Consultar el registro del sistema

1. dmesg

El comando **dmesg** permite recuperar los mensajes del núcleo emitidos al arrancar la máquina y los emitidos después. El registro de dmesg es circular. Después de cierto número de mensajes, los primeros desaparecen. Sin embargo, estas entradas en el registro no están perdidas, ya que el servicio syslogd (capítulo La red) las escribe en archivos.

El administrador, ingeniero o usuario del sistema Linux suele iniciar este comando para comprobar la presencia de posibles errores. En efecto, después del boot, los mensajes siguen llegando, en particular durante la conexión en caliente de periféricos, durante la carga de algunos módulos, cuando se producen averías, durante una corrupción del sistema de archivos, etc.

Se ha truncado voluntariamente el ejemplo siguiente a las primeras líneas, ya que la salida original contiene más de 500. Las primeras muestran todo el principio de la ejecución del núcleo (información facilitada por la BIOS). En medio se muestra la detección del primer disco duro y de sus particiones. El final muestra lo que ocurre al insertar de un pendrive, después del boot, durante una utilización normal y en el momento de desconectarse.

```
# dmesg

[    0.000000] Linux version 5.8.15-301.fc33.x86_64 (mockbuild@bkernel01.iad2.
fedoraproject.org) (gcc (GCC) 10.2.1 20200826 (Red Hat 10.2.1-3), GNU ld version
2.35-10.fc33) #1 SMP Thu Oct 15 16:58:06 UTC 2020
[    0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.8.15-301.fc33.x86_64
root=/dev/mapper/fedora_fedora-root ro rd.lvm.lv=fedora_fedora/root rhgb quiet
[    0.000000] x86/fpu: x87 FPU will use FXSAVE
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000007ffd9fff] usable
[    0.000000] BIOS-e820: [mem 0x000000007ffda000-0x000000007fffffff] reserved
```

```
[    0.000000] BIOS-e820: [mem 0x00000000feffc000-0x00000000feffffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.8 present.
[    0.000000] DMI: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014
[    0.000000] Hypervisor detected: KVM
[    0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[    0.000000] kvm-clock: cpu 0, msr 64601001, primary cpu clock
[    0.000000] kvm-clock: using sched offset of 27310944349 cycles
[    0.000007] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles:
0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[    0.000013] tsc: Detected 2993.198 MHz processor
[    0.000973] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[    0.000974] e820: remove [mem 0x000a0000-0x000fffff] usable
[    0.000977] last_pfn = 0x7ffda max_arch_pfn = 0x400000000
[    0.001011] MTRR default type: write-back
[    0.001012] MTRR fixed ranges enabled:
[    0.001013] 00000-9FFFF write-back
[    0.001014] A0000-BFFFF uncachable
[    0.001014] C0000-FFFFF write-protect
[    0.001015] MTRR variable ranges enabled:
[    0.001016] 0 base 0080000000 mask FF80000000 uncachable
... (y así hasta el final)
```

El resultado no es muy explícito. Podemos mejorar la visualización temporal para que sea más legible empleando -T.

```
# dmesg -T

[mié abr 21 22:37:01 2021] Linux version 5.8.15-301.fc33.x86_64 (mockbuild@bkernel01.iad2.
fedoraproject.org) (gcc (GCC) 10.2.1 20200826 (Red Hat 10.2.1-3), GNU ld version 2.35-10.fc33)
#1 SMP Thu Oct 15 16:58:06 UTC 2020
[mié abr 21 22:37:01 2021] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.8.15-301.fc33.
x86_64 root=/dev/mapper/fedora_fedora-root ro rd.lvm.lv=fedora_fedora/root rhgb quiet
[mié abr 21 22:37:01 2021] x86/fpu: x87 FPU will use FXSAVE
[mié abr 21 22:37:01 2021] BIOS-provided physical RAM map:
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
```

```
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x0000000000100000-0x000000007ffd9fff] usable
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x000000007ffda000-0x000000007fffffff] reserved
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x00000000feffc000-0x00000000feffffff] reserved
[mié abr 21 22:37:01 2021] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[mié abr 21 22:37:01 2021] NX (Execute Disable) protection: active
[mié abr 21 22:37:01 2021] SMBIOS 2.8 present.
[mié abr 21 22:37:01 2021] DMI: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014
[mié abr 21 22:37:01 2021] Hypervisor detected: KVM
[mié abr 21 22:37:01 2021] kvm-clock: Using msrs 4b564d01 and 4b564d00
[mié abr 21 22:37:01 2021] kvm-clock: cpu 0, msr 64601001, primary cpu clock
[mié abr 21 22:37:01 2021] kvm-clock: using sched offset of 27310944349 cycles
[mié abr 21 22:37:01 2021] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles:
0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[mié abr 21 22:37:01 2021] tsc: Detected 2993.198 MHz processor
```

También puede filtrar únicamente los mensajes del núcleo y el nivel de advertencia de la siguiente manera:

```
# dmesg -k -lwarn,err

[    0.387250] acpi PNP0A03:00: fail to add MMCONFIG information, can't access extended PCI
configuration space under this bridge.
[    2.917215] sd 2:0:0:0: Power-on or device reset occurred
[    2.917592] sd 2:0:0:1: Power-on or device reset occurred
[    9.635152] kauditd_printk_skb: 6 callbacks suppressed
[   16.516852] kauditd_printk_skb: 28 callbacks suppressed
[   28.409119] xfs filesystem being remounted at / supports timestamps until 2038 (0x7fffffff)
[   33.856700] kauditd_printk_skb: 22 callbacks suppressed
[   38.470417] xfs filesystem being mounted at /boot supports timestamps until 2038 (0x7fffffff)
[   39.932959] kauditd_printk_skb: 1 callbacks suppressed
```

Para sacarle partido al resultado, lo ideal sería o bien redireccionarlo a un archivo para un análisis más en frío, o bien utilizar el comando **grep** a propósito, si sabe lo que busca.

```
# dmesg|grep sdb
[    2.919925] sd 2:0:0:1: [sdb] 67108864 512-byte logical blocks: (34.4 GB/32.0 GiB)
[    2.919954] sd 2:0:0:1: [sdb] Write Protect is off
[    2.919955] sd 2:0:0:1: [sdb] Mode Sense: 63 00 00 08
[    2.920010] sd 2:0:0:1: [sdb] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
[    2.967984] sdb: sdb1
[    2.968595] sd 2:0:0:1: [sdb] Attached SCSI disk
[   38.145820] audit: type=1130 audit(1619037458.613:97): pid=1 uid=0 auid=4294967295
ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-fsck@dev-sdb1
comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
[   38.344473] EXT4-fs (sdb1): mounted filesystem with ordered data mode. Opts: usrquota
[   38.499214] audit: type=1400 audit(1619037458.966:99): avc: denied { quotaon } for
pid=547 comm="quotaon" name="aquota.user" dev="sdb1" ino=14 scontext=system_u:system_r:
quota_t:s0 tcontext=unconfined_u:object_r:unlabeled_t:s0 tclass=file permissive=0
```

2. /var/log/messages o /var/log/syslog

Sea cual sea la distribución empleada, **`/var/log/messages`** o **`/var/log/syslog`** es el archivo central de los mensajes del sistema, provengan del núcleo o de los servicios. El contenido de este archivo, gestionado por syslog, refleja el estado global del sistema (y no únicamente del núcleo) durante su utilización. En un sistema clásico, su contenido retoma el procedente del comando **dmesg** y el de varios servicios.

Las líneas tienen una firma temporal. Sin que se produzca alguna acción especial (ver logrotate en el capítulo Las tareas administrativas), el archivo crece con el tiempo y no se purga. Un archivo de mensajes puede contener varios miles de líneas, ¡sobre todo si se producen problemas!

```
# wc -l < messages
39208
```

Así, como con el comando **dmesg**, no olvide efectuar un grep para seleccionar sus líneas (o **`tail`**, **`head`**, etc.).

```
# grep drm messages | tail -100
Apr 19 08:48:24 localhost kernel: bochs-drm 0000:00:02.0: vgaarb: deactivate vga console
Apr 19 08:48:24 localhost kernel: [drm] Found bochs VGA, ID 0xb0c0.
Apr 19 08:48:24 localhost kernel: [drm] Framebuffer size 16384 kB @ 0xfd000000, mmio @
0xfea50000.
Apr 19 08:48:24 localhost kernel: [drm] Found EDID data blob.
Apr 19 08:48:24 localhost kernel: [drm] Initialized bochs-drm 1.0.0 20130925 for 0000:00:02.0
on minor 0
Apr 19 08:48:25 localhost kernel: fbcon: bochs-drmdrmfb (fb0) is primary device
Apr 19 08:48:25 localhost kernel: bochs-drm 0000:00:02.0: fb0: bochs-drmdrmfb frame buffer device
Apr 19 08:48:35 localhost systemd[1]: Condition check resulted in Load Kernel Module drm
being skipped.
Apr 21 22:37:03 localhost kernel: bochs-drm 0000:00:02.0: vgaarb: deactivate vga console
Apr 21 22:37:03 localhost kernel: [drm] Found bochs VGA, ID 0xb0c0.
Apr 21 22:37:03 localhost kernel: [drm] Framebuffer size 16384 kB @ 0xfd000000, mmio
@ 0xfea50000.
Apr 21 22:37:03 localhost kernel: [drm] Found EDID data blob.
Apr 21 22:37:03 localhost kernel: [drm] Initialized bochs-drm 1.0.0 20130925 for 0000:00:02.0
on minor 0
Apr 21 22:37:03 localhost kernel: fbcon: bochs-drmdrmfb (fb0) is primary device
Apr 21 22:37:04 localhost kernel: bochs-drm 0000:00:02.0: fb0: bochs-drmdrmfb frame buffer device
Apr 21 22:37:28 localhost systemd[1]: Condition check resulted in Load Kernel Module drm
being skipped.
Apr 22 22:42:16 pc-222 dracut[7592]: *** Including module: drm ***
```

En algunas distribuciones el archivo messages puede estar vacío o ausente, y lo mismo para el archivo syslog. Eso significará que el comando **syslog**, del que hay una descripción más adelante en este capítulo, ha distribuido los mensajes asociados en otros archivos.

3. journalctl

Verá con más detalle el comando **journalctl** en el capítulo de las tareas administrativas. Este es un comando que permite un acceso centralizado a las trazas del sistema y de las aplicaciones, y, por supuesto, el kernel. La opción `-k` proporciona la misma información que **dmesg**, y la opción `-p` para los niveles críticos (loglevels); aquí indica un intervalo: de

warning a emerg(ency).

```
# journalctl -k -p warning..emerg
-- Logs begin at Sat 2021-04-10 18:57:07 CEST, end at Thu 2021-04-22 22:53:27 CEST. --
abr 21 22:37:01 localhost.localdomain kernel: acpi PNP0A03:00: fail to add MMCONFIG
information, can't access>
abr 21 22:37:02 localhost.localdomain systemd[1]: /usr/lib/systemd/system/
plymouth-start.service:15: Unit con>
abr 21 22:37:03 localhost.localdomain kernel: sd 2:0:0:0: Power-on or device reset occurred
abr 21 22:37:03 localhost.localdomain kernel: sd 2:0:0:1: Power-on or device reset occurred
abr 21 22:37:10 localhost.localdomain kernel: kauditd_printk_skb: 6 callbacks suppressed
abr 21 22:37:28 localhost.localdomain kernel: kauditd_printk_skb: 28 callbacks suppressed
abr 21 22:37:28 localhost.localdomain systemd[1]: /usr/lib/systemd/system/
plymouth-start.service:15: Unit con>
abr 21 22:37:28 localhost.localdomain kernel: xfs filesystem being remounted at /
supports timestamps until 2>
abr 21 22:37:34 localhost.localdomain kernel: kauditd_printk_skb: 22 callbacks suppressed
abr 21 22:37:38 localhost.localdomain kernel: xfs filesystem being mounted at /boot
supports timestamps until>
abr 21 22:37:40 localhost.localdomain kernel: kauditd_printk_skb: 1 callbacks suppressed
abr 22 22:39:44 pc-222.home systemd[1]: /usr/lib/systemd/system/plymouth-start.service:15:
Unit configured to>
lines 1-13/13 (END)
```