

# Linux Configuration V6.19

## 0.1 Einführung

Dieses Dokument dient zur Beschreibung von diversen Einstellungen bei der Konfiguration zum Beispiel mittels `make nconfig` oder `make menuconfig` unter Linux.

Es wird nicht näher darauf eingegangen, wie der Kernel kompiliert wird oder welche Voreinstellungen, Programme usw. zum Kompilieren benötigt werden. Abweichungen zur bestehenden Arch-Linux-Konfiguration werden immer **gelb** markiert.

Original Texte, sofern geliefert (englischer Sprache) werden *kursiv* und etwas kleiner geschrieben. Zu Beginn der jeweiligen Konfigurationszeile wird der Standardwert (Default) angezeigt. Mein Vorschlag folgt danach.

Zum Beispiel bei `CONFIG_WERROR` `[=n] [Y]`

Hier ist der Standardwert ein Nein `[n]`, meine persönliche Einstellung ein Ja `[Y]`.

## 0.2 Konfiguration für ein verteiltes Kompilieren auf mehreren Rechnern

Sie sollten schon einiges an Erfahrung mit dem Kompilieren unter Linux mitbringen. Wenn Sie mit Linux Arch arbeiten, muss auf jedem Rechner das Programm `distcc` installiert sein. Man unterscheidet nun den **Client** auf dem üblicherweise das Kompilat erstellt wird und die weiteren beteiligten Rechner als **Volunteers**.

Nähere Informationen siehe <https://wiki.archlinux.org/title/Distcc>.

©KW4NZ, Thomas Kuschel

Wenn Sie Korrekturen wünschen, dann schicken Sie dies mit Erläuterungen und dem Hinweis auf die obenstehende Version V6.19 an: [oe1tk@gmail.com](mailto:oe1tk@gmail.com)

This is pdfTeX, Version 3.141592653-2.6-1.40.28 (TeX Live 2025) kpathsea version 6.4.1

## 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.

*Some drivers can be compiled on a different platform than they are intended to be run on. Despite they cannot be loaded there (or even when they load they cannot be used due to missing HW support), developers still, opposing to distributors, might want to build such drivers to compile-test them. If you are a developer and want to build everything available, say Y here. If you are a user/distributor, say N here to exclude useless drivers to be distributed.*

### 1.2 Compile the kernel with warnings as errors

CONFIG\_WERROR [=n] [ ]

*Den Kernel mit Fehlermeldungen bei Warnungen kompilieren*

Ein Build sollte keine Compiler-Warnungen ausgeben, dies aktiviert dieFlags **-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.

*A kernel build should not cause any compiler warnings, and this enables the ‘-Werror’ (for C) and ‘-Dwarnings’ (for Rust) flags to enforce that rule by default. Certain warnings from other tools such as the linker may be upgraded to errors with this option as well.*

*However, if you have a new (or very old) compiler or linker with odd and unusual warnings, or you have some architecture with problems, you may need to disable this config option in order to successfully build the kernel.*

*If in doubt, say Y.*

Der Kernel wird möglicherweise mit Warnungen kompiliert, deshalb sollten wir dies ausschalten.

### 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 **localversion\*** als Objekt und im Quellbaum, in dieser Reihenfolge angezeigt. Die Zeichenkette darf maximal 64 Zeichen lang sein.

*Append an extra string to the end of your kernel version. This will show up when you type uname, for example.*

*The string you set here will be appended after the contents of any files with a filename matching localversion\* in your object and source tree, in that order. Your total string can be a maximum of 64 characters.*

### 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 **-gxxxxxxx** 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.)

*This will try to automatically determine if the current tree is a release tree by looking for git tags that belong to the*

73 *current top of tree revision.*  
74 *A string of the format -gxxxxxxx will be added to the localversion if a git-based tree is found. The string generated by*  
75 *this will be appended after any matching localversion\* files, and after the value set in CONFIG\_LOCALVERSION.*  
76 *(The actual string used here is the first 12 characters produced by running the command:*  
77 *\$ git rev-parse -verify HEAD*  
78 *which is done within the script "scripts/setlocalversion".)*

## 79 1.5 Build ID Salt

80 CONFIG\_BUILD\_SALT [=] [ ]

81 Type: string

82 Dies wird verwendet, um die Binaries und ihre Debug-Infos zu verknüpfen. Wenn diese Option gesetzt ist,  
83 dann wird dieser Wert in die Berechnung der Build-ID einbezogen. Wird von Distributionen verwendet, die  
84 sicherstellen wollen, dass es eindeutige IDs zwischen verschiedenen Builds gibt. Üblicherweise brauchen  
85 wir das nicht.

86 *The build ID is used to link binaries and their debug info. Setting this option will use the value in the calculation*  
87 *of the build id. This is mostly useful for distributions which want to ensure the build is unique between builds. It's*  
88 *safe to leave the default.*

## 89 1.6 Kernel compression mode →

90 Der Linux-Kernel ist eine Art selbstextrahierende, ausführbare Datei. Es stehen mehrere Kompressi-  
91 onsalgorithmen zur Verfügung, die sich in Effizienz, Kompressions- und Dekompressionsgeschwindigkeit  
92 unterscheiden. Die Komprimierungsgeschwindigkeit ist nur bei der Erstellung eines Kernels relevant. Die  
93 Dekomprimierungsgeschwindigkeit ist bei jedem Systemstart von Bedeutung. (Eine ältere Version dieser  
94 Funktionalität (nur bzip2) für 2.4 wurde von Christian Ludwig bereitgestellt)

95 Hohe Komprimierungsoptionen sind vor allem für Benutzer nützlich, die wenig Festplattenplatz zur  
96 Verfügung haben (embedded systems), für die aber die Ram-Größe weniger wichtig ist.

97 Überblick: Gzip werden von den älteren Kernelversionen unterstützt,

98 Arch Linux (seit Linux/x86 5.9.0) Standard: ZSTD (former: XZ seit 4.14.4, predecessor GZIP,XZ)

99 Debian 11.6: XZ

100 @TODO Weitere Linux Distributionen

101 *The linux kernel is a kind of self-extracting executable. Several compression algorithms are available, which differ*  
102 *in efficiency, compression and decompression speed. Compression speed is only relevant when building a kernel.*  
103 *Decompression speed is relevant at each boot. If you have any problems with bzip2 or lzma compressed kernels,*  
104 *mail me (Alain Knaff) <alainknaff.lu>. (An older version of this functionality (bzip2 only), for 2.4, was supplied*  
105 *by Christian Ludwig)*

106 *High compression options are mostly useful for users, who are low on disk space (embedded systems), but for whom*  
107 *ram size matters less. If in doubt, select 'gzip'*

### 108 1.6.1 Gzip

109 CONFIG\_KERNEL\_GZIP [=n] [ ]

110 Die alte und bewährte gzip-Kompression. Sie bietet ein gutes Gleichgewicht zwischen Kompressionsrate  
111 und Dekompressionsgeschwindigkeit.

112 *The old and tried gzip compression. It provides a good balance between compression ratio and decompression speed.*

### 113 1.6.2 Bzip2

114 CONFIG\_KERNEL\_BZIP2 [=n] [ ]

115 Die Kompressionsrate und auch die Geschwindigkeit der ist durchschnittlich. Die Geschwindigkeit der  
116 Dekomprimierung ist die langsamste. Größe des Kernels ist etwa 10 % kleiner im Vergleich zu GZIP. Es  
117 benötigt auch einen großen Speicherbereich, bei modernen Kernen benötigt man zumindest 8 MB RAM  
118 oder mehr beim Booten.

119 *Its compression ratio and speed is intermediate. Decompression speed is slowest among the choices. The kernel*  
120 *size is about 10 % smaller with bzip2, in comparison to gzip. Bzip2 uses a large amount of memory. For modern*  
121 *kernels you will need at least 8MB RAM or more for booting.*

### 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.

*This compression algorithm's ratio is best. Decompression speed is between gzip and bzip2. Compression is slowest. The kernel size is about 33 % smaller with LZMA in comparison to 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.

*XZ uses the LZMA2 algorithm and instruction set specific BCJ filters which can improve compression ratio of executable code. The size of the kernel is about 30 % smaller with XZ in comparison to gzip. On architectures for which there is a BCJ filter (i386, x86\_64, ARM, ARM64, RISC-V, big endian PowerPC, and SPARC), XZ will create a few percent smaller kernel than plain LZMA.*

*The speed is about the same as with LZMA: The decompression speed of XZ is better than that of bzip2 but worse than gzip and LZO. Compression is slow.*

### 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.

*Its compression ratio is the poorest among the choices. The kernel size is about 10 % bigger than gzip; however its speed (both compression and decompression) is the fastest.*

### 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.

*LZ4 is an LZ77-type compressor with a fixed, byte-oriented encoding. A preliminary version of LZ4 de/compression tool is available at <https://code.google.com/p/lz4/>.*

*Its compression ratio is worse than LZO. The size of the kernel is about 8 % bigger than LZO. But the decompression speed is faster than 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.

*ZSTD is a compression algorithm targeting intermediate compression with fast decompression speed. It will compress better than GZIP and decompress around the same speed as LZO, but slower than LZ4. You will need at least 192 KB RAM or more for booting. The zstd command line tool is required for compression.*

## 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.

*This option determines the default init for the system if no `init=option` is passed on the kernel command line. If the requested path is not present, we will still then move on to attempting further locations (e.g. `/sbin/init`, etc). If this is empty, we will just use the fallback list when `init=` is not passed.*

## 1.8 Default hostname

CONFIG\_DEFAULT\_HOSTNAME [=archlinux] [=orange]

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.

*This option determines the default system hostname before userspace calls `sethostname(2)`. The kernel traditionally uses `"(none)"` here, but you may wish to use a different default here to make a minimal system more usable with less configuration.*

## 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. Insbesondere, wenn Sie den DOS-Emulator `dosemu` unter Linux laufen lassen wollen (lesen Sie das DOSEMU-HOWTO, verfügbar unter <http://www.tldp.org/docs.html#howto>), müssen Sie hier Y sagen.

Sie können Dokumentation über IPC mit `info ipc` und auch in Abschnitt 6.4 des Linux Programmer's Guide finden, verfügbar unter <http://www.tldp.org/guides.html>.

*Inter Process Communication is a suite of library functions and system calls which let processes (running programs) synchronize and exchange information. It is generally considered to be a good thing, and some programs won't run unless you say Y here. In particular, if you want to run the DOS emulator `dosemu` under Linux (read the DOSEMU-HOWTO, available from <http://www.tldp.org/docs.html#howto>), you'll need to say Y here. You can find documentation about IPC with `"info ipc"` and also in section 6.4 of the Linux Programmer's Guide, available from <http://www.tldp.org/guides.html>.*

## 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 als Dateisystem mit dem Namen „`mqueue`“ sichtbar und können irgendwo eingehängt werden, wenn Sie Dateisystemoperationen auf Nachrichtenwarteschlangen durchführen wollen. Wenn Sie unsicher sind, sagen Sie Y.

*POSIX variant of message queues is a part of IPC. In POSIX message queues every message has a priority which decides about succession of receiving it by a process. If you want to compile and run programs written e.g. for Solaris with use of its POSIX message queues (functions `mq_*`) say Y here.*

*POSIX message queues are visible as a filesystem called `'mqueue'` and can be mounted somewhere if you want to do filesystem operations on message queues.*

*If unsure, say Y.*

## 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-/Schlüsselbund-Änderungsbachrichtigungen (key/keyring) und Gerätebenachrichtigungen verwendet werden.

Siehe `Documentation/core-api/watch_queue.rst`.

*This is a general notification queue for the kernel to pass events to userspace by splicing them into pipes. It can be used in conjunction with watches for key/keyring change notifications and device notifications.*

*See `Documentation/core-api/watch_queue.rst`*

Bemerkung: Bei Debian Bullseye and Bookworm ist dies nicht gesetzt (N).

Eventuell benützt dies bereits GNOME, wir kommen derzeit vermutlich ohne aus.

## 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.

*Enabling this option adds the system calls `process_vm_readv` and `process_vm_writev` which allow a process with the correct privileges to directly read from or write to another process' address space. See the man page for more details.*

uselib syscall (for libc5 and earlier) bis 6.15

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.

*This option enables the `uselib` syscall, a system call used in the dynamic linker from `libc5` and earlier. `glibc` does not use this system call. If you intend to run programs built on `libc5` or earlier, you may need to enable this syscall. Current systems running `glibc` can safely disable this.*

Bemerkung: Debian Bullseye verwendet dies noch (Y).

## 1.13 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.

*Enable auditing infrastructure that can be used with another kernel subsystem, such as SELinux (which requires this for logging of `avc` messages output).*

*System call auditing is included on architectures which support it.*

## 1.14 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.

*This interface lets you select features and parameters for the build. Features can either be built-in, modularized, or ignored. Parameters must be entered in as decimal or hexadecimal numbers or text.*

### 1.14.1 Expose irq internals in debugfs

CONFIG\_GENERIC\_IRQ\_DEBUGFS [=n] [ ]

Legt interne Zustandsinformationen über `debugfs` offen. Hauptsächlich für Entwickler und zur Fehlersuche

bei schwer zu diagnostizierenden Interrupt-Problemen. Wenn Sie nicht wissen, was Sie hier tun sollen, sagen Sie N.

*Exposes internal state information through debugfs. Mostly for developers and debugging of hard to diagnose interrupt problems. If you don't know what to do here, say N.*

## 1.15 Timers subsystem →

*Teilsystem Zeitgeber*

### 1.15.1 Timer tick handling →

Sie müssen aus den folgenden drei Möglichkeiten eine wählen:

#### 1.15.1.1 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.

*This option keeps the tick running periodically at a constant rate, even when the CPU doesn't need it.*

#### 1.15.1.2 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.

*This option enables a tickless idle system: timer interrupts will only trigger on an as-needed basis when the system is idle. This is usually interesting for energy saving.*

*Most of the time you want to say Y here.*

#### 1.15.1.3 Full dynticks system (tickless)

CONFIG\_NO\_HZ\_FULL [=y] [Y]

Adaptiv versuchen, die Ticks abzuschalten, wann immer dies möglich ist, auch wenn die CPU Aufgaben ausführt. Normalerweise erfordert dies die Ausführung einer einzelnen Aufgabe auf der CPU. Die Chancen für einen ticklosen Betrieb sind am größten, wenn die Aufgabe größtenteils im Userspace läuft und wenig Kernelaktivität aufweist. Sie müssen den Boot-Parameter nohz\_full mit dem gewünschten Bereich von dynticks CPUs auffüllen, um ihn zu verwenden. Dies wird auf Kosten eines gewissen Overheads bei Benutzer ↔ Kernel-Übergängen implementiert: syscalls, exceptions und interrupts.

Standardmäßig, ohne Übergabe des nohz\_full-Parameters, verhält sich dies genau wie NO\_HZ\_IDLE. Wird bei Linux-Distributionen ausgewählt.

*Adaptively try to shutdown the tick whenever possible, even when the CPU is running tasks. Typically this requires running a single task on the CPU. Chances for running tickless are maximized when the task mostly runs in userspace and has few kernel activity.*

*You need to fill up the nohz\_full boot parameter with the desired range of dynticks CPUs to use it. This is implemented at the expense of some overhead in user ↔ kernel transitions: syscalls, exceptions and interrupts.*

*By default, without passing the nohz\_full parameter, this behaves just like NO\_HZ\_IDLE. If you're a distro say Y.*

### 1.15.2 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-erweiterter Ruhezustand und taktlose CPU-Zeitabrechnung. 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.

*The major pre-requirement for full dynticks to work is to support the user context tracking subsystem. But there are also other dependencies to provide in order to make the full dynticks working.*

*This option stands for testing when an arch implements the user context tracking backend but doesn't yet fulfill all the requirements to make the full dynticks feature working. Without the full dynticks, there is no way to test the support for user context tracking and the subsystems that rely on it: RCU userspace extended quiescent state and tickless cputime accounting. This option copes with the absence of the full dynticks subsystem by forcing the user context tracking on all CPUs in the system.*

*Say Y only if you're working on the development of an architecture backend for the user context tracking.*

*Say N otherwise, this option brings an overhead that you don't want in production.*

### 1.15.3 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 Abwärtskompatibilität mit älteren Konfigurationsdateien zu gewährleisten.*

*This is the old config entry that enables dynticks idle. We keep it around for a little while to enforce backward compatibility with older config files.*

Alte Dynticks-Konfiguration wird nicht mehr unterstützt.

### 1.15.4 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.

*This option enables high resolution timer support. If your hardware is not capable then this option only increases the size of the kernel image.*

### 1.15.5 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

*Specify the maximum amount of allowable watchdog skew in microseconds before reporting the clocksource to be unstable. The default is based on a half-second clocksource watchdog interval and NTP's maximum frequency drift of 500 parts per million. If the clocksource is good enough for NTP, it is good enough for the clocksource watchdog!*

### 1.15.6 Enable auxiliary POSIX clocks

CONFIG\_POSIX\_AUX\_CLOCKS [=n] [N]

Hilfs-POSIX-Uhren sind Uhren, die unabhängig vom Kernzeitgeber gesteuert werden können, der die MONOTONIC-, REALTIME-, BOOTTIME- und TAI-Uhren steuert. Sie sind nützlich, um beispielsweise lockless time accessors für unabhängige PTP-Uhren und andere Uhrdomänen bereitzustellen, die nicht mit dem TAI/NTP-Zeitbegriff korrelieren.

*Auxiliary POSIX clocks are clocks which can be steered independently of the core timekeeper, which controls the MONOTONIC, REALTIME, BOOTTIME and TAI clocks. They are useful to provide e.g. lockless time accessors to independent PTP clocks and other clock domains, which are not correlated to the TAI/NTP notion of time.*

## 1.16 BPF subsystem →

Berkeley Packet Filter, Firewall-Filtertechnik im Kernel

### 1.16.1 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.

*Enable the bpf() system call that allows to manipulate BPF programs and maps via file descriptors.*

### 1.16.2 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)
```

*BPF programs are normally handled by a BPF interpreter. This option allows the kernel to generate native code when a program is loaded into the kernel. This will significantly speed-up processing of BPF programs.*

*Note, an admin should enable this feature changing:*

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

#### 1.16.2.1 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.

*Enables BPF JIT and removes BPF interpreter to avoid speculative execution of BPF instructions by the interpreter. When CONFIG\_BPF\_JIT\_ALWAYS\_ON is enabled, /proc/sys/net/core/bpf\_jit\_enable is permanently set to 1 and setting any other value than that will return failure.*

### 1.16.3 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.

*Disables unprivileged BPF by default by setting the corresponding /proc/sys/kernel/unprivileged\_bpf\_disabled knob to 2. An admin can still reenale it by setting it to 0 later on, or permanently disable it by setting it to 1 (from which no other transition to 0 is possible anymore). Unprivileged BPF could be used to exploit certain potential speculative execution side-channel vulnerabilities on unmitigated affected hardware.*

*If you are unsure how to answer this question, answer Y.*

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

CONFIG\_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-Einhängpunkt eingefügt werden, was bei der Fehlersuche und der Untersuchung von BPF-Programmen und -Maps nützlich ist.

*This builds kernel module with several embedded BPF programs that are pinned into BPF FS mount point as human readable files that are useful in debugging and introspection of BPF programs and maps.*

### 1.16.5 Enable BPF LSM Instrumentation

CONFIG\_BPF\_LSM [=y] [Y]

Ermöglicht die Instrumentierung der Sicherheits-Hooks mit BPF-Programmen zur Implementierung dynamischer MAC- und Audit-Richtlinien. Wenn Sie sich bei der Beantwortung dieser Frage unsicher sind, antworten Sie mit N.

*Enables instrumentation of the security hooks with BPF programs for implementing dynamic MAC and Audit Policies. If you are unsure how to answer this question, answer N.*

#### 1.16.5.1 bpf\_preload kernel module

*Dies ist nur sichtbar wenn der übergeordnete Punkt aktiviert ist.*

CONFIG\_BPF\_PRELOAD\_UMD [=m] [M]

Damit wird das Kernelmodul bpf\_preload mit eingebetteten BPF-Programmen für die Introspektion in bpffs erstellt.

*This builds bpf\_preload kernel module with embedded BPF programs for introspection in bpffs.*

### 1.16.6 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.

*Enables instrumentation of the security hooks with BPF programs for implementing dynamic MAC and Audit Policies.*

*If you are unsure how to answer this question, answer N.*

### 1.17 Preemption Model (Scheduler controlled preemption model) →

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

#### 1.17.1 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.

*This is the traditional Linux preemption model, geared towards throughput. It will still provide good latencies most of the time, but there are no guarantees and occasional longer delays are possible.*

*Select this option if you are building a kernel for a server or scientific/computation system, or if you want to maximize the raw processing power of the kernel, irrespective of scheduling latencies.*

#### 1.17.2 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.

*This option reduces the latency of the kernel by adding more “explicit preemption points” to the kernel code. These new preemption points have been selected to reduce the maximum latency of rescheduling, providing faster application reactions, at the cost of slightly lower throughput.*

*This allows reaction to interactive events by allowing a low priority process to voluntarily preempt itself even if it is in kernel mode executing a system call. This allows applications to run more ‘smoothly’ even when the system is*

456 *under load.*  
457 *Select this if you are building a kernel for a desktop system.*

### 458 1.17.3 Preemptible Kernel (Low-Latency Desktop)

459 CONFIG\_PREEMPT [=y] [N]

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

465 *This option reduces the latency of the kernel by making all kernel code (that is not executing in a critical section)*  
466 *preemptible. This allows reaction to interactive events by permitting a low priority process to be preempted*  
467 *involuntarily even if it is in kernel mode executing a system call and would otherwise not be about to reach a*  
468 *natural preemption point.*

469 *This allows applications to run more ‘smoothly’ even when the system is under load, at the cost of slightly lower*  
470 *throughput and a slight runtime overhead to kernel code.*

471 *Select this if you are building a kernel for a desktop or embedded system with latency requirements in the milliseconds*  
472 *range.*

### 473 1.17.4 Scheduler controlled preemption model seit 6.13

474 CONFIG\_PREEMPT\_LAZY [=n] [Y]

475 Diese Option bietet ein Scheduler-gesteuertes Preemption-Modell, das dem vollständigen Preemption-Mo-  
476 dell grundsätzlich ähnelt, aber weniger darauf bedacht ist, SCHED\_NORMAL-Tasks zu preemptieren,  
477 um die Preemption von Lock-Holder-Tasks zu reduzieren und einen Teil der Leistungsgewinne aus der  
478 Verwendung von Voluntary Preemption wiederherzustellen.

479 *This option provides a scheduler driven preemption model that is fundamentally similar to full preemption, but is*  
480 *less eager to preempt SCHED\_NORMAL tasks in an attempt to reduce lock holder preemption and recover some*  
481 *of the performance gains seen from using Voluntary preemption.*

482 Wir testen die neue Einstellung des LAZY-Modes, deshalb hier seit Linux 6.14.2 ein Y.

## 483 1.18 Fully Preemptible Kernel (Real-Time)

484 CONFIG\_PREEMPT\_RT [=n] [N]

485 Mit dieser Option wird der Kernel zu einem Echtzeit-Kernel, indem verschiedene Sperrprimitive (spinlocks,  
486 rwlocks usw.) durch präemptive Varianten mit Prioritätsvererbung ersetzt werden, Interrupt-Threading  
487 erzwungen wird und Mechanismen zur Unterbrechung langer nicht präemptiver Abschnitte eingeführt  
488 werden. Dadurch wird der Kernel mit Ausnahme von sehr einfachen und kritischen Codepfaden (Ein-  
489 stiegscode, Scheduler, Low-Level-Interrupt-Handling) vollständig preemptibel und bringt die meisten  
490 Ausführungskontexte unter die Kontrolle des Schedulers. Wählen Sie dies, wenn Sie einen Kernel für  
491 Systeme erstellen, die Echtzeitgarantien erfordern.

492 *This option turns the kernel into a real-time kernel by replacing various locking primitives (spinlocks, rwlocks, etc.)*  
493 *with preemptible priority-inheritance aware variants, enforcing interrupt threading and introducing mechanisms to*  
494 *break up long non-preemptible sections. This makes the kernel, except for very low level and critical code paths*  
495 *(entry code, scheduler, low level interrupt handling) fully preemptible and brings most execution contexts under*  
496 *scheduler control.*

497 *Select this if you are building a kernel for systems which require real-time guarantees.*

### 498 1.18.1 Enforce softirq synchronisation on PREEMPT\_RT seit 6.18

499 CONFIG\_PREEMPT\_RT\_NEEDS\_BH\_LOCK [=n] [N]

500 Erzwingt die Synchronisation über den Softirqs-Kontext hinweg. Bei PREEMPT\_RT ist der Softirq präemp-  
501 tiv. Dadurch wird dieselbe BLK-Semantik pro CPU erzwungen, die auch in Nicht-PREEMPT\_RT-Builds  
502 vorhanden ist. Dies sollte nicht erforderlich sein, da Sperren pro CPU hinzugefügt wurden, um die BKL  
503 pro CPU zu vermeiden.

504 Dieser Schalter bietet das alte Verhalten zu Testzwecken. Wählen Sie diese Option, wenn Sie einen  
505 Fehler bei vorrangigen Softirqs vermuten und das alte synchronisierte Verhalten testen möchten.

Enforce synchronisation across the softirqs context. On `PREEMPT_RT` the softirq is preemptible. This enforces the same per-CPU BLK semantic non-`PREEMPT_RT` builds have. This should not be needed because per-CPU locks were added to avoid the per-CPU BKL. This switch provides the old behaviour for testing reasons. Select this if you suspect an error with preemptible softirq and want test the old synchronized behaviour.

## 1.19 Preemption behaviour defined on boot

`CONFIG_PREEMPT_DYNAMIC` [=y] [N]

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.

*This option allows to define the preemption model on the kernel command line parameter and thus override the default preemption model defined during compile time.*

*The feature is primarily interesting for Linux distributions which provide a pre-built kernel binary to reduce the number of kernel flavors they offer while still offering different usecases.*

*The runtime overhead is negligible with `HAVE_STATIC_CALL_INLINE` enabled but if runtime patching is not available for the specific architecture then the potential overhead should be considered.*

*Interesting if you want the same pre-built kernel should be used for both Server and Desktop workloads.*

Wir setzen dies auf Nein, da wir wissen, dass der Kernal für den Desktop kompiliert wird.

## 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.

*This option permits Core Scheduling, a means of coordinated task selection across SMT siblings. When enabled – see `prctl(PR_SCHED_CORE)` – task selection ensures that all SMT siblings will execute a task from the same ‘core group’, forcing idle when no matching task is found.*

*Use of this feature includes:*

- mitigation of some (not all) SMT side channels;
- limiting SMT interference to improve determinism and/or performance.

*`SCHED_CORE` is default disabled. When it is enabled and unused, which is the likely usage by Linux distributions, there should be no measurable impact on performance.*

## 1.21 Extensible Scheduling Class seit 6.12

`CONFIG_SCHED_CLASS_EXT` [=y] [Y]

Diese Option aktiviert eine neue Scheduler-Klasse `sched_ext` (SCX), die es ermöglicht, dass Scheduling-Richtlinien als BPF-Programme implementiert werden können, um Folgendes zu erreichen:

- Einfaches Experimentieren und Erforschen: Ermöglicht die schnelle Iteration neuer Zeitplanungsrichtlinien.

556 - Anpassungsfähigkeit: Erstellung von anwendungsspezifischen Schedulingern, die Richtlinien implementieren, die für allgemeine Scheduler nicht anwendbar sind.

558 - Schnelle Scheduler-Implementierungen: Unterbrechungsfreie Auslagerung von Planungsrichtlinien in Produktionsumgebungen.

560 `sched_ext` nutzt die BPF-Funktion `struct_ops`, um eine Struktur zu definieren, die Funktionsaufrufe und Flags an BPF-Programme exportiert, die Zeitplanungsrichtlinien implementieren möchten.

562 Die `struct_ops`-Struktur, die von `sched_ext` exportierte Struktur heißt `struct sched_ext_ops` und ist konzeptionell ähnlich wie `struct sched_class`.

564 Für weitere Informationen:

565 Dokumentation/scheduler/sched-ext.rst

566 <https://github.com/sched-ext/scx>

567 *This option enables a new scheduler class `sched_ext` (SCX), which allows scheduling policies to be implemented as BPF programs to achieve the following:*

569 - *Ease of experimentation and exploration: Enabling rapid iteration of new scheduling policies.*

570 - *Customization: Building application-specific schedulers which implement policies that are not applicable to general-purpose schedulers.*

572 - *Rapid scheduler deployments: Non-disruptive swap outs of scheduling policies in production environments.*

573 `sched_ext` leverages BPF `struct_ops` feature to define a structure which exports function callbacks and flags to BPF programs that wish to implement scheduling policies. The `struct_ops` structure exported by `sched_ext` is `struct sched_ext_ops`, and is conceptually similar to `struct sched_class`.

576 For more information: [Documentation/scheduler/sched-ext.rst](https://github.com/sched-ext/scx) <https://github.com/sched-ext/scx>

## 577 1.22 CPU/Task time and stats accounting →

### 578 1.22.1 Cputime accounting (Full dynticks CPU time accounting) →

#### 579 1.22.1.1 Simple tick based cputime accounting

580 `CONFIG_TICK_CPU_ACCOUNTING [=n] [N]`

581 Dies ist die grundlegende tick-basierte Rechenzeitabrechnung, die Statistiken über die Benutzer-, System- und Leerlaufzeit auf einer Granularität von wenigen Augenblicken führt.

583 Wenn Sie unsicher sind, sagen Sie Y.

584 *This is the basic tick based cputime accounting that maintains statistics about user, system and idle time spent on per jiffies granularity.*

586 *If unsure, say Y.*

587 Diese Auswahl erscheint nur dann, wenn `CONFIG_S390` sowie `CONFIG_NO_HZ_FULL` abgewählt sind (=N).

#### 588 1.22.1.2 Full dynticks CPU time accounting

589 `CONFIG_VIRT_CPU_ACCOUNTING_GEN [=y] [Y]`

590 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.

594 Im Moment ist dies nur sinnvoll, wenn Sie an der Entwicklung des vollständigen Dynticks-Subsystems arbeiten.

596 *Select this option to enable task and CPU time accounting on full dynticks systems. This accounting is implemented by watching every kernel-user boundaries using the context tracking subsystem. The accounting is thus performed at the expense of some significant overhead.*

599 *For now this is only useful if you are working on the full dynticks subsystem development. If unsure, say N.*

600 Wir können nur dies mit unseren Voreinstellungen auswählen.

### 601 1.22.2 Fine granularity task level IRQ time accounting

602 `CONFIG_IRQ_TIME_ACCOUNTING [=y] [N]`

603 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.

606 Im Zweifelsfall sagen Sie hier N für Nein.

607 *Select this option to enable fine granularity task irq time accounting. This is done by reading a timestamp on each*  
608 *transitions between softirq and hardirq state, so there can be a small performance impact.*

609 *If in doubt, say N here.*

610 Um etwas mehr Performance zu gewinnen, setzen wir dies auf N für Nein.

### 611 1.22.3 BSD Process Accounting

612 CONFIG\_BSD\_PROCESS\_ACCT [=y] [Y]

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

620 *If you say Y here, a user level program will be able to instruct the kernel (via a special system call) to write process*  
621 *accounting information to a file: whenever a process exits, information about that process will be appended to the*  
622 *file by the kernel. The information includes things such as creation time, owning user, command name, memory*  
623 *usage, controlling terminal etc. (the complete list is in the struct acct in <file:include/linux/acct.h>). It is up to*  
624 *the user level program to do useful things with this information. This is generally a good idea, so say Y.*

#### 625 1.22.3.1 BSD Process Accounting version 3 file format

626 CONFIG\_BSD\_PROCESS\_ACCT\_V3 [=y] [Y]

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

632 *If you say Y here, the process accounting information is written in a new file format that also logs the process*  
633 *IDs of each process and its parent. Note that this file format is incompatible with previous v0/v1/v2 file formats,*  
634 *so you will need updated tools for processing it. A preliminary version of these tools is available at [http:](http://www.gnu.org/software/acct/)*  
635 *//www.gnu.org/software/acct/.*

### 636 1.22.4 Export task/process statistics through netlink

637 CONFIG\_TASKSTATS [=y] [Y]

638 Export ausgewählter Statistiken für Aufgaben/Prozesse über die generische Netlink-Schnittstelle. Im  
639 Gegensatz zur BSD-Prozessabrechnung sind die Statistiken während der Lebensdauer von Aufgaben/Pro-  
640 zessen als Antwort auf Befehle verfügbar. Wie BSD-Accounting werden sie beim Beenden von Tasks in  
641 den Benutzerbereich gesendet.

642 Sagen Sie N, wenn Sie unsicher sind.

643 *Export selected statistics for tasks/processes through the generic netlink interface. Unlike BSD process accounting,*  
644 *the statistics are available during the lifetime of tasks/processes as responses to commands. Like BSD accounting,*  
645 *they are sent to user space on task exit.*

646 *Say N if unsure.*

#### 647 1.22.4.1 Enable per-task delay accounting

648 CONFIG\_TASK\_DELAY\_ACCT [=y] [Y]

649 Sammeln Sie Informationen über die Zeit, die eine Task für das Warten auf Systemressourcen wie CPU,  
650 synchrone Block-E/A-Abwicklung und Auslagerung von Seiten aufwendet. Solche Statistiken können bei  
651 der Festlegung der Prioritäten eines Tasks im Verhältnis zu anderen Tasks für CPU-, IO-, RSS-Limits  
652 usw. helfen.

653 Sagen Sie N, wenn Sie unsicher sind.

654 *Collect information on time spent by a task waiting for system resources like cpu, synchronous block I/O completion*  
655 *and swapping in pages. Such statistics can help in setting a task's priorities relative to other tasks for cpu, io, rss*

656 *limits etc.*  
657 *Say N if unsure.*

#### 658 **1.22.4.2 Enable extended accounting over taskstats**

659 **CONFIG\_TASK\_XACCT [=y] [Y]**

660 Sammeln von erweiterten Task-Accounting-Daten und Senden der Daten an das Userland zur Verarbeitung  
661 über die Taskstats-Schnittstelle.

662 Sagen Sie N, wenn Sie unsicher sind.

663 *Collect extended task accounting data and send the data to userland for processing over the taskstats interface.*

664 *Say N if unsure.*

#### 665 **1.22.4.2.1 Enable per-task storage I/O accounting**

666 **CONFIG\_TASK\_IO\_ACCOUNTING [=y] [Y]**

667 Sammeln von Informationen über die Anzahl der Bytes an Speicher-E/A, die dieser Task verursacht hat.

668 Sagen Sie N, wenn Sie unsicher sind.

669 *Collect information on the number of bytes of storage I/O which this task has caused.*

670 *Say N if unsure.*

#### 671 **1.22.5 Pressure stall information tracking**

672 **CONFIG\_PSI [=y] [Y]**

673 Sammeln Sie Metriken, die anzeigen, wie überlastet die CPU-, Speicher- und IO-Kapazität im System  
674 sind.

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

678 In Kernen mit `cgroup`-Unterstützung verfügen `cgroups` (nur `cgroup2`) über `cpu.pressure`-,  
679 `memory.pressure`- und `io.pressure`-Dateien, die nur die Druckstaus für die gruppierten Aufgaben zusam-  
680 menfassen.

681 Weitere Einzelheiten finden Sie unter `Documentation/accounting/psi.rst`.

682 Sagen Sie N, wenn Sie unsicher sind.

683 *Collect metrics that indicate how overcommitted the CPU, memory, and IO capacity are in the system.*

684 *If you say Y here, the kernel will create /proc/pressure/ with the pressure statistics files cpu, memory, and io.*

685 *These will indicate the share of walltime in which some or all tasks in the system are delayed due to contention of*  
686 *the respective resource.*

687 *In kernels with cgroup support, cgroups (cgroup2 only) will have cpu.pressure, memory.pressure, and io.pressure*  
688 *files, which aggregate pressure stalls for the grouped tasks only.*

689 *For more details see Documentation/accounting/psi.rst.*

690 *Say N if unsure.*

#### 691 **1.22.5.1 Require boot parameter to enable pressure stall information tracking**

692 **CONFIG\_PSI\_DEFAULT\_DISABLED [=n] [N]**

693 Wenn diese Option gesetzt ist, ist die Verfolgung von Druckstauinformationen standardmäßig deaktiviert,  
694 kann aber durch die Übergabe von `psi=1` auf der Kernel-Befehlszeile beim Booten aktiviert werden.

695 Diese Funktion fügt dem Task-Wakeup- und Sleep-Pfad des Schedulers etwas Code hinzu. Der Overhead  
696 ist zu gering, um gängige planungsintensive Arbeitslasten in der Praxis zu beeinträchtigen (z. B. Webserver,  
697 Memcache), aber es zeigt sich in künstlichen Scheduler-Stresstests, wie z. B. Hackbench.

698 Wenn Sie paranoid sind und nicht sicher, wofür der Kernel verwendet wird, sagen Sie Y für Ja.

699 Sagen Sie N, wenn Sie unsicher sind.

700 *If set, pressure stall information tracking will be disabled per default but can be enabled through passing psi=1 on*  
701 *the kernel commandline during boot.*

702 *This feature adds some code to the task wakeup and sleep paths of the scheduler. The overhead is too low to affect*  
703 *common scheduling-intense workloads in practice (such as web servers, memcache), but it does show up in artificial*  
704 *scheduler stress tests, such as hackbench.*

705 *If you are paranoid and not sure what the kernel will be used for, say Y.*

706 *Say N if unsure.*

## 1.23 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-CPU's verlagert. Dies wird durch den Boot-Parameter `isolcpus=` gesteuert.

Sagen Sie Y für ja, wenn Sie unsicher sind.

*Make sure that CPUs running critical tasks are not disturbed by any source of “noise” such as unbound workqueues, timers, kthreads ... Unbound jobs get offloaded to housekeeping CPUs. This is driven by the “isolcpus=” boot parameter.*

## 1.24 RCU Subsystem →

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

### 1.24.1 Make expert-level adjustments to RCU configuration

CONFIG\_RCU\_EXPERT [=y] [N]

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.

*This option needs to be enabled if you wish to make expert-level adjustments to RCU configuration. By default, no such adjustments can be made, which has the often-beneficial side-effect of preventing “make oldconfig” from asking you all sorts of detailed questions about how you would like numerous obscure RCU options to be set up.*

*Say Y if you need to make expert-level adjustments to RCU.*

*Say N if you are unsure.*

Wie bei Debian Bookworm setzen wir dies auf ein N für Nein.

### 1.24.2 Force selection of NEED\_SRCU\_NMI\_SAFE seit 6.2

CONFIG\_FORCE\_NEED\_SRCU\_NMI\_SAFE [=n] [N]

Diese Option erzwingt die Auswahl der NEED\_SRCU\_NMI\_SAFE Kconfig-Option und ermöglicht das Testen von `srcu_read_lock_nmisafe()` und `srcu_read_unlock_nmisafe()` auf Architekturen (wie x86), die die ARCH\_HAS\_NMI\_SAFE\_THIS\_CPU\_OPS Kconfig-Option auswählen.

*This option forces selection of the NEED\_SRCU\_NMI\_SAFE Kconfig option, allowing testing of srcu\_read\_lock\_nmisafe() and srcu\_read\_unlock\_nmisafe() on architectures (like x86) that select the ARCH\_HAS\_NMI\_SAFE\_THIS\_CPU\_OPS Kconfig option.*

### 1.24.3 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.

*This option force-enables a task-based RCU implementation that uses only voluntary context switch (not preemption!), idle, and user-mode execution as quiescent states. Not for manual selection in most cases.*

### 1.24.4 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-CPU's, auch auf den Idle-CPU's, also mit Vorsicht verwenden. In den meisten Fällen nicht für die manuelle Auswahl geeignet.

*This option force-enables a task-based RCU implementation that uses only context switch (including preemption)*

755 and user-mode execution as quiescent states. It forces IPIs and context switches on all online CPUs, including idle  
756 ones, so use with caution. Not for manual selection in most cases.

#### 757 1.24.5 Force selection of Tasks Trace RCU

758 CONFIG\_FORCE\_TASKS\_TRACE\_RCU [=n] [N]

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

763 *This option enables a task-based RCU implementation that uses explicit `rcu_read_lock_trace()` read-side markers,*  
764 *and allows these readers to appear in the idle loop as well as on the CPU hotplug code paths. It can force IPIs on*  
765 *online CPUs, including idle ones, so use with caution. Not for manual selection in most cases.*

#### 766 1.24.6 Tree-based hierarchical RCU fanout value

767 CONFIG\_RCU\_FANOUT [=64] [64]

768 Diese Option steuert den Fanout von hierarchischen Implementierungen von RCU, so dass RCU auf  
769 Maschinen mit einer großen Anzahl von CPUs effizient arbeiten kann. Dieser Wert muss mindestens die  
770 vierte Wurzel von `NR_CPUS` sein, wodurch `NR_CPUS` wahnsinnig groß werden kann. Der Standardwert  
771 von `RCU_FANOUT` sollte für Produktionssysteme verwendet werden, aber wenn Sie die RCU-Implementierung  
772 selbst einem Stresstest unterziehen, ermöglichen kleine `RCU_FANOUT`-Werte das Testen von  
773 Codepfaden für große Systeme auf kleinen (kleineren) Systemen.

774 Wählen Sie eine bestimmte Zahl, wenn Sie RCU selbst testen. Nehmen Sie den Standardwert, wenn Sie  
775 unsicher sind.

776 Symbol: `RCU_FANOUT` [=64]

777 Type : integer (Ganzzahl)

778 Bereich (range) : [2 64]

779 *This option controls the fanout of hierarchical implementations of RCU, allowing RCU to work efficiently on*  
780 *machines with large numbers of CPUs. This value must be at least the fourth root of `NR_CPUS`, which allows*  
781 *`NR_CPUS` to be insanely large. The default value of `RCU_FANOUT` should be used for production systems, but if*  
782 *you are stress-testing the RCU implementation itself, small `RCU_FANOUT` values allow you to test large-system*  
783 *code paths on small(er) systems.*

784 *Select a specific number if testing RCU itself. Take the default if unsure.*

#### 785 1.24.7 Tree-based hierarchical RCU leaf-level fanout value

786 CONFIG\_RCU\_FANOUT\_LEAF [=16] [16]

787 Diese Option steuert das Fanout auf Blattebene bei hierarchischen Implementierungen von RCU und  
788 ermöglicht es, Cache-Misses gegen Sperrkonflikte abzuwägen. Systeme, die ihre Scheduling-Clock-Interrupts  
789 aus Gründen der Energieeffizienz synchronisieren, werden die Standardeinstellung bevorzugen, da der  
790 kleinere Leaf-Level-Fanout die Lock-Contention-Level akzeptabel niedrig hält. Sehr große Systeme (Hun-  
791 derte oder Tausende von CPUs) werden stattdessen diesen Wert auf den maximal möglichen Wert setzen  
792 wollen, um die Anzahl der Cache-Misses zu reduzieren, die während der Initialisierung der RCU-Grace-  
793 Periode auftreten. Diese Systeme neigen dazu, CPU-gebunden zu laufen, und werden daher nicht von  
794 synchronisierten Interrupts unterstützt, und neigen daher dazu, sie zu verzerren, was den Sperrkonflikt  
795 so weit reduziert, dass große Fanouts auf Blattebene gut funktionieren. Das heißt, wenn Sie den Fanout  
796 auf Blattebene auf eine große Zahl setzen, wird dies wahrscheinlich zu problematischen Sperrkonflikten  
797 auf den `rcu_node`-Strukturen auf Blattebene führen, es sei denn, Sie booten mit dem Kernelparameter  
798 `skew_tick`.

799 Wählen Sie eine bestimmte Zahl, wenn Sie die RCU selbst testen.

800 Wählen Sie den maximal zulässigen Wert für große Systeme, aber bedenken Sie, dass Sie möglicherweise  
801 auch den Kernel-Boot-Parameter `skew_tick` setzen müssen, um Konflikte bei den Sperren den `rcu_node`-  
802 Strukturen zu vermeiden. Nehmen Sie den Standardwert, wenn Sie unsicher sind.

803 Symbol: `RCU_FANOUT_LEAF` [=64]

804 Type : integer (Ganzzahl)

805 Bereich (range) : [2 64]

806 *This option controls the leaf-level fanout of hierarchical implementations of RCU, and allows trading off cache*

misses against lock contention. Systems that synchronize their scheduling-clock interrupts for energy-efficiency reasons will want the default because the smaller leaf-level fanout keeps lock contention levels acceptably low. Very large systems (hundreds or thousands of CPUs) will instead want to set this value to the maximum value possible in order to reduce the number of cache misses incurred during RCU's grace-period initialization. These systems tend to run CPU-bound, and thus are not helped by synchronized interrupts, and thus tend to skew them, which reduces lock contention enough that large leaf-level fanouts work well. That said, setting leaf-level fanout to a large number will likely cause problematic lock contention on the leaf-level `rcu_node` structures unless you boot with the `skew_tick` kernel parameter.

Select a specific number if testing RCU itself.

Select the maximum permissible value for large systems, but please understand that you may also need to set the `skew_tick` kernel boot parameter to avoid contention on the `rcu_node` structure's locks.

Take the default if unsure.

#### 1.24.8 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.

*This option boosts the priority of preempted RCU readers that block the current preemptible RCU grace period for too long. This option also prevents heavy loads from blocking RCU callback invocation.*

*Say Y here if you are working with real-time apps or heavy loads*

*Say N here if you are unsure.*

Diese Option wird nur dann angezeigt, wenn Fully Preemptible Kernel (Real Time) mit `CONFIG_PREEMPT_RT` eingeschaltet ist.

##### 1.24.8.1 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]

*This option specifies the time to wait after the beginning of a given grace period before priority-boosting preempted RCU readers blocking that grace period. Note that any RCU reader blocking an expedited RCU grace period is boosted immediately.*

*Accept the default if unsure.*

##### 1.24.8.2 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`-Kernen 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.

*Use this option to further reduce the latencies of expedited grace periods at the expense of being more disruptive.*

*This option is disabled by default on `PREEMPT_RT=y` kernels which disable expedited grace periods after boot by unconditionally setting `rcupdate.rcu_normal_after_boot=1`.*

*Accept the default if unsure.*

#### 1.24.9 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.

*Use this option to reduce OS jitter for aggressive HPC or real-time workloads. It can also be used to offload RCU callback invocation to energy-efficient CPUs in battery-powered asymmetric multiprocessors. The price of this reduced jitter is that the overhead of `call_rcu()` increases and that some workloads will incur significant increases in context-switch rates.*

*This option offloads callback invocation from the set of CPUs specified at boot time by the `rcu_nocbs` parameter. For each such CPU, a kthread („`rcuox/N`“) will be created to invoke callbacks, where the „`N`“ is the CPU being offloaded, and where the „`x`“ is „`p`“ for RCU-preempt (PREEMPTION kernels) and „`s`“ for RCU-sched (PREEMPTION kernels). This option also creates another kthread for each  $\sqrt{\text{nr\_cpu\_ids}}$  CPUs („`rcuog/N`“, where `N` is the first CPU in that group to come online), which handles grace periods for its group. Nothing prevents these kthreads from running on the specified CPUs, but (1) the kthreads may be preempted between each callback, and (2) affinity or cgroups can be used to force the kthreads to run on whatever set of CPUs is desired.*

*The  $\sqrt{\text{nr\_cpu\_ids}}$  grouping may be overridden using the `rcutree.rcu_nocb_gp_stride` kernel boot parameter. This can be especially helpful for smaller numbers of CPUs, where  $\sqrt{\text{nr\_cpu\_ids}}$  can be a bit of a blunt instrument.*

*Say Y here if you need reduced OS jitter, despite added overhead. Say N here if you are unsure.*

#### 1.24.9.1 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.

*Use this option to offload callback processing from all CPUs by default, in the absence of the `rcu_nocbs` or `nohz_full` boot parameter. This also avoids the need to use any boot parameters to achieve the effect of offloading all CPUs on boot.*

*Say Y here if you want offload all CPUs by default on boot. Say N here if you are unsure.*

#### 1.24.9.2 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.

*Use this option to invoke offloaded callbacks as `SCHED_FIFO` to avoid starvation by heavy `SCHED_OTHER` background load. Of course, running as `SCHED_FIFO` during callback floods will cause the `rcuo[ps]` kthreads to monopolize the CPU for hundreds of milliseconds or more. Therefore, when enabling this option, it is your responsibility to ensure that latency-sensitive tasks either run with higher priority or run on some other CPU.*

*Say Y here if you want to set RT priority for offloading kthreads. Say N here if you are building a `!PREEMPT_RT`*

914 *kernel and are unsure.*

#### 915 **1.24.10 Tasks Trace RCU readers use memory barriers in user and idle**

916 `CONFIG_TASKS_TRACE_RCU_READ_MB [=n] [N]`

917 Verwenden Sie diese Option, um die Anzahl der IPIs (inter-processor interrupts), die an CPUs gesendet  
918 werden, die im Benutzerraum ausgeführt werden oder sich im Leerlauf befinden, während Tasks RCU-Til-  
919 gungsfristen verfolgen, weiter zu reduzieren. Da eine vernünftige Einstellung des Kernel-Boot-Parameters  
920 `rcupdate.rcu_task_ipi_delay` solche IPIs für viele Arbeitslasten eliminiert, ist die richtige Einstellung  
921 dieser Kconfig-Option vor allem für aggressive Echtzeitinstallationen und für batteriebetriebene Geräte  
922 wichtig, daher die oben gewählte Standardeinstellung.

923 Sagen Sie hier Y, wenn Sie IPIs hassen.

924 Sagen Sie hier N, wenn Sie leseseitige Speicherbarrieren hassen.

925 Nehmen Sie die Standardeinstellung, wenn Sie unsicher sind.

926 *Use this option to further reduce the number of IPIs sent to CPUs executing in userspace or idle during tasks trace*  
927 *RCU grace periods. Given that a reasonable setting of the `rcupdate.rcu_task_ipi_delay` kernel boot parameter*  
928 *eliminates such IPIs for many workloads, proper setting of this Kconfig option is important mostly for aggressive*  
929 *real-time installations and for battery-powered devices, hence the default chosen above.*

930 *Say Y here if you hate IPIs. Say N here if you hate read-side memory barriers. Take the default if you are unsure.*

#### 931 **1.24.11 RCU callback lazy invocation functionality**

932 `CONFIG_RCU_LAZY [=y] [Y]`

933 Um Strom zu sparen, sollten Sie RCU-Rückrufe stapeln und nach einer Verzögerung, einem Speicher-  
934 druck oder einer zu großen Rückrufliste flushen. Dazu muss `rcu_nocbs=all` gesetzt sein. Verwenden Sie  
935 `rcutree.enable_rcu_lazy=0`, um es beim Booten auszuschalten.

936 *To save power, batch RCU callbacks and flush after delay, memory pressure, or callback list growing too big.*

937 *Requires `rcu_nocbs=all` to be set.*

938 *Use `rcutree.enable_rcu_lazy=0` to turn it off at boot time.*

##### 939 **1.24.11.1 Turn RCU lazy invocation off by default** seit 6.9

940 `CONFIG_RCU_LAZY_DEFAULT_OFF [=n] [N]`

941 Erlaubt die Erstellung des Kernels mit `CONFIG_RCU_LAZY=y`, ist aber standardmäßig deaktiviert. Der  
942 Bootzeit-Parameter `rcutree.enable_rcu_lazy=1` kann verwendet werden, um es wieder einzuschalten.

943 *Allows building the kernel with `CONFIG_RCU_LAZY=y` yet keep it default off.*

944 *Boot time param `rcutree.enable_rcu_lazy=1` can be used to switch it back on.*

#### 945 **1.24.12 RCU callback-batch backup time check**

946 `CONFIG_RCU_DOUBLE_CHECK_CB_TIME [=y] [Y]`

947 Mit dieser Option kann der Modulparameter `rcutree.rcu_resched_ns` in Situationen präziser durch-  
948 gesetzt werden, in denen ein einzelner RCU-Callback Hunderte von Mikrosekunden lang laufen kann,  
949 wodurch die 32-Callback-Batching-Funktion, die zur Amortisierung der Kosten für die feinkörnige, aber  
950 teure `local_clock()`-Funktion verwendet wird, unterlaufen wird.

951 Diese Option rundet `rcutree.rcu_resched_ns` auf den nächsten Jiffy auf und setzt die 32-Callback-  
952 Batching-Funktion außer Kraft, wenn diese Grenze überschritten wird.

953 Sagen Sie hier Y, wenn Sie eine strengere Durchsetzung des Rückrufflimits benötigen.

954 Sagen Sie hier N, wenn Sie unsicher sind.

955 *Use this option to provide more precise enforcement of the `rcutree.rcu_resched_ns` module parameter in situations*  
956 *where a single RCU callback might run for hundreds of microseconds, thus defeating the 32-callback batching used*  
957 *to amortize the cost of the fine-grained but expensive `local_clock()` function.*

958 *This option rounds `rcutree.rcu_resched_ns` up to the next jiffy, and overrides the 32-callback batching if this limit*  
959 *is exceeded.*

960 *Say Y here if you need tighter callback-limit enforcement. Say N here if you are unsure.*

## 1.25 Kernel .config support

CONFIG\_IKCONFIG [=y] [N]

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).

*This option enables the complete Linux kernel “config” file contents to be saved in the kernel. It provides documentation of which kernel options are used in a running kernel or in an on-disk kernel. This information can be extracted from the kernel image file with the script `scripts/extract-ikconfig` and used as input to rebuild the current kernel or to build another kernel. It can also be extracted from a running kernel by reading `/proc/config.gz` if enabled (below).*

Ist nicht unbedingt notwendig, auch in Debian Bookworm ist dies ausgeschaltet.

### 1.25.1 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`.

*This option enables access to the kernel configuration file through `/proc/config.gz`.*

## 1.26 Enable kernel headers through /sys/kernel/kheaders.tar.xz

CONFIG\_IKHEADERS [=m] [N]

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.

*This option enables access to the in-kernel headers that are generated during the build process. These can be used to build eBPF tracing programs, or similar programs. If you build the headers as a module, a module called `kheaders.ko` is built which can be loaded on-demand to get access to headers.*

Ist auch als Modul nicht unbedingt notwendig, wie auch in Debian Bookworm wird dies ausgeschaltet.

## 1.27 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]

*Select the minimal kernel log buffer size as a power of 2.*

*The final size is affected by `LOG_CPU_MAX_BUF_SHIFT` config parameter, see below. Any higher size also might be forced by “`log_buf_len`” boot parameter.*

*Examples:*

17 => 128 KB

16 => 64 KB

1011 15 => 32 KB  
 1012 14 => 16 KB  
 1013 13 => 8 KB  
 1014 12 => 4 KB

## 1015 1.28 CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)

1016 CONFIG\_LOG\_BUF\_SHIFT [=12] [12]

1017 Diese Option ermöglicht es, die Standardgröße des Ringpuffers entsprechend der Anzahl der CPUs zu  
 1018 erhöhen. Der Wert definiert den Beitrag jeder CPU als eine Potenz von 2. Der beanspruchte Speicherplatz  
 1019 beträgt in der Regel nur wenige Zeilen, kann aber viel mehr sein, wenn Probleme gemeldet werden,  
 1020 z. B. bei Rückverfolgungen. Die erhöhte Größe bedeutet, dass ein neuer Puffer zugewiesen werden muss  
 1021 und der ursprüngliche statische Puffer ungenutzt ist. Dies ist nur auf Systemen mit mehreren CPUs  
 1022 sinnvoll. Daher wird dieser Wert nur verwendet, wenn die Summe der Beiträge größer ist als die Hälfte des  
 1023 Standard-Kernel-Ringpuffers, wie durch LOG\_BUF\_SHIFT definiert. Die Standardwerte sind so eingestellt,  
 1024 dass mehr als 16 CPUs erforderlich sind, um die Zuweisung auszulösen. Diese Option wird auch ignoriert,  
 1025 wenn der Kernelparameter log\_buf\_len verwendet wird, da er eine exakte (Zweierpotenz) Größe des  
 1026 Ringpuffers erzwingt. Die Anzahl der möglichen CPUs wird für diese Berechnung verwendet, wobei  
 1027 Hotplugging ignoriert wird, so dass die Berechnung für das Worst-Case-Szenario optimal ist und gleichzeitig  
 1028 ein einfacher Algorithmus ab dem Hochfahren verwendet werden kann. Beispiele für Verschiebungswerte  
 1029 und ihre Bedeutung:

1030 17 => 128 kB für jede CPU  
 1031 16 => 64 kB für jede CPU  
 1032 15 => 32 kB für jede CPU  
 1033 14 => 16 kB für jede CPU  
 1034 13 => 8 kB für jede CPU  
 1035 12 => 4 kB für jede CPU

1036 Symbol: LOG\_CPU\_MAX\_BUF\_SHIFT

1037 Type: Integer (Ganzzahl)

1038 Bereich (range): [0 21]

1039 *This option allows to increase the default ring buffer size according to the number of CPUs. The value defines the*  
 1040 *contribution of each CPU as a power of 2. The used space is typically only few lines however it might be much*  
 1041 *more when problems are reported, e.g. backtraces.*

1042 *The increased size means that a new buffer has to be allocated and the original static one is unused. It makes sense*  
 1043 *only on systems with more CPUs. Therefore this value is used only when the sum of contributions is greater than*  
 1044 *the half of the default kernel ring buffer as defined by LOG\_BUF\_SHIFT. The default values are set so that more*  
 1045 *than 16 CPUs are needed to trigger the allocation.*

1046 *Also this option is ignored when "log\_buf\_len" kernel parameter is used as it forces an exact (power of two) size of*  
 1047 *the ring buffer.*

1048 *The number of possible CPUs is used for this computation ignoring hotplugging making the computation optimal*  
 1049 *for the worst case scenario while allowing a simple algorithm to be used from bootup.*

1050 *Examples shift values and their meaning:*

1051 17 => 128 KB for each CPU  
 1052 16 => 64 KB for each CPU  
 1053 15 => 32 KB for each CPU  
 1054 14 => 16 KB for each CPU  
 1055 13 => 8 KB for each CPU  
 1056 12 => 4 KB for each CPU

## 1057 1.29 Printk indexing debugfs interface)

1058 CONFIG\_PRINTK\_INDEX [=y] [N]

1059 Unterstützung für die Indizierung aller zur Kompilierzeit bekannten printk-Formate unter  
 1060 <debugfs>/printk/index/<module> hinzufügen. Dies kann als Teil der Wartung von Daemonen, die  
 1061 /dev/kmsg überwachen, verwendet werden, da es die Überprüfung der in einem Kernel vorhandenen  
 1062 printk-Formate erlaubt, was die Erkennung von Fällen ermöglicht, in denen überwachte printks geändert  
 1063 oder nicht mehr vorhanden sind.

1064 Es gibt keine zusätzlichen Laufzeitkosten für printk, wenn dies aktiviert ist.

Add support for indexing of all printk formats known at compile time at `<debugfs>/printk/index/<module>`.  
This can be used as part of maintaining daemons which monitor `/dev/kmsg`, as it permits auditing the printk formats present in a kernel, allowing detection of cases where monitored printks are changed or no longer present.  
There is no additional runtime cost to printk with this enabled.

Wie bei Debian Bookworm wird diese Indizierung ausgeschaltet.

## 1.30 Scheduler features →

(Scheduler-Funktionen)

### 1.30.1 Enable utilization clamping for RT/FAIR tasks

`CONFIG_UCLAMP_TASK` [=y] [N]

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.

*This feature enables the scheduler to track the clamped utilization of each CPU based on RUNNABLE tasks scheduled on that CPU.*

*With this option, the user can specify the min and max CPU utilization allowed for RUNNABLE tasks. The max utilization defines the maximum frequency a task should use while the min utilization defines the minimum frequency it should use.*

*Both min and max utilization clamp values are hints to the scheduler, aiming at improving its frequency selection policy, but they do not enforce or grant any specific bandwidth for tasks.*

*If in doubt, say N.*

Wie bei Debian Bookworm und WSL2 wird dies ausgeschaltet.

#### 1.30.1.1 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.

*Defines the number of clamp buckets to use.*

*The range of each bucket will be `SCHED_CAPACITY_SCALE/UCLAMP_BUCKETS_COUNT`. The higher the number of clamp buckets the finer their granularity and the higher the precision of clamping aggregation and tracking at run-time.*

*For example, with the minimum configuration value we will have 5 clamp buckets tracking 20% utilization each. A 25% boosted tasks will be refcounted in the [20..39]% bucket and will set the bucket clamp effective value to 25%. If a second 30% boosted task should be co-scheduled on the same CPU, that task will be refcounted in the same bucket of the first task and it will boost the bucket clamp effective value to 30%. The clamp effective value of a*

1117 *bucket is reset to its nominal value (20% in the example above) when there are no more tasks refcounted in that*  
 1118 *bucket.*  
 1119 *An additional boost/capping margin can be added to some tasks. In the example above the 25% task will be boosted*  
 1120 *to 30% until it exits the CPU. If that should be considered not acceptable on certain systems, it's always possible*  
 1121 *to reduce the margin by increasing the number of clamp buckets to trade off used memory for run-time tracking*  
 1122 *precision.*  
 1123 *If in doubt, use the default value.*

### 1124 **1.31 Memory placement aware NUMA scheduler**

1125 **CONFIG\_NUMA\_BALANCING [=y] [Y]**  
 1126 Diese Option bietet Unterstützung für die automatische NUMA-kompatible Speicher-/Task-Platzierung.  
 1127 Der Mechanismus ist recht primitiv und basiert darauf, dass Speicher migriert wird, wenn er Referenzen  
 1128 auf den Knoten hat, auf dem die Aufgabe läuft. Dieses System ist auf UMA-Systemen inaktiv.  
 1129 *This option adds support for automatic NUMA aware memory/task placement. The mechanism is quite primitive*  
 1130 *and is based on migrating memory when it has references to the node the task is running on.*  
 1131 *This system will be inactive on UMA systems.*

#### 1132 **1.31.1 Automatically enable NUMA aware memory/task placement**

1133 **CONFIG\_NUMA\_BALANCING\_DEFAULT\_ENABLED [=y] [Y]**  
 1134 Wenn diese Option gesetzt ist, wird der automatische NUMA-Ausgleich aktiviert, wenn das System auf  
 1135 einem NUMA-Rechner läuft.  
 1136 *If set, automatic NUMA balancing will be enabled if running on a NUMA machine.*

### 1137 **1.32 Control Group support →**

1138 **CONFIG\_CGROUPS [=y] [Y]**  
 1139 (Unterstützung der Kontrollgruppe)  
 1140 Diese Option bietet Unterstützung für die Gruppierung von Prozessgruppen zur Verwendung mit Prozess-  
 1141 kontrollsubsystemen wie Cpusets, CFS, Speicherkontrolle oder Geräteisolierung.  
 1142 Siehe

- 1143 • [Dokumentation/scheduler/sched-design-CFS.rst](#) (CFS)
- 1144 • [Documentation/admin-guide/cgroup-v1/](#) (Funktionen für Gruppierung, Isolierung und Ressourcen-
- 1145 [kontrolle](#))

1146 Sagen Sie N, wenn Sie unsicher sind.  
 1147 *This option adds support for grouping sets of processes together, for use with process control subsystems such as*  
 1148 *Cpusets, CFS, memory controls or device isolation.*  
 1149 *See*  
 1150 *– Documentation/scheduler/sched-design-CFS.rst (CFS)*  
 1151 *– Documentation/admin-guide/cgroup-v1/ (features for grouping, isolation and resource control)*  
 1152 *Say N if unsure.*

#### 1153 **1.32.1 Favor dynamic modification latency reduction by default**

1154 **CONFIG\_CGROUP\_FAVOR\_DYNMODS [=n] [N]**  
 1155 Diese Option aktiviert standardmäßig die Einhängeoption `favordynmods`, die die Latenzzeiten dynamischer  
 1156 C-Gruppen-Änderungen wie Task-Migrationen und Controller-Ein-/Ausschaltungen auf Kosten von  
 1157 Hot-Path-Operationen wie Forks und Exits verteuert.  
 1158 Sagen Sie N, wenn Sie unsicher sind.  
 1159 *This option enables the “favordynmods” mount option by default which reduces the latencies of dynamic cgroup*  
 1160 *modifications such as task migrations and controller on/off at the cost of making hot path operations such as forks*  
 1161 *and exits more expensive.*  
 1162 *Say N if unsure.*

### 1.32.2 Memory controller

CONFIG\_MEMCG [=y] [Y]

Ermöglicht die Kontrolle über den Speicherbedarf von Tasks in einer cgroup.

*Provides control over the memory footprint of tasks in a cgroup.*

#### 1.32.2.1 Legacy cgroup v1 memory controller seit 6.11

CONFIG\_MEMCG\_V1 [=n] [N]

Legacy cgroup v1 memory controller, der durch die cgroup v2 Implementierung veraltet ist. Der v1 ist für ältere Anwendungen gedacht, die noch nicht auf die neue cgroup v2-Schnittstelle umgestellt wurden. Wenn Sie keine solche Anwendung haben, ist es völlig in Ordnung, wenn Sie diese Option deaktiviert lassen.

Bitte beachten Sie, dass der Funktionsumfang des Legacy-Speicher-Controllers aufgrund der Abkündigung wahrscheinlich schrumpfen wird. Von neuen Implementierungen mit v1-Controller wird dringend abgeraten.

*Legacy cgroup v1 memory controller which has been deprecated by cgroup v2 implementation. The v1 is there for legacy applications which haven't migrated to the new cgroup v2 interface yet. If you do not have any such application then you are completely fine leaving this option disabled.*

*Please note that feature set of the legacy memory controller is likely going to shrink due to deprecatoion process.*

*New deployments with v1 controller are highly discouraged.*

### 1.32.3 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.

*Generic block IO controller cgroup interface. This is the common cgroup interface which should be used by various IO controlling policies.*

*Currently, CFQ IO scheduler uses it to recognize task groups and control disk bandwidth allocation (proportional time slice allocation) to such task groups. It is also used by bio throttling logic in block layer to implement upper limit in IO rates on a device.*

*This option only enables generic Block IO controller infrastructure. One needs to also enable actual IO controlling logic/policy.*

*For enabling proportional weight division of disk bandwidth in CFQ, set CONFIG\_BFQ\_GROUP\_IOSCHED=y; for enabling throttling policy, set CONFIG\_BLK\_DEV\_THROTTLING=y.*

*See Documentation/admin-guide/cgroup-v1/blkio-controller.rst for more information.*

### 1.32.4 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.

*This feature lets CPU scheduler recognize task groups and control CPU bandwidth allocation to such task groups. It uses cgroups to group tasks.*

#### 1.32.4.1 Group scheduling for SCHED\_OTHER

CONFIG\_FAIR\_GROUP\_SCHED [=y] [Y]

*Für diese Option gibt es keine Hilfe.*

##### 1.32.4.1.1 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`.

*This option allows users to define CPU bandwidth rates (limits) for tasks running within the fair group scheduler. Groups with no limit set are considered to be unconstrained and will run with no restriction. See Documentation/scheduler/sched-bwc.rst for more information.*

#### 1.32.4.2 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`.

*This feature lets you explicitly allocate real CPU bandwidth to task groups. If enabled, it will also make it impossible to schedule realtime tasks for non-root users until you allocate realtime bandwidth for them. See Documentation/scheduler/sched-rt-group.rst for more information.*

#### 1.32.5 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.

*This feature enables the scheduler to track the clamped utilization of each CPU based on RUNNABLE tasks currently scheduled on that CPU.*

*When this option is enabled, the user can specify a min and max CPU bandwidth which is allowed for each single task in a group. The max bandwidth allows to clamp the maximum frequency a task can use, while the min bandwidth allows to define a minimum frequency a task will always use.*

*When task group based utilization clamping is enabled, an eventually specified task-specific clamp value is constrained by the cgroup specified clamp value. Both minimum and maximum task clamping cannot be bigger than the corresponding clamping defined at task group level.*

*If in doubt, say N.*

Wird nur in der „Control Group support“-Gruppe angezeigt, wenn `CGROUP_SCHED` (CPU Controller →) und `UCLAMP_TASK` (Enable utilization clamping for RT/FAIR tasks) gesetzt sind.

#### 1.32.6 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.

*Provides enforcement of process number limits in the scope of a cgroup. Any attempt to fork more processes than is allowed in the cgroup will fail. PIDs are fundamentally a global resource because it is fairly trivial to reach PID exhaustion before you reach even a conservative kmemcg limit. As a result, it is possible to grind a system to halt without being limited by other cgroup policies. The PIDs controller is designed to stop this from happening.*

*It should be noted that organisational operations (such as attaching to a cgroup hierarchy) will \*not\* be blocked by the PIDs controller, since the PIDs limit only affects a process's ability to fork, not to attach to a cgroup.*

### 1.32.7 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.

*Provides enforcement of RDMA resources defined by IB stack. It is fairly easy for consumers to exhaust RDMA resources, which can result into resource unavailability to other consumers. RDMA controller is designed to stop this from happening. Attaching processes with active RDMA resources to the cgroup hierarchy is allowed even if can cross the hierarchy's limit.*

### 1.32.8 Device memory controller (DMEM)

CONFIG\_CGROUP\_DMEN [=y] [Y]

Der DMEM-Controller ermöglicht es kompatiblen Geräten, die Nutzung des Gerätespeichers auf der Grundlage der cgroup-Hierarchie einzuschränken.

So können Sie zum Beispiel die VRAM-Nutzung für Anwendungen im DRM-Subsystem einschränken.

*The DMEM controller allows compatible devices to restrict device memory usage based on the cgroup hierarchy.*

*As an example, it allows you to restrict VRAM usage for applications in the DRM subsystem.*

### 1.32.9 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.

*Provides a way to freeze and unfreeze all tasks in a cgroup.*

*This option affects the ORIGINAL cgroup interface. The cgroup2 memory controller includes important in-kernel memory consumers per default.*

*If you're using cgroup2, say N.*

### 1.32.10 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.

*Provides a cgroup controller for HugeTLB pages. When you enable this, you can put a per cgroup limit on HugeTLB usage. The limit is enforced during page fault. Since HugeTLB doesn't support page reclaim, enforcing the limit at page fault time implies that, the application will get SIGBUS signal if it tries to access HugeTLB pages beyond its limit. This requires the application to know beforehand how much HugeTLB pages it would require for its use. The control group is tracked in the third page lru pointer. This means that we cannot use the controller with huge page less than 3 pages.*

### 1.32.11 Cpuset controller

CONFIG\_CPUSETS [=y] [Y]

Mit dieser Option können Sie CPUSETs erstellen und verwalten, die es ermöglichen, ein System dynamisch

1315 in Gruppen von CPUs und Speicherknoten zu partitionieren und Aufgaben zuzuweisen, die nur innerhalb  
1316 dieser Gruppen ausgeführt werden. Dies ist vor allem auf großen SMP- oder NUMA-Systemen nützlich.  
1317 Sagen Sie N, wenn Sie unsicher sind.

1318 *This option will let you create and manage CPUSets which allow dynamically partitioning a system into sets of*  
1319 *CPUs and Memory Nodes and assigning tasks to run only within those sets. This is primarily useful on large SMP*  
1320 *or NUMA systems.*

1321 *Say N if unsure.*

#### 1322 **1.32.11.1 Legacy cgroup v1 cpusets controller** seit 6.12

1323 CONFIG\_CPUSSETS\_V1 [=n] [N]

1324 Die Option v1 ist für ältere Anwendungen gedacht, die noch nicht auf die neue cgroup v2-Schnittstelle  
1325 migriert wurden. Wenn Sie keine solche Anwendung haben, ist es völlig in Ordnung, wenn Sie diese Option  
1326 deaktiviert lassen.

1327 Sagen Sie N, wenn Sie unsicher sind.

1328 *Legacy cgroup v1 cpusets controller which has been deprecated by cgroup v2 implementation. The v1 is there*  
1329 *for legacy applications which haven't migrated to the new cgroup v2 interface yet. If you do not have any such*  
1330 *application then you are completely fine leaving this option disabled.*

1331 *Say N if unsure.*

#### 1332 **1.32.11.1.1 Include legacy /proc/<pid>/cpuset file**

1333 CONFIG\_PROC\_PID\_CPUSET [=y] [Y]

1334 *Für diese Option gibt es keine Hilfe.*

1335 *There is no help available for this option.*

#### 1336 **1.32.12 Device controller**

1337 CONFIG\_CGROUP\_DEVICE [=y] [Y]

1338 Bietet einen cgroup-Controller an, der Whitelists für Geräte implementiert, die ein Prozess in der cgroup  
1339 mknod oder öffnen kann.

1340 *Provides a cgroup controller implementing whitelists for devices which a process in the cgroup can mknod or open.*

#### 1341 **1.32.13 Simple CPU accounting controller**

1342 CONFIG\_CGROUP\_CPUACCT [=y] [Y]

1343 (Einfacher CPU-Accounting-Controller)

1344 Bietet einen einfachen Controller für die Überwachung des gesamten CPU-Verbrauchs der Tasks in einer  
1345 cgroup an.

1346 *Provides a simple controller for monitoring the total CPU consumed by the tasks in a cgroup.*

#### 1347 **1.32.14 Perf controller**

1348 CONFIG\_CGROUP\_PERF [=y] [Y]

1349 Diese Option erweitert den Modus perf per-cpu, um die Überwachung auf Threads zu beschränken, die zu  
1350 der angegebenen cgroup gehören und auf der angegebenen CPU laufen. Sie kann auch verwendet werden,  
1351 um die cgroup ID in Stichproben zu haben, so dass sie Leistungsereignisse zwischen cgroups überwachen  
1352 kann.

1353 Sagen Sie N, wenn Sie unsicher sind.

1354 *This option extends the perf per-cpu mode to restrict monitoring to threads which belong to the cgroup specified and*  
1355 *run on the designated cpu. Or this can be used to have cgroup ID in samples so that it can monitor performance*  
1356 *events among cgroups.*

1357 *Say N if unsure.*

#### 1358 **1.32.15 Support for eBPF programs attached to cgroups**

1359 CONFIG\_CGROUP\_BPF [=y] [Y]

1360 Erlaubt das Anhängen von eBPF-Programmen an eine cgroup mit dem bpf(2)-Syscall-Befehl

1361 BPF\_PROG\_ATTACH.

1362 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.

*Allow attaching eBPF programs to a cgroup using the `bpf(2)` syscall command `BPF_PROG_ATTACH`. In which context these programs are accessed depends on the type of attachment. For instance, programs that are attached using `BPF_CGROUP_INET_INGRESS` will be executed on the ingress path of inet sockets.*

### 1.32.16 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`.

*Provides a controller for miscellaneous resources on a host.*

*Miscellaneous scalar resources are the resources on the host system which cannot be abstracted like the other cgroups.*

*This controller tracks and limits the miscellaneous resources used by a process attached to a cgroup hierarchy.*

*For more information, please check `misc cgroup` section in `/Documentation/admin-guide/cgroup-v2.rst`.*

### 1.32.17 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.

*This option enables a simple controller that exports debugging information about the cgroups framework. This controller is for control cgroup debugging only. Its interfaces are not stable.*

*Say N.*

## 1.33 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.

*Provides the way to make tasks work with different objects using the same id. For example same IPC id may refer to different objects or same user id or pid may refer to different tasks when used in different namespaces.*

### 1.33.1 UTS namespace

`CONFIG_UTS_NS [=y] [Y]`

In diesem Namensraum sehen Aufgaben verschiedene Informationen, die mit dem Systemaufruf `uname()` bereitgestellt werden.

*In this namespace tasks see different info provided with the `uname()` system call*

### 1.33.2 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.

*In this namespace boottime and monotonic clocks can be set. The time will keep going with the same pace.*

### 1.33.3 IPC namespace

`CONFIG_IPC_NS [=y] [Y]`

In diesem Namensraum arbeiten Aufgaben mit IPC-IDs (Interprozess-IDs), die jeweils verschiedenen IPC-

1410 Objekten in verschiedenen Namensräumen entsprechen.

1411 *In this namespace tasks work with IPC ids which correspond to different IPC objects in different namespaces.*

#### 1412 **1.33.4 User namespace**

1413 **CONFIG\_USER\_NS [=y] [Y]**

1414 Dies ermöglicht es Containern, d. h. V-Servern, Benutzernamensräume zu verwenden, um verschiedene  
1415 Benutzerinformationen für verschiedene Server bereitzustellen. Wenn Benutzernamensräume im Kernel  
1416 aktiviert sind, wird empfohlen, dass die Option MEMCG ebenfalls aktiviert wird und dass der Benutzerbereich  
1417 die Speicherkontrollgruppen verwendet, um die Speichermenge zu begrenzen, die nicht privilegierte Benutzer  
1418 verwenden können.

1419 *This allows containers, i.e. userservers, to use user namespaces to provide different user info for different servers.*

1420 *When user namespaces are enabled in the kernel it is recommended that the MEMCG option also be enabled and  
1421 that user-space use the memory control groups to limit the amount of memory a memory unprivileged users can use.*

1422 *If unsure, say N.*

1423 **Allow unprivileged users to create namespaces** bis 6.14

1424 **CONFIG\_USERS\_NS\_UNPRIVILEGED [=y] [Y]**

1425 Wenn diese Funktion deaktiviert ist, können unprivilegierte Benutzer keine neuen Namensräume erstellen.  
1426 Die Möglichkeit, dass Benutzer ihre eigenen Namespaces erstellen können, war Teil mehrerer kürzlich  
1427 erfolgter lokaler Privilegienerweiterungen. Wenn Sie also Benutzernamespaces benötigen, aber paranoid  
1428 bzw. sicherheitsbewusst sind, sollten Sie diese Funktion deaktivieren. Diese Einstellung kann zur Laufzeit  
1429 mit dem `kernel.unprivileged_usersns_clone` `sysctl` außer Kraft gesetzt werden.

1430 Wenn Sie unsicher sind, sagen Sie Y.

1431 *When disabled, unprivileged users will not be able to create new namespaces. Allowing users to create their own  
1432 namespaces has been part of several recent local privilege escalation exploits, so if you need user namespaces but are  
1433 paranoid, security-conscious you want to disable this.*

1434 *This setting can be overridden at runtime via the `kernel.unprivileged_usersns_clone` `sysctl`.*

1435 *If unsure, say Y.*

#### 1436 **1.33.5 PID namespace**

1437 **CONFIG\_PID\_NS [=y] [Y]**

1438 Unterstützung von Prozess-ID-Namensräumen. Dies ermöglicht es, mehrere Prozesse mit der gleichen  
1439 pid zu haben, solange sie sich in verschiedenen pid-Namensräumen befinden. Dies ist ein Baustein von  
1440 Containern.

1441 *Support process id namespaces. This allows having multiple processes with the same pid as long as they are in  
1442 different pid namespaces. This is a building block of containers.*

#### 1443 **1.33.6 Network namespace**

1444 **CONFIG\_NET\_NS [=y] [Y]**

1445 Ermöglicht es dem Benutzer, scheinbar mehrere Instanzen des Netzwerkstapels zu erstellen.

1446 *Allow user space to create what appear to be multiple instances of the network stack.*

### 1447 **1.34 Checkpoint/restore support**

1448 **CONFIG\_CHECKPOINT\_RESTORE [=y] [Y]**

1449 Ermöglicht zusätzliche Kernel-Funktionen in einer Art Checkpoint/Restore. Insbesondere fügt es zusätzliche  
1450 prctl-Codes zum Einrichten von Prozesstext, Daten- und Heap-Segmentgrößen sowie einige zusätzliche  
1451 `/proc`-Dateisystemeinträge hinzu.

1452 Wenn Sie unsicher sind, geben Sie hier N an.

1453 *Enables additional kernel features in a sake of checkpoint/restore. In particular it adds auxiliary prctl codes to  
1454 setup process text, data and heap segment sizes, and a few additional `/proc` filesystem entries.*

1455 *If unsure, say N here.*

### 1.35 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.

*This option optimizes the scheduler for common desktop workloads by automatically creating and populating task groups. This separation of workloads isolates aggressive CPU burners (like build jobs) from desktop applications. Task group autogeneration is currently based upon task session.*

### 1.36 Kernel→user space relay support (formerly relays)

CONFIG\_RELAY [=y] [Y]

Diese Option aktiviert die Unterstützung für die Relaischnittstelle 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.

*This option enables support for relay interface support in certain file systems (such as debugfs). It is designed to provide an efficient mechanism for tools and facilities to relay large amounts of data from kernel space to user space.*

*If unsure, say N.*

### 1.37 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 kB (auf einigen anderen Architekturen mehr) zur Kernelgröße hinzu.

Wenn Sie unsicher sind, sagen Sie Y.

*The initial RAM filesystem is a ramfs which is loaded by the boot loader (loadlin or lilo) and that is mounted as root before the normal boot procedure. It is typically used to load modules needed to mount the “real” root file system, etc. See <file:Documentation/admin-guide/initrd.rst> for details.*

*If RAM disk support (BLK\_DEV\_RAM) is also included, this also enables initial RAM disk (initrd) support and adds 15 Kbytes (more on some other architectures) to the kernel size.*

*If unsure say Y.*

#### 1.37.1 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)

*This can be either a single cpio archive with a .cpio suffix or a space-separated list of directories and files for building the initramfs image. A cpio archive should contain a filesystem archive to be used as an initramfs image. Directories should contain a filesystem layout to be included in the initramfs image. Files should contain entries according to the format described by the “usr/gen\_init\_cpio” program in the kernel tree.*

1507 *When multiple directories and files are specified then the initramfs image will be the aggregate of all of them.*  
1508 *See <file:Documentation/driver-api/early-userspace/early\_userspace\_support.rst> for more details. If you are not*  
1509 *sure, leave it blank.*

### 1510 **1.37.2 Support initial ramdisk/ramfs compressed using gzip**

1511 **CONFIG\_RD\_GZIP** [=y] [N]  
1512 Unterstützung des Ladens eines gzip-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1513 Wenn Sie unsicher sind, sagen Sie Y.  
1514 *Support loading of a gzip encoded initial ramdisk or cpio buffer. If unsure, say Y.*  
1515 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1516 **1.37.3 Support initial ramdisk/ramfs compressed using bzip2**

1517 **CONFIG\_RD\_BZIP2** [=y] [N]  
1518 Unterstützung des Ladens eines bzip2-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1519 Wenn Sie unsicher sind, sagen Sie Y.  
1520 *Support loading of a bzip2 encoded initial ramdisk or cpio buffer*  
1521 *If unsure, say N.*  
1522 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1523 **1.37.4 Support initial ramdisk/ramfs compressed using LZMA**

1524 **CONFIG\_RD\_LZMA** [=y] [N]  
1525 Unterstützung des Ladens eines LZMA-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1526 Wenn Sie unsicher sind, sagen Sie Y.  
1527 *Support loading of a LZMA encoded initial ramdisk or cpio buffer*  
1528 *If unsure, say N.*  
1529 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1530 **1.37.5 Support initial ramdisk/ramfs compressed using XZ**

1531 **CONFIG\_RD\_XZ** [=y] [N]  
1532 Unterstützung des Ladens eines XZ-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1533 Wenn Sie unsicher sind, sagen Sie Y.  
1534 *Support loading of a XZ encoded initial ramdisk or cpio buffer. If unsure, say N.*  
1535 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1536 **1.37.6 Support initial ramdisk/ramfs compressed using LZO**

1537 **CONFIG\_RD\_LZO** [=y] [N]  
1538 Unterstützung des Ladens eines LZO-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1539 Wenn Sie unsicher sind, sagen Sie Y.  
1540 *Support loading of a LZO encoded initial ramdisk or cpio buffer*  
1541 *If unsure, say N.*  
1542 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1543 **1.37.7 Support initial ramdisk/ramfs compressed using LZ4**

1544 **CONFIG\_RD\_LZ4** [=y] [N]  
1545 Unterstützung des Ladens eines LZ4-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.  
1546 Wenn Sie unsicher sind, sagen Sie Y.  
1547 *Support loading of a LZ4 encoded initial ramdisk or cpio buffer*  
1548 *If unsure, say N.*  
1549 Seit dem Kernel 5.9 wird standardmäßig mit ZSTD komprimiert.

### 1.37.8 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.

*Support loading of a ZSTD encoded initial ramdisk or cpio buffer.*

*If unsure, say N.*

## 1.38 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.

*Extra boot config allows system admin to pass a config file as complemental extension of kernel cmdline when booting. The boot config file must be attached at the end of initramfs with checksum, size and magic word. See*

*<file:Documentation/admin-guide/bootconfig.rst> for details. If unsure, say Y.*

### 1.38.1 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.

*With this Kconfig option set, BOOT\_CONFIG processing is carried out even when the "bootconfig" kernel-boot parameter is omitted. In fact, with this Kconfig option set, there is no way to make the kernel ignore the BOOT\_CONFIG-supplied kernel-boot parameters.*

*If unsure, say N.*

### 1.38.2 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.

*Embed a bootconfig file given by BOOT\_CONFIG\_EMBED\_FILE in the kernel. Usually, the bootconfig file is loaded with the initrd image. But if the system doesn't support initrd, this option will help you by embedding a bootconfig file while building the kernel.*

*If unsure, say N*

## 1.39 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.

*Each entry in an initramfs cpio archive carries an mtime value. When enabled, extracted cpio items take this mtime, with directory mtime setting deferred until after creation of any child entries.*

*If unsure, say Y.*

## 1.40 Compiler optimization level (Optimize for performance) →

Optimierungsgrad des Compilers, Auswahl aus den folgenden zwei Punkten:

1598 *There is no help available for this option.*

#### 1599 **1.40.1 Optimize for performance (-O2)**

1600 `CONFIG_CC_OPTIMIZE_FOR_PERFORMANCE [=y] [Y]`

1601 Dies ist die Standardoptimierungsstufe für den Kernel, die mit dem Compiler-Flag `-O2` erstellt wird, um  
1602 die beste Leistung und die hilfreichsten Warnungen bei der Kompilierung zu erhalten.

1603 *This is the default optimization level for the kernel, building with the “-O2” compiler flag for best performance and*  
1604 *most helpful compile-time warnings.*

#### 1605 **1.40.2 Optimize for size (-Os)**

1606 `CONFIG_CC_OPTIMIZE_FOR_SIZE [=n] [N]`

1607 Wenn Sie diese Option wählen, wird `-Os` an Ihren Compiler übergeben, was zu einem kleineren Kernel  
1608 führt.

1609 *Choosing this option will pass “-Os” to your compiler resulting in a smaller kernel.*

### 1610 **1.41 Sysfs syscall support**

1611 `CONFIG_SYSFS_SYSCALL [=y] [N]`

1612 `sys_sysfs` ist ein veralteter Systemaufruf, der von der libc nicht mehr unterstützt wird. Beachten Sie, dass  
1613 die Deaktivierung dieser Option sicherer ist, aber die Kompatibilität mit einigen Systemen beeinträchtigen  
1614 könnte.

1615 Wenn Sie unsicher sind, sagen Sie hier Y.

1616 *sys\_sysfs is an obsolete system call no longer supported in libc. Note that disabling this option is more secure but*  
1617 *might break compatibility with some systems.*

1618 *If unsure, say Y here.*

1619 Eventuell könnten wir dies auf N stellen.

### 1620 **1.42 Configure standard kernel features (expert users)**

1621 `CONFIG_EXPERT [=n] [Y]`

1622 Mit dieser Option können bestimmte Basis-Kerneloptionen und -einstellungen deaktiviert oder optimiert  
1623 werden. Dies ist für spezielle Umgebungen gedacht, die einen „Nicht-Standard“-Kernel tolerieren können.  
1624 Verwenden Sie diese Option nur, wenn Sie wirklich wissen, was Sie tun.

1625 *This option allows certain base kernel options and settings to be disabled or tweaked. This is for specialized*  
1626 *environments which can tolerate a “non-standard” kernel. Only use this if you really know what you are doing.*

#### 1627 **1.42.1 Enable 16-bit UID system calls**

1628 `CONFIG_UID16 [=y] [Y]`

1629 Damit werden die alten 16-Bit-UID-Syscall-Wrappers aktiviert.

1630 *This enables the legacy 16-bit UID syscall wrappers.*

#### 1631 **1.42.2 Multiple users, groups and capabilities support**

1632 `CONFIG_MULTIVER [=y] [Y]`

1633 Diese Option aktiviert die Unterstützung für Nicht-Root-Benutzer, -Gruppen und -Fähigkeiten. Wenn  
1634 Sie hier N angeben, werden alle Prozesse mit UID 0, GID 0 und allen möglichen Fähigkeiten ausgeführt.  
1635 Wenn Sie hier N angeben, wird auch die Unterstützung für Systemaufrufe, die sich auf UIDs, GIDs und  
1636 Fähigkeiten beziehen, wie `setuid`, `setgid` und `capset`, herauskompiliert.

1637 Wenn Sie unsicher sind, sagen Sie hier Y.

1638 *This option enables support for non-root users, groups and capabilities. If you say N here, all processes will run*  
1639 *with UID 0, GID 0, and all possible capabilities. Saying N here also compiles out support for system calls related*  
1640 *to UIDs, GIDs, and capabilities, such as setuid, setgid, and capset.*

1641 *If unsure, say Y here.*

1642 Dies wäre nur auf kleinen Systemen ohne Benutzer sinnvoll, auf N zu stellen.

### 1.42.3 sgetmask/ssetmask syscalls support

CONFIG\_SGETMASK\_SYSCALL [=y] [Y]

sys\_sgetmask und sys\_ssetmask sind veraltete Systemaufrufe, die in der libc nicht mehr unterstützt werden, aber in einigen Architekturen immer noch standardmäßig aktiviert sind.

Wenn Sie unsicher sind, lassen Sie die Standardoption hier.

*sys\_sgetmask and sys\_ssetmask are obsolete system calls no longer supported in libc but still enabled by default in some architectures.*

*If unsure, leave the default option here.*

Eventuell könnten wir dies auf N stellen.

### 1.42.4 open by fhandle syscalls

CONFIG\_FHANDLE [=y] [Y]

Wenn Sie hier Y angeben, kann ein Programm auf Benutzerebene Dateinamen auf Handles abbilden und diese Handles dann später für verschiedene Dateisystemoperationen verwenden. Dies ist nützlich bei der Implementierung von Userspace-Dateiservern, die jetzt Dateien über Handles anstelle von Namen verfolgen. Das Handle bleibt gleich, auch wenn die Dateinamen umbenannt werden. Ermöglicht `open_by_handle_at(2)` und `name_to_handle_at(2)` Syscalls.

*If you say Y here, a user level program will be able to map file names to handle and then later use the handle for different file system operations. This is useful in implementing userspace file servers, which now track files using handles instead of names. The handle would remain the same even if file names get renamed. Enables `open_by_handle_at(2)` and `name_to_handle_at(2)` syscalls.*

### 1.42.5 Posix Clocks & timers

CONFIG\_POSIX\_TIMERS [=y] [Y]

Dazu gehört die native Unterstützung für POSIX-Timer im Kernel. Einige eingebettete Systeme haben keine Verwendung für sie und können daher deaktiviert werden, um die Größe des Kernel-Images zu verringern.

Wenn diese Option deaktiviert ist, sind die folgenden Syscalls nicht verfügbar:

`timer_create`, `timer_gettime`, `timer_getoverrun`, `timer_settime`, `timer_delete`, `clock_adjtime`, `getitimer`, `setitimer`, `alarm`.

Außerdem werden die Syscalls `clock_settime`, `clock_gettime`, `clock_getres` und `clock_nanosleep` nur auf `CLOCK_REALTIME`, `CLOCK_MONOTONIC` und `CLOCK_BOOTTIME` beschränkt.

Wenn Sie unsicher sind, sagen Sie y.

*This includes native support for POSIX timers to the kernel. Some embedded systems have no use for them and therefore they can be configured out to reduce the size of the kernel image. When this option is disabled, the following syscalls won't be available: `timer_create`, `timer_gettime`, `timer_getoverrun`, `timer_settime`, `timer_delete`, `clock_adjtime`, `getitimer`, `setitimer`, `alarm`.*

*Furthermore, the `clock_settime`, `clock_gettime`, `clock_getres` and `clock_nanosleep` syscalls will be limited to `CLOCK_REALTIME`, `CLOCK_MONOTONIC` and `CLOCK_BOOTTIME` only.*

*If unsure say y.*

### 1.42.6 Enable support for printk

CONFIG\_PRINTK [=y] [Y]

Diese Option aktiviert die normale printk-Unterstützung. Wenn Sie diese Option entfernen, werden die meisten Meldungsstrings aus dem Kernel-Image entfernt und der Kernel wird mehr oder weniger still.

Da dies die Diagnose von Systemproblemen sehr erschwert, wird von der Angabe von N hier dringend abgeraten.

*This option enables normal printk support. Removing it eliminates most of the message strings from the kernel image and makes the kernel more or less silent. As this makes it very difficult to diagnose system problems, saying N here is strongly discouraged.*

### 1.42.7 BUG() support

CONFIG\_BUG [=y] [Y]

Durch die Deaktivierung dieser Option wird die Unterstützung für BUG und WARN eliminiert, wodurch die Größe Ihres Kernel-Images verringert und zahlreiche fatale Zustände möglicherweise stillschweigend ignoriert werden. Sie sollten die Deaktivierung dieser Option nur für eingebettete Systeme in Erwägung ziehen, die keine Möglichkeit haben, Fehler zu melden.

Sagen Sie einfach Y.

*Disabling this option eliminates support for BUG and WARN, reducing the size of your kernel image and potentially quietly ignoring numerous fatal conditions. You should only consider disabling this option for embedded systems with no facilities for reporting errors.*

*Just say Y.*

### 1.42.8 Enable ELF core dumps

CONFIG\_ELF\_CORE [=y] [Y]

Aktivieren Sie die Unterstützung für die Erzeugung von Kerneldumps. Die Deaktivierung spart etwa 4 kB.

*Enable support for generating core dumps. Disabling saves about 4k.*

### 1.42.9 Enable PC-Speaker support

CONFIG\_PCSPKR\_PLATFORM [=y] [N]

Mit dieser Option können Sie die Unterstützung des internen PC-Lautsprechers deaktivieren, um Speicherplatz zu sparen.

*This option allows to disable the internal PC speaker support, saving some memory.*

### 1.42.10 Enable smaller-sized data structures for core

CONFIG\_BASE\_SMALL [=n] [N]

Wenn Sie diese Option aktivieren, wird die Größe verschiedener Kernel-Datenstrukturen reduziert. Dies spart Speicher auf kleinen Rechnern, kann aber die Leistung verringern.

*Enabling this option reduces the size of miscellaneous core kernel data structures. This saves memory on small machines, but may reduce performance.*

### 1.42.11 Enable futex support

CONFIG\_FUTEX [=y] [Y]

Die Deaktivierung dieser Option führt dazu, dass der Kernel ohne Unterstützung für „schnelle User-space-Mutexe“ gebaut wird. Der resultierende Kernel führt möglicherweise glibc-basierte Anwendungen nicht korrekt aus.

*Disabling this option will cause the kernel to be built without support for “fast userspace mutexes”. The resulting kernel may not run glibc-based applications correctly.*

### 1.42.12 Enable eventpoll support

CONFIG\_EPOLL [=y] [Y]

Die Deaktivierung dieser Option führt dazu, dass der Kernel ohne Unterstützung für die epoll-Familie von Systemaufrufen gebaut wird.

*Disabling this option will cause the kernel to be built without support for epoll family of system calls.*

### 1.42.13 Enable signalfd() system call

CONFIG\_SIGNALFD [=y] [Y]

Aktivieren Sie den Systemaufruf `signalfd()`, der den Empfang von Signalen auf einem Dateideskriptor ermöglicht.

Wenn Sie unsicher sind, sagen Sie Y.

*Enable the signalfd() system call that allows to receive signals on a file descriptor.*

*If unsure, say Y.*

#### 1.42.14 Enable timerfd() system call

CONFIG\_TIMERFD [=y] [Y]

Aktivieren Sie den Systemaufruf `timerfd()`, der den Empfang von Timer-Ereignissen auf einem Dateideskriptor ermöglicht.

Wenn Sie unsicher sind, sagen Sie Y.

*Enable the timerfd() system call that allows to receive timer events on a file descriptor.*

*If unsure, say Y.*

#### 1.42.15 Enable eventfd() system call

CONFIG\_EVENTFD [=y] [Y]

Aktivieren Sie den Systemaufruf `eventfd()`, der es ermöglicht, sowohl Kernel-Benachrichtigungen (z. B. KAIO) als auch Userspace-Benachrichtigungen zu empfangen.

Wenn Sie unsicher sind, sagen Sie Y.

*Enable the eventfd() system call that allows to receive both kernel notification (ie. KAIO) or userspace notifications.*

*If unsure, say Y.*

#### 1.42.16 Use full shmem filesystem

CONFIG\_SHMEM [=y] [Y]

Das `shmem` ist ein internes Dateisystem, das zur Verwaltung des gemeinsamen Speichers verwendet wird. Es wird von `swap` unterstützt und verwaltet Ressourcengrenzen. Es wird auch in den Userspace als `tmpfs` exportiert, wenn TMPFS aktiviert ist. Wenn Sie diese Option deaktivieren, werden `shmem` und `tmpfs` durch den viel einfacheren `ramfs`-Code ersetzt, der auf kleinen Systemen ohne `swap` geeignet sein kann.

*The shmem is an internal filesystem used to manage shared memory. It is backed by swap and manages resource limits. It is also exported to userspace as tmpfs if TMPFS is enabled. Disabling this option replaces shmem and tmpfs with the much simpler ramfs code, which may be appropriate on small systems without swap.*

#### 1.42.17 Enable AIO support

CONFIG\_AIO [=y] [Y]

Diese Option aktiviert die asynchrone POSIX-E/A, die von einigen Hochleistungs-Thread-Anwendungen verwendet werden kann. Die Deaktivierung dieser Option spart etwa 7kB.

*This option enables POSIX asynchronous I/O which may be used by some high performance threaded applications.*

*Disabling this option saves about 7k.*

#### 1.42.18 Enable IO uring support

CONFIG\_IO\_URING [=y] [Y]

Diese Option aktiviert die Unterstützung für die `io_uring`-Schnittstelle, die es Anwendungen ermöglicht, IO über Übermittlungs- und Abschlussringe zu übermitteln und abzuschließen, die von Kernel und Anwendung gemeinsam genutzt werden.

*This option enables support for the io\_uring interface, enabling applications to submit and complete IO through submission and completion rings that are shared between the kernel and application.*

### 1.43 Enable madvise/fadvise syscalls

CONFIG\_ADVICE\_SYSCALLS [=y] [Y]

Diese Option aktiviert die Syscalls `madvise` und `fadvise`, die von Anwendungen verwendet werden, um den Kernel über ihre zukünftige Speicher- oder Dateinutzung zu informieren und so die Leistung zu verbessern. Wenn Sie ein eingebettetes System bauen, bei dem keine Anwendungen diese Syscalls verwenden, können Sie diese Option deaktivieren, um Platz zu sparen.

*This option enables the madvise and fadvise syscalls, used by applications to advise the kernel about their future memory or file usage, improving performance. If building an embedded system where no applications use these syscalls, you can disable this option to save space.*

## 1.44 Enable membarrier() system call

CONFIG\_MEMBARRIER [=y] [Y]

Aktivieren Sie den Systemaufruf `membarrier()`, der die Ausgabe von Speicherbarrieren für alle laufenden Threads ermöglicht. Damit können die Kosten für Speicherbarrieren im Benutzerbereich asymmetrisch verteilt werden, indem Paare von Speicherbarrieren in Paare bestehend aus `membarrier()` und einer Kompilerbarriere umgewandelt werden.

Wenn Sie unsicher sind, sagen Sie Y.

*Enable the membarrier() system call that allows issuing memory barriers across all running threads, which can be used to distribute the cost of user-space memory barriers asymmetrically by transforming pairs of memory barriers into pairs consisting of membarrier() and a compiler barrier.*

*If unsure, say Y.*

## 1.45 Enable kcmp() system call

CONFIG\_KCMP [=y] [Y]

Aktivieren Sie den Systemaufruf Kernel-Ressourcenvergleich. Er bietet dem Benutzerbereich die Möglichkeit, zwei Prozesse zu vergleichen, um festzustellen, ob sie eine gemeinsame Ressource teilen, wie z. B. einen Dateideskriptor oder sogar virtuellen Speicherplatz.

Wenn Sie unsicher sind, sagen Sie N.

*Enable the kernel resource comparison system call. It provides user-space with the ability to compare two processes to see if they share a common resource, such as a file descriptor or even virtual memory space.*

*If unsure, say N.*

## 1.46 Enable rseq() system call

CONFIG\_RSEQ [=y] [Y]

Aktivieren Sie den Systemaufruf „Neustartbare Sequenzen“. Er bietet einen User-Space-Cache für den aktuellen CPU-Nummernwert, der das Abrufen der aktuellen CPU-Nummer aus dem User-Space beschleunigt, sowie eine ABI zur Beschleunigung von User-Space-Operationen auf pro-CPU-Daten.

Wenn Sie unsicher sind, sagen Sie Y.

*Enable the restartable sequences system call. It provides a user-space cache for the current CPU number value, which speeds up getting the current CPU number from user-space, as well as an ABI to speed up user-space operations on per-CPU data.*

*If unsure, say Y.*

### 1.46.1 Enable debugging of rseq() system call

CONFIG\_DEBUG\_RSEQ [=n] [N]

Aktivieren Sie zusätzliche Debugging-Prüfungen für den Systemaufruf `rseq`.

Wenn Sie unsicher sind, sagen Sie N.

*Enable extra debugging checks for the rseq system call.*

*If unsure, say N.*

## 1.47 Enable cachestat() system call

CONFIG\_CACHESTAT\_SYSCALL [=y] [Y]

Aktivieren Sie den Systemaufruf `cachestat`, der die Seiten-Cache-Statistiken einer Datei abfragt (Anzahl der zwischengespeicherten Seiten, schmutzige Seiten, zum Zurückschreiben markierte Seiten, (kürzlich) entfernte Seiten).

Wenn Sie unsicher sind, sagen Sie hier Y.

*Enable the cachestat system call, which queries the page cache statistics of a file (number of cached pages, dirty pages, pages marked for writeback, (recently) evicted pages).*

*If unsure say Y here.*

## 1.48 PC/104 support

CONFIG\_PC104 [=n] [N]

Zeigt PC/104-Formfaktor-Gerätetreiber und Optionen zur Auswahl und Konfiguration an. Aktivieren Sie

diese Option, wenn Ihr Zielrechner über einen PC/104-Bus verfügt.

*Expose PC/104 form factor device drivers and options available for selection and configuration. Enable this option if your target machine has a PC/104 bus.*

## 1.49 Load all symbols for debugging/ksymoops

CONFIG\_KALLSYMS [=y] [Y]

(sichtbar wenn EXPERT [=y])

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.

*Say Y here to let the kernel print out symbolic crash information and symbolic stack backtraces. This increases the size of the kernel somewhat, as all symbols have to be loaded into the kernel image.*

### 1.49.1 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.

*Test the basic functions and performance of some interfaces, such as kallsyms\_lookup\_name. It also calculates the compression rate of the kallsyms compression algorithm for the current symbol set.*

*Start self-test automatically after system startup. Suggest executing "dmesg | grep kallsyms\_selftest" to collect test results. "finish" is displayed in the last line, indicating that the test is complete.*

### 1.49.2 Include all symbols in kallsyms

CONFIG\_KALLSYMS\_ALL [=y] [N]

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 300 KiB oder etwas Ähnliches betragen).

Sagen Sie N, es sei denn, Sie brauchen wirklich alle Symbole, oder Kernel-Live-Patching.

*Normally kallsyms only contains the symbols of functions for nicer OOPS messages and backtraces (i.e., symbols from the text and inittext sections). This is sufficient for most cases. And only if you want to enable kernel live patching, or other less common use cases (e.g., when a debugger is used) all symbols are required (i.e., names of variables from the data sections, etc).*

*This option makes sure that all symbols are loaded into the kernel image (i.e., symbols from all sections) in cost of increased kernel size (depending on the kernel configuration, it may be 300KiB or something like this).*

*Say N unless you really need all symbols, or kernel live patching.*

## 1.50 Kernel Performance Events And Counters →

Kernel-Leistungsereignisse und -Zähler

### 1.50.1 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 vom 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.

*Enable kernel support for various performance events provided by software and hardware.*

*Software events are supported either built-in or via the use of generic tracepoints.*

*Most modern CPUs support performance events via performance counter registers. These registers count the number of certain types of hw events: such as instructions executed, cachemisses suffered, or branches mis-predicted – without slowing down the kernel or applications. These registers can also trigger interrupts when a threshold number of events have passed – and can thus be used to profile the code that runs on that CPU.*

*The Linux Performance Event subsystem provides an abstraction of these software and hardware event capabilities, available via a system call and used by the “perf” utility in tools/perf/. It provides per task and per CPU counters, and it provides event capabilities on top of those.*

*Say Y if unsure.*

#### 1.50.1.1 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.

*Use vmalloc memory to back perf mmap() buffers.*

*Mostly useful for debugging the vmalloc code on platforms that don't require it.*

*Say N if unsure.*

### 1.51 Profiling support

`CONFIG_PROFILING` [=y] [Y]

Sagen Sie hier Y, um die erweiterten Unterstützungsmechanismen für das Profiling zu aktivieren, die von Profilern verwendet werden.

*Say Y here to enable the extended profiling support mechanisms used by profilers.*

### 1.52 Rust support

`CONFIG_RUST` [=y] [Y]

Aktiviert die Rust-Unterstützung im Kernel.

Dies ermöglicht die Auswahl anderer Rust-bezogener Optionen, wie z. B. in Rust geschriebene Treiber.

Es ist auch erforderlich, um externe, in Rust geschriebene Kernel-Module laden zu können.

Siehe `Dokumentation/rust/` für weitere Informationen.

Wenn Sie unsicher sind, sagen Sie N.

*Enables Rust support in the kernel.*

*This allows other Rust-related options, like drivers written in Rust, to be selected.*

*It is also required to be able to load external kernel modules written in Rust.*

*See `Dokumentation/rust/` for more information.*

*If unsure, say N.*

### 1.53 Kexec and crash features →

*Kexec und Absturzmerkmale*

### 1.53.1 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.

*kexec is a system call that implements the ability to shutdown your current kernel, and to start another kernel. It is like a reboot but it is independent of the system firmware. And like a reboot you can start any kernel with it, not just Linux.*

*The name comes from the similarity to the exec system call.*

*It is an ongoing process to be certain the hardware in a machine is properly shutdown, so do not be surprised if this code does not initially work for you. As of this writing the exact hardware interface is strongly in flux, so no good recommendation can be made.*

### 1.53.2 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.

*This is new version of kexec system call. This system call is file based and takes file descriptors as system call argument for kernel and initramfs as opposed to list of segments as accepted by kexec system call.*

#### 1.53.2.1 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.

*This option makes the kexec\_file\_load() syscall check for a valid signature of the kernel image. The image can still be loaded without a valid signature unless you also enable KEXEC\_SIG\_FORCE, though if there's a signature that we can check, then it must be valid.*

*In addition to this option, you need to enable signature verification for the corresponding kernel image type being loaded in order for this to work.*

##### 1.53.2.1.1 Require a valid signature in kexec\_file\_load() syscall

CONFIG\_KEXEC\_SIG\_FORCE [=n] [N]

Diese Option macht die Überprüfung der Kernelsignatur für den Syscall **kexec\_file\_load()** zwingend erforderlich.

*This option makes kernel signature verification mandatory for the kexec\_file\_load() syscall.*

##### 1.53.2.1.2 Enable bzImage signature verification support

CONFIG\_KEXEC\_BZIMAGE\_VERIFY\_SIG [=y] [Y]

Aktivierung der Unterstützung von bzImage für die Signaturprüfung.

*Enable bzImage signature verification support.*

### 1.53.3 kexec jump

CONFIG\_KEXEC\_JUMP [=y] [Y]

Sprung zwischen Original-Kernel und **kexec**-Kernel und Aufruf von Code im physikalischen Adressmodus über **KEXEC**

1973 *Jump between original kernel and kexeced kernel and invoke code in physical address mode via KEXEC*

#### 1974 **1.53.4 kernel crash dumps**

1975 **CONFIG\_CRASH\_DUMP [=y] [Y]**

1976 Erstellen Sie nach dem Start durch kexec einen Absturz-Dump. Dies sollte normalerweise nur in spe-  
1977 ziellen Absturz-Dump-Kernen eingestellt werden, die mit kexec-tools in den Hauptkernel in einen  
1978 speziell reservierten Bereich geladen und später nach einem Absturz von kdump/kexec ausgeführt  
1979 werden. Der Absturz-Dump-Kernel muss mit PHYSICAL\_START auf eine Speicheradresse kompi-  
1980 liert werden, die nicht vom Hauptkernel oder BIOS verwendet wird, oder er muss als verschiebbares  
1981 Image (CONFIG\_RELOCATABLE=y) erstellt werden. Weitere Details finden Sie unter Documen-  
1982 tation/admin-guide/kdump/kdump.rst

1983 Für s390 aktiviert diese Option auch zfcpdump.

1984 Siehe auch <file:Documentation/arch/s390/zfcpdump.rst>

1985 *Generate crash dump after being started by kexec. This should be normally only set in special crash dump kernels*  
1986 *which are loaded in the main kernel with kexec-tools into a specially reserved region and then later executed after a*  
1987 *crash by kdump/kexec. The crash dump kernel must be compiled to a memory address not used by the main kernel*  
1988 *or BIOS using PHYSICAL\_START, or it must be built as a relocatable image (CONFIG\_RELOCATABLE=y).*  
1989 *For more details see Documentation/admin-guide/kdump/kdump.rst*

1990 *For s390, this option also enables zfcpdump.*

1991 *See also <file:Documentation/arch/s390/zfcpdump.rst>*

#### 1992 **1.53.5 kexec crash dumps**

1993 **CONFIG\_KEXEC\_DUMP [=y] [Y]**

1994 Absturzdump (Speicherauszug) erzeugen, nachdem er von kexec gestartet wurde. Dies sollte normalerweise  
1995 nur in speziellen Crash-Dump-Kernen gesetzt werden, die im Hauptkernel mit kexec-tools in einen  
1996 speziell reservierten Bereich geladen werden und dann später nach einem Absturz von kdump/kexec  
1997 ausgeführt werden. Der Crash-Dump-Kernel muss mit PHYSICAL\_START auf eine Speicheradresse  
1998 kompiliert werden, die nicht vom Hauptkernel oder BIOS verwendet wird, oder er muss als relocatable  
1999 image (CONFIG\_RELOCATABLE=y) erstellt werden.

2000 Für weitere Details siehe Documentation/admin-guide/kdump/kdump.rst

2001 Für s390 aktiviert diese Option auch zfcpdump.

2002 Siehe auch <file:Documentation/s390/zfcpdump.rst>

2003 *Generate crash dump after being started by kexec. This should be normally only set in special crash dump kernels*  
2004 *which are loaded in the main kernel with kexec-tools into a specially reserved region and then later executed after a*  
2005 *crash by kdump/kexec. The crash dump kernel must be compiled to a memory address not used by the main kernel*  
2006 *or BIOS using PHYSICAL\_START, or it must be built as a relocatable image (CONFIG\_RELOCATABLE=y).*  
2007 *For more details see Documentation/admin-guide/kdump/kdump.rst*

2008 *For s390, this option also enables zfcpdump. See also <file:Documentation/arch/s390/zfcpdump.rst>*

#### 2009 **1.53.5.1 Support saving crash dump to dm-crypt encrypted volume** seit 6.6

2010 **CONFIG\_CRASH\_DM\_CRYPT [=y] [Y]**

2011 Ist diese Option aktiviert, kann der Benutzerbereich mit /sys/kernel/config/crash\_dm\_crypt\_keys  
2012 interagieren, um die dm-Verschlüsselungsschlüssel für den Dump-Capture-Kernel dauerhaft zu speichern.

2013 *With this option enabled, user space can interact with /sys/kernel/config/crash\_dm\_crypt\_keys to make the dm*  
2014 *crypt keys persistent for the dump-capture kernel.*

#### 2015 **1.53.6 Update the crash elfcorehdr on system configuration changes**

2016 **CONFIG\_CRASH\_HOTPLUG [=y] [Y]**

2017 Aktivierung der direkten Aktualisierung der Crash-Elfcorehdr (die die Liste der CPUs und Speicherbe-  
2018 reiche enthält, die bei einem Absturz gelöscht werden sollen) als Reaktion auf Hot-Plug/Unplug oder  
2019 Online/Offline von CPUs oder Speicher. Dies ist ein sehr viel fortschrittlicherer Ansatz als der Versuch  
2020 dies im Userspace zu tun.

2021 Wenn Sie unsicher sind, sagen Sie Y.

2022 *Enable direct update to the crash elfcorehdr (which contains the list of CPUs and memory regions to be dumped*  
2023 *upon a crash) in response to hot plug/unplug or online/offline of CPUs or memory. This is a much more advanced*

*approach than userspace attempting that.*  
*If unsure, say Y.*

### 1.53.6.1 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 bzw. 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) 1 TiB-System ab, das aus 128 MiB-Memblöcken besteht, und führt zu einer `elfcorehdr`-Speicherpuffer-/Segmentgröße von unter 1 MiB. 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.

*For the kexec\_file\_load() syscall path, specify the maximum number of memory regions that the elfcorehdr buffer / segment can accommodate. These regions are obtained via walk\_system\_ram\_res(); eg. the 'System RAM' entries in /proc/iomem. This value is combined with NR\_CPUS\_DEFAULT and multiplied by sizeof(Elf64\_Phdr) to determine the final elfcorehdr memory buffer/segment size. The value 8192, for example, covers a (sparsely populated) 1TiB system consisting of 128MiB memblocks, while resulting in an elfcorehdr memory buffer/segment size under 1MiB. This represents a sane choice to accommodate both baremetal and virtual machine configurations. For the kexec\_load() syscall path, CRASH\_MAX\_MEMORY\_RANGES is part of the computation behind the value provided through the /sys/kernel/crash\_elfcorehdr\_size attribute.*

## 1.54 Live Update and Kexec Handover → seit 6.19

*Live-Update und Kexec-Übergabe*

### 1.54.1 kexec handover seit 6.16

CONFIG\_KEXEC\_HANDOVER [=y] **[Y]**

Ermöglichen Sie `kexec` die Übergabe des Status zwischen Kernen, indem Sie zusätzliche Metadaten generieren und an den Zielkernel übergeben. Dies ist nützlich, um Daten oder den Status über `kexec` hinweg aufrechtzuerhalten. Damit dies funktioniert, muss diese Option sowohl im Quell- als auch im Zielkernel aktiviert sein.

*Allow kexec to hand over state across kernels by generating and passing additional metadata to the target kernel. This is useful to keep data or state alive across the kexec. For this to work, both source and target kernels need to have this option enabled.*

#### 1.54.1.1 Enable Kexec Handover debug checks seit 6.18

CONFIG\_KEXEC\_HANDOVER\_DEBUG [=n] **[N]**

Diese Option aktiviert zusätzliche Plausibilitätsprüfungen für das Kexec-Handover-Subsystem. Da die KHO-Leistung in Live-Update-Szenarien von entscheidender Bedeutung ist und der zusätzliche Code möglicherweise zu einem Mehraufwand führt, ist diese Option nur optional aktiviert.

*This option enables extra sanity checks for the Kexec Handover subsystem. Since, KHO performance is crucial in live update scenarios and the extra code might be adding overhead it is only optionally enabled.*

#### 1.54.1.2 kexec handover debugfs interface seit 6.19

CONFIG\_KEXEC\_HANDOVER\_DEBUGFS [=y] **[Y]**

Ermöglicht die Steuerung der Kexec-Übergabe des Device Tree über die Debugfs-Schnittstelle, d. h. die Fertigstellung des Status oder den Abbruch der Fertigstellung. Ermöglicht außerdem die Überprüfung der KHO-Fdt-Bäume mit den Debugfs-Binärblobs.

*Allow to control kexec handover device tree via debugfs interface, i.e. finalize the state or aborting the finalization. Also, enables inspecting the KHO fdt trees with the debugfs binary blobs.*

#### 1.54.1.3 Enable kexec handover by default seit 6.19

CONFIG\_KEXEC\_HANDOVER\_ENABLE\_DEFAULT [=n] **[N]**

2073 Ermöglicht standardmäßig die Kexec-Übergabe. Dadurch muss „kho=on“ nicht explizit in der Kernel-Be-  
2074 fehlszeile übergeben werden.  
2075 Dies ist nützlich für Systeme, bei denen KHO eine Voraussetzung für andere Funktionen wie Live Update  
2076 ist, da so sichergestellt wird, dass der Mechanismus immer aktiv ist.  
2077 Das Standardverhalten kann beim Booten weiterhin durch Übergeben von „kho=off“ überschrieben werden.  
2078 *Enable Kexec Handover by default. This avoids the need to explicitly pass 'kho=on' on the kernel command line.*  
2079 *This is useful for systems where KHO is a prerequisite for other features, such as Live Update, ensuring the*  
2080 *mechanism is always active.*  
2081 *The default behavior can still be overridden at boot time by passing 'kho=off'.*

## 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

*Say yes to build a 64-bit kernel – formerly known as x86\_64*

*Say no to build a 32-bit kernel – formerly known as i386*