



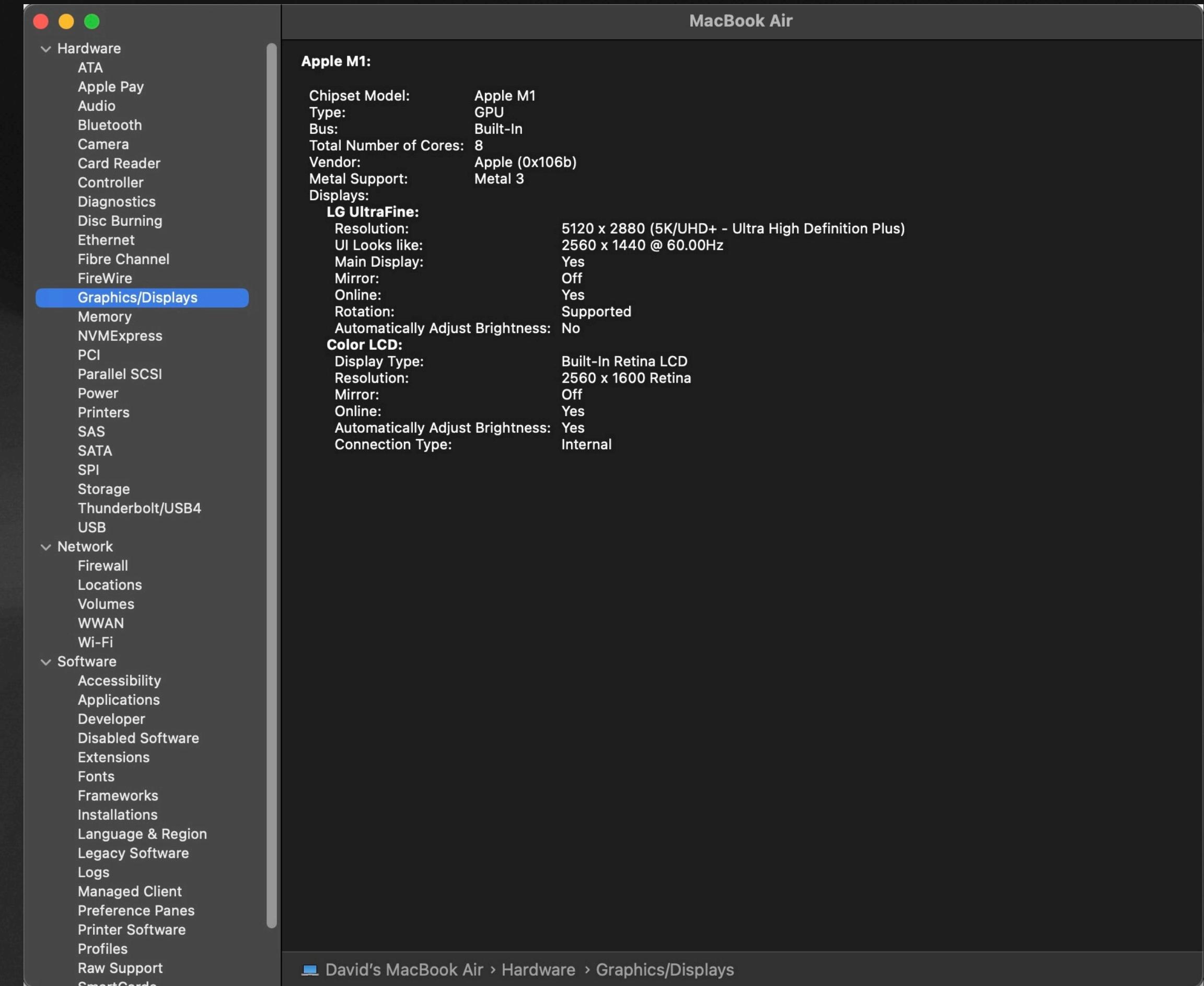
HARDWARE TROUBLESHOOTING ON THE LINUX COMMANDLINE

YOUTUBE.COM/@INFORMATIONTECHWITHDAVE



Introduction

- Modern operating systems offer various tools to gather information about hardware for use in diagnostics and troubleshooting.
- Windows 10 has Device Manager.
- macOS has System Report.
- Linux has several utilities that can be run from the commandline AKA shell.



dmesg

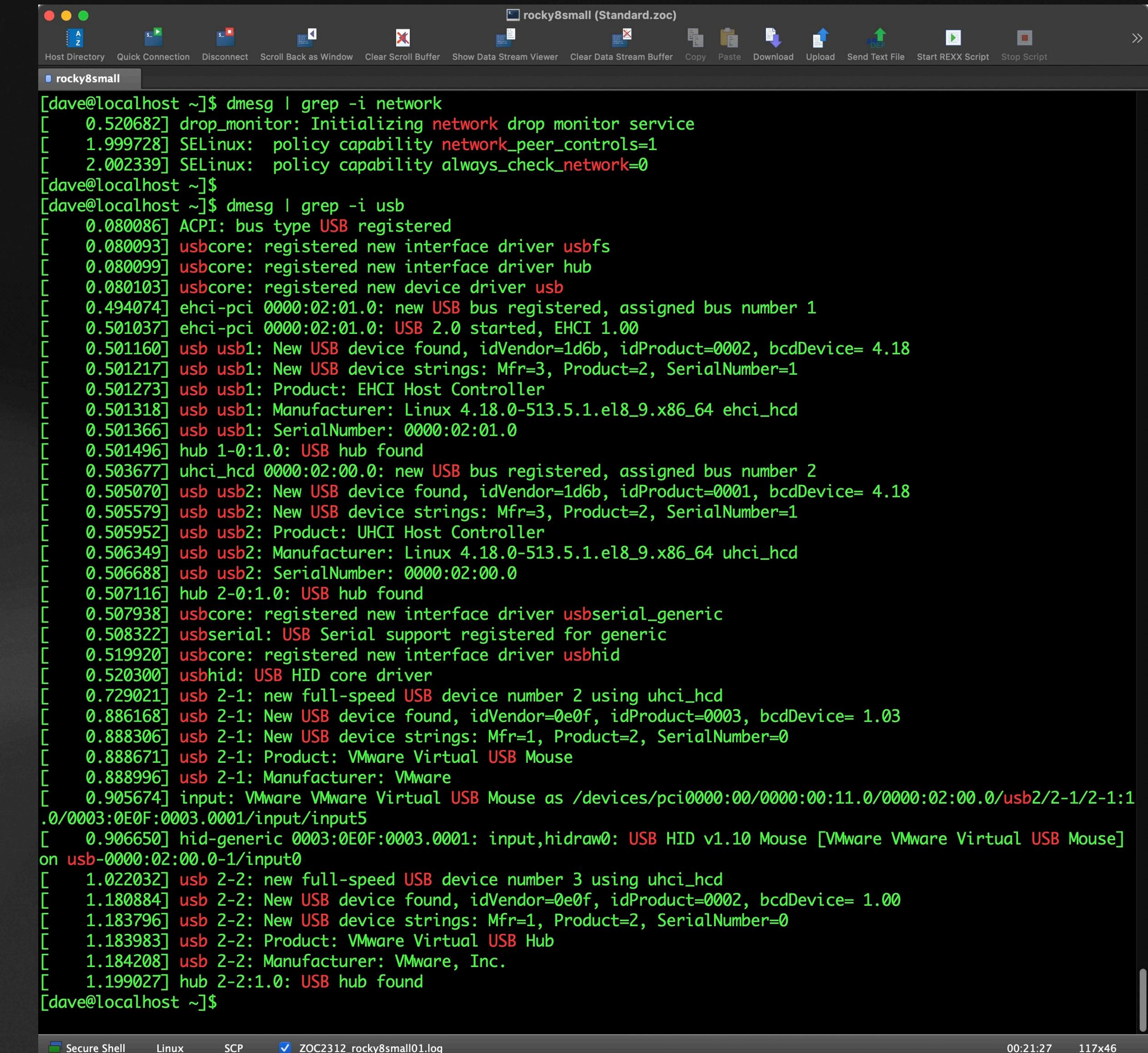
- dmesg displays messages from the kernel's ring buffer.
- The ring buffer stores information about kernel operations, including hardware-related messages.
- This information is helpful for troubleshooting hardware issues, identifying hardware components, and diagnosing system-related problems.

```
[dave@localhost ~]$ dmesg | more
[    0.000000] Linux version 4.18.0-513.5.1.el8_9.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-20) (GCC)) #1 SMP Fri Nov 17 03:31:10 UTC 2023
[    0.000000] Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-4.18.0-513.5.1.el8_9.x86_64 root=/dev/mapper/rl-root ro cr
ashkernel=auto resume=/dev/mapper/rl-swap rd.lvm.lv=rl/root rd.lvm.lv=rl/swap
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x020: 'AVX-512 opmask'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x040: 'AVX-512 Hi256'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x080: 'AVX-512 ZMM_Hi256'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x200: 'Protection Keys User registers'
[    0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[    0.000000] x86/fpu: xstate_offset[5]: 832, xstate_sizes[5]: 64
[    0.000000] x86/fpu: xstate_offset[6]: 896, xstate_sizes[6]: 512
[    0.000000] x86/fpu: xstate_offset[7]: 1408, xstate_sizes[7]: 1024
[    0.000000] x86/fpu: xstate_offset[9]: 2432, xstate_sizes[9]: 8
[    0.000000] x86/fpu: Enabled xstate features 0x2e7, context size is 2440 bytes, using 'compacted' format.
[    0.000000] signal: max sigframe size: 3632
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000000ffff] ACPI NVS
[    0.000000] BIOS-e820: [mem 0x0000000000001000-0x000000000009ffff] usable
[    0.000000] BIOS-e820: [mem 0x0000000000c000-0x0000000000ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000010000-0x000000000efaffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000efb0000-0x000000000efb3ffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000efb4000-0x000000000efc1ffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000efc2000-0x000000000efc6ffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000efc7000-0x000000000fee5ffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000fee6000-0x000000000ff5ffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000ff56000-0x000000000ff71ffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x000000000ff72000-0x000000000ff75ffff] ACPI NVS
[    0.000000] BIOS-e820: [mem 0x000000000ff76000-0x000000000ff7fffff] usable
[    0.000000] BIOS-e820: [mem 0x000000000ffc00000-0x000000000ffc29ffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] e820: update [mem 0xe5c6018-0xe5ce057] usable ==> usable
[    0.000000] e820: update [mem 0xe5c6018-0xe5ce057] usable ==> usable
[    0.000000] e820: update [mem 0xe5d8018-0xe5da057] usable ==> usable
[    0.000000] e820: update [mem 0xe5d8018-0xe5da057] usable ==> usable
[    0.000000] e820: update [mem 0xe5d5018-0xe5d7057] usable ==> usable
[    0.000000] e820: update [mem 0xe5d5018-0xe5d7057] usable ==> usable
[    0.000000] e820: update [mem 0xe5d3018-0xe5d4857] usable ==> usable
[    0.000000] e820: update [mem 0xe5d3018-0xe5d4857] usable ==> usable
[    0.000000] extended physical RAM map:
[    0.000000] reserve setup_data: [mem 0x0000000000000000-0x000000000000ffff] ACPI NVS
[    0.000000] reserve setup_data: [mem 0x0000000000001000-0x000000000009ffff] usable
--More--
```



Filtering dmesg output

- Run with no arguments, dmesg provides an overwhelming amount of information.
- \$ dmesg | grep -i network
- \$ dmesg | grep -i usb
- This filters the output to display only lines that contain the word, “network” or “usb”, respectively.
- The “-i” flag to grep makes it case-insensitive.



The screenshot shows a ZOC (Zotonic) terminal window titled "rocky8small (Standard.zoc)". The window displays the output of the command "dmesg | grep -i network" followed by "dmesg | grep -i usb". The output is color-coded, with red text indicating kernel messages and green text indicating device driver registrations and strings. The terminal interface includes a toolbar at the top with various functions like Host Directory, Quick Connection, Disconnect, and Show Data Stream Viewer, and a status bar at the bottom showing Secure Shell, Linux, SCP, and the log file ZOC2312_rocky8small01.log.

```
[dave@localhost ~]$ dmesg | grep -i network
[ 0.520682] drop_monitor: Initializing network drop monitor service
[ 1.999728] SELinux: policy capability network_peer_controls=1
[ 2.002339] SELinux: policy capability always_check_network=0
[dave@localhost ~]$ 
[dave@localhost ~]$ dmesg | grep -i usb
[ 0.080086] ACPI: bus type USB registered
[ 0.080093] usbcore: registered new interface driver usbf
[ 0.080099] usbcore: registered new interface driver hub
[ 0.080103] usbcore: registered new device driver usb
[ 0.494074] ehci-pci 0000:02:01.0: new USB bus registered, assigned bus number 1
[ 0.501037] ehci-pci 0000:02:01.0: USB 2.0 started, EHCI 1.00
[ 0.501160] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 4.18
[ 0.501217] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 0.501273] usb usb1: Product: EHCI Host Controller
[ 0.501318] usb usb1: Manufacturer: Linux 4.18.0-513.5.1.el8_9.x86_64 ehci_hcd
[ 0.501366] usb usb1: SerialNumber: 0000:02:01.0
[ 0.501496] hub 1-0:1.0: USB hub found
[ 0.503677] uhci_hcd 0000:02:00.0: new USB bus registered, assigned bus number 2
[ 0.505070] usb usb2: New USB device found, idVendor=1d6b, idProduct=0001, bcdDevice= 4.18
[ 0.505579] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 0.505952] usb usb2: Product: UHCI Host Controller
[ 0.506349] usb usb2: Manufacturer: Linux 4.18.0-513.5.1.el8_9.x86_64 uhci_hcd
[ 0.506688] usb usb2: SerialNumber: 0000:02:00.0
[ 0.507116] hub 2-0:1.0: USB hub found
[ 0.507938] usbcore: registered new interface driver usbserial_generic
[ 0.508322] usbserial: USB Serial support registered for generic
[ 0.519920] usbcore: registered new interface driver ushid
[ 0.520300] ushid: USB HID core driver
[ 0.729021] usb 2-1: new full-speed USB device number 2 using uhci_hcd
[ 0.886168] usb 2-1: New USB device found, idVendor=0e0f, idProduct=0003, bcdDevice= 1.03
[ 0.888306] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 0.888671] usb 2-1: Product: VMware Virtual USB Mouse
[ 0.888996] usb 2-1: Manufacturer: VMware
[ 0.905674] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1/2-1:1
[ 0.0003:0E0F:0003.0001/input/input5
[ 0.906650] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMware Virtual USB Mouse]
on usb-0000:02:00.0-1/input0
[ 1.022032] usb 2-2: new full-speed USB device number 3 using uhci_hcd
[ 1.180884] usb 2-2: New USB device found, idVendor=0e0f, idProduct=0002, bcdDevice= 1.00
[ 1.183796] usb 2-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1.183983] usb 2-2: Product: VMware Virtual USB Hub
[ 1.184208] usb 2-2: Manufacturer: VMware, Inc.
[ 1.199027] hub 2-2:1.0: USB hub found
[dave@localhost ~]$
```



Time-Based Filtering

- Kernel messages can be filtered based on the time they occurred, which can help narrow the scope of your troubleshooting.
- \$ dmesg -time-format iso | grep '2023-12-23T13'
- This shows lines on December 23, 2023 at 13:XX.

```
2023-12-23T13:43:00,114971-05:00 systemd[1]: systemd-journald.service: Scheduled restart job, restart counter is at 1.
2023-12-23T13:43:00,115376-05:00 systemd[1]: Stopped Journal Service.
2023-12-23T13:43:00,116594-05:00 systemd[1]: Starting Journal Service...
2023-12-23T13:43:00,144044-05:00 Adding 1679356k swap on /dev/mapper/r1-swap. Priority:-2 extents:1 across:1679356k SSFS
2023-12-23T13:43:00,294890-05:00 XFS (sda2): Mounting V5 Filesystem
2023-12-23T13:43:00,325384-05:00 XFS (sda2): Ending clean mount
2023-12-23T13:43:00,352212-05:00 piix4_smbus 0000:00:07.3: SMBus base address uninitialized - upgrade BIOS or use force_addr
r=0xaddr
2023-12-23T13:43:00,358265-05:00 vmw_vmci 0000:00:07.7: enabling device (0000 -> 0003)
2023-12-23T13:43:00,359174-05:00 vmw_vmci 0000:00:07.7: Using capabilities 0x1c
2023-12-23T13:43:00,363455-05:00 Guest personality initialized and is active
2023-12-23T13:43:00,366129-05:00 VMCI host device registered (name=vmci, major=10, minor=61)
2023-12-23T13:43:00,366459-05:00 Initialized host personality
2023-12-23T13:43:00,413889-05:00 input: PC Speaker as /devices/platform/pcspkr/input/input6
2023-12-23T13:43:00,425641-05:00 ACPI: bus type drm_connector registered
2023-12-23T13:43:00,501996-05:00 checking generic (f0000000 300000) vs hw (f0000000 800000)
2023-12-23T13:43:00,501999-05:00 fb0: switching to vmwgfx from EFI VGA
2023-12-23T13:43:00,502466-05:00 Console: switching to colour dummy device 80x25
2023-12-23T13:43:00,502500-05:00 vmwgfx 0000:00:0f.0: vgaarb: deactivate vga console
2023-12-23T13:43:00,502895-05:00 vmwgfx 0000:00:0f.0: [drm] FIFO at 0x0000000fb80000 size is 8192 kiB
2023-12-23T13:43:00,502915-05:00 vmwgfx 0000:00:0f.0: [drm] VRAM at 0x0000000f000000 size is 131072 kiB
2023-12-23T13:43:00,502934-05:00 vmwgfx 0000:00:0f.0: [drm] Running on SVGA version 2.
2023-12-23T13:43:00,502943-05:00 vmwgfx 0000:00:0f.0: [drm] Capabilities: rect copy, cursor, cursor bypass, cursor bypass 2
, 8bit emulation, alpha cursor, extended fifo, multimon, pitchlock, irq mask, display topology, gmr, traces, gmr2, screen o
bject 2, command buffers, command buffers 2, gbobject, dx, hp cmd queue, no bb restriction, cap2 register,
2023-12-23T13:43:00,502952-05:00 vmwgfx 0000:00:0f.0: [drm] Capabilities2: grow otble, intra surface copy, dx2, gb memsize
2, screendma reg, otble ptdepth2, non ms to ms stretchblt, cursor mob, mshint, cb max size 4mb, dx3, frame type, trace fu
ll fb, extra regs,
2023-12-23T13:43:00,502955-05:00 vmwgfx 0000:00:0f.0: [drm] DMA map mode: Caching DMA mappings.
2023-12-23T13:43:00,503100-05:00 vmwgfx 0000:00:0f.0: [drm] Legacy memory limits: VRAM = 16384 kB, FIFO = 256 kB, surface =
0 kB
2023-12-23T13:43:00,503103-05:00 vmwgfx 0000:00:0f.0: [drm] MOB limits: max mob size = 16384 kB, max mob pages = 12288
2023-12-23T13:43:00,503105-05:00 vmwgfx 0000:00:0f.0: [drm] Max GMR ids is 64
2023-12-23T13:43:00,503106-05:00 vmwgfx 0000:00:0f.0: [drm] Max number of GMR pages is 65536
2023-12-23T13:43:00,503107-05:00 vmwgfx 0000:00:0f.0: [drm] Maximum display memory size is 16384 kB
2023-12-23T13:43:00,508964-05:00 vmwgfx 0000:00:0f.0: [drm] Screen Target display unit initialized
2023-12-23T13:43:00,512076-05:00 vmwgfx 0000:00:0f.0: [drm] Fifo max 0x00040000 min 0x00001000 cap 0x0000077f
2023-12-23T13:43:00,512892-05:00 vmwgfx 0000:00:0f.0: [drm] Using command buffers with DMA pool.
2023-12-23T13:43:00,512901-05:00 vmwgfx 0000:00:0f.0: [drm] Available shader model: Legacy.
2023-12-23T13:43:00,513610-05:00 [drm] Initialized vmwgfx 2.20.0 20211206 for 0000:00:0f.0 on minor 0
2023-12-23T13:43:00,515063-05:00 fbcon: vmwgfxdrmfb (fb0) is primary device
2023-12-23T13:43:00,515417-05:00 Console: switching to colour frame buffer device 160x50
2023-12-23T13:43:00,517750-05:00 vmwgfx 0000:00:0f.0: [drm] fb0: vmwgfxdrmfb frame buffer device
2023-12-23T13:43:01,106134-05:00 IPv6: ADDRCONF(NETDEV_UP): ens192: link is not ready
2023-12-23T13:43:01,107057-05:00 vmxnet3 0000:0b:00.0 ens192: intr type 3, mode 0, 3 vectors allocated
2023-12-23T13:43:01,107796-05:00 vmxnet3 0000:0b:00.0 ens192: NIC Link is Up 10000 Mbps
[dave@localhost ~]$
```



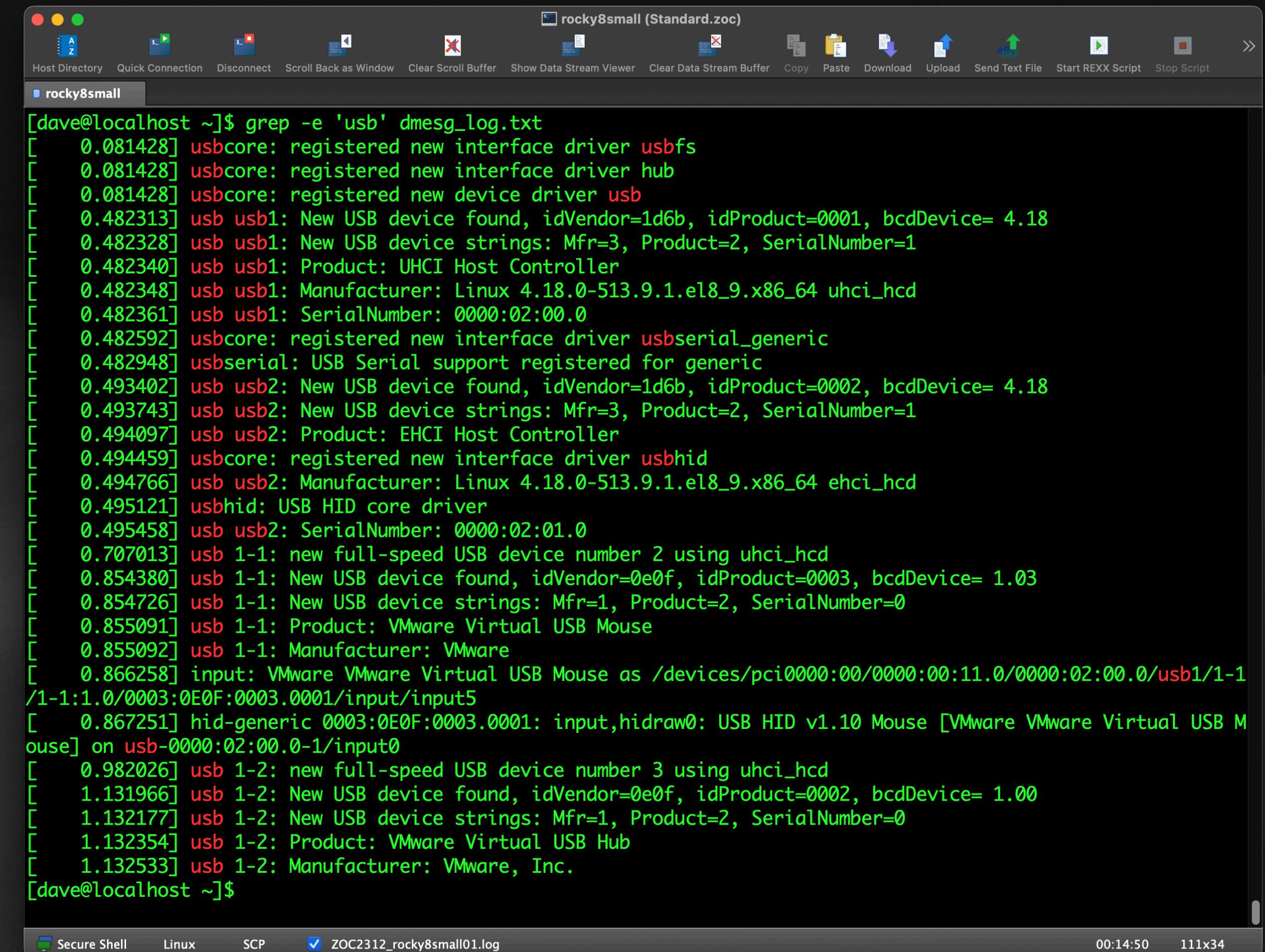
Checking for Hardware Issues

- You can use dmesg for identifying-hardware related problems.
- For example, if you're having a problem with USB, filter the output to display only lines containing “usb”.
- The same applies for other components, e.g., pci, sata, scsi, memory, or any other component.



Redirecting Output to a File

- dmesg generates a lot of output. Saving that to a file can make review and analysis easier.
- May also be useful if a vendor requests your hardware information.
- \$ dmesg > dmesg_log.txt
- Then use tools like grep to search, or less to page through it.



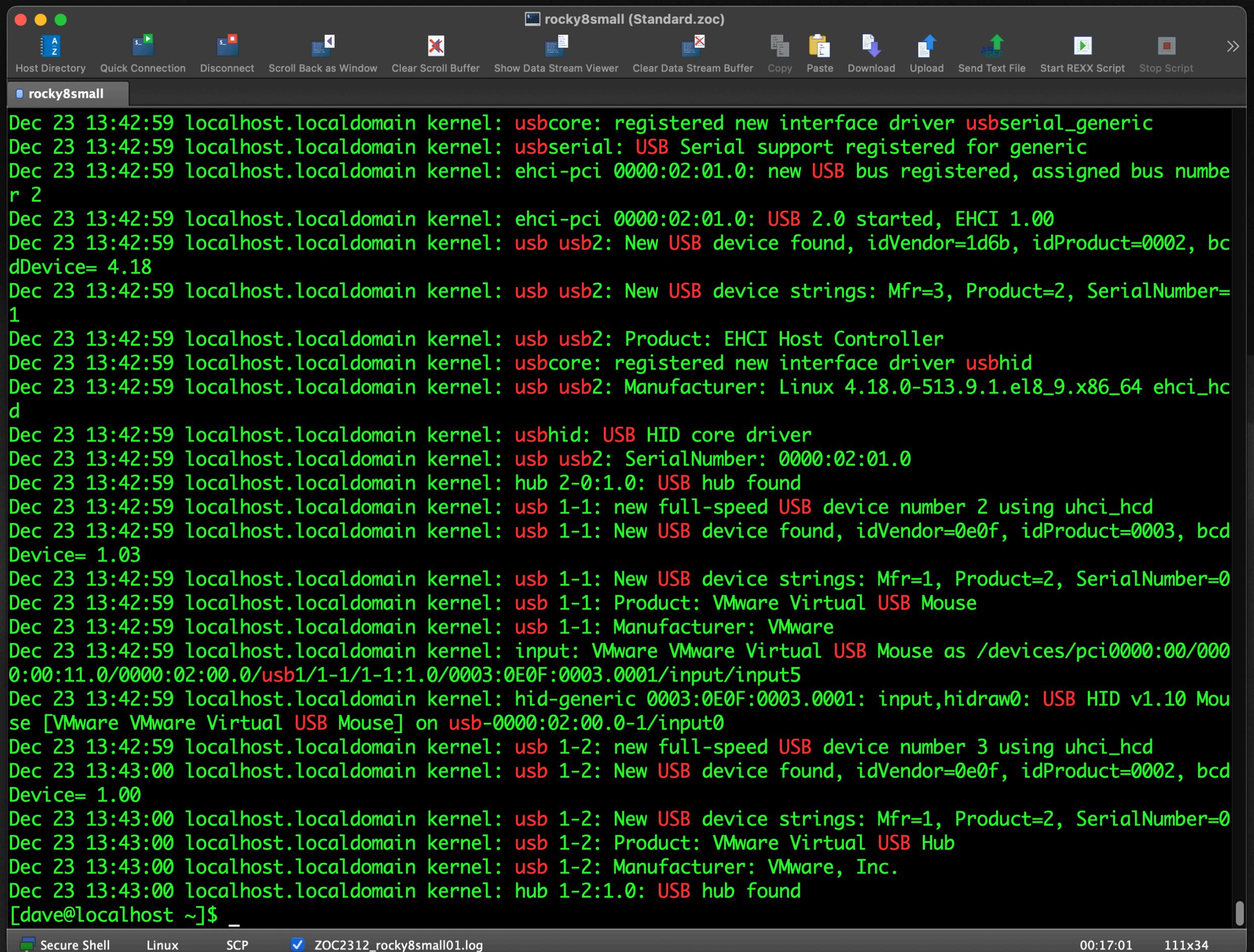
The screenshot shows a terminal window titled "rocky8small (Standard.zoc)" with the session name "rocky8small". The window contains the output of the command "dmesg | grep -e 'usb' dmesg_log.txt". The output lists various USB device detections, including hubs, controllers, and specific devices like VMware Virtual USB Mouse and Hub. The terminal interface includes a toolbar with various file operations and a status bar at the bottom.

```
[dave@localhost ~]$ grep -e 'usb' dmesg_log.txt
[ 0.081428] usbcore: registered new interface driver usbfs
[ 0.081428] usbcore: registered new interface driver hub
[ 0.081428] usbcore: registered new device driver usb
[ 0.482313] usb usb1: New USB device found, idVendor=1d6b, idProduct=0001, bcdDevice= 4.18
[ 0.482328] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 0.482340] usb usb1: Product: UHCI Host Controller
[ 0.482348] usb usb1: Manufacturer: Linux 4.18.0-513.9.1.el8_9.x86_64 uhci_hcd
[ 0.482361] usb usb1: SerialNumber: 0000:02:00.0
[ 0.482592] usbcore: registered new interface driver usbserial_generic
[ 0.482948] usbserial: USB Serial support registered for generic
[ 0.493402] usb usb2: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 4.18
[ 0.493743] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 0.494097] usb usb2: Product: EHCI Host Controller
[ 0.494459] usbcore: registered new interface driver usbhid
[ 0.494766] usb usb2: Manufacturer: Linux 4.18.0-513.9.1.el8_9.x86_64 ehci_hcd
[ 0.495121] usbhid: USB HID core driver
[ 0.495458] usb usb2: SerialNumber: 0000:02:01.0
[ 0.707013] usb 1-1: new full-speed USB device number 2 using uhci_hcd
[ 0.854380] usb 1-1: New USB device found, idVendor=0e0f, idProduct=0003, bcdDevice= 1.03
[ 0.854726] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 0.855091] usb 1-1: Product: VMware Virtual USB Mouse
[ 0.855092] usb 1-1: Manufacturer: VMware
[ 0.866258] input: VMware VMWare Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb1/1-1/1-1:1.0/0003:0E0F:0003.0001/input/input5
[ 0.867251] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMWare Virtual USB Mouse] on usb-0000:02:00.0-1/input0
[ 0.982026] usb 1-2: new full-speed USB device number 3 using uhci_hcd
[ 1.131966] usb 1-2: New USB device found, idVendor=0e0f, idProduct=0002, bcdDevice= 1.00
[ 1.132177] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1.132354] usb 1-2: Product: VMware Virtual USB Hub
[ 1.132533] usb 1-2: Manufacturer: VMware, Inc.
[dave@localhost ~]$
```



Using journalctl on systemd

- Another tool available on systemd Linux systems is journalctl with the ‘-k’ flag.
- Unlike dmesg, journalctl should be run as super user.
- `$ sudo journalctl -k | grep -i usb`



```
Dec 23 13:42:59 localhost.localdomain kernel: usbcore: registered new interface driver usbserial_generic
Dec 23 13:42:59 localhost.localdomain kernel: usbserial: USB Serial support registered for generic
Dec 23 13:42:59 localhost.localdomain kernel: ehci-pci 0000:02:01.0: new USB bus registered, assigned bus number 2
Dec 23 13:42:59 localhost.localdomain kernel: ehci-pci 0000:02:01.0: USB 2.0 started, EHCI 1.00
Dec 23 13:42:59 localhost.localdomain kernel: usb usb2: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 4.18
Dec 23 13:42:59 localhost.localdomain kernel: usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
Dec 23 13:42:59 localhost.localdomain kernel: usb usb2: Product: EHCI Host Controller
Dec 23 13:42:59 localhost.localdomain kernel: usbcore: registered new interface driver usbhid
Dec 23 13:42:59 localhost.localdomain kernel: usb usb2: Manufacturer: Linux 4.18.0-513.9.1.el8_9.x86_64 ehci_hcd
Dec 23 13:42:59 localhost.localdomain kernel: usbhid: USB HID core driver
Dec 23 13:42:59 localhost.localdomain kernel: usb usb2: SerialNumber: 0000:02:01.0
Dec 23 13:42:59 localhost.localdomain kernel: hub 2-0:1.0: USB hub found
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-1: new full-speed USB device number 2 using uhci_hcd
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-1: New USB device found, idVendor=0e0f, idProduct=0003, bcdDevice= 1.03
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-1: Product: VMware Virtual USB Mouse
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-1: Manufacturer: VMware
Dec 23 13:42:59 localhost.localdomain kernel: input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb1/1-1/1-1:1.0/0003:0E0F:0003.0001/input/input5
Dec 23 13:42:59 localhost.localdomain kernel: hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
Dec 23 13:42:59 localhost.localdomain kernel: usb 1-2: new full-speed USB device number 3 using uhci_hcd
Dec 23 13:43:00 localhost.localdomain kernel: usb 1-2: New USB device found, idVendor=0e0f, idProduct=0002, bcdDevice= 1.00
Dec 23 13:43:00 localhost.localdomain kernel: usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
Dec 23 13:43:00 localhost.localdomain kernel: usb 1-2: Product: VMware Virtual USB Hub
Dec 23 13:43:00 localhost.localdomain kernel: usb 1-2: Manufacturer: VMware, Inc.
Dec 23 13:43:00 localhost.localdomain kernel: hub 1-2:1.0: USB hub found
[dave@localhost ~]$
```



List Hardware with lshw

- lshw lists the hardware on the system and should be run as super user.
- \$ sudo lshw | more
- \$ sudo lshw > lshw-log.txt

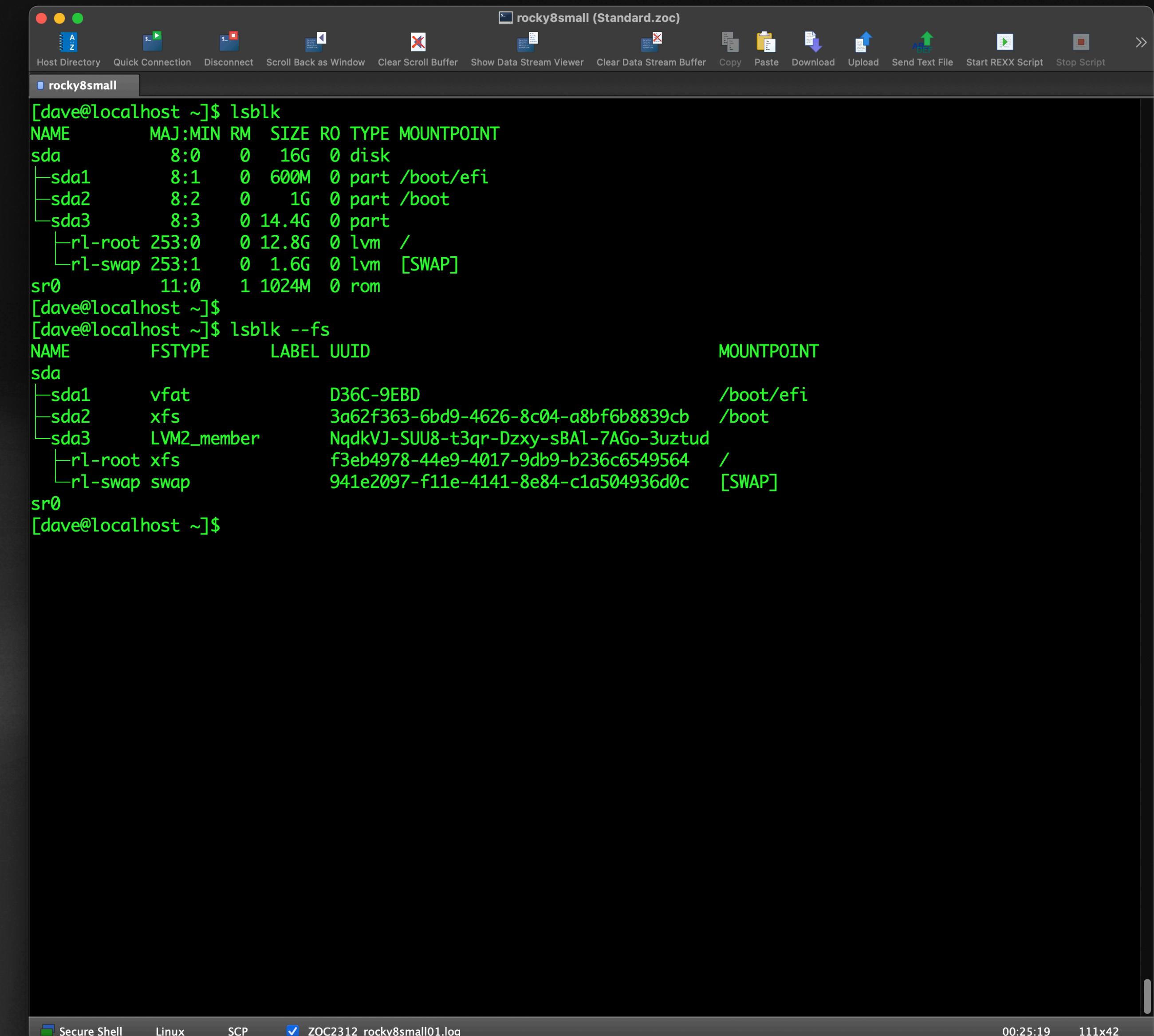
```
physical id: 34
slot: RAM slot #52
*-bank:53
    description: DIMM [empty]
    physical id: 35
    slot: RAM slot #53
*-bank:54
    description: DIMM [empty]
    physical id: 36
    slot: RAM slot #54
*-bank:55
    description: DIMM [empty]
    physical id: 37
    slot: RAM slot #55
*-bank:56
    description: DIMM [empty]
    physical id: 38
    slot: RAM slot #56
*-bank:57
    description: DIMM [empty]
    physical id: 39
    slot: RAM slot #57
*-bank:58
    description: DIMM [empty]
    physical id: 3a
    slot: RAM slot #58
*-bank:59
    description: DIMM [empty]
    physical id: 3b
    slot: RAM slot #59
*-bank:60
    description: DIMM [empty]
    physical id: 3c
    slot: RAM slot #60
*-bank:61
    description: DIMM [empty]
    physical id: 3d
    slot: RAM slot #61
*-bank:62
    description: DIMM [empty]
    physical id: 3e
--More--
```

Secure Shell Linux SCP ZOC2312_rocky8small01.log 00:24:04 111x42



List Block Devices

- `lsblk` displays a list of block storage devices, e.g., hard drives, SSDs, etc.
- When run with the '`-fs`' flag, it includes information about mount points.



The screenshot shows a terminal window titled "rocky8small (Standard.zoc)" with the command [dave@localhost ~]\$ lsblk. The output lists block devices with their major/minor numbers, sizes, types, and mount points. It then runs [dave@localhost ~]\$ lsblk --fs, which adds columns for FSTYPE, LABEL, and UUID, and includes mount points in the MOUNTPOINT column. The terminal also shows the file "ZOC2312_rocky8small01.log" is checked.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	600M	0	part	/boot/efi
└─sda2	8:2	0	1G	0	part	/boot
└─sda3	8:3	0	14.4G	0	part	
└─rl-root	253:0	0	12.8G	0	lvm	/
└─rl-swap	253:1	0	1.6G	0	lvm	[SWAP]
sr0	11:0	1	1024M	0	rom	

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
sda				
└─sda1	vfat		D36C-9EBD	/boot/efi
└─sda2	xfs		3a62f363-6bd9-4626-8c04-a8bf6b8839cb	/boot
└─sda3	LVM2_member		NqdkVJ-SUU8-t3qr-Dzxy-sBAL-7AGo-3uztud	
└─rl-root	xfs		f3eb4978-44e9-4017-9db9-b236c6549564	/
└─rl-swap	swap		941e2097-f11e-4141-8e84-c1a504936d0c	[SWAP]
sr0				

[dave@localhost ~]\$

Secure Shell Linux SCP ZOC2312_rocky8small01.log 00:25:19 111x42



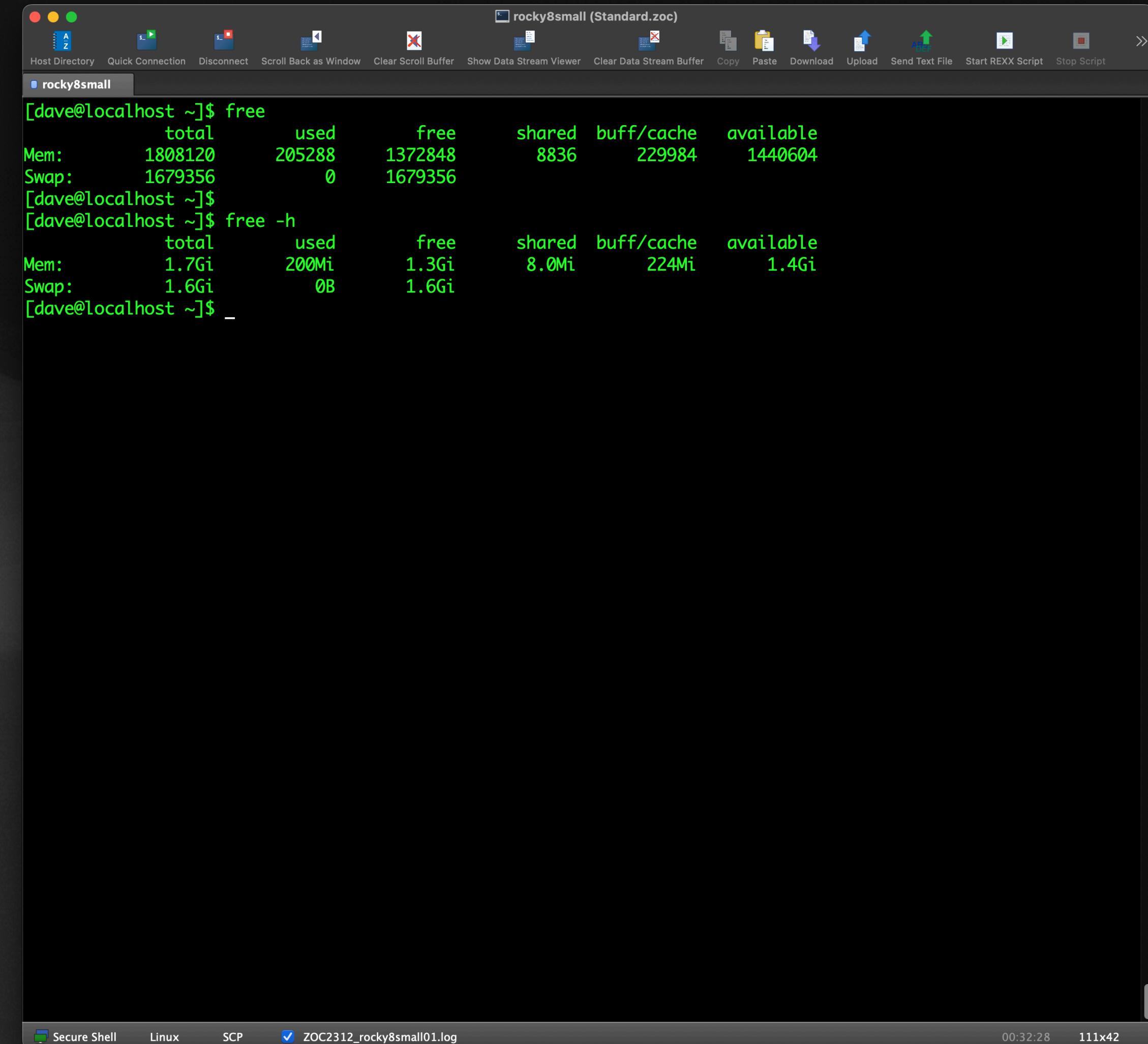
Other Commands

- `lscpu`: Lists CPU information.
- `lspci`: Lists PCI devices. Part of the `pciutils` package.
- `lsusb`: Lists USB devices. Part of the `usbutils` package.
- `lsscsi`: Lists SCSI and SATA devices.



Check Memory Usage with free

- free displays current memory usage.
- \$ free
- \$ free -h



The screenshot shows a terminal window titled "rocky8small (Standard.zoc)" with the following memory usage output:

```
[dave@localhost ~]$ free
              total        used        free      shared  buff/cache   available
Mem:       1808120      205288     1372848       8836      229984     1440604
Swap:      1679356          0     1679356
[dave@localhost ~]$
[dave@localhost ~]$ free -h
              total        used        free      shared  buff/cache   available
Mem:      1.7Gi      200Mi      1.3Gi      8.0Mi      224Mi      1.4Gi
Swap:      1.6Gi          0B      1.6Gi
[dave@localhost ~]$ _
```

The terminal also shows the status bar at the bottom with "Secure Shell", "Linux", "SCP", and "ZOC2312_rocky8small01.log".



Learning More About These Commands

- As with most Linux commands you can get information on more options and full capabilities by consulting the man page.
- \$ `man lspci`
- Or, google the man page so you can view the results outside your terminal.

The screenshot shows a web browser window with a dark theme. The address bar contains the query "man lspci - Google Search". Below the address bar, there's a navigation bar with links for Images, Videos, News, Perspectives, Command line, Example, Commands, Ubuntu, and Shopping. The main content area displays several search results for "lspci(8) - Linux manual page".

- man7**
https://man7.org › man-pages › man8 › lspci.8.html
lspci(8) - Linux manual page
lspci is a utility for displaying information about PCI buses in the system and devices connected to them. By default, it shows a brief list of devices. Use the ...
- Die.net**
https://linux.die.net › man › lspci
lspci(8): all PCI devices - Linux man page
lspci is a utility for displaying information about PCI buses in the system and devices connected to them. By default, it shows a brief list of devices.
- phoenixNAP**
https://phoenixnap.com › Home › SysAdmin
lspci Command: What Is It and How to Use It
Mar 10, 2022 — The **lspci** (list PCI) Linux command displays information about each PCI bus on your system. This includes information about the devices connected ...
- Stanford University**
http://ccrma.stanford.edu › man › man8 › lspci.8.html
lspci(8)
lspci is a utility for displaying information about all PCI buses in the system and all devices connected to them. To make use of all the features of this ...



Conclusion

- Even without a graphical interface, Linux provides several powerful tools that are usable from the shell.
- It behooves system administrators to become comfortable with using these commands to troubleshoot hardware issues and understand the kernel's interactions with hardware components.
- By analyzing their output, you can identify potential problems, view system boot messages, and gather information about the hardware configuration of your Linux system.



Thank you for watching!

Grab a copy of this slide deck from GitHub:

<https://github.com/InformationTechWithDave>

If you liked this video please hit the like button and subscribe.

[Subscribe](#)

Don't forget to click the notification button.

 Subscribed ▾

Please share with anyone who would find this helpful.

 Share

@InformationTechWithDave

