

## Basics zum Linux-Kernel, Teil 2

Antworten auf die häufigsten Fragen

Von Thorsten Leemhuis

### Kernel-Abbild

**?** Was ist ein Linux-Kernel-Image?

**!** Dabei handelt es sich um einen ausführbaren Linux-Kernel. Sprich: Jemand – zumeist Ihr Linux-Distributor – hat den Quellcode des Linux-Kernels heruntergeladen, konfiguriert und kompiliert. Derjenige legt dabei Funktionsumfang und Treiberausstattung fest, was auch die Art der Prozessorarchitektur umfasst.

Die auf einem System installierten Kernel-Images finden Sie typischerweise unterhalb von `/boot/` in Dateien, deren Namen mit „`vmlinuz-`“ beginnen. Darauf folgt typischerweise die Versionsbezeichnung, die `uname -r` auswirft.

### Initramfs & Initrd

**?** Neben den Kernel-Images liegen in `/boot/` noch einige Dateien, die mit „`initrd`“ oder „`initramfs`“ beginnen. Sie tragen Versionsnummern wie ein Kernel-Image. Was steckt in diesen Dateien?

**!** Beim Initramfs (Initial RAM File System) handelt es sich um ein Dateisystem-Image, mit dessen Hilfe die allermeisten Linux-Distributionen heutzutage booten. Hauptaufgabe der im Archiv enthaltenen Komponenten: Das Root-Dateisystem finden und einbinden, um dann die Kontrolle an das dort liegenden Init-System zu übergeben.

Für diese Aufgaben stecken im Initramfs eine Reihe von Kernel-Modulen, Firmware-Dateien und Werkzeugen. Darunter sind oft die Kernel-Module für Ihren Storage-Adapter und das Dateisystem, mit dem die Root-Partition formatiert wurde. Sollten Sie Letztere verschlüsselt haben, stecken im Archiv auch Programme, um die Passphrase abzufragen und die Partition damit zu entsperren. Ferner kann das Initramfs auch Netzwerktreiber, Konfigurationsdaten und Werkzeuge enthalten, um ein Root-Dateisystem auf anderen Wegen einzubinden –

etwa via Ethernet, iSCSI oder gar das Internet.

Die im Initramfs enthaltenen Module müssen zum eingesetzten Kernel gehören, daher gibt es normalerweise für jedes Kernel-Image ein eigenes Initramfs. Distributionen erzeugen es typischerweise automatisch bei der Kernel-Installation. Dazu nutzen sie Werkzeuge wie Dracut (Fedora, Red Hat, OpenSuse, Suse, ...), Initramfs-Tools/Mkinitramfs/Update-Initramfs (Debian, Ubuntu ...) oder Mkinitcpio (Arch Linux). Diese Programme legen meist ein Archiv an, das genau zur Hardware und der jeweiligen Linux-Installation passt. Das soll unnötigen Komponenten vermeiden, denn die würden das Archiv nur größer machen und so den Systemstart verlangsamen.

Früher nutzten Distributionen statt eines Initramfs ein Archiv namens Initrd (Initial Ramdisk), das ähnlich arbeitet. Die Begriffe werden daher häufig synonym verwendet. Prinzipiell lässt sich eine Linux-Installation auch ohne diese Techniken starten – das Kernel-Image muss dann aber alles enthalten, was zum Einbinden der Root-Partition erforderlich ist. In dem Fall braucht in der Konfiguration des Boot-Managers kein Initramfs referenziert werden; stattdessen muss er nur das Kernel-Image in den Speicher heben und ausführen, um Linux zu starten.

### Kernel Oops & Panic

**?** In den Ausgaben von `dmesg` sehe ich einen „Kernel Oops“. Ist das eine dieser „Kernel Panics“, von der ich mal gehört habe? Oder nur ein Warnhinweis, den ich ignorieren kann?

**Kernel.org nennt die prognostizierten Wartungszeiten der verschiedenen Linux-Versionen.**

**!** Der Kernel kennt verschiedene Fehlerarten. Am schwerwiegendsten ist die „Kernel Panic“: Bei einer solchen stoppt sich Linux absichtlich selbst, weil es nicht weiter arbeiten kann. Das kann passieren, wenn es plötzlich nicht mehr auf die Root-Partition zugreifen kann. Manchmal gibt der Kernel aus Sicherheitsgründen auf, damit es weder Hardware noch Daten beschädigt – etwa wenn Linux die Root-Partition noch erreicht, aber nicht sicherstellen kann, ob geschriebene Daten auch an dem vorgesehenen Ort landen.

Auch beim „Kernel Oops“ ist ein schwerwiegender Fehler aufgetreten – diesen konnte der Kernel allerdings so weit abfangen, dass ein Weiterbetrieb möglich ist. Der Kernel legt dabei aber womöglich Teilfunktionen seiner selbst lahm. Im dümmsten Fall funktioniert daher ein Treiber für eine Hardware-Komponente nicht mehr. Dann müssen Sie meist neu starten, um die vom Treiber betreute Hardware wieder verwenden zu können.

Auf kleinere Probleme weist der Kernel mit Meldungen hin, die typischerweise mit „BUG“ oder „WARNING“ beginnen. Auch hinter diesen können sich Probleme verbergen, die größere Auswirkungen haben. So warnen Treiber manchmal auf diese Weise, wenn eine zwingend benötigte Firmware-Datei fehlt.

Von der jeweiligen Situation hängt ab, ob und wie Sie auf solche Fehlerhinweise reagieren müssen. Lassen Sie sich dabei nicht davon einschüchtern, dass die Ausgaben von Oops und Panic ähnlich umfassend und abschreckend wirken wie jene eines Blue Screen von Windows. Auch Ungewübte finden in den Kernel-Ausgaben leicht Informationen zur Fehlerursache. Eine Kurzbeschreibung des Grunds steht

Longterm release kernels			
Version	Maintainer	Released	Projected EOL
4.9	Greg Kroah-Hartman	2016-12-11	Jan, 2019
4.4	Greg Kroah-Hartman	2016-01-10	Feb, 2018
4.1	Sasha Levin	2015-06-21	Sep, 2017
3.16	Ben Hutchings	2014-08-03	Apr, 2020
3.10	Willy Tarreau	2013-06-30	Oct, 2017
3.4	Li Zefan	2012-05-20	Apr, 2017
3.2	Ben Hutchings	2012-01-04	May, 2018

meist schon in der ersten Zeile. Bei der Panic „Attempted to kill init“ ist etwa der Init-Prozess abgestürzt. Daran kann ein beschädigtes Initramfs schuld sein, denn schon dort läuft ein Prozess, der Init-ähnliche Aufgaben übernimmt. Es können auch fehlende oder beschädigte Dateien auf dem Root-Dateisystem sein, die der Init-Daemon nutzt; das sind auch mögliche Ursachen, falls der Kernel „No init found. Try passing init= option to the kernel“ meldet.

Vorsicht: Achten Sie bei der Analyse eines Oops auf den vierstelligen Zähler in der mit „Oops“ beginnenden Zeile. Wenn dort nicht „0000“ steht, dann war es nicht das erste schwerwiegende Problem. Letzterem sollten Sie dann zuerst nachspüren, denn möglicherweise ist der angezeigte Fehler nur ein Folgefehler, der mit der eigentlichen Ursache nur wenig zu tun hat: Ein Oops in einem einzelnen Treiber kann manchmal ein Fehlverhalten in anderen Treibern des Kernels nach sich ziehen.

## Root unauffindbar

**?** Mein System startet nicht; stattdessen zeigt der Kernel eine Fehlermeldung, es habe die Root-Partition nicht einbinden können. Was läuft schief?

**!** Das kann verschiedene Ursachen haben. Hauptverdächtig ist das Initramfs; manchmal kann auch ein Hardware-Defekt oder eine fehlerhafte Boot-Loader-Konfiguration die Ursache sein.

Viele Distributionen halten ältere Kernel im Boot-Menü für Notfälle bereit. Wenn einer der Kernel startet, ist klar: Die Hardware funktioniert, sie trifft keine Schuld.

Untersuchen Sie anschließend die Fehlermeldungen genauer. Steht dort etwas von „Kernel panic - not syncing: VFS: Unable to mount root fs“, dann wurde das Initramfs wahrscheinlich gar nicht geladen. Das kann passieren, wenn das Initramfs fehlt, beschädigt ist oder in der Boot-Manager-Konfiguration nicht angegeben wurde. Bei Grub können Sie Letzteres zur Laufzeit über die Edit-Funk-

```
[ 0.585779] VFS: Cannot open root device "UUID=ebbd4734-596f-4244-ab4c-0f3cd0
cb9409" or unknown-block(0,0): error -6
[ 0.594256] Please append a correct "root=" boot option; here are the availab
le partitions:
[ 0.595007] Kernel panic - not syncing: VFS: Unable to mount root fs on unkno
wn-block(0,0)
[ 0.595759] CPU: 1 PID: 1 Comm: swapper/0 Not tainted 4.11.1-300.fc26.x86_64
#1
[ 0.596496] Hardware name: Red Hat KVM, BIOS 0.5.1 01/01/2011
[ 0.596909] Call Trace:
[ 0.597287] dump_stack+0x63/0x84
[ 0.597673] panic+0xe4/0x22d
[ 0.598053] mount_block_root+0x1e6/0x2cf
[ 0.598450] ? set_debug_rodata+0x12/0x12
[ 0.598843] mount_root+0x65/0x68
[ 0.599229] prepare_namespace+0x134/0x16c
[ 0.599920] kernel_init_freeable+0x227/0x24f
[ 0.600323] ? rest_init+0x80/0x80
[ 0.600708] kernel_init+0xe/0x100
[ 0.601093] ret_from_fork+0x2c/0x40
[ 0.601523] Kernel Offset: 0x25000000 from 0xffffffff81000000 (relocation ran
ge: 0xffffffff80000000-0xffffffffbfffffff)
[ 0.602307] ---[ end Kernel panic - not syncing: VFS: Unable to mount root fs
on unknown-block(0,0)
```

Bei dieser Kernel Panic konnte Linux die Root-Partition nicht einbinden, weil das Initramfs fehlte.

tion korrigieren, die sie nach dem Auswählen eines Boot-Eintrags mit der Taste „e“ aufrufen.

Für die anderen Problemursachen müssen Sie die Root-Partition einbinden. Falls Sie keinen der installierten Kernel mehr starten können, müssen Sie dazu ein anderes Linux bemühen. Das kann ein parallel installiertes oder ein vom USB Stick startendes Live-Linux sein. Mit ihm korrigieren Sie dann das Problem. Oft ist es das einfachste, in die Linux-Installation hineinzuwechseln. Dazu müssen Sie diese zuerst finden, wobei das Kommando `lsblk` hilft. Bei einem auf `/dev/sda7` installierten Distribution gelingt das mit den folgenden Kommandos, die Sie als Root oder mithilfe von `sudo` ausführen:

```
mount /dev/sda7 /mnt/
mount --bind /dev/ /mnt/dev/
mount --bind /proc/ /mnt/proc/
mount --bind /sys/ /mnt/sys/
chroot /mnt/
mount -a
```

Dort können Sie dann das Werkzeug nutzen, mit dem Ihre Distribution das zu Ihrem System passende Initramfs erzeugt. Durch Eingabe von `exit` kehren Sie zum gestarteten Linux zurück.

## Userland

**?** Was sind eigentlich „Userland“ und „User space“, von denen ich im Kontext des Kernels immer wieder höre?

**!** Userland bezeichnet die Umgebung, die zusammen mit dem Kernel das Betriebssystem bildet. Das umfasst abge-

sehen vom Kernel-Image alles, was bei der Linux-Installation aufgespielt wird – im Wesentlichen also Anwendungen samt der von ihnen benötigten Bibliotheken, Interpreter und Dateien.

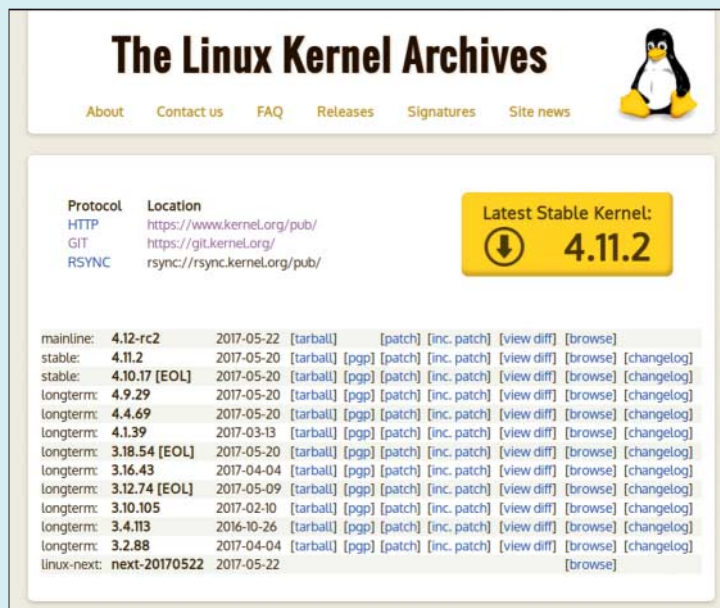
Häufig wird diese Umgebung auch als „User space“ bezeichnet. Streng genommen steht dieser Begriff aber nur für den Teil des Arbeitsspeichers, den der Kernel einem gestarteten Programm zur freien Verfügung bereitstellt. Der Speicherbereich, in dem der Kernel läuft, nennt sich hingegen „Kernel space“. Auf ihn können Programme nur über Systemaufrufe (System calls/Syscalls) zugreifen, über die sie mit dem Kernel interagieren und dabei etwa Daten austauschen.

## Versionswirrwarr

**?** Auf Kernel.org gibt es ein Dutzend verschiedener Kernel-Versionen, die als „mainline“, „stable“, „longterm“ und „linux-next“ kategorisiert sind. Manche Versionsnummern bestehen aus zwei, andere aus drei Nummern, teilweise steht auch „EOL“ oder „rc1“ daneben. Was hat das alles zu bedeuten?

**!** Die Bezeichnungen sollen klar machen, aus welcher Entwicklungslinie die jeweilige Version hervorgegangen ist.


Mit „mainline“ gekennzeichnete Kernel entstammen der Hauptentwicklungslinie von Linux und tragen Versionsnummern wie 4.11. Direkt nach der Veröffentlichung solch einer Version beginnt die Entwicklung des Nachfolgers, die typischerweise zehn Wochen dauert. In der Zeit gibt Linus Torvalds meist wöchent-



**The Linux Kernel Archives**

About Contact us FAQ Releases Signatures Site news

Protocol Location  
 HTTP <https://www.kernel.org/pub/>  
 GIT <https://git.kernel.org/>  
 RSYNC <rsync://rsync.kernel.org/pub/>

**Latest Stable Kernel:**  
 **4.11.2**

mainline:	4.12-rc2	2017-05-22	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]
stable:	4.11.2	2017-05-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
stable:	4.10.17 [EOL]	2017-05-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.9.29	2017-05-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.4.69	2017-05-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	4.1.39	2017-03-13	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.18.54 [EOL]	2017-05-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.16.43	2017-04-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.12.74 [EOL]	2017-05-09	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.10.105	2017-02-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.4.113	2016-10-26	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.2.88	2017-04-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
linux-next:	next-20170522	2017-05-22					[browse]

lich Vorabversionen dieser Mainline-Kernel frei, die er mit „-rc“ kennzeichnet – Linux 4.12-rc2 etwa ist die zweite Vorabversion (auch „Prepatch“ genannt) von 4.12. Die tragen alle ein „rc“ – man solle es daher nicht als „Release Candidate“ betrachten, denn es sei vielmehr ein „Ridiculous Count“ (lächerlicher Zähler), wie Torvalds vor Jahren schrieb.

Parallel zur Arbeit in der Hauptentwicklungslinie fallen Fehler in den zuvor freigegebenen Versionen auf. Einige Entwickler veröffentlichen daher mit Korrekturen und kleinen Verbesserungen angereicherte Fassungen dieser Kernel und kennzeichnen sie mit einer zusätzlichen Versionsnummer. Linux 4.11.1 etwa war die erste Überarbeitung von 4.11.

Die Pflege der jeweils aktuellen Version erfolgt dabei im Rahmen der „stable“-Series. Deren Betreuer stellt die Wartung allerdings oft schon drei bis fünf Wochen nach Erscheinen des nächsten Mainline-Kernels ein. 4.10 wurde etwa parallel mit dem Erscheinen von 4.11.2 aufgegeben – 4.11 war zu dem Zeitpunkt drei Wochen alt. Wenn die Entwickler die Pflege einer Linie einstellen, wird die zuletzt freigegebene Version mit „EOL“ (End of Life) gekennzeichnet. Das soll klar machen: Wechselt besser auf eine neuere Linux-Version, denn diese Kernel-Serie wird nicht mehr gepflegt, daher enthält diese Version womöglich bekannte Sicherheitslücken. Etwas später verschwindet die Zeile zu dieser Version komplett von Kernel.org, um das noch klarer zu machen.

Einige Kernel-Entwickler versorgen ausgewählte Versionen länger mit Korrek-

turen. Das sind dann „longterm“-Kernel. Zu denen gehören beispielsweise die älteren Linien 4.4 und 4.9, von denen mindestens zwei Jahre lang überarbeitete Versionen erscheinen. Manche Reihen erhalten noch länger Korrekturen, weil jemand daran ein Eigeninteresse hat; oft sind das Entwickler von Distributionen, deren Kernel auf der jeweiligen Linux-Version aufbaut. In diese Klasse fällt etwa Linux 3.2, dessen Wartung im Mai 2018 nach sechs Jahren enden soll. Hier sorgt ein Debian-Entwickler für die Langzeitwartung im Rahmen von Kernel.org, weil der Debian-7-Kernel auf 3.2 aufbaut. Ähnlich verhält es sich mit dem von Debian 8 genutzten 3.16. Auch Oracle und Suse-Mitarbeiter haben die Pflege einzelner Versionsreihen übernommen, weil sie auf den jeweiligen Linux-Versionen aufbauende Kernel in ihren Produkten eingesetzt haben.

Typischerweise wird zumeist die im Januar aktuelle Linux-Version ein Longterm-Kernel. Eine Übersicht der verfügbaren Longterm-Kernel samt des geplanten Pflegezeitraums findet sich bei Kernel.org auf der Seite „Releases“. Manchmal wird die dort angegebene Wartungsdauer aber noch ausgedehnt. Das ist beispielsweise beim Longterm-Kernel 4.9 zu erwarten: Nach Ablauf der beiden regulären Pflegejahre wird vermutlich wieder ein Debian-Entwickler übernehmen und die Linie drei weitere Jahre warten, weil der Debian-9-Kernel auf dieser Version aufbaut.

„Linux-next“ richtet sich an Kernel-Enthusiasten und Entwickler, denn Letztere koordinieren über diesen Kernel-Zweig die Änderungen für die jeweils

Kernel.org nennt neben den aktuell gepflegten oder entwickelten Linux-Versionen auch einige, deren Wartung jüngst aufgegeben wurde.

übernächste Mainline-Version. So können die Programmierer ihre Patches für 4.12 aufeinander abstimmen, noch bevor Linux 4.11 fertig ist. Außerdem können Interessierte die Änderungen so frühzeitig testen.

Einige Distributoren nutzen bewusst Stable- oder Longterm-Kernel als Basis für die Kernel ihrer Produkte, um von der Wartung der Kernel-Entwickler zu profitieren. Manche greifen bei der Entwicklung neuer Distributionen einfach eine gerade aktuelle Version auf und reichern sie dann mit Änderungen an. Unabhängig von Kernel.org versorgen sie diesen Kernel dann so lange mit Korrekturen, wie sie die jeweilige Distribution pflegen. Daher ist es auch kein Problem, dass Ubuntu 16.04.2 und 16.10 derzeit einen auf Linux 4.8 basierenden Kernel einsetzen: Canonical wartet diese nach wie vor, obwohl Kernel.org die Pflege von 4.8 bereits vor Monaten aufgegeben hat.

## Vanilla Kernel

**?** Wofür stehen die Bezeichnungen „offizieller Linux-Kernel“ und „Vanilla Kernel“?

**!** Die Begriffe meinen Linux-Quellen und damit gebaute Kernel-Images, bei denen der Quellcode nicht modifiziert wurde. Er ist somit in genau der Form bei Kernel.org erhältlich und stammt komplett von den Kernel-Entwicklern rund um Linus Torvalds. Eben das ist bei den Kernel-Images vieler Linux-Distributoren nicht der Fall, weil sie Zusatzfunktionen einbauen – beispielsweise Treiber, die dem offiziellen Kernel fehlen, weil sie etwa den Qualitätsansprüchen der Entwickler des offiziellen Linux-Kernels nicht genügen. Dessen Programmierer scheren sich daher teilweise nicht um Fehlerberichte von Anwendern, die sich auf die Kernel von Fedora, OpenSuse & Co. beziehen. Kein Wunder, schließlich steckt die Fehlerursache womöglich in Änderungen des Distributors. (thl@ct.de)

**Basiswissen zum Linux-Kernel, Teil 1:**  
[ct.de/yfzf](http://ct.de/yfzf)