1. boot kernel parameter, ignore_loglevel, loglevel and log_buf_len

```
	ignore_loglevel	[KNL]
			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.
```

You could also modify "ignore_loglevel" dynamically.

```
root@granite2:~# cat  /sys/module/printk/parameters/ignore_loglevel
Y
root@granite2:~# echo N >  /sys/module/printk/parameters/ignore_loglevel
root@granite2:~# cat  /sys/module/printk/parameters/ignore_loglevel
N
```

```
	loglevel=	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]	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.

==

2. Modify loglevel when build kernel

CONFIG_MESSAGE_LOGLEVEL_DEFAULT=4

==

3. modify loglevel by /proc/sys/kernel/printk

```
root@granite2:~# cat /proc/sys/kernel/printk
```

```
7 4 1 7

root@granite2:~# echo 6 4 1 6 > /proc/sys/kernel/printk

root@granite2:~# cat /proc/sys/kernel/printk

6 4 1 6
```

===

4. cat /dev/kmsg

5. cat /proc/kmsg