

Linux Kernel ProCamp

Presentation

Sam Protsenko

Senior Software Engineer, Linux Kernel ProCamp Trainer

August 28, 2019

Linux Usages

Linux Usages: Non-Embedded

Non-embedded systems:

- Servers
- Desktops
- Supercomputers



Linux Usages: Embedded

Embedded Systems:

- Mobile
- Automotive
- Networking
- Smart TVs, game consoles, set-top boxes
- IoT
- Medical
- Aerospace
- Industry



Android Rise

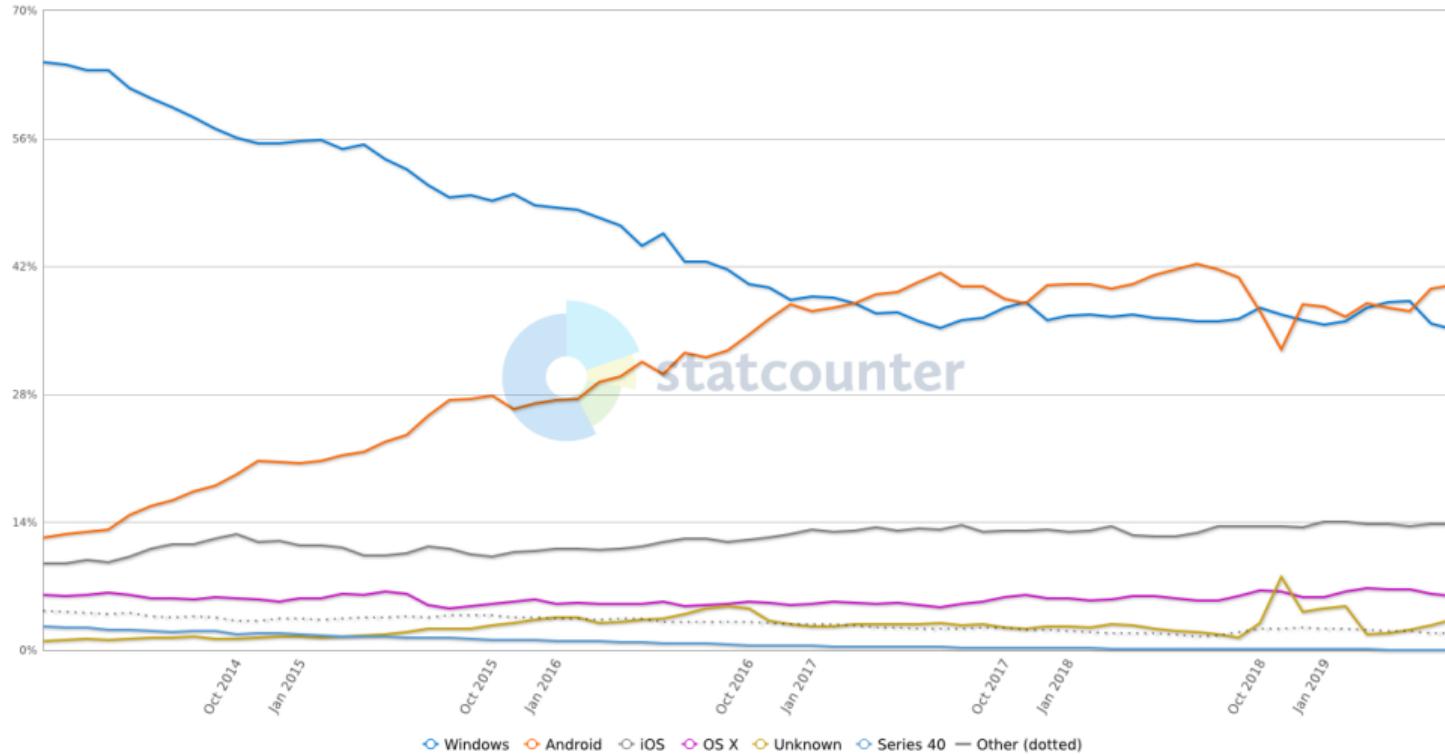


Figure 1: Operating System Market Share Worldwide

Automotive Market Size

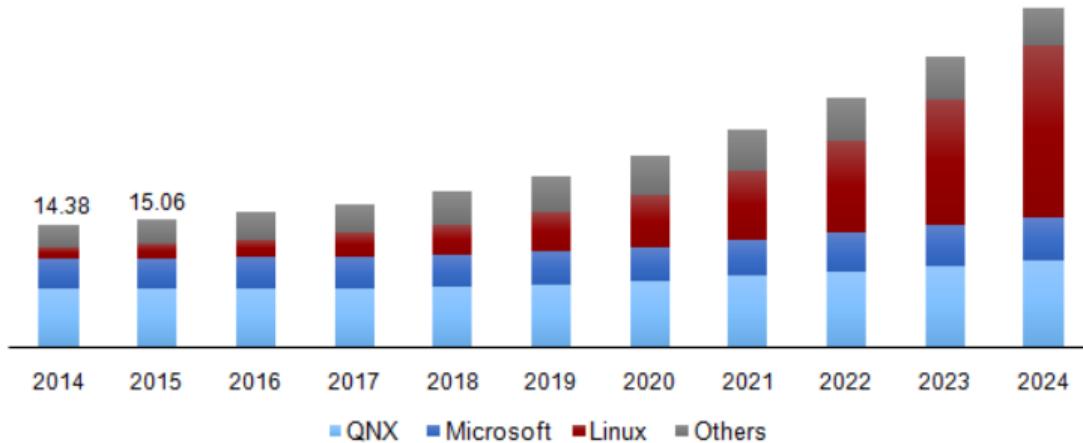
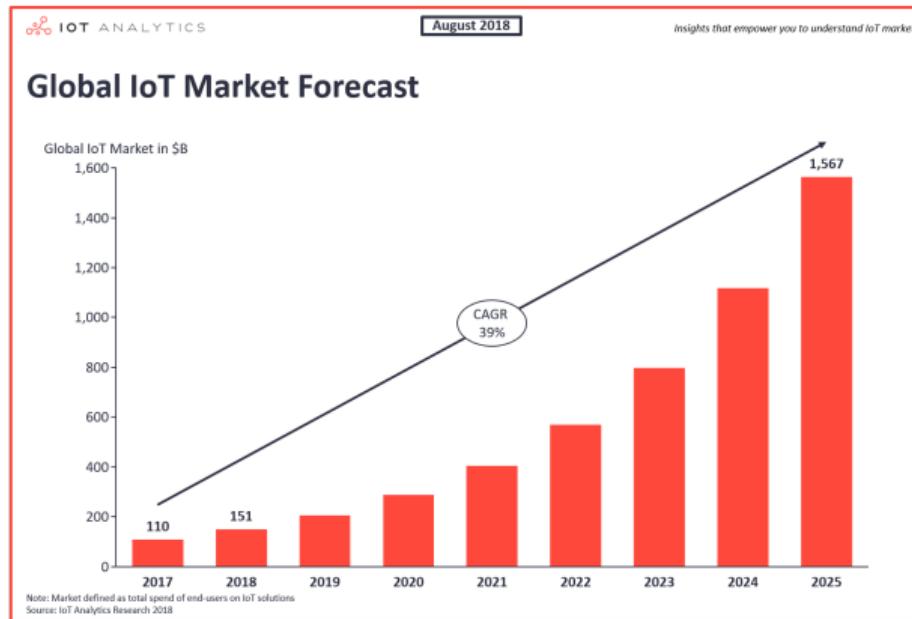


Figure 2: Global automotive infotainment market, by operating system;
(2014 - 2024, USD Billion)

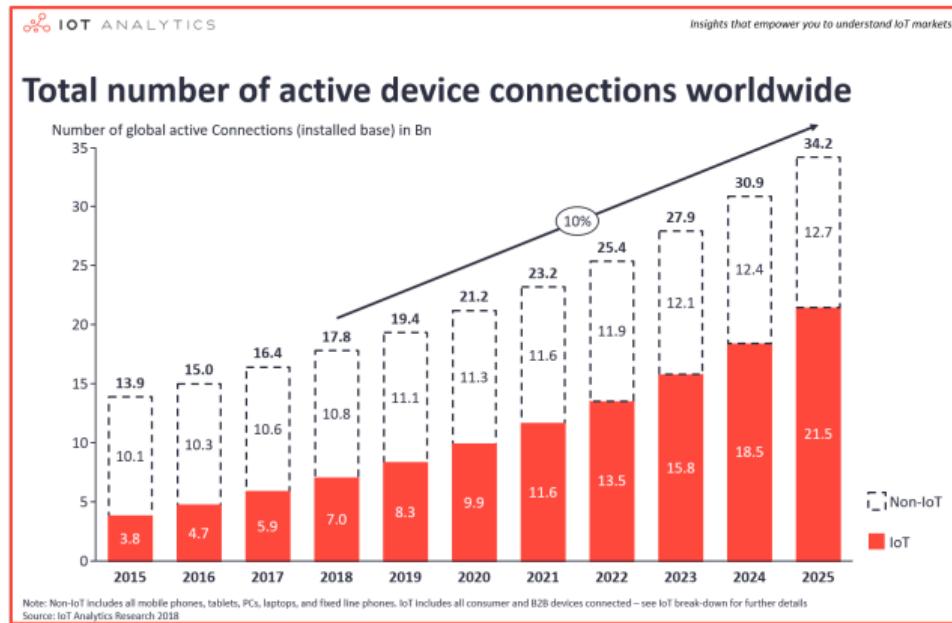
Source: <https://www.hexaresearch.com>

IoT Market Size



Source: <https://iot-analytics.com>

IoT Number of Devices



Source: <https://iot-analytics.com>

How Linux kernel fits in?

Good fit for embedded:

- Open-source (eases the collaboration)
- Supports a lot of architectures
- A lot of infrastructure exists (bootloaders, RootFS's, dev tools)
- Already widely used in Embedded world

How Linux kernel fits in?

Good fit for embedded:

- Open-source (eases the collaboration)
- Supports a lot of architectures
- A lot of infrastructure exists (bootloaders, RootFS's, dev tools)
- Already widely used in Embedded world

Recent developments:

- Reducing the kernel footprint (Nicolas Pitre)
- **PREEMPT_RT** is being merged
- Xen
- OpenDataPlane vs DPDK
- RISC-V support
- Android kernel is mostly upstreamed now

Embedded Linux Kernel Tasks

- Board bring-up (porting the kernel to a new board)
- Writing the drivers and device tree
- Migrating downstream kernel to a new version
- Bug fixing
- Boot time optimization
- Upstreaming
- Related work: bootloader, rootfs, hardware debugging

My Experience

- Implementing the driver for automotive radio chip
- NOR flash support (XIP boot)
- Updating/upstreaming MAX732x driver
- Adopting upstream kernel VPN in Android kernel
- Fixing VPN protocols in kernel
- Implementing Android features (fastboot, boot flow, etc)

Overture: Course Embedded Part

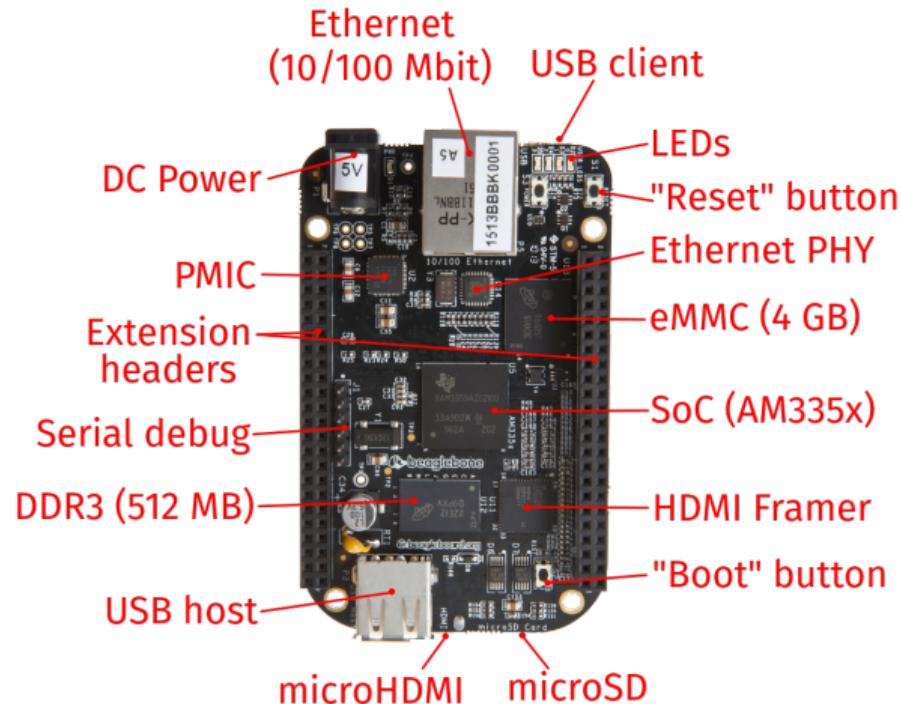
Embedded Programming

Differences from regular system:

- Cross-compiling
- Flashing
- Serial console
- Testing concerns
- Working with hardware
- Non discoverable buses on board
(device tree, platform drivers)



BeagleBone Black



AM335x SoC

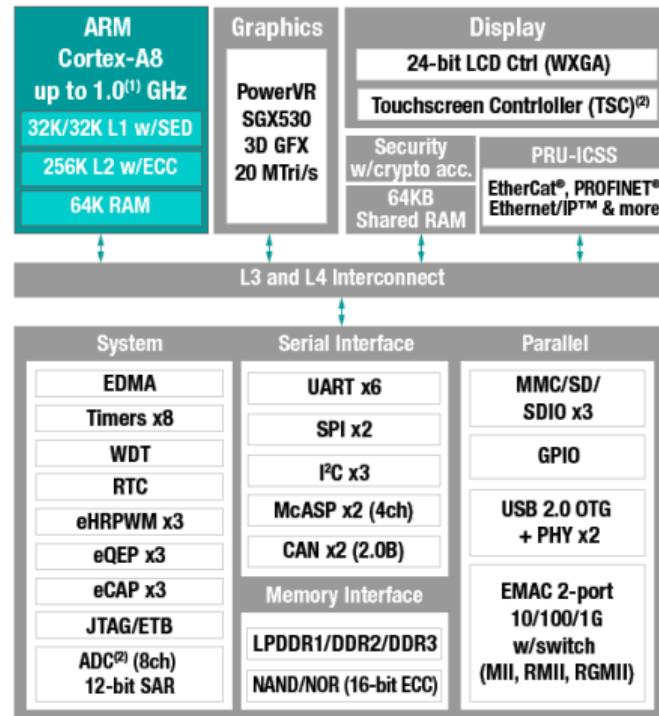


Figure 3: AM335x Functional Diagram

BeagleBone Black: Pros and Cons

Pros:

- Open Hardware
 - Public TRM
 - Schematic
 - PCB files
- Supported in upstream
 - Kernel
 - U-Boot
- Conventional ARM architecture
- Very popular
- Low cost (\$55)

Cons:

- Old 32-bit architecture
- Single core processor
- Android is not supported officially
- No WiFi

Embedded Topics

We will cover next Embedded topics during the course:

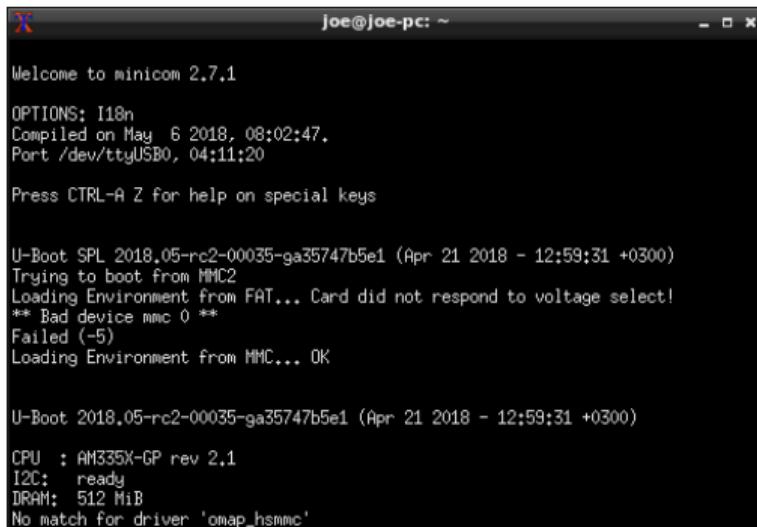
- Building, flashing and booting



Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs



The screenshot shows a terminal window titled "minicom 2.7.1" with the command "joe@joe-pc: ~". The window displays the following text:

```
Welcome to minicom 2.7.1
OPTIONS: I18n
Compiled on May 6 2018, 08:02:47.
Port /dev/ttyUSB0, 04:11:20

Press CTRL-A Z for help on special keys

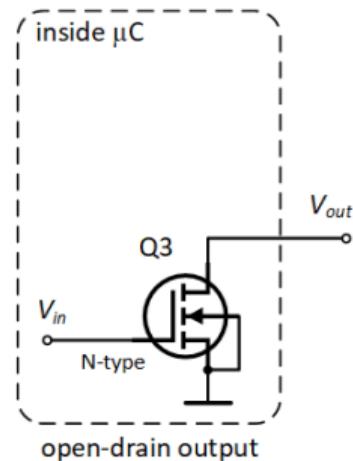
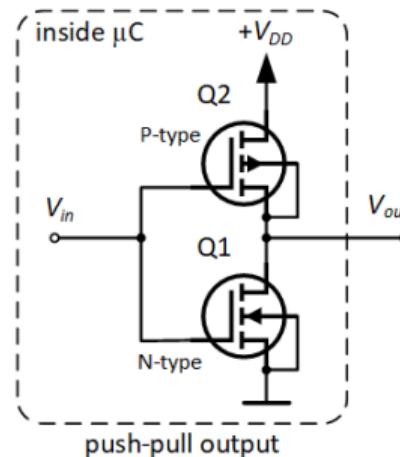
U-Boot SPL 2018.05-rc2-00035-ga35747b5e1 (Apr 21 2018 - 12:59:31 +0300)
Trying to boot from MMC2
Loading Environment from FAT... Card did not respond to voltage select!
** Bad device mmc 0 **
Failed (-5)
Loading Environment from MMC... OK

U-Boot 2018.05-rc2-00035-ga35747b5e1 (Apr 21 2018 - 12:59:31 +0300)
CPU : AM335X-GP rev 2.1
I2C: ready
DRAM: 512 MiB
No match for driver 'omap_hsmmc'
```

Embedded Topics

We will cover next Embedded topics during the course:

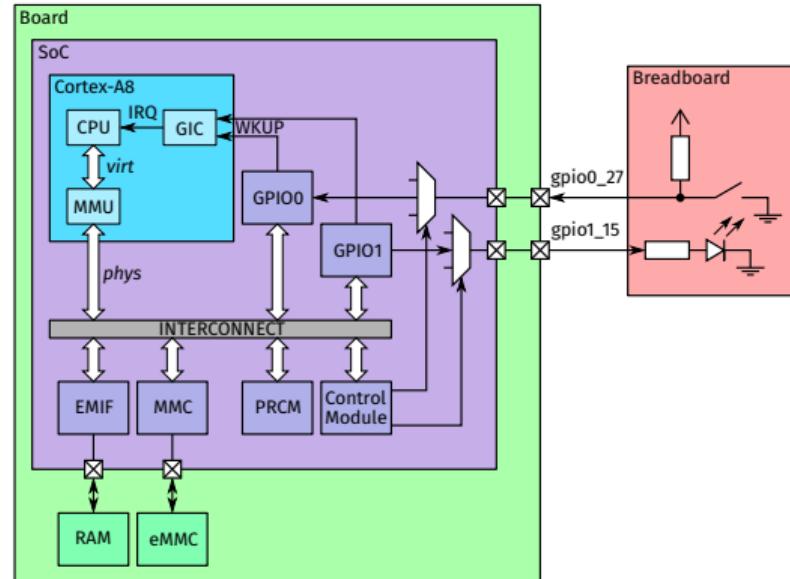
- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics



Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture

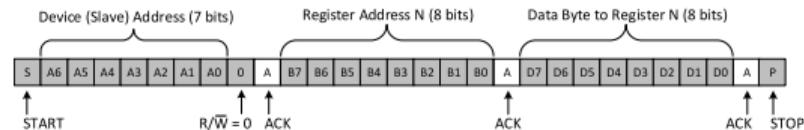


Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture
- Buses and protocols

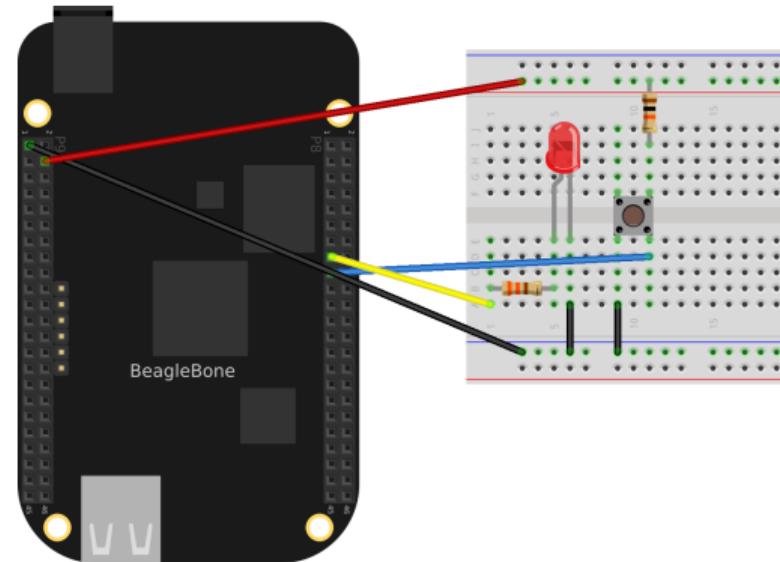
■ Master Controls SDA Line
□ Slave Controls SDA Line



Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture
- Buses and protocols
- Connecting external devices



Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture
- Buses and protocols
- Connecting external devices
- Writing the drivers

```
160 static const struct file_operations hw3_fops = {
161     .owner      = THIS_MODULE,
162     .read       = hw3_read,
163     .write      = hw3_write,
164     .poll       = hw3_poll,
165     .unlocked_ioctl = hw3_ioctl,
166     .llseek     = no_llseek,
167 };
168
169 static irqreturn_t hw3_btn_isr(int irq, void *data)
170 {
171     struct hw3 *hw3 = data;
172     unsigned long flags;
173
174     pr_err("### isr\n");
175
176     spin_lock_irqsave(&hw3->lock, flags);
177     hw3->data_ready = true;
178     hw3->btn_on = gpiod_get_value(hw3->btn_gpio);
179     if (hw3->btn_on && hw3->control) {
180         hw3->led_on ^= 0x1;
181         gpiod_set_value(hw3->led_gpio, hw3->led_on);
```

Embedded Topics

We will cover next Embedded topics during the course:

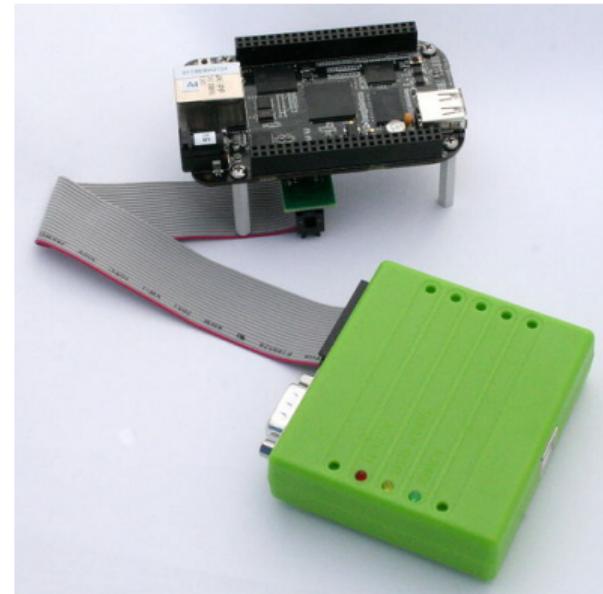
- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture
- Buses and protocols
- Connecting external devices
- Writing the drivers
- Device Tree

```
249 &i2c2 {  
250     pinctrl-names = "default";  
251     pinctrl-0 = <&i2c2_pins>;  
252  
253     status = "okay";  
254     clock-frequency = <100000>;  
255  
256     cape_eeprom0: cape_eeprom0@54 {  
257         compatible = "atmel,24c256";  
258         reg = <0x54>;  
259         #address-cells = <1>;  
260         #size-cells = <1>;  
261         cape0_data: cape_data@0 {  
262             reg = <0 0x100>;  
263         };  
264     };  
265  
266     cape_eeprom1: cape_eeprom1@55 {  
267         compatible = "atmel,24c256";  
268         reg = <0x55>;  
269         #address-cells = <1>;  
270         #size-cells = <1>;  
271         cape1_data: cape_data@0 {  
272             reg = <0 0x100>;  
273         };  
274     };
```

Embedded Topics

We will cover next Embedded topics during the course:

- Building, flashing and booting
- Bootloader and rootfs
- Electronics basics
- SoC architecture
- Buses and protocols
- Connecting external devices
- Writing the drivers
- Device Tree
- Hardware debugging



What does it take?

- We don't cover: C, Linux, Git, Makefiles
 - ...learn along the way!

What does it take?

- We don't cover: C, Linux, Git, Makefiles
 - ...learn along the way!
- 3 months
 - ...is not too long!

What does it take?

- We don't cover: C, Linux, Git, Makefiles
 - ...learn along the way!
- 3 months
 - ...is not too long!
- A lot of homework
 - ...I encourage you to do even more hacking and test your ideas!

What does it take?

- We don't cover: C, Linux, Git, Makefiles
 - ...learn along the way!
- 3 months
 - ...is not too long!
- A lot of homework
 - ...I encourage you to do even more hacking and test your ideas!
- Books: there are old ones and new ones; also LWN
 - ...but *practice makes perfect!*

What does it take?

- We don't cover: C, Linux, Git, Makefiles
 - ...learn along the way!
- 3 months
 - ...is not too long!
- A lot of homework
 - ...I encourage you to do even more hacking and test your ideas!
- Books: there are old ones and new ones; also LWN
 - ...but *practice makes perfect!*
- Working environment (IDE, toolchain, tools)
 - ...people matter even more; Linux is all about community

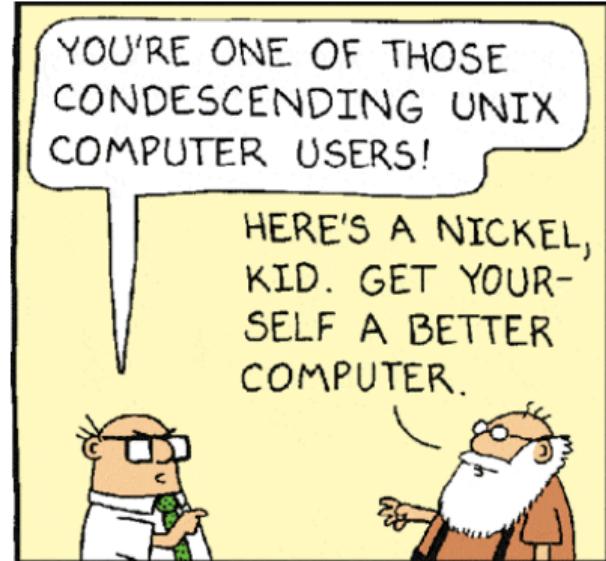
UNIX/Linux History

2019: The Anniversaries

- 25 years: Linux kernel v1.0
- 30 years: WWW
- 50 years:
 - UNIX
 - Internet (ARPANET)
 - People walking on the moon
 - Linus Torvalds

2019: The Anniversaries

- 25 years: Linux kernel v1.0
- 30 years: WWW
- 50 years:
 - UNIX
 - Internet (ARPANET)
 - People walking on the moon
 - Linus Torvalds



(Credits: Maddog presentation at FOSDEM 2019)

UNIX

“...When BTL withdrew from the project, they needed to rewrite an OS in order to play space war on another smaller machine (a DEC PDP-7 with 4K memory for user programs). The result was a system which a punning colleague called UNICS (UNiplexed Information and Computing Service) – an ‘emasculated Multics’; no one recalls whose idea the change to UNIX was.”



UNIX (cont'd)

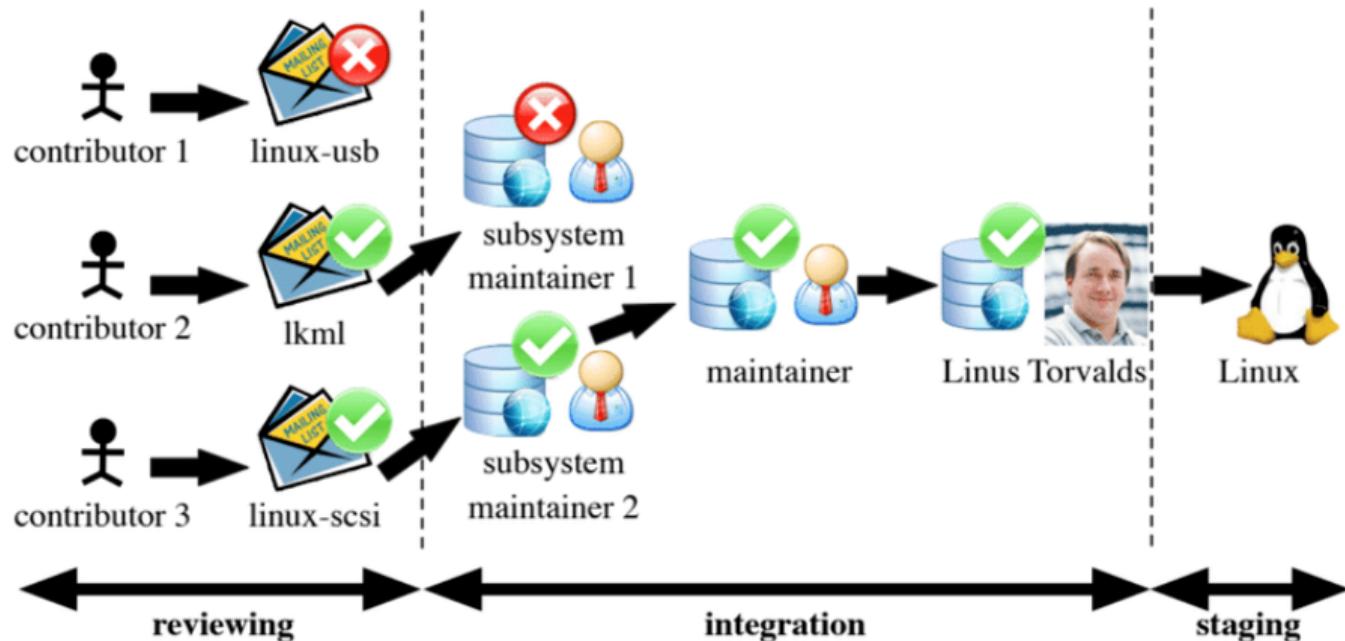
- 1969: UNIX development started (PDP-7)
- 1971: V1, written in assembler (PDP-11/20)
- 1973: **V4 was re-written in C**
- 1975: V6 was available outside of Bell Labs (basis for BSD)
- 1979: V7: K&R C, Bourne shell; ported to VAX (32-bit); kernel is only 40 KiB!

GNU/Linux

- 1984: GNU is not UNIX
- 1991: LINUX v0.01
- 1994: LINUX v1.0
- 2005: **First Git release**

"Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready."

Linux Development Process



References

References

-  Eric S. Raymond.
The Cathedral and the Bazaar.
-  Tracy Kidder.
The Soul of a New Machine.
-  Steven Levy.
Hackers: Heroes of the Computer Revolution, 25th Anniversary Edition.

Thank you!

[joe\(skb7@gmail.com](mailto:joe(skb7@gmail.com)