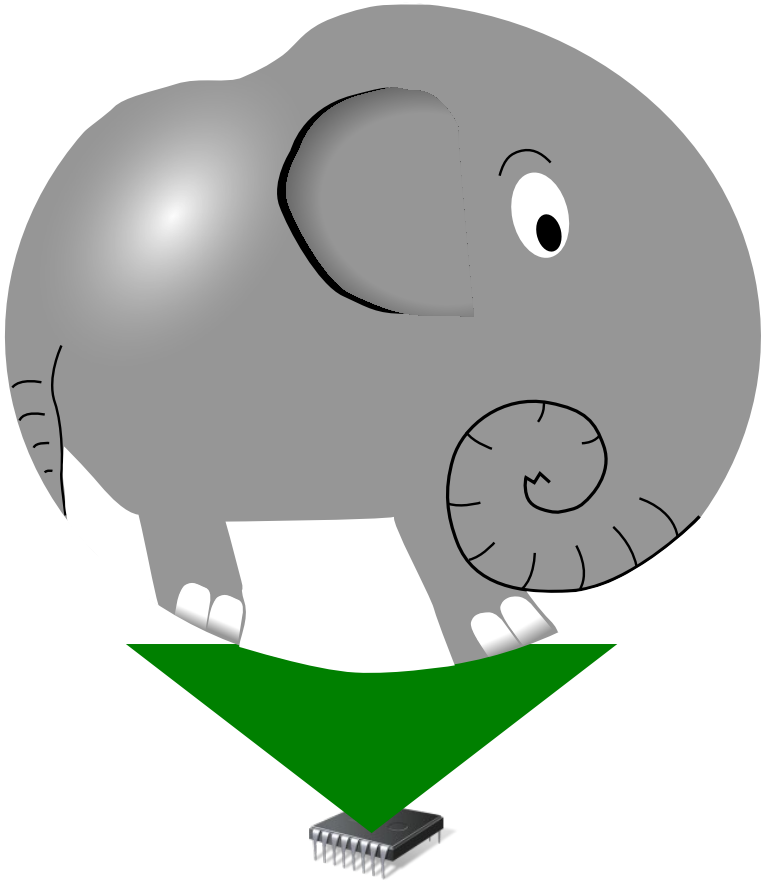
Professional Seminars II:
Operating Systems for Embedded Systems

Introduction



r05
Ángel Perles

Contents

- Operating Systems
- OS for Embedded
- Linux for Embedded
- Raspberry Pi

Operating Systems

- A set of computer programs that provide an abstraction of the physical machine to give you an enhanced/virtual machine

Bare metal machine

OS1

OS2

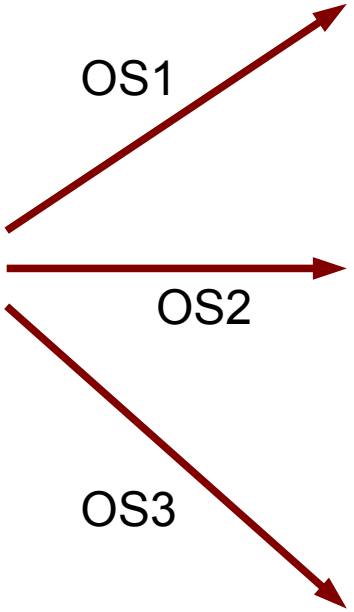
OS3

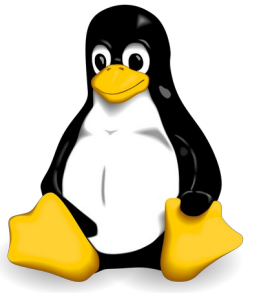
Every OS has specific advantages and inconveniences

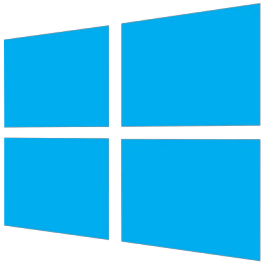
Operating Systems

- A set of computer programs that provide an abstraction of the physical machine to give you an enhanced/virtual machine

Bare metal machine







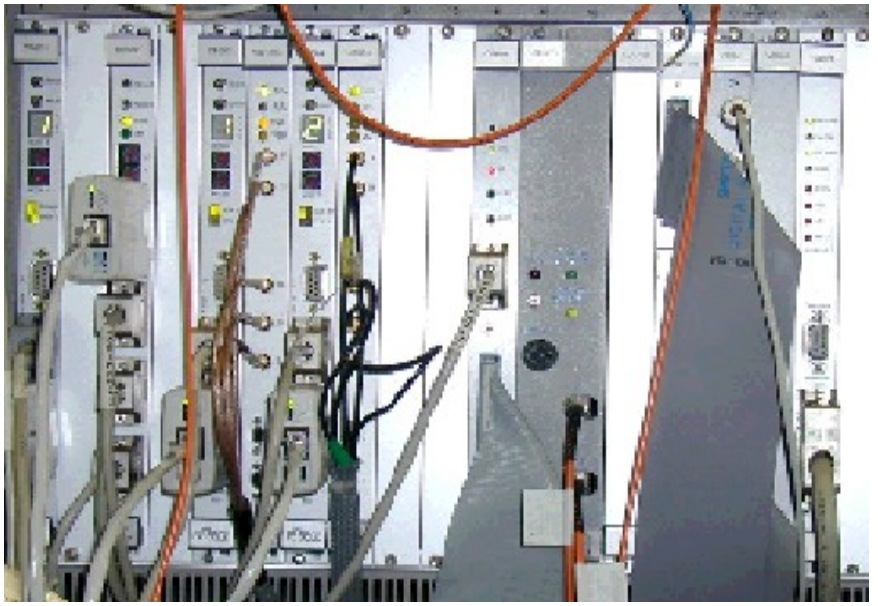
These are desktop-oriented OSes

Operating Systems

- A set of computer programs that provide an abstraction of the physical machine to give you an enhanced/virtual machine
  - Hardware and software resource management. For example:
    - Process creation, destruction and CPU assignment
    - Memory assignment
  - Persistence
    - How data is stored: files, directories, ...
  - Application services:
    - User applications invoke services of the OS

OS for Embedded

- Every embedded system has different needs
  - smartphones -> best user experience
  - router -> high throughput, lowest price
  - autopilot, ABS brake -> hard real-time, safety



copter control (German Aerospace Centre courtesy)

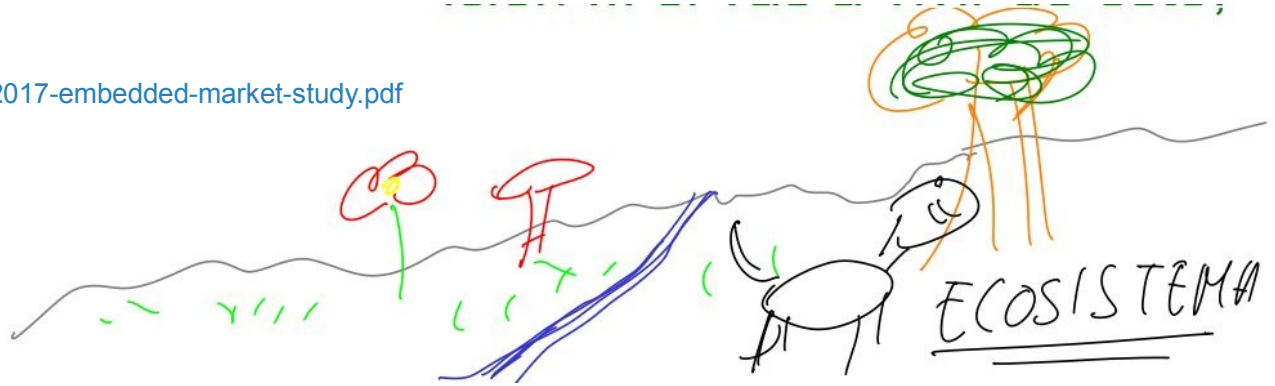
OS for Embedded

- There are commercial and free OSes focused to this segment
  - for microcontrollers: FreeRTOS/SafeRTOS, uCos, RTX, ...
  - for general pourpose processors:
    - vxWorks: http://windriver.com/products/vxworks/
    - QNX: http://www.qnx.com/
    - Nucleus: https://www.mentor.com/embedded-software/nucleus/
    - Linux: http://www.linuxfoundation.org/
    - etc.

- 5 min. to take a look to vxWorks and QNX

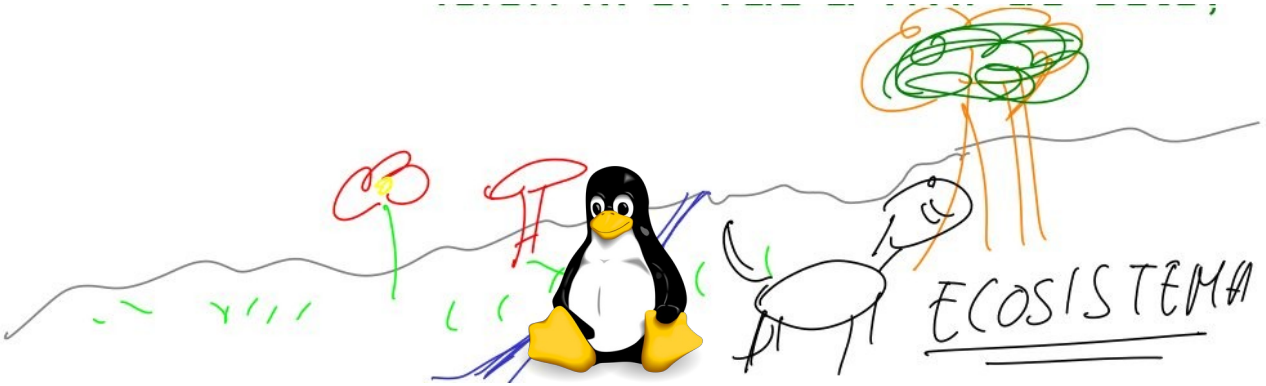
OS for Embedded

- An OS is insufficient. Please, take care of the ECOSYSTEM
  - Available programming languages
  - Information and reference examples
  - Community
  - Debugging aids
  - Supported platforms (ARM, MIPS, x86, RISC-V, ...)
  - Commercial support
  - Licensing
  - Open source
  - https://m.eet.com/media/1246048/2017-embedded-market-study.pdf



OS for Embedded

- The GNU and LINUX combo excels because:
  - Available programming languages -> free and open
  - Information and reference examples
  - Community -> BIG, and not only engineers
  - Debugging aids -> free and open
  - Supported platforms (ARM, MIPS, x86, RISC-V, ...)
  - Commercial support -> e.g. Red Hat
  - Licensing -> well some problems here: GPL, LGPL, ...
  - Open source
  - etc.

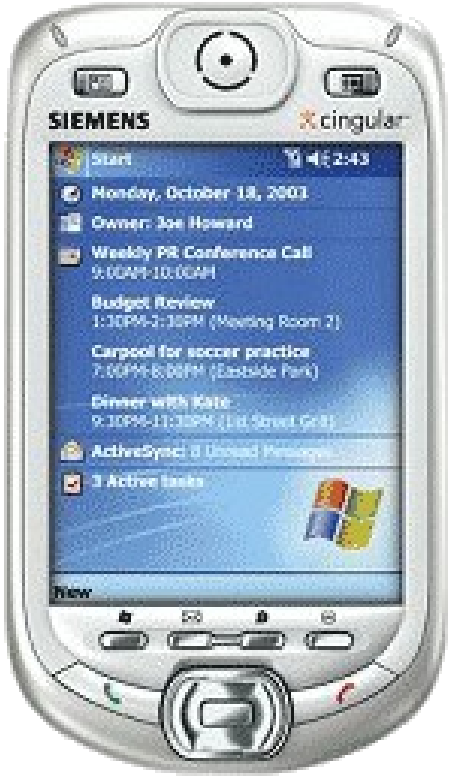


Linux for Embedded

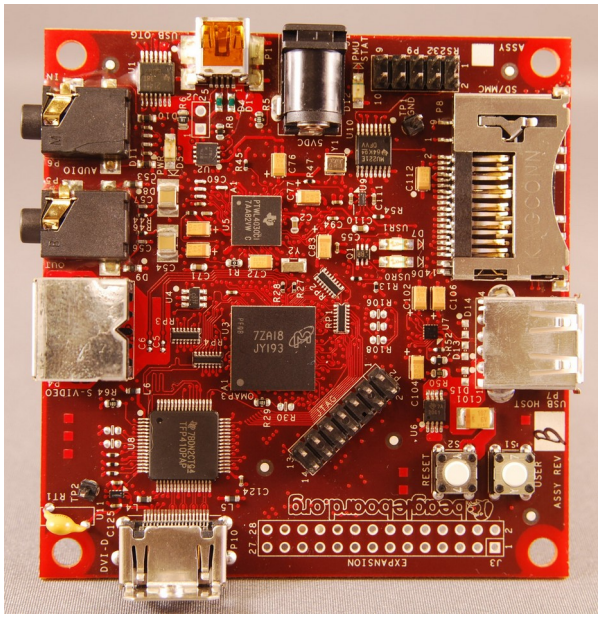
- Present in lots of devices
  - Smartphones: Android is based on Linux
  - Routers, IoT infrastructures
  - NAS
  - Drones (Parrot, ...)
  - etc.

Linux for Embedded

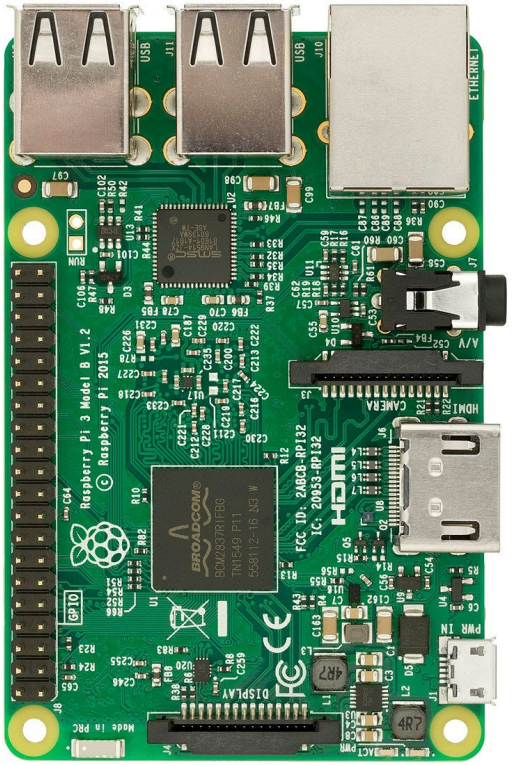
"Hack" community ⇨ "Geek" community ⇨ "Democratic" community



HTC Blueangel
Intel PXA 253



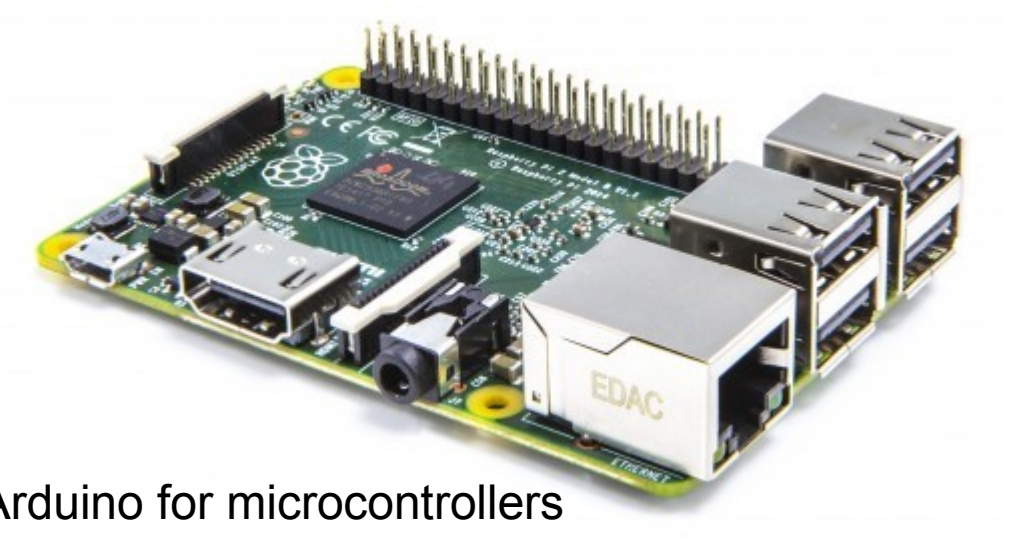
Beagleboard
Texas Instruments OMAP3



Raspberry Pi
Broadcom BCM 2xxx

Raspberry Pi

- The perfect fit for noobs
  - To learn Linux
  - To learn embedded
  - To learn hardware interface
  - ... and fantastic community of experts and noobs



Raspberry Pi has the same approach than Arduino for microcontrollers

http://www.raspberrypi.org/

Raspberry Pi

- For the lab (slightly outdated)
- Raspberry Pi 2 model B
  - "SoC" (Sytem-on-Chip) Broadcom BCM2836
    - 900MHz quad-core ARM Cortex-A7 processor
    - GPU VideoCore 4 (plays 1080p video)
  - 1 GiB of RAM
  - HDMI video output
  - Audio output
  - 4 USB 2.0 ports
  - Ethernet 10/100 LAN connection