# 101.1 Determine and configure hardware settings

Weight: 2

Candidates should be able to determine and configure fundamental system hardware.

## Objectives

* Enable and disable integrated peripherals.
* Configure systems with or without external peripherals such as keyboards.
* Differentiate between the various types of mass storage devices.
* Set the correct hardware ID for different devices, especially the boot device.
* Know the differences between coldplug and hotplug devices.
* Determine hardware resources for devices.
* Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.)
* Tools and utilities to manipulate USB devices
* Conceptual understanding of sysfs, udev, hald, dbus
* /sys
* /proc
* /dev
* modprobe
* lsmod
* lspci
* lsusb

## Find out about the hardware

### HAL

HAL is Hardware Abstraction Layer. It abstracts your hardware details from you, say any first network card will be eth0. This way Linux will see any hardware as an standard hardware and you will be able to replace the hardware easily.

Hardware Abstraction Layer همان HAL است که نامی یکسان برای تمامی سخت افزارها از یک نوع روی تمامی توزیع های لینوکس در نظر میگیرد. به طور مثال در اکثر توزیع های لینوکس ماژول اول شبکه شما eth0 نام دارد و این نام گذاری توسط hal انجام میگردد.

هدف اصلی **HAL** این است که سخت افزارهای مختلف با کارایی های مختلف و معماری های مختلف به صورت یک سیستم یکپارچه برای سیستم عامل به نمایش گذاشته شوند. HAL به این دلیل در سیستم عامل ها قرار گرفته است برای اینکه دیگر نیازی نباشد برای اینکه یک نرم افزار بتواند از یک سخت افزار استفاده کند لازم باشد در Kernel سیستم عامل دستکاری شود ، شما با HAL دیگر درگیری های سخت افزاری و معماری های مختلف سخت افزاری و هماهنگ کردن آنها با سیستم عامل و نرم افزارهایتان را نخواهید داشت. HAL قابلیت های زیر را به شما می تواند ارائه بدهد :

1. امکان استفاده حداکثری از کارایی منابع سخت افزاری
2. امکان کار کردن نرم افزارها فارق از معماری موجود در سخت افزارهای مختلف
3. امکان مستقل کردن نرم افزارها از سخت افزارها و از بردن وابستگی های آنها
4. امکان دسترسی مستقیم درایورها به سخت افزارها
5. ایجاد کردن امکانات توسعه نرم افزارهای قابل حمل یا portable

به زبانی بسیار ساده مثلا اگر موسی را به سیستم متصل میکنید HAL باعث میشود که این موس از مدلش مستقل شود

در دایرکتوری dev/ تمامی device هایی که به سیستم متصل شده اند به صورت acstract شده وجود دارند

### dbus

A line like a bus that connects all parts of the OS to each other. dbus lets different parts of the system to communicate with each other. For example, when you install a USB into your computer, dbus lets GNOME know about it. Using dbus, hardware & software can talk with each other.

همینطور که از اسمش مشخصه dBUS مثل یک اتوبوسه ! که وظیفه ی اطلاع رسانی رو داره ٬‌یعنی چی ؟! یعنی وقتی فلان دیوایس رو متصل کردید به نرم افزار میگه که فلان مثلا کیبورد به دستگاه متصل شد میتونی ازش استفاده کنی ...

کاربرد **dbus** چی هست ؟ برای اینکه پاسخ این سئوال رو پیدا کنید از خودتون یک سئوال بپرسید ، آیا همه نرم افزارها از جمله سیستم عامل و application های کاربردی سیستم باید مستقیما با دایرکتوری های dev یا proc یا sys کار کنن ؟ طبیعتا خیر نیازی نیست ، این بین ما میایم یک واسط قرار میدیم که نقش دلال رو داره و واسطه بین همه این مجموعه ها و نرم افزارها هست ، الان فرض کنید که یک سخت افزار به سیستم شما اضافه میشه و شما از یک دسکتاپ به نام KDE استفاده می کنید ، این dbuss هست که میره اطلاعات مربوط به سخت افزار رو از لایه های پایین دریافت می کنه و سیگنال اینکار رو به سمت Desktop Managerr میفرسته و شما از محیط گرافیکی متوجه میشید که خوب یک سخت افزار جدید به سیستم اضافه شده یا حذف شده ، هر موقع اسم dbus رو شنیدید دقیقا یاد یک اتوبوس بیوفتید که قرار هست مسافرانش رو به مقصد برسونه و به محل کار مسافراش هم میگه کاری نداشته باشید مسافراتون از کجا میان شما باهاشون کار داشتید من آوردمشون ، کاری نداشته باشید هم می برمشون .

#### udev

Supplies the software with the events and access info of devices and can handle rules.

There are a lot of devices in /dev/ and if you plugin any device, it will have a file in /dev (say /dev/sdb2). udev lets you control what will be what in /dev. For example, you can use a rule to force your 8GB flash drive with one specific vendor to be /dev/mybackup all the time or you can tell it to copy all photos to your home directory as soon as your camera is connected.

یودو یه قانون گذار به تمام معناست ٬ چیزی که وظیفه ی قانون گذاری روی دستگاه های سخت افزاری رو داره . مثلا شما یک فلش به دستگاه وصل کردی با udev میتونی بهش دستور بدی که هر وقت مثلا این فلش وصل شد فایل های درایو D رو فقط عکس هاش رو کپی کن داخل همین فلش مخصوص ! خیلی هم باحاله !

### sysfs

The /sys directory is where HAL keeps its database of everything connected to the system.

jadi@funlife:~$ ls /sys

block bus class dev devices firmware fs hypervisor kernel module power

All block devices are at the block and bus directory has all the connected PCI, USB, serial, .. devices. Note that here in sys we have the devices based on their technology but /dev/ is abstracted.

یک دایرکتوری که اطلاعات HAL رو ذخیره میکنه و HAL وظیفه کنترل کردن sys رو داره . مثلا شما میخوایید یک آمار از سیستمتون بگیرید ببینید چه سخت افزارهایی متصل هستش میتونید با دایرکتوری sys کل سخت افزارهای بخش های مختلفی که متصل هستند رو دربیارید . sysfs هم در حقیقت مخفف sys file system هستش که کاربرد نداره و همون از sys باید استفاده بشه .

### proc directory

This is where kernel keeps its settings and properties. This directory is created on ram and files might have write accessible.

مکانی برای ذخیره اطلاعات پردازنده ها هست و شما با استفاده از دایرکتوری proc میتونید اطلاعات پردازنده هاتون رو داشته باشید .

البته این مفهوم خیلی خیلی قدیمی هست و برمیگرده به سالهای 1990 و همون حدودها بنابراین یه مقدار دیدتون رو قدیمیتر کنید ، خوب در اون وهله زمانی ما مجبور بودیم دونه دونه پردازش ها یا Process های سیستم رو مدیریت کنیم ، خوب قبل از اینکه ادامه بدیم یک یادآوری هم بکنیم ، همه چیز در سیستم عامل لینوکس و خانواده یونیکس در قالب فایل هست و procfs هم همین هست یعنی یک فایل هست که داخلش مدیریت پردازش های CPU ما انجام میشه ، وقتی یک دستگاه یه سیستم ما متصل میشه داخل این فایل محتوایی قرار می گیره که باعث میشه ما بتونیم پردازش های مربوط به اون سخت افزار رو انجام بدیم ، خوب با توجه به حرف خودمون که همه چیز در قالب فایل ذخیره میشه بنابراین محتویات sysfs هم باید داخل یک فایل و یک فولدر قرار بگیرند که این محتویات در دایرکتوری به نام proc/ در سیستم عامل لینوکس وجود داره ، توجه کنید که proc/ یک فایل سیستم هست .

$ ls /proc/

1 1249 1451 1565 18069 20346 2426 2765 2926 3175 3317 3537 39 468 4921 53 689 969 filesystems misc sysvipc

10 13 146 157 18093 20681 2452 2766 2929 3183 3318 354 397 4694 4934 538 7 97 fs modules timer\_list

1039 1321 147 1572 18243 21 2456 28 2934 3187 34 3541 404 4695 4955 54 737 acpi interrupts mounts timer\_stats

10899 13346 148 1576 18274 21021 2462 2841 2936 3191 3450 3550 41 47 4970 546 74 asound iomem mtrr tty

10960 13438 14817 158 1859 21139 25 2851 2945 32 3459 357 42 4720 4982 55 742 buddyinfo ioports net uptime

11 13619 149 16 18617 2129 2592 2852 2947 3202 3466 36 43 4731 4995 551 75 bus irq pagetypeinfo version

11120 13661 15 1613 18781 214 26 2862 2948 3206 3467 3683 44 4756 5 56 77 cgroups kallsyms partitions version\_signature

11145 13671 150 1630 1880 215 27 2865 2952 3208 3469 3699 4484 4774 50 577 8 cmdline kcore sched\_debug vmallocinfo

1159 13927 151 1633 1882 2199 2707 2866 2955 3212 3470 37 4495 4795 5008 5806 892 consoles keys schedstat vmstat

1163 14 1512 1634 19 22 2708 2884 2957 3225 3474 3710 45 48 5013 60 9 cpuinfo key-users scsi zoneinfo

1164 14045 1515 1693 19061 2219 2709 2887 2961 3236 3475 3752 4506 4811 5077 61 904 crypto kmsg self

1170 14047 152 17 19068 23 2710 2891 3 324 3477 3761 4529 4821 5082 62 9061 devices kpagecount slabinfo

1174 14052 153 17173 19069 23055 2711 2895 3047 3261 3517 3778 4558 484 5091 677 915 diskstats kpageflags softirqs

12 1409 154 1732 19075 2354 2718 29 3093 3284 3522 38 4562 4861 51 678 923 dma loadavg stat

1231 1444 155 17413 2 2390 2719 2904 31 3287 3525 3803 46 4891 52 679 939 driver locks swaps

1234 1446 156 17751 20 24 2723 2908 3132 3298 3528 3823 4622 49 5202 680 940 execdomains mdstat sys

1236 145 1563 18 2028 2418 2763 2911 3171 33 3533 3845 4661 4907 525 687 96 fb meminfo sysrq-trigger

The numbers are the process IDs! There are also other files like cpuinfo, mounts, meminfo, ...

$ cat /proc/cpuinfo

processor : 0

vendor\_id : GenuineIntel

cpu family : 6

model : 42

model name : Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz

stepping : 7

microcode : 0x15

cpu MHz : 3195.312

cache size : 3072 KB

physical id : 0

siblings : 4

core id : 0

cpu cores : 2

apicid : 0

initial apicid : 0

fpu : yes

fpu\_exception : yes

cpuid level : 13

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx rdtscp lm constant\_tsc arch\_perfmon pebs bts nopl xtopology nonstop\_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds\_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4\_1 sse4\_2 x2apic popcnt tsc\_deadline\_timer aes xsave avx lahf\_lm ida arat epb xsaveopt pln pts dtherm tpr\_shadow vnmi flexpriority ept vpid

bogomips : 4983.79

clflush size : 64

cache\_alignment : 64

address sizes : 36 bits physical, 48 bits virtual

power management:

processor : 1

vendor\_id : GenuineIntel

cpu family : 6

model : 42

model name : Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz

stepping : 7

microcode : 0x15

cpu MHz : 3010.839

cache size : 3072 KB

physical id : 0

siblings : 4

core id : 0

cpu cores : 2

apicid : 1

initial apicid : 1

fpu : yes

fpu\_exception : yes

cpuid level : 13

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx rdtscp lm constant\_tsc arch\_perfmon pebs bts nopl xtopology nonstop\_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds\_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4\_1 sse4\_2 x2apic popcnt tsc\_deadline\_timer aes xsave avx lahf\_lm ida arat epb xsaveopt pln pts dtherm tpr\_shadow vnmi flexpriority ept vpid

bogomips : 4983.79

clflush size : 64

cache\_alignment : 64

address sizes : 36 bits physical, 48 bits virtual

power management:

We can also write here. Since I'm on an IBM Lenovo laptop I can turn my LED on and off by writing here:

root@funlife:/proc/acpi/ibm# echo on > light

root@funlife:/proc/acpi/ibm# echo off > light

One more traditional example is changing the max number of open files per user:

root@funlife:/proc/sys/fs# cat file-max

797946

root@funlife:/proc/sys/fs# echo 1000000 > file-max

root@funlife:/proc/sys/fs# cat file-max

1000000

Another very useful directory here, is /proc/sys/net/ipv4 which controls real time networking configurations.

All these changes will be reverted after a boot. You have to write into config files in /etc/ to make these changes permanent

### dev

udev controls /dev/ directory. There are abstracted devices like a hard, is /dev/sda or /dev/hd0 regardless of its brand, model or technology:

root@funlife:/dev# ls /dev/sda\*

/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sda5 /dev/sda6

### lsmod, lsusb, lspci

These commands show list of modules and hardwares on the system.

#### lsmod

Shows kernel modules.

root@funlife:/dev# lsmod

Module Size Used by

pci\_stub 12622 1

vboxpci 23256 0

vboxnetadp 25670 0

vboxnetflt 27605 0

vboxdrv 418013 3 vboxnetadp,vboxnetflt,vboxpci

ctr 13049 3

ccm 17731 3

dm\_crypt 23172 1

bnep 19543 2

rfcomm 69509 8

uvcvideo 81065 0

arc4 12608 2

videobuf2\_vmalloc 13216 1 uvcvideo

intel\_rapl 18783 0

iwldvm 236430 0

x86\_pkg\_temp\_thermal 14205 0

intel\_powerclamp 18786 0

btusb 32448 0

videobuf2\_memops 13362 1 videobuf2\_vmalloc

videobuf2\_core 59104 1 uvcvideo

v4l2\_common 15682 1 videobuf2\_core

mac80211 660592 1 iwldvm

coretemp 13441 0

videodev 149725 3 uvcvideo,v4l2\_common,videobuf2\_core

media 21963 2 uvcvideo,videodev

bluetooth 446190 22 bnep,btusb,rfcomm

kvm\_intel 143592 0

kvm 459835 1 kvm\_intel

snd\_hda\_codec\_hdmi 47547 1

crct10dif\_pclmul 14307 0

6lowpan\_iphc 18702 1 bluetooth

crc32\_pclmul 13133 0

snd\_hda\_codec\_conexant 23064 1

ghash\_clmulni\_intel 13230 0

snd\_hda\_codec\_generic 68914 1 snd\_hda\_codec\_conexant

aesni\_intel 152552 10

snd\_seq\_midi 13564 0

snd\_seq\_midi\_event 14899 1 snd\_seq\_midi

aes\_x86\_64 17131 1 aesni\_intel

mei\_me 19742 0

lrw 13287 1 aesni\_intel

iwlwifi 183038 1 iwldvm

These are the kernel modules which are loaded.

If you need to add a module to your kernel (say a new driver for a hardware) or remove it (uninstall a driver) you can use rmmod and modprobe.

# rmmod iwlwifi

And this is for installing the module:

# insmod kernel/drivers/net/wireless/lwlwifi.ko

but nobody uses insmod because it does not understand dependencies and you need to give it the whole path to the module file. Instead, use the modprobe command:

# modprobe iwlwifi

you can use -f switch to FORCE rmmod to remove the module even if it is in use.

If you need to load some modules every time your system boots, do one of the following:

1. add their names to this file /etc/modules
2. add their configs files to the /etc/modprobe.d/

ls مخفف list size و بخش بعدی mod مخفف ماژول module هستش . یعنی لیست ماژول های که کرنل (هسته لینوکس) لود کرده است .  
برای حذف کردن Drive Kernel Modulee ها ما می توانیم به راحتی با لیست کردن ماژول های کرنل با دستور lsmod و مشاهده نام آن ماژول و در نهایت وارد کردن دستور rmmod آن ماژول را حذف کنیم ، اما چگونه می توانیم آن را نصب کنیم ؟ برای نصب کردن Drive Kernel Module ما از دو دستور insmod و modprobe استفاده می کنیم ، اما مهمترین نکته تفاوت بین این دو دستور است ، زمانیکه صحبت از insmod می شود فقط و فقط درایورهایی نصب می شوند که چند فایل وابسته ندارند ، یعنی در اصطلاح فایل شما دارای Dependency نیست ، در چنین حالتی دستور insmod کار می کند ، اما در بسیاری از موارد ما هیچوقت از چنین دستوری استفاده نمی کنیم زیرا فاقد کارایی می باشد و اکثر درایورهای ما امروزه دارای وابستگی یا Dependency هستند ، برای اینکه در سیستم بصورت خودکار این وابستگی ها یا Dependency ها نصب شوند ما از دستور modprobe استفاده می کنیم .

#### lspci

Shows PCI devices that are connected to the computer.

# lspci

00:00.0 Host bridge: Intel Corporation 2nd Generation Core Processor Family DRAM Controller (rev 09)

00:02.0 VGA compatible controller: Intel Corporation 2nd Generation Core Processor Family Integrated Graphics Controller (rev 09)

00:16.0 Communication controller: Intel Corporation 6 Series/C200 Series Chipset Family MEI Controller #1 (rev 04)

00:19.0 Ethernet controller: Intel Corporation 82579LM Gigabit Network Connection (rev 04)

00:1a.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB Enhanced Host Controller #2 (rev 04)

00:1b.0 Audio device: Intel Corporation 6 Series/C200 Series Chipset Family High Definition Audio Controller (rev 04)

00:1c.0 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Express Root Port 1 (rev b4)

00:1c.1 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Express Root Port 2 (rev b4)

00:1c.4 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI Express Root Port 5 (rev b4)

00:1d.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB Enhanced Host Controller #1 (rev 04)

00:1f.0 ISA bridge: Intel Corporation QM67 Express Chipset Family LPC Controller (rev 04)

00:1f.2 SATA controller: Intel Corporation 6 Series/C200 Series Chipset Family 6 port SATA AHCI Controller (rev 04)

00:1f.3 SMBus: Intel Corporation 6 Series/C200 Series Chipset Family SMBus Controller (rev 04)

03:00.0 Network controller: Intel Corporation Centrino Wireless-N 1000 [Condor Peak]

0d:00.0 System peripheral: Ricoh Co Ltd MMC/SD Host Controller (rev 07)

#### lsusb

Shows all the USB devices connected to the system.

# lsusb

Bus 002 Device 003: ID 1c4f:0026 SiGma Micro Keyboard

Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub

Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 001 Device 005: ID 04f2:b217 Chicony Electronics Co., Ltd Lenovo Integrated Camera (0.3MP)

Bus 001 Device 004: ID 0a5c:217f Broadcom Corp. BCM2045B (BDC-2.1)

Bus 001 Device 003: ID 192f:0916 Avago Technologies, Pte.

Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

#### lspcmcia

Shows available PCMCIA cards on this computer.

#### lshal

Shows HAL data.

#### lshw

Shows hardware. Test it!

## Device UUIDs

Each device has an ID. If you speak about /dev/sda, you are speaking about the "first hard" but if you want a specific drive to be your /home, you have to use UUID.

root@funlife:/dev# cat /proc/mounts

rootfs / rootfs rw 0 0

sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0

proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0

udev /dev devtmpfs rw,relatime,size=4014804k,nr\_inodes=1003701,mode=755 0 0

devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0

tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=806028k,mode=755 0 0

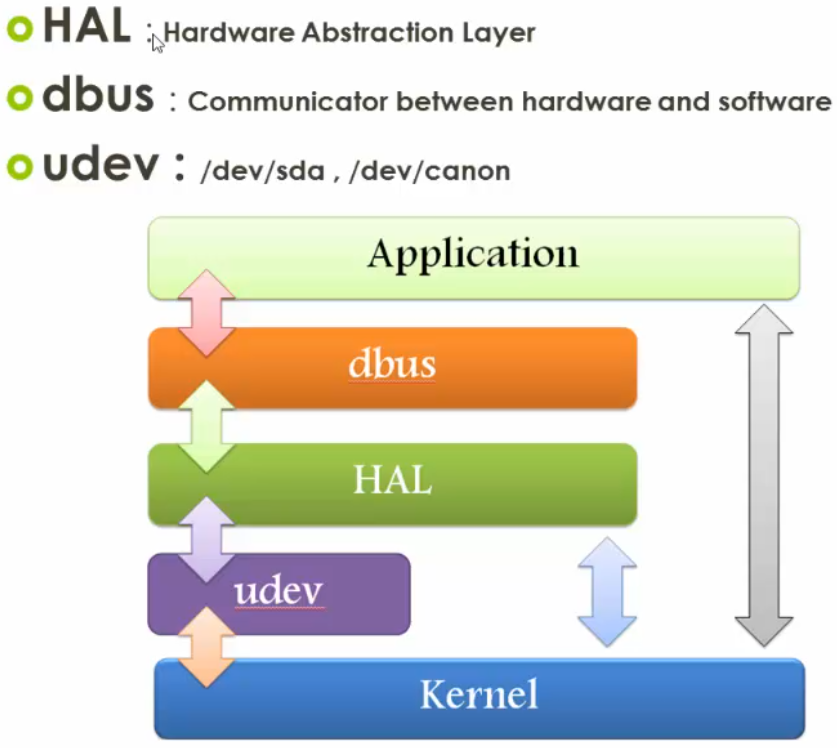
/dev/disk/by-uuid/1651a94e-0b4e-47fb-aca0-f77e05714617 / ext4 rw,relatime,errors=remount-ro,data=ordered 0 0

Every other device has its own ID which can be used to identify it.

با استفاده از دستور blkid میتوانید id تمام block device ها یعنی هاردها رو مشاهده کنید

## hotplug & coldplug

Hotplug is when you insert a hardware into a running computer and coldplug is when you have to turn your computer off to install a hardware. USB devices are hot pluggable while PCI cards should be cold-plugged.



# 101.2. Boot the system

weight: 3

Candidates should be able to guide the system through the booting process.

## Key Knowledge Areas

* Provide common commands to the boot loader and options to the kernel at boot time.
* Demonstrate knowledge of the boot sequence from BIOS to boot completion.
* Check boot events in the log files.

## Terms

* BIOS
* bootloader
* kernel
* init
* /var/log/messages
* dmesg

### BIOS

BIOS is Basic Input Output System and does the first steps of the PC bootup. For example is does a POST (Power On Self Test) and decides which hardware should boot the system.

### bootloader

Bootloader can be GRUB (1&2) or LILO which are great for disks less than 2TB.

/etc/lilo.conf

/boot/grub/grub.cfg

/boot/grub/menu.lst

### Kernel

Kernel parameters (sometimes called boot parameters) supply the kernel with information about hardware parameters that it might not determine on its own - say single user mod boot (S)

### init

When the kernel finishes loading, it usually starts /sbin/init. This program remains running until the system is shut down. It is always assigned process ID 1.

first process, process in charge, a big family tree of commands:

pstree

init is being replaced in many distros (say ubuntu with upstart) but still is in exam and has its own section.

### dmesg

Funny fact: During the bootup, only The Kernel is running so it should record and keep its own logs!

dmesg command will show the full data from kernel ring buffer up to now. But

cat /var/log/dmesg

will show only the data during the boot

### /var/log/messages

After the init process comes up, syslog daemon will log messages. It has timestamps and will persist during restarts.

* Kernel is still logging its own messages in dmesg
* in some systems it might be called /var/log/syslog
* there are many other logs at /var/log