

¹ Linux Configuration V6.19

² 0.1 Einführung

³ Dieses Dokument dient zur Beschreibung von diversen Einstellungen bei der Konfiguration zum Beispiel
⁴ mittels `make nconfig` oder `make menuconfig` unter Linux.

⁵ Es wird nicht näher darauf eingegangen, wie der Kernel kompiliert wird oder welche Voreinstellungen,
⁶ Programme usw. zum Kompilieren benötigt werden. Abweichungen zur bestehenden Arch-Linux-Konfigu-
⁷ ration werden immer **gelb** markiert.

⁸ Original Texte, sofern geliefert (englischer Sprache) werden *kursiv* und etwas kleiner geschrieben. Zu
⁹ Beginn der jeweiligen Konfigurationszeile wird der Standardwert (Default) angezeigt. Mein Vorschlag folgt
¹⁰ danach.

¹¹ Zum Beispiel bei `CONFIG_WERROR [=n] [Y]`

¹² Hier ist der Standardwert ein Nein [n], meine persönliche Einstellung ein Ja [Y].

¹³ 0.2 Konfiguration für ein verteiltes Kompilieren auf mehreren Rechnern

¹⁴ Sie sollten schon einiges an Erfahrung mit dem Kompilieren unter Linux mitbringen. Wenn Sie mit Linux
¹⁵ Arch arbeiten, muss auf jedem Rechner das Programm `distcc` installiert sein. Man unterscheidet nun
¹⁶ den **Client** auf dem üblicherweise das Kompilat erstellt wird und die weiteren beteiligten Rechner als
¹⁷ **Volunteers**.

¹⁸ Nähere Informationen siehe <https://wiki.archlinux.org/title/Distcc>.

¹⁹ ©KW4NZ, Thomas Kuschel

²⁰ Wenn Sie Korrekturen wünschen, dann schicken Sie dies mit Erläuterungen und dem Hinweis auf die
²¹ obenstehende Version V6.19 an: oe1kt@gmail.com

²² This is pdfTeX, Version 3.141592653-2.6-1.40.28 (TeX Live 2025) kpathsea version 6.4.1

23 **1 General setup →**

24 **1.1 Compile also drivers which will not load**

25 CONFIG_COMPILE_TEST [=n] []

26 *Kompilieren Sie auch Treiber, die nicht geladen werden können*

27 Einige Treiber können auf einer anderen Plattform kompiliert werden als auf der, für die sie gedacht sind.

28 Obwohl sie dort nicht geladen werden können (oder selbst wenn sie geladen werden können, können sie
29 aufgrund fehlender Hardware-Unterstützung nicht verwendet werden), möchten Entwickler, im Gegensatz
30 zu Distributoren, solche Treiber vielleicht trotzdem kompilieren und testen.

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

35 **1.2 Compile the kernel with warnings as errors**

36 CONFIG_WERROR [=n] []

37 *Den Kernel mit Fehlermeldungen bei Warnungen kompilieren*

38 Ein Build sollte keine Compiler-Warnungen ausgeben, dies aktiviert dieFlags -Werror (für C) und
39 -Dwarnings (für Rust) um diese Regel standardmäßig zu setzen. Bestimmte Warnungen von anderen Tools
40 z. B. der Linker könnte mit dieser Option Fehler generieren. Deaktivieren ist sinnvoll, wenn Sie einen
41 neuen (oder sehr alten) Compiler bzw. Linker mit seltenen, ungewöhnlichen Warnungen haben. Haben
42 Sie auf Ihrer Architektur Probleme, dann müssen Sie diese Konfiguration deaktivieren, um den Kernel
43 erfolgreich zu bauen. Im Zweifelsfall sagen sie Y für Ja.

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

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

49 *If in doubt, say Y.*

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

51 **1.3 Local version – append to kernel release**

52 CONFIG_LOCALVERSION [=] []

53 *Lokale Version – an die Kernelversion anhängen*

54 Type: string

55 Hängen Sie eine zusätzliche Zeichenkette an das Ende Ihrer Kernelversion an.

56 Dies wird angezeigt, wenn Sie z. B. uname eingeben. Die hier angegebene Zeichenfolge wird an den Inhalt
57 von einem Dateinamen mit localversion* als Objekt und im Quellbaum, in dieser Reihenfolge angezeigt.
58 Die Zeichenkette darf maximal 64 Zeichen lang sein.

59 *Append an extra string to the end of your kernel version. This will show up when you type uname, for example.
60 The string you set here will be appended after the contents of any files with a filename matching localversion* in
61 your object and source tree, in that order. Your total string can be a maximum of 64 characters.*

62 **1.4 Automatically append version information to the version string**

63 CONFIG_LOCALVERSION_AUTO [=y] [Y]

64 Dies versucht automatisch festzustellen, ob der aktuelle Baum ein Release-Tree ist, indem es nach Git-Tags
65 sucht, die zur aktuellen Top-of-Tree-Revision gehören.

66 Eine Zeichenkette des Formats -gxxxxxxxxx wird der lokalen Version hinzugefügt, wenn ein git-basierter
67 Baum gefunden wird. Die so erzeugte Zeichenkette wird nach allen passenden „localversion**“-Dateien und
68 nach dem in CONFIG_LOCALVERSION eingestellten Wert angehängt. (Die hier tatsächlich verwendete
69 Zeichenkette sind die ersten 12 Zeichen, die durch die Ausführung des Befehls erzeugt werden:

70 \$ git rev-parse -verify HEAD

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

72 *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 -xxxxxxxxx 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 eineindeutige 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.

122 **1.6.3 LZMA**

123 CONFIG_KERNEL_LZMA [=n] []

124 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.

127 *This compression algorithm's ratio is best. Decompression speed is between gzip and bzip2. Compression is slowest.*

128 *The kernel size is about 33% smaller with LZMA in comparison to gzip.*

129 **1.6.4 XZ**

130 CONFIG_KERNEL_XZ [=n] []

131 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.

137 *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.*

141 *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.*

143 **1.6.5 LZO**

144 CONFIG_KERNEL_LZO [=n] []

145 Kompressionsrate ist die schlechteste aller anderen. Kernelgröße ist etwa 10 % größer als GZIP. Jedoch ist die Geschwindigkeit beim Komprimieren und Dekomprimieren die höchste.

147 *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.*

149 **1.6.6 LZ4**

150 CONFIG_KERNEL_LZ4 [=n] []

151 LZ4 ist eine LZ77-Typ-Komprimierung mit einer festen, byte-orientierten Enkodierung.

152 Siehe auch <http://code.google.com/p/lz4>.

153 Komprimierungsverhältnis ist noch schlechter als LZO. 8 % größere Kernelgröße als bei LZO. Dekomprimierung ist jedoch von der Geschwindigkeit her schneller als LZO.

155 *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/>.*

157 *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.*

159 **1.6.7 ZSTD**

160 CONFIG_KERNEL_ZSTD [=y] [Y]

161 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.

165 *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.*

168 **1.7 Default init path**

169 CONFIG_DEFAULT_INIT [=] []

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

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

177 **1.8 Default hostname**

178 CONFIG_DEFAULT_HOSTNAME [=archlinux] [=orange]

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

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

185 **1.9 System V IPC**

186 CONFIG_SYSVIPC [=y] [Y]

187 Die Inter-Prozess-Kommunikation IPC ist eine Zusammenstellung aus Bibliotheksfunktionen (libraries) und
188 Systemaufrufen die Prozesse (laufende Programme) synchronisiert und Daten untereinander austauschen
189 kann. Generell ist das eine gute Sache, einige Programme würden auch nicht funktionieren wenn Sie hier
190 kein Y (ja) setzen. Insbesondere, wenn Sie den DOS-Emulator dosemu unter Linux laufen lassen wollen
191 (lesen Sie das DOSEMU-HOWTO, verfügbar unter <http://www.tldp.org/docs.html#howto>), müssen
192 Sie hier Y sagen.

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

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

201 **1.10 POSIX Message Queues**

202 CONFIG_POSIX_MQUEUE [=y] [Y]

203 Die POSIX-Variante der Nachrichtenwarteschlangen (message queues) ist ein Teil der IPC. In POSIX-
204 Nachrichtenwarteschlangen hat jede Nachricht eine Priorität, die über die Reihenfolge des Empfangs
205 durch einen Prozess entscheidet. Wenn Sie Programme kompilieren und ausführen wollen, die z. B. für
206 Solaris geschrieben wurden und die POSIX-Warteschlangen (Funktionen mq_*) verwenden, sagen Sie
207 hier Y. POSIX-Nachrichtenwarteschlangen sind als Dateisystem mit dem Namen „mqueue“ sichtbar und
208 können irgendwo eingehängt werden, wenn Sie Dateisystemoperationen auf Nachrichtenwarteschlangen
209 durchführen wollen. Wenn Sie unsicher sind, sagen Sie Y.

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

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

215 *If unsure, say Y.*

216 **1.11 General notification queue**

217 CONFIG_WATCH_QUEUE [=y] [N]

218 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-Änderungsbenachrichtigungen (key/keyring) und Gerätebenachrichtigungen verwendet werden.

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

223 *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.*

225 See Documentation/core-api/watch_queue.rst

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

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

228 **1.12 Enable process_vm_readv/writev syscalls**

229 CONFIG_CROSS_MEMORY_ATTACH [=y] [Y]

230 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.

233 *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.*

236 uselib syscall (for libc5 and earlier) bis 6.15

237 CONFIG_USELIB [=n] [N]

238 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) kompiliert wurden.

241

242 *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.*

245 Bemerkung: Debian Bullseye verwendet dies noch (Y).

246 **1.13 Auditing support**

247 CONFIG_AUDIT [=y] [Y]

248 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.

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

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

254 **1.14 IRQ subsystem →**

255 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.

258 *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.*

260 **1.14.1 Expose irq internals in debugfs**

261 CONFIG_GENERIC_IRQ_DEBUGFS [=n] []

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

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

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

267 **1.15 Timers subsystem →**

268 *Teilsystem Zeitgeber*

269 **1.15.1 Timer tick handling →**

270 Sie müssen aus den folgenden drei Möglichkeiten eine wählen:

271 **1.15.1.1 Periodic timer ticks (constant rate, no dynticks)**

272 CONFIG_HZ_PERIODIC [=n] [N]

273 Diese Option sorgt dafür, dass der Tick periodisch mit einer konstanten Rate läuft, auch wenn die CPU
274 ihn nicht braucht.

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

276 **1.15.1.2 Idle dynticks system (tickless idle)**

277 CONFIG_NO_HZ_IDLE [=n] [N]

278 Diese Option ermöglicht ein tickloses idle-System (Leerlaufsystem): Timer-Interrupts werden nur bei
279 Bedarf ausgelöst, wenn das System im Leerlauf ist. Dies ist v. a. zum Energiesparen interessant.

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

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

283 **1.15.1.3 Full dynticks system (tickless)**

284 CONFIG_NO_HZ_FULL [=y] [Y]

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

291 Standardmäßig, ohne Übergabe des nohz_full-Parameters, verhält sich dies genau wie NO_HZ_IDLE.
292 Wird bei Linux-Distributionen ausgewählt.

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

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

298 *By default, without passing the nohz_full parameter, this behaves just like NO_HZ_IDLE. If you're a distro say Y.*

299 **1.15.2 Force user context tracking**

300 CONFIG_CONTEXT_TRACKING_USER_FORCE [=n] [N]

301 Die wichtigste Voraussetzung für das Funktionieren von Full-Dynticks ist die Unterstützung des Subsystems
302 zur Verfolgung des Benutzerkontextes. Es gibt aber auch noch andere Abhängigkeiten, die erfüllt werden
303 müssen, damit die vollständigen Dynticks funktionieren.

304 Diese Option dient zum Testen, wenn eine Systemarchitektur das Backend für die Benutzerkontextver-
305 folgung implementiert, aber noch nicht alle Anforderungen erfüllt, um die volle Dynticks-Funktion zu
306 ermöglichen. Ohne die vollständigen Dynticks gibt es keine Möglichkeit, die Unterstützung für die Benutzer-
307 kontextverfolgung und die Teilsysteme, die darauf angewiesen sind, zu testen: RCU-Userspace-erweiterter
308 Ruhezustand und taktlose CPU-Zeitabrechnung. Diese Option kommt mit dem Fehlen des vollständigen
309 dynticks-Subsystems zurecht, indem sie die Benutzerkontextverfolgung auf allen CPUs im System erzwingt.

310 Sagen Sie nur dann ja (Y), wenn Sie an der Entwicklung eines Architektur-Backends für die Benutzer-
311 kontextverfolgung arbeiten. Sagen Sie ansonsten N, da diese Option einen Overhead mit sich bringt, den

312 Sie in der Praxis nicht haben wollen.
313 *The major pre-requirement for full dynticks to work is to support the user context tracking subsystem. But there*
314 *are also other dependencies to provide in order to make the full dynticks working.*
315 *This option stands for testing when an arch implements the user context tracking backend but doesn't yet fulfill all*
316 *the requirements to make the full dynticks feature working. Without the full dynticks, there is no way to test the*
317 *support for user context tracking and the subsystems that rely on it: RCU userspace extended quiescent state and*
318 *tickless cputime accounting. This option copes with the absence of the full dynticks subsystem by forcing the user*
319 *context tracking on all CPUs in the system.*
320 *Say Y only if you're working on the development of an architecture backend for the user context tracking.*
321 *Say N otherwise, this option brings an overhead that you don't want in production.*

322 1.15.3 Old Idle dynticks config

323 CONFIG_NO_HZ [=y] [N]
324 *Alte Leerlauf-Dynticks-Konfiguration*
325 Dies ist der alte Konfigurationseintrag, der Dynticks im Leerlauf aktiviert. ~~Wir behalten ihn noch eine~~
326 ~~Weile bei, um die Abwärtskompatibilität mit älteren Konfigurationsdateien zu gewährleisten.~~
327 *This is the old config entry that enables dynticks idle. We keep it around for a little while to enforce backward*
328 *compatibility with older config files.*
329 Alte Dynticks-Konfiguration wird nicht mehr unterstützt.

330 1.15.4 High Resolution Timer Support

331 CONFIG_HIGH_RES_TIMERS [=y] [Y]
332 *Unterstützung von Timern mit hoher Auflösung*
333 Diese Option aktiviert die Unterstützung hochauflösender Timer. Wenn ihre Hardware dazu nicht in der
334 Lage ist, erhöht diese Option nur die Größe des Kernel-Images.
335 *This option enables high resolution timer support. If your hardware is not capable then this option only increases*
336 *the size of the kernel image.*

337 1.15.5 Clocksource watchdog maximum allowable skew

338 CONFIG_CLOCKSOURCE_WATCHDOG_MAX_SKEW_US [=100] [100]
339 *Maximal zulässige Abweichung der Watchdog-Taktquelle*
340 Geben Sie den maximal zulässigen Wert für den Watchdog-Versatz in Mikrosekunden an, bevor die
341 Clocksource als instabil gemeldet wird. Der Standardwert basiert auf einem Watchdog-Intervall von
342 einer halben Sekunde und der maximalen Frequenzdrift von NTP von 500 Teilen pro Million. Wenn die
343 Clocksource gut genug für NTP ist, ist sie auch gut genug für den Watchdog der Clocksource!
344 Bereich (Range): 50 – 1000
345 *Specify the maximum amount of allowable watchdog skew in microseconds before reporting the clocksource to be*
346 *unstable. The default is based on a half-second clocksource watchdog interval and NTP's maximum frequency drift*
347 *of 500 parts per million. If the clocksource is good enough for NTP, it is good enough for the clocksource watchdog!*

348 1.15.6 Enable auxiliary POSIX clocks

349 CONFIG_POSIX_AUX_CLOCKS [=n] [N]
350 Hilfs-POSIX-Uhren sind Uhren, die unabhängig vom Kernzeitgeber gesteuert werden können, der die
351 MONOTONIC-, REALTIME-, BOOTTIME- und TAI-Uhren steuert. Sie sind nützlich, um beispielsweise
352 lockless time accessors für unabhängige PTP-Uhren und andere Uhrdomänen bereitzustellen, die nicht
353 mit dem TAI/NTP-Zeitbegriff korrelieren.
354 *Auxiliary POSIX clocks are clocks which can be steered independently of the core timekeeper, which controls the*
355 *MONOTONIC, REALTIME, BOOTTIME and TAI clocks. They are useful to provide e.g. lockless time accessors*
356 *to independent PTP clocks and other clock domains, which are not correlated to the TAI/NTP notion of time.*

357 1.16 BPF subsystem →

358 Berkeley Packet Filter, Firewall-Filtertechnik im Kernel

359 **1.16.1 Enable bpf() system call**

360 CONFIG_BPF_SYSCALL [=y] [Y]

361 Aktivieren Sie den Systemaufruf bpf(), der es ermöglicht, BPF-Programme und -Maps über Dateideskriptoren zu manipulieren.

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

364 **1.16.2 Enable BPF Just In Time compiler**

365 CONFIG_BPF_JIT [=y] [Y]

366 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.

369 Beachten Sie, dass ein Administrator diese Funktion durch Ändern aktivieren sollte:

370 /proc/sys/net/core/bpf_jit_enable
371 /proc/sys/net/core/bpf_jit_harden (optional)
372 /proc/sys/net/core/bpf_jit_kallsyms (optional)

373 *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.*

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

376 /proc/sys/net/core/bpf_jit_enable
377 /proc/sys/net/core/bpf_jit_harden (optional)
378 /proc/sys/net/core/bpf_jit_kallsyms (optional)

379 **1.16.2.1 Permanently enable BPF JIT and remove BPF interpreter**

380 CONFIG_BPF_JIT_ALWAYS_ON [=y] [Y]

381 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.

385 *Enables BPF JIT and removes BPF interpreter to avoid speculative execution of BPF instructions by the interpreter.*
386 *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.*

388 **1.16.3 Disable unprivileged BPF by default**

389 CONFIG_BPF_UNPRIV_DEFAULT_OFF [=y] [Y]

390 Deaktiviert die unprivilegierte BPF standardmäßig, indem der entsprechende Eintrag

391 /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).

394 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.

397 *Disables unprivileged BPF by default by setting the corresponding /proc/sys/kernel/unprivileged_bpf_disabled knob to 2. An admin can still reenable 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.*

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

402 **1.16.4 Preload BPF file system with kernel specific program and map iterators →**

403 CONFIG_BPF_PRELOAD [=n] [N]

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

407 *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.*

409 **1.16.5 Enable BPF LSM Instrumentation**

410 CONFIG_BPF_LSM [=y] [Y]

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

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

416 **1.16.5.1 bpf_preload kernel module**

417

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

419 CONFIG_BPF_PRELOAD_UMD [=m] [M]

420 Damit wird das Kernelmodul bpf_preload mit eingebetteten BPF-Programmen für die Introspektion in
421 bpffs erstellt.

422 *This builds bpf_preload kernel module with embedded BPF programs for introspection in bpffs.*

423 **1.16.6 Enable BPF LSM Instrumentation**

424 CONFIG_BPF_LSM [=y] [Y]

425 Ermöglicht die Instrumentierung der Sicherheitshaken mit BPF-Programmen zur Implementierung dyna-
426 mischer MAC- und Prüfungsrichtlinien.

427 Wenn Sie unsicher sind, wie Sie diese Frage beantworten sollten, antworten Sie mit N.

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

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

431 **1.17 Preemption Model (Scheduler controlled preemption model) →**

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

434 **1.17.1 No Forced Preemption (Server)**

435 CONFIG_PREEMPT_NONE [=n] [N]

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

439 Für einen Serverbetrieb wird diese Einstellung empfohlen, damit der maximale Durchsatz an Rechen-
440 leistung entsteht.

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

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

445 **1.17.2 Voluntary Kernel Preemption (Desktop)**

446 CONFIG_PREEMPT_VOLUNTARY [=n] [N]

447 Diese Einstellung reduziert die Latenz des Kernels durch zusätzliche „explizite Unterbrechungspunkte“, im
448 Kernel. Diese neuen Unterbrechungspunkte wurden ausgewählt, um die maximale Latenz beim neuerlichen
449 Zuordnen des Schedulers zu reduzieren und dadurch schnelle Reaktionszeiten der Applikationen zu
450 gewährleisten. – Auf Kosten eines geringeren Durchsatzes wird dies erreicht.

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

454 *This allows reaction to interactive events by allowing a low priority process to voluntarily preempt itself even if it is
455 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 geringerem 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-Modus, dehalb 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.

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

511 1.19 Preemption behaviour defined on boot

512 CONFIG_PREEMPT_DYNAMIC [=y] [N]

513 Diese Option ermöglicht es, das Präemptionsmodell über den Kernel-Kommandozeilenparameter zu
514 definieren und damit das während der Kompilierung definierte Standard-Präemptionsmodell außer Kraft
515 zu setzen. Diese Funktion ist vor allem für Linux-Distributionen interessant, die eine vorgefertigte Kernel-
516 Binärdatei bereitstellen, um die Anzahl der angebotenen Kernel-Varianten zu reduzieren und dennoch
517 verschiedene Anwendungsfälle zu ermöglichen.

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

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

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

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

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

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

530 1.20 Core Scheduling for SMT

531 CONFIG_SCHED_CORE [=y] [Y]

532 Kern-Scheduling für SMT

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

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

540 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.

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

545 *Use of this feature includes:*

- 546 - mitigation of some (not all) SMT side channels;
 - 547 - limiting SMT interference to improve determinism and/or performance.
- 548 *SCHED_CORE is default disabled. When it is enabled and unused, which is the likely usage by Linux distributions,
549 there should be no measurable impact on performance.*

550 1.21 Extensible Scheduling Class seit 6.12

551 CONFIG_SCHED_CLASS_EXT [=y] [Y]

552 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:

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

556 - Anpassungsfähigkeit: Erstellung von anwendungsspezifischen Schedulern, die Richtlinien implementieren, die für allgemeine Scheduler nicht anwendbar sind.
 557
 558 - Schnelle Scheduler-Implementierungen: Unterbrechungsfreie Auslagerung von Planungsrichtlinien in Produktionsumgebungen.
 559
 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.
 561
 562 Die `struct_ops`-Struktur, die von `sched_ext` exportierte Struktur heißt `struct sched_ext_ops` und ist konzeptionell ähnlich wie `struct sched_class`.
 563
 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:*
 568
 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.
 571
 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.*
 574
 575
 576 *For more information: Documentation/scheduler/sched-ext.rst <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.
 582

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.*
 585

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.
 591

592 Im Moment ist dies nur sinnvoll, wenn Sie an der Entwicklung des vollständigen Dynticks-Subsystems arbeiten.
 593

594 *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.*
 595

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

597 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.
 604
 605

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://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
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 sind.

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

678 In Kerneln 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 webservers, 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.*

707 1.23 CPU isolation

708 CONFIG_CPU_ISOLATION [=y] [Y]

709 Stellen Sie sicher, dass CPUs, auf denen kritische Aufgaben laufen, nicht durch irgendwelche „Störquellen“
710 wie ungebundene Workqueues, Timers, kthreads usw. gestört werden.

711 Ungebundene Aufgaben werden auf Housekeeping-CPUs verlagert. Dies wird durch den Boot-Parameter
712 isolcpus= gesteuert.

713 Sagen Sie Y für ja, wenn Sie unsicher sind.

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

717 1.24 RCU Subsystem →

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

719 1.24.1 Make expert-level adjustments to RCU configuration

720 CONFIG_RCU_EXPERT [=y] [N]

721 Diese Option muss aktiviert werden, wenn Sie Anpassungen der RCU-Konfiguration auf Expertenebene
722 vornehmen möchten. Standardmäßig können solche Anpassungen nicht vorgenommen werden, was den oft
723 vorteilhaften Nebeneffekt hat, dass „make oldconfig“ Sie davon abhält, alle möglichen detaillierten Fragen
724 darüber zu stellen, wie Sie zahlreiche obskure RCU-Optionen eingerichtet haben möchten.

725 Sagen Sie Y, wenn Sie Anpassungen an RCU auf Expertenebene vornehmen müssen.

726 Sagen Sie N, wenn Sie unsicher sind.

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

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

731 *Say N if you are unsure.*

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

733 1.24.2 Force selection of NEED_SRCU_NMI_SAFE seit 6.2

734 CONFIG_FORCE_NEED_SRCU_NMI_SAFE [=n] [N]

735 Diese Option erzwingt die Auswahl der NEED_SRCU_NMI_SAFE Kconfig-Option und ermöglicht
736 das Testen von `srcu_read_lock_nmisafe()` und `srcu_read_unlock_nmisafe()` auf Architekturen (wie
737 x86), die die ARCH_HAS_NMI_SAFE_THIS_CPU_OPS Kconfig-Option auswählen.

738 *This option forces selection of the NEED_SRCU_NMI_SAFE Kconfig option, allowing testing of
739 `srcu_read_lock_nmisafe()` and `srcu_read_unlock_nmisafe()` on architectures (like x86) that select the
740 ARCH_HAS_NMI_SAFE_THIS_CPU_OPS Kconfig option.*

741 1.24.3 Force selection of TASKS_RCU

742 CONFIG_FORCE_TASKS_RCU [=n] [N]

743 Diese Option erzwingt eine aufgabenbasierte RCU-Implementierung die nur freiwillige Kontextwechsel
744 verwendet (keine Preemption!), Leerlauf und Benutzermodus-Ausführung als Ruhezustände verwendet.
745 Nicht für manuelle Auswahl in den meisten Fällen.

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

748 1.24.4 Force selection of Tasks Rude RCU

749 CONFIG_FORCE_TASKS_RUDE_RCU [=n] [N]

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

754 *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-CPUs, 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 Produktionsysteme verwendet werden, aber wenn Sie die RCU-Implemen-
772 tierung selbst einem Stresstest unterziehen, ermöglichen kleinen 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 verzerrten, 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 Latenzen 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-

859 Workloads zu reduzieren. Sie kann auch verwendet werden, um RCU-Callback-Aufrufe auf energieeffiziente
 860 CPUs in batteriebetriebenen asymmetrischen Multiprozessoren auszulagern. Der Preis für diesen reduzierten
 861 Jitter ist, dass der Overhead von `call_rcu()` ansteigt und dass bei einigen Workloads ein erheblicher
 862 Anstieg der Kontextwechselraten zu verzeichnen ist.
 863 Diese Option entlastet den Aufruf von Callbacks von der Gruppe von CPUs, die zur Boot-Zeit durch den
 864 `rcu_nocbs`-Parameter angegeben wird. Für jede dieser CPUs wird ein kthread („`rcuox/N`“) erstellt, um
 865 Callbacks aufzurufen, wobei „N“ die CPU ist, die entlastet wird, und wobei „x“ „p“ für RCU-preempt
 866 (PREEMPTION-Kernel) und „s“ für RCU-sched (!PREEMPTION-Kernel) ist. Nichts hindert diesen
 867 kthread daran, auf den angegebenen CPUs zu laufen, aber (1) die kthreads können zwischen jedem
 868 Callback preempted werden, und (2) Affinität oder cgroups können verwendet werden, um die kthreads zu
 869 zwingen, auf jeder gewünschten Gruppe von CPUs zu laufen.
 870 Sagen Sie hier Y, wenn Sie trotz des zusätzlichen Overheads ein geringeres OS-Jitter benötigen.
 871 Sagen Sie hier N, wenn Sie unsicher sind.
 872 *Use this option to reduce OS jitter for aggressive HPC or real-time workloads. It can also be used to offload RCU
 873 callback invocation to energy-efficient CPUs in battery-powered asymmetric multiprocessors. The price of this
 874 reduced jitter is that the overhead of `call_rcu()` increases and that some workloads will incur significant increases
 875 in context-switch rates.*
 876 *This option offloads callback invocation from the set of CPUs specified at boot time by the `rcu_nocbs` parameter. For
 877 each such CPU, a kthread (“`rcuox/N`”) will be created to invoke callbacks, where the “N” is the CPU being offloaded,
 878 and where the “x” is “p” for RCU-preempt (PREEMPTION kernels) and “s” for RCU-sched (!PREEMPTION
 879 kernels). This option also creates another kthread for each $\sqrt{\text{nr_cpu_ids}}$ CPUs (“`rcuog/N`”, where N is the
 880 first CPU in that group to come online), which handles grace periods for its group. Nothing prevents these kthreads
 881 from running on the specified CPUs, but (1) the kthreads may be preempted between each callback, and (2) affinity
 882 or cgroups can be used to force the kthreads to run on whatever set of CPUs is desired.*
 883 *The $\sqrt{\text{nr_cpu_ids}}$ grouping may be overridden using the `rcutree.rcu_nocb_gp_stride` kernel boot parameter.
 884 This can be especially helpful for smaller numbers of CPUs, where $\sqrt{\text{nr_cpu_ids}}$ can be a bit of a blunt
 885 instrument.*
 886 *Say Y here if you need reduced OS jitter, despite added overhead. Say N here if you are unsure.*

887 **1.24.9.1 Offload RCU callback processing from all CPUs by default**
 888 `CONFIG_RCU_NOCB_CPU_DEFAULT_ALL [=n] [N]`
 889 Verwenden Sie diese Option, um die Callback-Verarbeitung standardmäßig von allen CPUs zu entlasten,
 890 wenn der Boot-Parameter `rcu_nocbs` oder `nohz_full` nicht vorhanden ist. Dadurch wird auch die Notwen-
 891 digkeit vermieden, Boot-Parameter zu verwenden, um den Effekt der Entlastung aller CPUs beim Booten
 892 zu erreichen.
 893 Geben Sie hier Y an, wenn Sie alle CPUs standardmäßig beim Booten entlasten wollen.
 894 Sagen Sie hier N, wenn Sie sich nicht sicher sind.
 895 *Use this option to offload callback processing from all CPUs by default, in the absence of the `rcu_nocbs` or `nohz_full`
 896 boot parameter. This also avoids the need to use any boot parameters to achieve the effect of offloading all CPUs
 897 on boot.*
 898 *Say Y here if you want offload all CPUs by default on boot. Say N here if you are unsure.*

899 **1.24.9.2 Offload RCU callback from real-time kthread**
 900 `CONFIG_RCU_NOCB_CPU_CB_BOOST [=n] [N]`
 901 Verwenden Sie diese Option, um ausgelagerte Rückrufe als `SCHED_FIFO` aufzurufen, um ein Aushungern
 902 durch schwere `SCHED_OTHER`-Hintergrundlast zu vermeiden. Natürlich führt die Ausführung als
 903 `SCHED_FIFO` während Callback Floods dazu, dass die `rcuo[ps]` kthreads die CPU für Hunderte von
 904 Millisekunden oder mehr monopolisieren. Wenn Sie diese Option aktivieren, müssen Sie daher sicherstellen,
 905 dass latenzempfindliche Aufgaben entweder mit höherer Priorität oder auf einer anderen CPU ausgeführt
 906 werden.
 907 Geben Sie hier Y an, wenn Sie die RT-Priorität für die Auslagerung von kthreads festlegen möchten.
 908 Sagen Sie hier N, wenn Sie einen !PREEMPT_RT-Kernel bauen und sich unsicher sind.
 909 *Use this option to invoke offloaded callbacks as `SCHED_FIFO` to avoid starvation by heavy `SCHED_OTHER`
 910 background load. Of course, running as `SCHED_FIFO` during callback floods will cause the `rcuo[ps]` kthreads
 911 to monopolize the CPU for hundreds of milliseconds or more. Therefore, when enabling this option, it is your
 912 responsibility to ensure that latency-sensitive tasks either run with higher priority or run on some other CPU.*
 913 *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.*

961 1.25 Kernel .config support

962 CONFIG_IKCONFIG [=y] [N]

963 Mit dieser Option kann der gesamte Inhalt der „config“-Datei des Linux-Kernels im Kernel gespeichert
964 werden. Sie dokumentiert, welche Kernel-Optionen in einem laufenden Kernel oder in einem On-Disk-
965 Kernel verwendet werden. Diese Informationen können mit dem Skript scripts/extract-ikconfig aus der
966 Kernel-Image-Datei extrahiert und als Eingabe verwendet werden, um den aktuellen Kernel neu zu
967 erstellen oder einen anderen Kernel zu bauen. Sie können auch aus einem laufenden Kernel extrahiert
968 werden, indem /proc/config.gz gelesen wird, falls dies aktiviert ist (siehe unten).

969

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

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

976 1.25.1 Enable access to .config through /proc/config.gz

977 CONFIG_IKCONFIG_PROC [=y] [Y]

978 Diese Option ermöglicht den Zugriff auf die Kernelkonfigurationsdatei über /proc/config.gz.

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

980 1.26 Enable kernel headers through /sys/kernel/kheaders.tar.xz

981 CONFIG_IKHEADERS [=m] [N]

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

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

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

990 1.27 Kernel log buffer size (16 ⇒ 64KB, 17 ⇒ 128KB)

991 CONFIG_LOG_BUF_SHIFT [=17] [17]

992 Wählen Sie die minimale Größe des Kernel-Protokollpuffers als eine Potenz von 2 aus. Die endgültige
993 Größe wird durch den Konfigurationsparameter LOG_CPU_MAX_BUF_SHIFT beeinflusst, siehe unten.
994 Eine höhere Größe kann auch durch den Boot-Parameter log_buf_len erzwungen werden.

995 Beispiele:

996 17 ⇒ 128 kB

997 16 ⇒ 64 kB

998 15 ⇒ 32 kB

999 14 ⇒ 16 kB

1000 13 ⇒ 8 kB

1001 12 ⇒ 4 kB

1002 Symbol: LOG_BUF_SHIFT

1003 Type: Integer (Ganzzahl)

1004 Bereich (range): [12 25]

1005 *Select the minimal kernel log buffer size as a power of 2.*

1006 *The final size is affected by LOG_CPU_MAX_BUF_SHIFT config parameter, see below. Any higher size also*
1007 *might be forced by “log_buf_len” boot parameter.*

1008 Examples:

1009 17 => 128 KB

1010 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.

1065 Add support for indexing of all printk formats known at compile time at <debugfs>/printk/index/<module>. 1066 This can be used as part of maintaining daemons which monitor /dev/kmsg, as it permits auditing the printk 1067 formats present in a kernel, allowing detection of cases where monitored printks are changed or no longer present. 1068 There is no additional runtime cost to printk with this enabled.

1069 Wie bei Debian Bookworm wird diese Indizierung ausgeschaltet.

1070 1.30 Scheduler features →

1071 (*Scheduler-Funktionen*)

1072 1.30.1 Enable utilization clamping for RT/FAIR tasks

1073 CONFIG_UCLAMP_TASK [=y] [N]

1074 Diese Funktion ermöglicht es dem Scheduler, die geklemmte Auslastung jeder CPU auf der Grundlage 1075 der auf dieser CPU geplanten RUNNABLE-Tasks zu verfolgen. Mit dieser Option kann der Benutzer 1076 die minimale und maximale CPU-Auslastung angeben, die für RUNNABLE-Aufgaben zulässig ist. Die 1077 maximale Auslastung definiert die maximale Häufigkeit, mit der ein Task laufen soll, während die minimale 1078 Auslastung die minimale Häufigkeit definiert, mit der er laufen soll.

1079 Sowohl die Minimal- als auch die Maximalwerte für die Auslastung sind Hinweise für den Scheduler, um 1080 seine Frequenzauswahl zu verbessern, aber sie erzwingen oder gewähren keine bestimmte Bandbreite für 1081 Tasks.

1082 Im Zweifelsfall sagen Sie N für Nein.

1083 *This feature enables the scheduler to track the clamped utilization of each CPU based on RUNNABLE tasks scheduled on that CPU.*

1084 *With this option, the user can specify the min and max CPU utilization allowed for RUNNABLE tasks. The 1085 max utilization defines the maximum frequency a task should use while the min utilization defines the minimum 1086 frequency it should use.*

1087 *Both min and max utilization clamp values are hints to the scheduler, aiming at improving its frequency selection 1088 policy, but they do not enforce or grant any specific bandwidth for tasks.*

1089 *If in doubt, say N.*

1091 Wie bei Debian Bookworm und WSL2 wird dies ausgeschaltet.

1092 1.30.1.1 Number of supported utilization clamp buckets

1093 CONFIG_UCLAMP_BUCKETS_COUNT [=5] [5]

1094 Legt die Anzahl der zu verwendenden Klammerbereiche fest. Der Bereich der einzelnen Buckets ist 1095 SCHED_CAPACITY_SCALE/UCLAMP_BUCKETS_COUNT. Je höher die Anzahl der Clamp-Buckets, 1096 desto feiner die Granularität und desto höher die Präzision der Clamp-Aggregation und -Verfolgung 1097 während der Laufzeit. Mit dem minimalen Konfigurationswert haben wir beispielsweise 5 Clamp-Buckets, 1098 die jeweils 20 % Auslastung verfolgen. Eine um 25 % gesteigerte Aufgabe wird im Bucket [20..39] % gezählt 1099 und setzt den effektiven Wert der Bucketklemme auf 25 %. Wenn eine zweite, um 30 % erhöhte Aufgabe 1100 auf derselben CPU eingeplant wird, wird diese Aufgabe im selben Bucket wie die erste Aufgabe gezählt 1101 und erhöht den effektiven Bucket-Clamp-Wert auf 30 %. Der effektive Klemmwert eines Bereichs wird 1102 auf seinen Nennwert (20 % im obigen Beispiel) zurückgesetzt, wenn keine weiteren Aufgaben mehr in 1103 diesem Bereich gezählt werden. Bei einigen Aufgaben kann eine zusätzliche Verstärkungs-/Kappungsmarge 1104 hinzugefügt werden. Im obigen Beispiel wird die 25 %-Aufgabe auf 30 % angehoben, bis sie die CPU 1105 verlässt. Sollte dies auf bestimmten Systemen nicht akzeptabel sein, ist es immer möglich, den Spielraum 1106 zu verringern, indem die Anzahl der Clamp-Buckets erhöht wird, um den verbrauchten Speicher gegen die 1107 Genauigkeit der Laufzeitverfolgung einzutauschen.

1108 Im Zweifelsfall sollten Sie den Standardwert verwenden.

1109 *Defines the number of clamp buckets to use.*

1110 *The range of each bucket will be SCHED_CAPACITY_SCALE/UCLAMP_BUCKETS_COUNT. The higher 1111 the number of clamp buckets the finer their granularity and the higher the precision of clamping aggregation and 1112 tracking at run-time.*

1113 *For example, with the minimum configuration value we will have 5 clamp buckets tracking 20% utilization each. A 1114 25% boosted tasks will be refcounted in the [20..39] % bucket and will set the bucket clamp effective value to 25%.*

1115 *If a second 30% boosted task should be co-scheduled on the same CPU, that task will be refcounted in the same 1116 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 placemnent

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 Cupsets, 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 Cupsets, 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.

1163 **1.32.2 Memory controller**

1164 CONFIG_MEMCG [=y] [Y]

1165 Ermöglicht die Kontrolle über den Speicherbedarf von Tasks in einer cgroup.

1166 *Provides control over the memory footprint of tasks in a cgroup.*

1167 **1.32.2.1 Legacy cgroup v1 memory controller** seit 6.11

1168 CONFIG_MEMCG_V1 [=n] [N]

1169 Legacy cgroup v1 memory controller, der durch die cgroup v2 Implementierung veraltet ist. Der v1 ist für
1170 ältere Anwendungen gedacht, die noch nicht auf die neue cgroup v2-Schnittstelle umgestellt wurden. Wenn
1171 Sie keine solche Anwendung haben, ist es völlig in Ordnung, wenn Sie diese Option deaktiviert lassen.
1172 Bitte beachten Sie, dass der Funktionsumfang des Legacy-Speicher-Controllers aufgrund der Abkündigung
1173 wahrscheinlich schrumpfen wird. Von neuen Implementierungen mit v1-Controller wird dringend abgeraten.

1174 *Legacy cgroup v1 memory controller which has been deprecated by cgroup v2 implementation. The v1 is there
1175 for legacy applications which haven't migrated to the new cgroup v2 interface yet. If you do not have any such
1176 application then you are completely fine leaving this option disabled.*

1177 *Please note that feature set of the legacy memory controller is likely going to shrink due to deprecation process.
1178 New deployments with v1 controller are highly discouraged.*

1179 **1.32.3 IO controller**

1180 CONFIG_BLK_CGROUP [=y] [Y]

1181 Generische Block IO Controller cgroup Schnittstelle. Dies ist die gemeinsame cgroup-Schnittstelle, die von
1182 verschiedenen IO-Kontrollstrategien verwendet werden sollte.

1183 Derzeit wird sie vom CFQ IO Scheduler zur Erkennung von Task-Gruppen und zur Steuerung der Zuwei-
1184 sung von Festplattenbandbreite (proportionale Zeitscheibenzuweisung) an solche Task-Gruppen verwendet.
1185 Sie wird auch von der Bio-Throttling-Logik in der Blockschicht verwendet, um eine Obergrenze für die IO-
1186 Raten auf einem Gerät einzuführen.

1187 Diese Option aktiviert nur die generische Infrastruktur des Block-IO-Controllers. Man muss auch die tat-
1188 sächliche IO-Kontrolllogik/-Politik aktivieren. Um die proportionale Aufteilung der Festplattenbandbreite
1189 in CFQ zu aktivieren, setzen Sie CONFIG_BFQ_GROUP_IOSCHED=y; für die Aktivierung der Drosselungspoli-
1190 tik setzen Sie CONFIG_BLK_DEV_THROTTLING=y.

1191 Weitere Informationen finden Sie unter Documentation/admin-guide/cgroup-v1/blkio-controller.rst.

1192 *Generic block IO controller cgroup interface. This is the common cgroup interface which should be used by various
1193 IO controlling policies.*

1194 *Currently, CFQ IO scheduler uses it to recognize task groups and control disk bandwidth allocation (proportional
1195 time slice allocation) to such task groups. It is also used by bio throttling logic in block layer to implement upper
1196 limit in IO rates on a device.*

1197 *This option only enables generic Block IO controller infrastructure. One needs to also enable actual IO controlling
1198 logic/policy.*

1199 *For enabling proportional weight division of disk bandwidth in CFQ, set CONFIG_BFQ_GROUP_IOSCHED=y;
1200 for enabling throttling policy, set CONFIG_BLK_DEV_THROTTLING=y.*

1201 *See Documentation/admin-guide/cgroup-v1/blkio-controller.rst for more information.*

1202 **1.32.4 CPU controller →**

1203 CONFIG_CGROUP_SCHED [=y] [Y]

1204 Diese Funktion ermöglicht es dem CPU-Scheduler, Task-Gruppen zu erkennen und die Zuweisung von
1205 CPU-Bandbreite an solche Task-Gruppen zu steuern. Er verwendet cgroups, um Tasks zu gruppieren.

1206 *This feature lets CPU scheduler recognize task groups and control CPU bandwidth allocation to such task groups. It
1207 uses cgroups to group tasks.*

1208 **1.32.4.1 Group scheduling for SCHED_OTHER**

1209 CONFIG_FAIR_GROUP_SCHED [=y] [Y]

1210 *Für diese Option gibt es keine Hilfe.*

1211 **1.32.4.1.1 CPU bandwidth provisioning for FAIR_GROUP_SCHED**

1212 CONFIG_CFS_BANDWIDTH [=y] [Y]

1213 Mit dieser Option können Benutzer CPU-Bandbreitenraten (Limits) für Aufgaben festlegen, die innerhalb
1214 des Fair Group Schedulers laufen. Gruppen, für die kein Limit festgelegt wurde, gelten als uneingeschränkt
1215 und werden ohne Einschränkung ausgeführt.

1216 Weitere Informationen finden Sie unter Documentation/scheduler/sched-bwc.rst.

1217 *This option allows users to define CPU bandwidth rates (limits) for tasks running within the fair group sched-
1218 uler. Groups with no limit set are considered to be unconstrained and will run with no restriction. See
1219 Documentation/scheduler/sched-bwc.rst for more information.*

1220 1.32.4.2 Group scheduling for SCHED_RR/FIFO

1221 CONFIG_RT_GROUP_SCHED [=n] [N]

1222 Mit dieser Funktion können Sie den Task-Gruppen explizit echte CPU-Bandbreite zuweisen. Wenn sie
1223 aktiviert ist, wird es auch unmöglich, Echtzeitaufgaben für Nicht-Root-Benutzer zu planen, bis Sie ihnen
1224 Echtzeitbandbreite zuweisen.

1225 Weitere Informationen finden Sie unter Documentation/scheduler/sched-rt-group.rst.

1226 *This feature lets you explicitly allocate real CPU bandwidth to task groups. If enabled, it will also make it
1227 impossible to schedule realtime tasks for non-root users until you allocate realtime bandwidth for them. See
1228 Documentation/scheduler/sched-rt-group.rst for more information.*

1229 1.32.5 Utilization clamping per group of tasks

1230 CONFIG_UCLAMP_TASK_GROUP [=y] [Y]

1231 Mit dieser Funktion kann der Scheduler die geklemmte Auslastung jeder CPU auf der Grundlage der
1232 RUNNABLE-Tasks, die derzeit auf dieser CPU geplant sind, verfolgen. Wenn diese Option aktiviert ist,
1233 kann der Benutzer eine minimale und maximale CPU-Bandbreite angeben, die für jede einzelne Aufgabe
1234 in einer Gruppe zulässig ist. Mit der maximalen Bandbreite kann die maximale Frequenz, die ein Task
1235 verwenden kann, festgelegt werden, während mit der minimalen Bandbreite eine minimale Frequenz
1236 festgelegt werden kann, die ein Task immer verwenden wird. Bei aktiverter aufgabengruppenbasierter
1237 Auslastungsbegrenzung wird ein eventuell angegebener aufgabenspezifischer Begrenzungswert durch den
1238 von cgroup angegebenen Begrenzungswert eingeschränkt. Sowohl die minimale als auch die maximale Task-
1239 Klemmung kann nicht größer sein als die entsprechende auf Task-Gruppen-Ebene definierte Klemmung.
1240 Im Zweifelsfall sagen Sie N.

1241 *This feature enables the scheduler to track the clamped utilization of each CPU based on RUNNABLE tasks currently
1242 scheduled on that CPU.*

1243 *When this option is enabled, the user can specify a min and max CPU bandwidth which is allowed for each single
1244 task in a group. The max bandwidth allows to clamp the maximum frequency a task can use, while the min
1245 bandwidth allows to define a minimum frequency a task will always use.*

1246 *When task group based utilization clamping is enabled, an eventually specified task-specific clamp value is constrained
1247 by the cgroup specified clamp value. Both minimum and maximum task clamping cannot be bigger than the
1248 corresponding clamping defined at task group level.*

1249 *If in doubt, say N.*

1250 Wird nur in der „Control Group support“-Gruppe angezeigt, wenn CGROUP_SCHED (CPU Controller →) und UCLAMP_TASK
1251 (Enable utilization clamping for RT/FAIR tasks) gesetzt sind.

1252 1.32.6 PIDs controller

1253 CONFIG_CGROUP_PIDS [=y] [Y]

1254 Erzwingt die Begrenzung der Prozessanzahl im Bereich einer cgroup. Jeder Versuch, mehr Prozesse zu
1255 forken, als in der cgroup erlaubt sind, schlägt fehl. PIDs sind grundsätzlich eine globale Ressource, da es
1256 ziemlich trivial ist, eine PID-Erschöpfung zu erreichen, bevor man auch nur eine konservative kmemcg-
1257 Grenze erreicht. Infolgedessen ist es möglich, ein System zum Stillstand zu bringen, ohne durch andere
1258 cgroup-Richtlinien eingeschränkt zu werden. Der PID-Regler ist dafür ausgelegt, dies zu verhindern. Es sollte
1259 beachtet werden, dass organisatorische Operationen (wie z. B. das Anhängen an eine cgroup-Hierarchie)
1260 nicht durch den PIDs-Controller blockiert werden, da das PIDs-Limit nur die Fähigkeit eines Prozesses
1261 zum Forking, nicht aber zum Anhängen an eine cgroup beeinflusst.

1262 *Provides enforcement of process number limits in the scope of a cgroup. Any attempt to fork more processes than is
1263 allowed in the cgroup will fail. PIDs are fundamentally a global resource because it is fairly trivial to reach PID
1264 exhaustion before you reach even a conservative kmemcg limit. As a result, it is possible to grind a system to halt
1265 without being limited by other cgroup policies. The PIDs controller is designed to stop this from happening.*

1266 It should be noted that organisational operations (such as attaching to a cgroup hierarchy) will *not* be blocked by
1267 the PIDs controller, since the PIDs limit only affects a process's ability to fork, not to attach to a cgroup.

1268 1.32.7 RDMA controller

1269 CONFIG_CGROUP_RDMA [=y] [Y]

1270 Ermöglicht die Durchsetzung der vom IB-Stack definierten RDMA-Ressourcen. Es ist relativ einfach
1271 für Verbraucher, RDMA-Ressourcen zu erschöpfen, was dazu führen kann, dass Ressourcen für andere
1272 Verbraucher nicht mehr verfügbar sind. Der RDMA-Controller ist dafür ausgelegt, dies zu verhindern.
1273 Das Anhängen von Prozessen mit aktiven RDMA-Ressourcen an die cgroup-Hierarchie ist erlaubt, auch
1274 wenn die Grenze der Hierarchie überschritten werden kann.

1275 Provides enforcement of RDMA resources defined by IB stack. It is fairly easy for consumers to exhaust RDMA
1276 resources, which can result into resource unavailability to other consumers. RDMA controller is designed to stop
1277 this from happening. Attaching processes with active RDMA resources to the cgroup hierarchy is allowed even if
1278 can cross the hierarchy's limit.

1279 1.32.8 Device memory controller (DMEM)

1280 CONFIG_CGROUP_DMEM [=y] [Y]

1281 Der DMEM-Controller ermöglicht es kompatiblen Geräten, die Nutzung des Gerätespeichers auf der
1282 Grundlage der cgroup-Hierarchie einzuschränken.

1283 So können Sie zum Beispiel die VRAM-Nutzung für Anwendungen im DRM-Subsystem einschränken.

1284 The DMEM controller allows compatible devices to restrict device memory usage based on the cgroup hierarchy.
1285 As an example, it allows you to restrict VRAM usage for applications in the DRM subsystem.

1286 1.32.9 Freezer controller

1287 CONFIG_CGROUP_FREEZER [=y] [Y]

1288 Ermöglicht das Einfrieren und Aufheben des Einfrierens aller Aufgaben in einer C-Group. Diese Option
1289 betrifft die ORIGINAL cgroup-Schnittstelle. Der cgroup2-Speicher-Controller enthält standardmäßig
1290 wichtige In-Kernel-Speicherverbraucher.

1291 Wenn Sie cgroup2 verwenden, sagen Sie N.

1292 Provides a way to freeze and unfreeze all tasks in a cgroup.

1293 This option affects the ORIGINAL cgroup interface. The cgroup2 memory controller includes important in-kernel
1294 memory consumers per default.

1295 If you're using cgroup2, say N.

1296 1.32.10 HugeTLB controller

1297 CONFIG_CGROUP_HUGETLB [=y] [Y]

1298 Bietet eine cgroup-Steuerung für HugeTLB-Seiten. Wenn Sie dies aktivieren, können Sie die HugeTLB-
1299 Nutzung pro cgroup begrenzen. Die Begrenzung wird während eines Seitenfehlers durchgesetzt. Da
1300 HugeTLB keine Seitenrückforderung unterstützt, bedeutet die Durchsetzung des Limits zum Zeitpunkt des
1301 Seitenfehlers, dass die Anwendung ein SIGBUS-Signal erhält, wenn sie versucht, über das Limit hinaus auf
1302 HugeTLB-Seiten zuzugreifen. Dies setzt voraus, dass die Anwendung im Voraus weiß, wie viele HugeTLB-
1303 Seiten sie für ihre Nutzung benötigt. Die Kontrollgruppe wird im dritten Page-lru-Zeiger verfolgt. Dies
1304 bedeutet, dass wir die Steuergruppe nicht mit einer riesigen Seite von weniger als 3 Seiten verwenden
1305 können.

1306 Provides a cgroup controller for HugeTLB pages. When you enable this, you can put a per cgroup limit on HugeTLB
1307 usage. The limit is enforced during page fault. Since HugeTLB doesn't support page reclaim, enforcing the limit at
1308 page fault time implies that, the application will get SIGBUS signal if it tries to access HugeTLB pages beyond its
1309 limit. This requires the application to know beforehand how much HugeTLB pages it would require for its use. The
1310 control group is tracked in the third page lru pointer. This means that we cannot use the controller with huge page
1311 less than 3 pages.

1312 1.32.11 Cpuset controller

1313 CONFIG_CPUSETS [=y] [Y]

1314 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_CPUSETS_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

1363 Beispiel werden Programme, die mit BPF_CGROUP_INET_INGRESS angehängt werden, auf dem
1364 Ingress-Pfad von inet-Sockets ausgeführt.

1365 *Allow attaching eBPF programs to a cgroup using the bpf(2) syscall command BPF_PROG_ATTACH.*

1366 *In which context these programs are accessed depends on the type of attachment. For instance, programs that are
1367 attached using BPF_CGROUP_INET_INGRESS will be executed on the ingress path of inet sockets.*

1368 1.32.16 Misc resource controller

1369 CONFIG_CGROUP_MISC [=y] [Y]

1370 Bietet einen Controller für verschiedene Ressourcen auf einem Host. Verschiedene skalare Ressourcen
1371 sind die Ressourcen auf dem Host-System, die nicht wie die anderen cgroups abstrahiert werden können.
1372 Dieser Controller verfolgt und begrenzt die verschiedenen Ressourcen, die von einem Prozess verwendet
1373 werden, der an eine cgroup-Hierarchie angeschlossen ist. Weitere Informationen finden Sie im Abschnitt
1374 misc cgroup in /Documentation/admin-guide/cgroup-v2.rst.

1375 *Provides a controller for miscellaneous resources on a host.*

1376 *Miscellaneous scalar resources are the resources on the host system which cannot be abstracted like the other cgroups.*

1377 *This controller tracks and limits the miscellaneous resources used by a process attached to a cgroup hierarchy.*

1378 *For more information, please check misc cgroup section in /Documentation/admin-guide/cgroup-v2.rst.*

1379 1.32.17 Debug controller

1380 CONFIG_CGROUP_DEBUG [=n] [N]

1381 Diese Option aktiviert einen einfachen Controller, der Debugging-Informationen über das cgroups-Framework
1382 exportiert. Dieser Controller ist nur für das Debugging von Kontroll-C-Gruppen gedacht. Seine
1383 Schnittstellen sind nicht stabil.

1384 Sagen Sie N.

1385 *This option enables a simple controller that exports debugging information about the cgroups framework. This
1386 controller is for control cgroup debugging only. Its interfaces are not stable.*

1387 *Say N.*

1388 1.33 Namespaces support →

1389 CONFIG_NAMESPACES [=y] [Y]

1390 (Unterstützung von Namensräumen, namespaces)

1391 Bietet die Möglichkeit, Aufgaben mit verschiedenen Objekten unter Verwendung derselben Kennung
1392 arbeiten zu lassen. Zum Beispiel kann sich dieselbe IPC-ID auf verschiedene Objekte beziehen oder
1393 dieselbe Benutzer-ID oder pid kann sich auf verschiedene Aufgaben beziehen, wenn sie in verschiedenen
1394 Namensräumen verwendet werden.

1395 *Provides the way to make tasks work with different objects using the same id. For example same IPC id may refer
1396 to different objects or same user id or pid may refer to different tasks when used in different namespaces.*

1397 1.33.1 UTS namespace

1398 CONFIG_UTS_NS [=y] [Y]

1399 In diesem Namensraum sehen Aufgaben verschiedene Informationen, die mit dem Systemaufruf `uname()`
1400 bereitgestellt werden.

1401 *In this namespace tasks see different info provided with the uname() system call*

1402 1.33.2 TIME namespace

1403 CONFIG_TIME_NS [=y] [Y]

1404 In diesem Namespace können boottime und monotone Uhren eingestellt werden. Die Zeit läuft dann mit
1405 der gleichen Geschwindigkeit weiter.

1406 *In this namespace boottime and monotonic clocks can be set. The time will keep going with the same pace.*

1407 1.33.3 IPC namespace

1408 CONFIG_IPC_NS [=y] [Y]

1409 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. vservers, 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_userns_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_userns_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.*

1456 **1.35 Automatic process group scheduling**

1457 CONFIG_SCHED_AUTOGROUP [=y] [Y]

1458 Mit dieser Option wird der Scheduler für gängige Desktop-Workloads optimiert, indem automatisch
1459 Aufgabengruppen erstellt und aufgefüllt werden. Diese Trennung von Arbeitslasten isoliert aggressive CPU-
1460 Brenner (wie Build-Jobs) von Desktop-Anwendungen. Die automatische Erstellung von Aufgabengruppen
1461 basiert derzeit auf der Aufgabensitzung.

1462 *This option optimizes the scheduler for common desktop workloads by automatically creating and populating task
1463 groups. This separation of workloads isolates aggressive CPU burners (like build jobs) from desktop applications.
1464 Task group autogeneration is currently based upon task session.*

1465 **1.36 Kernel→user space relay support (formerly relayfs)**

1466 CONFIG_RELAY [=y] [Y]

1467 Diese Option aktiviert die Unterstützung für die Relaischnittstelle in bestimmten Dateisystemen (wie
1468 debugfs). Sie wurde entwickelt, um einen effizienten Mechanismus für Werkzeuge und Einrichtungen zur
1469 Weiterleitung großer Datenmengen aus dem Kernelbereich in den Benutzerbereich bereitzustellen.
1470 Wenn Sie unsicher sind, sagen Sie N.

1471 *This option enables support for relay interface support in certain file systems (such as debugfs). It is designed to
1472 provide an efficient mechanism for tools and facilities to relay large amounts of data from kernel space to user
1473 space.*

1474 *If unsure, say N.*

1475 **1.37 Initial RAM filesystem and RAM disk (initramfs/initrd) support**

1476 CONFIG_BLK_DEV_INITRD [=y] [Y]

1477 Das anfängliche RAM-Dateisystem ist ein ramfs, das vom Bootloader (loadlin oder lilo) geladen und vor
1478 dem normalen Bootvorgang als root eingehängt wird. Es wird typischerweise verwendet, um Module zu
1479 laden, die zum Einhängen des „echten“ Root-Dateisystems benötigt werden, usw.

1480 Siehe <file:Documentation/admin-guide/initrd.rst> für Details. Wenn die RAM-Disk-Unterstützung
1481 (BLK_DEV_RAM) ebenfalls enthalten ist, aktiviert dies auch die anfängliche RAM-Disk-Unterstützung
1482 (initrd) und fügt 15 kB (auf einigen anderen Architekturen mehr) zur Kernelgröße hinzu.
1483 Wenn Sie unsicher sind, sagen Sie Y.

1484 *The initial RAM filesystem is a ramfs which is loaded by the boot loader (loadlin or lilo) and that is mounted as root
1485 before the normal boot procedure. It is typically used to load modules needed to mount the “real” root file system,
1486 etc. See <file:Documentation/admin-guide/initrd.rst> for details.*

1487 *If RAM disk support (BLK_DEV_RAM) is also included, this also enables initial RAM disk (initrd) support and
1488 adds 15 Kbytes (more on some other architectures) to the kernel size.*

1489 *If unsure say Y.*

1490 **1.37.1 Initramfs source file(s)**

1491 CONFIG_INITRAMFS_SOURCE [=] []

1492 Dies kann entweder ein einzelnes cpio-Archiv mit der Endung .cpio oder eine durch Leerzeichen getrennte
1493 Liste von Verzeichnissen und Dateien zur Erstellung des initramfs-Abbilds sein. Ein cpio-Archiv sollte ein
1494 Dateisystemarchiv enthalten, das als initramfs-Abbild verwendet werden soll. Verzeichnisse sollten ein
1495 Dateisystem-Layout enthalten, das in das initramfs-Abbild aufgenommen werden soll. Die Dateien sollten
1496 Einträge in dem Format enthalten, das vom Programm `usr/gen_init_cpio` im Kernelbaum beschrieben
1497 wird. Wenn mehrere Verzeichnisse und Dateien angegeben werden, wird das initramfs-Abbild die Summe
1498 aller dieser Verzeichnisse und Dateien sein.

1499 Siehe <file:Documentation/driver-api/early-userspace/early_userspace_support.rst> für weitere Details.

1500 Wenn Sie sich nicht sicher sind, lassen Sie das Feld leer.

1501 Symbol: INITRAMFS_SOURCE [=]

1502 Type : string (Zeichenkette)

1503 *This can be either a single cpio archive with a .cpio suffix or a space-separated list of directories and files for
1504 building the initramfs image. A cpio archive should contain a filesystem archive to be used as an initramfs image.
1505 Directories should contain a filesystem layout to be included in the initramfs image. Files should contain entries
1506 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.

1550 **1.37.8 Support initial ramdisk/ramfs compressed using ZSTD**

1551 CONFIG_RD_ZSTD [=y] [Y]

1552 Unterstützung des Ladens eines ZSTD-kodierten Anfangs-Ramdisk- oder Cpio-Puffers.

1553 Wenn Sie unsicher sind, sagen Sie Y.

1554 *Support loading of a ZSTD encoded initial ramdisk or cpio buffer.*

1555 *If unsure, say N.*

1556 **1.38 Boot config support**

1557 CONFIG_BOOT_CONFIG [=y] [Y]

1558 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.

1561 Siehe <file:Documentation/admin-guide/bootconfig.rst> für Details.

1562 Wenn Sie unsicher sind, sagen Sie Y.

1563 *Extra boot config allows system admin to pass a config file as complementary 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.*

1566 **1.38.1 Force unconditional bootconfig processing**

1567 CONFIG_BOOT_CONFIG_FORCE [=n] [N]

1568 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.

1572 Wenn Sie unsicher sind, sagen Sie N.

1573 *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.*

1576 *If unsure, say N.*

1577 **1.38.2 Embed bootconfig file in the kernel**

1578 CONFIG_BOOT_CONFIG_EMBED [=n] [N]

1579 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.

1582 Wenn Sie unsicher sind, sagen Sie N.

1583 *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.*

1586 *If unsure, say N*

1587 **1.39 Preserve cpio archive mtimes in initramfs**

1588 CONFIG_INITRAMFS_PRESERVE_MTIME [=y] [Y]

1589 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.

1592 Wenn Sie unsicher sind, sagen Sie Y.

1593 *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.*

1595 *If unsure, say Y.*

1596 **1.40 Compiler optimization level (Optimize for performance) →**

1597 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-Wrapper aktiviert.

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

1631 1.42.2 Multiple users, groups and capabilities support

1632 CONFIG_MULTIUSER [=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.

1643 **1.42.3 sgetmask/ssetmask syscalls support**

1644 **CONFIG_SGETMASK_SYSCALL [=y] [Y]**

1645 *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.*

1647 *Wenn Sie unsicher sind, lassen Sie die Standardoption hier.*

1648 *sys_sgetmask and sys_ssetmask are obsolete system calls no longer supported in libc but still enabled by default in some architectures.*

1650 *If unsure, leave the default option here.*

1651 Eventuell könnten wir dies auf N stellen.

1652 **1.42.4 open by fhandle syscalls**

1653 **CONFIG_FHANDLE [=y] [Y]**

1654 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.

1659 *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.*

1663 **1.42.5 Posix Clocks & timers**

1664 **CONFIG_POSIX_TIMERS [=y] [Y]**

1665 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.

1668 Wenn diese Option deaktiviert ist, sind die folgenden Syscalls nicht verfügbar:

1669 `timer_create`, `timer_gettime`: `timer_getoverrun`, `timer_settime`, `timer_delete`, `clock_adjtime`,
1670 `getitimer`, `setitimer`, `alarm`.

1671 Außerdem werden die Syscalls `clock_settime`, `clock_gettime`, `clock_getres` und `clock_nanosleep`
1672 nur auf `CLOCK_REALTIME`, `CLOCK_MONOTONIC` und `CLOCK_BOOTTIME` beschränkt.

1673 Wenn Sie unsicher sind, sagen Sie y.

1674 *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.*

1678 *Furthermore, the clock_settime, clock_gettime, clock_getres and clock_nanosleep syscalls will be limited to CLOCK_REALTIME, CLOCK_MONOTONIC and CLOCK_BOOTTIME only.*

1680 *If unsure say y.*

1681 **1.42.6 Enable support for printk**

1682 **CONFIG_PRINTK [=y] [Y]**

1683 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.

1685 Da dies die Diagnose von Systemproblemen sehr erschwert, wird von der Angabe von N hier dringend abgeraten.

1687 *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.*

1690 **1.42.7 BUG() support**

1691 CONFIG_BUG [=y] [Y]

1692 Durch die Deaktivierung dieser Option wird die Unterstützung für BUG und WARN eliminiert, wodurch
1693 die Größe Ihres Kernel-Images verringert und zahlreiche fatale Zustände möglicherweise stillschweigend
1694 ignoriert werden. Sie sollten die Deaktivierung dieser Option nur für eingebettete Systeme in Erwägung
1695 ziehen, die keine Möglichkeit haben, Fehler zu melden.

1696 Sagen Sie einfach Y.

1697 *Disabling this option eliminates support for BUG and WARN, reducing the size of your kernel image and potentially
1698 quietly ignoring numerous fatal conditions. You should only consider disabling this option for embedded systems
1699 with no facilities for reporting errors.*

1700 *Just say Y.*

1701 **1.42.8 Enable ELF core dumps**

1702 CONFIG_ELF_CORE [=y] [Y]

1703 Aktivieren Sie die Unterstützung für die Erzeugung von Kerneldumps. Die Deaktivierung spart etwa 4 kB.

1704 *Enable support for generating core dumps. Disabling saves about 4k.*

1705 **1.42.9 Enable PC-Speaker support**

1706 CONFIG_PCSPKR_PLATFORM [=y] [N]

1707 Mit dieser Option können Sie die Unterstützung des internen PC-Lautsprechers deaktivieren, um Spei-
1708 cherplatz zu sparen.

1709 *This option allows to disable the internal PC speaker support, saving some memory.*

1710 **1.42.10 Enable smaller-sized data structures for core**

1711 CONFIG_BASE_SMALL [=n] [N]

1712 Wenn Sie diese Option aktivieren, wird die Größe verschiedener Kernel-Datenstrukturen reduziert. Dies
1713 spart Speicher auf kleinen Rechnern, kann aber die Leistung verringern.

1714 *Enabling this option reduces the size of miscellaneous core kernel data structures. This saves memory on small
1715 machines, but may reduce performance.*

1716 **1.42.11 Enable futex support**

1717 CONFIG_FUTEX [=y] [Y]

1718 Die Deaktivierung dieser Option führt dazu, dass der Kernel ohne Unterstützung für „schnelle User-
1719 space-Mutexe“ gebaut wird. Der resultierende Kernel führt möglicherweise glibc-basierte Anwendungen
1720 nicht korrekt aus.

1721 *Disabling this option will cause the kernel to be built without support for “fast userspace mutexes”. The resulting
1722 kernel may not run glibc-based applications correctly.*

1723 **1.42.12 Enable eventpoll support**

1724 CONFIG_EPOLL [=y] [Y]

1725 Die Deaktivierung dieser Option führt dazu, dass der Kernel ohne Unterstützung für die epoll-Familie von
1726 Systemaufrufen gebaut wird.

1727 *Disabling this option will cause the kernel to be built without support for epoll family of system calls.*

1728 **1.42.13 Enable signalfd() system call**

1729 CONFIG_SIGNALFD [=y] [Y]

1730 Aktivieren Sie den Systemaufruf `signalfd()`, der den Empfang von Signalen auf einem Dateideskriptor
1731 ermöglicht.

1732 Wenn Sie unsicher sind, sagen Sie Y.

1733 *Enable the signalfd() system call that allows to receive signals on a file descriptor.*

1734 *If unsure, say Y.*

1735 **1.42.14 Enable timerfd() system call**

1736 CONFIG_TIMERFD [=y] [Y]

1737 Aktivieren Sie den Systemaufruf `timerfd()`, der den Empfang von Timer-Ereignissen auf einem Datei-
1738 deskriptor ermöglicht.

1739 Wenn Sie unsicher sind, sagen Sie Y.

1740 *Enable the timerfd() system call that allows to receive timer events on a file descriptor.*

1741 *If unsure, say Y.*

1742 **1.42.15 Enable eventfd() system call**

1743 CONFIG_EVENTFD [=y] [Y]

1744 Aktivieren Sie den Systemaufruf `eventfd()`, der es ermöglicht, sowohl Kernel-Benachrichtigungen (z. B.
1745 KAIO) als auch Userspace-Benachrichtigungen zu empfangen.

1746 Wenn Sie unsicher sind, sagen Sie Y.

1747 *Enable the eventfd() system call that allows to receive both kernel notification (ie. KAIO) or userspace notifications.*

1748 *If unsure, say Y.*

1749 **1.42.16 Use full shmem filesystem**

1750 CONFIG_SHMEM [=y] [Y]

1751 Das `shmem` ist ein internes Dateisystem, das zur Verwaltung des gemeinsamen Speichers verwendet wird.
1752 Es wird von `swap` unterstützt und verwaltet Ressourcengrenzen. Es wird auch in den Userspace als `tmpfs`
1753 exportiert, wenn TMPFS aktiviert ist. Wenn Sie diese Option deaktivieren, werden `shmem` und `tmpfs`
1754 durch den viel einfacheren `ramfs`-Code ersetzt, der auf kleinen Systemen ohne `swap` geeignet sein kann.

1755 *The shmem is an internal filesystem used to manage shared memory. It is backed by swap and manages resource
1756 limits. It is also exported to userspace as tmpfs if TMPFS is enabled. Disabling this option replaces shmem and
1757 tmpfs with the much simpler ramfs code, which may be appropriate on small systems without swap.*

1758 **1.42.17 Enable AIO support**

1759 CONFIG_AIO [=y] [Y]

1760 Diese Option aktiviert die asynchrone POSIX-E/A, die von einigen Hochleistungs-Thread-Anwendungen
1761 verwendet werden kann. Die Deaktivierung dieser Option spart etwa 7kB.

1762 *This option enables POSIX asynchronous I/O which may be used by some high performance threaded applications.
1763 Disabling this option saves about 7k.*

1764 **1.42.18 Enable IO_uring support**

1765 CONFIG_IO_URING [=y] [Y]

1766 Diese Option aktiviert die Unterstützung für die `io_uring`-Schnittstelle, die es Anwendungen ermöglicht,
1767 IO über Übermittlungs- und Abschlussringe zu übermitteln und abzuschließen, die von Kernel und
1768 Anwendung gemeinsam genutzt werden.

1769 *This option enables support for the io_uring interface, enabling applications to submit and complete IO through
1770 submission and completion rings that are shared between the kernel and application.*

1771 **1.43 Enable madvise/fadvise syscalls**

1772 CONFIG_ADVISE_SYSCALLS [=y] [Y]

1773 Diese Option aktiviert die Syscalls `madvise` und `fadvise`, die von Anwendungen verwendet werden, um den
1774 Kernel über ihre zukünftige Speicher- oder Dateinutzung zu informieren und so die Leistung zu verbessern.
1775 Wenn Sie ein eingebettetes System bauen, bei dem keine Anwendungen diese Syscalls verwenden, können
1776 Sie diese Option deaktivieren, um Platz zu sparen.

1777 *This option enables the madvise and fadvise syscalls, used by applications to advise the kernel about their future
1778 memory or file usage, improving performance. If building an embedded system where no applications use these
1779 syscalls, you can disable this option to save space.*

1780 **1.44 Enable membarrier() system call**

1781 CONFIG_MEMBARRIER [=y] [Y]

1782 Aktivieren Sie den Systemaufruf `membarrier()`, der die Ausgabe von Speicherbarrieren für alle laufenden
1783 Threads ermöglicht. Damit können die Kosten für Speicherbarrieren im Benutzerbereich asymmetrisch
1784 verteilt werden, indem Paare von Speicherbarrieren in Paare bestehend aus `membarrier()` und einer
1785 Kompilerbarriere umgewandelt werden.

1786 Wenn Sie unsicher sind, sagen Sie Y.

1787 *Enable the membarrier() system call that allows issuing memory barriers across all running threads, which can be
1788 used to distribute the cost of user-space memory barriers asymmetrically by transforming pairs of memory barriers
1789 into pairs consisting of membarrier() and a compiler barrier.*

1790 *If unsure, say Y.*

1791 **1.45 Enable kcmp() system call**

1792 CONFIG_KCMP [=y] [Y]

1793 Aktivieren Sie den Systemaufruf Kernel-Ressourcenvergleich. Er bietet dem Benutzerbereich die Möglichkeit,
1794 zwei Prozesse zu vergleichen, um festzustellen, ob sie eine gemeinsame Ressource teilen, wie z. B. einen
1795 Dateideskriptor oder sogar virtuellen Speicherplatz.

1796 Wenn Sie unsicher sind, sagen Sie N.

1797 *Enable the kernel resource comparison system call. It provides user-space with the ability to compare two processes
1798 to see if they share a common resource, such as a file descriptor or even virtual memory space.*

1799 *If unsure, say N.*

1800 **1.46 Enable rseq() system call**

1801 CONFIG_RSEQ [=y] [Y]

1802 Aktivieren Sie den Systemaufruf „Neustartbare Sequenzen“. Er bietet einen User-Space-Cache für den
1803 aktuellen CPU-Nummernwert, der das Abrufen der aktuellen CPU-Nummer aus dem User-Space beschleu-
1804 nigt, sowie eine ABI zur Beschleunigung von User-Space-Operationen auf pro-CPU-Daten.

1805 Wenn Sie unsicher sind, sagen Sie Y.

1806 *Enable the restartable sequences system call. It provides a user-space cache for the current CPU number value,
1807 which speeds up getting the current CPU number from user-space, as well as an ABI to speed up user-space
1808 operations on per-CPU data.*

1809 *If unsure, say Y.*

1810 **1.46.1 Enable debugging of rseq() system call**

1811 CONFIG_DEBUG_RSEQ [=n] [N]

1812 Aktivieren Sie zusätzliche Debugging-Prüfungen für den Systemaufruf `rseq`.

1813 Wenn Sie unsicher sind, sagen Sie N.

1814 *Enable extra debugging checks for the rseq system call.*

1815 *If unsure, say N.*

1816 **1.47 Enable cachestat() system call**

1817 CONFIG_CACHESTAT_SYSCALL [=y] [Y]

1818 Aktivieren Sie den Systemaufruf `cachestat`, der die Seiten-Cache-Statistiken einer Datei abfragt (Anzahl
1819 der zwischengespeicherten Seiten, schmutzige Seiten, zum Zurückschreiben markierte Seiten, (kürzlich)
1820 entfernte Seiten).

1821 Wenn Sie unsicher sind, sagen Sie hier Y.

1822 *Enable the cachestat system call, which queries the page cache statistics of a file (number of cached pages, dirty
1823 pages, pages marked for writeback, (recently) evicted pages).*

1824 *If unsure say Y here.*

1825 **1.48 PC/104 support**

1826 CONFIG_PC104 [=n] [N]

1827 Zeigt PC/104-Formfaktor-Gerätetreiber und Optionen zur Auswahl und Konfiguration an. Aktivieren Sie

1828 diese Option, wenn Ihr Zielrechner über einen PC/104-Bus verfügt.
1829 Expose PC/104 form factor device drivers and options available for selection and configuration. Enable this option
1830 if your target machine has a PC/104 bus.

1831 1.49 Load all symbols for debugging/ksymoops

1832 CONFIG_KALLSYMS [=y] [Y]
1833 (sichtbar wenn EXPERT [=y])
1834 Geben Sie hier Y ein, damit der Kernel symbolische Absturzinformationen und symbolische Stack-
1835 Backtraces ausgibt. Dies erhöht die Größe des Kernels etwas, da alle Symbole in das Kernel-Image geladen
1836 werden müssen.
1837 Say Y here to let the kernel print out symbolic crash information and symbolic stack backtraces. This increases the
1838 size of the kernel somewhat, as all symbols have to be loaded into the kernel image.

1839 1.49.1 Test the basic functions and performance of kallsyms

1840 CONFIG_KALLSYMS_SELFTEST [=n] [N]
1841 Testen Sie die Grundfunktionen und die Leistung einiger Schnittstellen, wie z. B. `kallsyms_lookup_name`.
1842 Außerdem wird die Kompressionsrate des `kallsyms`-Kompressionsalgorithmus für den aktuellen Symbolsatz
1843 berechnet. Starten Sie den Selbsttest automatisch nach dem Systemstart.
1844 Es wird empfohlen, `dmesg | grep kallsyms_selftest` auszuführen, um die Testergebnisse zu sammeln.
1845 In der letzten Zeile wird `finish` angezeigt, was bedeutet, dass der Test abgeschlossen ist.
1846 Test the basic functions and performance of some interfaces, such as `kallsyms_lookup_name`. It also calculates the
1847 compression rate of the `kallsyms` compression algorithm for the current symbol set.
1848 Start self-test automatically after system startup. Suggest executing “`dmesg | grep kallsyms_selftest`” to collect test
1849 results. “`finish`” is displayed in the last line, indicating that the test is complete.

1850 1.49.2 Include all symbols in kallsyms

1851 CONFIG_KALLSYMS_ALL [=y] [N]
1852 Normalerweise enthält `kallsyms` nur die Symbole von Funktionen für schönere OOPS-Meldungen und
1853 Backtraces (d. h. Symbole aus den Abschnitten text und inittext). Dies ist für die meisten Fälle ausreichend.
1854 Nur wenn Sie Kernel-Live-Patching oder andere weniger häufige Anwendungsfälle (z. B. wenn ein Debugger
1855 verwendet wird) aktivieren wollen, sind alle Symbole erforderlich (d. h. die Namen von Variablen aus den
1856 Data-Abschnitten usw.).
1857 Diese Option stellt sicher, dass alle Symbole in das Kernel-Image geladen werden (d. h. Symbole aus
1858 allen Sektionen), was die Kernelgröße erhöht (je nach Kernelkonfiguration kann sie 300 KiB oder etwas
1859 Ähnliches betragen).
1860 Sagen Sie N, es sei denn, Sie brauchen wirklich alle Symbole, oder Kernel-Live-Patching.
1861 Normally `kallsyms` only contains the symbols of functions for nicer OOPS messages and backtraces (i.e., symbols
1862 from the text and inittext sections). This is sufficient for most cases. And only if you want to enable kernel live
1863 patching, or other less common use cases (e.g., when a debugger is used) all symbols are required (i.e., names of
1864 variables from the data sections, etc).
1865 This option makes sure that all symbols are loaded into the kernel image (i.e., symbols from all sections) in cost of
1866 increased kernel size (depending on the kernel configuration, it may be 300KiB or something like this).
1867 Say N unless you really need all symbols, or kernel live patching.

1868 1.50 Kernel Performance Events And Counters →

1869 Kernel-Leistungsereignisse und -Zähler

1870 1.50.1 Kernel performance events and counters

1871 CONFIG_PERF_EVENTS [=y] [Y]
1872 Aktivieren Sie die Kernel-Unterstützung für verschiedene von Software und Hardware bereitgestellte
1873 Leistungsereignisse.
1874 Software-Ereignisse werden entweder integriert oder über die Verwendung von generischen Tracepoints
1875 unterstützt.

1876 Die meisten modernen CPUs unterstützen Leistungsereignisse über Leistungszählerregister. Diese
1877 Register zählen die Anzahl bestimmter Arten von hw-Ereignissen: z. B. ausgeführte Anweisungen, erlittene
1878 Cachemisses oder falsch vorhergesagte Verzweigungen – ohne den Kernel oder Anwendungen zu verlangsamen.
1879 Diese Register können auch Unterbrechungen auslösen, wenn eine bestimmte Anzahl von Ereignissen
1880 überschritten wird – und können so dazu verwendet werden, ein Profil des Codes zu erstellen, der auf
1881 dieser CPU läuft.

1882 Das Linux-Performance-Event-Subsystem bietet eine Abstraktion dieser Software- und Hardware-Event-
1883 Fähigkeiten, die über einen Systemaufruf zugänglich sind und vom Dienstprogramm `perf` in `tools/perf/`
1884 verwendet werden. Es stellt Zähler pro Task und pro CPU zur Verfügung und bietet darüber hinaus
1885 Ereignisfunktionen.

1886 Sagen Sie Y, wenn Sie unsicher sind.

1887 *Enable kernel support for various performance events provided by software and hardware.*

1888 *Software events are supported either built-in or via the use of generic tracepoints.*

1889 *Most modern CPUs support performance events via performance counter registers. These registers count the
1890 number of certain types of hw events: such as instructions executed, cachemisses suffered, or branches mis-predicted
1891 – without slowing down the kernel or applications. These registers can also trigger interrupts when a threshold
1892 number of events have passed – and can thus be used to profile the code that runs on that CPU.*

1893 *The Linux Performance Event subsystem provides an abstraction of these software and hardware event capabilities,
1894 available via a system call and used by the “perf” utility in tools/perf/. It provides per task and per CPU counters,
1895 and it provides event capabilities on top of those.*

1896 *Say Y if unsure.*

1897 1.50.1.1 Debug: use vmalloc to back perf mmap() buffers

1898 CONFIG_DEBUG_PERF_USE_VMALLOC [=n] [N]

1899 Verwendung von `vmalloc`-Speicher zur Sicherung von `mmap()`-Puffern. Hauptsächlich nützlich zum De-
1900 buggen des `vmalloc`-Codes auf Plattformen, die dies nicht erfordern. Sagen Sie N, wenn Sie unsicher sind.

1901 *Use vmalloc memory to back perf mmap() buffers.*

1902 *Mostly useful for debugging the vmalloc code on platforms that don't require it.*

1903 *Say N if unsure.*

1904 1.51 Profiling support

1905 CONFIG_PROFILING [=y] [Y]

1906 Sagen Sie hier Y, um die erweiterten Unterstützungsmechanismen für das Profiling zu aktivieren, die von
1907 Profilern verwendet werden.

1908 *Say Y here to enable the extended profiling support mechanisms used by profilers.*

1909 1.52 Rust support

1910 CONFIG_RUST [=y] [Y]

1911 Aktiviert die Rust-Unterstützung im Kernel.

1912 Dies ermöglicht die Auswahl anderer Rust-bezogener Optionen, wie z. B. in Rust geschriebene Treiber.

1913 Es ist auch erforderlich, um externe, in Rust geschriebene Kernel-Module laden zu können.

1914 Siehe Dokumentation/rust/ für weitere Informationen.

1915 Wenn Sie unsicher sind, sagen Sie N.

1916 *Enables Rust support in the kernel.*

1917 *This allows other Rust-related options, like drivers written in Rust, to be selected.*

1918 *It is also required to be able to load external kernel modules written in Rust.*

1919 *See Documentation/rust/ for more information.*

1920 *If unsure, say N.*

1921 1.53 Kexec and crash features →

1922 *Kexec und Absturzmerkmale*

1923 **1.53.1 Enable kexec system call**

1924 **CONFIG_KEXEC [=y] [Y]**

1925 **kexec** ist ein Systemaufruf, der die Fähigkeit implementiert, den aktuellen Kernel herunterzufahren und
1926 einen anderen Kernel zu starten. Es ist wie ein Neustart, aber er ist unabhängig von der System-Firmware.
1927 Und wie ein Neustart können Sie damit jeden Kernel starten, nicht nur Linux. Der Name kommt von
1928 der Anlehnung mit dem Systemaufruf **exec**. Es ist ein fortlaufender Prozess, um sicher zu sein, dass die
1929 Hardware eines Rechners ordnungsgemäß heruntergefahren wird, seien Sie also nicht überrascht, wenn
1930 dieser Code bei Ihnen zunächst nicht funktioniert. Zum Zeitpunkt des Verfassens dieses Artikels ist die
1931 genaue Hardwareschnittstelle noch stark im Wandel, so dass keine gute Empfehlung ausgesprochen werden
1932 kann.

1933 *kexec is a system call that implements the ability to shutdown your current kernel, and to start another kernel. It
1934 is like a reboot but it is independent of the system firmware. And like a reboot you can start any kernel with it, not
1935 just Linux.*

1936 *The name comes from the similarity to the exec system call.*

1937 *It is an ongoing process to be certain the hardware in a machine is properly shutdown, so do not be surprised if
1938 this code does not initially work for you. As of this writing the exact hardware interface is strongly in flux, so no
1939 good recommendation can be made.*

1940 **1.53.2 Enable kexec file based system call**

1941 **CONFIG_KEXEC_FILE [=y] [Y]**

1942 (Aktivieren des dateibasierten Systemaufrufs **kexec**)

1943 Dies ist eine neue Version des Systemaufrufs **kexec**. Dieser Systemaufruf ist dateibasiert und nimmt
1944 Dateideskriptoren als Systemaufrufsargument für Kernel und **initramfs** anstelle einer Liste von Segmenten,
1945 wie sie vom **kexec**-Systemaufruf akzeptiert wird.

1946 *This is new version of kexec system call. This system call is file based and takes file descriptors as system call
1947 argument for kernel and initramfs as opposed to list of segments as accepted by kexec system call.*

1948 **1.53.2.1 Verify kernel signature during kexec_file_load() syscall**

1949 **CONFIG_KEXEC_SIG [=y] [Y]**

1950 Mit dieser Option wird der Syscall **kexec_file_load()** auf eine gültige Signatur des Kernel-Images
1951 geprüft. Das Image kann immer noch ohne gültige Signatur geladen werden, es sei denn, Sie aktivieren
1952 auch **KEXEC_SIG_FORCE**, aber wenn es eine Signatur gibt, die überprüft werden kann, dann muss
1953 sie auch gültig sein. Zusätzlich zu dieser Option müssen Sie die Signaturprüfung für den entsprechenden
1954 Kernel-Image-Typ, der geladen wird, aktivieren, damit dies funktioniert.

1955 *This option makes the kexec_file_load() syscall check for a valid signature of the kernel image. The image can still
1956 be loaded without a valid signature unless you also enable KEXEC_SIG_FORCE, though if there's a signature
1957 that we can check, then it must be valid.*

1958 *In addition to this option, you need to enable signature verification for the corresponding kernel image type being
1959 loaded in order for this to work.*

1960 **1.53.2.1.1 Require a valid signature in kexec_file_load() syscall**

1961 **CONFIG_KEXEC_SIG_FORCE [=n] [N]**

1962 Diese Option macht die Überprüfung der Kernelsignatur für den Syscall **kexec_file_load()** zwingend
1963 erforderlich.

1964 *This option makes kernel signature verification mandatory for the kexec_file_load() syscall.*

1965 **1.53.2.1.2 Enable bzImage signature verification support**

1966 **CONFIG_KEXEC_BZIMAGE_VERIFY_SIG [=y] [Y]**

1967 Aktivierung der Unterstützung von bzImage für die Signaturprüfung.

1968 *Enable bzImage signature verification support.*

1969 **1.53.3 kexec jump**

1970 **CONFIG_KEXEC_JUMP [=y] [Y]**

1971 Sprung zwischen Original-Kernel und **kexec**-Kernel und Aufruf von Code im physikalischen Adressmodus
1972 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 speziellen Absturz-Dump-Kernen eingestellt werden, die mit kexec-tools in den Hauptkernel in einen speziell reservierten Bereich geladen und später nach einem Absturz von kdump/kexec ausgeführt werden. Der Absturz-Dump-Kernel muss mit PHYSICAL_START auf eine Speicheradresse kompiliert werden, die nicht vom Hauptkernel oder BIOS verwendet wird, oder er muss als verschiebbbares Image (CONFIG_RELOCATABLE=y) erstellt werden. Weitere Details finden Sie unter Documentation/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 which are loaded in the main kernel with kexec-tools into a specially reserved region and then later executed after a crash by kdump/kexec. The crash dump kernel must be compiled to a memory address not used by the main kernel 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 nur in speziellen Crash-Dump-Kernen gesetzt werden, die im Hauptkernel mit kexec-tools in einen speziell reservierten Bereich geladen werden und dann später nach einem Absturz von kdump/kexec ausgeführt werden. Der Crash-Dump-Kernel muss mit PHYSICAL_START auf eine Speicheradresse kompiliert werden, die nicht vom Hauptkernel oder BIOS verwendet wird, oder er muss als relocatable image (CONFIG_RELOCATABLE=y) erstellt werden.

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 which are loaded in the main kernel with kexec-tools into a specially reserved region and then later executed after a crash by kdump/kexec. The crash dump kernel must be compiled to a memory address not used by the main kernel 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 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 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 Speicherbereiche enthält, die bei einem Absturz gelöscht werden sollen) als Reaktion auf Hot-Plug/Unplug oder Online/Offline von CPUs oder Speicher. Dies ist ein sehr viel fortschrittlicherer Ansatz als der Versuch dies im Userspace zu tun.

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 upon a crash) in response to hot plug/unplug or online/offline of CPUs or memory. This is a much more advanced*

2024 approach than userspace attempting that.

2025 If unsure, say Y.

2026 1.53.6.1 Specify the maximum number of memory regions for the elfcorehdr

2027 CONFIG_CRASH_MAX_MEMORY_RANGES [=8192] [8192]

2028 Für den Pfad des Systemaufrufs `kexec_file_load()` ist die maximale Anzahl der Speicherbereiche
2029 anzugeben, die der `elfcorehdr`-Puffer bzw. das `elfcorehdr`-Segment aufnehmen kann. Diese Regionen
2030 werden über `walk_system_ram_res()` ermittelt, z. B. die ‚System RAM‘-Einträge in `/proc/iomem`.
2031 Dieser Wert wird mit `NR_CPUS_DEFAULT` kombiniert und mit `sizeof(Elf64_Phdr)` multipliziert, um
2032 die endgültige `elfcorehdr`-Speicherpuffer-/Segmentgröße zu bestimmen. Der Wert 8192 beispielsweise
2033 deckt ein (dünn besiedeltes) 1 TiB-System ab, das aus 128 MiB-Memblocken besteht, und führt zu einer
2034 `elfcorehdr`-Speicherpuffer-/Segmentgröße von unter 1 MiB. Dies ist eine vernünftige Wahl, um sowohl
2035 Baremetal- als auch virtuelle Maschinenkonfigurationen zu unterstützen.

2036 Für den Syscall-Pfad `kexec_load()` ist `CRASH_MAX_MEMORY_RANGES` Teil der Berechnung hinter
2037 dem Wert, der über das Attribut `/sys/kernel/crash_elfcorehdr_size` bereitgestellt wird.

2038 *For the kexec_file_load() syscall path, specify the maximum number of memory regions that the elfcorehdr buffer
2039 / segment can accommodate. These regions are obtained via walk_system_ram_res(); eg. the ‘System RAM’
2040 entries in /proc/iomem. This value is combined with NR_CPUS_DEFAULT and multiplied by sizeof(Elf64_Phdr)
2041 to determine the final elfcorehdr memory buffer/segment size. The value 8192, for example, covers a (sparsely
2042 populated) 1TiB system consisting of 128MiB memblocks, while resulting in an elfcorehdr memory buffer/segment
2043 size under 1MiB. This represents a sane choice to accommodate both baremetal and virtual machine configurations.
2044 For the kexec_load() syscall path, CRASH_MAX_MEMORY_RANGES is part of the computation behind the
2045 value provided through the /sys/kernel/crash_elfcorehdr_size attribute.*

2046 1.54 Live Update and Kexec Handover → seit 6.19

2047 *Live-Update und Kexec-Übergabe*

2048 1.54.1 kexec handover seit 6.16

2049 CONFIG_KEXEC_HANOVER [=y] [Y]

2050 Ermöglichen Sie `kexec` die Übergabe des Status zwischen Kerneln, indem Sie zusätzliche Metadaten
2051 generieren und an den Zielkernel übergeben. Dies ist nützlich, um Daten oder den Status über `kexec`
2052 hinweg aufrechtzuerhalten. Damit dies funktioniert, muss diese Option sowohl im Quell- als auch im
2053 Zielkernel aktiviert sein.

2054 *Allow kexec to hand over state across kernels by generating and passing additional metadata to the target kernel.
2055 This is useful to keep data or state alive across the kexec. For this to work, both source and target kernels need to
2056 have this option enabled.*

2057 1.54.1.1 Enable Kexec Handover debug checks seit 6.18

2058 CONFIG_KEXEC_HANOVER_DEBUG [=n] [N]

2059 Diese Option aktiviert zusätzliche Plausibilitätsprüfungen für das Kexec-Handover-Subsystem. Da die
2060 KHO-Leistung in Live-Update-Szenarien von entscheidender Bedeutung ist und der zusätzliche Code
2061 möglicherweise zu einem Mehraufwand führt, ist diese Option nur optional aktiviert.

2062 *This option enables extra sanity checks for the Kexec Handover subsystem. Since, KHO performance is crucial in
2063 live update scenarios and the extra code might be adding overhead it is only optionally enabled.*

2064 1.54.1.2 kexec handover debugfs interface seit 6.19

2065 CONFIG_KEXEC_HANOVER_DEBUGFS [=y] [Y]

2066 Ermöglicht die Steuerung der Kexec-Übergabe des Device Tree über die Debugfs-Schnittstelle, d. h. die
2067 Fertigstellung des Status oder den Abbruch der Fertigstellung. Ermöglicht außerdem die Überprüfung der
2068 KHO-Fdt-Bäume mit den Debugfs-Binärblobs.

2069 *Allow to control kexec handover device tree via debugfs interface, i.e. finalize the state or aborting the finalization.
2070 Also, enables inspecting the KHO fdt trees with the debugfs binary blobs.*

2071 1.54.1.3 Enable kexec handover by default seit 6.19

2072 CONFIG_KEXEC_HANOVER_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

2082 CONFIG_64BIT [=y] [Y]

2084 Sagen Sie Y für ja, zur Erstellung eines 64-Bit-Kernels – früher bekannt als x86_64

2085 Sagen Sie N für nein, um einen 32-Bit-Kernel zu erstellen – früher bekannt als i386

2086 *Say yes to build a 64-bit kernel – formerly known as x86_64*

2087 *Say no to build a 32-bit kernel – formerly known as i386*