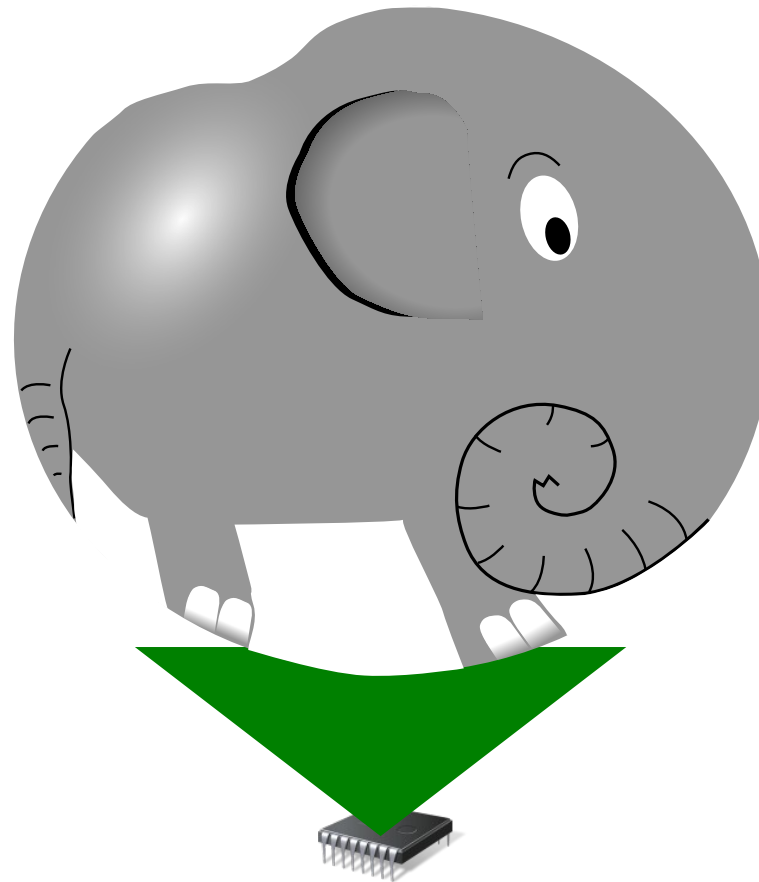


Professional Seminars II: Operating Systems for Embedded Systems

Booting *nix



R04
Ángel Perles

Contents

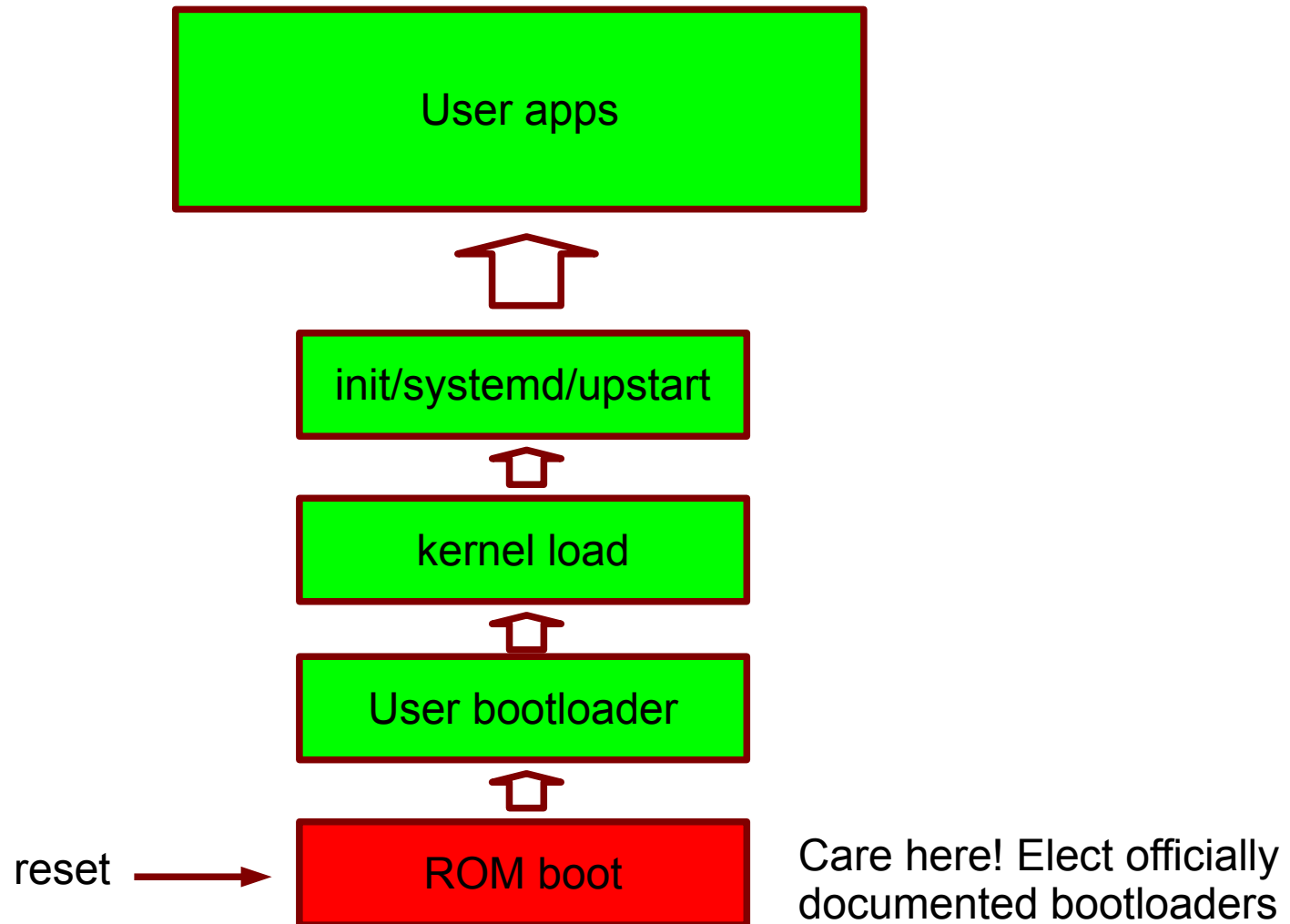
- Objectives
- Boot sequence
- Bootloader
- Linux kernel
- Root filesystem (/)

Objectives

- To understand the general booting sequence of *nix OSes
- To know the particular start-up of Linux
- To practice applying it to ARM-based boards
 - Example with Raspberry Pi board
 - The same for your iPhone, Google Android, ...

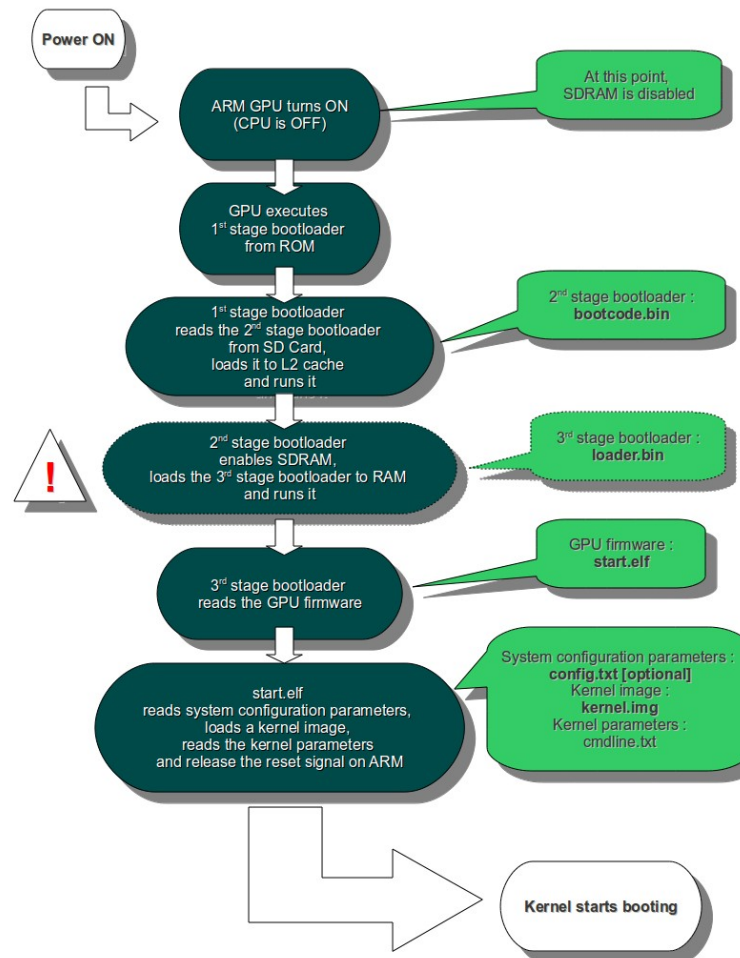
Boot sequence

- Very generic one for Linux



Boot sequence

- Raspberry Pi (old image, boot 2+ boot 3 are now only one stage)

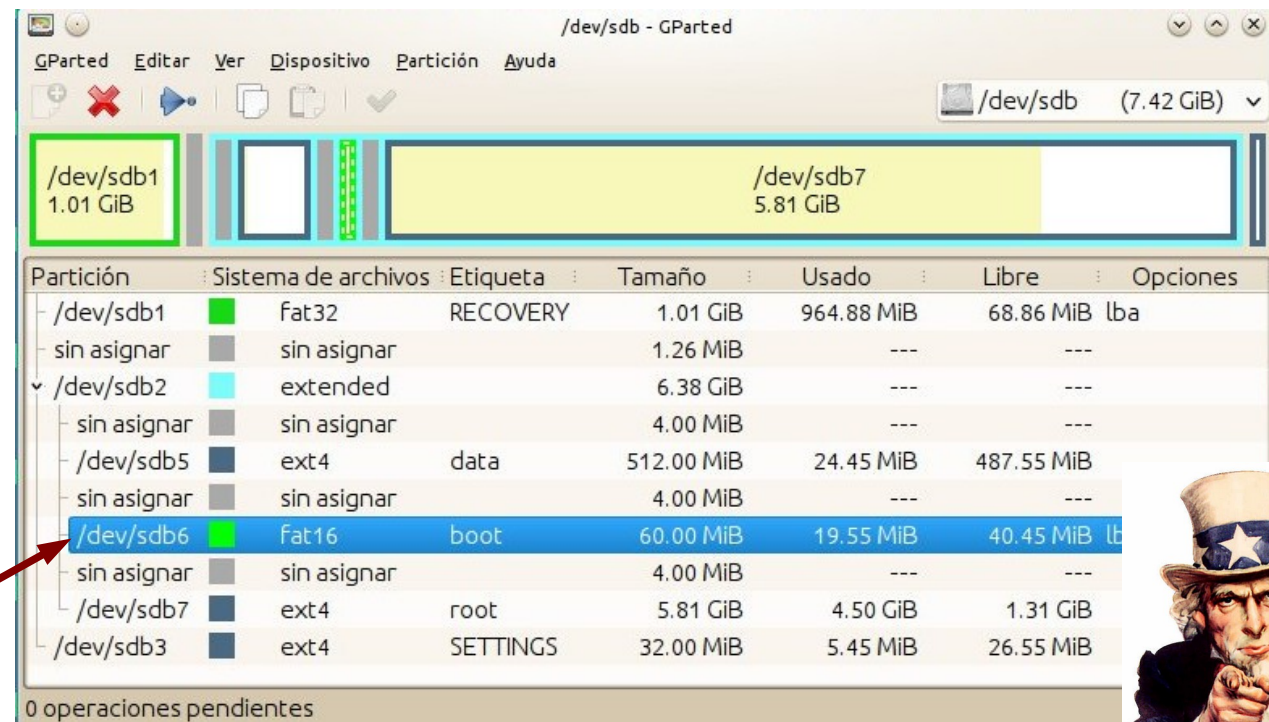


source <http://myembeddedlinux.blogspot.com.es/2013/05/raspberry-pi-boot-sequence.html>

Bootloader

- Hell! Where are these things?
 - Some things in ROM <- closed source !!!!!!!!!!!!!
 - Some in the boot partition of the SD (FAT -> pay Microsoft)

see here your
Raspberry OS
based SD partition



The screenshot shows the GParted interface for the device /dev/sdb (7.42 GiB). The partition table is as follows:

Partición	Sistema de archivos	Etiqueta	Tamaño	Usado	Libre	Opciones
/dev/sdb1	fat32	RECOVERY	1.01 GiB	964.88 MiB	68.86 MiB	lba
sin asignar	sin asignar		1.26 MiB	---	---	
/dev/sdb2	extended		6.38 GiB	---	---	
sin asignar	sin asignar		4.00 MiB	---	---	
/dev/sdb5	ext4	data	512.00 MiB	24.45 MiB	487.55 MiB	
sin asignar	sin asignar		4.00 MiB	---	---	
/dev/sdb6	fat16	boot	60.00 MiB	19.55 MiB	40.45 MiB	lba
sin asignar	sin asignar		4.00 MiB	---	---	
/dev/sdb7	ext4	root	5.81 GiB	4.50 GiB	1.31 GiB	
/dev/sdb3	ext4	SETTINGS	32.00 MiB	5.45 MiB	26.55 MiB	

0 operaciones pendientes

boot, boot

Use *gparted* to take a look at your SD



Bootloader

- Let's take a look assuming you are using a Linux box
 - Insert SD
 - Mount the boot partition
 - Look at config.txt file


Boot 1st stage →

Boot configuration →

Linux kernel →

Boot 2nd + 3rd stage →

Nombre	Fecha	Tamaño
> overlays	01-01-80 00:00	45 elementos
bcm2708-rpi-b-plus.dtb	23-09-15 16:10	9,9 KiB
bcm2708-rpi-b.dtb	23-09-15 16:10	9,6 KiB
bcm2708-rpi-cm.dtb	23-09-15 16:10	9,6 KiB
bcm2709-rpi-2-b.dtb	23-09-15 16:10	10,9 KiB
bootcode.bin	23-09-15 16:10	17,5 KiB
cmdline.txt	01-01-80 00:00	136 B
config.txt	01-01-80 00:00	1,7 KiB
COPYING.txt	21-08-15 17:04	18,3 KiB
fixup.dat	23-09-15 16:10	6,2 KiB
fixup_cd.dat	23-09-15 16:10	2,3 KiB
fixup_db.dat	23-09-15 16:10	9,4 KiB
fixup_x.dat	23-09-15 16:10	9,4 KiB
issue.txt	24-09-15 16:10	1 B
kernel.img	23-09-15 16:10	1 KiB
kernel7.img	23-09-15 16:10	1 KiB
LICENCE.broadcom	21-08-15 17:04	1 KiB
LICENSE.oracle	25-09-13 22:00	1 KiB
os_config.json	01-01-80 00:00	1 B
start.elf	23-09-15 16:10	1 KiB
start_cd.elf	23-09-15 16:10	1 KiB
start_db.elf	23-09-15 16:10	1 KiB
start_x.elf	23-09-15 16:10	1 KiB



Linux kernel

- The 3rd boot stage loads the OS
 - Microsoft Windows, RISC OS, Linux ...
 - kernel.img file is Linux!
 - and cmdline.txt the params
 - look at cmdline.txt



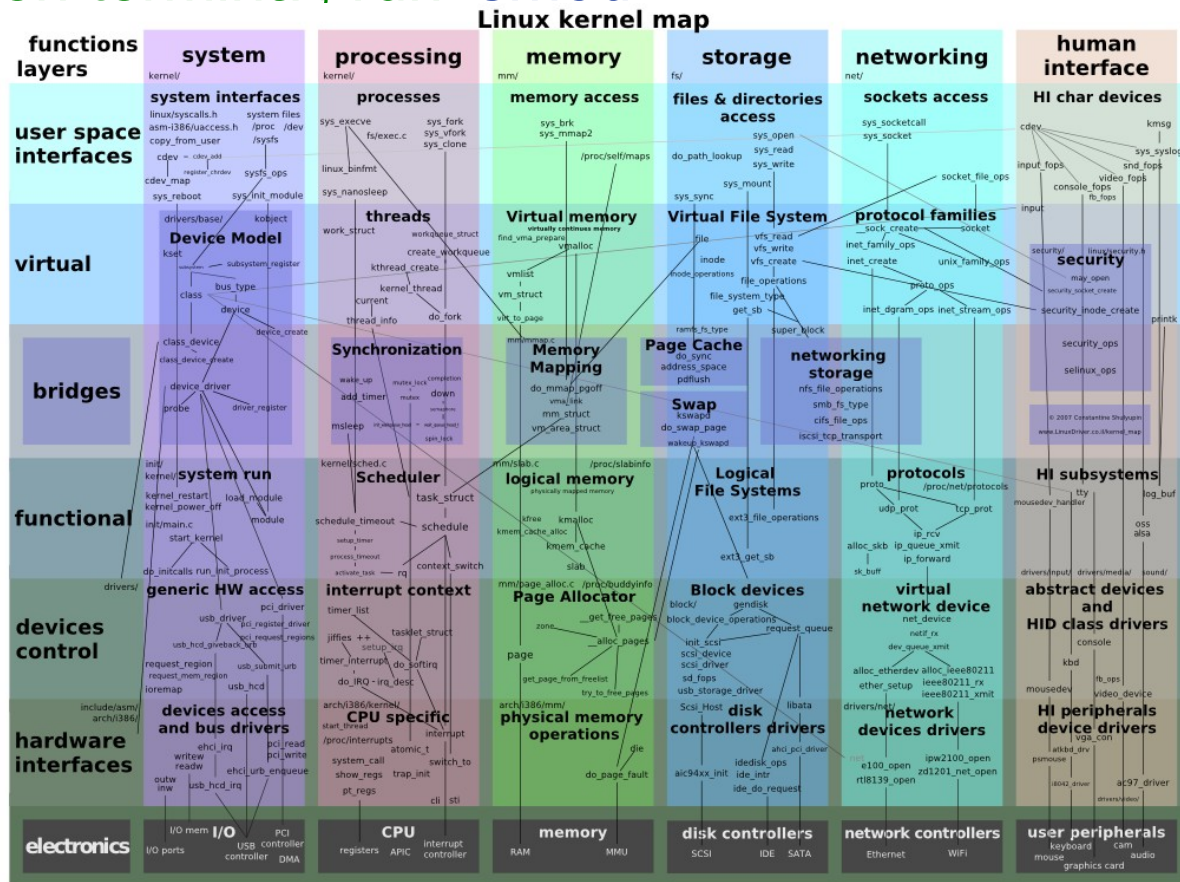
Linux kernel basic conf. →

Linux kernel →

> boot			
Nombre	Fecha	Tamaño	
> overlays	01-01-80 00:00	45 elementos	
bcm2708-rpi-b-plus.dtb	23-09-15 16:10	9,9 KiB	
bcm2708-rpi-b.dtb	23-09-15 16:10	9,6 KiB	
bcm2708-rpi-cm.dtb	23-09-15 16:10	9,6 KiB	
bcm2709-rpi-2-b.dtb	23-09-15 16:10	10,9 KiB	
bootcode.bin	23-09-15 16:10	17,5 KiB	
cmdline.txt	01-01-80 00:00	136 B	
config.txt	01-01-80 00:00	1,7 KiB	
COPYING.linux	21-08-15 17:04	18,3 KiB	
fixup.dat	23-09-15 16:10	6,2 KiB	
fixup_cd.dat	23-09-15 16:10	2,3 KiB	
fixup_db.dat	23-09-15 16:10	9,4 KiB	
fixup_x.dat	23-09-15 16:10	9,4 KiB	
issue.txt	24-09-15 16:33	137 B	
kernel.img	23-09-15 16:10	3,9 MiB	
kernel7.img	23-09-15 16:10	3,8 MiB	
LICENCE.broadcom	21-08-15 17:04	1,4 KiB	
LICENSE.oracle	25-09-13 22:57	18,5 KiB	
os_config.json	01-01-80 00:00	305 B	
start.elf	23-09-15 16:10	2,6 MiB	
start_cd.elf	23-09-15 16:10	582,8 KiB	
start_db.elf	23-09-15 16:10	4,6 MiB	
start_x.elf	23-09-15 16:10	3,6 MiB	

Linux kernel

- The Linux kernel is an static file (3rd boot stage loads the OS)
 - Monolithic + modular design: part static, part dynamic
 - open terminal, run `lsmod`

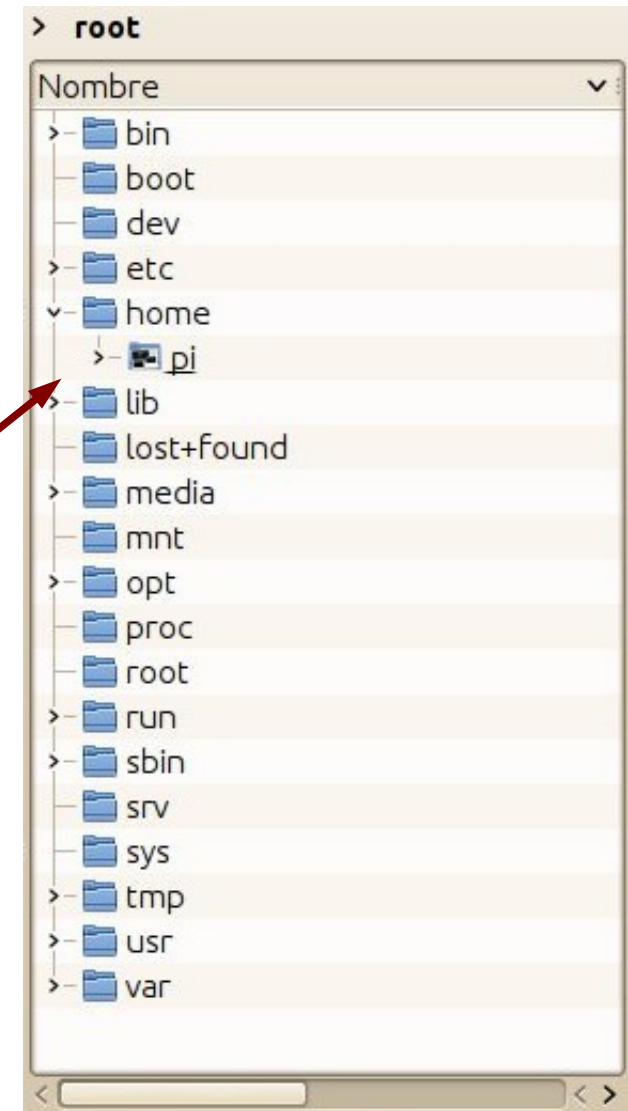


See <https://makelinux.github.io/kernel/map/>

Root filesystem (/)

- Next step is to mount the “root” (/)
 - Basic filesystem structure
 - Libraries, kernel modules, device files, ...
 - Applications (main and customizations)
 - and system initialization ...

take a look to
your SD root partition



Root filesystem (/)

- ... system initialization
 - spawn process PID 1 (init or systemd or upstart)
 - that loads other processes
 - and configurations
 - and services (e.g. check /etc/rc.d)
 - ...
 - ... till user prompt or a given app

\$ dmesg | less



Root filesystem (/)

- Raspbian based on Debian wheezy use init (sysvinit)
- Raspbian based on Debian stretch use systemd
 - parallelization
 - lots of information are now obsolete in Internet

```
[ 2.231932] mmc0: host does not support reading read-only switch, assuming write-en
able
[ 2.247779] mmc0: new high speed SDHC card at address 9149
[ 2.259437] mmcblk0: mmc0:9149 SD08G 7.42 GiB
[ 2.273583] mmcblk0: p1 p2 < p5 p6 p7 > p3
[ 2.347212] EXT4-fs (mmcblk0p7): mounted filesystem with ordered data mode. Opts: (
null)
[ 2.361122] VFS: Mounted root (ext4 filesystem) readonly on device 179:7.
[ 2.378496] devtmpfs: mounted
[ 2.387908] Freeing unused kernel memory: 416K (80776000 - 807de000)
[ 2.413539] usb 1-1: New USB device found, idVendor=0424, idProduct=9514
[ 2.426072] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 2.439934] hub 1-1:1.0: USB hub found
[ 2.449655] hub 1-1:1.0: 5 ports detected
[ 2.711384] random: systemd urandom read with 65 bits of entropy available
[ 2.729034] systemd[1]: systemd 215 running in system mode. (+PAM +AUDIT +SELINUX +
IMA +SYSVINIT +LIBCRYPTSETUP +GCRYPT +ACL +XZ -SECCOMP -APPARMOR)
[ 2.733201] usb 1-1.1: new high-speed USB device number 3 using dwc_otg
[ 2.761692] systemd[1]: Detected architecture 'arm'.
[ 2.853526] usb 1-1.1: New USB device found, idVendor=0424, idProduct=ec00
[ 2.866864] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
```




And yes, I'm aware that this is too much information.