Documentation for /proc/sys/kernel/* kernel version 2.2.10

(c) 1998, 1999, Rik van Riel <riel@nl.linux.org>
(c) 2009, Shen Feng<shen@cn.fujitsu.com>

For general info and legal blurb, please look in README.

==

This file contains documentation for the sysctl files in
/proc/sys/kernel/ and is valid for Linux kernel version 2.2.

The files in this directory can be used to tune and monitor
miscellaneous and general things in the operation of the Linux
kernel. Since some of the files _can_ be used to screw up your
system, it is advisable to read both documentation and source
before actually making adjustments.

Currently, these files might (depending on your configuration)
show up in /proc/sys/kernel:

- acpi_video_flags
- acct
- bootloader_type [X86 only]
- bootloader_version [X86 only]
- callhome [S390 only]
- auto_msgmni
- core_pattern
- core_pipe_limit
- core_uses_pid
- ctrl-alt-del
- dentry-state
- domainname
- hostname
- hotplug
- java-appletviewer [binfmt_java, obsolete]
- java-interpreter [binfmt_java, obsolete]
- kstack_depth_to_print [X86 only]
- l2cr [PPC only]
- modprobe ==> Documentation/debugging-modules.txt
- modules_disabled
- msgmax
- msgmnb
- msgmni
- nmi_watchdog
- osrelease
- ostype
- overflowgid
- overflowuid
- panic
- pid_max
- powersave-nap [PPC only]
- panic_on_unrecovered_nmi
- printk
- randomize_va_space
- real-root-dev ==> Documentation/initrd.txt
- reboot-cmd [SPARC only]
- rtsig-max

- rtsig-nr
- sem
- sg-big-buff [generic SCSI device (sg)]
- shmall
- shmmax [sysv ipc]
- shmmni
- stop-a [SPARC only]
- sysrq ==> Documentation/sysrq.txt
- tainted
- threads-max
- unknown_nmi_panic
- version

==

acpi_video_flags:

flags

See Doc*/kernel/power/video.txt, it allows mode of video boot to be
set during run time.

==

acct:

highwater lowwater frequency

If BSD-style process accounting is enabled these values control
its behaviour. If free space on filesystem where the log lives
goes below <lowwater>% accounting suspends. If free space gets
above <highwater>% accounting resumes. <Frequency> determines
how often do we check the amount of free space (value is in
seconds). Default:
4 2 30
That is, suspend accounting if there left <= 2% free; resume it
if we got >=4%; consider information about amount of free space
valid for 30 seconds.

==

bootloader_type:

x86 bootloader identification

This gives the bootloader type number as indicated by the bootloader,
shifted left by 4, and OR'd with the low four bits of the bootloader
version. The reason for this encoding is that this used to match the
type_of_loader field in the kernel header; the encoding is kept for
backwards compatibility. That is, if the full bootloader type number
is 0x15 and the full version number is 0x234, this file will contain
the value 340 = 0x154.

See the type_of_loader and ext_loader_type fields in
Documentation/x86/boot.txt for additional information.

==

bootloader_version:

x86 bootloader version

The complete bootloader version number. In the example above, this
file will contain the value 564 = 0x234.

See the type_of_loader and ext_loader_ver fields in
Documentation/x86/boot.txt for additional information.

==

callhome:

Controls the kernel's callhome behavior in case of a kernel panic.

The s390 hardware allows an operating system to send a notification
to a service organization (callhome) in case of an operating system panic.

When the value in this file is 0 (which is the default behavior)
nothing happens in case of a kernel panic. If this value is set to "1"
the complete kernel oops message is send to the IBM customer service
organization in case the mainframe the Linux operating system is running
on has a service contract with IBM.

==

core_pattern:

core_pattern is used to specify a core dumpfile pattern name.
. max length 128 characters; default value is "core"
. core_pattern is used as a pattern template for the output filename;
  certain string patterns (beginning with '%') are substituted with
  their actual values.
. backward compatibility with core_uses_pid:
        If core_pattern does not include "%p" (default does not)
        and core_uses_pid is set, then .PID will be appended to
        the filename.
. corename format specifiers:
        %<NUL> '%' is dropped
        %%   output one '%'
        %p   pid
        %u   uid
        %g   gid
        %s   signal number
        %t   UNIX time of dump
        %h   hostname
        %e   executable filename
        %<OTHER> both are dropped
. If the first character of the pattern is a '|', the kernel will treat
  the rest of the pattern as a command to run. The core dump will be
  written to the standard input of that program instead of to a file.

==

core_pipe_limit:

This sysctl is only applicable when core_pattern is configured to pipe core files to a user space helper (when the first character of core_pattern is a '|', see above). When collecting cores via a pipe to an application, it is occasionally useful for the collecting application to gather data about the crashing process from its /proc/pid directory. In order to do this safely, the kernel must wait for the collecting process to exit, so as not to remove the crashing processes proc files prematurely. This in turn creates the possibility that a misbehaving userspace collecting process can block the reaping of a crashed process simply by never exiting. This sysctl defends against that. It defines how many concurrent crashing processes may be piped to user space applications in parallel. If this value is exceeded, then those crashing processes above that value are noted via the kernel log and their cores are skipped. 0 is a special value, indicating that unlimited processes may be captured in parallel, but that no waiting will take place (i.e. the collecting process is not guaranteed access to /proc/<crashing pid>/). This value defaults to 0.

==

core_uses_pid:

The default coredump filename is "core". By setting
core_uses_pid to 1, the coredump filename becomes core.PID.
If core_pattern does not include "%p" (default does not)
and core_uses_pid is set, then .PID will be appended to
the filename.

==

ctrl-alt-del:

When the value in this file is 0, ctrl-alt-del is trapped and
sent to the init(1) program to handle a graceful restart.
When, however, the value is > 0, Linux's reaction to a Vulcan
Nerve Pinch (tm) will be an immediate reboot, without even
syncing its dirty buffers.

Note: when a program (like dosemu) has the keyboard in 'raw'
mode, the ctrl-alt-del is intercepted by the program before it
ever reaches the kernel tty layer, and it's up to the program
to decide what to do with it.

==

domainname & hostname:

These files can be used to set the NIS/YP domainname and the
hostname of your box in exactly the same way as the commands
domainname and hostname, i.e.:
echo "darkstar" > /proc/sys/kernel/hostname
echo "mydomain" > /proc/sys/kernel/domainname
has the same effect as
hostname "darkstar"

```
# domainname "mydomain"
```

Note, however, that the classic darkstar.frop.org has the hostname "darkstar" and DNS (Internet Domain Name Server) domainname "frop.org", not to be confused with the NIS (Network Information Service) or YP (Yellow Pages) domainname. These two domain names are in general different. For a detailed discussion see the hostname(1) man page.

==========

hotplug:

Path for the hotplug policy agent.
Default value is "/sbin/hotplug".

==========

l2cr: (PPC only)

This flag controls the L2 cache of G3 processor boards. If 0, the cache is disabled. Enabled if nonzero.

==========

kstack_depth_to_print: (X86 only)

Controls the number of words to print when dumping the raw kernel stack.

==========

modules_disabled:

A toggle value indicating if modules are allowed to be loaded in an otherwise modular kernel. This toggle defaults to off (0), but can be set true (1). Once true, modules can be neither loaded nor unloaded, and the toggle cannot be set back to false.

==========

osrelease, ostype & version:

```
# cat osrelease
2.1.88
# cat ostype
Linux
# cat version
#5 Wed Feb 25 21:49:24 MET 1998
```

The files osrelease and ostype should be clear enough. Version needs a little more clarification however. The '#5' means that this is the fifth kernel built from this source base and the date behind it indicates the time the kernel was built.
The only way to tune these values is to rebuild the kernel :-)

==

overflowgid & overflowuid:

if your architecture did not always support 32-bit UIDs (i.e. arm, i386, m68k, sh, and sparc32), a fixed UID and GID will be returned to applications that use the old 16-bit UID/GID system calls, if the actual UID or GID would exceed 65535.

These sysctls allow you to change the value of the fixed UID and GID. The default is 65534.

==

panic:

The value in this file represents the number of seconds the kernel waits before rebooting on a panic. When you use the software watchdog, the recommended setting is 60.

==

panic_on_oops:

Controls the kernel's behaviour when an oops or BUG is encountered.

0: try to continue operation

1: panic immediately. If the `panic' sysctl is also non-zero then the machine will be rebooted.

==

pid_max:

PID allocation wrap value. When the kernel's next PID value reaches this value, it wraps back to a minimum PID value. PIDs of value pid_max or larger are not allocated.

==

powersave-nap: (PPC only)

If set, Linux-PPC will use the 'nap' mode of powersaving, otherwise the 'doze' mode will be used.

==

printk:

The four values in printk denote: console_loglevel, default_message_loglevel, minimum_console_loglevel and default_console_loglevel respectively.

These values influence printk() behavior when printing or

logging error messages. See 'man 2 syslog' for more info on
the different loglevels.

- console_loglevel: messages with a higher priority than
  this will be printed to the console
- default_message_level: messages without an explicit priority
  will be printed with this priority
- minimum_console_loglevel: minimum (highest) value to which
  console_loglevel can be set
- default_console_loglevel: default value for console_loglevel

==

printk_ratelimit:

Some warning messages are rate limited. printk_ratelimit specifies
the minimum length of time between these messages (in jiffies), by
default we allow one every 5 seconds.

A value of 0 will disable rate limiting.

==

printk_ratelimit_burst:

While long term we enforce one message per printk_ratelimit
seconds, we do allow a burst of messages to pass through.
printk_ratelimit_burst specifies the number of messages we can
send before ratelimiting kicks in.

==

printk_delay:

Delay each printk message in printk_delay milliseconds

Value from 0 - 10000 is allowed.

==

randomize-va-space:

This option can be used to select the type of process address
space randomization that is used in the system, for architectures
that support this feature.

0 - Turn the process address space randomization off. This is the
    default for architectures that do not support this feature anyways,
    and kernels that are booted with the "norandmaps" parameter.

1 - Make the addresses of mmap base, stack and VDSO page randomized.
    This, among other things, implies that shared libraries will be
    loaded to random addresses. Also for PIE-linked binaries, the
    location of code start is randomized. This is the default if the
    CONFIG_COMPAT_BRK option is enabled.

2 - Additionally enable heap randomization. This is the default if
    CONFIG_COMPAT_BRK is disabled.

    There are a few legacy applications out there (such as some ancient
    versions of libc.so.5 from 1996) that assume that brk area starts
    just after the end of the code+bss. These applications break when
    start of the brk area is randomized. There are however no known
    non-legacy applications that would be broken this way, so for most
    systems it is safe to choose full randomization.

    Systems with ancient and/or broken binaries should be configured
    with CONFIG_COMPAT_BRK enabled, which excludes the heap from process
    address space randomization.

==

reboot-cmd: (Sparc only)

??? This seems to be a way to give an argument to the Sparc
ROM/Flash boot loader. Maybe to tell it what to do after
rebooting. ???

==

rtsig-max & rtsig-nr:

The file rtsig-max can be used to tune the maximum number
of POSIX realtime (queued) signals that can be outstanding
in the system.

rtsig-nr shows the number of RT signals currently queued.

==

sg-big-buff:

This file shows the size of the generic SCSI (sg) buffer.
You can't tune it just yet, but you could change it on
compile time by editing include/scsi/sg.h and changing
the value of SG_BIG_BUFF.

There shouldn't be any reason to change this value. If
you can come up with one, you probably know what you
are doing anyway :)

==

shmmax:

This value can be used to query and set the run time limit
on the maximum shared memory segment size that can be created.
Shared memory segments up to 1Gb are now supported in the
kernel. This value defaults to SHMMAX.

==

softlockup_thresh:

This value can be used to lower the softlockup tolerance threshold. The
default threshold is 60 seconds. If a cpu is locked up for 60 seconds,
the kernel complains. Valid values are 1-60 seconds. Setting this
tunable to zero will disable the softlockup detection altogether.

==

tainted:

Non-zero if the kernel has been tainted. Numeric values, which
can be ORed together:

```
   1 - A module with a non-GPL license has been loaded, this
       includes modules with no license.
       Set by modutils >= 2.4.9 and module-init-tools.
   2 - A module was force loaded by insmod -f.
       Set by modutils >= 2.4.9 and module-init-tools.
   4 - Unsafe SMP processors: SMP with CPUs not designed for SMP.
   8 - A module was forcibly unloaded from the system by rmmod -f.
  16 - A hardware machine check error occurred on the system.
  32 - A bad page was discovered on the system.
  64 - The user has asked that the system be marked "tainted". This
       could be because they are running software that directly modifies
       the hardware, or for other reasons.
 128 - The system has died.
 256 - The ACPI DSDT has been overridden with one supplied by the user
        instead of using the one provided by the hardware.
 512 - A kernel warning has occurred.
1024 - A module from drivers/staging was loaded.
```

==

auto_msgmni:

Enables/Disables automatic recomputing of msgmni upon memory add/remove or
upon ipc namespace creation/removal (see the msgmni description above).
Echoing "1" into this file enables msgmni automatic recomputing.
Echoing "0" turns it off.
auto_msgmni default value is 1.

==

nmi_watchdog:

Enables/Disables the NMI watchdog on x86 systems. When the value is non-zero
the NMI watchdog is enabled and will continuously test all online cpus to
determine whether or not they are still functioning properly. Currently,
passing "nmi_watchdog=" parameter at boot time is required for this function
to work.

If LAPIC NMI watchdog method is in use (nmi_watchdog=2 kernel parameter), the
NMI watchdog shares registers with oprofile. By disabling the NMI watchdog,
oprofile may have more registers to utilize.

==========

unknown_nmi_panic:

The value in this file affects behavior of handling NMI. When the value is non-zero, unknown NMI is trapped and then panic occurs. At that time, kernel debugging information is displayed on console.

NMI switch that most IA32 servers have fires unknown NMI up, for example.
If a system hangs up, try pressing the NMI switch.

==========

panic_on_unrecovered_nmi:

The default Linux behaviour on an NMI of either memory or unknown is to continue operation. For many environments such as scientific computing it is preferable that the box is taken out and the error dealt with than an uncorrected parity/ECC error get propogated.

A small number of systems do generate NMI's for bizarre random reasons such as power management so the default is off. That sysctl works like the existing panic controls already in that directory.