file:Documentation/kbuild/modules.rst>. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.10.10 Security table

CONFIG_IP_NF_SECURITY [=m] [**M**]
Diese Option fügt eine „Security“-Tabelle zu iptables hinzu, für die Verwendung mit der Mandatory Access Control (MAC) Richtlinie. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.11 ARP tables support

CONFIG_IP_NF_ARPTABLES [=m] [**M**]
arptables ist ein allgemeiner, erweiterbarer Rahmen für die Paketidentifizierung. Die ARP-Paketfilter- und -Manipulations-Subsysteme verwenden es: Sagen Sie hier Y oder M, wenn Sie eines von beiden verwenden wollen. Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.

14.1.15.5.11.1 ARP packet filtering

CONFIG_IP_NF_ARPFILTER [=m] [**M**]
Die ARP-Paketfilterung definiert eine Tabelle „filter“, die eine Reihe von Regeln für die einfache ARP-Paketfilterung am lokalen Eingang und am lokalen Ausgang enthält. Auf einer Bridge können Sie auch Filterregeln für weitergeleitete ARP-Pakete angeben. Siehe die Manpage für arptables(8). Um es als Modul zu kompilieren, wählen Sie hier M. Wenn Sie unsicher sind, sagen Sie N.