The kernel’s command-line parameters

English

The following is a consolidated list of the kernel parameters as implemented by the **`__setup()`**, **`early_param()`**, **`core_param()`** and **`module_param()`** macros and sorted into English Dictionary order (defined as ignoring all punctuation and sorting digits before letters in a case insensitive manner), and with descriptions where known.

The kernel parses parameters from the kernel command line up to “`--`“; if it doesn’t recognize a parameter and it doesn’t contain a ‘.’, the parameter gets passed to init: parameters with ‘=’ go into init’s environment, others are passed as command line arguments to init. Everything after “`--`” is passed as an argument to init.

Module parameters can be specified in two ways: via the kernel command line with a module name prefix, or via modprobe, e.g.:

```
(kernel command line) usbcore.blinkenlights=1
(modprobe command line) modprobe usbcore blinkenlights=1
```

Parameters for modules which are built into the kernel need to be specified on the kernel command line. modprobe looks through the kernel command line (/proc/cmdline) and collects module parameters when it loads a module, so the kernel command line can be used for loadable modules too.

This document may not be entirely up to date and comprehensive. The command “modinfo -p ${modulename}” shows a current list of all parameters of a loadable module. Loadable modules, after being loaded into the running kernel, also reveal their parameters in /sys/module/${modulename}/parameters/. Some of these parameters may be changed at runtime by the command `echo -n ${value} > /sys/module/${modulename}/parameters/${parm}`.

Special handling

Hyphens (dashes) and underscores are equivalent in parameter names, so:

```
log_buf_len=1M print-fatal-signals=1
```

can also be entered as:

```
log-buf-len=1M print_fatal_signals=1
```

Double-quotes can be used to protect spaces in values, e.g.:

```
param="spaces in here"
```

cpu lists

Some kernel parameters take a list of CPUs as a value, e.g. isolcpus, nohz_full, irqaffinity, rcu_nocbs. The format of this list is:

<cpu number>,...,<cpu number>

or

<cpu number>-<cpu number> (must be a positive range in ascending order)

or a mixture

<cpu number>,...,<cpu number>-<cpu number>

Note that for the special case of a range one can split the range into equal sized groups and for each group use some amount from the beginning of that group:

<cpu number>-<cpu number>:<used size>/<group size>

For example one can add to the command line following parameter:

isolcpus=1,2,10-20,100-2000:2/25

where the final item represents CPUs 100,101,125,126,150,151,...

The value “N” can be used to represent the numerically last CPU on the system, i.e “foo_cpus=16-N” would be equivalent to “16-31” on a 32 core system.

Keep in mind that “N” is dynamic, so if system changes cause the bitmap width to change, such as less cores in the CPU list, then N and any ranges using N will also change. Use the same on a small 4 core system, and “16-N” becomes “16-3” and now the same boot input will be flagged as invalid (start > end).

The special case-tolerant group name “all” has a meaning of selecting all CPUs, so that “nohz_full=all” is the equivalent of “nohz_full=0-N”.

The semantics of “N” and “all” is supported on a level of bitmaps and holds for all users of **`bitmap_parselist()`**.

Metric suffixes

The [KMG] suffix is commonly described after a number of kernel parameter values. ‘K’, ‘M’, ‘G’, ‘T’, ‘P’, and ‘E’ suffixes are allowed. These letters represent the _binary_ multipliers ‘Kilo’, ‘Mega’, ‘Giga’, ‘Tera’, ‘Peta’, and ‘Exa’, equaling 2^10, 2^20, 2^30, 2^40, 2^50, and 2^60 bytes respectively. Such letter suffixes can also be entirely omitted.

Kernel Build Options

The parameters listed below are only valid if certain kernel build options were enabled and if respective hardware is present. This list should be kept in alphabetical order. The text in square brackets at the beginning of each description states the restrictions within which a parameter is applicable.

Parameters denoted with BOOT are actually interpreted by the boot loader, and have no meaning to the kernel directly. Do not modify the syntax of boot loader parameters without extreme need or coordination with <The Linux/x86 Boot Protocol>.

There are also arch-specific kernel-parameters not documented here.

Note that ALL kernel parameters listed below are CASE SENSITIVE, and that a trailing = on the name of any parameter states that the parameter will be entered as an environment variable, whereas its absence indicates that it will appear as a kernel argument readable via /proc/cmdline by programs running once the system is up.

The number of kernel parameters is not limited, but the length of the complete command line (parameters including spaces etc.) is limited to a fixed number of characters. This limit depends on the architecture and is between 256 and 4096 characters. It is defined in the file ./include/uapi/asm-generic/setup.h as COMMAND_LINE_SIZE.

```
        ACPI    ACPI support is enabled.
        AGP     AGP (Accelerated Graphics Port) is enabled.
        ALSA    ALSA sound support is enabled.
        APIC    APIC support is enabled.
        APM     Advanced Power Management support is enabled.
        APPARMOR AppArmor support is enabled.
        ARM     ARM architecture is enabled.
        ARM64   ARM64 architecture is enabled.
        CLK     Common clock infrastructure is enabled.
        CMA     Contiguous Memory Area support is enabled.
        DRM     Direct Rendering Management support is enabled.
        DYNAMIC_DEBUG Build in debug messages and enable them at runtime
        EARLY   Parameter processed too early to be embedded in initrd.
        EDD     BIOS Enhanced Disk Drive Services (EDD) is enabled
        EFI     EFI Partitioning (GPT) is enabled
        EVM     Extended Verification Module
        FB      The frame buffer device is enabled.
        FTRACE  Function tracing enabled.
        GCOV    GCOV profiling is enabled.
        HIBERNATION HIBERNATION is enabled.
        HW      Appropriate hardware is enabled.
        HYPER_V HYPERV support is enabled.
        IMA     Integrity measurement architecture is enabled.
        IP_PNP  IP DHCP, BOOTP, or RARP is enabled.
        IPV6    IPv6 support is enabled.
        ISAPNP  ISA PnP code is enabled.
        ISOL    CPU Isolation is enabled.
        JOY     Appropriate joystick support is enabled.
        KGDB    Kernel debugger support is enabled.
        KVM     Kernel Virtual Machine support is enabled.
        LIBATA  Libata driver is enabled
```

```
LOONGARCH LoongArch architecture is enabled.
LOOP    Loopback device support is enabled.
LP      Printer support is enabled.
M68k    M68k architecture is enabled.
                These options have more detailed description inside of
                Documentation/arch/m68k/kernel-options.rst.
MIPS    MIPS architecture is enabled.
MOUSE   Appropriate mouse support is enabled.
MSI     Message Signaled Interrupts (PCI).
MTD     MTD (Memory Technology Device) support is enabled.
NET     Appropriate network support is enabled.
NFS     Appropriate NFS support is enabled.
NUMA    NUMA support is enabled.
OF      Devicetree is enabled.
PARISC  The PA-RISC architecture is enabled.
PCI     PCI bus support is enabled.
PCIE    PCI Express support is enabled.
PCMCIA  The PCMCIA subsystem is enabled.
PNP     Plug & Play support is enabled.
PPC     PowerPC architecture is enabled.
PPT     Parallel port support is enabled.
PS2     Appropriate PS/2 support is enabled.
PV_OPS  A paravirtualized kernel is enabled.
RAM     RAM disk support is enabled.
RDT     Intel Resource Director Technology.
RISCV   RISCV architecture is enabled.
S390    S390 architecture is enabled.
SCSI    Appropriate SCSI support is enabled.
                A lot of drivers have their options described inside
                the Documentation/scsi/ sub-directory.
SDW     SoundWire support is enabled.
SECURITY Different security models are enabled.
SELINUX SELinux support is enabled.
SERIAL  Serial support is enabled.
SH      SuperH architecture is enabled.
SMP     The kernel is an SMP kernel.
SPARC   Sparc architecture is enabled.
SUSPEND System suspend states are enabled.
SWSUSP  Software suspend (hibernation) is enabled.
TPM     TPM drivers are enabled.
UMS     USB Mass Storage support is enabled.
USB     USB support is enabled.
NVME    NVMe support is enabled
USBHID  USB Human Interface Device support is enabled.
V4L     Video For Linux support is enabled.
VGA     The VGA console has been enabled.
VMMIO   Driver for memory mapped virtio devices is enabled.
VT      Virtual terminal support is enabled.
WDT     Watchdog support is enabled.
X86-32  X86-32, aka i386 architecture is enabled.
X86-64  X86-64 architecture is enabled.
X86     Either 32-bit or 64-bit x86 (same as X86-32+X86-64)
```

```
        X86_UV  SGI UV support is enabled.
        XEN     Xen support is enabled
        XTENSA  xtensa architecture is enabled.

In addition, the following text indicates that the option

        BOOT    Is a boot loader parameter.
        BUGS=   Relates to possible processor bugs on the said processor.
        KNL     Is a kernel start-up parameter.


Kernel parameters

        accept_memory=  [MM]
                        Format: { eager | lazy }
                        default: lazy
                        By default, unaccepted memory is accepted lazily to
                        avoid prolonged boot times. The lazy option will add
                        some runtime overhead until all memory is eventually
                        accepted. In most cases the overhead is negligible.
                        For some workloads or for debugging purposes
                        accept_memory=eager can be used to accept all memory
                        at once during boot.

        acpi=           [HW,ACPI,X86,ARM64,RISCV64,EARLY]
                        Advanced Configuration and Power Interface
                        Format: { force | on | off | strict | noirq | rsdt |
                                  copy_dsdt | nospcr }
                        force -- enable ACPI if default was off
                        on -- enable ACPI but allow fallback to DT [arm64,riscv64]
                        off -- disable ACPI if default was on
                        noirq -- do not use ACPI for IRQ routing
                        strict -- Be less tolerant of platforms that are not
                                strictly ACPI specification compliant.
                        rsdt -- prefer RSDT over (default) XSDT
                        copy_dsdt -- copy DSDT to memory
                        nocmcff -- Disable firmware first mode for corrected
                        errors. This disables parsing the HEST CMC error
                        source to check if firmware has set the FF flag. This
                        may result in duplicate corrected error reports.
                        nospcr -- disable console in ACPI SPCR table as
                                default _serial_ console on ARM64
                        spcr -- enable console in ACPI SPCR table as
                                default _serial_ console on x86
                        For ARM64, ONLY "acpi=off", "acpi=on", "acpi=force" or
                        "acpi=nospcr" are available
                        For RISCV64, ONLY "acpi=off", "acpi=on" or "acpi=force"
                        are available

                        See also Documentation/power/runtime_pm.rst, pci=noacpi

        acpi_apic_instance=     [ACPI,IOAPIC,EARLY]
```

```
                Format: <int>
                2: use 2nd APIC table, if available
                1,0: use 1st APIC table
                default: 0

    acpi_backlight= [HW,ACPI]
                { vendor | video | native | none }
                If set to vendor, prefer vendor-specific driver
                (e.g. thinkpad_acpi, sony_acpi, etc.) instead
                of the ACPI video.ko driver.
                If set to video, use the ACPI video.ko driver.
                If set to native, use the device's native backlight mode.
                If set to none, disable the ACPI backlight interface.

    acpi_force_32bit_fadt_addr [ACPI,EARLY]
                force FADT to use 32 bit addresses rather than the
                64 bit X_* addresses. Some firmware have broken 64
                bit addresses for force ACPI ignore these and use
                the older legacy 32 bit addresses.

    acpica_no_return_repair [HW, ACPI]
                Disable AML predefined validation mechanism
                This mechanism can repair the evaluation result to make
                the return objects more ACPI specification compliant.
                This option is useful for developers to identify the
                root cause of an AML interpreter issue when the issue
                has something to do with the repair mechanism.

    acpi.debug_layer=       [HW,ACPI,ACPI_DEBUG]
    acpi.debug_level=       [HW,ACPI,ACPI_DEBUG]
                Format: <int>
                CONFIG_ACPI_DEBUG must be enabled to produce any ACPI
                debug output.  Bits in debug_layer correspond to a
                _COMPONENT in an ACPI source file, e.g.,
                    #define _COMPONENT ACPI_EVENTS
                Bits in debug_level correspond to a level in
                ACPI_DEBUG_PRINT statements, e.g.,
                    ACPI_DEBUG_PRINT((ACPI_DB_INFO, ...
                The debug_level mask defaults to "info".  See
                Documentation/firmware-guide/acpi/debug.rst for more inform
                debug layers and levels.

                Enable processor driver info messages:
                    acpi.debug_layer=0x20000000
                Enable AML "Debug" output, i.e., stores to the Debug
                object while interpreting AML:
                    acpi.debug_layer=0xffffffff acpi.debug_level=0x2
                Enable all messages related to ACPI hardware:
                    acpi.debug_layer=0x2 acpi.debug_level=0xffffffff

                Some values produce so much output that the system is
                unusable.  The "log_buf_len" parameter may be useful
```

```
                if you need to capture more output.

        acpi.poweroff_on_fatal= [ACPI]
                {0 | 1}
                Causes the system to poweroff when the ACPI bytecode signal
                a fatal error. The default value of this setting is 1.
                Overriding this value should only be done for diagnosing
                ACPI firmware problems, as the system might behave erratica
                after having encountered a fatal ACPI error.

        acpi_enforce_resources= [ACPI]
                { strict | lax | no }
                Check for resource conflicts between native drivers
                and ACPI OperationRegions (SystemIO and SystemMemory
                only). IO ports and memory declared in ACPI might be
                used by the ACPI subsystem in arbitrary AML code and
                can interfere with legacy drivers.
                strict (default): access to resources claimed by ACPI
                is denied; legacy drivers trying to access reserved
                resources will fail to bind to device using them.
                lax: access to resources claimed by ACPI is allowed;
                legacy drivers trying to access reserved resources
                will bind successfully but a warning message is logged.
                no: ACPI OperationRegions are not marked as reserved,
                no further checks are performed.

        acpi_force_table_verification   [HW,ACPI,EARLY]
                Enable table checksum verification during early stage.
                By default, this is disabled due to x86 early mapping
                size limitation.

        acpi_irq_balance [HW,ACPI]
                ACPI will balance active IRQs
                default in APIC mode

        acpi_irq_nobalance [HW,ACPI]
                ACPI will not move active IRQs (default)
                default in PIC mode

        acpi_irq_isa=   [HW,ACPI] If irq_balance, mark listed IRQs used by ISA
                Format: <irq>,<irq>...

        acpi_irq_pci=   [HW,ACPI] If irq_balance, clear listed IRQs for
                use by PCI
                Format: <irq>,<irq>...

        acpi_mask_gpe=  [HW,ACPI]
                Due to the existence of _Lxx/_Exx, some GPEs triggered
                by unsupported hardware/firmware features can result in
                GPE floodings that cannot be automatically disabled by
                the GPE dispatcher.
                This facility can be used to prevent such uncontrolled
```

```
                GPE floodings.
                Format: <byte> or <bitmap-list>

    acpi_no_auto_serialize  [HW,ACPI]
                Disable auto-serialization of AML methods
                AML control methods that contain the opcodes to create
                named objects will be marked as "Serialized" by the
                auto-serialization feature.
                This feature is enabled by default.
                This option allows to turn off the feature.

    acpi_no_memhotplug [ACPI] Disable memory hotplug.  Useful for kdump
                       kernels.

    acpi_no_static_ssdt     [HW,ACPI,EARLY]
                Disable installation of static SSDTs at early boot time
                By default, SSDTs contained in the RSDT/XSDT will be
                installed automatically and they will appear under
                /sys/firmware/acpi/tables.
                This option turns off this feature.
                Note that specifying this option does not affect
                dynamic table installation which will install SSDT
                tables to /sys/firmware/acpi/tables/dynamic.

    acpi_no_watchdog        [HW,ACPI,WDT]
                Ignore the ACPI-based watchdog interface (WDAT) and let
                a native driver control the watchdog device instead.

    acpi_rsdp=      [ACPI,EFI,KEXEC,EARLY]
                Pass the RSDP address to the kernel, mostly used
                on machines running EFI runtime service to boot the
                second kernel for kdump.

    acpi_os_name=   [HW,ACPI] Tell ACPI BIOS the name of the OS
                Format: To spoof as Windows 98: ="Microsoft Windows"

    acpi_rev_override [ACPI] Override the _REV object to return 5 (instead
                of 2 which is mandated by ACPI 6) as the supported ACPI
                specification revision (when using this switch, it may
                be necessary to carry out a cold reboot _twice_ in a
                row to make it take effect on the platform firmware).

    acpi_osi=       [HW,ACPI] Modify list of supported OS interface strings
                acpi_osi="string1"      # add string1
                acpi_osi="!string2"     # remove string2
                acpi_osi=!*             # remove all strings
                acpi_osi=!              # disable all built-in OS vendor
                                          strings
                acpi_osi=!!             # enable all built-in OS vendor
                                          strings
                acpi_osi=               # disable all strings
```

```
                'acpi_osi=!' can be used in combination with single or
                multiple 'acpi_osi="string1"' to support specific OS
                vendor string(s).  Note that such command can only
                affect the default state of the OS vendor strings, thus
                it cannot affect the default state of the feature group
                strings and the current state of the OS vendor strings,
                specifying it multiple times through kernel command line
                is meaningless.  This command is useful when one do not
                care about the state of the feature group strings which
                should be controlled by the OSPM.
                Examples:
                  1. 'acpi_osi=! acpi_osi="Windows 2000"' is equivalent
                     to 'acpi_osi="Windows 2000" acpi_osi=!', they all
                     can make '_OSI("Windows 2000")' TRUE.

                'acpi_osi=' cannot be used in combination with other
                'acpi_osi=' command lines, the _OSI method will not
                exist in the ACPI namespace.  NOTE that such command can
                only affect the _OSI support state, thus specifying it
                multiple times through kernel command line is also
                meaningless.
                Examples:
                  1. 'acpi_osi=' can make 'CondRefOf(_OSI, Local1)'
                     FALSE.

                'acpi_osi=!*' can be used in combination with single or
                multiple 'acpi_osi="string1"' to support specific
                string(s).  Note that such command can affect the
                current state of both the OS vendor strings and the
                feature group strings, thus specifying it multiple times
                through kernel command line is meaningful.  But it may
                still not able to affect the final state of a string if
                there are quirks related to this string.  This command
                is useful when one want to control the state of the
                feature group strings to debug BIOS issues related to
                the OSPM features.
                Examples:
                  1. 'acpi_osi="Module Device" acpi_osi=!*' can make
                     '_OSI("Module Device")' FALSE.
                  2. 'acpi_osi=!* acpi_osi="Module Device"' can make
                     '_OSI("Module Device")' TRUE.
                  3. 'acpi_osi=! acpi_osi=!* acpi_osi="Windows 2000"' is
                     equivalent to
                     'acpi_osi=!* acpi_osi=! acpi_osi="Windows 2000"'
                     and
                     'acpi_osi=!* acpi_osi="Windows 2000" acpi_osi=!',
                     they all will make '_OSI("Windows 2000")' TRUE.

        acpi_pm_good    [X86]
                Override the pmtimer bug detection: force the kernel
                to assume that this machine's pmtimer latches its value
                and always returns good values.
```

```
acpi_sci=       [HW,ACPI,EARLY] ACPI System Control Interrupt trigger mode
                Format: { level | edge | high | low }

acpi_skip_timer_override [HW,ACPI,EARLY]
                Recognize and ignore IRQ0/pin2 Interrupt Override.
                For broken nForce2 BIOS resulting in XT-PIC timer.

acpi_sleep=     [HW,ACPI] Sleep options
                Format: { s3_bios, s3_mode, s3_beep, s4_hwsig,
                          s4_nohwsig, old_ordering, nonvs,
                          sci_force_enable, nobl }
                See Documentation/power/video.rst for information on
                s3_bios and s3_mode.
                s3_beep is for debugging; it makes the PC's speaker beep
                as soon as the kernel's real-mode entry point is called.
                s4_hwsig causes the kernel to check the ACPI hardware
                signature during resume from hibernation, and gracefully
                refuse to resume if it has changed. This complies with
                the ACPI specification but not with reality, since
                Windows does not do this and many laptops do change it
                on docking. So the default behaviour is to allow resume
                and simply warn when the signature changes, unless the
                s4_hwsig option is enabled.
                s4_nohwsig prevents ACPI hardware signature from being
                used (or even warned about) during resume.
                old_ordering causes the ACPI 1.0 ordering of the _PTS
                control method, with respect to putting devices into
                low power states, to be enforced (the ACPI 2.0 ordering
                of _PTS is used by default).
                nonvs prevents the kernel from saving/restoring the
                ACPI NVS memory during suspend/hibernation and resume.
                sci_force_enable causes the kernel to set SCI_EN directly
                on resume from S1/S3 (which is against the ACPI spec,
                but some broken systems don't work without it).
                nobl causes the internal blacklist of systems known to
                behave incorrectly in some ways with respect to system
                suspend and resume to be ignored (use wisely).

acpi_use_timer_override [HW,ACPI,EARLY]
                Use timer override. For some broken Nvidia NF5 boards
                that require a timer override, but don't have HPET

add_efi_memmap  [EFI,X86,EARLY] Include EFI memory map in
                kernel's map of available physical RAM.

agp=            [AGP]
                { off | try_unsupported }
                off: disable AGP support
                try_unsupported: try to drive unsupported chipsets
                        (may crash computer or cause data corruption)
```

```
        ALSA            [HW,ALSA]
                        See Documentation/sound/alsa-configuration.rst

        alignment=      [KNL,ARM]
                        Allow the default userspace alignment fault handler
                        behaviour to be specified.  Bit 0 enables warnings,
                        bit 1 enables fixups, and bit 2 sends a segfault.

        align_va_addr=  [X86-64]
                        Align virtual addresses by clearing slice [14:12] when
                        allocating a VMA at process creation time. This option
                        gives you up to 3% performance improvement on AMD F15h
                        machines (where it is enabled by default) for a
                        CPU-intensive style benchmark, and it can vary highly in
                        a microbenchmark depending on workload and compiler.

                        32: only for 32-bit processes
                        64: only for 64-bit processes
                        on: enable for both 32- and 64-bit processes
                        off: disable for both 32- and 64-bit processes

        alloc_snapshot  [FTRACE]
                        Allocate the ftrace snapshot buffer on boot up when the
                        main buffer is allocated. This is handy if debugging
                        and you need to use tracing_snapshot() on boot up, and
                        do not want to use tracing_snapshot_alloc() as it needs
                        to be done where GFP_KERNEL allocations are allowed.

        allow_mismatched_32bit_el0 [ARM64,EARLY]
                        Allow execve() of 32-bit applications and setting of the
                        PER_LINUX32 personality on systems where only a strict
                        subset of the CPUs support 32-bit EL0. When this
                        parameter is present, the set of CPUs supporting 32-bit
                        EL0 is indicated by /sys/devices/system/cpu/aarch32_el0
                        and hot-unplug operations may be restricted.

                        See Documentation/arch/arm64/asymmetric-32bit.rst for more
                        information.

        amd_iommu=      [HW,X86-64]
                        Pass parameters to the AMD IOMMU driver in the system.
                        Possible values are:
                        fullflush - Deprecated, equivalent to iommu.strict=1
                        off       - do not initialize any AMD IOMMU found in
                                    the system
                        force_isolation - Force device isolation for all
                                          devices. The IOMMU driver is not
                                          allowed anymore to lift isolation
                                          requirements as needed. This option
                                          does not override iommu=pt
                        force_enable    - Force enable the IOMMU on platforms known
                                          to be buggy with IOMMU enabled. Use this
```

```
                                  option with care.
                pgtbl_v1        - Use v1 page table for DMA-API (Default).
                pgtbl_v2        - Use v2 page table for DMA-API.
                irtcachedis     - Disable Interrupt Remapping Table (IRT) c
                nohugepages     - Limit page-sizes used for v1 page-tables
                                  to 4 KiB.
                v2_pgsizes_only - Limit page-sizes used for v1 page-tables
                                  to 4KiB/2Mib/1GiB.


        amd_iommu_dump= [HW,X86-64]
                        Enable AMD IOMMU driver option to dump the ACPI table
                        for AMD IOMMU. With this option enabled, AMD IOMMU
                        driver will print ACPI tables for AMD IOMMU during
                        IOMMU initialization.

        amd_iommu_intr= [HW,X86-64]
                        Specifies one of the following AMD IOMMU interrupt
                        remapping modes:
                        legacy     - Use legacy interrupt remapping mode.
                        vapic      - Use virtual APIC mode, which allows IOMMU
                                     to inject interrupts directly into guest.
                                     This mode requires kvm-amd.avic=1.
                                     (Default when IOMMU HW support is present.)

        amd_pstate=     [X86,EARLY]
                        disable
                          Do not enable amd_pstate as the default
                          scaling driver for the supported processors
                        passive
                          Use amd_pstate with passive mode as a scaling driver.
                          In this mode autonomous selection is disabled.
                          Driver requests a desired performance level and platform
                          tries to match the same performance level if it is
                          satisfied by guaranteed performance level.
                        active
                          Use amd_pstate_epp driver instance as the scaling driver,
                          driver provides a hint to the hardware if software wants
                          to bias toward performance (0x0) or energy efficiency (0x
                          to the CPPC firmware. then CPPC power algorithm will
                          calculate the runtime workload and adjust the realtime co
                          frequency.
                        guided
                          Activate guided autonomous mode. Driver requests minimum
                          maximum performance level and the platform autonomously
                          selects a performance level in this range and appropriate
                          to the current workload.

        amd_prefcore=
                        [X86]
                        disable
                          Disable amd-pstate preferred core.
```

```
amd_dynamic_epp=
                [X86]
                disable
                  Disable amd-pstate dynamic EPP.
                enable
                  Enable amd-pstate dynamic EPP.

amijoy.map=     [HW,JOY] Amiga joystick support
                Map of devices attached to JOY0DAT and JOY1DAT
                Format: <a>,<b>
                See also Documentation/input/joydev/joystick.rst

analog.map=     [HW,JOY] Analog joystick and gamepad support
                Specifies type or capabilities of an analog joystick
                connected to one of 16 gameports
                Format: <type1>,<type2>,..<type16>

apc=            [HW,SPARC]
                Power management functions (SPARCstation-4/5 + deriv.)
                Format: noidle
                Disable APC CPU standby support. SPARCstation-Fox does
                not play well with APC CPU idle - disable it if you have
                APC and your system crashes randomly.

apic            [APIC,X86-64] Use IO-APIC. Default.

apic=           [APIC,X86,EARLY] Advanced Programmable Interrupt Controller
                Change the output verbosity while booting
                Format: { quiet (default) | verbose | debug }
                Change the amount of debugging information output
                when initialising the APIC and IO-APIC components.

apic_extnmi=    [APIC,X86,EARLY] External NMI delivery setting
                Format: { bsp (default) | all | none }
                bsp:  External NMI is delivered only to CPU 0
                all:  External NMIs are broadcast to all CPUs as a
                      backup of CPU 0
                none: External NMI is masked for all CPUs. This is
                      useful so that a dump capture kernel won't be
                      shot down by NMI

apicpmtimer     Do APIC timer calibration using the pmtimer. Implies
                apicmaintimer. Useful when your PIT timer is totally
                broken.

autoconf=       [IPV6]
                See Documentation/networking/ipv6.rst.

apm=            [APM] Advanced Power Management
                See header of arch/x86/kernel/apm_32.c.
```

```
        apparmor=       [APPARMOR] Disable or enable AppArmor at boot time
                        Format: { "0" | "1" }
                        See security/apparmor/Kconfig help text
                        0 -- disable.
                        1 -- enable.
                        Default value is set via kernel config option.

        arm64.no32bit_el0 [ARM64] Unconditionally disable the execution of
                        32 bit applications.

        arm64.nobti     [ARM64] Unconditionally disable Branch Target
                        Identification support

        arm64.nogcs     [ARM64] Unconditionally disable Guarded Control Stack
                        support

        arm64.nomops    [ARM64] Unconditionally disable Memory Copy and Memory
                        Set instructions support

        arm64.nompam    [ARM64] Unconditionally disable Memory Partitioning And
                        Monitoring support

        arm64.nomte     [ARM64] Unconditionally disable Memory Tagging Extension
                        support

        arm64.nopauth   [ARM64] Unconditionally disable Pointer Authentication
                        support

        arm64.nosme     [ARM64] Unconditionally disable Scalable Matrix
                        Extension support

        arm64.nosve     [ARM64] Unconditionally disable Scalable Vector
                        Extension support

        ataflop=        [HW,M68k]

        atarimouse=     [HW,MOUSE] Atari Mouse

        atkbd.extra=    [HW] Enable extra LEDs and keys on IBM RapidAccess,
                        EzKey and similar keyboards

        atkbd.reset=    [HW] Reset keyboard during initialization

        atkbd.set=      [HW] Select keyboard code set
                        Format: <int> (2 = AT (default), 3 = PS/2)

        atkbd.scroll=   [HW] Enable scroll wheel on MS Office and similar
                        keyboards

        atkbd.softraw=  [HW] Choose between synthetic and real raw mode
                        Format: <bool> (0 = real, 1 = synthetic (default))
```

```
	atkbd.softrepeat= [HW]
			Use software keyboard repeat

	audit=		[KNL] Enable the audit sub-system
			Format: { "0" | "1" | "off" | "on" }
			0 | off - kernel audit is disabled and can not be
			    enabled until the next reboot
			unset - kernel audit is initialized but disabled and
			    will be fully enabled by the userspace auditd.
			1 | on - kernel audit is initialized and partially
			    enabled, storing at most audit_backlog_limit
			    messages in RAM until it is fully enabled by the
			    userspace auditd.
			Default: unset

	audit_backlog_limit= [KNL] Set the audit queue size limit.
			Format: <int> (must be >=0)
			Default: 64

	bau=		[X86_UV] Enable the BAU on SGI UV.  The default
			behavior is to disable the BAU (i.e. bau=0).
			Format: { "0" | "1" }
			0 - Disable the BAU.
			1 - Enable the BAU.
			unset - Disable the BAU.

	bdev_allow_write_mounted=
			Format: <bool>
			Control the ability to open a mounted block device
			for writing, i.e., allow / disallow writes that bypass
			the FS. This was implemented as a means to prevent
			fuzzers from crashing the kernel by overwriting the
			metadata underneath a mounted FS without its awareness.
			This also prevents destructive formatting of mounted
			filesystems by naive storage tooling that don't use
			O_EXCL. Default is Y and can be changed through the
			Kconfig option CONFIG_BLK_DEV_WRITE_MOUNTED.

	bert_disable	[ACPI]
			Disable BERT OS support on buggy BIOSes.

	bgrt_disable	[ACPI,X86,EARLY]
			Disable BGRT to avoid flickering OEM logo.

	blkdevparts=	Manual partition parsing of block device(s) for
			embedded devices based on command line input.
			See Documentation/block/cmdline-partition.rst

	boot_delay=	[KNL,EARLY]
			Milliseconds to delay each printk during boot.
			Only works if CONFIG_BOOT_PRINTK_DELAY is enabled,
			and you may also have to specify "lpj=".  Boot_delay
```

```
                values larger than 10 seconds (10000) are assumed
                erroneous and ignored.
                Format: integer

    bootconfig  [KNL,EARLY]
                Extended command line options can be added to an initrd
                and this will cause the kernel to look for it.

                See Documentation/admin-guide/bootconfig.rst

    bttv.card=  [HW,V4L] bttv (bt848 + bt878 based grabber cards)
    bttv.radio= Most important insmod options are available as
                kernel args too.
    bttv.pll=   See Documentation/admin-guide/media/bttv.rst
    bttv.tuner=

    bulk_remove=off [PPC]  This parameter disables the use of the pSeries
                firmware feature for flushing multiple hpte entries
                at a time.

    c101=       [NET] Moxa C101 synchronous serial card

    cachesize=  [BUGS=X86-32] Override level 2 CPU cache size detection.
                Sometimes CPU hardware bugs make them report the cache
                size incorrectly. The kernel will attempt work arounds
                to fix known problems, but for some CPUs it is not
                possible to determine what the correct size should be.
                This option provides an override for these situations.

    carrier_timeout=
                [NET] Specifies amount of time (in seconds) that
                the kernel should wait for a network carrier. By default
                it waits 120 seconds.

    ca_keys=    [KEYS] This parameter identifies a specific key(s) on
                the system trusted keyring to be used for certificate
                trust validation.
                format: { id:<keyid> | builtin }

    cca=        [MIPS,EARLY] Override the kernel pages' cache coherency
                algorithm.  Accepted values range from 0 to 7
                inclusive. See arch/mips/include/asm/pgtable-bits.h
                for platform specific values (SB1, Loongson3 and
                others).

    ccw_timeout_log [S390]
                See Documentation/arch/s390/common_io.rst for details.

    cfi=        [X86-64] Set Control Flow Integrity checking features
                when CONFIG_FINEIBT is enabled.
                Format: feature[,feature...]
                Default: auto
```

```
                auto:     Use FineIBT if IBT available, otherwise kCFI.
                          Under FineIBT, enable "paranoid" mode when
                          FRED is not available.
                off:      Turn off CFI checking.
                kcfi:     Use kCFI (disable FineIBT).
                fineibt:  Use FineIBT (even if IBT not available).
                norand:   Do not re-randomize CFI hashes.
                paranoid: Add caller hash checking under FineIBT.
                bhi:      Enable register poisoning to stop speculation
                          across FineIBT. (Disabled by default.)
                warn:     Do not enforce CFI checking: warn only.
                debug:    Report CFI initialization details.

    cgroup_disable= [KNL] Disable a particular controller or optional feature
                Format: {name of the controller(s) or feature(s) to disable
                The effects of cgroup_disable=foo are:
                - foo isn't auto-mounted if you mount all cgroups in
                  a single hierarchy
                - foo isn't visible as an individually mountable
                  subsystem
                - if foo is an optional feature then the feature is
                  disabled and corresponding cgroup files are not
                  created
                {Currently only "memory" controller deal with this and
                cut the overhead, others just disable the usage. So
                only cgroup_disable=memory is actually worthy}
                Specifying "pressure" disables per-cgroup pressure
                stall information accounting feature

    cgroup_no_v1=   [KNL] Disable cgroup controllers and named hierarchies in v
                Format: { { controller | "all" | "named" }
                          [,{ controller | "all" | "named" }...] }
                Like cgroup_disable, but only applies to cgroup v1;
                the blacklisted controllers remain available in cgroup2.
                "all" blacklists all controllers and "named" disables
                named mounts. Specifying both "all" and "named" disables
                all v1 hierarchies.

    cgroup_v1_proc= [KNL] Show also missing controllers in /proc/cgroups
                Format: { "true" | "false" }
                /proc/cgroups lists only v1 controllers by default.
                This compatibility option enables listing also v2
                controllers (whose v1 code is not compiled!), so that
                semi-legacy software can check this file to decide
                about usage of v2 (sic) controllers.

    cgroup_favordynmods= [KNL] Enable or Disable favordynmods.
                Format: { "true" | "false" }
                Defaults to the value of CONFIG_CGROUP_FAVOR_DYNMODS.

    cgroup.memory=  [KNL] Pass options to the cgroup memory controller.
```

```
                Format: <string>
                nosocket -- Disable socket memory accounting.
                nokmem -- Disable kernel memory accounting.
                nobpf -- Disable BPF memory accounting.

    check_pages=    [MM,EARLY] Enable sanity checking of pages after
                allocations / before freeing. This adds checks to catch
                double-frees, use-after-frees, and other sources of
                page corruption by inspecting page internals (flags,
                mapcount/refcount, memcg_data, etc.).
                Format: { "0" | "1" }
                Default: 0 (1 if CONFIG_DEBUG_VM is set)

    checkreqprot=   [SELINUX] Set initial checkreqprot flag value.
                Format: { "0" | "1" }
                See security/selinux/Kconfig help text.
                0 -- check protection applied by kernel (includes
                        any implied execute protection).
                1 -- check protection requested by application.
                Default value is set via a kernel config option.
                Value can be changed at runtime via
                        /sys/fs/selinux/checkreqprot.
                Setting checkreqprot to 1 is deprecated.

    cio_ignore=     [S390]
                See Documentation/arch/s390/common_io.rst for details.

    clk_ignore_unused
                [CLK]
                Prevents the clock framework from automatically gating
                clocks that have not been explicitly enabled by a Linux
                device driver but are enabled in hardware at reset or
                by the bootloader/firmware. Note that this does not
                force such clocks to be always-on nor does it reserve
                those clocks in any way. This parameter is useful for
                debug and development, but should not be needed on a
                platform with proper driver support.  For more
                information, see Documentation/driver-api/clk.rst.

    clock=          [BUGS=X86-32, HW] gettimeofday clocksource override.
                [Deprecated]
                Forces specified clocksource (if available) to be used
                when calculating gettimeofday(). If specified
                clocksource is not available, it defaults to PIT.
                Format: { pit | tsc | cyclone | pmtmr }

    clocksource=    Override the default clocksource
                Format: <string>
                Override the default clocksource and use the clocksource
                with the name specified.
                Some clocksource names to choose from, depending on
                the platform:
```

```
                [all] jiffies (this is the base, fallback clocksource)
                [ACPI] acpi_pm
                [ARM] imx_timer1,OSTS,netx_timer,mpu_timer2,
                        pxa_timer,timer3,32k_counter,timer0_1
                [X86-32] pit,hpet,tsc;
                        scx200_hrt on Geode; cyclone on IBM x440
                [MIPS] MIPS
                [PARISC] cr16
                [S390] tod
                [SH] SuperH
                [SPARC64] tick
                [X86-64] hpet,tsc

    clocksource.arm_arch_timer.evtstrm=
                [ARM,ARM64,EARLY]
                Format: <bool>
                Enable/disable the eventstream feature of the ARM
                architected timer so that code using WFE-based polling
                loops can be debugged more effectively on production
                systems.

    clocksource.verify_n_cpus= [KNL]
                Limit the number of CPUs checked for clocksources
                marked with CLOCK_SOURCE_VERIFY_PERCPU that
                are marked unstable due to excessive skew.
                A negative value says to check all CPUs, while
                zero says not to check any.  Values larger than
                nr_cpu_ids are silently truncated to nr_cpu_ids.
                The actual CPUs are chosen randomly, with
                no replacement if the same CPU is chosen twice.

    clocksource-wdtest.holdoff= [KNL]
                Set the time in seconds that the clocksource
                watchdog test waits before commencing its tests.
                Defaults to zero when built as a module and to
                10 seconds when built into the kernel.

    cma=nn[MG]@[start[MG][-end[MG]]]
                [KNL,CMA,EARLY]
                Sets the size of kernel global memory area for
                contiguous memory allocations and optionally the
                placement constraint by the physical address range of
                memory allocations. A value of 0 disables CMA
                altogether. For more information, see
                kernel/dma/contiguous.c

    cma_pernuma=nn[MG]
                [KNL,CMA,EARLY]
                Sets the size of kernel per-numa memory area for
                contiguous memory allocations. A value of 0 disables
                per-numa CMA altogether. And If this option is not
                specified, the default value is 0.
```

```
                With per-numa CMA enabled, DMA users on node nid will
                first try to allocate buffer from the pernuma area
                which is located in node nid, if the allocation fails,
                they will fallback to the global default memory area.

    numa_cma=<node>:nn[MG][,<node>:nn[MG]]
                [KNL,CMA,EARLY]
                Sets the size of kernel numa memory area for
                contiguous memory allocations. It will reserve CMA
                area for the specified node.

                If it is setup together with upper 'cmd_pernuma='
                (unlikely), its size setting takes priority for the
                specified numa nodes.

                With numa CMA enabled, DMA users on node nid will
                first try to allocate buffer from the numa area
                which is located in node nid, if the allocation fails,
                they will fallback to the global default memory area.

    cmo_free_hint=  [PPC] Format: { yes | no }
                Specify whether pages are marked as being inactive
                when they are freed.  This is used in CMO environments
                to determine OS memory pressure for page stealing by
                a hypervisor.
                Default: yes

    coherent_pool=nn[KMG]   [ARM,KNL,EARLY]
                Sets the size of memory pool for coherent, atomic dma
                allocations, by default set to 256K.

    condev=         [HW,S390] console device
    conmode=

    con3215_drop=   [S390,EARLY] 3215 console drop mode.
                Format: y|n|Y|N|1|0
                When set to true, drop data on the 3215 console when
                the console buffer is full. In this case the
                operator using a 3270 terminal emulator (for example
                x3270) does not have to enter the clear key for the
                console output to advance and the kernel to continue.
                This leads to a much faster boot time when a 3270
                terminal emulator is active. If no 3270 terminal
                emulator is used, this parameter has no effect.

    console=        [KNL] Output console device and options.

            tty<n>  Use the virtual console device <n>.

            ttyS<n>[,options]
            ttyUSB0[,options]
                    Use the specified serial port.  The options are of
```

```
                the form "bbbbpnf", where "bbbb" is the baud rate,
                "p" is parity ("n", "o", or "e"), "n" is number of
                bits, and "f" is flow control ("r" for RTS or
                omit it).  Default is "9600n8".

                See Documentation/admin-guide/serial-console.rst for more
                information.  See
                Documentation/networking/netconsole.rst for an
                alternative.

        <DEVNAME>:<n>.<n>[,options]
                Use the specified serial port on the serial core bus.
                The addressing uses DEVNAME of the physical serial port
                device, followed by the serial core controller instance,
                and the serial port instance. The options are the same
                as documented for the ttyS addressing above.

                The mapping of the serial ports to the tty instances
                can be viewed with:

                $ ls -d /sys/bus/serial-base/devices/*:*.*/tty/*
                /sys/bus/serial-base/devices/00:04:0.0/tty/ttyS0

                In the above example, the console can be addressed with
                console=00:04:0.0. Note that a console addressed this
                way will only get added when the related device driver
                is ready. The use of an earlycon parameter in addition to
                the console may be desired for console output early on.

        uart[8250],io,<addr>[,options]
        uart[8250],mmio,<addr>[,options]
        uart[8250],mmio16,<addr>[,options]
        uart[8250],mmio32,<addr>[,options]
        uart[8250],0x<addr>[,options]
                Start an early, polled-mode console on the 8250/16550
                UART at the specified I/O port or MMIO address,
                switching to the matching ttyS device later.
                MMIO inter-register address stride is either 8-bit
                (mmio), 16-bit (mmio16), or 32-bit (mmio32).
                If none of [io|mmio|mmio16|mmio32], <addr> is assumed
                to be equivalent to 'mmio'. 'options' are specified in
                the same format described for ttyS above; if unspecified,
                the h/w is not re-initialized.

        hvc<n>  Use the hypervisor console device <n>. This is for
                both Xen and PowerPC hypervisors.

        { null | "" }
                Use to disable console output, i.e., to have kernel
                console messages discarded.
                This must be the only console= parameter used on the
                kernel command line.
```

```
        If the device connected to the port is not a TTY but a braille
        device, prepend "brl," before the device type, for instance
                console=brl,ttyS0
        For now, only VisioBraille is supported.

    console_msg_format=
                [KNL] Change console messages format
        default
                By default we print messages on consoles in
                "[time stamp] text\n" format (time stamp may not be
                printed, depending on CONFIG_PRINTK_TIME or
                `printk_time' param).
        syslog
                Switch to syslog format: "<%u>[time stamp] text\n"
                IOW, each message will have a facility and loglevel
                prefix. The format is similar to one used by syslog()
                syscall, or to executing "dmesg -S --raw" or to reading
                from /proc/kmsg.

    consoleblank=   [KNL] The console blank (screen saver) timeout in
                seconds. A value of 0 disables the blank timer.
                Defaults to 0.

    coredump_filter=
                [KNL] Change the default value for
                /proc/<pid>/coredump_filter.
                See also Documentation/filesystems/proc.rst.

    coresight_cpu_debug.enable
                [ARM,ARM64]
                Format: <bool>
                Enable/disable the CPU sampling based debugging.
                0: default value, disable debugging
                1: enable debugging at boot time

    cpcihp_generic= [HW,PCI] Generic port I/O CompactPCI driver
                Format:
                <first_slot>,<last_slot>,<port>,<enum_bit>[,<debug>]

    cpuidle.off=1   [CPU_IDLE]
                disable the cpuidle sub-system

    cpuidle.governor=
                [CPU_IDLE] Name of the cpuidle governor to use.

    cpufreq.off=1   [CPU_FREQ]
                disable the cpufreq sub-system

    cpufreq.default_governor=
                [CPU_FREQ] Name of the default cpufreq governor or
                policy to use. This governor must be registered in the
```

```
                kernel before the cpufreq driver probes.

        cpu_init_udelay=N
                [X86,EARLY] Delay for N microsec between assert and de-asse
                of APIC INIT to start processors.  This delay occurs
                on every CPU online, such as boot, and resume from suspend.
                Default: 10000

        cpuhp.parallel=
                [SMP] Enable/disable parallel bringup of secondary CPUs
                Format: <bool>
                Default is enabled if CONFIG_HOTPLUG_PARALLEL=y. Otherwise
                the parameter has no effect.

        crash_kexec_post_notifiers
                Only jump to kdump kernel after running the panic
                notifiers and dumping kmsg. This option increases
                the risks of a kdump failure, since some panic
                notifiers can make the crashed kernel more unstable.
                In configurations where kdump may not be reliable,
                running the panic notifiers could allow collecting
                more data on dmesg, like stack traces from other CPUS
                or extra data dumped by panic_print. Note that some
                configurations enable this option unconditionally,
                like Hyper-V, PowerPC (fadump) and AMD SEV-SNP.

        crashkernel=size[KMG][@offset[KMG]]
                [KNL,EARLY] Using kexec, Linux can switch to a 'crash kerne
                upon panic. This parameter reserves the physical
                memory region [offset, offset + size] for that kernel
                image. If '@offset' is omitted, then a suitable offset
                is selected automatically.
                [KNL, X86-64, ARM64, RISCV, LoongArch] Select a region
                under 4G first, and fall back to reserve region above
                4G when '@offset' hasn't been specified.
                See Documentation/admin-guide/kdump/kdump.rst for further d

        crashkernel=range1:size1[,range2:size2,...][@offset]
                [KNL] Same as above, but depends on the memory
                in the running system. The syntax of range is
                start-[end] where start and end are both
                a memory unit (amount[KMG]). See also
                Documentation/admin-guide/kdump/kdump.rst for an example.

        crashkernel=size[KMG],high
                [KNL, X86-64, ARM64, RISCV, LoongArch] range could be
                above 4G.
                Allow kernel to allocate physical memory region from top,
                so could be above 4G if system have more than 4G ram
                installed. Otherwise memory region will be allocated
                below 4G, if available.
                It will be ignored if crashkernel=X is specified.
```

```
        crashkernel=size[KMG],low
                        [KNL, X86-64, ARM64, RISCV, LoongArch] range under 4G.
                        When crashkernel=X,high is passed, kernel could allocate
                        physical memory region above 4G, that cause second kernel
                        crash on system that require some amount of low memory,
                        e.g. swiotlb requires at least 64M+32K low memory, also
                        enough extra low memory is needed to make sure DMA buffers
                        for 32-bit devices won't run out. Kernel would try to alloc
                        default size of memory below 4G automatically. The default
                        size is platform dependent.
                          --> x86: max(swiotlb_size_or_default() + 8MiB, 256MiB)
                          --> arm64: 128MiB
                          --> riscv: 128MiB
                          --> loongarch: 128MiB
                        This one lets the user specify own low range under 4G
                        for second kernel instead.
                        0: to disable low allocation.
                        It will be ignored when crashkernel=X,high is not used
                        or memory reserved is below 4G.
        crashkernel=size[KMG],cma
                        [KNL, X86, ppc] Reserve additional crash kernel memory from
                        CMA. This reservation is usable by the first system's
                        userspace memory and kernel movable allocations (memory
                        balloon, zswap). Pages allocated from this memory range
                        will not be included in the vmcore so this should not
                        be used if dumping of userspace memory is intended and
                        it has to be expected that some movable kernel pages
                        may be missing from the dump.

                        A standard crashkernel reservation, as described above,
                        is still needed to hold the crash kernel and initrd.

                        This option increases the risk of a kdump failure: DMA
                        transfers configured by the first kernel may end up
                        corrupting the second kernel's memory.

                        This reservation method is intended for systems that
                        can't afford to sacrifice enough memory for standard
                        crashkernel reservation and where less reliable and
                        possibly incomplete kdump is preferable to no kdump at
                        all.

        cryptomgr.notests
                        [KNL] Disable crypto self-tests

        cs89x0_dma=     [HW,NET]
                        Format: <dma>

        cs89x0_media=   [HW,NET]
                        Format: { rj45 | aui | bnc }

        csdlock_debug=  [KNL] Enable or disable debug add-ons of cross-CPU
```

```
                function call handling. When switched on,
                additional debug data is printed to the console
                in case a hanging CPU is detected, and that
                CPU is pinged again in order to try to resolve
                the hang situation.  The default value of this
                option depends on the CSD_LOCK_WAIT_DEBUG_DEFAULT
                Kconfig option.

    dasd=       [HW,NET]
                See header of drivers/s390/block/dasd_devmap.c.

    db9.dev[2|3]=   [HW,JOY] Multisystem joystick support via parallel port
                (one device per port)
                Format: <port#>,<type>
                See also Documentation/input/devices/joystick-parport.rst

    debug       [KNL,EARLY] Enable kernel debugging (events log level).

    debug_boot_weak_hash
                [KNL,EARLY] Enable printing [hashed] pointers early in the
                boot sequence.  If enabled, we use a weak hash instead
                of siphash to hash pointers.  Use this option if you are
                seeing instances of '(___ptrval___)') and need to see a
                value (hashed pointer) instead. Cryptographically
                insecure, please do not use on production kernels.

    debug_locks_verbose=
                [KNL] verbose locking self-tests
                Format: <int>
                Print debugging info while doing the locking API
                self-tests.
                Bitmask for the various LOCKTYPE_ tests. Defaults to 0
                (no extra messages), setting it to -1 (all bits set)
                will print _a_lot_ more information - normally only
                useful to lockdep developers.

    debug_objects   [KNL,EARLY] Enable object debugging

    debug_guardpage_minorder=
                [KNL,EARLY] When CONFIG_DEBUG_PAGEALLOC is set, this
                parameter allows control of the order of pages that will
                be intentionally kept free (and hence protected) by the
                buddy allocator. Bigger value increase the probability
                of catching random memory corruption, but reduce the
                amount of memory for normal system use. The maximum
                possible value is MAX_PAGE_ORDER/2.  Setting this
                parameter to 1 or 2 should be enough to identify most
                random memory corruption problems caused by bugs in
                kernel or driver code when a CPU writes to (or reads
                from) a random memory location. Note that there exists
                a class of memory corruptions problems caused by buggy
                H/W or F/W or by drivers badly programming DMA
```

```
                (basically when memory is written at bus level and the
                CPU MMU is bypassed) which are not detectable by
                CONFIG_DEBUG_PAGEALLOC, hence this option will not
                help tracking down these problems.

    debug_pagealloc=
                [KNL,EARLY] When CONFIG_DEBUG_PAGEALLOC is set, this parame
                enables the feature at boot time. By default, it is
                disabled and the system will work mostly the same as a
                kernel built without CONFIG_DEBUG_PAGEALLOC.
                Note: to get most of debug_pagealloc error reports, it's
                useful to also enable the page_owner functionality.
                on: enable the feature

    debugfs=    [KNL,EARLY] This parameter enables what is exposed to
                userspace and debugfs internal clients.
                Format: { on, off }
                on:     All functions are enabled.
                off:    Filesystem is not registered and clients
                        get a -EPERM as result when trying to register file
                        or directories within debugfs.
                        This is equivalent of the runtime functionality if
                        debugfs was not enabled in the kernel at all.
                Default value is set in build-time with a kernel configurat

    debugpat    [X86] Enable PAT debugging

    default_hugepagesz=
                [HW] The size of the default HugeTLB page. This is
                the size represented by the legacy /proc/ hugepages
                APIs.  In addition, this is the default hugetlb size
                used for shmget(), mmap() and mounting hugetlbfs
                filesystems.  If not specified, defaults to the
                architecture's default huge page size.  Huge page
                sizes are architecture dependent.  See also
                Documentation/admin-guide/mm/hugetlbpage.rst.
                Format: size[KMG]

    deferred_probe_timeout=
                [KNL] Debugging option to set a timeout in seconds for
                deferred probe to give up waiting on dependencies to
                probe. Only specific dependencies (subsystems or
                drivers) that have opted in will be ignored. A timeout
                of 0 will timeout at the end of initcalls; a negative
                value is treated as an infinite timeout value. If the
                timeout hasn't expired, it'll be restarted by each
                successful driver registration. This option will also
                dump out devices still on the deferred probe list after
                retrying.

    delayacct   [KNL] Enable per-task delay accounting
```

```
	dell_smm_hwmon.ignore_dmi=
	                [HW] Continue probing hardware even if DMI data
	                indicates that the driver is running on unsupported
	                hardware.

	dell_smm_hwmon.force=
	                [HW] Activate driver even if SMM BIOS signature does
	                not match list of supported models and enable otherwise
	                blacklisted features.

	dell_smm_hwmon.power_status=
	                [HW] Report power status in /proc/i8k
	                (disabled by default).

	dell_smm_hwmon.restricted=
	                [HW] Allow controlling fans only if SYS_ADMIN
	                capability is set.

	dell_smm_hwmon.fan_mult=
	                [HW] Factor to multiply fan speed with.

	dell_smm_hwmon.fan_max=
	                [HW] Maximum configurable fan speed.

	dfltcc=         [HW,S390]
	                Format: { on | off | def_only | inf_only | always }
	                on:       s390 zlib hardware support for compression on
	                          level 1 and decompression (default)
	                off:      No s390 zlib hardware support
	                def_only: s390 zlib hardware support for deflate
	                          only (compression on level 1)
	                inf_only: s390 zlib hardware support for inflate
	                          only (decompression)
	                always:   Same as 'on' but ignores the selected compression
	                          level always using hardware support (used for deb

	dhash_entries=  [KNL]
	                Set number of hash buckets for dentry cache.

	disable_1tb_segments [PPC,EARLY]
	                Disables the use of 1TB hash page table segments. This
	                causes the kernel to fall back to 256MB segments which
	                can be useful when debugging issues that require an SLB
	                miss to occur.

	disable=        [IPV6]
	                See Documentation/networking/ipv6.rst.

	disable_radix   [PPC,EARLY]
	                Disable RADIX MMU mode on POWER9

	disable_tlbie   [PPC]
```

```
                Disable TLBIE instruction. Currently does not work
                with KVM, with HASH MMU, or with coherent accelerators.

        disable_ddw     [PPC/PSERIES,EARLY]
                Disable Dynamic DMA Window support. Use this
                to workaround buggy firmware.

        disable_ipv6=   [IPV6]
                See Documentation/networking/ipv6.rst.

        disable_mtrr_cleanup [X86,EARLY]
                The kernel tries to adjust MTRR layout from continuous
                to discrete, to make X server driver able to add WB
                entry later. This parameter disables that.

        disable_mtrr_trim [X86, Intel and AMD only,EARLY]
                By default the kernel will trim any uncacheable
                memory out of your available memory pool based on
                MTRR settings.  This parameter disables that behavior,
                possibly causing your machine to run very slowly.

        disable_timer_pin_1 [X86,EARLY]
                Disable PIN 1 of APIC timer
                Can be useful to work around chipset bugs.

        dis_ucode_ldr   [X86] Disable the microcode loader.

        dma_debug=off   If the kernel is compiled with DMA_API_DEBUG support,
                this option disables the debugging code at boot.

        dma_debug_entries=<number>
                This option allows to tune the number of preallocated
                entries for DMA-API debugging code. One entry is
                required per DMA-API allocation. Use this if the
                DMA-API debugging code disables itself because the
                architectural default is too low.

        dma_debug_driver=<driver_name>
                With this option the DMA-API debugging driver
                filter feature can be enabled at boot time. Just
                pass the driver to filter for as the parameter.
                The filter can be disabled or changed to another
                driver later using sysfs.

        reg_file_data_sampling=
                [X86] Controls mitigation for Register File Data
                Sampling (RFDS) vulnerability. RFDS is a CPU
                vulnerability which may allow userspace to infer
                kernel data values previously stored in floating point
                registers, vector registers, or integer registers.
                RFDS only affects Intel Atom processors.
```

```
                on:     Turns ON the mitigation.
                off:    Turns OFF the mitigation.

                This parameter overrides the compile time default set
                by CONFIG_MITIGATION_RFDS. Mitigation cannot be
                disabled when other VERW based mitigations (like MDS)
                are enabled. In order to disable RFDS mitigation all
                VERW based mitigations need to be disabled.

                For details see:
                Documentation/admin-guide/hw-vuln/reg-file-data-sampling.rs

    dm_verity.keyring_unsealed=
                [KNL] When set to 1, leave the dm-verity keyring
                unsealed after initialization so userspace can
                provision keys. Once the keyring is restricted
                it becomes active and is searched during signature
                verification.

    driver_async_probe=  [KNL]
                List of driver names to be probed asynchronously. *
                matches with all driver names. If * is specified, the
                rest of the listed driver names are those that will NOT
                match the *.
                Format: <driver_name1>,<driver_name2>...

    drm.edid_firmware=[<connector>:]<file>[,[<connector>:]<file>]
                Broken monitors, graphic adapters, KVMs and EDIDless
                panels may send no or incorrect EDID data sets.
                This parameter allows to specify an EDID data sets
                in the /lib/firmware directory that are used instead.
                An EDID data set will only be used for a particular
                connector, if its name and a colon are prepended to
                the EDID name. Each connector may use a unique EDID
                data set by separating the files with a comma. An EDID
                data set with no connector name will be used for
                any connectors not explicitly specified.

    dscc4.setup=    [NET]

    dt_cpu_ftrs=    [PPC,EARLY]
                Format: {"off" | "known"}
                Control how the dt_cpu_ftrs device-tree binding is
                used for CPU feature discovery and setup (if it
                exists).
                off: Do not use it, fall back to legacy cpu table.
                known: Do not pass through unknown features to guests
                or userspace, only those that the kernel is aware of.

    dump_apple_properties   [X86]
                Dump name and content of EFI device properties on
                x86 Macs.  Useful for driver authors to determine
```

```
                what data is available or for reverse-engineering.

dyndbg[="val"]          [KNL,DYNAMIC_DEBUG]
<module>.dyndbg[="val"]
                Enable debug messages at boot time.  See
                Documentation/admin-guide/dynamic-debug-howto.rst
                for details.

early_ioremap_debug [KNL,EARLY]
                Enable debug messages in early_ioremap support. This
                is useful for tracking down temporary early mappings
                which are not unmapped.

earlycon=       [KNL,EARLY] Output early console device and options.

                When used with no options, the early console is
                determined by stdout-path property in device tree's
                chosen node or the ACPI SPCR table if supported by
                the platform.

        cdns,<addr>[,options]
                Start an early, polled-mode console on a Cadence
                (xuartps) serial port at the specified address. Only
                supported option is baud rate. If baud rate is not
                specified, the serial port must already be setup and
                configured.

        uart[8250],io,<addr>[,options[,uartclk]]
        uart[8250],mmio,<addr>[,options[,uartclk]]
        uart[8250],mmio32,<addr>[,options[,uartclk]]
        uart[8250],mmio32be,<addr>[,options[,uartclk]]
        uart[8250],0x<addr>[,options]
                Start an early, polled-mode console on the 8250/16550
                UART at the specified I/O port or MMIO address.
                MMIO inter-register address stride is either 8-bit
                (mmio) or 32-bit (mmio32 or mmio32be).
                If none of [io|mmio|mmio32|mmio32be], <addr> is assumed
                to be equivalent to 'mmio'. 'options' are specified
                in the same format described for "console=ttyS<n>"; if
                unspecified, the h/w is not initialized. 'uartclk' is
                the uart clock frequency; if unspecified, it is set
                to 'BASE_BAUD' * 16.

        pl011,<addr>
        pl011,mmio32,<addr>
                Start an early, polled-mode console on a pl011 serial
                port at the specified address. The pl011 serial port
                must already be setup and configured. Options are not
                yet supported.  If 'mmio32' is specified, then only
                the driver will use only 32-bit accessors to read/write
                the device registers.
```

```
            liteuart,<addr>
                    Start an early console on a litex serial port at the
                    specified address. The serial port must already be
                    setup and configured. Options are not yet supported.

            meson,<addr>
                    Start an early, polled-mode console on a meson serial
                    port at the specified address. The serial port must
                    already be setup and configured. Options are not yet
                    supported.

            msm_serial,<addr>
                    Start an early, polled-mode console on an msm serial
                    port at the specified address. The serial port
                    must already be setup and configured. Options are not
                    yet supported.

            msm_serial_dm,<addr>
                    Start an early, polled-mode console on an msm serial
                    dm port at the specified address. The serial port
                    must already be setup and configured. Options are not
                    yet supported.

            owl,<addr>
                    Start an early, polled-mode console on a serial port
                    of an Actions Semi SoC, such as S500 or S900, at the
                    specified address. The serial port must already be
                    setup and configured. Options are not yet supported.

            rda,<addr>
                    Start an early, polled-mode console on a serial port
                    of an RDA Micro SoC, such as RDA8810PL, at the
                    specified address. The serial port must already be
                    setup and configured. Options are not yet supported.

            sbi
                    Use RISC-V SBI (Supervisor Binary Interface) for early
                    console.

            smh     Use ARM semihosting calls for early console.

            s3c2410,<addr>
            s3c2412,<addr>
            s3c2440,<addr>
            s3c6400,<addr>
            s5pv210,<addr>
            exynos4210,<addr>
                    Use early console provided by serial driver available
                    on Samsung SoCs, requires selecting proper type and
                    a correct base address of the selected UART port. The
                    serial port must already be setup and configured.
                    Options are not yet supported.
```

```
            lantiq,<addr>
                    Start an early, polled-mode console on a lantiq serial
                    (lqasc) port at the specified address. The serial port
                    must already be setup and configured. Options are not
                    yet supported.

            lpuart,<addr>
            lpuart32,<addr>
                    Use early console provided by Freescale LP UART driver
                    found on Freescale Vybrid and QorIQ LS1021A processors.
                    A valid base address must be provided, and the serial
                    port must already be setup and configured.

            ec_imx21,<addr>
            ec_imx6q,<addr>
                    Start an early, polled-mode, output-only console on the
                    Freescale i.MX UART at the specified address. The UART
                    must already be setup and configured.

            ar3700_uart,<addr>
                    Start an early, polled-mode console on the
                    Armada 3700 serial port at the specified
                    address. The serial port must already be setup
                    and configured. Options are not yet supported.

            qcom_geni,<addr>
                    Start an early, polled-mode console on a Qualcomm
                    Generic Interface (GENI) based serial port at the
                    specified address. The serial port must already be
                    setup and configured. Options are not yet supported.

            efifb,[options]
                    Start an early, unaccelerated console on the EFI
                    memory mapped framebuffer (if available). On cache
                    coherent non-x86 systems that use system memory for
                    the framebuffer, pass the 'ram' option so that it is
                    mapped with the correct attributes.

            linflex,<addr>
                    Use early console provided by Freescale LINFlexD UART
                    serial driver for NXP S32V234 SoCs. A valid base
                    address must be provided, and the serial port must
                    already be setup and configured.

    earlyprintk=    [X86,SH,ARM,M68k,S390,UM,EARLY]
                    earlyprintk=vga
                    earlyprintk=sclp
                    earlyprintk=xen
                    earlyprintk=serial[,ttySn[,baudrate]]
                    earlyprintk=serial[,0x...[,baudrate]]
                    earlyprintk=ttySn[,baudrate]
```

```
                    earlyprintk=dbgp[debugController#]
                    earlyprintk=mmio32,membase[,{nocfg|baudrate}]
                    earlyprintk=pciserial[,force],bus:device.function[,{nocfg|b
                    earlyprintk=xdbc[xhciController#]
                    earlyprintk=bios

                    earlyprintk is useful when the kernel crashes before
                    the normal console is initialized. It is not enabled by
                    default because it has some cosmetic problems.

                    Use "nocfg" to skip UART configuration, assume
                    BIOS/firmware has configured UART correctly.

                    Append ",keep" to not disable it when the real console
                    takes over.

                    Only one of vga, serial, or usb debug port can
                    be used at a time.

                    Currently only ttyS0 and ttyS1 may be specified by
                    name.  Other I/O ports may be explicitly specified
                    on some architectures (x86 and arm at least) by
                    replacing ttySn with an I/O port address, like this:
                            earlyprintk=serial,0x1008,115200
                    You can find the port for a given device in
                    /proc/tty/driver/serial:
                            2: uart:ST16650V2 port:00001008 irq:18 ...

                    Interaction with the standard serial driver is not
                    very good.

                    The VGA output is eventually overwritten by
                    the real console.

                    The xen option can only be used in Xen domains.

                    The sclp output can only be used on s390.

                    The bios output can only be used on SuperH.

                    The optional "force" to "pciserial" enables use of a
                    PCI device even when its classcode is not of the
                    UART class.

        edac_report=    [HW,EDAC] Control how to report EDAC event
                    Format: {"on" | "off" | "force"}
                    on: enable EDAC to report H/W event. May be overridden
                    by other higher priority error reporting module.
                    off: disable H/W event reporting through EDAC.
                    force: enforce the use of EDAC to report H/W event.
                    default: on.
```

```
        edd=            [EDD]
                        Format: {"off" | "on" | "skip[mbr]"}

        efi=            [EFI,EARLY]
                        Format: { "debug", "disable_early_pci_dma",
                                  "nochunk", "noruntime", "nosoftreserve",
                                  "novamap", "no_disable_early_pci_dma" }
                        debug: enable misc debug output.
                        disable_early_pci_dma: disable the busmaster bit on all
                        PCI bridges while in the EFI boot stub.
                        nochunk: disable reading files in "chunks" in the EFI
                        boot stub, as chunking can cause problems with some
                        firmware implementations.
                        noruntime : disable EFI runtime services support
                        nosoftreserve: The EFI_MEMORY_SP (Specific Purpose)
                        attribute may cause the kernel to reserve the
                        memory range for a memory mapping driver to
                        claim. Specify efi=nosoftreserve to disable this
                        reservation and treat the memory by its base type
                        (i.e. EFI_CONVENTIONAL_MEMORY / "System RAM").
                        novamap: do not call SetVirtualAddressMap().
                        no_disable_early_pci_dma: Leave the busmaster bit set
                        on all PCI bridges while in the EFI boot stub

        efi_no_storage_paranoia [EFI,X86,EARLY]
                        Using this parameter you can use more than 50% of
                        your efi variable storage. Use this parameter only if
                        you are really sure that your UEFI does sane gc and
                        fulfills the spec otherwise your board may brick.

        efivar_ssdt=    [EFI; X86] Name of an EFI variable that contains an SSDT
                        that is to be dynamically loaded by Linux. If there are
                        multiple variables with the same name but with different
                        vendor GUIDs, all of them will be loaded. See
                        Documentation/admin-guide/acpi/ssdt-overlays.rst for detail


        eisa_irq_edge=  [PARISC,HW]
                        See header of drivers/parisc/eisa.c.

        ekgdboc=        [X86,KGDB,EARLY] Allow early kernel console debugging
                        Format: ekgdboc=kbd

                        This is designed to be used in conjunction with
                        the boot argument: earlyprintk=vga

                        This parameter works in place of the kgdboc parameter
                        but can only be used if the backing tty is available
                        very early in the boot process. For early debugging
                        via a serial port see kgdboc_earlycon instead.

        elfcorehdr=[size[KMG]@]offset[KMG] [PPC,SH,X86,S390,EARLY]
```

```
                Specifies physical address of start of kernel core
                image elf header and optionally the size. Generally
                kexec loader will pass this option to capture kernel.
                See Documentation/admin-guide/kdump/kdump.rst for details.

    enable_mtrr_cleanup [X86,EARLY]
                The kernel tries to adjust MTRR layout from continuous
                to discrete, to make X server driver able to add WB
                entry later. This parameter enables that.

    enable_timer_pin_1 [X86]
                Enable PIN 1 of APIC timer
                Can be useful to work around chipset bugs
                (in particular on some ATI chipsets).
                The kernel tries to set a reasonable default.

    enforcing=      [SELINUX] Set initial enforcing status.
                Format: {"0" | "1"}
                See security/selinux/Kconfig help text.
                0 -- permissive (log only, no denials).
                1 -- enforcing (deny and log).
                Default value is 0.
                Value can be changed at runtime via
                /sys/fs/selinux/enforce.

    erst_disable    [ACPI]
                Disable Error Record Serialization Table (ERST)
                support.

    ether=          [HW,NET] Ethernet cards parameters
                This option is obsoleted by the "netdev=" option, which
                has equivalent usage. See its documentation for details.

    evm=            [EVM]
                Format: { "fix" }
                Permit 'security.evm' to be updated regardless of
                current integrity status.

    early_page_ext [KNL,EARLY] Enforces page_ext initialization to earlier
                stages so cover more early boot allocations.
                Please note that as side effect some optimizations
                might be disabled to achieve that (e.g. parallelized
                memory initialization is disabled) so the boot process
                might take longer, especially on systems with a lot of
                memory. Available with CONFIG_PAGE_EXTENSION=y.

    failslab=
    fail_usercopy=
    fail_page_alloc=
    fail_skb_realloc=
    fail_make_request=[KNL]
                General fault injection mechanism.
```

```
                Format: <interval>,<probability>,<space>,<times>
                See also Documentation/fault-injection/.

        fb_tunnels=     [NET]
                Format: { initns | none }
                See Documentation/admin-guide/sysctl/net.rst for
                fb_tunnels_only_for_init_ns

        floppy=         [HW]
                See Documentation/admin-guide/blockdev/floppy.rst.

        forcepae        [X86-32]
                Forcefully enable Physical Address Extension (PAE).
                Many Pentium M systems disable PAE but may have a
                functionally usable PAE implementation.
                Warning: use of this parameter will taint the kernel
                and may cause unknown problems.

        fred=           [X86-64]
                Enable/disable Flexible Return and Event Delivery.
                Format: { on | off }
                on: enable FRED when it's present, the default setting.
                off: disable FRED.

        ftrace=[tracer]
                [FTRACE] will set and start the specified tracer
                as early as possible in order to facilitate early
                boot debugging.

        ftrace_boot_snapshot
                [FTRACE] On boot up, a snapshot will be taken of the
                ftrace ring buffer that can be read at:
                /sys/kernel/tracing/snapshot.
                This is useful if you need tracing information from kernel
                boot up that is likely to be overridden by user space
                start up functionality.

                Optionally, the snapshot can also be defined for a tracing
                instance that was created by the trace_instance= command
                line parameter.

                trace_instance=foo,sched_switch ftrace_boot_snapshot=foo

                The above will cause the "foo" tracing instance to trigger
                a snapshot at the end of boot up.

        ftrace_dump_on_oops[=2(orig_cpu) | =<instance>][,<instance> |
                          ,<instance>=2(orig_cpu)]
                [FTRACE] will dump the trace buffers on oops.
                If no parameter is passed, ftrace will dump global
                buffers of all CPUs, if you pass 2 or orig_cpu, it
                will dump only the buffer of the CPU that triggered
```

```
                the oops, or the specific instance will be dumped if
                its name is passed. Multiple instance dump is also
                supported, and instances are separated by commas. Each
                instance supports only dump on CPU that triggered the
                oops by passing 2 or orig_cpu to it.

                ftrace_dump_on_oops=foo=orig_cpu

                The above will dump only the buffer of "foo" instance
                on CPU that triggered the oops.

                ftrace_dump_on_oops,foo,bar=orig_cpu

                The above will dump global buffer on all CPUs, the
                buffer of "foo" instance on all CPUs and the buffer
                of "bar" instance on CPU that triggered the oops.

    ftrace_filter=[function-list]
                [FTRACE] Limit the functions traced by the function
                tracer at boot up. function-list is a comma-separated
                list of functions. This list can be changed at run
                time by the set_ftrace_filter file in the debugfs
                tracing directory.

    ftrace_notrace=[function-list]
                [FTRACE] Do not trace the functions specified in
                function-list. This list can be changed at run time
                by the set_ftrace_notrace file in the debugfs
                tracing directory.

    ftrace_graph_filter=[function-list]
                [FTRACE] Limit the top level callers functions traced
                by the function graph tracer at boot up.
                function-list is a comma-separated list of functions
                that can be changed at run time by the
                set_graph_function file in the debugfs tracing directory.

    ftrace_graph_notrace=[function-list]
                [FTRACE] Do not trace from the functions specified in
                function-list.  This list is a comma-separated list of
                functions that can be changed at run time by the
                set_graph_notrace file in the debugfs tracing directory.

    ftrace_graph_max_depth=<uint>
                [FTRACE] Used with the function graph tracer. This is
                the max depth it will trace into a function. This value
                can be changed at run time by the max_graph_depth file
                in the tracefs tracing directory. default: 0 (no limit)

    fw_devlink=     [KNL,EARLY] Create device links between consumer and suppli
                devices by scanning the firmware to infer the
                consumer/supplier relationships. This feature is
```

```
                especially useful when drivers are loaded as modules as
                it ensures proper ordering of tasks like device probing
                (suppliers first, then consumers), supplier boot state
                clean up (only after all consumers have probed),
                suspend/resume & runtime PM (consumers first, then
                suppliers).
                Format: { off | permissive | on | rpm }
                off --  Don't create device links from firmware info.
                permissive -- Create device links from firmware info
                        but use it only for ordering boot state clean
                        up (sync_state() calls).
                on --   Create device links from firmware info and use it
                        to enforce probe and suspend/resume ordering.
                rpm --  Like "on", but also use to order runtime PM.

    fw_devlink.strict=<bool>
                [KNL,EARLY] Treat all inferred dependencies as mandatory
                dependencies. This only applies for fw_devlink=on|rpm.
                Format: <bool>

    fw_devlink.sync_state =
                [KNL,EARLY] When all devices that could probe have finished
                probing, this parameter controls what to do with
                devices that haven't yet received their sync_state()
                calls.
                Format: { strict | timeout }
                strict -- Default. Continue waiting on consumers to
                        probe successfully.
                timeout -- Give up waiting on consumers and call
                        sync_state() on any devices that haven't yet
                        received their sync_state() calls after
                        deferred_probe_timeout has expired or by
                        late_initcall() if !CONFIG_MODULES.

    gamecon.map[2|3]=
                [HW,JOY] Multisystem joystick and NES/SNES/PSX pad
                support via parallel port (up to 5 devices per port)
                Format: <port#>,<pad1>,<pad2>,<pad3>,<pad4>,<pad5>
                See also Documentation/input/devices/joystick-parport.rst

    gamma=          [HW,DRM]

    gart_fix_e820=  [X86-64,EARLY] disable the fix e820 for K8 GART
                Format: off | on
                default: on

    gather_data_sampling=
                [X86,INTEL,EARLY] Control the Gather Data Sampling (GDS)
                mitigation.

                Gather Data Sampling is a hardware vulnerability which
                allows unprivileged speculative access to data which was
```

```
                        previously stored in vector registers.

                        This issue is mitigated by default in updated microcode.
                        The mitigation may have a performance impact but can be
                        disabled. On systems without the microcode mitigation
                        disabling AVX serves as a mitigation.

                        force:  Disable AVX to mitigate systems without
                                microcode mitigation. No effect if the microcode
                                mitigation is present. Known to cause crashes in
                                userspace with buggy AVX enumeration.

                        off:    Disable GDS mitigation.

        gbpages         [X86] Use GB pages for kernel direct mappings.

        gcov_persist=   [GCOV] When non-zero (default), profiling data for
                        kernel modules is saved and remains accessible via
                        debugfs, even when the module is unloaded/reloaded.
                        When zero, profiling data is discarded and associated
                        debugfs files are removed at module unload time.

        goldfish        [X86] Enable the goldfish android emulator platform.
                        Don't use this when you are not running on the
                        android emulator

        gpio-mockup.gpio_mockup_ranges
                        [HW] Sets the ranges of gpiochip of for this device.
                        Format: <start1>,<end1>,<start2>,<end2>...
        gpio-mockup.gpio_mockup_named_lines
                        [HW] Let the driver know GPIO lines should be named.

        gpt             [EFI] Forces disk with valid GPT signature but
                        invalid Protective MBR to be treated as GPT. If the
                        primary GPT is corrupted, it enables the backup/alternate
                        GPT to be used instead.

        grcan.enable0=  [HW] Configuration of physical interface 0. Determines
                        the "Enable 0" bit of the configuration register.
                        Format: 0 | 1
                        Default: 0
        grcan.enable1=  [HW] Configuration of physical interface 1. Determines
                        the "Enable 0" bit of the configuration register.
                        Format: 0 | 1
                        Default: 0
        grcan.select=   [HW] Select which physical interface to use.
                        Format: 0 | 1
                        Default: 0
        grcan.txsize=   [HW] Sets the size of the tx buffer.
                        Format: <unsigned int> such that (txsize & ~0x1fffc0) == 0.
                        Default: 1024
        grcan.rxsize=   [HW] Sets the size of the rx buffer.
```

```
                Format: <unsigned int> such that (rxsize & ~0x1fffc0) == 0.
                Default: 1024

    hardened_usercopy=
                [KNL] Under CONFIG_HARDENED_USERCOPY, whether
                hardening is enabled for this boot. Hardened
                usercopy checking is used to protect the kernel
                from reading or writing beyond known memory
                allocation boundaries as a proactive defense
                against bounds-checking flaws in the kernel's
                copy_to_user()/copy_from_user() interface.
                The default is determined by
                CONFIG_HARDENED_USERCOPY_DEFAULT_ON.
            on      Perform hardened usercopy checks.
            off     Disable hardened usercopy checks.

    hardlockup_all_cpu_backtrace=
                [KNL] Should the hard-lockup detector generate
                backtraces on all cpus.
                Format: 0 | 1

    hash_pointers=
                [KNL,EARLY]
                By default, when pointers are printed to the console
                or buffers via the %p format string, that pointer is
                "hashed", i.e. obscured by hashing the pointer value.
                This is a security feature that hides actual kernel
                addresses from unprivileged users, but it also makes
                debugging the kernel more difficult since unequal
                pointers can no longer be compared. The choices are:
                Format: { auto | always | never }
                Default: auto

                auto   - Hash pointers unless slab_debug is enabled.
                always - Always hash pointers (even if slab_debug is
                         enabled).
                never  - Never hash pointers. This option should only
                         be specified when debugging the kernel. Do
                         not use on production kernels. The boot
                         param "no_hash_pointers" is an alias for
                         this mode.

                For controlling hashing dynamically at runtime,
                use the "kernel.kptr_restrict" sysctl instead.

    hashdist=       [KNL,NUMA] Large hashes allocated during boot
                are distributed across NUMA nodes.  Defaults on
                for 64-bit NUMA, off otherwise.
                Format: 0 | 1 (for off | on)

    hd=             [EIDE] (E)IDE hard drive subsystem geometry
                Format: <cyl>,<head>,<sect>
```

```
hest_disable    [ACPI]
                Disable Hardware Error Source Table (HEST) support;
                corresponding firmware-first mode error processing
                logic will be disabled.

hibernate=      [HIBERNATION]
        noresume        Don't check if there's a hibernation image
                        present during boot.
        nocompress      Don't compress/decompress hibernation images.
        no              Disable hibernation and resume.
        protect_image   Turn on image protection during restoration
                        (that will set all pages holding image data
                        during restoration read-only).

hibernate.compressor=   [HIBERNATION] Compression algorithm to be
                        used with hibernation.
                        Format: { lzo | lz4 }
                        Default: lzo

                        lzo: Select LZO compression algorithm to
                        compress/decompress hibernation image.

                        lz4: Select LZ4 compression algorithm to
                        compress/decompress hibernation image.

hibernate.pm_test_delay=
                [HIBERNATION]
                Sets the number of seconds to remain in a hibernation test
                mode before resuming the system (see
                /sys/power/pm_test). Only available when CONFIG_PM_DEBUG
                is set. Default value is 5.

hibernate_compression_threads=
                [HIBERNATION]
                Set the number of threads used for compressing or decompres
                hibernation images.

                Format: <integer>
                Default: 3
                Minimum: 1
                Example: hibernate_compression_threads=4

highmem=nn[KMG] [KNL,BOOT,EARLY] forces the highmem zone to have an exact
                size of <nn>. This works even on boxes that have no
                highmem otherwise. This also works to reduce highmem
                size on bigger boxes.

highres=        [KNL] Enable/disable high resolution timer mode.
                Valid parameters: "on", "off"
                Default: "on"
```

```
        hlt             [BUGS=ARM,SH]

        hostname=       [KNL,EARLY] Set the hostname (aka UTS nodename).
                        Format: <string>
                        This allows setting the system's hostname during early
                        startup. This sets the name returned by gethostname.
                        Using this parameter to set the hostname makes it
                        possible to ensure the hostname is correctly set before
                        any userspace processes run, avoiding the possibility
                        that a process may call gethostname before the hostname
                        has been explicitly set, resulting in the calling
                        process getting an incorrect result. The string must
                        not exceed the maximum allowed hostname length (usually
                        64 characters) and will be truncated otherwise.

        hpet=           [X86-32,HPET] option to control HPET usage
                        Format: { enable (default) | disable | force |
                                verbose }
                        disable: disable HPET and use PIT instead
                        force: allow force enabled of undocumented chips (ICH4,
                                VIA, nVidia)
                        verbose: show contents of HPET registers during setup

        hpet_mmap=      [X86, HPET_MMAP] Allow userspace to mmap HPET
                        registers.  Default set by CONFIG_HPET_MMAP_DEFAULT.

        hugepages=      [HW,EARLY] Number of HugeTLB pages to allocate at boot.
                        If this follows hugepagesz (below), it specifies
                        the number of pages of hugepagesz to be allocated.
                        If this is the first HugeTLB parameter on the command
                        line, it specifies the number of pages to allocate for
                        the default huge page size. If using node format, the
                        number of pages to allocate per-node can be specified.
                        See also Documentation/admin-guide/mm/hugetlbpage.rst.
                        Format: <integer> or (node format)
                                <node>:<integer>[,<node>:<integer>]

        hugepagesz=
                        [HW,EARLY] The size of the HugeTLB pages.  This is
                        used in conjunction with hugepages (above) to
                        allocate huge pages of a specific size at boot. The
                        pair hugepagesz=X hugepages=Y can be specified once
                        for each supported huge page size. Huge page sizes
                        are architecture dependent. See also
                        Documentation/admin-guide/mm/hugetlbpage.rst.
                        Format: size[KMG]

        hugepage_alloc_threads=
                        [HW] The number of threads that should be used to
                        allocate hugepages during boot. This option can be
                        used to improve system bootup time when allocating
                        a large amount of huge pages.
```

```
                The default value is 25% of the available hardware threads.

                Note that this parameter only applies to non-gigantic huge

    hugetlb_cma=    [HW,CMA,EARLY] The size of a CMA area used for allocation
                of gigantic hugepages. Or using node format, the size
                of a CMA area per node can be specified.
                Format: nn[KMGTPE] or (node format)
                        <node>:nn[KMGTPE][,<node>:nn[KMGTPE]]

                The size must be a multiple of the gigantic page size.
                When using node format, this applies to each per-node size.
                Missaligned values are dropped with a warning.

                Reserve a CMA area of given size and allocate gigantic
                hugepages using the CMA allocator. If enabled, the
                boot-time allocation of gigantic hugepages is skipped.

    hugetlb_cma_only=
                [HW,CMA,EARLY] When allocating new HugeTLB pages, only
                try to allocate from the CMA areas.

                This option does nothing if hugetlb_cma= is not also
                specified.

    hugetlb_free_vmemmap=
                [KNL] Requires CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP
                enabled.
                Control if HugeTLB Vmemmap Optimization (HVO) is enabled.
                Allows heavy hugetlb users to free up some more
                memory (7 * PAGE_SIZE for each 2MB hugetlb page).
                Format: { on | off (default) }

                on: enable HVO
                off: disable HVO

                Built with CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP_DEFAULT_ON=
                the default is on.

                Note that the vmemmap pages may be allocated from the added
                memory block itself when memory_hotplug.memmap_on_memory is
                enabled, those vmemmap pages cannot be optimized even if th
                feature is enabled.  Other vmemmap pages not allocated from
                the added memory block itself do not be affected.

    hung_task_panic=
                [KNL] Number of hung tasks to trigger kernel panic.
                Format: <int>

                When set to a non-zero value, a kernel panic will be trigge
                the number of detected hung tasks reaches this value.
```

```
                    0: don't panic
                    1: panic immediately on first hung task
                    N: panic after N hung tasks are detected in a single scan

                    The default value is controlled by the
                    CONFIG_BOOTPARAM_HUNG_TASK_PANIC build-time option. The val
                    selected by this boot parameter can be changed later by the
                    kernel.hung_task_panic sysctl.

        hvc_iucv=       [S390]  Number of z/VM IUCV hypervisor console (HVC)
                                terminal devices. Valid values: 0..8
        hvc_iucv_allow= [S390]  Comma-separated list of z/VM user IDs.
                                If specified, z/VM IUCV HVC accepts connections
                                from listed z/VM user IDs only.

        hv_nopvspin     [X86,HYPER_V,EARLY]
                        Disables the paravirt spinlock optimizations
                        which allow the hypervisor to 'idle' the guest
                        on lock contention.

        hw_protection=  [HW]
                        Format: reboot | shutdown

                        Hardware protection action taken on critical events like
                        overtemperature or imminent voltage loss.

        i2c_bus=        [HW]    Override the default board specific I2C bus speed
                                or register an additional I2C bus that is not
                                registered from board initialization code.
                                Format:
                                <bus_id>,<clkrate>

        i2c_touchscreen_props= [HW,ACPI,X86]
                        Set device-properties for ACPI-enumerated I2C-attached
                        touchscreen, to e.g. fix coordinates of upside-down
                        mounted touchscreens. If you need this option please
                        submit a drivers/platform/x86/touchscreen_dmi.c patch
                        adding a DMI quirk for this.

                        Format:
                        <ACPI_HW_ID>:<prop_name>=<val>[:prop_name=val][:...]
                        Where <val> is one of:
                        Omit "=<val>" entirely  Set a boolean device-property
                        Unsigned number         Set a u32 device-property
                        Anything else           Set a string device-property

                        Examples (split over multiple lines):
                        i2c_touchscreen_props=GDIX1001:touchscreen-inverted-x:
                        touchscreen-inverted-y

                        i2c_touchscreen_props=MSSL1680:touchscreen-size-x=1920:
                        touchscreen-size-y=1080:touchscreen-inverted-y:
```

```
                firmware-name=gsl1680-vendor-model.fw:silead,home-button

    i8042.debug     [HW] Toggle i8042 debug mode
    i8042.unmask_kbd_data
                    [HW] Enable printing of interrupt data from the KBD port
                         (disabled by default, and as a pre-condition
                         requires that i8042.debug=1 be enabled)
    i8042.direct    [HW] Put keyboard port into non-translated mode
    i8042.dumbkbd   [HW] Pretend that controller can only read data from
                         keyboard and cannot control its state
                         (Don't attempt to blink the leds)
    i8042.noaux     [HW] Don't check for auxiliary (== mouse) port
    i8042.nokbd     [HW] Don't check/create keyboard port
    i8042.noloop    [HW] Disable the AUX Loopback command while probing
                         for the AUX port
    i8042.nomux     [HW] Don't check presence of an active multiplexing
                         controller
    i8042.nopnp     [HW] Don't use ACPIPnP / PnPBIOS to discover KBD/AUX
                         controllers
    i8042.notimeout [HW] Ignore timeout condition signalled by controller
    i8042.reset     [HW] Reset the controller during init, cleanup and
                         suspend-to-ram transitions, only during s2r
                         transitions, or never reset
                    Format: { 1 | Y | y | 0 | N | n }
                    1, Y, y: always reset controller
                    0, N, n: don't ever reset controller
                    Default: only on s2r transitions on x86; most other
                    architectures force reset to be always executed
    i8042.unlock    [HW] Unlock (ignore) the keylock
    i8042.kbdreset  [HW] Reset device connected to KBD port
    i8042.probe_defer
                    [HW] Allow deferred probing upon i8042 probe errors

    i810=           [HW,DRM]

    i915.invert_brightness=
                    [DRM] Invert the sense of the variable that is used to
                    set the brightness of the panel backlight. Normally a
                    brightness value of 0 indicates backlight switched off,
                    and the maximum of the brightness value sets the backlight
                    to maximum brightness. If this parameter is set to 0
                    (default) and the machine requires it, or this parameter
                    is set to 1, a brightness value of 0 sets the backlight
                    to maximum brightness, and the maximum of the brightness
                    value switches the backlight off.
                    -1 -- never invert brightness
                     0 -- machine default
                     1 -- force brightness inversion

    ia32_emulation= [X86-64]
                    Format: <bool>
                    When true, allows loading 32-bit programs and executing 32-
```

```
                  syscalls, essentially overriding IA32_EMULATION_DEFAULT_DIS
                  boot time. When false, unconditionally disables IA32 emulat


      idle=       [X86,EARLY]
                  Format: idle=poll, idle=halt, idle=nomwait

                  idle=poll:  Don't do power saving in the idle loop
                  using HLT, but poll for rescheduling event. This will
                  make the CPUs eat a lot more power, but may be useful
                  to get slightly better performance in multiprocessor
                  benchmarks. It also makes some profiling using
                  performance counters more accurate.  Please note that
                  on systems with MONITOR/MWAIT support (like Intel
                  EM64T CPUs) this option has no performance advantage
                  over the normal idle loop.  It may also interact badly
                  with hyperthreading.

                  idle=halt: Halt is forced to be used for CPU idle.
                  In such case C2/C3 won't be used again.

                  idle=nomwait: Disable mwait for CPU C-states

      idxd.sva=   [HW]
                  Format: <bool>
                  Allow force disabling of Shared Virtual Memory (SVA)
                  support for the idxd driver. By default it is set to
                  true (1).

      idxd.tc_override= [HW]
                  Format: <bool>
                  Allow override of default traffic class configuration
                  for the device. By default it is set to false (0).

      ieee754=    [MIPS] Select IEEE Std 754 conformance mode
                  Format: { strict | legacy | 2008 | relaxed | emulated }
                  Default: strict

                  Choose which programs will be accepted for execution
                  based on the IEEE 754 NaN encoding(s) supported by
                  the FPU and the NaN encoding requested with the value
                  of an ELF file header flag individually set by each
                  binary.  Hardware implementations are permitted to
                  support either or both of the legacy and the 2008 NaN
                  encoding mode.

                  Available settings are as follows:
                  strict  accept binaries that request a NaN encoding
                          supported by the FPU
                  legacy  only accept legacy-NaN binaries, if supported
                          by the FPU
                  2008    only accept 2008-NaN binaries, if supported
```

```
                        by the FPU
                relaxed accept any binaries regardless of whether
                        supported by the FPU
                emulated accept any binaries but enable FPU emulator
                        if binary mode is unsupported by the FPU.

                The FPU emulator is always able to support both NaN
                encodings, so if no FPU hardware is present or it has
                been disabled with 'nofpu', then the settings of
                'legacy' and '2008' strap the emulator accordingly,
                'relaxed' straps the emulator for both legacy-NaN and
                2008-NaN, whereas 'strict' enables legacy-NaN only on
                legacy processors and both NaN encodings on MIPS32 or
                MIPS64 CPUs.

                The setting for ABS.fmt/NEG.fmt instruction execution
                mode generally follows that for the NaN encoding,
                except where unsupported by hardware.

    ignore_loglevel [KNL,EARLY]
                Ignore loglevel setting - this will print /all/
                kernel messages to the console. Useful for debugging.
                We also add it as printk module parameter, so users
                could change it dynamically, usually by
                /sys/module/printk/parameters/ignore_loglevel.

    ignore_rlimit_data
                Ignore RLIMIT_DATA setting for data mappings,
                print warning at first misuse.  Can be changed via
                /sys/module/kernel/parameters/ignore_rlimit_data.

    ihash_entries=  [KNL]
                Set number of hash buckets for inode cache.

    ima_appraise=   [IMA] appraise integrity measurements
                Format: { "off" | "enforce" | "fix" | "log" }
                default: "enforce"

    ima_appraise_tcb [IMA] Deprecated.  Use ima_policy= instead.
                The builtin appraise policy appraises all files
                owned by uid=0.

    ima_canonical_fmt [IMA]
                Use the canonical format for the binary runtime
                measurements, instead of host native format.

    ima_flush_htable  [IMA]
                Flush the IMA hash table when deleting all the
                staged measurement records, to achieve maximum
                memory saving at the cost of having duplicate
                records across the staged measurement lists.
```

```
        ima_hash=       [IMA]
                        Format: { md5 | sha1 | rmd160 | sha256 | sha384
                                  | sha512 | ... }
                        default: "sha1"

                        The list of supported hash algorithms is defined
                        in crypto/hash_info.h.

        ima_policy=     [IMA]
                        The builtin policies to load during IMA setup.
                        Format: "tcb | appraise_tcb | secure_boot |
                                 fail_securely | critical_data"

                        The "tcb" policy measures all programs exec'd, files
                        mmap'd for exec, and all files opened with the read
                        mode bit set by either the effective uid (euid=0) or
                        uid=0.

                        The "appraise_tcb" policy appraises the integrity of
                        all files owned by root.

                        The "secure_boot" policy appraises the integrity
                        of files (eg. kexec kernel image, kernel modules,
                        firmware, policy, etc) based on file signatures.

                        The "fail_securely" policy forces file signature
                        verification failure also on privileged mounted
                        filesystems with the SB_I_UNVERIFIABLE_SIGNATURE
                        flag.

                        The "critical_data" policy measures kernel integrity
                        critical data.

        ima_tcb         [IMA] Deprecated.  Use ima_policy= instead.
                        Load a policy which meets the needs of the Trusted
                        Computing Base.  This means IMA will measure all
                        programs exec'd, files mmap'd for exec, and all files
                        opened for read by uid=0.

        ima_template=   [IMA]
                        Select one of defined IMA measurements template formats.
                        Formats: { "ima" | "ima-ng" | "ima-ngv2" | "ima-sig" |
                                   "ima-sigv2" }
                        Default: "ima-ng"

        ima_template_fmt=
                        [IMA] Define a custom template format.
                        Format: { "field1|...|fieldN" }

        ima=            [IMA] Enable or disable IMA
                        Format: { "off" | "on" }
                        Default: "on"
```

```
                Note that disabling IMA is limited to kdump kernel.

indirect_target_selection= [X86,Intel] Mitigation control for Indirect
                Target Selection(ITS) bug in Intel CPUs. Updated
                microcode is also required for a fix in IBPB.

                on:     Enable mitigation (default).
                off:    Disable mitigation.
                force:  Force the ITS bug and deploy default
                        mitigation.
                vmexit: Only deploy mitigation if CPU is affected by
                        guest/host isolation part of ITS.
                stuff:  Deploy RSB-fill mitigation when retpoline is
                        also deployed. Otherwise, deploy the default
                        mitigation.

                For details see:
                Documentation/admin-guide/hw-vuln/indirect-target-selection

init=           [KNL]
                Format: <full_path>
                Run specified binary instead of /sbin/init as init
                process.

initcall_debug  [KNL] Trace initcalls as they are executed.  Useful
                for working out where the kernel is dying during
                startup.

initcall_blacklist=  [KNL] Do not execute a comma-separated list of
                initcall functions.  Useful for debugging built-in
                modules and initcalls.

initramfs_async= [KNL]
                Format: <bool>
                Default: 1
                This parameter controls whether the initramfs
                image is unpacked asynchronously, concurrently
                with devices being probed and
                initialized. This should normally just work,
                but as a debugging aid, one can get the
                historical behaviour of the initramfs
                unpacking being completed before device_ and
                late_ initcalls.

initrd=         [BOOT,EARLY] Specify the location of the initial ramdisk

initrdmem=      [KNL,EARLY] Specify a physical address and size from which
                load the initrd. If an initrd is compiled in or
                specified in the bootparams, it takes priority over this
                setting.
                Format: ss[KMG],nn[KMG]
                Default is 0, 0
```

```
	init_on_alloc=  [MM,EARLY] Fill newly allocated pages and heap objects with
	                zeroes.
	                Format: 0 | 1
	                Default set by CONFIG_INIT_ON_ALLOC_DEFAULT_ON.

	init_on_free=   [MM,EARLY] Fill freed pages and heap objects with zeroes.
	                Format: 0 | 1
	                Default set by CONFIG_INIT_ON_FREE_DEFAULT_ON.

	init_pkru=      [X86] Specify the default memory protection keys rights
	                register contents for all processes.  0x55555554 by
	                default (disallow access to all but pkey 0).  Can
	                override in debugfs after boot.

	inport.irq=     [HW] Inport (ATI XL and Microsoft) busmouse driver
	                Format: <irq>

	int_pln_enable  [X86] Enable power limit notification interrupt

	integrity_audit=[IMA]
	                Format: { "0" | "1" }
	                0 -- basic integrity auditing messages. (Default)
	                1 -- additional integrity auditing messages.

	intel_iommu=    [DMAR] Intel IOMMU driver (DMAR) option
	        on
	                Enable intel iommu driver.
	        off
	                Disable intel iommu driver.
	        igfx_off [Default Off]
	                By default, gfx is mapped as normal device. If a gfx
	                device has a dedicated DMAR unit, the DMAR unit is
	                bypassed by not enabling DMAR with this option. In
	                this case, gfx device will use physical address for
	                DMA.
	        strict [Default Off]
	                Deprecated, equivalent to iommu.strict=1.
	        sp_off [Default Off]
	                By default, super page will be supported if Intel IOMMU
	                has the capability. With this option, super page will
	                not be supported.
	        sm_on
	                Enable the Intel IOMMU scalable mode if the hardware
	                advertises that it has support for the scalable mode
	                translation.
	        sm_off
	                Disallow use of the Intel IOMMU scalable mode.
	        tboot_noforce [Default Off]
	                Do not force the Intel IOMMU enabled under tboot.
	                By default, tboot will force Intel IOMMU on, which
	                could harm performance of some high-throughput
```

```
                devices like 40GBit network cards, even if identity
                mapping is enabled.
                Note that using this option lowers the security
                provided by tboot because it makes the system
                vulnerable to DMA attacks.

    intel_idle.max_cstate=  [KNL,HW,ACPI,X86]
                0       disables intel_idle and fall back on acpi_idle.
                1 to 9  specify maximum depth of C-state.

    intel_pstate=   [X86,EARLY]
                disable
                  Do not enable intel_pstate as the default
                  scaling driver for the supported processors
                active
                  Use intel_pstate driver to bypass the scaling
                  governors layer of cpufreq and provides it own
                  algorithms for p-state selection. There are two
                  P-state selection algorithms provided by
                  intel_pstate in the active mode: powersave and
                  performance.  The way they both operate depends
                  on whether or not the hardware managed P-states
                  (HWP) feature has been enabled in the processor
                  and possibly on the processor model.
                passive
                  Use intel_pstate as a scaling driver, but configure it
                  to work with generic cpufreq governors (instead of
                  enabling its internal governor).  This mode cannot be
                  used along with the hardware-managed P-states (HWP)
                  feature.
                force
                  Enable intel_pstate on systems that prohibit it by defaul
                  in favor of acpi-cpufreq. Forcing the intel_pstate driver
                  instead of acpi-cpufreq may disable platform features, su
                  as thermal controls and power capping, that rely on ACPI
                  P-States information being indicated to OSPM and therefor
                  should be used with caution. This option does not work wi
                  processors that aren't supported by the intel_pstate driv
                  or on platforms that use pcc-cpufreq instead of acpi-cpuf
                no_hwp
                  Do not enable hardware P state control (HWP)
                  if available.
                hwp_only
                  Only load intel_pstate on systems which support
                  hardware P state control (HWP) if available.
                support_acpi_ppc
                  Enforce ACPI _PPC performance limits. If the Fixed ACPI
                  Description Table, specifies preferred power management
                  profile as "Enterprise Server" or "Performance Server",
                  then this feature is turned on by default.
                per_cpu_perf_limits
                  Allow per-logical-CPU P-State performance control limits
```

```
                    cpufreq sysfs interface
                  no_cas
                    Do not enable capacity-aware scheduling (CAS) on
                    hybrid systems

        intremap=   [X86-64,Intel-IOMMU,EARLY]
                  on      enable Interrupt Remapping (default)
                  off     disable Interrupt Remapping
                  nosid   disable Source ID checking
                  no_x2apic_optout
                          BIOS x2APIC opt-out request will be ignored
                  nopost  disable Interrupt Posting
                  posted_msi
                          enable MSIs delivered as posted interrupts

        iomem=    Disable strict checking of access to MMIO memory
                strict  regions from userspace.
                relaxed

        iommu=    [X86,EARLY]

                off
                        Don't initialize and use any kind of IOMMU.

                force
                        Force the use of the hardware IOMMU even when
                        it is not actually needed (e.g. because < 3 GB
                        memory).

                noforce
                        Don't force hardware IOMMU usage when it is not
                        needed. (default).

                merge
                        Do scatter-gather (SG) merging. Implies "force"
                        (experimental).

                nomerge
                        Don't do scatter-gather (SG) merging.

                biomerge
                        Do scatter-gather (SG) merging. Implies "force"
                        (experimental). [same as "merge"]

                panic
                        Always panic when IOMMU overflows.

                nopanic
                        Don't panic on IOMMU overflows.

                pt
                        Use passthrough mode by default
```

```
                (Equivalent to iommu.passthrough=1)

        nopt
                Use translated mode for DMA by default
                (Equivalent to iommu.passthrough=0)

        soft
                Use software bounce buffering (SWIOTLB) (default for
                Intel machines). This can be used to prevent the usage
                of an available hardware IOMMU.

        usedac
                Use the DAC on VIA PCI bridge
                (default: disable the VIA PCI bridge DAC)

    AMD Gart HW IOMMU-specific options: (CONFIG_GART_IOMMU)

        <size>
                Set the size of the remapping area in bytes.

        allowed
                Overwrite iommu off workarounds for specific chipsets

        force
                Overwrite iommu off workarounds for specific chipsets

        fullflush
                Flush IOMMU on each allocation (default).

        nofullflush
                Don't use IOMMU fullflush.

        memaper[=<order>]
                Allocate an own aperture over RAM with size
                32MB<<order.  (default: order=1, i.e. 64MB)

        noaperture
                Ask the IOMMU not to touch the aperture for AGP.

        noagp
                Don't initialize the AGP driver and use full aperture.

iommu=          [PPC/POWERNV]

        nobypass
                Disable IOMMU bypass, using IOMMU for PCI devices.

iommu.forcedac= [ARM64,X86,EARLY] Control IOVA allocation for PCI devices.
                Format: { "0" | "1" }
                0 - Try to allocate a 32-bit DMA address first, before
                  falling back to the full range if needed.
                1 - Allocate directly from the full usable range,
```

```
                  forcing Dual Address Cycle for PCI cards supporting
                  greater than 32-bit addressing.

    iommu.strict=   [ARM64,X86,S390,EARLY] Configure TLB invalidation behaviour
                    Format: { "0" | "1" }
                    0 - Lazy mode.
                      Request that DMA unmap operations use deferred
                      invalidation of hardware TLBs, for increased
                      throughput at the cost of reduced device isolation.
                      Will fall back to strict mode if not supported by
                      the relevant IOMMU driver.
                    1 - Strict mode.
                      DMA unmap operations invalidate IOMMU hardware TLBs
                      synchronously.
                    unset - Use value of CONFIG_IOMMU_DEFAULT_DMA_{LAZY,STRICT}
                    Note: on x86, strict mode specified via one of the
                    legacy driver-specific options takes precedence.

    iommu.passthrough=
                    [ARM64,X86,EARLY] Configure DMA to bypass the IOMMU by defa
                    Format: { "0" | "1" }
                    0 - Use IOMMU translation for DMA.
                    1 - Bypass the IOMMU for DMA.
                    unset - Use value of CONFIG_IOMMU_DEFAULT_PASSTHROUGH.

    iommu.debug_pagealloc=
                    [KNL,EARLY] When CONFIG_IOMMU_DEBUG_PAGEALLOC is set, this
                    parameter enables the feature at boot time. By default, it
                    is disabled and the system behaves the same way as a kernel
                    built without CONFIG_IOMMU_DEBUG_PAGEALLOC.
                    Format: { "0" | "1" }
                    0 - Sanitizer disabled.
                    1 - Sanitizer enabled, expect runtime overhead.

    io7=            [HW] IO7 for Marvel-based Alpha systems
                    See comment before marvel_specify_io7 in
                    arch/alpha/kernel/core_marvel.c.

    io_delay=       [X86,EARLY] I/O delay method
            0x80
                    Standard port 0x80 based delay
            0xed
                    Alternate port 0xed based delay (needed on some systems)
            udelay
                    Simple two microseconds delay
            none
                    No delay

    ip=             [IP_PNP]
                    See Documentation/admin-guide/nfs/nfsroot.rst.

    ipcmni_extend   [KNL,EARLY] Extend the maximum number of unique System V
```

```
                IPC identifiers from 32,768 to 16,777,216.

    ipe.enforce=    [IPE]
                Format: <bool>
                Determine whether IPE starts in permissive (0) or
                enforce (1) mode. The default is enforce.

    ipe.success_audit=
                [IPE]
                Format: <bool>
                Start IPE with success auditing enabled, emitting
                an audit event when a binary is allowed. The default
                is 0.

    irqaffinity=    [SMP] Set the default irq affinity mask
                The argument is a cpu list, as described above.

    irqchip.gicv2_force_probe=
                [ARM,ARM64,EARLY]
                Format: <bool>
                Force the kernel to look for the second 4kB page
                of a GICv2 controller even if the memory range
                exposed by the device tree is too small.

    irqchip.gicv3_nolpi=
                [ARM,ARM64,EARLY]
                Force the kernel to ignore the availability of
                LPIs (and by consequence ITSs). Intended for system
                that use the kernel as a bootloader, and thus want
                to let secondary kernels in charge of setting up
                LPIs.

    irqchip.gicv3_pseudo_nmi= [ARM64,EARLY]
                Enables support for pseudo-NMIs in the kernel. This
                requires the kernel to be built with
                CONFIG_ARM64_PSEUDO_NMI.

    irqchip.riscv_imsic_noipi
                [RISC-V,EARLY]
                Force the kernel to not use IMSIC software injected MSIs
                as IPIs. Intended for system where IMSIC is trap-n-emulated
                and thus want to reduce MMIO traps when triggering IPIs
                to multiple harts.

    irqfixup        [HW]
                When an interrupt is not handled search all handlers
                for it. Intended to get systems with badly broken
                firmware running.

    irqhandler.duration_warn_us= [KNL]
                Warn if an IRQ handler exceeds the specified duration
                threshold in microseconds. Useful for identifying
```

```
                long-running IRQs in the system.

    irqpoll         [HW]
                    When an interrupt is not handled search all handlers
                    for it. Also check all handlers each timer
                    interrupt. Intended to get systems with badly broken
                    firmware running.

    isapnp=         [ISAPNP]
                    Format: <RDP>,<reset>,<pci_scan>,<verbosity>

    isolcpus=       [KNL,SMP,ISOL] Isolate a given set of CPUs from disturbance
                    Format: [flag-list,]<cpu-list>

                    Specify one or more CPUs to isolate from disturbances
                    specified in the flag list (default: domain):

                    nohz
                      Disable the tick when a single task runs as well as
                      disabling other kernel noises like having RCU callbacks
                      offloaded. This is equivalent to the nohz_full parameter.

                      A residual 1Hz tick is offloaded to workqueues, which you
                      need to affine to housekeeping through the global
                      workqueue's affinity configured via the
                      /sys/devices/virtual/workqueue/cpumask sysfs file, or
                      by using the 'domain' flag described below.

                      NOTE: by default the global workqueue runs on all CPUs,
                      so to protect individual CPUs the 'cpumask' file has to
                      be configured manually after bootup.

                    domain
                      Isolate from the general SMP balancing and scheduling
                      algorithms. Note that performing domain isolation this wa
                      is irreversible: it's not possible to bring back a CPU to
                      the domains once isolated through this boot time
                      configuration. Use cpusets for a dynamic configuration
                      which can be altered at runtime. For details see
                      Documentation/admin-guide/cpu-isolation.rst.

                      You can move a process onto or off an "isolated" CPU via
                      the CPU affinity syscalls or cpuset.
                      <cpu number> begins at 0 and the maximum value is
                      "number of CPUs in system - 1".

                    managed_irq

                      Isolate from being targeted by managed interrupts
                      which have an interrupt mask containing isolated
                      CPUs. The affinity of managed interrupts is
                      handled by the kernel and cannot be changed via
```

```
                        the /proc/irq/* interfaces.

                        This isolation is best effort and only effective
                        if the automatically assigned interrupt mask of a
                        device queue contains isolated and housekeeping
                        CPUs. If housekeeping CPUs are online then such
                        interrupts are directed to the housekeeping CPU
                        so that IO submitted on the housekeeping CPU
                        cannot disturb the isolated CPU.

                        If a queue's affinity mask contains only isolated
                        CPUs then this parameter has no effect on the
                        interrupt routing decision, though interrupts are
                        only delivered when tasks running on those
                        isolated CPUs submit IO. IO submitted on
                        housekeeping CPUs has no influence on those
                        queues.

                    The format of <cpu-list> is described above.

        iucv=       [HW,NET]

        ivrs_ioapic [HW,X86-64]
                    Provide an override to the IOAPIC-ID<->DEVICE-ID
                    mapping provided in the IVRS ACPI table.
                    By default, PCI segment is 0, and can be omitted.

                    For example, to map IOAPIC-ID decimal 10 to
                    PCI segment 0x1 and PCI device 00:14.0,
                    write the parameter as:
                            ivrs_ioapic=10@0001:00:14.0

                    Deprecated formats:
                    * To map IOAPIC-ID decimal 10 to PCI device 00:14.0
                      write the parameter as:
                            ivrs_ioapic[10]=00:14.0
                    * To map IOAPIC-ID decimal 10 to PCI segment 0x1 and
                      PCI device 00:14.0 write the parameter as:
                            ivrs_ioapic[10]=0001:00:14.0

        ivrs_hpet   [HW,X86-64]
                    Provide an override to the HPET-ID<->DEVICE-ID
                    mapping provided in the IVRS ACPI table.
                    By default, PCI segment is 0, and can be omitted.

                    For example, to map HPET-ID decimal 10 to
                    PCI segment 0x1 and PCI device 00:14.0,
                    write the parameter as:
                            ivrs_hpet=10@0001:00:14.0

                    Deprecated formats:
                    * To map HPET-ID decimal 0 to PCI device 00:14.0
```

```
                        write the parameter as:
                              ivrs_hpet[0]=00:14.0
                        * To map HPET-ID decimal 10 to PCI segment 0x1 and
                          PCI device 00:14.0 write the parameter as:
                              ivrs_ioapic[10]=0001:00:14.0

        ivrs_acpihid    [HW,X86-64]
                        Provide an override to the ACPI-HID:UID<->DEVICE-ID
                        mapping provided in the IVRS ACPI table.
                        By default, PCI segment is 0, and can be omitted.

                        For example, to map UART-HID:UID AMD0020:0 to
                        PCI segment 0x1 and PCI device ID 00:14.5,
                        write the parameter as:
                              ivrs_acpihid=AMD0020:0@0001:00:14.5

                        Deprecated formats:
                        * To map UART-HID:UID AMD0020:0 to PCI segment is 0,
                          PCI device ID 00:14.5, write the parameter as:
                              ivrs_acpihid[00:14.5]=AMD0020:0
                        * To map UART-HID:UID AMD0020:0 to PCI segment 0x1 and
                          PCI device ID 00:14.5, write the parameter as:
                              ivrs_acpihid[0001:00:14.5]=AMD0020:0

        js=             [HW,JOY] Analog joystick
                        See Documentation/input/joydev/joystick.rst.

        kasan_multi_shot
                        [KNL] Enforce KASAN (Kernel Address Sanitizer) to print
                        report on every invalid memory access. Without this
                        parameter KASAN will print report only for the first
                        invalid access.

        keep_bootcon    [KNL,EARLY]
                        Do not unregister boot console at start. This is only
                        useful for debugging when something happens in the window
                        between unregistering the boot console and initializing
                        the real console.

        keepinitrd      [HW,ARM] See retain_initrd.

        kernelcore=     [KNL,X86,PPC,EARLY]
                        Format: nn[KMGTPE] | nn% | "mirror"
                        This parameter specifies the amount of memory usable by
                        the kernel for non-movable allocations.  The requested
                        amount is spread evenly throughout all nodes in the
                        system as ZONE_NORMAL.  The remaining memory is used for
                        movable memory in its own zone, ZONE_MOVABLE.  In the
                        event, a node is too small to have both ZONE_NORMAL and
                        ZONE_MOVABLE, kernelcore memory will take priority and
                        other nodes will have a larger ZONE_MOVABLE.
```

```
                ZONE_MOVABLE is used for the allocation of pages that
                may be reclaimed or moved by the page migration
                subsystem.  Note that allocations like PTEs-from-HighMem
                still use the HighMem zone if it exists, and the Normal
                zone if it does not.

                It is possible to specify the exact amount of memory in
                the form of "nn[KMGTPE]", a percentage of total system
                memory in the form of "nn%", or "mirror".  If "mirror"
                option is specified, mirrored (reliable) memory is used
                for non-movable allocations and remaining memory is used
                for Movable pages.  "nn[KMGTPE]", "nn%", and "mirror"
                are exclusive, so you cannot specify multiple forms.

    kfence.burst=   [MM,KFENCE] The number of additional successive
                allocations to be attempted through KFENCE for each
                sample interval.
                Format: <unsigned integer>
                Default: 0

    kfence.check_on_panic=
                [MM,KFENCE] Whether to check all KFENCE-managed objects'
                canaries on panic.
                Format: <bool>
                Default: false

    kfence.deferrable=
                [MM,KFENCE] Whether to use a deferrable timer to trigger
                allocations. This avoids forcing CPU wake-ups if the
                system is idle, at the risk of a less predictable
                sample interval.
                Format: <bool>
                Default: CONFIG_KFENCE_DEFERRABLE

    kfence.fault=   [MM,KFENCE] Controls the behavior when a KFENCE
                error is detected.
                report - print the error report and continue (default).
                oops   - print the error report and oops.
                panic  - print the error report and panic.

    kfence.sample_interval=
                [MM,KFENCE] KFENCE's sample interval in milliseconds.
                Format: <unsigned integer>
                 0 - Disable KFENCE.
                >0 - Enabled KFENCE with given sample interval.
                Default: CONFIG_KFENCE_SAMPLE_INTERVAL

    kfence.skip_covered_thresh=
                [MM,KFENCE] If pool utilization reaches this threshold
                (pool usage%), KFENCE limits currently covered
                allocations of the same source from further filling
                up the pool.
```

```
                Format: <unsigned integer>
                Default: 75

    kgdbdbgp=   [KGDB,HW,EARLY] kgdb over EHCI usb debug port.
                Format: <Controller#>[,poll interval]
                The controller # is the number of the ehci usb debug
                port as it is probed via PCI.  The poll interval is
                optional and is the number seconds in between
                each poll cycle to the debug port in case you need
                the functionality for interrupting the kernel with
                gdb or control-c on the dbgp connection.  When
                not using this parameter you use sysrq-g to break into
                the kernel debugger.

    kgdboc=     [KGDB,HW] kgdb over consoles.
                Requires a tty driver that supports console polling,
                or a supported polling keyboard driver (non-usb).
                 Serial only format: <serial_device>[,baud]
                 keyboard only format: kbd
                 keyboard and serial format: kbd,<serial_device>[,baud]
                Optional Kernel mode setting:
                 kms, kbd format: kms,kbd
                 kms, kbd and serial format: kms,kbd,<ser_dev>[,baud]

    kgdboc_earlycon=        [KGDB,HW,EARLY]
                If the boot console provides the ability to read
                characters and can work in polling mode, you can use
                this parameter to tell kgdb to use it as a backend
                until the normal console is registered. Intended to
                be used together with the kgdboc parameter which
                specifies the normal console to transition to.

                The name of the early console should be specified
                as the value of this parameter. Note that the name of
                the early console might be different than the tty
                name passed to kgdboc. It's OK to leave the value
                blank and the first boot console that implements
                read() will be picked.

    kgdbwait    [KGDB,EARLY] Stop kernel execution and enter the
                kernel debugger at the earliest opportunity.

    kho=        [KEXEC,EARLY]
                Format: { "0" | "1" | "off" | "on" | "y" | "n" }
                Enables or disables Kexec HandOver.
                "0" | "off" | "n" - kexec handover is disabled
                "1" | "on" | "y" - kexec handover is enabled

    kho_scratch=    [KEXEC,EARLY]
                Format: ll[KMG],mm[KMG],nn[KMG] | nn%
                Defines the size of the KHO scratch region. The KHO
                scratch regions are physically contiguous memory
```

```
                ranges that can only be used for non-kernel
                allocations. That way, even when memory is heavily
                fragmented with handed over memory, the kexeced
                kernel will always have enough contiguous ranges to
                bootstrap itself.

                It is possible to specify the exact amount of
                memory in the form of "ll[KMG],mm[KMG],nn[KMG]"
                where the first parameter defines the size of a low
                memory scratch area, the second parameter defines
                the size of a global scratch area and the third
                parameter defines the size of additional per-node
                scratch areas.  The form "nn%" defines scale factor
                (in percents) of memory that was used during boot.

    kmac=       [MIPS] Korina ethernet MAC address.
                Configure the RouterBoard 532 series on-chip
                Ethernet adapter MAC address.

    kmemleak=   [KNL,EARLY] Boot-time kmemleak enable/disable
                Valid arguments: on, off
                Default: on
                Built with CONFIG_DEBUG_KMEMLEAK_DEFAULT_OFF=y,
                the default is off.

    kprobe_event=[probe-list]
                [FTRACE] Add kprobe events and enable at boot time.
                The probe-list is a semicolon delimited list of probe
                definitions. Each definition is same as kprobe_events
                interface, but the parameters are comma delimited.
                For example, to add a kprobe event on vfs_read with
                arg1 and arg2, add to the command line;

                      kprobe_event=p,vfs_read,$arg1,$arg2

                See also Documentation/trace/kprobetrace.rst "Kernel
                Boot Parameter" section.

    kpti=       [ARM64,EARLY] Control page table isolation of
                user and kernel address spaces.
                Default: enabled on cores which need mitigation.
                0: force disabled
                1: force enabled

    kunit.enable=   [KUNIT] Enable executing KUnit tests. Requires
                CONFIG_KUNIT to be set to be fully enabled. The
                default value can be overridden via
                KUNIT_DEFAULT_ENABLED.
                Default is 1 (enabled)

    kvm.ignore_msrs=[KVM] Ignore guest accesses to unhandled MSRs.
                Default is 0 (don't ignore, but inject #GP)
```

```
kvm.eager_page_split=
                [KVM,X86] Controls whether or not KVM will try to
                proactively split all huge pages during dirty logging.
                Eager page splitting reduces interruptions to vCPU
                execution by eliminating the write-protection faults
                and MMU lock contention that would otherwise be
                required to split huge pages lazily.

                VM workloads that rarely perform writes or that write
                only to a small region of VM memory may benefit from
                disabling eager page splitting to allow huge pages to
                still be used for reads.

                The behavior of eager page splitting depends on whether
                KVM_DIRTY_LOG_INITIALLY_SET is enabled or disabled. If
                disabled, all huge pages in a memslot will be eagerly
                split when dirty logging is enabled on that memslot. If
                enabled, eager page splitting will be performed during
                the KVM_CLEAR_DIRTY ioctl, and only for the pages being
                cleared.

                Eager page splitting is only supported when kvm.tdp_mmu=Y.

                Default is Y (on).

kvm.enable_pmu=[KVM,X86]
                If enabled, KVM will virtualize PMU functionality based
                on the virtual CPU model defined by userspace.  This
                can be overridden on a per-VM basis via
                KVM_CAP_PMU_CAPABILITY.

                If disabled, KVM will not virtualize PMU functionality,
                e.g. MSRs, PMCs, PMIs, etc., even if userspace defines
                a virtual CPU model that contains PMU assets.

                Note, KVM's vPMU support implicitly requires running
                with an in-kernel local APIC, e.g. to deliver PMIs to
                the guest.  Running without an in-kernel local APIC is
                not supported, though KVM will allow such a combination
                (with severely degraded functionality).

                See also enable_mediated_pmu.

                Default is Y (on).

kvm.enable_virt_at_load=[KVM,ARM64,LOONGARCH,MIPS,RISCV,X86]
                If enabled, KVM will enable virtualization in hardware
                when KVM is loaded, and disable virtualization when KVM
                is unloaded (if KVM is built as a module).

                If disabled, KVM will dynamically enable and disable
```

```
                virtualization on-demand when creating and destroying
                VMs, i.e. on the 0=>1 and 1=>0 transitions of the
                number of VMs.

                Enabling virtualization at module load avoids potential
                latency for creation of the 0=>1 VM, as KVM serializes
                virtualization enabling across all online CPUs.  The
                "cost" of enabling virtualization when KVM is loaded,
                is that doing so may interfere with using out-of-tree
                hypervisors that want to "own" virtualization hardware.

kvm.enable_vmware_backdoor=[KVM] Support VMware backdoor PV interface.
                           Default is false (don't support).

kvm.nx_huge_pages=
                [KVM] Controls the software workaround for the
                X86_BUG_ITLB_MULTIHIT bug.
                force   : Always deploy workaround.
                off     : Never deploy workaround.
                auto    : Deploy workaround based on the presence of
                          X86_BUG_ITLB_MULTIHIT.

                Default is 'auto'.

                If the software workaround is enabled for the host,
                guests do need not to enable it for nested guests.

kvm.nx_huge_pages_recovery_ratio=
                [KVM] Controls how many 4KiB pages are periodically zapped
                back to huge pages.  0 disables the recovery, otherwise if
                the value is N KVM will zap 1/Nth of the 4KiB pages every
                period (see below).  The default is 60.

kvm.nx_huge_pages_recovery_period_ms=
                [KVM] Controls the time period at which KVM zaps 4KiB pages
                back to huge pages. If the value is a non-zero N, KVM will
                zap a portion (see ratio above) of the pages every N msecs.
                If the value is 0 (the default), KVM will pick a period bas
                on the ratio, such that a page is zapped after 1 hour on av

kvm-{amd,intel}.enable_mediated_pmu=[KVM,AMD,INTEL]
                If enabled, KVM will provide a mediated virtual PMU,
                instead of the default perf-based virtual PMU (if
                kvm.enable_pmu is true and PMU is enumerated via the
                virtual CPU model).

                With a perf-based vPMU, KVM operates as a user of perf,
                i.e. emulates guest PMU counters using perf events.
                KVM-created perf events are managed by perf as regular
                (guest-only) events, e.g. are scheduled in/out, contend
                for hardware resources, etc.  Using a perf-based vPMU
                allows guest and host usage of the PMU to co-exist, but
```

```
                incurs non-trivial overhead and can result in silently
                dropped guest events (due to resource contention).

                With a mediated vPMU, hardware PMU state is context
                switched around the world switch to/from the guest.
                KVM mediates which events the guest can utilize, but
                gives the guest direct access to all other PMU assets
                when possible (KVM may intercept some accesses if the
                virtual CPU model provides a subset of hardware PMU
                functionality).  Using a mediated vPMU significantly
                reduces PMU virtualization overhead and eliminates lost
                guest events, but is mutually exclusive with using perf
                to profile KVM guests and adds latency to most VM-Exits
                (to context switch PMU state).

                Default is N (off).

    kvm-amd.nested= [KVM,AMD] Control nested virtualization feature in
                KVM/SVM. Default is 1 (enabled).

    kvm-amd.npt=    [KVM,AMD] Control KVM's use of Nested Page Tables,
                a.k.a. Two-Dimensional Page Tables. Default is 1
                (enabled). Disable by KVM if hardware lacks support
                for NPT.

    kvm-amd.ciphertext_hiding_asids=
                [KVM,AMD] Ciphertext hiding prevents disallowed accesses
                to SNP private memory from reading ciphertext.  Instead,
                reads will see constant default values (0xff).

                If ciphertext hiding is enabled, the joint SEV-ES and
                SEV-SNP ASID space is partitioned into separate SEV-ES
                and SEV-SNP ASID ranges, with the SEV-SNP range being
                [1..max_snp_asid] and the SEV-ES range being
                (max_snp_asid..min_sev_asid), where min_sev_asid is
                enumerated by CPUID.0x.8000_001F[EDX].

                A non-zero value enables SEV-SNP ciphertext hiding and
                adjusts the ASID ranges for SEV-ES and SEV-SNP guests.
                KVM caps the number of SEV-SNP ASIDs at the maximum
                possible value, e.g. specifying -1u will assign all
                joint SEV-ES and SEV-SNP ASIDs to SEV-SNP.  Note,
                assigning all joint ASIDs to SEV-SNP, i.e. configuring
                max_snp_asid == min_sev_asid-1, will effectively make
                SEV-ES unusable.

    kvm-arm.mode=
                [KVM,ARM,EARLY] Select one of KVM/arm64's modes of
                operation.

                none: Forcefully disable KVM.
```

```
                  nvhe: Standard nVHE-based mode, without support for
                        protected guests.

                  protected: Mode with support for guests whose state is
                             kept private from the host, using VHE or
                             nVHE depending on HW support.

                  nested: VHE-based mode with support for nested
                          virtualization. Requires at least ARMv8.4
                          hardware (with FEAT_NV2).

                  Defaults to VHE/nVHE based on hardware support. Setting
                  mode to "protected" will disable kexec and hibernation
                  for the host. To force nVHE on VHE hardware, add
                  "arm64_sw.hvhe=0 id_aa64mmfr1.vh=0" to the
                  command-line.
                  "nested" and "protected" are experimental and should be
                  used with extreme caution.

kvm-arm.vgic_v3_group0_trap=
                  [KVM,ARM,EARLY] Trap guest accesses to GICv3 group-0
                  system registers

kvm-arm.vgic_v3_group1_trap=
                  [KVM,ARM,EARLY] Trap guest accesses to GICv3 group-1
                  system registers

kvm-arm.vgic_v3_common_trap=
                  [KVM,ARM,EARLY] Trap guest accesses to GICv3 common
                  system registers

kvm-arm.vgic_v4_enable=
                  [KVM,ARM,EARLY] Allow use of GICv4 for direct
                  injection of LPIs.

kvm-arm.wfe_trap_policy=
                  [KVM,ARM] Control when to set WFE instruction trap for
                  KVM VMs. Traps are allowed but not guaranteed by the
                  CPU architecture.

                  trap: set WFE instruction trap

                  notrap: clear WFE instruction trap

kvm-arm.wfi_trap_policy=
                  [KVM,ARM] Control when to set WFI instruction trap for
                  KVM VMs. Traps are allowed but not guaranteed by the
                  CPU architecture.

                  trap: set WFI instruction trap

                  notrap: clear WFI instruction trap
```

```
kvm_cma_resv_ratio=n [PPC,EARLY]
                Reserves given percentage from system memory area for
                contiguous memory allocation for KVM hash pagetable
                allocation.
                By default it reserves 5% of total system memory.
                Format: <integer>
                Default: 5

kvm-intel.ept=  [KVM,Intel] Control KVM's use of Extended Page Tables,
                a.k.a. Two-Dimensional Page Tables.  Default is 1
                (enabled). Disable by KVM if hardware lacks support
                for EPT.

kvm-intel.emulate_invalid_guest_state=
                [KVM,Intel] Control whether to emulate invalid guest
                state. Ignored if kvm-intel.enable_unrestricted_guest=1,
                as guest state is never invalid for unrestricted
                guests. This param doesn't apply to nested guests (L2),
                as KVM never emulates invalid L2 guest state.
                Default is 1 (enabled).

kvm-intel.flexpriority=
                [KVM,Intel] Control KVM's use of FlexPriority feature
                (TPR shadow). Default is 1 (enabled). Disable by KVM if
                hardware lacks support for it.

kvm-intel.nested=
                [KVM,Intel] Control nested virtualization feature in
                KVM/VMX. Default is 1 (enabled).

kvm-intel.unrestricted_guest=
                [KVM,Intel] Control KVM's use of unrestricted guest
                feature (virtualized real and unpaged mode). Default
                is 1 (enabled). Disable by KVM if EPT is disabled or
                hardware lacks support for it.

kvm-intel.vmentry_l1d_flush=[KVM,Intel] Mitigation for L1 Terminal Fault
                CVE-2018-3620.

                Valid arguments: never, cond, always

                always: L1D cache flush on every VMENTER.
                cond:   Flush L1D on VMENTER only when the code between
                        VMEXIT and VMENTER can leak host memory.
                never:  Disables the mitigation

                Default is cond (do L1 cache flush in specific instances)

kvm-intel.vpid= [KVM,Intel] Control KVM's use of Virtual Processor
                Identification feature (tagged TLBs). Default is 1
                (enabled). Disable by KVM if hardware lacks support
```

```
                        for it.

        l1d_flush=      [X86,INTEL,EARLY]
                        Control mitigation for L1D based snooping vulnerability.

                        Certain CPUs are vulnerable to an exploit against CPU
                        internal buffers which can forward information to a
                        disclosure gadget under certain conditions.

                        In vulnerable processors, the speculatively
                        forwarded data can be used in a cache side channel
                        attack, to access data to which the attacker does
                        not have direct access.

                        This parameter controls the mitigation. The
                        options are:

                        on         - enable the interface for the mitigation

        l1tf=           [X86,EARLY] Control mitigation of the L1TF vulnerability on
                              affected CPUs

                        The kernel PTE inversion protection is unconditionally
                        enabled and cannot be disabled.

                        full
                                Provides all available mitigations for the
                                L1TF vulnerability. Disables SMT and
                                enables all mitigations in the
                                hypervisors, i.e. unconditional L1D flush.

                                SMT control and L1D flush control via the
                                sysfs interface is still possible after
                                boot.  Hypervisors will issue a warning
                                when the first VM is started in a
                                potentially insecure configuration,
                                i.e. SMT enabled or L1D flush disabled.

                        full,force
                                Same as 'full', but disables SMT and L1D
                                flush runtime control. Implies the
                                'nosmt=force' command line option.
                                (i.e. sysfs control of SMT is disabled.)

                        flush
                                Leaves SMT enabled and enables the default
                                hypervisor mitigation, i.e. conditional
                                L1D flush.

                                SMT control and L1D flush control via the
                                sysfs interface is still possible after
                                boot.  Hypervisors will issue a warning
```

```
                        when the first VM is started in a
                        potentially insecure configuration,
                        i.e. SMT enabled or L1D flush disabled.

                flush,nosmt

                        Disables SMT and enables the default
                        hypervisor mitigation.

                        SMT control and L1D flush control via the
                        sysfs interface is still possible after
                        boot.  Hypervisors will issue a warning
                        when the first VM is started in a
                        potentially insecure configuration,
                        i.e. SMT enabled or L1D flush disabled.

                flush,nowarn
                        Same as 'flush', but hypervisors will not
                        warn when a VM is started in a potentially
                        insecure configuration.

                off
                        Disables hypervisor mitigations and doesn't
                        emit any warnings.
                        It also drops the swap size and available
                        RAM limit restriction on both hypervisor and
                        bare metal.

                Default is 'flush'.

                For details see: Documentation/admin-guide/hw-vuln/l1tf.rst

    l2cr=       [PPC]

    l3cr=       [PPC]

    lapic       [X86-32,APIC,EARLY] Enable the local APIC even if BIOS
                disabled it.

    lapic=      [X86,APIC] Do not use TSC deadline
                value for LAPIC timer one-shot implementation. Default
                back to the programmable timer unit in the LAPIC.
                Format: notscdeadline

    lapic_timer_c2_ok       [X86,APIC,EARLY] trust the local apic timer
                in C2 power state.

    libata.dma= [LIBATA] DMA control
                libata.dma=0      Disable all PATA and SATA DMA
                libata.dma=1      PATA and SATA Disk DMA only
                libata.dma=2      ATAPI (CDROM) DMA only
                libata.dma=4      Compact Flash DMA only
```

```
                Combinations also work, so libata.dma=3 enables DMA
                for disks and CDROMs, but not CFs.

    libata.ignore_hpa=      [LIBATA] Ignore HPA limit
                libata.ignore_hpa=0       keep BIOS limits (default)
                libata.ignore_hpa=1       ignore limits, using full disk

    libata.noacpi   [LIBATA] Disables use of ACPI in libata suspend/resume
                when set.
                Format: <int>

    libata.force=   [LIBATA] Force configurations.  The format is a comma-
                separated list of "[ID:]VAL" where ID is PORT[.DEVICE].
                PORT and DEVICE are decimal numbers matching port, link
                or device.  Basically, it matches the ATA ID string
                printed on console by libata.  If the whole ID part is
                omitted, the last PORT and DEVICE values are used.  If
                ID hasn't been specified yet, the configuration applies
                to all ports, links and devices.

                If only DEVICE is omitted, the parameter applies to
                the port and all links and devices behind it.  DEVICE
                number of 0 either selects the first device or the
                first fan-out link behind PMP device.  It does not
                select the host link.  DEVICE number of 15 selects the
                host link and device attached to it.

                The VAL specifies the configuration to force.  As long
                as there is no ambiguity, shortcut notation is allowed.
                For example, both 1.5 and 1.5G would work for 1.5Gbps.
                The following configurations can be forced.

                * Cable type: 40c, 80c, short40c, unk, ign or sata.
                  Any ID with matching PORT is used.

                * SATA link speed limit: 1.5Gbps or 3.0Gbps.

                * Transfer mode: pio[0-7], mwdma[0-4] and udma[0-7].
                  udma[/][16,25,33,44,66,100,133] notation is also
                  allowed.

                * nohrst, nosrst, norst: suppress hard, soft and both
                  resets.

                * rstonce: only attempt one reset during hot-unplug
                  link recovery.

                * [no]dbdelay: Enable or disable the extra 200ms delay
                  before debouncing a link PHY and device presence
                  detection.

                * [no]ncq: Turn on or off NCQ.
```

```
                * [no]ncqtrim: Enable or disable queued DSM TRIM.

                * [no]ncqati: Enable or disable NCQ trim on ATI chipset.

                * [no]trim: Enable or disable (unqueued) TRIM.

                * trim_zero: Indicate that TRIM command zeroes data.

                * max_trim_128m: Set 128M maximum trim size limit.

                * [no]dma: Turn on or off DMA transfers.

                * atapi_dmadir: Enable ATAPI DMADIR bridge support.

                * atapi_mod16_dma: Enable the use of ATAPI DMA for
                  commands that are not a multiple of 16 bytes.

                * [no]dmalog: Enable or disable the use of the
                  READ LOG DMA EXT command to access logs.

                * [no]iddevlog: Enable or disable access to the
                  identify device data log.

                * [no]logdir: Enable or disable access to the general
                  purpose log directory.

                * max_sec=<sectors>: Set the transfer size limit, in
                  number of 512-byte sectors, to the value specified in
                  <sectors>. The value specified in <sectors> has to be
                  a non-zero positive integer.

                * max_sec_128: Set transfer size limit to 128 sectors.

                * max_sec_1024: Set or clear transfer size limit to
                  1024 sectors.

                * max_sec_lba48: Set or clear transfer size limit to
                  65535 sectors.

                * external: Mark port as external (hotplug-capable).

                * [no]lpm: Enable or disable link power management.

                * [no]setxfer: Indicate if transfer speed mode setting
                  should be skipped.

                * [no]fua: Disable or enable FUA (Force Unit Access)
                  support for devices supporting this feature.

                * dump_id: Dump IDENTIFY data.
```

```
                * disable: Disable this device.

                If there are multiple matching configurations changing
                the same attribute, the last one is used.

    liveupdate=     [KNL,EARLY]
                Format: <bool>
                Enable Live Update Orchestrator (LUO).
                Default: off.

    lockd.nlm_grace_period=P  [NFS] Assign grace period.
                Format: <integer>

    lockd.nlm_tcpport=N     [NFS] Assign TCP port.
                Format: <integer>

    lockd.nlm_timeout=T     [NFS] Assign timeout value.
                Format: <integer>

    lockd.nlm_udpport=M     [NFS] Assign UDP port.
                Format: <integer>

    lockdown=       [SECURITY,EARLY]
                { integrity | confidentiality }
                Enable the kernel lockdown feature. If set to
                integrity, kernel features that allow userland to
                modify the running kernel are disabled. If set to
                confidentiality, kernel features that allow userland
                to extract confidential information from the kernel
                are also disabled.

    locktorture.acq_writer_lim= [KNL]
                Set the time limit in jiffies for a lock
                acquisition.  Acquisitions exceeding this limit
                will result in a splat once they do complete.

    locktorture.bind_readers= [KNL]
                Specify the list of CPUs to which the readers are
                to be bound.

    locktorture.bind_writers= [KNL]
                Specify the list of CPUs to which the writers are
                to be bound.

    locktorture.call_rcu_chains= [KNL]
                Specify the number of self-propagating call_rcu()
                chains to set up.  These are used to ensure that
                there is a high probability of an RCU grace period
                in progress at any given time.  Defaults to 0,
                which disables these call_rcu() chains.

    locktorture.long_hold= [KNL]
```

```
                Specify the duration in milliseconds for the
                occasional long-duration lock hold time.  Defaults
                to 100 milliseconds.  Select 0 to disable.

    locktorture.nested_locks= [KNL]
                Specify the maximum lock nesting depth that
                locktorture is to exercise, up to a limit of 8
                (MAX_NESTED_LOCKS).  Specify zero to disable.
                Note that this parameter is ineffective on types
                of locks that do not support nested acquisition.

    locktorture.nreaders_stress= [KNL]
                Set the number of locking read-acquisition kthreads.
                Defaults to being automatically set based on the
                number of online CPUs.

    locktorture.nwriters_stress= [KNL]
                Set the number of locking write-acquisition kthreads.

    locktorture.onoff_holdoff= [KNL]
                Set time (s) after boot for CPU-hotplug testing.

    locktorture.onoff_interval= [KNL]
                Set time (s) between CPU-hotplug operations, or
                zero to disable CPU-hotplug testing.

    locktorture.rt_boost= [KNL]
                Do periodic testing of real-time lock priority
                boosting.  Select 0 to disable, 1 to boost
                only rt_mutex, and 2 to boost unconditionally.
                Defaults to 2, which might seem to be an
                odd choice, but which should be harmless for
                non-real-time spinlocks, due to their disabling
                of preemption.  Note that non-realtime mutexes
                disable boosting.

    locktorture.rt_boost_factor= [KNL]
                Number that determines how often and for how
                long priority boosting is exercised.  This is
                scaled down by the number of writers, so that the
                number of boosts per unit time remains roughly
                constant as the number of writers increases.
                On the other hand, the duration of each boost
                increases with the number of writers.

    locktorture.shuffle_interval= [KNL]
                Set task-shuffle interval (jiffies).  Shuffling
                tasks allows some CPUs to go into dyntick-idle
                mode during the locktorture test.

    locktorture.shutdown_secs= [KNL]
                Set time (s) after boot system shutdown.  This
```

```
                is useful for hands-off automated testing.

    locktorture.stat_interval= [KNL]
                Time (s) between statistics printk()s.

    locktorture.stutter= [KNL]
                Time (s) to stutter testing, for example,
                specifying five seconds causes the test to run for
                five seconds, wait for five seconds, and so on.
                This tests the locking primitive's ability to
                transition abruptly to and from idle.

    locktorture.torture_type= [KNL]
                Specify the locking implementation to test.

    locktorture.verbose= [KNL]
                Enable additional printk() statements.

    locktorture.writer_fifo= [KNL]
                Run the write-side locktorture kthreads at
                sched_set_fifo() real-time priority.

    logibm.irq=     [HW,MOUSE] Logitech Bus Mouse Driver
                Format: <irq>

    loglevel=       [KNL,EARLY]
                All Kernel Messages with a loglevel smaller than the
                console loglevel will be printed to the console. It can
                also be changed with klogd or other programs. The
                loglevels are defined as follows:

                0 (KERN_EMERG)          system is unusable
                1 (KERN_ALERT)          action must be taken immediately
                2 (KERN_CRIT)           critical conditions
                3 (KERN_ERR)            error conditions
                4 (KERN_WARNING)        warning conditions
                5 (KERN_NOTICE)         normal but significant condition
                6 (KERN_INFO)           informational
                7 (KERN_DEBUG)          debug-level messages

    log_buf_len=n[KMG] [KNL,EARLY]
                Sets the size of the printk ring buffer, in bytes.
                n must be a power of two and greater than the
                minimal size. The minimal size is defined by
                LOG_BUF_SHIFT kernel config parameter. There
                is also CONFIG_LOG_CPU_MAX_BUF_SHIFT config
                parameter that allows to increase the default size
                depending on the number of CPUs. See init/Kconfig
                for more details.

    logo.nologo     [FB] Disables display of the built-in Linux logo.
                This may be used to provide more screen space for
```

```
                kernel log messages and is useful when debugging
                kernel boot problems.

lp=0            [LP]    Specify parallel ports to use, e.g,
lp=port[,port...]       lp=none,parport0 (lp0 not configured, lp1 uses
lp=reset                first parallel port). 'lp=0' disables the
lp=auto                 printer driver. 'lp=reset' (which can be
                        specified in addition to the ports) causes
                        attached printers to be reset. Using
                        lp=port1,port2,... specifies the parallel ports
                        to associate lp devices with, starting with
                        lp0. A port specification may be 'none' to skip
                        that lp device, or a parport name such as
                        'parport0'. Specifying 'lp=auto' instead of a
                        port specification list means that device IDs
                        from each port should be examined, to see if
                        an IEEE 1284-compliant printer is attached; if
                        so, the driver will manage that printer.
                        See also header of drivers/char/lp.c.

lpj=n           [KNL]
                Sets loops_per_jiffy to given constant, thus avoiding
                time-consuming boot-time autodetection (up to 250 ms per
                CPU). 0 enables autodetection (default). To determine
                the correct value for your kernel, boot with normal
                autodetection and see what value is printed. Note that
                on SMP systems the preset will be applied to all CPUs,
                which is likely to cause problems if your CPUs need
                significantly divergent settings. An incorrect value
                will cause delays in the kernel to be wrong, leading to
                unpredictable I/O errors and other breakage. Although
                unlikely, in the extreme case this might damage your
                hardware.

lsm.debug       [SECURITY] Enable LSM initialization debugging output.

lsm=lsm1,...,lsmN
                [SECURITY] Choose order of LSM initialization. This
                overrides CONFIG_LSM, and the "security=" parameter.

machtype=       [Loongson] Share the same kernel image file between
                different yeeloong laptops.
                Example: machtype=lemote-yeeloong-2f-7inch

maxcpus=        [SMP,EARLY] Maximum number of processors that an SMP kernel
                will bring up during bootup.  maxcpus=n : n >= 0 limits
                the kernel to bring up 'n' processors. Surely after
                bootup you can bring up the other plugged cpu by executing
                "echo 1 > /sys/devices/system/cpu/cpuX/online". So maxcpus
                only takes effect during system bootup.
                While n=0 is a special case, it is equivalent to "nosmp",
                which also disables the IO APIC.
```

```
max_loop=       [LOOP] The number of loop block devices that get
(loop.max_loop) unconditionally pre-created at init time. The default
                number is configured by BLK_DEV_LOOP_MIN_COUNT. Instead
                of statically allocating a predefined number, loop
                devices can be requested on-demand with the
                /dev/loop-control interface.

mce=            [X86-{32,64}]

                Please see Documentation/arch/x86/x86_64/machinecheck.rst f

        off
                disable machine check

        no_cmci
                disable CMCI(Corrected Machine Check Interrupt) that
                Intel processor supports.  Usually this disablement is
                not recommended, but it might be handy if your
                hardware is misbehaving.

                Note that you'll get more problems without CMCI than
                with due to the shared banks, i.e. you might get
                duplicated error logs.

        dont_log_ce
                don't make logs for corrected errors.  All events
                reported as corrected are silently cleared by OS. This
                option will be useful if you have no interest in any
                of corrected errors.

        ignore_ce
                disable features for corrected errors, e.g.
                polling timer and CMCI.  All events reported as
                corrected are not cleared by OS and remained in its
                error banks.

                Usually this disablement is not recommended, however
                if there is an agent checking/clearing corrected
                errors (e.g. BIOS or hardware monitoring
                applications), conflicting with OS's error handling,
                and you cannot deactivate the agent, then this option
                will be a help.

        no_lmce
                do not opt-in to Local MCE delivery. Use legacy method
                to broadcast MCEs.

        bootlog
                enable logging of machine checks left over from
                booting. Disabled by default on AMD Fam10h and older
                because some BIOS leave bogus ones.
```

```
                If your BIOS doesn't do that it's a good idea to
                enable though to make sure you log even machine check
                events that result in a reboot. On Intel systems it is
                enabled by default.

        nobootlog
                disable boot machine check logging.

        monarchtimeout (number)
                sets the time in us to wait for other CPUs on machine
                checks. 0 to disable.

        bios_cmci_threshold
                don't overwrite the bios-set CMCI threshold. This boot
                option prevents Linux from overwriting the CMCI
                threshold set by the bios.  Without this option, Linux
                always sets the CMCI threshold to 1. Enabling this may
                make memory predictive failure analysis less effective
                if the bios sets thresholds for memory errors since we
                will not see details for all errors.

        recovery
                force-enable recoverable machine check code paths

                Everything else is in sysfs now.


md=             [HW] RAID subsystems devices and level
                See Documentation/admin-guide/md.rst.

mds=            [X86,INTEL,EARLY]
                Control mitigation for the Micro-architectural Data
                Sampling (MDS) vulnerability.

                Certain CPUs are vulnerable to an exploit against CPU
                internal buffers which can forward information to a
                disclosure gadget under certain conditions.

                In vulnerable processors, the speculatively
                forwarded data can be used in a cache side channel
                attack, to access data to which the attacker does
                not have direct access.

                This parameter controls the MDS mitigation. The
                options are:

                full       - Enable MDS mitigation on vulnerable CPUs
                full,nosmt - Enable MDS mitigation and disable
                             SMT on vulnerable CPUs
                off        - Unconditionally disable MDS mitigation
```

```
                On TAA-affected machines, mds=off can be prevented by
                an active TAA mitigation as both vulnerabilities are
                mitigated with the same mechanism so in order to disable
                this mitigation, you need to specify tsx_async_abort=off
                too.

                Not specifying this option is equivalent to
                mds=full.

                For details see: Documentation/admin-guide/hw-vuln/mds.rst

    mem=nn[KMG] [HEXAGON,EARLY] Set the memory size.
                Must be specified, otherwise memory size will be 0.

    mem=nn[KMG] [KNL,BOOT,EARLY] Force usage of a specific amount
                of memory Amount of memory to be used in cases
                as follows:

                1 for test;
                2 when the kernel is not able to see the whole system memor
                3 memory that lies after 'mem=' boundary is excluded from
                 the hypervisor, then assigned to KVM guests.
                4 to limit the memory available for kdump kernel.

                [ARC,MICROBLAZE] - the limit applies only to low memory,
                high memory is not affected.

                [ARM64] - only limits memory covered by the linear
                mapping. The NOMAP regions are not affected.

                [X86] Work as limiting max address. Use together
                with memmap= to avoid physical address space collisions.
                Without memmap= PCI devices could be placed at addresses
                belonging to unused RAM.

                Note that this only takes effects during boot time since
                in above case 3, memory may need be hot added after boot
                if system memory of hypervisor is not sufficient.

    mem=nn[KMG]@ss[KMG]
                [ARM,MIPS,EARLY] - override the memory layout
                reported by firmware.
                Define a memory region of size nn[KMG] starting at
                ss[KMG].
                Multiple different regions can be specified with
                multiple mem= parameters on the command line.

    mem=nopentium   [BUGS=X86-32] Disable usage of 4MB pages for kernel
                memory.

    memblock=debug  [KNL,EARLY] Enable memblock debug messages.
```

```
        memchunk=nn[KMG]
                        [KNL,SH] Allow user to override the default size for
                        per-device physically contiguous DMA buffers.

        memhp_default_state=online/offline/online_kernel/online_movable
                        [KNL] Set the initial state for the memory hotplug
                        onlining policy. If not specified, the default value is
                        set according to the
                        CONFIG_MHP_DEFAULT_ONLINE_TYPE kernel config
                        options.
                        See Documentation/admin-guide/mm/memory-hotplug.rst.

        memmap=exactmap [KNL,X86,EARLY] Enable setting of an exact
                        E820 memory map, as specified by the user.
                        Such memmap=exactmap lines can be constructed based on
                        BIOS output or other requirements. See the memmap=nn@ss
                        option description.

        memmap=nn[KMG]@ss[KMG]
                        [KNL, X86,MIPS,XTENSA,EARLY] Force usage of a specific regi
                        Region of memory to be used is from ss to ss+nn.
                        If @ss[KMG] is omitted, it is equivalent to mem=nn[KMG],
                        which limits max address to nn[KMG].
                        Multiple different regions can be specified,
                        comma delimited.
                        Example:
                                memmap=100M@2G,100M#3G,1G!1024G

        memmap=nn[KMG]#ss[KMG]
                        [KNL,ACPI,EARLY] Mark specific memory as ACPI data.
                        Region of memory to be marked is from ss to ss+nn.

        memmap=nn[KMG]$ss[KMG]
                        [KNL,ACPI,EARLY] Mark specific memory as reserved.
                        Region of memory to be reserved is from ss to ss+nn.
                        Example: Exclude memory from 0x18690000-0x1869ffff
                                 memmap=64K$0x18690000
                                 or
                                 memmap=0x10000$0x18690000
                        Some bootloaders may need an escape character before '$',
                        like Grub2, otherwise '$' and the following number
                        will be eaten.

        memmap=nn[KMG]!ss[KMG,EARLY]
                        [KNL,X86] Mark specific memory as protected.
                        Region of memory to be used, from ss to ss+nn.
                        The memory region may be marked as e820 type 12 (0xc)
                        and is NVDIMM or ADR memory.

        memmap=<size>%<offset>-<oldtype>+<newtype>
                        [KNL,ACPI,EARLY] Convert memory within the specified region
                        from <oldtype> to <newtype>. If "-<oldtype>" is left
```

```
                out, the whole region will be marked as <newtype>,
                even if previously unavailable. If "+<newtype>" is left
                out, matching memory will be removed. Types are
                specified as e820 types, e.g., 1 = RAM, 2 = reserved,
                3 = ACPI, 12 = PRAM.

    memory_corruption_check=0/1 [X86,EARLY]
                Some BIOSes seem to corrupt the first 64k of
                memory when doing things like suspend/resume.
                Setting this option will scan the memory
                looking for corruption.  Enabling this will
                both detect corruption and prevent the kernel
                from using the memory being corrupted.
                However, it's intended as a diagnostic tool; if
                repeatable BIOS-originated corruption always
                affects the same memory, you can use memmap=
                to prevent the kernel from using that memory.

    memory_corruption_check_size=size [X86,EARLY]
                By default it checks for corruption in the low
                64k, making this memory unavailable for normal
                use.  Use this parameter to scan for
                corruption in more or less memory.

    memory_corruption_check_period=seconds [X86,EARLY]
                By default it checks for corruption every 60
                seconds.  Use this parameter to check at some
                other rate.  0 disables periodic checking.

    memory_hotplug.memmap_on_memory
                [KNL,X86,ARM] Boolean flag to enable this feature.
                Format: {on | off (default)}
                When enabled, runtime hotplugged memory will
                allocate its internal metadata (struct pages,
                those vmemmap pages cannot be optimized even
                if hugetlb_free_vmemmap is enabled) from the
                hotadded memory which will allow to hotadd a
                lot of memory without requiring additional
                memory to do so.
                This feature is disabled by default because it
                has some implication on large (e.g. GB)
                allocations in some configurations (e.g. small
                memory blocks).
                The state of the flag can be read in
                /sys/module/memory_hotplug/parameters/memmap_on_memory.
                Note that even when enabled, there are a few cases where
                the feature is not effective.

    memtest=    [KNL,X86,ARM,M68K,PPC,RISCV,EARLY] Enable memtest
                Format: <integer>
                default : 0 <disable>
                Specifies the number of memtest passes to be
```

```
                performed. Each pass selects another test
                pattern from a given set of patterns. Memtest
                fills the memory with this pattern, validates
                memory contents and reserves bad memory
                regions that are detected.

    mem_encrypt=    [X86-64] AMD Secure Memory Encryption (SME) control
                Valid arguments: on, off
                Default: off
                mem_encrypt=on:         Activate SME
                mem_encrypt=off:        Do not activate SME

                Refer to Documentation/virt/kvm/x86/amd-memory-encryption.r
                for details on when memory encryption can be activated.

    mem_sleep_default=      [SUSPEND] Default system suspend mode:
                s2idle  - Suspend-To-Idle
                shallow - Power-On Suspend or equivalent (if supported)
                deep    - Suspend-To-RAM or equivalent (if supported)
                See Documentation/admin-guide/pm/sleep-states.rst.

    mfgptfix        [X86-32] Fix MFGPT timers on AMD Geode platforms when
                the BIOS has incorrectly applied a workaround. TinyBIOS
                version 0.98 is known to be affected, 0.99 fixes the
                problem by letting the user disable the workaround.

    mga=            [HW,DRM]

    microcode=      [X86] Control the behavior of the microcode loader.
                Available options, comma separated:

                base_rev=X - with <X> with format: <u32>
                Set the base microcode revision of each thread when in
                debug mode.

                dis_ucode_ldr: disable the microcode loader

                force_minrev:
                Enable or disable the microcode minimal revision
                enforcement for the runtime microcode loader.

    mini2440=       [ARM,HW,KNL]
                Format:[0..2][b][c][t]
                Default: "0tb"
                MINI2440 configuration specification:
                0 - The attached screen is the 3.5" TFT
                1 - The attached screen is the 7" TFT
                2 - The VGA Shield is attached (1024x768)
                Leaving out the screen size parameter will not load
                the TFT driver, and the framebuffer will be left
                unconfigured.
                b - Enable backlight. The TFT backlight pin will be
```

```
                    linked to the kernel VESA blanking code and a GPIO
                    LED. This parameter is not necessary when using the
                    VGA shield.
                    c - Enable the s3c camera interface.
                    t - Reserved for enabling touchscreen support. The
                    touchscreen support is not enabled in the mainstream
                    kernel as of 2.6.30, a preliminary port can be found
                    in the "bleeding edge" mini2440 support kernel at
                    https://repo.or.cz/w/linux-2.6/mini2440.git

        mitigations=
                    [X86,PPC,S390,ARM64,EARLY] Control optional mitigations for
                    CPU vulnerabilities.  This is a set of curated,
                    arch-independent options, each of which is an
                    aggregation of existing arch-specific options.

                    Note, "mitigations" is supported if and only if the
                    kernel was built with CPU_MITIGATIONS=y.

                    off
                            Disable all optional CPU mitigations.  This
                            improves system performance, but it may also
                            expose users to several CPU vulnerabilities.
                            Equivalent to: if nokaslr then kpti=0 [ARM64]
                                           gather_data_sampling=off [X86]
                                           indirect_target_selection=off [X86]
                                           kvm.nx_huge_pages=off [X86]
                                           l1tf=off [X86]
                                           mds=off [X86]
                                           mmio_stale_data=off [X86]
                                           no_entry_flush [PPC]
                                           no_uaccess_flush [PPC]
                                           nobp=0 [S390]
                                           nopti [X86,PPC]
                                           nospectre_bhb [ARM64]
                                           nospectre_v1 [X86,PPC]
                                           nospectre_v2 [X86,PPC,S390,ARM64]
                                           reg_file_data_sampling=off [X86]
                                           retbleed=off [X86]
                                           spec_rstack_overflow=off [X86]
                                           spec_store_bypass_disable=off [X86,P
                                           spectre_bhi=off [X86]
                                           spectre_v2_user=off [X86]
                                           srbds=off [X86,INTEL]
                                           ssbd=force-off [ARM64]
                                           tsa=off [X86,AMD]
                                           tsx_async_abort=off [X86]
                                           vmscape=off [X86]

                            Exceptions:
                                           This does not have any effect on
                                           kvm.nx_huge_pages when
```

```
                                        kvm.nx_huge_pages=force.

                    auto (default)
                            Mitigate all CPU vulnerabilities, but leave SMT
                            enabled, even if it's vulnerable.  This is for
                            users who don't want to be surprised by SMT
                            getting disabled across kernel upgrades, or who
                            have other ways of avoiding SMT-based attacks.
                            Equivalent to: (default behavior)

                    auto,nosmt
                            Mitigate all CPU vulnerabilities, disabling SMT
                            if needed.  This is for users who always want to
                            be fully mitigated, even if it means losing SMT.
                            Equivalent to: l1tf=flush,nosmt [X86]
                                           mds=full,nosmt [X86]
                                           tsx_async_abort=full,nosmt [X86]
                                           mmio_stale_data=full,nosmt [X86]
                                           retbleed=auto,nosmt [X86]

                    [X86] After one of the above options, additionally
                    supports attack-vector based controls as documented in
                    Documentation/admin-guide/hw-vuln/attack_vector_controls.rs

    mminit_loglevel=
                    [KNL,EARLY] When CONFIG_DEBUG_MEMORY_INIT is set, this
                    parameter allows control of the logging verbosity for
                    the additional memory initialisation checks. A value
                    of 0 disables mminit logging and a level of 4 will
                    log everything. Information is printed at KERN_DEBUG
                    so loglevel=8 may also need to be specified.

    mmio_stale_data=
                    [X86,INTEL,EARLY] Control mitigation for the Processor
                    MMIO Stale Data vulnerabilities.

                    Processor MMIO Stale Data is a class of
                    vulnerabilities that may expose data after an MMIO
                    operation. Exposed data could originate or end in
                    the same CPU buffers as affected by MDS and TAA.
                    Therefore, similar to MDS and TAA, the mitigation
                    is to clear the affected CPU buffers.

                    This parameter controls the mitigation. The
                    options are:

                    full       - Enable mitigation on vulnerable CPUs

                    full,nosmt - Enable mitigation and disable SMT on
                                 vulnerable CPUs.

                    off        - Unconditionally disable mitigation
```

```
                On MDS or TAA affected machines,
                mmio_stale_data=off can be prevented by an active
                MDS or TAA mitigation as these vulnerabilities are
                mitigated with the same mechanism so in order to
                disable this mitigation, you need to specify
                mds=off and tsx_async_abort=off too.

                Not specifying this option is equivalent to
                mmio_stale_data=full.

                For details see:
                Documentation/admin-guide/hw-vuln/processor_mmio_stale_data

    <module>.async_probe[=<bool>] [KNL]
                If no <bool> value is specified or if the value
                specified is not a valid <bool>, enable asynchronous
                probe on this module.  Otherwise, enable/disable
                asynchronous probe on this module as indicated by the
                <bool> value. See also: module.async_probe

    module.async_probe=<bool>
                [KNL] When set to true, modules will use async probing
                by default. To enable/disable async probing for a
                specific module, use the module specific control that
                is documented under <module>.async_probe. When both
                module.async_probe and <module>.async_probe are
                specified, <module>.async_probe takes precedence for
                the specific module.

    module.enable_dups_trace
                [KNL] When CONFIG_MODULE_DEBUG_AUTOLOAD_DUPS is set,
                this means that duplicate request_module() calls will
                trigger a WARN_ON() instead of a pr_warn(). Note that
                if MODULE_DEBUG_AUTOLOAD_DUPS_TRACE is set, WARN_ON()s
                will always be issued and this option does nothing.
    module.sig_enforce
                [KNL] When CONFIG_MODULE_SIG is set, this means that
                modules without (valid) signatures will fail to load.
                Note that if CONFIG_MODULE_SIG_FORCE is set, that
                is always true, so this option does nothing.

    module_blacklist=  [KNL] Do not load a comma-separated list of
                modules.  Useful for debugging problem modules.

    mousedev.tap_time=
                [MOUSE] Maximum time between finger touching and
                leaving touchpad surface for touch to be considered
                a tap and be reported as a left button click (for
                touchpads working in absolute mode only).
                Format: <msecs>
    mousedev.xres=  [MOUSE] Horizontal screen resolution, used for devices
```

```
                reporting absolute coordinates, such as tablets
mousedev.yres=  [MOUSE] Vertical screen resolution, used for devices
                reporting absolute coordinates, such as tablets

movablecore=    [KNL,X86,PPC,EARLY]
                Format: nn[KMGTPE] | nn%
                This parameter is the complement to kernelcore=, it
                specifies the amount of memory used for migratable
                allocations.  If both kernelcore and movablecore is
                specified, then kernelcore will be at *least* the
                specified value but may be more.  If movablecore on its
                own is specified, the administrator must be careful
                that the amount of memory usable for all allocations
                is not too small.

movable_node    [KNL,EARLY] Boot-time switch to make hotplugable memory
                NUMA nodes to be movable. This means that the memory
                of such nodes will be usable only for movable
                allocations which rules out almost all kernel
                allocations. Use with caution!

MTD_Partition=  [MTD]
                Format: <name>,<region-number>,<size>,<offset>

MTD_Region=     [MTD] Format:
                <name>,<region-number>[,<base>,<size>,<buswidth>,<altbuswid

mtdparts=       [MTD]
                See drivers/mtd/parsers/cmdlinepart.c

mtouchusb.raw_coordinates=
                [HW] Make the MicroTouch USB driver use raw coordinates
                ('y', default) or cooked coordinates ('n')

mtrr=debug      [X86,EARLY]
                Enable printing debug information related to MTRR
                registers at boot time.

mtrr_chunk_size=nn[KMG,X86,EARLY]
                used for mtrr cleanup. It is largest continuous chunk
                that could hold holes aka. UC entries.

mtrr_gran_size=nn[KMG,X86,EARLY]
                Used for mtrr cleanup. It is granularity of mtrr block.
                Default is 1.
                Large value could prevent small alignment from
                using up MTRRs.

mtrr_spare_reg_nr=n [X86,EARLY]
                Format: <integer>
                Range: 0,7 : spare reg number
                Default : 1
```

```
                Used for mtrr cleanup. It is spare mtrr entries number.
                Set to 2 or more if your graphical card needs more.

    multitce=off    [PPC]  This parameter disables the use of the pSeries
                firmware feature for updating multiple TCE entries
                at a time.

    n2=             [NET] SDL Inc. RISCom/N2 synchronous serial card

    netdev=         [NET] NE2000 ISA network devices parameters
                Format: <irq>,<io>,<mem_start>,<mem_end>,<name>

    netpoll.carrier_timeout=
                [NET] Specifies amount of time (in seconds) that
                netpoll should wait for a carrier. By default netpoll
                waits 4 seconds.

    nf_conntrack.acct=
                [NETFILTER] Enable connection tracking flow accounting
                0 to disable accounting
                1 to enable accounting
                Default value is 0.

    nfs.cache_getent=
                [NFS] sets the pathname to the program which is used
                to update the NFS client cache entries.

    nfs.cache_getent_timeout=
                [NFS] sets the timeout after which an attempt to
                update a cache entry is deemed to have failed.

    nfs.callback_nr_threads=
                [NFSv4] set the total number of threads that the
                NFS client will assign to service NFSv4 callback
                requests.

    nfs.callback_tcpport=
                [NFS] set the TCP port on which the NFSv4 callback
                channel should listen.

    nfs.delay_retrans=
                [NFS] specifies the number of times the NFSv4 client
                retries the request before returning an EAGAIN error,
                after a reply of NFS4ERR_DELAY from the server.
                Only applies if the softerr mount option is enabled,
                and the specified value is >= 0.

    nfs.idmap_cache_timeout=
                [NFS] set the maximum lifetime for idmapper cache
                entries.

    nfs.max_session_cb_slots=
```

```
                [NFSv4.1] Sets the maximum number of session
                slots the client will assign to the callback
                channel. This determines the maximum number of
                callbacks the client will process in parallel for
                a particular server.

    nfs.max_session_slots=
                [NFSv4.1] Sets the maximum number of session slots
                the client will attempt to negotiate with the server.
                This limits the number of simultaneous RPC requests
                that the client can send to the NFSv4.1 server.
                Note that there is little point in setting this
                value higher than the max_tcp_slot_table_limit.

    nfs.nfs4_disable_idmapping=
                [NFSv4] When set to the default of '1', this option
                ensures that both the RPC level authentication
                scheme and the NFS level operations agree to use
                numeric uids/gids if the mount is using the
                'sec=sys' security flavour. In effect it is
                disabling idmapping, which can make migration from
                legacy NFSv2/v3 systems to NFSv4 easier.
                Servers that do not support this mode of operation
                will be autodetected by the client, and it will fall
                back to using the idmapper.
                To turn off this behaviour, set the value to '0'.

    nfs.nfs4_unique_id=
                [NFS4] Specify an additional fixed unique ident-
                ification string that NFSv4 clients can insert into
                their nfs_client_id4 string.  This is typically a
                UUID that is generated at system install time.

    nfs.recover_lost_locks=
                [NFSv4] Attempt to recover locks that were lost due
                to a lease timeout on the server. Please note that
                doing this risks data corruption, since there are
                no guarantees that the file will remain unchanged
                after the locks are lost.
                If you want to enable the kernel legacy behaviour of
                attempting to recover these locks, then set this
                parameter to '1'.
                The default parameter value of '0' causes the kernel
                not to attempt recovery of lost locks.

    nfs.send_implementation_id=
                [NFSv4.1] Send client implementation identification
                information in exchange_id requests.
                If zero, no implementation identification information
                will be sent.
                The default is to send the implementation identification
                information.
```

```
nfs4.layoutstats_timer=
                [NFSv4.2] Change the rate at which the kernel sends
                layoutstats to the pNFS metadata server.

                Setting this to value to 0 causes the kernel to use
                whatever value is the default set by the layout
                driver. A non-zero value sets the minimum interval
                in seconds between layoutstats transmissions.

nfsd.inter_copy_offload_enable=
                [NFSv4.2] When set to 1, the server will support
                server-to-server copies for which this server is
                the destination of the copy.

nfsd.nfs4_disable_idmapping=
                [NFSv4] When set to the default of '1', the NFSv4
                server will return only numeric uids and gids to
                clients using auth_sys, and will accept numeric uids
                and gids from such clients.  This is intended to ease
                migration from NFSv2/v3.

nfsd.nfsd4_ssc_umount_timeout=
                [NFSv4.2] When used as the destination of a
                server-to-server copy, knfsd temporarily mounts
                the source server.  It caches the mount in case
                it will be needed again, and discards it if not
                used for the number of milliseconds specified by
                this parameter.

nfsaddrs=       [NFS] Deprecated.  Use ip= instead.
                See Documentation/admin-guide/nfs/nfsroot.rst.

nfsroot=        [NFS] nfs root filesystem for disk-less boxes.
                See Documentation/admin-guide/nfs/nfsroot.rst.

nfsrootdebug    [NFS] enable nfsroot debugging messages.
                See Documentation/admin-guide/nfs/nfsroot.rst.

nmi_backtrace.backtrace_idle [KNL]
                Dump stacks even of idle CPUs in response to an
                NMI stack-backtrace request.

nmi_debug=      [KNL,SH] Specify one or more actions to take
                when a NMI is triggered.
                Format: [state][,regs][,debounce][,die]

nmi_watchdog=   [KNL,BUGS=X86] Debugging features for SMP kernels
                Format: [panic,][nopanic,][rNNN,][num]
                Valid num: 0 or 1
                0 - turn hardlockup detector in nmi_watchdog off
                1 - turn hardlockup detector in nmi_watchdog on
```

```
                rNNN - configure the watchdog with raw perf event 0xNNN

                When panic is specified, panic when an NMI watchdog
                timeout occurs (or 'nopanic' to not panic on an NMI
                watchdog, if CONFIG_BOOTPARAM_HARDLOCKUP_PANIC is set)
                To disable both hard and soft lockup detectors,
                please see 'nowatchdog'.
                This is useful when you use a panic=... timeout and
                need the box quickly up again.

                These settings can be accessed at runtime via
                the nmi_watchdog and hardlockup_panic sysctls.

    no4lvl      [RISCV,EARLY] Disable 4-level and 5-level paging modes.
                Forces kernel to use 3-level paging instead.

    no5lvl      [X86-64,RISCV,EARLY] Disable 5-level paging mode. Forces
                kernel to use 4-level paging instead.

    noalign     [KNL,ARM]

    noapic      [SMP,APIC,EARLY] Tells the kernel to not make use of any
                IOAPICs that may be present in the system.

    noapictimer [APIC,X86] Don't set up the APIC timer

    noautogroup Disable scheduler automatic task group creation.

    nocache     [ARM,EARLY]

    no_console_suspend
                [HW] Never suspend the console
                Disable suspending of consoles during suspend and
                hibernate operations.  Once disabled, debugging
                messages can reach various consoles while the rest
                of the system is being put to sleep (ie, while
                debugging driver suspend/resume hooks).  This may
                not work reliably with all consoles, but is known
                to work with serial and VGA consoles.
                To facilitate more flexible debugging, we also add
                console_suspend, a printk module parameter to control
                it. Users could use console_suspend (usually
                /sys/module/printk/parameters/console_suspend) to
                turn on/off it dynamically.

    no_debug_objects
                [KNL,EARLY] Disable object debugging

    nodsp       [SH] Disable hardware DSP at boot time.

    noefi       [EFI,EARLY] Disable EFI runtime services support.
```

```
	no_entry_flush  [PPC,EARLY] Don't flush the L1-D cache when entering the ke

	noexec32        [X86-64]
	                This affects only 32-bit executables.
	                noexec32=on: enable non-executable mappings (default)
	                        read doesn't imply executable mappings
	                noexec32=off: disable non-executable mappings
	                        read implies executable mappings

	no_file_caps    Tells the kernel not to honor file capabilities.  The
	                only way then for a file to be executed with privilege
	                is to be setuid root or executed by root.

	nofpu           [MIPS,SH] Disable hardware FPU at boot time.

	nofsgsbase      [X86] Disables FSGSBASE instructions.

	nofxsr          [BUGS=X86-32] Disables x86 floating point extended
	                register save and restore. The kernel will only save
	                legacy floating-point registers on task switch.

	nogbpages       [X86] Do not use GB pages for kernel direct mappings.

	no_hash_pointers
	                [KNL,EARLY]
	                Alias for "hash_pointers=never".

	nohibernate     [HIBERNATION] Disable hibernation and resume.

	nohlt           [ARM,ARM64,MICROBLAZE,MIPS,PPC,RISCV,SH] Forces the kernel
	                busy wait in do_idle() and not use the arch_cpu_idle()
	                implementation; requires CONFIG_GENERIC_IDLE_POLL_SETUP
	                to be effective. This is useful on platforms where the
	                sleep(SH) or wfi(ARM,ARM64) instructions do not work
	                correctly or when doing power measurements to evaluate
	                the impact of the sleep instructions. This is also
	                useful when using JTAG debugger.

	nohpet          [X86] Don't use the HPET timer.

	nohugeiomap     [KNL,X86,PPC,ARM64,EARLY] Disable kernel huge I/O mappings.

	nohugevmalloc   [KNL,X86,PPC,ARM64,EARLY] Disable kernel huge vmalloc mappi

	nohz=           [KNL] Boottime enable/disable dynamic ticks
	                Valid arguments: on, off
	                Default: on

	nohz_full=      [KNL,BOOT,SMP,ISOL]
	                The argument is a cpu list, as described above.
	                In kernels built with CONFIG_NO_HZ_FULL=y, set
	                the specified list of CPUs whose tick will be stopped
```

```
                whenever possible. The boot CPU will be forced outside
                the range to maintain the timekeeping.  Any CPUs
                in this list will have their RCU callbacks offloaded,
                just as if they had also been called out in the
                rcu_nocbs= boot parameter.

                Note that this argument takes precedence over
                the CONFIG_RCU_NOCB_CPU_DEFAULT_ALL option.

    noinitrd    [Deprecated,RAM] Tells the kernel not to load any configure
                initial RAM disk. Currently this parameter applies to
                initrd only, not to initramfs. But it applies to both
                in EFI mode.

    nointremap  [X86-64,Intel-IOMMU,EARLY] Do not enable interrupt
                remapping.
                [Deprecated - use intremap=off]

    noinvpcid   [X86,EARLY] Disable the INVPCID cpu feature.

    noiotrap    [SH] Disables trapped I/O port accesses.

    noirqdebug  [X86-32] Disables the code which attempts to detect and
                disable unhandled interrupt sources.

    noisapnp    [ISAPNP] Disables ISA PnP code.

    nokaslr     [KNL,EARLY]
                When CONFIG_RANDOMIZE_BASE is set, this disables
                kernel and module base offset ASLR (Address Space
                Layout Randomization).

    no-kvmapf   [X86,KVM,EARLY] Disable paravirtualized asynchronous page
                fault handling.

    no-kvmclock [X86,KVM,EARLY] Disable paravirtualized KVM clock driver

    nolapic     [X86-32,APIC,EARLY] Do not enable or use the local APIC.

    nolapic_timer [X86-32,APIC,EARLY] Do not use the local APIC timer.

    nomce       [X86-32] Disable Machine Check Exception

    nomfgpt     [X86-32] Disable Multi-Function General Purpose
                Timer usage (for AMD Geode machines).

    nomodeset   Disable kernel modesetting. Most systems' firmware
                sets up a display mode and provides framebuffer memory
                for output. With nomodeset, DRM and fbdev drivers will
                not load if they could possibly displace the pre-
                initialized output. Only the system framebuffer will
                be available for use. The respective drivers will not
```

```
                perform display-mode changes or accelerated rendering.

                Useful as error fallback, or for testing and debugging.

    nomodule    Disable module load

    nonmi_ipi   [X86] Disable using NMI IPIs during panic/reboot to
                shutdown the other cpus.  Instead use the REBOOT_VECTOR
                irq.

    nopat       [X86,EARLY] Disable PAT (page attribute table extension of
                pagetables) support.

    nopcid      [X86-64,EARLY] Disable the PCID cpu feature.

    nopku       [X86] Disable Memory Protection Keys CPU feature found
                in some Intel CPUs.

    nopti       [X86-64,EARLY]
                Equivalent to pti=off

    nopv=       [X86,XEN,KVM,HYPER_V,VMWARE,EARLY]
                Disables the PV optimizations forcing the guest to run
                as generic guest with no PV drivers. Currently support
                XEN HVM, KVM, HYPER_V and VMWARE guest.

    nopvspin    [X86,XEN,KVM,EARLY]
                Disables the qspinlock slow path using PV optimizations
                which allow the hypervisor to 'idle' the guest on lock
                contention.

    norandmaps  Don't use address space randomization.  Equivalent to
                echo 0 > /proc/sys/kernel/randomize_va_space

    noreplace-smp   [X86-32,SMP] Don't replace SMP instructions
                with UP alternatives

    noresume    [SWSUSP] Disables resume and restores original swap
                space.

    no-scroll   [VGA] Disables scrollback.
                This is required for the Braillex ib80-piezo Braille
                reader made by F.H. Papenmeier (Germany).

    nosgx       [X86-64,SGX,EARLY] Disables Intel SGX kernel support.

    nosmap      [PPC,EARLY]
                Disable SMAP (Supervisor Mode Access Prevention)
                even if it is supported by processor.

    nosmep      [PPC64s,EARLY]
                Disable SMEP (Supervisor Mode Execution Prevention)
```

```
                even if it is supported by processor.

    nosmp       [SMP,EARLY] Tells an SMP kernel to act as a UP kernel,
                and disable the IO APIC.  legacy for "maxcpus=0".

    nosmt       [KNL,MIPS,PPC,EARLY] Disable symmetric multithreading (SMT)
                Equivalent to smt=1.

                [KNL,LOONGARCH,X86,ARM64,PPC,S390] Disable symmetric multit
                nosmt=force: Force disable SMT, cannot be undone
                             via the sysfs control file.

    nosoftlockup    [KNL] Disable the soft-lockup detector.

    nospec_store_bypass_disable
                [HW,EARLY] Disable all mitigations for the Speculative
                Store Bypass vulnerability

    nospectre_bhb   [ARM64,EARLY] Disable all mitigations for Spectre-BHB (bran
                history injection) vulnerability. System may allow data lea
                with this option.

    nospectre_v1    [X86,PPC,EARLY] Disable mitigations for Spectre Variant 1
                (bounds check bypass). With this option data leaks are
                possible in the system.

    nospectre_v2    [X86,PPC_E500,ARM64,EARLY] Disable all mitigations
                for the Spectre variant 2 (indirect branch
                prediction) vulnerability. System may allow data
                leaks with this option.

    no-steal-acc    [X86,PV_OPS,ARM64,PPC/PSERIES,RISCV,LOONGARCH,EARLY]
                Disable paravirtualized steal time accounting. steal time
                is computed, but won't influence scheduler behaviour

    nosync      [HW,M68K] Disables sync negotiation for all devices.

    no_timer_check  [X86,APIC] Disables the code which tests for broken
                timer IRQ sources, i.e., the IO-APIC timer. This can
                work around problems with incorrect timer
                initialization on some boards.

    no_uaccess_flush
                [PPC,EARLY] Don't flush the L1-D cache after accessing user

    novmcoredd      [KNL,KDUMP]
                Disable device dump. Device dump allows drivers to
                append dump data to vmcore so you can collect driver
                specified debug info.  Drivers can append the data
                without any limit and this data is stored in memory,
                so this may cause significant memory stress.  Disabling
                device dump can help save memory but the driver debug
```

```
                data will be no longer available.  This parameter
                is only available when CONFIG_PROC_VMCORE_DEVICE_DUMP
                is set.

    no-vmw-sched-clock
                [X86,PV_OPS,EARLY] Disable paravirtualized VMware
                scheduler clock and use the default one.

    nowatchdog  [KNL] Disable both lockup detectors, i.e.
                soft-lockup and NMI watchdog (hard-lockup).

    nowb        [ARM,EARLY]

    nox2apic    [X86-64,APIC,EARLY] Do not enable x2APIC mode.

                NOTE: this parameter will be ignored on systems with the
                LEGACY_XAPIC_DISABLED bit set in the
                IA32_XAPIC_DISABLE_STATUS MSR.

    noxsave     [BUGS=X86] Disables x86 extended register state save
                and restore using xsave. The kernel will fallback to
                enabling legacy floating-point and sse state.

    noxsaveopt  [X86] Disables xsaveopt used in saving x86 extended
                register states. The kernel will fall back to use
                xsave to save the states. By using this parameter,
                performance of saving the states is degraded because
                xsave doesn't support modified optimization while
                xsaveopt supports it on xsaveopt enabled systems.

    noxsaves    [X86] Disables xsaves and xrstors used in saving and
                restoring x86 extended register state in compacted
                form of xsave area. The kernel will fall back to use
                xsaveopt and xrstor to save and restore the states
                in standard form of xsave area. By using this
                parameter, xsave area per process might occupy more
                memory on xsaves enabled systems.

    nr_cpus=    [SMP,EARLY] Maximum number of processors that an SMP kernel
                could support.  nr_cpus=n : n >= 1 limits the kernel to
                support 'n' processors. It could be larger than the
                number of already plugged CPU during bootup, later in
                runtime you can physically add extra cpu until it reaches
                n. So during boot up some boot time memory for per-cpu
                variables need be pre-allocated for later physical cpu
                hot plugging.

    nr_uarts=   [SERIAL] maximum number of UARTs to be registered.

    numa=off    [KNL, ARM64, PPC, RISCV, SPARC, X86, EARLY]
                Disable NUMA, Only set up a single NUMA node
                spanning all memory.
```

```
numa=fake=<size>[MG]
                [KNL, ARM64, RISCV, X86, EARLY]
                If given as a memory unit, fills all system RAM with
                nodes of size interleaved over physical nodes.

numa=fake=<N>
                [KNL, ARM64, RISCV, X86, EARLY]
                If given as an integer, fills all system RAM with N
                fake nodes interleaved over physical nodes.

numa=fake=<N>U
                [KNL, ARM64, RISCV, X86, EARLY]
                If given as an integer followed by 'U', it will
                divide each physical node into N emulated nodes.

numa=noacpi     [X86] Don't parse the SRAT table for NUMA setup

numa=nohmat     [X86] Don't parse the HMAT table for NUMA setup, or
                soft-reserved memory partitioning.

numa_balancing= [KNL,ARM64,PPC,RISCV,S390,X86] Enable or disable automatic
                NUMA balancing.
                Allowed values are enable and disable

numa_zonelist_order= [KNL, BOOT] Select zonelist order for NUMA.
                'node', 'default' can be specified
                This can be set from sysctl after boot.
                See Documentation/admin-guide/sysctl/vm.rst for details.

nvme.quirks=    [NVME] A list of quirk entries to augment the built-in
                nvme quirk list. List entries are separated by a
                '-' character.
                Each entry has the form VendorID:ProductID:quirk_names.
                The IDs are 4-digits hex numbers and quirk_names is a
                list of quirk names separated by commas. A quirk name
                can be prefixed by '^', meaning that the specified
                quirk must be disabled.

                Example:
                nvme.quirks=7710:2267:bogus_nid,^identify_cns-9900:7711:bro

ohci1394_dma=early      [HW,EARLY] enable debugging via the ohci1394 driver
                See Documentation/core-api/debugging-via-ohci1394.rst for m
                info.

olpc_ec_timeout= [OLPC] ms delay when issuing EC commands
                Rather than timing out after 20 ms if an EC
                command is not properly ACKed, override the length
                of the timeout.  We have interrupts disabled while
                waiting for the ACK, so if this is set too high
                interrupts *may* be lost!
```

```
omap_mux=       [OMAP] Override bootloader pin multiplexing.
                Format: <mux_mode0.mode_name=value>...
                For example, to override I2C bus2:
                omap_mux=i2c2_scl.i2c2_scl=0x100,i2c2_sda.i2c2_sda=0x100

onenand.bdry=   [HW,MTD] Flex-OneNAND Boundary Configuration

                Format: [die0_boundary][,die0_lock][,die1_boundary][,die1_l

                boundary - index of last SLC block on Flex-OneNAND.
                           The remaining blocks are configured as MLC block
                lock     - Configure if Flex-OneNAND boundary should be loc
                           Once locked, the boundary cannot be changed.
                           1 indicates lock status, 0 indicates unlock stat

oops=panic      [KNL,EARLY]
                Always panic on oopses. Default is to just kill the
                process, but there is a small probability of
                deadlocking the machine.
                This will also cause panics on machine check exceptions.
                Useful together with panic=30 to trigger a reboot.

page_alloc.shuffle=
                [KNL] Boolean flag to control whether the page allocator
                should randomize its free lists. This parameter can be
                used to enable/disable page randomization. The state of
                the flag can be read from sysfs at:
                /sys/module/page_alloc/parameters/shuffle.
                This parameter is only available if CONFIG_SHUFFLE_PAGE_ALL

page_owner=     [KNL,EARLY] Boot-time page_owner enabling option.
                Storage of the information about who allocated
                each page is disabled in default. With this switch,
                we can turn it on.
                on: enable the feature

page_poison=    [KNL,EARLY] Boot-time parameter changing the state of
                poisoning on the buddy allocator, available with
                CONFIG_PAGE_POISONING=y.
                off: turn off poisoning (default)
                on: turn on poisoning

page_reporting.page_reporting_order=
                [KNL] Minimal page reporting order
                Format: <integer>
                Adjust the minimal page reporting order. The page
                reporting is disabled when it exceeds MAX_PAGE_ORDER.

panic=          [KNL] Kernel behaviour on panic: delay <timeout>
                timeout > 0: seconds before rebooting
                timeout = 0: wait forever
```

```
                timeout < 0: reboot immediately
                Format: <timeout>

    panic_on_taint= [KNL,EARLY]
                Bitmask for conditionally calling panic() in add_taint()
                Format: <hex>[,nousertaint]
                Hexadecimal bitmask representing the set of TAINT flags
                that will cause the kernel to panic when add_taint() is
                called with any of the flags in this set.
                The optional switch "nousertaint" can be utilized to
                prevent userspace forced crashes by writing to sysctl
                /proc/sys/kernel/tainted any flagset matching with the
                bitmask set on panic_on_taint.
                See Documentation/admin-guide/tainted-kernels.rst for
                extra details on the taint flags that users can pick
                to compose the bitmask to assign to panic_on_taint.

    panic_on_warn=1 panic() instead of WARN().  Useful to cause kdump
                on a WARN().

    panic_force_cpu=
                [KNL,SMP] Force panic handling to execute on a specific CPU
                Format: <cpu number>
                Some platforms require panic handling to occur on a
                specific CPU for the crash kernel to function correctly.
                This can be due to firmware limitations, interrupt routing
                constraints, or platform-specific requirements where only
                a particular CPU can safely enter the crash kernel.
                When set, panic() will redirect execution to the specified
                CPU before proceeding with the normal panic and kexec flow.
                If the target CPU is offline or unavailable, panic proceeds
                on the current CPU.
                This option should only be used for systems with the above
                constraints as it might cause the panic operation to be les

    panic_print=    Bitmask for printing system info when panic happens.
                User can chose combination of the following bits:
                bit 0: print all tasks info
                bit 1: print system memory info
                bit 2: print timer info
                bit 3: print locks info if CONFIG_LOCKDEP is on
                bit 4: print ftrace buffer
                bit 5: replay all kernel messages on consoles at the end of
                bit 6: print all CPUs backtrace (if available in the arch)
                bit 7: print only tasks in uninterruptible (blocked) state
                *Be aware* that this option may print a _lot_ of lines,
                so there are risks of losing older messages in the log.
                Use this option carefully, maybe worth to setup a
                bigger log buffer with "log_buf_len" along with this.

    panic_sys_info= A comma separated list of extra information to be dumped
                on panic.
```

```
                Format: val[,val...]
                Where @val can be any of the following:

                tasks:          print all tasks info
                mem:            print system memory info
                timers:         print timers info
                locks:          print locks info if CONFIG_LOCKDEP is on
                ftrace:         print ftrace buffer
                all_bt:         print all CPUs backtrace (if available in t
                blocked_tasks:  print only tasks in uninterruptible (blocke

                This is a human readable alternative to the 'panic_print' o

    panic_console_replay
                When panic happens, replay all kernel messages on
                consoles at the end of panic.

    parkbd.port=    [HW] Parallel port number the keyboard adapter is
                connected to, default is 0.
                Format: <parport#>
    parkbd.mode=    [HW] Parallel port keyboard adapter mode of operation,
                0 for XT, 1 for AT (default is AT).
                Format: <mode>

    parport=        [HW,PPT] Specify parallel ports. 0 disables.
                Format: { 0 | auto | 0xBBB[,IRQ[,DMA]] }
                Use 'auto' to force the driver to use any
                IRQ/DMA settings detected (the default is to
                ignore detected IRQ/DMA settings because of
                possible conflicts). You can specify the base
                address, IRQ, and DMA settings; IRQ and DMA
                should be numbers, or 'auto' (for using detected
                settings on that particular port), or 'nofifo'
                (to avoid using a FIFO even if it is detected).
                Parallel ports are assigned in the order they
                are specified on the command line, starting
                with parport0.

    parport_init_mode=      [HW,PPT]
                Configure VIA parallel port to operate in
                a specific mode. This is necessary on Pegasos
                computer where firmware has no options for setting
                up parallel port mode and sets it to spp.
                Currently this function knows 686a and 8231 chips.
                Format: [spp|ps2|epp|ecp|ecpepp]

    pata_legacy.all=        [HW,LIBATA]
                Format: <int>
                Set to non-zero to probe primary and secondary ISA
                port ranges on PCI systems where no PCI PATA device
                has been found at either range.  Disabled by default.
```

```
	pata_legacy.autospeed=  [HW,LIBATA]
				Format: <int>
				Set to non-zero if a chip is present that snoops speed
				changes.  Disabled by default.

	pata_legacy.ht6560a=    [HW,LIBATA]
				Format: <int>
				Set to 1, 2, or 3 for HT 6560A on the primary channel,
				the secondary channel, or both channels respectively.
				Disabled by default.

	pata_legacy.ht6560b=    [HW,LIBATA]
				Format: <int>
				Set to 1, 2, or 3 for HT 6560B on the primary channel,
				the secondary channel, or both channels respectively.
				Disabled by default.

	pata_legacy.iordy_mask= [HW,LIBATA]
				Format: <int>
				IORDY enable mask.  Set individual bits to allow IORDY
				for the respective channel.  Bit 0 is for the first
				legacy channel handled by this driver, bit 1 is for
				the second channel, and so on.  The sequence will often
				correspond to the primary legacy channel, the secondary
				legacy channel, and so on, but the handling of a PCI
				bus and the use of other driver options may interfere
				with the sequence.  By default IORDY is allowed across
				all channels.

	pata_legacy.opti82c46x= [HW,LIBATA]
				Format: <int>
				Set to 1, 2, or 3 for Opti 82c611A on the primary
				channel, the secondary channel, or both channels
				respectively.  Disabled by default.

	pata_legacy.opti82c611a=        [HW,LIBATA]
				Format: <int>
				Set to 1, 2, or 3 for Opti 82c465MV on the primary
				channel, the secondary channel, or both channels
				respectively.  Disabled by default.

	pata_legacy.pio_mask=   [HW,LIBATA]
				Format: <int>
				PIO mode mask for autospeed devices.  Set individual
				bits to allow the use of the respective PIO modes.
				Bit 0 is for mode 0, bit 1 is for mode 1, and so on.
				All modes allowed by default.

	pata_legacy.probe_all=  [HW,LIBATA]
				Format: <int>
				Set to non-zero to probe tertiary and further ISA
				port ranges on PCI systems.  Disabled by default.
```

```
pata_legacy.probe_mask= [HW,LIBATA]
                Format: <int>
                Probe mask for legacy ISA PATA ports.  Depending on
                platform configuration and the use of other driver
                options up to 6 legacy ports are supported: 0x1f0,
                0x170, 0x1e8, 0x168, 0x1e0, 0x160, however probing
                of individual ports can be disabled by setting the
                corresponding bits in the mask to 1.  Bit 0 is for
                the first port in the list above (0x1f0), and so on.
                By default all supported ports are probed.

pata_legacy.qdi=        [HW,LIBATA]
                Format: <int>
                Set to non-zero to probe QDI controllers.  By default
                set to 1 if CONFIG_PATA_QDI_MODULE, 0 otherwise.

pata_legacy.winbond=    [HW,LIBATA]
                Format: <int>
                Set to non-zero to probe Winbond controllers.  Use
                the standard I/O port (0x130) if 1, otherwise the
                value given is the I/O port to use (typically 0x1b0).
                By default set to 1 if CONFIG_PATA_WINBOND_VLB_MODULE,
                0 otherwise.

pata_platform.pio_mask= [HW,LIBATA]
                Format: <int>
                Supported PIO mode mask.  Set individual bits to allow
                the use of the respective PIO modes.  Bit 0 is for
                mode 0, bit 1 is for mode 1, and so on.  Mode 0 only
                allowed by default.

pause_on_oops=<int>
                Halt all CPUs after the first oops has been printed for
                the specified number of seconds.  This is to be used if
                your oopses keep scrolling off the screen.

pci=option[,option...]  [PCI,EARLY] various PCI subsystem options.

                        Some options herein operate on a specific device
                        or a set of devices (<pci_dev>). These are
                        specified in one of the following formats:

                        [<domain>:]<bus>:<dev>.<func>[/<dev>.<func>]*
                        pci:<vendor>:<device>[:<subvendor>:<subdevice>]

                        Note: the first format specifies a PCI
                        bus/device/function address which may change
                        if new hardware is inserted, if motherboard
                        firmware changes, or due to changes caused
                        by other kernel parameters. If the
                        domain is left unspecified, it is
```

```
                    taken to be zero. Optionally, a path
                    to a device through multiple device/function
                    addresses can be specified after the base
                    address (this is more robust against
                    renumbering issues).  The second format
                    selects devices using IDs from the
                    configuration space which may match multiple
                    devices in the system.

        earlydump   dump PCI config space before the kernel
                    changes anything
        off         [X86] don't probe for the PCI bus
        bios        [X86-32] force use of PCI BIOS, don't access
                    the hardware directly. Use this if your machine
                    has a non-standard PCI host bridge.
        nobios      [X86-32] disallow use of PCI BIOS, only direct
                    hardware access methods are allowed. Use this
                    if you experience crashes upon bootup and you
                    suspect they are caused by the BIOS.
        conf1       [X86] Force use of PCI Configuration Access
                    Mechanism 1 (config address in IO port 0xCF8,
                    data in IO port 0xCFC, both 32-bit).
        conf2       [X86] Force use of PCI Configuration Access
                    Mechanism 2 (IO port 0xCF8 is an 8-bit port for
                    the function, IO port 0xCFA, also 8-bit, sets
                    bus number. The config space is then accessed
                    through ports 0xC000-0xCFFF).
                    See http://wiki.osdev.org/PCI for more info
                    on the configuration access mechanisms.
        noaer       [PCIE] If the PCIEAER kernel config parameter is
                    enabled, this kernel boot option can be used to
                    disable the use of PCIE advanced error reporting.
        nodomains   [PCI] Disable support for multiple PCI
                    root domains (aka PCI segments, in ACPI-speak).
        nommconf    [X86] Disable use of MMCONFIG for PCI
                    Configuration
        check_enable_amd_mmconf [X86] check for and enable
                    properly configured MMIO access to PCI
                    config space on AMD family 10h CPU
        nomsi       [MSI] If the PCI_MSI kernel config parameter is
                    enabled, this kernel boot option can be used to
                    disable the use of MSI interrupts system-wide.
        noioapicquirk [APIC] Disable all boot interrupt quirks.
                    Safety option to keep boot IRQs enabled. This
                    should never be necessary.
        ioapicreroute [APIC] Enable rerouting of boot IRQs to the
                    primary IO-APIC for bridges that cannot disable
                    boot IRQs. This fixes a source of spurious IRQs
                    when the system masks IRQs.
        noioapicreroute [APIC] Disable workaround that uses the
                    boot IRQ equivalent of an IRQ that connects to
                    a chipset where boot IRQs cannot be disabled.
```

```
                The opposite of ioapicreroute.
        biosirq         [X86-32] Use PCI BIOS calls to get the interrupt
                routing table. These calls are known to be buggy
                on several machines and they hang the machine
                when used, but on other computers it's the only
                way to get the interrupt routing table. Try
                this option if the kernel is unable to allocate
                IRQs or discover secondary PCI buses on your
                motherboard.
        rom             [X86] Assign address space to expansion ROMs.
                Use with caution as certain devices share
                address decoders between ROMs and other
                resources.
        norom           [X86] Do not assign address space to
                expansion ROMs that do not already have
                BIOS assigned address ranges.
        nobar           [X86] Do not assign address space to the
                BARs that weren't assigned by the BIOS.
        irqmask=0xMMMM  [X86] Set a bit mask of IRQs allowed to be
                assigned automatically to PCI devices. You can
                make the kernel exclude IRQs of your ISA cards
                this way.
        pirqaddr=0xAAAAA        [X86] Specify the physical address
                of the PIRQ table (normally generated
                by the BIOS) if it is outside the
                F0000h-100000h range.
        lastbus=N       [X86] Scan all buses thru bus #N. Can be
                useful if the kernel is unable to find your
                secondary buses and you want to tell it
                explicitly which ones they are.
        assign-busses   [X86] Always assign all PCI bus
                numbers ourselves, overriding
                whatever the firmware may have done.
        usepirqmask     [X86] Honor the possible IRQ mask stored
                in the BIOS $PIR table. This is needed on
                some systems with broken BIOSes, notably
                some HP Pavilion N5400 and Omnibook XE3
                notebooks. This will have no effect if ACPI
                IRQ routing is enabled.
        noacpi          [X86] Do not use ACPI for IRQ routing
                or for PCI scanning.
        use_crs         [X86] Use PCI host bridge window information
                from ACPI.  On BIOSes from 2008 or later, this
                is enabled by default.  If you need to use this,
                please report a bug.
        nocrs           [X86] Ignore PCI host bridge windows from ACPI.
                If you need to use this, please report a bug.
        use_e820        [X86] Use E820 reservations to exclude parts of
                PCI host bridge windows. This is a workaround
                for BIOS defects in host bridge _CRS methods.
                If you need to use this, please report a bug to
                <linux-pci@vger.kernel.org>.
```

```
                no_e820         [X86] Ignore E820 reservations for PCI host
                                bridge windows. This is the default on modern
                                hardware. If you need to use this, please report
                                a bug to <linux-pci@vger.kernel.org>.
                routeirq        Do IRQ routing for all PCI devices.
                                This is normally done in pci_enable_device(),
                                so this option is a temporary workaround
                                for broken drivers that don't call it.
                skip_isa_align  [X86] do not align io start addr, so can
                                handle more pci cards
                noearly         [X86] Don't do any early type 1 scanning.
                                This might help on some broken boards which
                                machine check when some devices' config space
                                is read. But various workarounds are disabled
                                and some IOMMU drivers will not work.
                bfsort          Sort PCI devices into breadth-first order.
                                This sorting is done to get a device
                                order compatible with older (<= 2.4) kernels.
                nobfsort        Don't sort PCI devices into breadth-first order.
                pcie_bus_tune_off       Disable PCIe MPS (Max Payload Size)
                                tuning and use the BIOS-configured MPS defaults.
                pcie_bus_safe   Set every device's MPS to the largest value
                                supported by all devices below the root complex.
                pcie_bus_perf   Set device MPS to the largest allowable MPS
                                based on its parent bus. Also set MRRS (Max
                                Read Request Size) to the largest supported
                                value (no larger than the MPS that the device
                                or bus can support) for best performance.
                pcie_bus_peer2peer      Set every device's MPS to 128B, which
                                every device is guaranteed to support. This
                                configuration allows peer-to-peer DMA between
                                any pair of devices, possibly at the cost of
                                reduced performance.  This also guarantees
                                that hot-added devices will work.
                cbiosize=nn[KMG]        The fixed amount of bus space which is
                                reserved for the CardBus bridge's IO window.
                                The default value is 256 bytes.
                cbmemsize=nn[KMG]       The fixed amount of bus space which is
                                reserved for the CardBus bridge's memory
                                window. The default value is 64 megabytes.
                resource_alignment=
                                Format:
                                [<order of align>@]<pci_dev>[; ...]
                                Specifies alignment and device to reassign
                                aligned memory resources. How to
                                specify the device is described above.
                                If <order of align> is not specified,
                                PAGE_SIZE is used as alignment.
                                A PCI-PCI bridge can be specified if resource
                                windows need to be expanded.
                                To specify the alignment for several
                                instances of a device, the PCI vendor,
```

```
                device, subvendor, and subdevice may be
                specified, e.g., 12@pci:8086:9c22:103c:198f
                for 4096-byte alignment.
        ecrc=           Enable/disable PCIe ECRC (transaction layer
                end-to-end CRC checking). Only effective if
                OS has native AER control (either granted by
                ACPI _OSC or forced via "pcie_ports=native")
                bios: Use BIOS/firmware settings. This is the
                the default.
                off: Turn ECRC off
                on: Turn ECRC on.
        hpiosize=nn[KMG]        The fixed amount of bus space which is
                reserved for hotplug bridge's IO window.
                Default size is 256 bytes.
        hpmmiosize=nn[KMG]      The fixed amount of bus space which is
                reserved for hotplug bridge's MMIO window.
                Default size is 2 megabytes.
        hpmmioprefsize=nn[KMG]  The fixed amount of bus space which is
                reserved for hotplug bridge's MMIO_PREF window.
                Default size is 2 megabytes.
        hpmemsize=nn[KMG]       The fixed amount of bus space which is
                reserved for hotplug bridge's MMIO and
                MMIO_PREF window.
                Default size is 2 megabytes.
        hpbussize=nn    The minimum amount of additional bus numbers
                reserved for buses below a hotplug bridge.
                Default is 1.
        realloc=        Enable/disable reallocating PCI bridge resources
                if allocations done by BIOS are too small to
                accommodate resources required by all child
                devices.
                off: Turn realloc off
                on: Turn realloc on
        realloc         same as realloc=on
        noari           do not use PCIe ARI.
        noats           [PCIE, Intel-IOMMU, AMD-IOMMU]
                do not use PCIe ATS (and IOMMU device IOTLB).
        pcie_scan_all   Scan all possible PCIe devices.  Otherwise we
                only look for one device below a PCIe downstream
                port.
        big_root_window Try to add a big 64bit memory window to the PCIe
                root complex on AMD CPUs. Some GFX hardware
                can resize a BAR to allow access to all VRAM.
                Adding the window is slightly risky (it may
                conflict with unreported devices), so this
                taints the kernel.
        disable_acs_redir=<pci_dev>[; ...]
                Specify one or more PCI devices (in the format
                specified above) separated by semicolons.
                Each device specified will have the PCI ACS
                redirect capabilities forced off which will
                allow P2P traffic between devices through
```

```
                        bridges without forcing it upstream. Note:
                        this removes isolation between devices and
                        may put more devices in an IOMMU group.
            config_acs=
                        Format:
                        <ACS flags>@<pci_dev>[; ...]
                        Specify one or more PCI devices (in the format
                        specified above) optionally prepended with flags
                        and separated by semicolons. The respective
                        capabilities will be enabled, disabled or
                        unchanged based on what is specified in
                        flags.

                        ACS Flags is defined as follows:
                          bit-0 : ACS Source Validation
                          bit-1 : ACS Translation Blocking
                          bit-2 : ACS P2P Request Redirect
                          bit-3 : ACS P2P Completion Redirect
                          bit-4 : ACS Upstream Forwarding
                          bit-5 : ACS P2P Egress Control
                          bit-6 : ACS Direct Translated P2P
                        Each bit can be marked as:
                          '0' – force disabled
                          '1' – force enabled
                          'x' – unchanged
                        For example,
                          pci=config_acs=10x@pci:0:0
                        would configure all devices that support
                        ACS to enable P2P Request Redirect, disable
                        Translation Blocking, and leave Source
                        Validation unchanged from whatever power-up
                        or firmware set it to.

                        Note: this may remove isolation between devices
                        and may put more devices in an IOMMU group.
            force_floating  [S390] Force usage of floating interrupts.
            nomio           [S390] Do not use MIO instructions.
            norid           [S390] ignore the RID field and force use of
                            one PCI domain per PCI function
            notph           [PCIE] If the PCIE_TPH kernel config parameter
                            is enabled, this kernel boot option can be used
                            to disable PCIe TLP Processing Hints support
                            system-wide.

    pcie_aspm=      [PCIE] Forcibly enable or ignore PCIe Active State Power
                    Management.
            off     Don't touch ASPM configuration at all.  Leave any
                    configuration done by firmware unchanged.
            force   Enable ASPM even on devices that claim not to support it.
                    WARNING: Forcing ASPM on may cause system lockups.

    pcie_ports=     [PCIE] PCIe port services handling:
```

```
                native  Use native PCIe services (PME, AER, DPC, PCIe hotplug)
                        even if the platform doesn't give the OS permission to
                        use them.  This may cause conflicts if the platform
                        also tries to use these services.
                dpc-native      Use native PCIe service for DPC only.  May
                                cause conflicts if firmware uses AER or DPC.
                compat  Disable native PCIe services (PME, AER, DPC, PCIe
                        hotplug).

        pcie_port_pm=   [PCIE] PCIe port power management handling:
                off     Disable power management of all PCIe ports
                force   Forcibly enable power management of all PCIe ports

        pcie_pme=       [PCIE,PM] Native PCIe PME signaling options:
                nomsi   Do not use MSI for native PCIe PME signaling (this makes
                        all PCIe root ports use INTx for all services).

        pcmv=           [HW,PCMCIA] BadgePAD 4

        pd_ignore_unused
                        [PM]
                        Keep all power-domains already enabled by bootloader on,
                        even if no driver has claimed them. This is useful
                        for debug and development, but should not be
                        needed on a platform with proper driver support.

        pdcchassis=     [PARISC,HW] Disable/Enable PDC Chassis Status codes at
                        boot time.
                        Format: { 0 | 1 }
                        See arch/parisc/kernel/pdc_chassis.c

        percpu_alloc=   [MM,EARLY]
                        Select which percpu first chunk allocator to use.
                        Currently supported values are "embed" and "page".
                        Archs may support subset or none of the selections.
                        See comments in mm/percpu.c for details on each
                        allocator.  This parameter is primarily for debugging
                        and performance comparison.

        pirq=           [SMP,APIC] Manual mp-table setup
                        See Documentation/arch/x86/i386/IO-APIC.rst.

        plip=           [PPT,NET] Parallel port network link
                        Format: { parport<nr> | timid | 0 }
                        See also Documentation/admin-guide/parport.rst.

        pmtmr=          [X86] Manual setup of pmtmr I/O Port.
                        Override pmtimer IOPort with a hex value.
                        e.g. pmtmr=0x508

        pmu_override=   [PPC] Override the PMU.
                        This option takes over the PMU facility, so it is no
```

```
                longer usable by perf. Setting this option starts the
                PMU counters by setting MMCR0 to 0 (the FC bit is
                cleared). If a number is given, then MMCR1 is set to
                that number, otherwise (e.g., 'pmu_override=on'), MMCR1
                remains 0.

pm_async=       [PM]
                Format: off
                This parameter sets the initial value of the
                /sys/power/pm_async sysfs knob at boot time.
                If set to "off", disables asynchronous suspend and
                resume of devices during system-wide power transitions.
                This can be useful on platforms where device
                dependencies are not well-defined, or for debugging
                power management issues. Asynchronous operations are
                enabled by default.


pm_debug_messages       [SUSPEND,KNL]
                Enable suspend/resume debug messages during boot up.

pnp.debug=1     [PNP]
                Enable PNP debug messages (depends on the
                CONFIG_PNP_DEBUG_MESSAGES option).  Change at run-time
                via /sys/module/pnp/parameters/debug.  We always show
                current resource usage; turning this on also shows
                possible settings and some assignment information.

pnpacpi=        [ACPI]
                { off }

pnpbios=        [ISAPNP]
                { on | off | curr | res | no-curr | no-res }

pnp_reserve_irq=
                [ISAPNP] Exclude IRQs for the autoconfiguration

pnp_reserve_dma=
                [ISAPNP] Exclude DMAs for the autoconfiguration

pnp_reserve_io= [ISAPNP] Exclude I/O ports for the autoconfiguration
                Ranges are in pairs (I/O port base and size).

pnp_reserve_mem=
                [ISAPNP] Exclude memory regions for the
                autoconfiguration.
                Ranges are in pairs (memory base and size).

ports=          [IP_VS_FTP] IPVS ftp helper module
                Default is 21.
                Up to 8 (IP_VS_APP_MAX_PORTS) ports
                may be specified.
```

```
                Format: <port>,<port>....

possible_cpus=  [SMP,S390,X86]
                Format: <unsigned int>
                Set the number of possible CPUs, overriding the
                regular discovery mechanisms (such as ACPI/FW, etc).

powersave=off   [PPC] This option disables power saving features.
                It specifically disables cpuidle and sets the
                platform machine description specific power_save
                function to NULL. On Idle the CPU just reduces
                execution priority.

ppc_strict_facility_enable
                [PPC,ENABLE] This option catches any kernel floating point,
                Altivec, VSX and SPE outside of regions specifically
                allowed (eg kernel_enable_fpu()/kernel_disable_fpu()).
                There is some performance impact when enabling this.

ppc_tm=         [PPC,EARLY]
                Format: {"off"}
                Disable Hardware Transactional Memory

preempt=        [KNL]
                Select preemption mode if you have CONFIG_PREEMPT_DYNAMIC
                none - Limited to cond_resched() calls
                voluntary - Limited to cond_resched() and might_sleep() cal
                full - Any section that isn't explicitly preempt disabled
                       can be preempted anytime.  Tasks will also yield
                       contended spinlocks (if the critical section isn't
                       explicitly preempt disabled beyond the lock itself).
                lazy - Scheduler controlled. Similar to full but instead
                       of preempting the task immediately, the task gets
                       one HZ tick time to yield itself before the
                       preemption will be forced. One preemption is when th
                       task returns to user space.

print-fatal-signals=
                [KNL] debug: print fatal signals

                If enabled, warn about various signal handling
                related application anomalies: too many signals,
                too many POSIX.1 timers, fatal signals causing a
                coredump - etc.

                If you hit the warning due to signal overflow,
                you might want to try "ulimit -i unlimited".

                default: off.

printk.always_kmsg_dump=
                Trigger kmsg_dump for cases other than kernel oops or
```

```
                panics
                Format: <bool>  (1/Y/y=enable, 0/N/n=disable)
                default: disabled

    printk.console_no_auto_verbose=
                Disable console loglevel raise on oops, panic
                or lockdep-detected issues (only if lock debug is on).
                With an exception to setups with low baudrate on
                serial console, keeping this 0 is a good choice
                in order to provide more debug information.
                Format: <bool>
                default: 0 (auto_verbose is enabled)

    printk.debug_non_panic_cpus=
                Allows storing messages from non-panic CPUs into
                the printk log buffer during panic(). They are
                flushed to consoles by the panic-CPU on
                a best-effort basis.
                Format: <bool> (1/Y/y=enable, 0/N/n=disable)
                Default: disabled

    printk.devkmsg={on,off,ratelimit}
                Control writing to /dev/kmsg.
                on - unlimited logging to /dev/kmsg from userspace
                off - logging to /dev/kmsg disabled
                ratelimit - ratelimit the logging
                Default: ratelimit

    printk.time=    Show timing data prefixed to each printk message line
                Format: <bool>  (1/Y/y=enable, 0/N/n=disable)

    proc_mem.force_override= [KNL]
                Format: {always | ptrace | never}
                Traditionally /proc/pid/mem allows memory permissions to be
                overridden without restrictions. This option may be set to
                restrict that. Can be one of:
                - 'always': traditional behavior always allows mem override
                - 'ptrace': only allow mem overrides for active ptracers.
                - 'never':  never allow mem overrides.
                If not specified, default is the CONFIG_PROC_MEM_* choice.

    processor.max_cstate=   [HW,ACPI]
                Limit processor to maximum C-state
                max_cstate=9 overrides any DMI blacklist limit.

    processor.nocst [HW,ACPI]
                Ignore the _CST method to determine C-states,
                instead using the legacy FADT method

    profile=        [KNL] Enable kernel profiling via /proc/profile
                Format: [<profiletype>,]<number>
                Param: <profiletype>: "schedule" or "kvm"
```

```
                        [defaults to kernel profiling]
                Param: "schedule" - profile schedule points.
                Param: "kvm" - profile VM exits.
                Param: <number> - step/bucket size as a power of 2 for
                        statistical time based profiling.

prot_virt=      [S390] enable hosting protected virtual machines
                isolated from the hypervisor (if hardware supports
                that). If enabled, the default kernel base address
                might be overridden even when Kernel Address Space
                Layout Randomization is disabled.
                Format: <bool>

psi=            [KNL] Enable or disable pressure stall information
                tracking.
                Format: <bool>

psmouse.proto=  [HW,MOUSE] Highest PS2 mouse protocol extension to
                probe for; one of (bare|imps|exps|lifebook|any).
psmouse.rate=   [HW,MOUSE] Set desired mouse report rate, in reports
                per second.
psmouse.resetafter=     [HW,MOUSE]
                Try to reset the device after so many bad packets
                (0 = never).
psmouse.resolution=
                [HW,MOUSE] Set desired mouse resolution, in dpi.
psmouse.smartscroll=
                [HW,MOUSE] Controls Logitech smartscroll autorepeat.
                0 = disabled, 1 = enabled (default).

pstore.backend= Specify the name of the pstore backend to use

pti=            [X86-64] Control Page Table Isolation of user and
                kernel address spaces.  Disabling this feature
                removes hardening, but improves performance of
                system calls and interrupts.

                on   - unconditionally enable
                off  - unconditionally disable
                auto - kernel detects whether your CPU model is
                       vulnerable to issues that PTI mitigates

                Not specifying this option is equivalent to pti=auto.

pty.legacy_count=
                [KNL] Number of legacy pty's. Overwrites compiled-in
                default number.

quiet           [KNL,EARLY] Disable most log messages

r128=           [HW,DRM]
```

```
	radix_hcall_invalidate=on  [PPC/PSERIES]
			Disable RADIX GTSE feature and use hcall for TLB
			invalidate.

	raid=		[HW,RAID]
			See Documentation/admin-guide/md.rst.

	ramdisk_size=	[RAM] Sizes of RAM disks in kilobytes
			See Documentation/admin-guide/blockdev/ramdisk.rst.

	ramdisk_start=	[Deprecated,RAM] RAM disk image start address

	random.trust_cpu=off
			[KNL,EARLY] Disable trusting the use of the CPU's
			random number generator (if available) to
			initialize the kernel's RNG.

	random.trust_bootloader=off
			[KNL,EARLY] Disable trusting the use of the a seed
			passed by the bootloader (if available) to
			initialize the kernel's RNG.

	randomize_kstack_offset=
			[KNL,EARLY] Enable or disable kernel stack offset
			randomization, which provides roughly 5 bits of
			entropy, frustrating memory corruption attacks
			that depend on stack address determinism or
			cross-syscall address exposures. This is only
			available on architectures that have defined
			CONFIG_HAVE_ARCH_RANDOMIZE_KSTACK_OFFSET.
			Format: <bool>  (1/Y/y=enable, 0/N/n=disable)
			Default is CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT.

	ras=option[,option,...] [KNL] RAS-specific options

		cec_disable	[X86]
				Disable the Correctable Errors Collector,
				see CONFIG_RAS_CEC help text.

	rcu_nocbs[=cpu-list]
			[KNL] The optional argument is a cpu list,
			as described above.

			In kernels built with CONFIG_RCU_NOCB_CPU=y,
			enable the no-callback CPU mode, which prevents
			such CPUs' callbacks from being invoked in
			softirq context.  Invocation of such CPUs' RCU
			callbacks will instead be offloaded to "rcuox/N"
			kthreads created for that purpose, where "x" is
			"p" for RCU-preempt, "s" for RCU-sched, and "g"
			for the kthreads that mediate grace periods; and
			"N" is the CPU number. This reduces OS jitter on
```

```
                the offloaded CPUs, which can be useful for HPC
                and real-time workloads.  It can also improve
                energy efficiency for asymmetric multiprocessors.

                If a cpulist is passed as an argument, the specified
                list of CPUs is set to no-callback mode from boot.

                Otherwise, if the '=' sign and the cpulist
                arguments are omitted, no CPU will be set to
                no-callback mode from boot but the mode may be
                toggled at runtime via cpusets.

                Note that this argument takes precedence over
                the CONFIG_RCU_NOCB_CPU_DEFAULT_ALL option.

    rcu_nocb_poll   [KNL]
                Rather than requiring that offloaded CPUs
                (specified by rcu_nocbs= above) explicitly
                awaken the corresponding "rcuoN" kthreads,
                make these kthreads poll for callbacks.
                This improves the real-time response for the
                offloaded CPUs by relieving them of the need to
                wake up the corresponding kthread, but degrades
                energy efficiency by requiring that the kthreads
                periodically wake up to do the polling.

    rcutree.blimit= [KNL]
                Set maximum number of finished RCU callbacks to
                process in one batch.

    rcutree.csd_lock_suppress_rcu_stall=    [KNL]
                Do only a one-line RCU CPU stall warning when
                there is an ongoing too-long CSD-lock wait.

    rcutree.do_rcu_barrier= [KNL]
                Request a call to rcu_barrier().  This is
                throttled so that userspace tests can safely
                hammer on the sysfs variable if they so choose.
                If triggered before the RCU grace-period machinery
                is fully active, this will error out with EAGAIN.

    rcutree.dump_tree=      [KNL]
                Dump the structure of the rcu_node combining tree
                out at early boot.  This is used for diagnostic
                purposes, to verify correct tree setup.

    rcutree.gp_cleanup_delay=       [KNL]
                Set the number of jiffies to delay each step of
                RCU grace-period cleanup.

    rcutree.gp_init_delay=  [KNL]
                Set the number of jiffies to delay each step of
```

```
                RCU grace-period initialization.

        rcutree.gp_preinit_delay=       [KNL]
                Set the number of jiffies to delay each step of
                RCU grace-period pre-initialization, that is,
                the propagation of recent CPU-hotplug changes up
                the rcu_node combining tree.

        rcutree.jiffies_till_first_fqs= [KNL]
                Set delay from grace-period initialization to
                first attempt to force quiescent states.
                Units are jiffies, minimum value is zero,
                and maximum value is HZ.

        rcutree.jiffies_till_next_fqs= [KNL]
                Set delay between subsequent attempts to force
                quiescent states.  Units are jiffies, minimum
                value is one, and maximum value is HZ.

        rcutree.jiffies_till_sched_qs= [KNL]
                Set required age in jiffies for a
                given grace period before RCU starts
                soliciting quiescent-state help from
                rcu_note_context_switch() and cond_resched().
                If not specified, the kernel will calculate
                a value based on the most recent settings
                of rcutree.jiffies_till_first_fqs
                and rcutree.jiffies_till_next_fqs.
                This calculated value may be viewed in
                rcutree.jiffies_to_sched_qs.  Any attempt to set
                rcutree.jiffies_to_sched_qs will be cheerfully
                overwritten.

        rcutree.kthread_prio=    [KNL,BOOT]
                Set the SCHED_FIFO priority of the RCU per-CPU
                kthreads (rcuc/N). This value is also used for
                the priority of the RCU boost threads (rcub/N)
                and for the RCU grace-period kthreads (rcu_bh,
                rcu_preempt, and rcu_sched). If RCU_BOOST is
                set, valid values are 1-99 and the default is 1
                (the least-favored priority).  Otherwise, when
                RCU_BOOST is not set, valid values are 0-99 and
                the default is zero (non-realtime operation).
                When RCU_NOCB_CPU is set, also adjust the
                priority of NOCB callback kthreads.

        rcutree.nocb_nobypass_lim_per_jiffy= [KNL]
                On callback-offloaded (rcu_nocbs) CPUs,
                RCU reduces the lock contention that would
                otherwise be caused by callback floods through
                use of the ->nocb_bypass list.  However, in the
                common non-flooded case, RCU queues directly to
```

```
                the main ->cblist in order to avoid the extra
                overhead of the ->nocb_bypass list and its lock.
                But if there are too many callbacks queued during
                a single jiffy, RCU pre-queues the callbacks into
                the ->nocb_bypass queue.  The definition of "too
                many" is supplied by this kernel boot parameter.

    rcutree.nohz_full_patience_delay= [KNL]
                On callback-offloaded (rcu_nocbs) CPUs, avoid
                disturbing RCU unless the grace period has
                reached the specified age in milliseconds.
                Defaults to zero.  Large values will be capped
                at five seconds.  All values will be rounded down
                to the nearest value representable by jiffies.

    rcutree.qhimark= [KNL]
                Set threshold of queued RCU callbacks beyond which
                batch limiting is disabled.

    rcutree.qlowmark= [KNL]
                Set threshold of queued RCU callbacks below which
                batch limiting is re-enabled.

    rcutree.qovld= [KNL]
                Set threshold of queued RCU callbacks beyond which
                RCU's force-quiescent-state scan will aggressively
                enlist help from cond_resched() and sched IPIs to
                help CPUs more quickly reach quiescent states.
                Set to less than zero to make this be set based
                on rcutree.qhimark at boot time and to zero to
                disable more aggressive help enlistment.

    rcutree.rcu_delay_page_cache_fill_msec= [KNL]
                Set the page-cache refill delay (in milliseconds)
                in response to low-memory conditions.  The range
                of permitted values is in the range 0:100000.

    rcutree.rcu_divisor= [KNL]
                Set the shift-right count to use to compute
                the callback-invocation batch limit bl from
                the number of callbacks queued on this CPU.
                The result will be bounded below by the value of
                the rcutree.blimit kernel parameter.  Every bl
                callbacks, the softirq handler will exit in
                order to allow the CPU to do other work.

                Please note that this callback-invocation batch
                limit applies only to non-offloaded callback
                invocation.  Offloaded callbacks are instead
                invoked in the context of an rcuoc kthread, which
                scheduler will preempt as it does any other task.
```

```
	rcutree.rcu_fanout_exact= [KNL]
			Disable autobalancing of the rcu_node combining
			tree.  This is used by rcutorture, and might
			possibly be useful for architectures having high
			cache-to-cache transfer latencies.

	rcutree.rcu_fanout_leaf= [KNL]
			Change the number of CPUs assigned to each
			leaf rcu_node structure.  Useful for very
			large systems, which will choose the value 64,
			and for NUMA systems with large remote-access
			latencies, which will choose a value aligned
			with the appropriate hardware boundaries.

	rcutree.rcu_min_cached_objs= [KNL]
			Minimum number of objects which are cached and
			maintained per one CPU. Object size is equal
			to PAGE_SIZE. The cache allows to reduce the
			pressure to page allocator, also it makes the
			whole algorithm to behave better in low memory
			condition.

	rcutree.rcu_nocb_gp_stride= [KNL]
			Set the number of NOCB callback kthreads in
			each group, which defaults to the square root
			of the number of CPUs.  Larger numbers reduce
			the wakeup overhead on the global grace-period
			kthread, but increases that same overhead on
			each group's NOCB grace-period kthread.

	rcutree.rcu_kick_kthreads= [KNL]
			Cause the grace-period kthread to get an extra
			wake_up() if it sleeps three times longer than
			it should at force-quiescent-state time.
			This wake_up() will be accompanied by a
			WARN_ONCE() splat and an ftrace_dump().

	rcutree.rcu_resched_ns= [KNL]
			Limit the time spend invoking a batch of RCU
			callbacks to the specified number of nanoseconds.
			By default, this limit is checked only once
			every 32 callbacks in order to limit the pain
			inflicted by local_clock() overhead.

	rcutree.rcu_unlock_delay= [KNL]
			In CONFIG_RCU_STRICT_GRACE_PERIOD=y kernels,
			this specifies an rcu_read_unlock()-time delay
			in microseconds.  This defaults to zero.
			Larger delays increase the probability of
			catching RCU pointer leaks, that is, buggy use
			of RCU-protected pointers after the relevant
			rcu_read_unlock() has completed.
```

```
	rcutree.sysrq_rcu= [KNL]
			Commandeer a sysrq key to dump out Tree RCU's
			rcu_node tree with an eye towards determining
			why a new grace period has not yet started.

	rcutree.use_softirq=	[KNL]
			If set to zero, move all RCU_SOFTIRQ processing to
			per-CPU rcuc kthreads.  Defaults to a non-zero
			value, meaning that RCU_SOFTIRQ is used by default.
			Specify rcutree.use_softirq=0 to use rcuc kthreads.

			But note that CONFIG_PREEMPT_RT=y kernels disable
			this kernel boot parameter, forcibly setting it
			to zero.

	rcutree.enable_rcu_lazy= [KNL]
			To save power, batch RCU callbacks and flush after
			delay, memory pressure or callback list growing too
			big.

	rcutree.rcu_normal_wake_from_gp= [KNL]
			Reduces a latency of synchronize_rcu() call. This approach
			maintains its own track of synchronize_rcu() callers, so it
			does not interact with regular callbacks because it does no
			use a call_rcu[_hurry]() path. Please note, this is for a
			normal grace period.

			How to disable it:

			echo 0 > /sys/module/rcutree/parameters/rcu_normal_wake_fro
			or pass a boot parameter "rcutree.rcu_normal_wake_from_gp=0

			Default is 1 if it is not explicitly disabled by the boot p
			passing 0.

	rcuscale.gp_async= [KNL]
			Measure performance of asynchronous
			grace-period primitives such as call_rcu().

	rcuscale.gp_async_max= [KNL]
			Specify the maximum number of outstanding
			callbacks per writer thread.  When a writer
			thread exceeds this limit, it invokes the
			corresponding flavor of rcu_barrier() to allow
			previously posted callbacks to drain.

	rcuscale.gp_exp= [KNL]
			Measure performance of expedited synchronous
			grace-period primitives.

	rcuscale.holdoff= [KNL]
```

```
                Set test-start holdoff period.  The purpose of
                this parameter is to delay the start of the
                test until boot completes in order to avoid
                interference.

        rcuscale.kfree_by_call_rcu= [KNL]
                In kernels built with CONFIG_RCU_LAZY=y, test
                call_rcu() instead of kfree_rcu().

        rcuscale.kfree_mult= [KNL]
                Instead of allocating an object of size kfree_obj,
                allocate one of kfree_mult * sizeof(kfree_obj).
                Defaults to 1.

        rcuscale.kfree_rcu_test= [KNL]
                Set to measure performance of kfree_rcu() flooding.

        rcuscale.kfree_rcu_test_double= [KNL]
                Test the double-argument variant of kfree_rcu().
                If this parameter has the same value as
                rcuscale.kfree_rcu_test_single, both the single-
                and double-argument variants are tested.

        rcuscale.kfree_rcu_test_single= [KNL]
                Test the single-argument variant of kfree_rcu().
                If this parameter has the same value as
                rcuscale.kfree_rcu_test_double, both the single-
                and double-argument variants are tested.

        rcuscale.kfree_nthreads= [KNL]
                The number of threads running loops of kfree_rcu().

        rcuscale.kfree_alloc_num= [KNL]
                Number of allocations and frees done in an iteration.

        rcuscale.kfree_loops= [KNL]
                Number of loops doing rcuscale.kfree_alloc_num number
                of allocations and frees.

        rcuscale.minruntime= [KNL]
                Set the minimum test run time in seconds.  This
                does not affect the data-collection interval,
                but instead allows better measurement of things
                like CPU consumption.

        rcuscale.nreaders= [KNL]
                Set number of RCU readers.  The value -1 selects
                N, where N is the number of CPUs.  A value
                "n" less than -1 selects N-n+1, where N is again
                the number of CPUs.  For example, -2 selects N
                (the number of CPUs), -3 selects N+1, and so on.
                A value of "n" less than or equal to -N selects
```

```
                a single reader.

    rcuscale.nwriters= [KNL]
                Set number of RCU writers.  The values operate
                the same as for rcuscale.nreaders.
                N, where N is the number of CPUs

    rcuscale.scale_type= [KNL]
                Specify the RCU implementation to test.

    rcuscale.shutdown= [KNL]
                Shut the system down after performance tests
                complete.  This is useful for hands-off automated
                testing.

    rcuscale.verbose= [KNL]
                Enable additional printk() statements.

    rcuscale.writer_holdoff= [KNL]
                Write-side holdoff between grace periods,
                in microseconds.  The default of zero says
                no holdoff.

    rcuscale.writer_holdoff_jiffies= [KNL]
                Additional write-side holdoff between grace
                periods, but in jiffies.  The default of zero
                says no holdoff.

    rcutorture.fqs_duration= [KNL]
                Set duration of force_quiescent_state bursts
                in microseconds.

    rcutorture.fqs_holdoff= [KNL]
                Set holdoff time within force_quiescent_state bursts
                in microseconds.

    rcutorture.fqs_stutter= [KNL]
                Set wait time between force_quiescent_state bursts
                in seconds.

    rcutorture.fwd_progress= [KNL]
                Specifies the number of kthreads to be used
                for  RCU grace-period forward-progress testing
                for the types of RCU supporting this notion.
                Defaults to 1 kthread, values less than zero or
                greater than the number of CPUs cause the number
                of CPUs to be used.

    rcutorture.fwd_progress_div= [KNL]
                Specify the fraction of a CPU-stall-warning
                period to do tight-loop forward-progress testing.
```

```
	rcutorture.fwd_progress_holdoff= [KNL]
			Number of seconds to wait between successive
			forward-progress tests.

	rcutorture.fwd_progress_need_resched= [KNL]
			Enclose cond_resched() calls within checks for
			need_resched() during tight-loop forward-progress
			testing.

	rcutorture.gp_cond= [KNL]
			Use conditional/asynchronous update-side
			normal-grace-period primitives, if available.

	rcutorture.gp_cond_exp= [KNL]
			Use conditional/asynchronous update-side
			expedited-grace-period primitives, if available.

	rcutorture.gp_cond_full= [KNL]
			Use conditional/asynchronous update-side
			normal-grace-period primitives that also take
			concurrent expedited grace periods into account,
			if available.

	rcutorture.gp_cond_exp_full= [KNL]
			Use conditional/asynchronous update-side
			expedited-grace-period primitives that also take
			concurrent normal grace periods into account,
			if available.

	rcutorture.gp_cond_wi= [KNL]
			Nominal wait interval for normal conditional
			grace periods (specified by rcutorture's
			gp_cond and gp_cond_full module parameters),
			in microseconds.  The actual wait interval will
			be randomly selected to nanosecond granularity up
			to this wait interval.  Defaults to 16 jiffies,
			for example, 16,000 microseconds on a system
			with HZ=1000.

	rcutorture.gp_cond_wi_exp= [KNL]
			Nominal wait interval for expedited conditional
			grace periods (specified by rcutorture's
			gp_cond_exp and gp_cond_exp_full module
			parameters), in microseconds.  The actual wait
			interval will be randomly selected to nanosecond
			granularity up to this wait interval.  Defaults to
			128 microseconds.

	rcutorture.gp_exp= [KNL]
			Use expedited update-side primitives, if available.

	rcutorture.gp_normal= [KNL]
```

```
                Use normal (non-expedited) asynchronous
                update-side primitives, if available.

    rcutorture.gp_poll= [KNL]
                Use polled update-side normal-grace-period
                primitives, if available.

    rcutorture.gp_poll_exp= [KNL]
                Use polled update-side expedited-grace-period
                primitives, if available.

    rcutorture.gp_poll_full= [KNL]
                Use polled update-side normal-grace-period
                primitives that also take concurrent expedited
                grace periods into account, if available.

    rcutorture.gp_poll_exp_full= [KNL]
                Use polled update-side expedited-grace-period
                primitives that also take concurrent normal
                grace periods into account, if available.

    rcutorture.gp_poll_wi= [KNL]
                Nominal wait interval for normal conditional
                grace periods (specified by rcutorture's
                gp_poll and gp_poll_full module parameters),
                in microseconds.  The actual wait interval will
                be randomly selected to nanosecond granularity up
                to this wait interval.  Defaults to 16 jiffies,
                for example, 16,000 microseconds on a system
                with HZ=1000.

    rcutorture.gp_poll_wi_exp= [KNL]
                Nominal wait interval for expedited conditional
                grace periods (specified by rcutorture's
                gp_poll_exp and gp_poll_exp_full module
                parameters), in microseconds.  The actual wait
                interval will be randomly selected to nanosecond
                granularity up to this wait interval.  Defaults to
                128 microseconds.

    rcutorture.gp_sync= [KNL]
                Use normal (non-expedited) synchronous
                update-side primitives, if available.  If all
                of rcutorture.gp_cond=, rcutorture.gp_exp=,
                rcutorture.gp_normal=, and rcutorture.gp_sync=
                are zero, rcutorture acts as if is interpreted
                they are all non-zero.

    rcutorture.gpwrap_lag= [KNL]
                Enable grace-period wrap lag testing. Setting
                to false prevents the gpwrap lag test from
                running. Default is true.
```

```
rcutorture.gpwrap_lag_gps= [KNL]
                Set the value for grace-period wrap lag during
                active lag testing periods. This controls how many
                grace periods differences we tolerate between
                rdp and rnp's gp_seq before setting overflow flag.
                The default is always set to 8.

rcutorture.gpwrap_lag_cycle_mins= [KNL]
                Set the total cycle duration for gpwrap lag
                testing in minutes. This is the total time for
                one complete cycle of active and inactive
                testing periods. Default is 30 minutes.

rcutorture.gpwrap_lag_active_mins= [KNL]
                Set the duration for which gpwrap lag is active
                within each cycle, in minutes. During this time,
                the grace-period wrap lag will be set to the
                value specified by gpwrap_lag_gps. Default is
                5 minutes.

rcutorture.irqreader= [KNL]
                Run RCU readers from irq handlers, or, more
                accurately, from a timer handler.  Not all RCU
                flavors take kindly to this sort of thing.

rcutorture.leakpointer= [KNL]
                Leak an RCU-protected pointer out of the reader.
                This can of course result in splats, and is
                intended to test the ability of things like
                CONFIG_RCU_STRICT_GRACE_PERIOD=y to detect
                such leaks.

rcutorture.n_barrier_cbs= [KNL]
                Set callbacks/threads for rcu_barrier() testing.

rcutorture.nfakewriters= [KNL]
                Set number of concurrent RCU writers.  These just
                stress RCU, they don't participate in the actual
                test, hence the "fake".

rcutorture.nocbs_nthreads= [KNL]
                Set number of RCU callback-offload togglers.
                Zero (the default) disables toggling.

rcutorture.nocbs_toggle= [KNL]
                Set the delay in milliseconds between successive
                callback-offload toggling attempts.

rcutorture.nreaders= [KNL]
                Set number of RCU readers.  The value -1 selects
                N-1, where N is the number of CPUs.  A value
```

```
                "n" less than -1 selects N-n-2, where N is again
                the number of CPUs.  For example, -2 selects N
                (the number of CPUs), -3 selects N+1, and so on.

        rcutorture.object_debug= [KNL]
                Enable debug-object double-call_rcu() testing.

        rcutorture.onoff_holdoff= [KNL]
                Set time (s) after boot for CPU-hotplug testing.

        rcutorture.onoff_interval= [KNL]
                Set time (jiffies) between CPU-hotplug operations,
                or zero to disable CPU-hotplug testing.

        rcutorture.preempt_duration= [KNL]
                Set duration (in milliseconds) of preemptions
                by a high-priority FIFO real-time task.  Set to
                zero (the default) to disable.  The CPUs to
                preempt are selected randomly from the set that
                are online at a given point in time.  Races with
                CPUs going offline are ignored, with that attempt
                at preemption skipped.

        rcutorture.preempt_interval= [KNL]
                Set interval (in milliseconds, defaulting to one
                second) between preemptions by a high-priority
                FIFO real-time task.  This delay is mediated
                by an hrtimer and is further fuzzed to avoid
                inadvertent synchronizations.

        rcutorture.read_exit_burst= [KNL]
                The number of times in a given read-then-exit
                episode that a set of read-then-exit kthreads
                is spawned.

        rcutorture.read_exit_delay= [KNL]
                The delay, in seconds, between successive
                read-then-exit testing episodes.

        rcutorture.reader_flavor= [KNL]
                A bit mask indicating which readers to use.
                If there is more than one bit set, the readers
                are entered from low-order bit up, and are
                exited in the opposite order.  For SRCU, the
                0x1 bit is normal readers, 0x2 NMI-safe readers,
                and 0x4 light-weight readers.

        rcutorture.shuffle_interval= [KNL]
                Set task-shuffle interval (s).  Shuffling tasks
                allows some CPUs to go into dyntick-idle mode
                during the rcutorture test.
```

```
	rcutorture.shutdown_secs= [KNL]
			Set time (s) after boot system shutdown.  This
			is useful for hands-off automated testing.

	rcutorture.stall_cpu= [KNL]
			Duration of CPU stall (s) to test RCU CPU stall
			warnings, zero to disable.

	rcutorture.stall_cpu_block= [KNL]
			Sleep while stalling if set.  This will result
			in warnings from preemptible RCU in addition to
			any other stall-related activity.  Note that
			in kernels built with CONFIG_PREEMPTION=n and
			CONFIG_PREEMPT_COUNT=y, this parameter will
			cause the CPU to pass through a quiescent state.
			Given CONFIG_PREEMPTION=n, this will suppress
			RCU CPU stall warnings, but will instead result
			in scheduling-while-atomic splats.

			Use of this module parameter results in splats.


	rcutorture.stall_cpu_holdoff= [KNL]
			Time to wait (s) after boot before inducing stall.

	rcutorture.stall_cpu_irqsoff= [KNL]
			Disable interrupts while stalling if set, but only
			on the first stall in the set.

	rcutorture.stall_cpu_repeat= [KNL]
			Number of times to repeat the stall sequence,
			so that rcutorture.stall_cpu_repeat=3 will result
			in four stall sequences.

	rcutorture.stall_gp_kthread= [KNL]
			Duration (s) of forced sleep within RCU
			grace-period kthread to test RCU CPU stall
			warnings, zero to disable.  If both stall_cpu
			and stall_gp_kthread are specified, the
			kthread is starved first, then the CPU.

	rcutorture.stat_interval= [KNL]
			Time (s) between statistics printk()s.

	rcutorture.stutter= [KNL]
			Time (s) to stutter testing, for example, specifying
			five seconds causes the test to run for five seconds,
			wait for five seconds, and so on.  This tests RCU's
			ability to transition abruptly to and from idle.

	rcutorture.test_boost= [KNL]
			Test RCU priority boosting?  0=no, 1=maybe, 2=yes.
```

```
                "Maybe" means test if the RCU implementation
                under test support RCU priority boosting.

    rcutorture.test_boost_duration= [KNL]
                Duration (s) of each individual boost test.

    rcutorture.test_boost_holdoff= [KNL]
                Holdoff time (s) from start of test to the start
                of RCU priority-boost testing.  Defaults to zero,
                that is, no holdoff.

    rcutorture.test_boost_interval= [KNL]
                Interval (s) between each boost test.

    rcutorture.test_no_idle_hz= [KNL]
                Test RCU's dyntick-idle handling.  See also the
                rcutorture.shuffle_interval parameter.

    rcutorture.torture_type= [KNL]
                Specify the RCU implementation to test.

    rcutorture.verbose= [KNL]
                Enable additional printk() statements.

    rcupdate.rcu_cpu_stall_ftrace_dump= [KNL]
                Dump ftrace buffer after reporting RCU CPU
                stall warning.

    rcupdate.rcu_cpu_stall_notifiers= [KNL]
                Provide RCU CPU stall notifiers, but see the
                warnings in the RCU_CPU_STALL_NOTIFIER Kconfig
                option's help text.  TL;DR:  You almost certainly
                do not want rcupdate.rcu_cpu_stall_notifiers.

    rcupdate.rcu_cpu_stall_suppress= [KNL]
                Suppress RCU CPU stall warning messages.

    rcupdate.rcu_cpu_stall_suppress_at_boot= [KNL]
                Suppress RCU CPU stall warning messages and
                rcutorture writer stall warnings that occur
                during early boot, that is, during the time
                before the init task is spawned.

    rcupdate.rcu_cpu_stall_timeout= [KNL]
                Set timeout for RCU CPU stall warning messages.
                The value is in seconds and the maximum allowed
                value is 300 seconds.

    rcupdate.rcu_exp_cpu_stall_timeout= [KNL]
                Set timeout for expedited RCU CPU stall warning
                messages.  The value is in milliseconds
                and the maximum allowed value is 21000
```

```
                milliseconds. Please note that this value is
                adjusted to an arch timer tick resolution.
                Setting this to zero causes the value from
                rcupdate.rcu_cpu_stall_timeout to be used (after
                conversion from seconds to milliseconds).

    rcupdate.rcu_cpu_stall_cputime= [KNL]
                Provide statistics on the cputime and count of
                interrupts and tasks during the sampling period. For
                multiple continuous RCU stalls, all sampling periods
                begin at half of the first RCU stall timeout.

    rcupdate.rcu_exp_stall_task_details= [KNL]
                Print stack dumps of any tasks blocking the
                current expedited RCU grace period during an
                expedited RCU CPU stall warning.

    rcupdate.rcu_expedited= [KNL]
                Use expedited grace-period primitives, for
                example, synchronize_rcu_expedited() instead
                of synchronize_rcu().  This reduces latency,
                but can increase CPU utilization, degrade
                real-time latency, and degrade energy efficiency.
                No effect on CONFIG_TINY_RCU kernels.

    rcupdate.rcu_normal= [KNL]
                Use only normal grace-period primitives,
                for example, synchronize_rcu() instead of
                synchronize_rcu_expedited().  This improves
                real-time latency, CPU utilization, and
                energy efficiency, but can expose users to
                increased grace-period latency.  This parameter
                overrides rcupdate.rcu_expedited.  No effect on
                CONFIG_TINY_RCU kernels.

    rcupdate.rcu_normal_after_boot= [KNL]
                Once boot has completed (that is, after
                rcu_end_inkernel_boot() has been invoked), use
                only normal grace-period primitives.  No effect
                on CONFIG_TINY_RCU kernels.

                But note that CONFIG_PREEMPT_RT=y kernels enables
                this kernel boot parameter, forcibly setting
                it to the value one, that is, converting any
                post-boot attempt at an expedited RCU grace
                period to instead use normal non-expedited
                grace-period processing.

    rcupdate.rcu_task_collapse_lim= [KNL]
                Set the maximum number of callbacks present
                at the beginning of a grace period that allows
                the RCU Tasks flavors to collapse back to using
```

```
                a single callback queue.  This switching only
                occurs when rcupdate.rcu_task_enqueue_lim is
                set to the default value of -1.

    rcupdate.rcu_task_contend_lim= [KNL]
                Set the minimum number of callback-queuing-time
                lock-contention events per jiffy required to
                cause the RCU Tasks flavors to switch to per-CPU
                callback queuing.  This switching only occurs
                when rcupdate.rcu_task_enqueue_lim is set to
                the default value of -1.

    rcupdate.rcu_task_enqueue_lim= [KNL]
                Set the number of callback queues to use for the
                RCU Tasks family of RCU flavors.  The default
                of -1 allows this to be automatically (and
                dynamically) adjusted.  This parameter is intended
                for use in testing.

    rcupdate.rcu_task_lazy_lim= [KNL]
                Number of callbacks on a given CPU that will
                cancel laziness on that CPU.  Use -1 to disable
                cancellation of laziness, but be advised that
                doing so increases the danger of OOM due to
                callback flooding.

    rcupdate.rcu_task_stall_info= [KNL]
                Set initial timeout in jiffies for RCU task stall
                informational messages, which give some indication
                of the problem for those not patient enough to
                wait for ten minutes.  Informational messages are
                only printed prior to the stall-warning message
                for a given grace period. Disable with a value
                less than or equal to zero.  Defaults to ten
                seconds.  A change in value does not take effect
                until the beginning of the next grace period.

    rcupdate.rcu_task_stall_info_mult= [KNL]
                Multiplier for time interval between successive
                RCU task stall informational messages for a given
                RCU tasks grace period.  This value is clamped
                to one through ten, inclusive.  It defaults to
                the value three, so that the first informational
                message is printed 10 seconds into the grace
                period, the second at 40 seconds, the third at
                160 seconds, and then the stall warning at 600
                seconds would prevent a fourth at 640 seconds.

    rcupdate.rcu_task_stall_timeout= [KNL]
                Set timeout in jiffies for RCU task stall
                warning messages.  Disable with a value less
                than or equal to zero.  Defaults to ten minutes.
```

```
                A change in value does not take effect until
                the beginning of the next grace period.

    rcupdate.rcu_tasks_lazy_ms= [KNL]
                Set timeout in milliseconds RCU Tasks asynchronous
                callback batching for call_rcu_tasks().
                A negative value will take the default.  A value
                of zero will disable batching.  Batching is
                always disabled for synchronize_rcu_tasks().

    rcupdate.rcu_self_test= [KNL]
                Run the RCU early boot self tests

    rdinit=         [KNL]
                Format: <full_path>
                Run specified binary instead of /init from the ramdisk,
                used for early userspace startup. See initrd.

    rdrand=         [X86,EARLY]
                force - Override the decision by the kernel to hide the
                        advertisement of RDRAND support (this affects
                        certain AMD processors because of buggy BIOS
                        support, specifically around the suspend/resume
                        path).

    rdt=            [HW,X86,RDT]
                Turn on/off individual RDT features. List is:
                cmt, mbmtotal, mbmlocal, l3cat, l3cdp, l2cat, l2cdp,
                mba, smba, bmec, abmc, sdciae, energy[:guid],
                perf[:guid].
                E.g. to turn on cmt and turn off mba use:
                        rdt=cmt,!mba
                To turn off all energy telemetry monitoring and ensure that
                perf telemetry monitoring associated with guid 0x12345
                is enabled use:
                        rdt=!energy,perf:0x12345

    reboot=         [KNL]
                Format (x86 or x86_64):
                        [w[arm] | c[old] | h[ard] | s[oft] | g[pio]] | d[ef
                        [[,]s[mp]#### \
                        [[,]b[ios] | a[cpi] | k[bd] | t[riple] | e[fi] | p[
                        [[,]f[orce]
                Where reboot_mode is one of warm (soft) or cold (hard) or g
                                (prefix with 'panic_' to set mode for panic
                                reboot only),
                      reboot_type is one of bios, acpi, kbd, triple, efi, o
                      reboot_force is either force or not specified,
                      reboot_cpu is s[mp]#### with #### being the processor
                                to be used for rebooting.

            acpi
```

```
                Use the ACPI RESET_REG in the FADT. If ACPI is not
                configured or the ACPI reset does not work, the reboot
                path attempts the reset using the keyboard controller.

        bios
                Use the CPU reboot vector for warm reset

        cold
                Set the cold reboot flag

        default
                There are some built-in platform specific "quirks"
                - you may see: "reboot: <name> series board detected.
                Selecting <type> for reboots." In the case where you
                think the quirk is in error (e.g. you have newer BIOS,
                or newer board) using this option will ignore the
                built-in quirk table, and use the generic default
                reboot actions.

        efi
                Use efi reset_system runtime service. If EFI is not
                configured or the EFI reset does not work, the reboot
                path attempts the reset using the keyboard controller.

        force
                Don't stop other CPUs on reboot. This can make reboot
                more reliable in some cases.

        kbd
                Use the keyboard controller. cold reset (default)

        pci
                Use a write to the PCI config space register 0xcf9 to
                trigger reboot.

        triple
                Force a triple fault (init)

        warm
                Don't set the cold reboot flag

                Using warm reset will be much faster especially on big
                memory systems because the BIOS will not go through
                the memory check.  Disadvantage is that not all
                hardware will be completely reinitialized on reboot so
                there may be boot problems on some systems.


refscale.holdoff= [KNL]
                Set test-start holdoff period.  The purpose of
                this parameter is to delay the start of the
                test until boot completes in order to avoid
```

```
                interference.

    refscale.lookup_instances= [KNL]
                Number of data elements to use for the forms of
                SLAB_TYPESAFE_BY_RCU testing.  A negative number
                is negated and multiplied by nr_cpu_ids, while
                zero specifies nr_cpu_ids.

    refscale.loops= [KNL]
                Set the number of loops over the synchronization
                primitive under test.  Increasing this number
                reduces noise due to loop start/end overhead,
                but the default has already reduced the per-pass
                noise to a handful of picoseconds on ca. 2020
                x86 laptops.

    refscale.nreaders= [KNL]
                Set number of readers.  The default value of -1
                selects N, where N is roughly 75% of the number
                of CPUs.  A value of zero is an interesting choice.

    refscale.nruns= [KNL]
                Set number of runs, each of which is dumped onto
                the console log.

    refscale.readdelay= [KNL]
                Set the read-side critical-section duration,
                measured in microseconds.

    refscale.scale_type= [KNL]
                Specify the read-protection implementation to test.

    refscale.shutdown= [KNL]
                Shut down the system at the end of the performance
                test.  This defaults to 1 (shut it down) when
                refscale is built into the kernel and to 0 (leave
                it running) when refscale is built as a module.

    refscale.verbose= [KNL]
                Enable additional printk() statements.

    refscale.verbose_batched= [KNL]
                Batch the additional printk() statements.  If zero
                (the default) or negative, print everything.  Otherwise,
                print every Nth verbose statement, where N is the value
                specified.

    regulator_ignore_unused
                [REGULATOR]
                Prevents regulator framework from disabling regulators
                that are unused, due no driver claiming them. This may
                be useful for debug and development, but should not be
```

```
                needed on a platform with proper driver support.

    relax_domain_level=
                [KNL, SMP] Set scheduler's default relax_domain_level.
                See Documentation/admin-guide/cgroup-v1/cpusets.rst.

    reserve=    [KNL,BUGS] Force kernel to ignore I/O ports or memory
                Format: <base1>,<size1>[,<base2>,<size2>,...]
                Reserve I/O ports or memory so the kernel won't use
                them.  If <base> is less than 0x10000, the region
                is assumed to be I/O ports; otherwise it is memory.

    reserve_mem=    [RAM]
                Format: nn[KMG]:<align>:<label>
                Reserve physical memory and label it with a name that
                other subsystems can use to access it. This is typically
                used for systems that do not wipe the RAM, and this command
                line will try to reserve the same physical memory on
                soft reboots. Note, it is not guaranteed to be the same
                location. For example, if anything about the system changes
                or if booting a different kernel. It can also fail if KASLR
                places the kernel at the location of where the RAM reservat
                was from a previous boot, the new reservation will be at a
                different location.
                Any subsystem using this feature must add a way to verify
                that the contents of the physical memory is from a previous
                boot, as there may be cases where the memory will not be
                located at the same location.

                The format is size:align:label for example, to request
                12 megabytes of 4096 alignment for ramoops:

                reserve_mem=12M:4096:oops ramoops.mem_name=oops

    reservetop=     [X86-32,EARLY]
                Format: nn[KMG]
                Reserves a hole at the top of the kernel virtual
                address space.

    reset_devices   [KNL] Force drivers to reset the underlying device
                during initialization.

    resume=     [SWSUSP]
                Specify the partition device for software suspend
                Format:
                {/dev/<dev> | PARTUUID=<uuid> | <int>:<int> | <hex>}

    resume_offset=  [SWSUSP]
                Specify the offset from the beginning of the partition
                given by "resume=" at which the swap header is located,
                in <PAGE_SIZE> units (needed only for swap files).
                See  Documentation/power/swsusp-and-swap-files.rst
```

```
resumedelay=    [HIBERNATION] Delay (in seconds) to pause before attempting
                read the resume files

resumewait      [HIBERNATION] Wait (indefinitely) for resume device to show
                Useful for devices that are detected asynchronously
                (e.g. USB and MMC devices).

retain_initrd   [RAM] Keep initrd memory after extraction. After boot, it w
                be accessible via /sys/firmware/initrd.

retbleed=       [X86] Control mitigation of RETBleed (Arbitrary
                Speculative Code Execution with Return Instructions)
                vulnerability.

                AMD-based UNRET and IBPB mitigations alone do not stop
                sibling threads from influencing the predictions of other
                sibling threads. For that reason, STIBP is used on pro-
                cessors that support it, and mitigate SMT on processors
                that don't.

                off          - no mitigation
                auto         - automatically select a mitigation
                auto,nosmt   - automatically select a mitigation,
                               disabling SMT if necessary for
                               the full mitigation (only on Zen1
                               and older without STIBP).
                ibpb         - On AMD, mitigate short speculation
                               windows on basic block boundaries too.
                               Safe, highest perf impact. It also
                               enables STIBP if present. Not suitable
                               on Intel.
                ibpb,nosmt   - Like "ibpb" above but will disable SMT
                               when STIBP is not available. This is
                               the alternative for systems which do not
                               have STIBP.
                unret        - Force enable untrained return thunks,
                               only effective on AMD f15h-f17h based
                               systems.
                unret,nosmt  - Like unret, but will disable SMT when STIBP
                               is not available. This is the alternative fo
                               systems which do not have STIBP.

                Selecting 'auto' will choose a mitigation method at run
                time according to the CPU.

                Not specifying this option is equivalent to retbleed=auto.

rfkill.default_state=
        0       "airplane mode".  All wifi, bluetooth, wimax, gps, fm,
                etc. communication is blocked by default.
        1       Unblocked.
```

```
rfkill.master_switch_mode=
        0       The "airplane mode" button does nothing.
        1       The "airplane mode" button toggles between everything
                blocked and the previous configuration.
        2       The "airplane mode" button toggles between everything
                blocked and everything unblocked.

ring3mwait=disable
                [KNL] Disable ring 3 MONITOR/MWAIT feature on supported
                CPUs.

riscv_isa_fallback [RISCV,EARLY]
                When CONFIG_RISCV_ISA_FALLBACK is not enabled, permit
                falling back to detecting extension support by parsing
                "riscv,isa" property on devicetree systems when the
                replacement properties are not found. See the Kconfig
                entry for RISCV_ISA_FALLBACK.

riscv_nousercfi=
        all     Disable user CFI ABI to userspace even if cpu extension
                are available.
        bcfi    Disable user backward CFI ABI to userspace even if
                the shadow stack extension is available.
        fcfi    Disable user forward CFI ABI to userspace even if the
                landing pad extension is available.

ro              [KNL] Mount root device read-only on boot

rodata=         [KNL,EARLY]
        on      Mark read-only kernel memory as read-only (default).
        off     Leave read-only kernel memory writable for debugging.
        noalias Mark read-only kernel memory as read-only but retain
                writable aliases in the direct map for regions outside
                of the kernel image. [arm64]

rockchip.usb_uart
                [EARLY]
                Enable the uart passthrough on the designated usb port
                on Rockchip SoCs. When active, the signals of the
                debug-uart get routed to the D+ and D- pins of the usb
                port and the regular usb controller gets disabled.

root=           [KNL] Root filesystem
                Usually this is a block device specifier of some kind,
                see the early_lookup_bdev comment in
                block/early-lookup.c for details.
                Alternatively this can be "ram" for the legacy initial
                ramdisk, "nfs" and "cifs" for root on a network file
                system, or "mtd" and "ubi" for mounting from raw flash.

rootdelay=      [KNL] Delay (in seconds) to pause before attempting to
```

```
                mount the root filesystem

rootflags=      [KNL] Set root filesystem mount option string

rseq_slice_ext= [KNL] RSEQ based time slice extension
                Format: boolean
                Control enablement of RSEQ based time slice extension.
                Default is 'on'.

initramfs_options= [KNL]
                Specify mount options for the initramfs mount.

rootfstype=     [KNL] Set root filesystem type

rootwait        [KNL] Wait (indefinitely) for root device to show up.
                Useful for devices that are detected asynchronously
                (e.g. USB and MMC devices).

rootwait=       [KNL] Maximum time (in seconds) to wait for root device
                to show up before attempting to mount the root
                filesystem.

rproc_mem=nn[KMG][@address]
                [KNL,ARM,CMA] Remoteproc physical memory block.
                Memory area to be used by remote processor image,
                managed by CMA.

rseq_debug=     [KNL] Enable or disable restartable sequence
                debug mode. Defaults to CONFIG_RSEQ_DEBUG_DEFAULT_ENABLE.
                Format: <bool>

rt_group_sched= [KNL] Enable or disable SCHED_RR/FIFO group scheduling
                when CONFIG_RT_GROUP_SCHED=y. Defaults to
                !CONFIG_RT_GROUP_SCHED_DEFAULT_DISABLED.
                Format: <bool>

rw              [KNL] Mount root device read-write on boot

S               [KNL] Run init in single mode

s390_iommu=     [HW,S390]
                Set s390 IOTLB flushing mode
        strict
                With strict flushing every unmap operation will result
                in an IOTLB flush. Default is lazy flushing before
                reuse, which is faster. Deprecated, equivalent to
                iommu.strict=1.

s390_iommu_aperture=    [KNL,S390]
                Specifies the size of the per device DMA address space
                accessible through the DMA and IOMMU APIs as a decimal
                factor of the size of main memory.
```

```
                The default is 1 meaning that one can concurrently use
                as many DMA addresses as physical memory is installed,
                if supported by hardware, and thus map all of memory
                once. With a value of 2 one can map all of memory twice
                and so on. As a special case a factor of 0 imposes no
                restrictions other than those given by hardware at the
                cost of significant additional memory use for tables.


    sched_proxy_exec= [KNL]
                Enables or disables "proxy execution" style
                solution to mutex-based priority inversion.
                Format: <bool>

    sched_verbose   [KNL,EARLY] Enables verbose scheduler debug messages.

    schedstats=     [KNL,X86] Enable or disable scheduled statistics.
                Allowed values are enable and disable. This feature
                incurs a small amount of overhead in the scheduler
                but is useful for debugging and performance tuning.

    sched_thermal_decay_shift=
                [Deprecated]
                [KNL, SMP] Set a decay shift for scheduler thermal
                pressure signal. Thermal pressure signal follows the
                default decay period of other scheduler pelt
                signals(usually 32 ms but configurable). Setting
                sched_thermal_decay_shift will left shift the decay
                period for the thermal pressure signal by the shift
                value.
                i.e. with the default pelt decay period of 32 ms
                sched_thermal_decay_shift   thermal pressure decay pr
                        1                       64 ms
                        2                       128 ms
                and so on.
                Format: integer between 0 and 10
                Default is 0.

    scftorture.holdoff= [KNL]
                Number of seconds to hold off before starting
                test.  Defaults to zero for module insertion and
                to 10 seconds for built-in smp_call_function()
                tests.

    scftorture.longwait= [KNL]
                Request ridiculously long waits randomly selected
                up to the chosen limit in seconds.  Zero (the
                default) disables this feature.  Please note
                that requesting even small non-zero numbers of
                seconds can result in RCU CPU stall warnings,
                softlockup complaints, and so on.
```

```
	scftorture.nthreads= [KNL]
			Number of kthreads to spawn to invoke the
			smp_call_function() family of functions.
			The default of -1 specifies a number of kthreads
			equal to the number of CPUs.

	scftorture.onoff_holdoff= [KNL]
			Number seconds to wait after the start of the
			test before initiating CPU-hotplug operations.

	scftorture.onoff_interval= [KNL]
			Number seconds to wait between successive
			CPU-hotplug operations.  Specifying zero (which
			is the default) disables CPU-hotplug operations.

	scftorture.shutdown_secs= [KNL]
			The number of seconds following the start of the
			test after which to shut down the system.  The
			default of zero avoids shutting down the system.
			Non-zero values are useful for automated tests.

	scftorture.stat_interval= [KNL]
			The number of seconds between outputting the
			current test statistics to the console.  A value
			of zero disables statistics output.

	scftorture.stutter_cpus= [KNL]
			The number of jiffies to wait between each change
			to the set of CPUs under test.

	scftorture.use_cpus_read_lock= [KNL]
			Use use_cpus_read_lock() instead of the default
			preempt_disable() to disable CPU hotplug
			while invoking one of the smp_call_function*()
			functions.

	scftorture.verbose= [KNL]
			Enable additional printk() statements.

	scftorture.weight_single= [KNL]
			The probability weighting to use for the
			smp_call_function_single() function with a zero
			"wait" parameter.  A value of -1 selects the
			default if all other weights are -1.  However,
			if at least one weight has some other value, a
			value of -1 will instead select a weight of zero.

	scftorture.weight_single_wait= [KNL]
			The probability weighting to use for the
			smp_call_function_single() function with a
			non-zero "wait" parameter.  See weight_single.
```

```
	scftorture.weight_many= [KNL]
			The probability weighting to use for the
			smp_call_function_many() function with a zero
			"wait" parameter.  See weight_single.
			Note well that setting a high probability for
			this weighting can place serious IPI load
			on the system.

	scftorture.weight_many_wait= [KNL]
			The probability weighting to use for the
			smp_call_function_many() function with a
			non-zero "wait" parameter.  See weight_single
			and weight_many.

	scftorture.weight_all= [KNL]
			The probability weighting to use for the
			smp_call_function_all() function with a zero
			"wait" parameter.  See weight_single and
			weight_many.

	scftorture.weight_all_wait= [KNL]
			The probability weighting to use for the
			smp_call_function_all() function with a
			non-zero "wait" parameter.  See weight_single
			and weight_many.

	sdw_mclk_divider=[SDW]
			Specify the MCLK divider for Intel SoundWire buses in
			case the BIOS does not provide the clock rate properly.

	skew_tick=	[KNL,EARLY] Offset the periodic timer tick per cpu to mitig
			xtime_lock contention on larger systems, and/or RCU lock
			contention on all systems with CONFIG_MAXSMP set.
			Format: { "0" | "1" }
			0 -- disable. (may be 1 via CONFIG_CMDLINE="skew_tick=1"
			1 -- enable.
			Note: increases power consumption, thus should only be
			enabled if running jitter sensitive (HPC/RT) workloads.

	security=	[SECURITY] Choose a legacy "major" security module to
			enable at boot. This has been deprecated by the
			"lsm=" parameter.

	selinux=	[SELINUX] Disable or enable SELinux at boot time.
			Format: { "0" | "1" }
			See security/selinux/Kconfig help text.
			0 -- disable.
			1 -- enable.
			Default value is 1.

	serialnumber	[BUGS=X86-32]
```

```
	sev=option[,option...] [X86-64]

		debug
			Enable debug messages.

		nosnp
			Do not enable SEV-SNP (applies to host/hypervisor
			only). Setting 'nosnp' avoids the RMP check overhead
			in memory accesses when users do not want to run
			SEV-SNP guests.

	shapers=	[NET]
			Maximal number of shapers.

	show_lapic=	[APIC,X86] Advanced Programmable Interrupt Controller
			Limit apic dumping. The parameter defines the maximal
			number of local apics being dumped. Also it is possible
			to set it to "all" by meaning -- no limit here.
			Format: { 1 (default) | 2 | ... | all }.
			The parameter valid if only apic=debug or
			apic=verbose is specified.
			Example: apic=debug show_lapic=all

	slab_debug[=options[,slabs][;[options[,slabs]]...]	[MM]
			Enabling slab_debug allows one to determine the
			culprit if slab objects become corrupted. Enabling
			slab_debug can create guard zones around objects and
			may poison objects when not in use. Also tracks the
			last alloc / free. For more information see
			Documentation/admin-guide/mm/slab.rst.
			(slub_debug legacy name also accepted for now)

			Using this option implies the "no_hash_pointers"
			option which can be undone by adding the
			"hash_pointers=always" option.

	slab_max_order= [MM]
			Determines the maximum allowed order for slabs.
			A high setting may cause OOMs due to memory
			fragmentation. For more information see
			Documentation/admin-guide/mm/slab.rst.
			(slub_max_order legacy name also accepted for now)

	slab_merge	[MM]
			Enable merging of slabs with similar size when the
			kernel is built without CONFIG_SLAB_MERGE_DEFAULT.
			(slub_merge legacy name also accepted for now)

	slab_min_objects=	[MM]
			The minimum number of objects per slab. SLUB will
			increase the slab order up to slab_max_order to
			generate a sufficiently large slab able to contain
```

```
                the number of objects indicated. The higher the number
                of objects the smaller the overhead of tracking slabs
                and the less frequently locks need to be acquired.
                For more information see
                Documentation/admin-guide/mm/slab.rst.
                (slub_min_objects legacy name also accepted for now)

        slab_min_order= [MM]
                Determines the minimum page order for slabs. Must be
                lower or equal to slab_max_order. For more information see
                Documentation/admin-guide/mm/slab.rst.
                (slub_min_order legacy name also accepted for now)

        slab_nomerge    [MM]
                Disable merging of slabs with similar size. May be
                necessary if there is some reason to distinguish
                allocs to different slabs, especially in hardened
                environments where the risk of heap overflows and
                layout control by attackers can usually be
                frustrated by disabling merging. This will reduce
                most of the exposure of a heap attack to a single
                cache (risks via metadata attacks are mostly
                unchanged). Debug options disable merging on their
                own.
                For more information see
                Documentation/admin-guide/mm/slab.rst.
                (slub_nomerge legacy name also accepted for now)

        slab_strict_numa        [MM]
                Support memory policies on a per object level
                in the slab allocator. The default is for memory
                policies to be applied at the folio level when
                a new folio is needed or a partial folio is
                retrieved from the lists. Increases overhead
                in the slab fastpaths but gains more accurate
                NUMA kernel object placement which helps with slow
                interconnects in NUMA systems.

        slram=          [HW,MTD]

        smart2=         [HW]
                Format: <io1>[,<io2>[,...,<io8>]]

        smp.csd_lock_timeout= [KNL]
                Specify the period of time in milliseconds
                that smp_call_function() and friends will wait
                for a CPU to release the CSD lock.  This is
                useful when diagnosing bugs involving CPUs
                disabling interrupts for extended periods
                of time.  Defaults to 5,000 milliseconds, and
                setting a value of zero disables this feature.
                This feature may be more efficiently disabled
```

```
                using the csdlock_debug- kernel parameter.

    smp.panic_on_ipistall= [KNL]
                If a csd_lock_timeout extends for more than
                the specified number of milliseconds, panic the
                system.  By default, let CSD-lock acquisition
                take as long as they take.  Specifying 300,000
                for this value provides a 5-minute timeout.

    smsc-ircc2.nopnp        [HW] Don't use PNP to discover SMC devices
    smsc-ircc2.ircc_cfg=    [HW] Device configuration I/O port
    smsc-ircc2.ircc_sir=    [HW] SIR base I/O port
    smsc-ircc2.ircc_fir=    [HW] FIR base I/O port
    smsc-ircc2.ircc_irq=    [HW] IRQ line
    smsc-ircc2.ircc_dma=    [HW] DMA channel
    smsc-ircc2.ircc_transceiver= [HW] Transceiver type:
                            0: Toshiba Satellite 1800 (GP data pin select)
                            1: Fast pin select (default)
                            2: ATC IRMode

    smt=            [KNL,MIPS,S390,EARLY] Set the maximum number of threads
                (logical CPUs) to use per physical CPU on systems
                capable of symmetric multithreading (SMT). Will
                be capped to the actual hardware limit.
                Format: <integer>
                Default: -1 (no limit)

    softlockup_panic=
                [KNL] Should the soft-lockup detector generate panics.
                Format: <int>

                A value of non-zero instructs the soft-lockup detector
                to panic the machine when a soft-lockup duration exceeds
                N thresholds. It is also controlled by the kernel.softlocku
                sysctl and CONFIG_BOOTPARAM_SOFTLOCKUP_PANIC, which is the
                respective build-time switch to that functionality.

    softlockup_all_cpu_backtrace=
                [KNL] Should the soft-lockup detector generate
                backtraces on all cpus.
                Format: 0 | 1

    sonypi.*=       [HW] Sony Programmable I/O Control Device driver
                See Documentation/admin-guide/laptops/sonypi.rst

    spectre_bhi=    [X86] Control mitigation of Branch History Injection
                (BHI) vulnerability.  This setting affects the
                deployment of the HW BHI control and the SW BHB
                clearing sequence.

                on     - (default) Enable the HW or SW mitigation as
                         needed.  This protects the kernel from
```

```
                        both syscalls and VMs.
                vmexit - On systems which don't have the HW mitigation
                        available, enable the SW mitigation on vmexit
                        ONLY.  On such systems, the host kernel is
                        protected from VM-originated BHI attacks, but
                        may still be vulnerable to syscall attacks.
                off    - Disable the mitigation.

    spectre_v2=     [X86,EARLY] Control mitigation of Spectre variant 2
                (indirect branch speculation) vulnerability.
                The default operation protects the kernel from
                user space attacks.

                on   - unconditionally enable, implies
                       spectre_v2_user=on
                off  - unconditionally disable, implies
                       spectre_v2_user=off
                auto - kernel detects whether your CPU model is
                       vulnerable

                Selecting 'on' will, and 'auto' may, choose a
                mitigation method at run time according to the
                CPU, the available microcode, the setting of the
                CONFIG_MITIGATION_RETPOLINE configuration option,
                and the compiler with which the kernel was built.

                Selecting 'on' will also enable the mitigation
                against user space to user space task attacks.
                Selecting specific mitigation does not force enable
                user mitigations.

                Selecting 'off' will disable both the kernel and
                the user space protections.

                Specific mitigations can also be selected manually:

                retpoline         - replace indirect branches
                retpoline,generic - Retpolines
                retpoline,lfence  - LFENCE; indirect branch
                retpoline,amd     - alias for retpoline,lfence
                eibrs             - Enhanced/Auto IBRS
                eibrs,retpoline   - Enhanced/Auto IBRS + Retpolines
                eibrs,lfence      - Enhanced/Auto IBRS + LFENCE
                ibrs              - use IBRS to protect kernel

                Not specifying this option is equivalent to
                spectre_v2=auto.

    spectre_v2_user=
                [X86] Control mitigation of Spectre variant 2
                (indirect branch speculation) vulnerability between
                user space tasks
```

```
                on      - Unconditionally enable mitigations. Is
                          enforced by spectre_v2=on

                off     - Unconditionally disable mitigations. Is
                          enforced by spectre_v2=off

                prctl   - Indirect branch speculation is enabled,
                          but mitigation can be enabled via prctl
                          per thread.  The mitigation control state
                          is inherited on fork.

                prctl,ibpb
                        - Like "prctl" above, but only STIBP is
                          controlled per thread. IBPB is issued
                          always when switching between different user
                          space processes.

                seccomp
                        - Same as "prctl" above, but all seccomp
                          threads will enable the mitigation unless
                          they explicitly opt out.

                seccomp,ibpb
                        - Like "seccomp" above, but only STIBP is
                          controlled per thread. IBPB is issued
                          always when switching between different
                          user space processes.

                auto    - Kernel selects the mitigation depending on
                          the available CPU features and vulnerability.

                Default mitigation: "prctl"

                Not specifying this option is equivalent to
                spectre_v2_user=auto.

    spec_rstack_overflow=
                [X86,EARLY] Control RAS overflow mitigation on AMD Zen CPUs

                off             - Disable mitigation
                microcode       - Enable microcode mitigation only
                safe-ret        - Enable sw-only safe RET mitigation (defau
                ibpb            - Enable mitigation by issuing IBPB on
                                  kernel entry
                ibpb-vmexit     - Issue IBPB only on VMEXIT
                                  (cloud-specific mitigation)

    spec_store_bypass_disable=
                [HW,EARLY] Control Speculative Store Bypass (SSB) Disable m
                (Speculative Store Bypass vulnerability)
```

```
                Certain CPUs are vulnerable to an exploit against a
                a common industry wide performance optimization known
                as "Speculative Store Bypass" in which recent stores
                to the same memory location may not be observed by
                later loads during speculative execution. The idea
                is that such stores are unlikely and that they can
                be detected prior to instruction retirement at the
                end of a particular speculation execution window.

                In vulnerable processors, the speculatively forwarded
                store can be used in a cache side channel attack, for
                example to read memory to which the attacker does not
                directly have access (e.g. inside sandboxed code).

                This parameter controls whether the Speculative Store
                Bypass optimization is used.

                On x86 the options are:

                on      - Unconditionally disable Speculative Store Bypass
                off     - Unconditionally enable Speculative Store Bypass
                auto    - Kernel detects whether the CPU model contains an
                          implementation of Speculative Store Bypass and
                          picks the most appropriate mitigation. If the
                          CPU is not vulnerable, "off" is selected. If the
                          CPU is vulnerable the default mitigation is
                          architecture and Kconfig dependent. See below.
                prctl   - Control Speculative Store Bypass per thread
                          via prctl. Speculative Store Bypass is enabled
                          for a process by default. The state of the contro
                          is inherited on fork.
                seccomp - Same as "prctl" above, but all seccomp threads
                          will disable SSB unless they explicitly opt out.

                Default mitigations:
                X86:    "prctl"

                On powerpc the options are:

                on,auto - On Power8 and Power9 insert a store-forwarding
                          barrier on kernel entry and exit. On Power7
                          perform a software flush on kernel entry and
                          exit.
                off     - No action.

                Not specifying this option is equivalent to
                spec_store_bypass_disable=auto.

        split_llc=
                [X86,EARLY] Split the LLC N-ways

                When set, the LLC is split this many ways by matching
```

```
                'core_id % n'. This is setup before SMP bringup and
                used during SMP bringup before it knows the full
                topology. If your core count doesn't nicely divide by
                the number given, you get to keep the pieces.

                This is mostly a debug feature to emulate multiple LLCs
                on hardware that only have a single LLC.

    split_lock_detect=
                [X86] Enable split lock detection or bus lock detection

                When enabled (and if hardware support is present), atomic
                instructions that access data across cache line
                boundaries will result in an alignment check exception
                for split lock detection or a debug exception for
                bus lock detection.

                off     - not enabled

                warn    - the kernel will emit rate-limited warnings
                          about applications triggering the #AC
                          exception or the #DB exception. This mode is
                          the default on CPUs that support split lock
                          detection or bus lock detection. Default
                          behavior is by #AC if both features are
                          enabled in hardware.

                fatal   - the kernel will send SIGBUS to applications
                          that trigger the #AC exception or the #DB
                          exception. Default behavior is by #AC if
                          both features are enabled in hardware.

                ratelimit:N -
                          Set system wide rate limit to N bus locks
                          per second for bus lock detection.
                          0 < N <= 1000.

                          N/A for split lock detection.


                If an #AC exception is hit in the kernel or in
                firmware (i.e. not while executing in user mode)
                the kernel will oops in either "warn" or "fatal"
                mode.

                #DB exception for bus lock is triggered only when
                CPL > 0.

    srbds=      [X86,INTEL,EARLY]
                Control the Special Register Buffer Data Sampling
                (SRBDS) mitigation.
```

```
                Certain CPUs are vulnerable to an MDS-like
                exploit which can leak bits from the random
                number generator.

                By default, this issue is mitigated by
                microcode.  However, the microcode fix can cause
                the RDRAND and RDSEED instructions to become
                much slower.  Among other effects, this will
                result in reduced throughput from /dev/urandom.

                The microcode mitigation can be disabled with
                the following option:

                off:    Disable mitigation and remove
                        performance impact to RDRAND and RDSEED

    srcutree.big_cpu_lim [KNL]
                Specifies the number of CPUs constituting a
                large system, such that srcu_struct structures
                should immediately allocate an srcu_node array.
                This kernel-boot parameter defaults to 128,
                but takes effect only when the low-order four
                bits of srcutree.convert_to_big is equal to 3
                (decide at boot).

    srcutree.convert_to_big [KNL]
                Specifies under what conditions an SRCU tree
                srcu_struct structure will be converted to big
                form, that is, with an rcu_node tree:

                          0:  Never.
                          1:  At init_srcu_struct() time.
                          2:  When rcutorture decides to.
                          3:  Decide at boot time (default).
                       0x1X:  Above plus if high contention.

                Either way, the srcu_node tree will be sized based
                on the actual runtime number of CPUs (nr_cpu_ids)
                instead of the compile-time CONFIG_NR_CPUS.

    srcutree.counter_wrap_check [KNL]
                Specifies how frequently to check for
                grace-period sequence counter wrap for the
                srcu_data structure's ->srcu_gp_seq_needed field.
                The greater the number of bits set in this kernel
                parameter, the less frequently counter wrap will
                be checked for.  Note that the bottom two bits
                are ignored.

    srcutree.exp_holdoff [KNL]
                Specifies how many nanoseconds must elapse
                since the end of the last SRCU grace period for
```

```
                a given srcu_struct until the next normal SRCU
                grace period will be considered for automatic
                expediting.  Set to zero to disable automatic
                expediting.

    srcutree.srcu_max_nodelay [KNL]
                Specifies the number of no-delay instances
                per jiffy for which the SRCU grace period
                worker thread will be rescheduled with zero
                delay. Beyond this limit, worker thread will
                be rescheduled with a sleep delay of one jiffy.

    srcutree.srcu_max_nodelay_phase [KNL]
                Specifies the per-grace-period phase, number of
                non-sleeping polls of readers. Beyond this limit,
                grace period worker thread will be rescheduled
                with a sleep delay of one jiffy, between each
                rescan of the readers, for a grace period phase.

    srcutree.srcu_retry_check_delay [KNL]
                Specifies number of microseconds of non-sleeping
                delay between each non-sleeping poll of readers.

    srcutree.small_contention_lim [KNL]
                Specifies the number of update-side contention
                events per jiffy will be tolerated before
                initiating a conversion of an srcu_struct
                structure to big form.  Note that the value of
                srcutree.convert_to_big must have the 0x10 bit
                set for contention-based conversions to occur.

    ssbd=       [ARM64,HW,EARLY]
                Speculative Store Bypass Disable control

                On CPUs that are vulnerable to the Speculative
                Store Bypass vulnerability and offer a
                firmware based mitigation, this parameter
                indicates how the mitigation should be used:

                force-on:  Unconditionally enable mitigation for
                           for both kernel and userspace
                force-off: Unconditionally disable mitigation for
                           for both kernel and userspace
                kernel:    Always enable mitigation in the
                           kernel, and offer a prctl interface
                           to allow userspace to register its
                           interest in being mitigated too.

    stack_guard_gap=        [MM]
                override the default stack gap protection. The value
                is in page units and it defines how many pages prior
                to (for stacks growing down) resp. after (for stacks
```

```
                growing up) the main stack are reserved for no other
                mapping. Default value is 256 pages.

    stack_depot_disable= [KNL,EARLY]
                Setting this to true through kernel command line will
                disable the stack depot thereby saving the static memory
                consumed by the stack hash table. By default this is set
                to false.

    stack_depot_max_pools= [KNL,EARLY]
                Specify the maximum number of pools to use for storing
                stack traces. Pools are allocated on-demand up to this
                limit. Default value is 8191 pools.

    stacktrace  [FTRACE]
                Enable the stack tracer on boot up.

    stacktrace_filter=[function-list]
                [FTRACE] Limit the functions that the stack tracer
                will trace at boot up. function-list is a comma-separated
                list of functions. This list can be changed at run
                time by the stack_trace_filter file in the debugfs
                tracing directory. Note, this enables stack tracing
                and the stacktrace above is not needed.

    sti=        [PARISC,HW]
                Format: <num>
                Set the STI (builtin display/keyboard on the HP-PARISC
                machines) console (graphic card) which should be used
                as the initial boot-console.
                See also comment in drivers/video/sticore.c.

    sti_font=   [HW]
                See comment in drivers/video/sticore.c.

    stifb=      [HW]
                Format: bpp:<bpp1>[:<bpp2>[:<bpp3>...]]

    strict_sas_size=
                [X86]
                Format: <bool>
                Enable or disable strict sigaltstack size checks
                against the required signal frame size which
                depends on the supported FPU features. This can
                be used to filter out binaries which have
                not yet been made aware of AT_MINSIGSTKSZ.

    stress_hpt  [PPC,EARLY]
                Limits the number of kernel HPT entries in the hash
                page table to increase the rate of hash page table
                faults on kernel addresses.
```

```
	stress_slb	[PPC,EARLY]
			Limits the number of kernel SLB entries, and flushes
			them frequently to increase the rate of SLB faults
			on kernel addresses.

	no_slb_preload	[PPC,EARLY]
			Disables slb preloading for userspace.

	sunrpc.min_resvport=
	sunrpc.max_resvport=
			[NFS,SUNRPC]
			SunRPC servers often require that client requests
			originate from a privileged port (i.e. a port in the
			range 0 < portnr < 1024).
			An administrator who wishes to reserve some of these
			ports for other uses may adjust the range that the
			kernel's sunrpc client considers to be privileged
			using these two parameters to set the minimum and
			maximum port values.

	sunrpc.svc_rpc_per_connection_limit=
			[NFS,SUNRPC]
			Limit the number of requests that the server will
			process in parallel from a single connection.
			The default value is 0 (no limit).

	sunrpc.pool_mode=
			[NFS]
			Control how the NFS server code allocates CPUs to
			service thread pools.  Depending on how many NICs
			you have and where their interrupts are bound, this
			option will affect which CPUs will do NFS serving.
			Note: this parameter cannot be changed while the
			NFS server is running.

			auto		the server chooses an appropriate mode
					automatically using heuristics
			global		a single global pool contains all CPUs
			percpu		one pool for each CPU
			pernode		one pool for each NUMA node (equivalent
					to global on non-NUMA machines)

	sunrpc.tcp_slot_table_entries=
	sunrpc.udp_slot_table_entries=
			[NFS,SUNRPC]
			Sets the upper limit on the number of simultaneous
			RPC calls that can be sent from the client to a
			server. Increasing these values may allow you to
			improve throughput, but will also increase the
			amount of memory reserved for use by the client.

	suspend.pm_test_delay=
```

```
                [SUSPEND]
                Sets the number of seconds to remain in a suspend test
                mode before resuming the system (see
                /sys/power/pm_test). Only available when CONFIG_PM_DEBUG
                is set. Default value is 5.

svm=            [PPC]
                Format: { on | off | y | n | 1 | 0 }
                This parameter controls use of the Protected
                Execution Facility on pSeries.

swiotlb=        [ARM,PPC,MIPS,X86,S390,EARLY]
                Format: { <int> [,<int>] | force | noforce }
                <int> -- Number of I/O TLB slabs
                <int> -- Second integer after comma. Number of swiotlb
                         areas with their own lock. Will be rounded up
                         to a power of 2.
                force -- force using of bounce buffers even if they
                         wouldn't be automatically used by the kernel
                noforce -- Never use bounce buffers (for debugging)

switches=       [HW,M68k,EARLY]

sysctl.*=       [KNL]
                Set a sysctl parameter, right before loading the init
                process, as if the value was written to the respective
                /proc/sys/... file. Both '.' and '/' are recognized as
                separators. Unrecognized parameters and invalid values
                are reported in the kernel log. Sysctls registered
                later by a loaded module cannot be set this way.
                Example: sysctl.vm.swappiness=40

sysrq_always_enabled
                [KNL]
                Ignore sysrq setting - this boot parameter will
                neutralize any effect of /proc/sys/kernel/sysrq.
                Useful for debugging.

tcpmhash_entries= [KNL,NET]
                Set the number of tcp_metrics_hash slots.
                Default value is 8192 or 16384 depending on total
                ram pages. This is used to specify the TCP metrics
                cache size. See Documentation/networking/ip-sysctl.rst
                "tcp_no_metrics_save" section for more details.

tdfx=           [HW,DRM]

test_suspend=   [SUSPEND]
                Format: { "mem" | "standby" | "freeze" }[,N]
                Specify "mem" (for Suspend-to-RAM) or "standby" (for
                standby suspend) or "freeze" (for suspend type freeze)
                as the system sleep state during system startup with
```

```
                the optional capability to repeat N number of times.
                The system is woken from this state using a
                wakeup-capable RTC alarm.

    thash_entries=  [KNL,NET]
                Set number of hash buckets for TCP connection

    thermal.act=    [HW,ACPI]
                -1: disable all active trip points in all thermal zones
                <degrees C>: override all lowest active trip points

    thermal.crt=    [HW,ACPI]
                -1: disable all critical trip points in all thermal zones
                <degrees C>: override all critical trip points

    thermal.off=    [HW,ACPI]
                1: disable ACPI thermal control

    thermal.psv=    [HW,ACPI]
                -1: disable all passive trip points
                <degrees C>: override all passive trip points to this
                value

    thermal.tzp=    [HW,ACPI]
                Specify global default ACPI thermal zone polling rate
                <deci-seconds>: poll all this frequency
                0: no polling (default)

    thp_anon=       [KNL]
                Format: <size>[KMG],<size>[KMG]:<state>;<size>[KMG]-<size>[
                state is one of "always", "madvise", "never" or "inherit".
                Control the default behavior of the system with respect
                to anonymous transparent hugepages.
                Can be used multiple times for multiple anon THP sizes.
                See Documentation/admin-guide/mm/transhuge.rst for more
                details.

    threadirqs      [KNL,EARLY]
                Force threading of all interrupt handlers except those
                marked explicitly IRQF_NO_THREAD.

    thp_shmem=      [KNL]
                Format: <size>[KMG],<size>[KMG]:<policy>;<size>[KMG]-<size>
                Control the default policy of each hugepage size for the
                internal shmem mount. <policy> is one of policies available
                for the shmem mount ("always", "inherit", "never", "within_
                and "advise").
                It can be used multiple times for multiple shmem THP sizes.
                See Documentation/admin-guide/mm/transhuge.rst for more
                details.

    topology=       [S390,EARLY]
```

```
                Format: {off | on}
                Specify if the kernel should make use of the cpu
                topology information if the hardware supports this.
                The scheduler will make use of this information and
                e.g. base its process migration decisions on it.
                Default is on.

    torture.disable_onoff_at_boot= [KNL]
                Prevent the CPU-hotplug component of torturing
                until after init has spawned.

    torture.ftrace_dump_at_shutdown= [KNL]
                Dump the ftrace buffer at torture-test shutdown,
                even if there were no errors.  This can be a
                very costly operation when many torture tests
                are running concurrently, especially on systems
                with rotating-rust storage.

    torture.verbose_sleep_frequency= [KNL]
                Specifies how many verbose printk()s should be
                emitted between each sleep.  The default of zero
                disables verbose-printk() sleeping.

    torture.verbose_sleep_duration= [KNL]
                Duration of each verbose-printk() sleep in jiffies.

    tpm.disable_pcr_integrity= [HW,TPM]
                Do not protect PCR registers from unintended physical
                access, or interposers in the bus by the means of
                having an integrity protected session wrapped around
                TPM2_PCR_Extend command. Consider this in a situation
                where TPM is heavily utilized by IMA, thus protection
                causing a major performance hit, and the space where
                machines are deployed is by other means guarded.

    tpm_crb_ffa.busy_timeout_ms= [ARM64,TPM]
                Maximum time in milliseconds to retry sending a message
                to the TPM service before giving up. This parameter control
                how long the system will continue retrying when the TPM
                service is busy.
                Format: <unsigned int>
                Default: 2000 (2 seconds)

    tpm_suspend_pcr=[HW,TPM]
                Format: integer pcr id
                Specify that at suspend time, the tpm driver
                should extend the specified pcr with zeros,
                as a workaround for some chips which fail to
                flush the last written pcr on TPM_SaveState.
                This will guarantee that all the other pcrs
                are saved.
```

```
	tpm_tis.interrupts= [HW,TPM]
			Enable interrupts for the MMIO based physical layer
			for the FIFO interface. By default it is set to false
			(0). For more information about TPM hardware interfaces
			defined by Trusted Computing Group (TCG) see
			https://trustedcomputinggroup.org/resource/pc-client-platfo

	tp_printk	[FTRACE]
			Have the tracepoints sent to printk as well as the
			tracing ring buffer. This is useful for early boot up
			where the system hangs or reboots and does not give the
			option for reading the tracing buffer or performing a
			ftrace_dump_on_oops.

			To turn off having tracepoints sent to printk,
			 echo 0 > /proc/sys/kernel/tracepoint_printk
			Note, echoing 1 into this file without the
			tp_printk kernel cmdline option has no effect.

			The tp_printk_stop_on_boot (see below) can also be used
			to stop the printing of events to console at
			late_initcall_sync.

			** CAUTION **

			Having tracepoints sent to printk() and activating high
			frequency tracepoints such as irq or sched, can cause
			the system to live lock.

	tp_printk_stop_on_boot [FTRACE]
			When tp_printk (above) is set, it can cause a lot of noise
			on the console. It may be useful to only include the
			printing of events during boot up, as user space may
			make the system inoperable.

			This command line option will stop the printing of events
			to console at the late_initcall_sync() time frame.

	trace_buf_size=nn[KMG]
			[FTRACE] will set tracing buffer size on each cpu.

	trace_clock=	[FTRACE] Set the clock used for tracing events
			at boot up.
			local - Use the per CPU time stamp counter
				(converted into nanoseconds). Fast, but
				depending on the architecture, may not be
				in sync between CPUs.
			global - Event time stamps are synchronized across
				CPUs. May be slower than the local clock,
				but better for some race conditions.
			counter - Simple counting of events (1, 2, ..)
				note, some counts may be skipped due to the
```

```
                        infrastructure grabbing the clock more than
                        once per event.
                uptime - Use jiffies as the time stamp.
                perf - Use the same clock that perf uses.
                mono - Use ktime_get_mono_fast_ns() for time stamps.
                mono_raw - Use ktime_get_raw_fast_ns() for time
                        stamps.
                boot - Use ktime_get_boot_fast_ns() for time stamps.
                Architectures may add more clocks. See
                Documentation/trace/ftrace.rst for more details.

    trace_event=[event-list]
                [FTRACE] Set and start specified trace events in order
                to facilitate early boot debugging. The event-list is a
                comma-separated list of trace events to enable. See
                also Documentation/trace/events.rst

                To enable modules, use :mod: keyword:

                trace_event=:mod:<module>

                The value before :mod: will only enable specific events
                that are part of the module. See the above mentioned
                document for more information.

    trace_instance=[instance-info]
                [FTRACE] Create a ring buffer instance early in boot up.
                This will be listed in:

                        /sys/kernel/tracing/instances

                Events can be enabled at the time the instance is created
                via:

                        trace_instance=<name>,<system1>:<event1>,<system2>:

                Note, the "<system*>:" portion is optional if the event is
                unique.

                        trace_instance=foo,sched:sched_switch,irq_handler_e

                will enable the "sched_switch" event (note, the "sched:" is
                the same thing would happen if it was left off). The irq_ha
                event, and all events under the "initcall" system.

                Flags can be added to the instance to modify its behavior w
                created. The flags are separated by '^'.

                The available flags are:

                    traceoff    - Have the tracing instance tracing disable
                    traceprintk - Have trace_printk() write into this trace
```

```
                                        (note, "printk" and "trace_printk" can al

                        trace_instance=foo^traceoff^traceprintk,sched,irq

                    The flags must come before the defined events.

                    If memory has been reserved (see memmap for x86), the insta
                    can use that memory:

                            memmap=12M$0x284500000 trace_instance=boot_map@0x28

                    The above will create a "boot_map" instance that uses the p
                    memory at 0x284500000 that is 12Megs. The per CPU buffers o
                    instance will be split up accordingly.

                    Alternatively, the memory can be reserved by the reserve_me

                            reserve_mem=12M:4096:trace trace_instance=boot_map@

                    This will reserve 12 megabytes at boot up with a 4096 byte
                    and place the ring buffer in this memory. Note that due to
                    memory may not be the same location each time, which will n
                    the buffer content.

                    Also note that the layout of the ring buffer data may chang
                    kernel versions where the validator will fail and reset the
                    if the layout is not the same as the previous kernel.

                    If the ring buffer is used for persistent bootups and has e
                    it is recommend to disable tracing so that events from a pr
                    mix with events of the current boot (unless you are debuggi
                    at boot up).

                            reserve_mem=12M:4096:trace trace_instance=boot_map^

                    Note, saving the trace buffer across reboots does require t
                    is set up to not wipe memory. For instance, CONFIG_RESET_AT
                    can force a memory reset on boot which will clear any trace
                    This is just one of many ways that can clear memory. Make s
                    keeps the content of memory across reboots before relying o

                    NB: Both the mapped address and size must be page aligned f

                    See also Documentation/trace/debugging.rst


    trace_options=[option-list]
                    [FTRACE] Enable or disable tracer options at boot.
                    The option-list is a comma delimited list of options
                    that can be enabled or disabled just as if you were
                    to echo the option name into
```

```
                        /sys/kernel/tracing/trace_options

                    For example, to enable stacktrace option (to dump the
                    stack trace of each event), add to the command line:

                          trace_options=stacktrace

                    See also Documentation/trace/ftrace.rst "trace options"
                    section.

        trace_trigger=[trigger-list]
                    [FTRACE] Add an event trigger on specific events.
                    Set a trigger on top of a specific event, with an optional
                    filter.

                    The format is "trace_trigger=<event>.<trigger>[ if <filter>
                    Where more than one trigger may be specified that are comma

                    For example:

                      trace_trigger="sched_switch.stacktrace if prev_state == 2

                    The above will enable the "stacktrace" trigger on the "sche
                    event but only trigger it if the "prev_state" of the "sched
                    event is "2" (TASK_UNINTERRUPTIBLE).

                    See also "Event triggers" in Documentation/trace/events.rst


        traceoff_after_boot
                    [FTRACE] Sometimes tracing is used to debug issues
                    during the boot process. Since the trace buffer has a
                    limited amount of storage, it may be prudent to
                    disable tracing after the boot is finished, otherwise
                    the critical information may be overwritten.  With this
                    option, the main tracing buffer will be turned off at
                    the end of the boot process.

        traceoff_on_warning
                    [FTRACE] enable this option to disable tracing when a
                    warning is hit. This turns off "tracing_on". Tracing can
                    be enabled again by echoing '1' into the "tracing_on"
                    file located in /sys/kernel/tracing/

                    This option is useful, as it disables the trace before
                    the WARNING dump is called, which prevents the trace to
                    be filled with content caused by the warning output.

                    This option can also be set at run time via the sysctl
                    option:  kernel/traceoff_on_warning

        transparent_hugepage=
```

```
                        [KNL]
                        Format: [always|madvise|never]
                        Can be used to control the default behavior of the system
                        with respect to transparent hugepages.
                        See Documentation/admin-guide/mm/transhuge.rst
                        for more details.

        transparent_hugepage_shmem= [KNL]
                        Format: [always|within_size|advise|never|deny|force]
                        Can be used to control the hugepage allocation policy for
                        the internal shmem mount.
                        See Documentation/admin-guide/mm/transhuge.rst
                        for more details.

        transparent_hugepage_tmpfs= [KNL]
                        Format: [always|within_size|advise|never]
                        Can be used to control the default hugepage allocation poli
                        for the tmpfs mount.
                        See Documentation/admin-guide/mm/transhuge.rst
                        for more details.

        trusted.source= [KEYS]
                        Format: <string>
                        This parameter identifies the trust source as a backend
                        for trusted keys implementation. Supported trust
                        sources:
                        - "tpm"
                        - "tee"
                        - "caam"
                        - "dcp"
                        - "pkwm"
                        If not specified then it defaults to iterating through
                        the trust source list starting with TPM and assigns the
                        first trust source as a backend which is initialized
                        successfully during iteration.

        trusted.debug=  [KEYS]
                        Format: <bool>
                        Enable trusted keys debug traces at runtime when
                        CONFIG_TRUSTED_KEYS_DEBUG=y.

                        To make the traces visible after enabling the option,
                        use trusted.dyndbg='+p' as needed. By convention,
                        the subsystem uses pr_debug() for these traces.

                        SAFETY: The traces can leak sensitive data, so be
                        cautious before enabling this. They remain inactive
                        unless this parameter is set this option to  a true
                        value.

                        Default: false
```

```
        trusted.rng=    [KEYS]
                        Format: <string>
                        The RNG used to generate key material for trusted keys.
                        Can be one of:
                        - "kernel"
                        - the same value as trusted.source: "tpm" or "tee"
                        - "default"
                        If not specified, "default" is used. In this case,
                        the RNG's choice is left to each individual trust source.

        trusted.dcp_use_otp_key
                        This is intended to be used in combination with
                        trusted.source=dcp and will select the DCP OTP key
                        instead of the DCP UNIQUE key blob encryption.

        trusted.dcp_skip_zk_test
                        This is intended to be used in combination with
                        trusted.source=dcp and will disable the check if the
                        blob key is all zeros. This is helpful for situations where
                        having this key zero'ed is acceptable. E.g. in testing
                        scenarios.

        tsa=            [X86] Control mitigation for Transient Scheduler
                        Attacks on AMD CPUs. Search the following in your
                        favourite search engine for more details:

                        "Technical guidance for mitigating transient scheduler
                        attacks".

                        off             - disable the mitigation
                        on              - enable the mitigation (default)
                        user            - mitigate only user/kernel transitions
                        vm              - mitigate only guest/host transitions


        tsc=            Disable clocksource stability checks for TSC.
                        Format: <string>
                        [x86] reliable: mark tsc clocksource as reliable, this
                        disables clocksource verification at runtime, as well
                        as the stability checks done at bootup. Used to enable
                        high-resolution timer mode on older hardware, and in
                        virtualized environment.
                        [x86] noirqtime: Do not use TSC to do irq accounting.
                        Used to run time disable IRQ_TIME_ACCOUNTING on any
                        platforms where RDTSC is slow and this accounting
                        can add overhead.
                        [x86] unstable: mark the TSC clocksource as unstable, this
                        marks the TSC unconditionally unstable at bootup and
                        avoids any further wobbles once the TSC watchdog notices.
                        [x86] nowatchdog: disable clocksource watchdog. Used
                        in situations with strict latency requirements (where
                        interruptions from clocksource watchdog are not
```

```
                  acceptable).
                  [x86] recalibrate: force recalibration against a HW timer
                  (HPET or PM timer) on systems whose TSC frequency was
                  obtained from HW or FW using either an MSR or CPUID(0x15).
                  Warn if the difference is more than 500 ppm.
                  [x86] watchdog: Enforce the clocksource watchdog on TSC

      tsc_early_khz=  [X86,EARLY] Skip early TSC calibration and use the given
                  value instead. Useful when the early TSC frequency discover
                  procedure is not reliable, such as on overclocked systems
                  with CPUID.16h support and partial CPUID.15h support.
                  Format: <unsigned int>

      tsx=        [X86] Control Transactional Synchronization
                  Extensions (TSX) feature in Intel processors that
                  support TSX control.

                  This parameter controls the TSX feature. The options are:

                  on      - Enable TSX on the system. Although there are
                         mitigations for all known security vulnerabilities,
                         TSX has been known to be an accelerator for
                         several previous speculation-related CVEs, and
                         so there may be unknown security risks associated
                         with leaving it enabled.

                  off     - Disable TSX on the system. (Note that this
                         option takes effect only on newer CPUs which are
                         not vulnerable to MDS, i.e., have
                         MSR_IA32_ARCH_CAPABILITIES.MDS_NO=1 and which get
                         the new IA32_TSX_CTRL MSR through a microcode
                         update. This new MSR allows for the reliable
                         deactivation of the TSX functionality.)

                  auto    - Disable TSX if X86_BUG_TAA is present,
                           otherwise enable TSX on the system.

                  Not specifying this option is equivalent to tsx=off.

                  See Documentation/admin-guide/hw-vuln/tsx_async_abort.rst
                  for more details.

      tsx_async_abort= [X86,INTEL,EARLY] Control mitigation for the TSX Async
                  Abort (TAA) vulnerability.

                  Similar to Micro-architectural Data Sampling (MDS)
                  certain CPUs that support Transactional
                  Synchronization Extensions (TSX) are vulnerable to an
                  exploit against CPU internal buffers which can forward
                  information to a disclosure gadget under certain
                  conditions.
```

```
                In vulnerable processors, the speculatively forwarded
                data can be used in a cache side channel attack, to
                access data to which the attacker does not have direct
                access.

                This parameter controls the TAA mitigation.  The
                options are:

                full       - Enable TAA mitigation on vulnerable CPUs
                             if TSX is enabled.

                full,nosmt - Enable TAA mitigation and disable SMT on
                             vulnerable CPUs. If TSX is disabled, SMT
                             is not disabled because CPU is not
                             vulnerable to cross-thread TAA attacks.
                off        - Unconditionally disable TAA mitigation

                On MDS-affected machines, tsx_async_abort=off can be
                prevented by an active MDS mitigation as both vulnerabiliti
                are mitigated with the same mechanism so in order to disabl
                this mitigation, you need to specify mds=off too.

                Not specifying this option is equivalent to
                tsx_async_abort=full.  On CPUs which are MDS affected
                and deploy MDS mitigation, TAA mitigation is not
                required and doesn't provide any additional
                mitigation.

                For details see:
                Documentation/admin-guide/hw-vuln/tsx_async_abort.rst

    turbografx.map[2|3]=    [HW,JOY]
                TurboGraFX parallel port interface
                Format:
                <port#>,<js1>,<js2>,<js3>,<js4>,<js5>,<js6>,<js7>
                See also Documentation/input/devices/joystick-parport.rst

    udbg-immortal   [PPC] When debugging early kernel crashes that
                happen after console_init() and before a proper
                console driver takes over, this boot options might
                help "seeing" what's going on.

    uhash_entries=  [KNL,NET]
                Set number of hash buckets for UDP/UDP-Lite connections

    uhci-hcd.ignore_oc=
                [USB] Ignore overcurrent events (default N).
                Some badly-designed motherboards generate lots of
                bogus events, for ports that aren't wired to
                anything.  Set this parameter to avoid log spamming.
                Note that genuine overcurrent events won't be
                reported either.
```

```
unaligned_scalar_speed=
                [RISCV]
                Format: {slow | fast | unsupported}
                Allow skipping scalar unaligned access speed tests. This
                is useful for testing alternative code paths and to skip
                the tests in environments where they run too slowly. All
                CPUs must have the same scalar unaligned access speed.

unaligned_vector_speed=
                [RISCV]
                Format: {slow | fast | unsupported}
                Allow skipping vector unaligned access speed tests. This
                is useful for testing alternative code paths and to skip
                the tests in environments where they run too slowly. All
                CPUs must have the same vector unaligned access speed.

unknown_nmi_panic
                [X86] Cause panic on unknown NMI.

unwind_debug    [X86-64,EARLY]
                Enable unwinder debug output.  This can be
                useful for debugging certain unwinder error
                conditions, including corrupt stacks and
                bad/missing unwinder metadata.

usbcore.authorized_default=
                [USB] Default USB device authorization:
                (default -1 = authorized (same as 1),
                0 = not authorized, 1 = authorized, 2 = authorized
                if device connected to internal port)

usbcore.autosuspend=
                [USB] The autosuspend time delay (in seconds) used
                for newly-detected USB devices (default 2).  This
                is the time required before an idle device will be
                autosuspended.  Devices for which the delay is set
                to a negative value won't be autosuspended at all.

usbcore.usbfs_snoop=
                [USB] Set to log all usbfs traffic (default 0 = off).

usbcore.usbfs_snoop_max=
                [USB] Maximum number of bytes to snoop in each URB
                (default = 65536).

usbcore.blinkenlights=
                [USB] Set to cycle leds on hubs (default 0 = off).

usbcore.old_scheme_first=
                [USB] Start with the old device initialization
                scheme (default 0 = off).
```

```
usbcore.usbfs_memory_mb=
                [USB] Memory limit (in MB) for buffers allocated by
                usbfs (default = 16, 0 = max = 2047).

usbcore.use_both_schemes=
                [USB] Try the other device initialization scheme
                if the first one fails (default 1 = enabled).

usbcore.initial_descriptor_timeout=
                [USB] Specifies timeout for the initial 64-byte
                USB_REQ_GET_DESCRIPTOR request in milliseconds
                (default 5000 = 5.0 seconds).

usbcore.nousb   [USB] Disable the USB subsystem

usbcore.quirks=
                [USB] A list of quirk entries to augment the built-in
                usb core quirk list. List entries are separated by
                commas. Each entry has the form
                VendorID:ProductID:Flags. The IDs are 4-digit hex
                numbers and Flags is a set of letters. Each letter
                will change the built-in quirk; setting it if it is
                clear and clearing it if it is set. The letters have
                the following meanings:
                        a = USB_QUIRK_STRING_FETCH_255 (string
                                descriptors must not be fetched using
                                a 255-byte read);
                        b = USB_QUIRK_RESET_RESUME (device can't resume
                                correctly so reset it instead);
                        c = USB_QUIRK_NO_SET_INTF (device can't handle
                                Set-Interface requests);
                        d = USB_QUIRK_CONFIG_INTF_STRINGS (device can't
                                handle its Configuration or Interface
                                strings);
                        e = USB_QUIRK_RESET (device can't be reset
                                (e.g morph devices), don't use reset);
                        f = USB_QUIRK_HONOR_BNUMINTERFACES (device has
                                more interface descriptions than the
                                bNumInterfaces count, and can't handle
                                talking to these interfaces);
                        g = USB_QUIRK_DELAY_INIT (device needs a pause
                                during initialization, after we read
                                the device descriptor);
                        h = USB_QUIRK_LINEAR_UFRAME_INTR_BINTERVAL (For
                                high speed and super speed interrupt
                                endpoints, the USB 2.0 and USB 3.0 spec
                                require the interval in microframes (1
                                microframe = 125 microseconds) to be
                                calculated as interval = 2 ^
                                (bInterval-1).
                                Devices with this quirk report their
```

```
                                bInterval as the result of this
                                calculation instead of the exponent
                                variable used in the calculation);
                        i = USB_QUIRK_DEVICE_QUALIFIER (device can't
                                handle device_qualifier descriptor
                                requests);
                        j = USB_QUIRK_IGNORE_REMOTE_WAKEUP (device
                                generates spurious wakeup, ignore
                                remote wakeup capability);
                        k = USB_QUIRK_NO_LPM (device can't handle Link
                                Power Management);
                        l = USB_QUIRK_LINEAR_FRAME_INTR_BINTERVAL
                                (Device reports its bInterval as linear
                                frames instead of the USB 2.0
                                calculation);
                        m = USB_QUIRK_DISCONNECT_SUSPEND (Device needs
                                to be disconnected before suspend to
                                prevent spurious wakeup);
                        n = USB_QUIRK_DELAY_CTRL_MSG (Device needs a
                                pause after every control message);
                        o = USB_QUIRK_HUB_SLOW_RESET (Hub needs extra
                                delay after resetting its port);
                        p = USB_QUIRK_SHORT_SET_ADDRESS_REQ_TIMEOUT
                                (Reduce timeout of the SET_ADDRESS
                                request from 5000 ms to 500 ms);
                        q = USB_QUIRK_FORCE_ONE_CONFIG (Device
                                claims zero configurations,
                                forcing to 1);
                        r = USB_QUIRK_WINDOWS_CONFIG_REQ_SIZE (Device
                                fails during initialization when asked for
                                9-bytes configuration descriptor request.
                                Ask for 255-bytes request instead to mirror
                                Windows' behavior);
                Example: quirks=0781:5580:bk,0a5c:5834:gij

usbhid.mousepoll=
                [USBHID] The interval which mice are to be polled at.

usbhid.jspoll=
                [USBHID] The interval which joysticks are to be polled at.

usbhid.kbpoll=
                [USBHID] The interval which keyboards are to be polled at.

usb-storage.delay_use=
                [UMS] The delay in seconds before a new device is
                scanned for Logical Units (default 1).
                Optionally the delay in milliseconds if the value has
                suffix with "ms".
                Example: delay_use=2567ms

usb-storage.quirks=
```

```
                [UMS] A list of quirks entries to supplement or
                override the built-in unusual_devs list.  List
                entries are separated by commas.  Each entry has
                the form VID:PID:Flags where VID and PID are Vendor
                and Product ID values (4-digit hex numbers) and
                Flags is a set of characters, each corresponding
                to a common usb-storage quirk flag as follows:
                        a = SANE_SENSE (collect more than 18 bytes
                                of sense data, not on uas);
                        b = BAD_SENSE (don't collect more than 18
                                bytes of sense data, not on uas);
                        c = FIX_CAPACITY (decrease the reported
                                device capacity by one sector);
                        d = NO_READ_DISC_INFO (don't use
                                READ_DISC_INFO command, not on uas);
                        e = NO_READ_CAPACITY_16 (don't use
                                READ_CAPACITY_16 command);
                        f = NO_REPORT_OPCODES (don't use report opcodes
                                command, uas only);
                        g = MAX_SECTORS_240 (don't transfer more than
                                240 sectors at a time, uas only);
                        h = CAPACITY_HEURISTICS (decrease the
                                reported device capacity by one
                                sector if the number is odd);
                        i = IGNORE_DEVICE (don't bind to this
                                device);
                        j = NO_REPORT_LUNS (don't use report luns
                                command, uas only);
                        k = NO_SAME (do not use WRITE_SAME, uas only)
                        l = NOT_LOCKABLE (don't try to lock and
                                unlock ejectable media, not on uas);
                        m = MAX_SECTORS_64 (don't transfer more
                                than 64 sectors = 32 KB at a time,
                                not on uas);
                        n = INITIAL_READ10 (force a retry of the
                                initial READ(10) command, not on uas);
                        o = CAPACITY_OK (accept the capacity
                                reported by the device, not on uas);
                        p = WRITE_CACHE (the device cache is ON
                                by default, not on uas);
                        r = IGNORE_RESIDUE (the device reports
                                bogus residue values, not on uas);
                        s = SINGLE_LUN (the device has only one
                                Logical Unit);
                        t = NO_ATA_1X (don't allow ATA(12) and ATA(16)
                                commands, uas only);
                        u = IGNORE_UAS (don't bind to the uas driver);
                        w = NO_WP_DETECT (don't test whether the
                                medium is write-protected).
                        y = ALWAYS_SYNC (issue a SYNCHRONIZE_CACHE
                                even if the device claims no cache,
                                not on uas)
```

```
                Example: quirks=0419:aaf5:rl,0421:0433:rc

    user_debug=     [KNL,ARM]
                Format: <int>
                See arch/arm/Kconfig.debug help text.
                         1 - undefined instruction events
                         2 - system calls
                         4 - invalid data aborts
                         8 - SIGSEGV faults
                        16 - SIGBUS faults
                Example: user_debug=31

    vdso=           [X86,SH,SPARC]
                On X86_32, this is an alias for vdso32=.  Otherwise:

                vdso=1: enable VDSO (the default)
                vdso=0: disable VDSO mapping

    vdso32=         [X86] Control the 32-bit vDSO
                vdso32=1: enable 32-bit VDSO
                vdso32=0 or vdso32=2: disable 32-bit VDSO

                See the help text for CONFIG_COMPAT_VDSO for more
                details.  If CONFIG_COMPAT_VDSO is set, the default is
                vdso32=0; otherwise, the default is vdso32=1.

                For compatibility with older kernels, vdso32=2 is an
                alias for vdso32=0.

                Try vdso32=0 if you encounter an error that says:
                dl_main: Assertion `(void *) ph->p_vaddr == _rtld_local._dl

    video=          [FB,EARLY] Frame buffer configuration
                See Documentation/fb/modedb.rst.

    video.brightness_switch_enabled= [ACPI]
                Format: [0|1]
                If set to 1, on receiving an ACPI notify event
                generated by hotkey, video driver will adjust brightness
                level and then send out the event to user space through
                the allocated input device. If set to 0, video driver
                will only send out the event without touching backlight
                brightness level.
                default: 1

    virtio_mmio.device=
                [VMMIO] Memory mapped virtio (platform) device.

                        <size>@<baseaddr>:<irq>[:<id>]
                where:
                        <size>     := size (can use standard suffixes
                                        like K, M and G)
```

```
                        <baseaddr> := physical base address
                        <irq>      := interrupt number (as passed to
                                      request_irq())
                        <id>       := (optional) platform device id
                    example:
                        virtio_mmio.device=1K@0x100b0000:48:7

                    Can be used multiple times for multiple devices.

    vga=            [BOOT,X86-32] Select a particular video mode
                    See Documentation/arch/x86/boot.rst and
                    Documentation/admin-guide/svga.rst.
                    Use vga=ask for menu.
                    This is actually a boot loader parameter; the value is
                    passed to the kernel using a special protocol.

    vm_debug[=options]      [KNL] Available with CONFIG_DEBUG_VM=y.
                    May slow down system boot speed, especially when
                    enabled on systems with a large amount of memory.
                    All options are enabled by default, and this
                    interface is meant to allow for selectively
                    enabling or disabling specific virtual memory
                    debugging features.

                    Available options are:
                      P     Enable page structure init time poisoning
                      -     Disable all of the above options

    vmalloc=nn[KMG] [KNL,BOOT,EARLY] Forces the vmalloc area to have an
                    exact size of <nn>. This can be used to increase
                    the minimum size (128MB on x86, arm32 platforms).
                    It can also be used to decrease the size and leave more roo
                    for directly mapped kernel RAM. Note that this parameter do
                    not exist on many other platforms (including arm64, alpha,
                    loongarch, arc, csky, hexagon, microblaze, mips, nios2, ope
                    parisc, m64k, powerpc, riscv, sh, um, xtensa, s390, sparc).

    vmcp_cma=nn[MG] [KNL,S390,EARLY]
                    Sets the memory size reserved for contiguous memory
                    allocations for the vmcp device driver.

    vmhalt=         [KNL,S390] Perform z/VM CP command after system halt.
                    Format: <command>

    vmpanic=        [KNL,S390] Perform z/VM CP command after kernel panic.
                    Format: <command>

    vmpoff=         [KNL,S390] Perform z/VM CP command after power off.
                    Format: <command>

    vmscape=        [X86] Controls mitigation for VMscape attacks.
                    VMscape attacks can leak information from a userspace
```

```
                hypervisor to a guest via speculative side-channels.

                off             - disable the mitigation
                ibpb            - use Indirect Branch Prediction Barrier
                                  (IBPB) mitigation (default)
                force           - force vulnerability detection even on
                                  unaffected processors

    vsyscall=   [X86-64,EARLY]
                Controls the behavior of vsyscalls (i.e. calls to
                fixed addresses of 0xffffffffff600x00 from legacy
                code).  Most statically-linked binaries and older
                versions of glibc use these calls.  Because these
                functions are at fixed addresses, they make nice
                targets for exploits that can control RIP.

                emulate     Vsyscalls turn into traps and are emulated
                            reasonably safely.  The vsyscall page is
                            readable.  This disables the Linear
                            Address Space Separation (LASS) security
                            feature and makes the system less secure.

                xonly       [default] Vsyscalls turn into traps and are
                            emulated reasonably safely.  The vsyscall
                            page is not readable.

                none        Vsyscalls don't work at all.  This makes
                            them quite hard to use for exploits but
                            might break your system.

    vt.color=   [VT] Default text color.
                Format: 0xYX, X = foreground, Y = background.
                Default: 0x07 = light gray on black.

    vt.cur_default= [VT] Default cursor shape.
                Format: 0xCCBBAA, where AA, BB, and CC are the same as
                the parameters of the <Esc>[?A;B;Cc escape sequence;
                see vga-softcursor.rst. Default: 2 = underline.

    vt.default_blu= [VT]
                Format: <blue0>,<blue1>,<blue2>,...,<blue15>
                Change the default blue palette of the console.
                This is a 16-member array composed of values
                ranging from 0-255.

    vt.default_grn= [VT]
                Format: <green0>,<green1>,<green2>,...,<green15>
                Change the default green palette of the console.
                This is a 16-member array composed of values
                ranging from 0-255.

    vt.default_red= [VT]
```

```
                Format: <red0>,<red1>,<red2>,...,<red15>
                Change the default red palette of the console.
                This is a 16-member array composed of values
                ranging from 0-255.

    vt.default_utf8=
                [VT]
                Format=<0|1>
                Set system-wide default UTF-8 mode for all tty's.
                Default is 1, i.e. UTF-8 mode is enabled for all
                newly opened terminals.

    vt.global_cursor_default=
                [VT]
                Format=<-1|0|1>
                Set system-wide default for whether a cursor
                is shown on new VTs. Default is -1,
                i.e. cursors will be created by default unless
                overridden by individual drivers. 0 will hide
                cursors, 1 will display them.

    vt.italic=      [VT] Default color for italic text; 0-15.
                Default: 2 = green.

    vt.underline=   [VT] Default color for underlined text; 0-15.
                Default: 3 = cyan.

    watchdog timers [HW,WDT] For information on watchdog timers,
                see Documentation/watchdog/watchdog-parameters.rst
                or other driver-specific files in the
                Documentation/watchdog/ directory.

    watchdog_thresh=
                [KNL]
                Set the hard lockup detector stall duration
                threshold in seconds. The soft lockup detector
                threshold is set to twice the value. A value of 0
                disables both lockup detectors. Default is 10
                seconds.

    workqueue.unbound_cpus=
                [KNL,SMP] Specify to constrain one or some CPUs
                to use in unbound workqueues.
                Format: <cpu-list>
                By default, all online CPUs are available for
                unbound workqueues.

    workqueue.watchdog_thresh=
                If CONFIG_WQ_WATCHDOG is configured, workqueue can
                warn stall conditions and dump internal state to
                help debugging.  0 disables workqueue stall
                detection; otherwise, it's the stall threshold
```

```
                duration in seconds.  The default value is 30 and
                it can be updated at runtime by writing to the
                corresponding sysfs file.

    workqueue.panic_on_stall=<uint>
                Panic when workqueue stall is detected by
                CONFIG_WQ_WATCHDOG. It sets the number times of the
                stall to trigger panic.

                The default is set by CONFIG_BOOTPARAM_WQ_STALL_PANIC,
                which is 0 (disabled) if not configured.

    workqueue.panic_on_stall_time=<uint>
                Panic when a workqueue stall has been continuous for
                the specified number of seconds. Unlike panic_on_stall
                which counts accumulated stall events, this triggers
                based on the duration of a single continuous stall.

                The default is 0, which disables the time-based panic.

    workqueue.cpu_intensive_thresh_us=
                Per-cpu work items which run for longer than this
                threshold are automatically considered CPU intensive
                and excluded from concurrency management to prevent
                them from noticeably delaying other per-cpu work
                items. Default is 10000 (10ms).

                If CONFIG_WQ_CPU_INTENSIVE_REPORT is set, the kernel
                will report the work functions which violate this
                threshold repeatedly. They are likely good
                candidates for using WQ_UNBOUND workqueues instead.

    workqueue.cpu_intensive_warning_thresh=<uint>
                If CONFIG_WQ_CPU_INTENSIVE_REPORT is set, the kernel
                will report the work functions which violate the
                intensive_threshold_us repeatedly. In order to prevent
                spurious warnings, start printing only after a work
                function has violated this threshold number of times.

                The default is 4 times. 0 disables the warning.

    workqueue.power_efficient
                Per-cpu workqueues are generally preferred because
                they show better performance thanks to cache
                locality; unfortunately, per-cpu workqueues tend to
                be more power hungry than unbound workqueues.

                Enabling this makes the per-cpu workqueues which
                were observed to contribute significantly to power
                consumption unbound, leading to measurably lower
                power usage at the cost of small performance
                overhead.
```

```
                The default value of this parameter is determined by
                the config option CONFIG_WQ_POWER_EFFICIENT_DEFAULT.

    workqueue.default_affinity_scope=
                Select the default affinity scope to use for unbound
                workqueues. Can be one of "cpu", "smt", "cache",
                "cache_shard", "numa" and "system". Default is
                "cache_shard". For more
                information, see the Affinity Scopes section in
                Documentation/core-api/workqueue.rst.

                This can be changed after boot by writing to the
                matching /sys/module/workqueue/parameters file. All
                workqueues with the "default" affinity scope will be
                updated accordingly.

    workqueue.debug_force_rr_cpu
                Workqueue used to implicitly guarantee that work
                items queued without explicit CPU specified are put
                on the local CPU.  This guarantee is no longer true
                and while local CPU is still preferred work items
                may be put on foreign CPUs.  This debug option
                forces round-robin CPU selection to flush out
                usages which depend on the now broken guarantee.
                When enabled, memory and cache locality will be
                impacted.

    writecombine=   [LOONGARCH,EARLY] Control the MAT (Memory Access
                Type) of ioremap_wc().

                on   - Enable writecombine, use WUC for ioremap_wc()
                off  - Disable writecombine, use SUC for ioremap_wc()

    x2apic_phys     [X86-64,APIC,EARLY] Use x2apic physical mode instead of
                default x2apic cluster mode on platforms
                supporting x2apic.

    xen_512gb_limit         [KNL,X86-64,XEN]
                Restricts the kernel running paravirtualized under Xen
                to use only up to 512 GB of RAM. The reason to do so is
                crash analysis tools and Xen tools for doing domain
                save/restore/migration must be enabled to handle larger
                domains.

    xen_console_io  [XEN,EARLY]
                Boolean option to enable/disable the usage of the Xen
                console_io hypercalls to read and write to the console.
                Mostly useful for debugging and development.

    xen_emul_unplug=                [HW,X86,XEN,EARLY]
                Unplug Xen emulated devices
```

```
                Format: [unplug0,][unplug1]
                ide-disks -- unplug primary master IDE devices
                aux-ide-disks -- unplug non-primary-master IDE devices
                nics -- unplug network devices
                all -- unplug all emulated devices (NICs and IDE disks)
                unnecessary -- unplugging emulated devices is
                        unnecessary even if the host did not respond to
                        the unplug protocol
                never -- do not unplug even if version check succeeds

    xen_legacy_crash        [X86,XEN,EARLY]
                Crash from Xen panic notifier, without executing late
                panic() code such as dumping handler.

    xen_mc_debug    [X86,XEN,EARLY]
                Enable multicall debugging when running as a Xen PV guest.
                Enabling this feature will reduce performance a little
                bit, so it should only be enabled for obtaining extended
                debug data in case of multicall errors.

    xen_msr_safe=   [X86,XEN,EARLY]
                Format: <bool>
                Select whether to always use non-faulting (safe) MSR
                access functions when running as Xen PV guest. The
                default value is controlled by CONFIG_XEN_PV_MSR_SAFE.

    xen_nopv        [X86]
                Disables the PV optimizations forcing the HVM guest to
                run as generic HVM guest with no PV drivers.
                This option is obsoleted by the "nopv" option, which
                has equivalent effect for XEN platform.

    xen_no_vector_callback
                [KNL,X86,XEN,EARLY] Disable the vector callback for Xen
                event channel interrupts.

    xen_scrub_pages=        [XEN]
                Boolean option to control scrubbing pages before giving the
                to Xen, for use by other domains. Can be also changed at ru
                with /sys/devices/system/xen_memory/xen_memory0/scrub_pages
                Default value controlled with CONFIG_XEN_SCRUB_PAGES_DEFAUL

    xen_timer_slop= [X86-64,XEN,EARLY]
                Set the timer slop (in nanoseconds) for the virtual Xen
                timers (default is 100000). This adjusts the minimum
                delta of virtualized Xen timers, where lower values
                improve timer resolution at the expense of processing
                more timer interrupts.

    xen.balloon_boot_timeout= [XEN]
                The time (in seconds) to wait before giving up to boot
                in case initial ballooning fails to free enough memory.
```

```
                Applies only when running as HVM or PVH guest and
                started with less memory configured than allowed at
                max. Default is 180.

    xen.event_eoi_delay=    [XEN]
                How long to delay EOI handling in case of event
                storms (jiffies). Default is 10.

    xen.event_loop_timeout= [XEN]
                After which time (jiffies) the event handling loop
                should start to delay EOI handling. Default is 2.

    xen.fifo_events=        [XEN]
                Boolean parameter to disable using fifo event handling
                even if available. Normally fifo event handling is
                preferred over the 2-level event handling, as it is
                fairer and the number of possible event channels is
                much higher. Default is on (use fifo events).

    xirc2ps_cs=     [NET,PCMCIA]
                Format:
                <irq>,<irq_mask>,<io>,<full_duplex>,<do_sound>,<lockup_hack

    xive=           [PPC]
                By default on POWER9 and above, the kernel will
                natively use the XIVE interrupt controller. This option
                allows the fallback firmware mode to be used:

                off       Fallback to firmware control of XIVE interrupt
                          controller on both pseries and powernv
                          platforms. Only useful on POWER9 and above.

    xive.store-eoi=off      [PPC]
                By default on POWER10 and above, the kernel will use
                stores for EOI handling when the XIVE interrupt mode
                is active. This option allows the XIVE driver to use
                loads instead, as on POWER9.

    xhci-hcd.quirks         [USB,KNL]
                A hex value specifying bitmask with supplemental xhci
                host controller quirks. Meaning of each bit can be
                consulted in header drivers/usb/host/xhci.h.

    xmon            [PPC,EARLY]
                Format: { early | on | rw | ro | off }
                Controls if xmon debugger is enabled. Default is off.
                Passing only "xmon" is equivalent to "xmon=early".
                early   Call xmon as early as possible on boot; xmon
                        debugger is called from setup_arch().
                on      xmon debugger hooks will be installed so xmon
                        is only called on a kernel crash. Default mode,
                        i.e. either "ro" or "rw" mode, is controlled
```

```
                        with CONFIG_XMON_DEFAULT_RO_MODE.
                rw      xmon debugger hooks will be installed so xmon
                        is called only on a kernel crash, mode is write,
                        meaning SPR registers, memory and, other data
                        can be written using xmon commands.
                ro      same as "rw" option above but SPR registers,
                        memory, and other data can't be written using
                        xmon commands.
                off     xmon is disabled.
```