Linux For Embedded Systems

For Arabs

Cairo University
Computer Eng. Dept.
CMP445-Embedded Systems

Ahmed ElArabawy

Lecture 4:

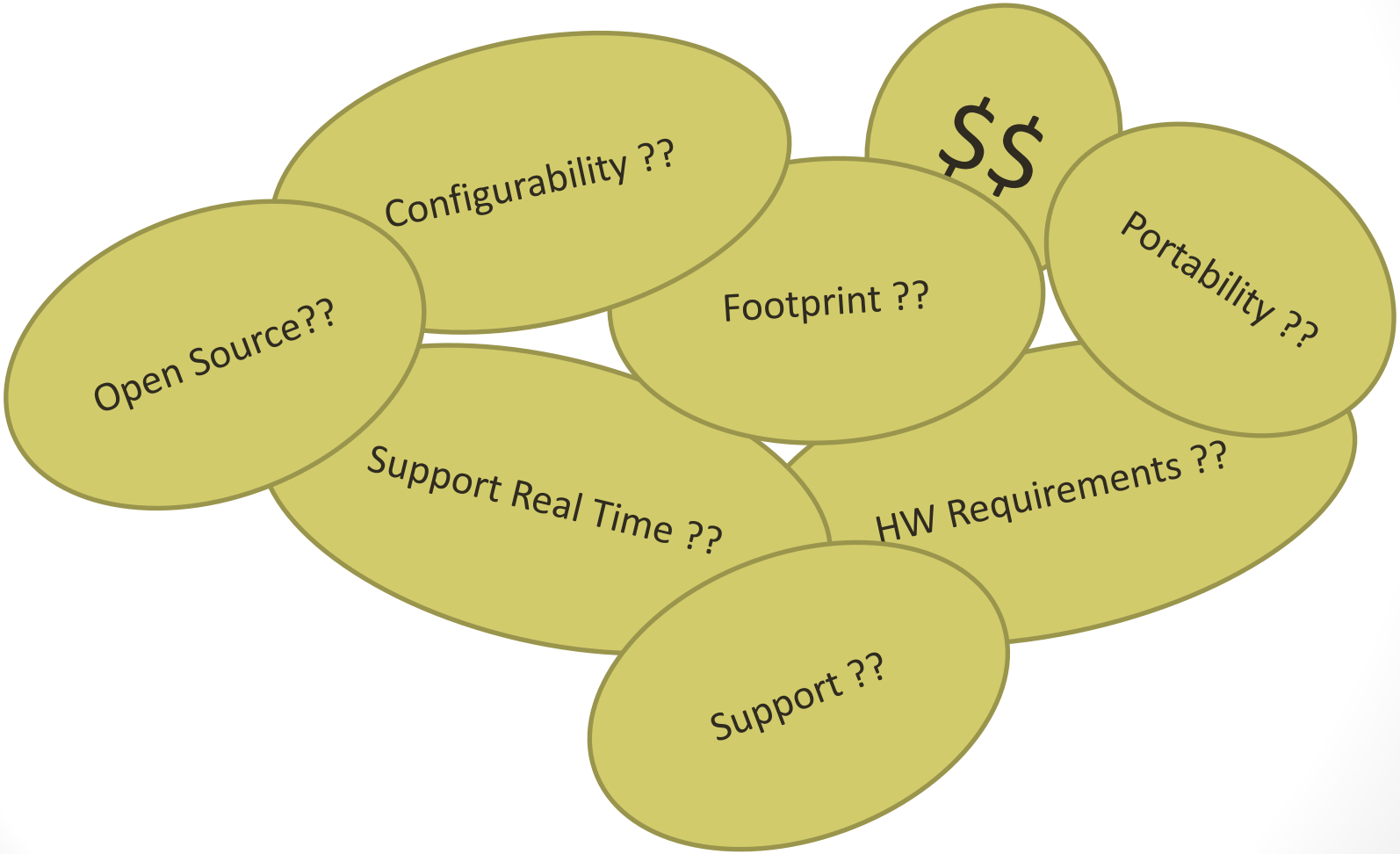
Selecting the Proper OS

Do We Have to Have an OS ?

Out of our Scope

- Not really
- Very simple embedded systems don't need an operating system, and development happens directly on the hardware
- Also, code for a bootloader or a kernel does not assume an OS
- Development is normally done in assembly language and in C
- No support of multiple tasks, or multiple processes, a simple loop runs the different functionality back to back
- No or minimal support of Timers…
- Storing info in the flash is done in a very primitive way
- These systems are very limited in functionality

Which OS Should We Use ??



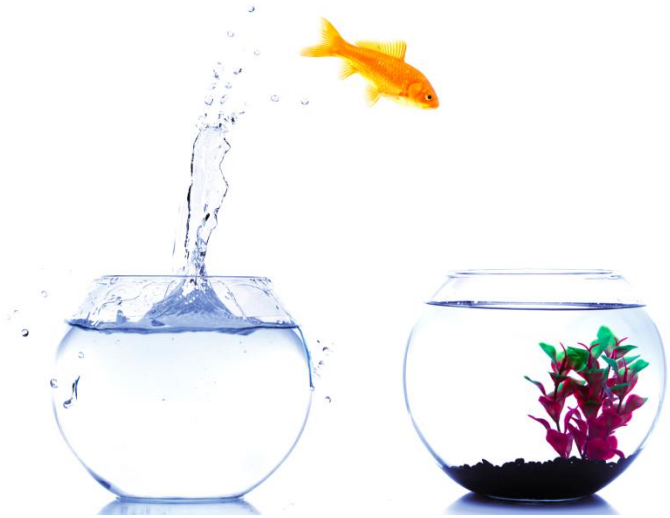
Cost ….

- Commercial OS's
  - Different cost models:
    - Fixed price for OS License
      - Per device type or family
      - Per processor
      - Per device line
    - Per seat license for associated tools
    - License fee per unit (Royalty cost)
    - Others…..
- Free OS's
  - No direct cost for using the OS and plenty of tools
  - Sometimes there is indirect costs such as
    - Support
    - Special services
    - Special versions (more reliability, or better performance)
    - Some associated tools
  - With time more and more free tools
  - Examples include Linux, uClinux, Ecos,..

Hardware Requirements

- Each OS has its own hardware requirements
  - Some OS's require 32 bit processors, others are OK with 8 bits
  - Some OS's require a processor with MMU, others are fine without it
  - Each OS has its own minimum memory and storage requirements
- Also, the embedded platform may have some requirements
  - Does it need a USB 2.0 Support ?
  - Does it need TCP/IP support ?

Portability



- We may need to migrate to a different Hardware Platform
  - Each OS has its own list of supported architectures
- We may need to migrate our code to a different OS
  - Software portability can be achieved by support of APIs (such as POSIX)

Footprint

- Footprint is the required size of storage and memory required to start and operate the OS
- Some OS's require very small footprint (as low as 2 KB), while others require much bigger size (Several Mega Bytes)
- The size of the embedded device and its storage resources may dictate which OS to use

Support of Real Time

- Embedded Applications usually come with real time constraints
- This require support for Real Time in the OS
- Real Time Operating System = RTOS
- Real time support does not mean fast response only, it means **Predictable Response** also

Support of Real Time

- Fast Response does not necessary mean Predictable response
- RTOS requires Predictable response, to make sure tasks meet the target hard real time constraints
- This means,
  - No background tasks that startup in random fashion and affect response time
  - Scheduling provides some guarantees
  - Short Interrupt Latency
  - Fast Context Switch

Configurability

- Some embedded OS's come with the option to be configurable
- This means that some of their features are optional, and can be carved out if not needed by the embedded application or the hardware
- This way we can customize our OS to the needs of the application
- Accordingly, we reach a smaller footprint and proper functionality support

Support

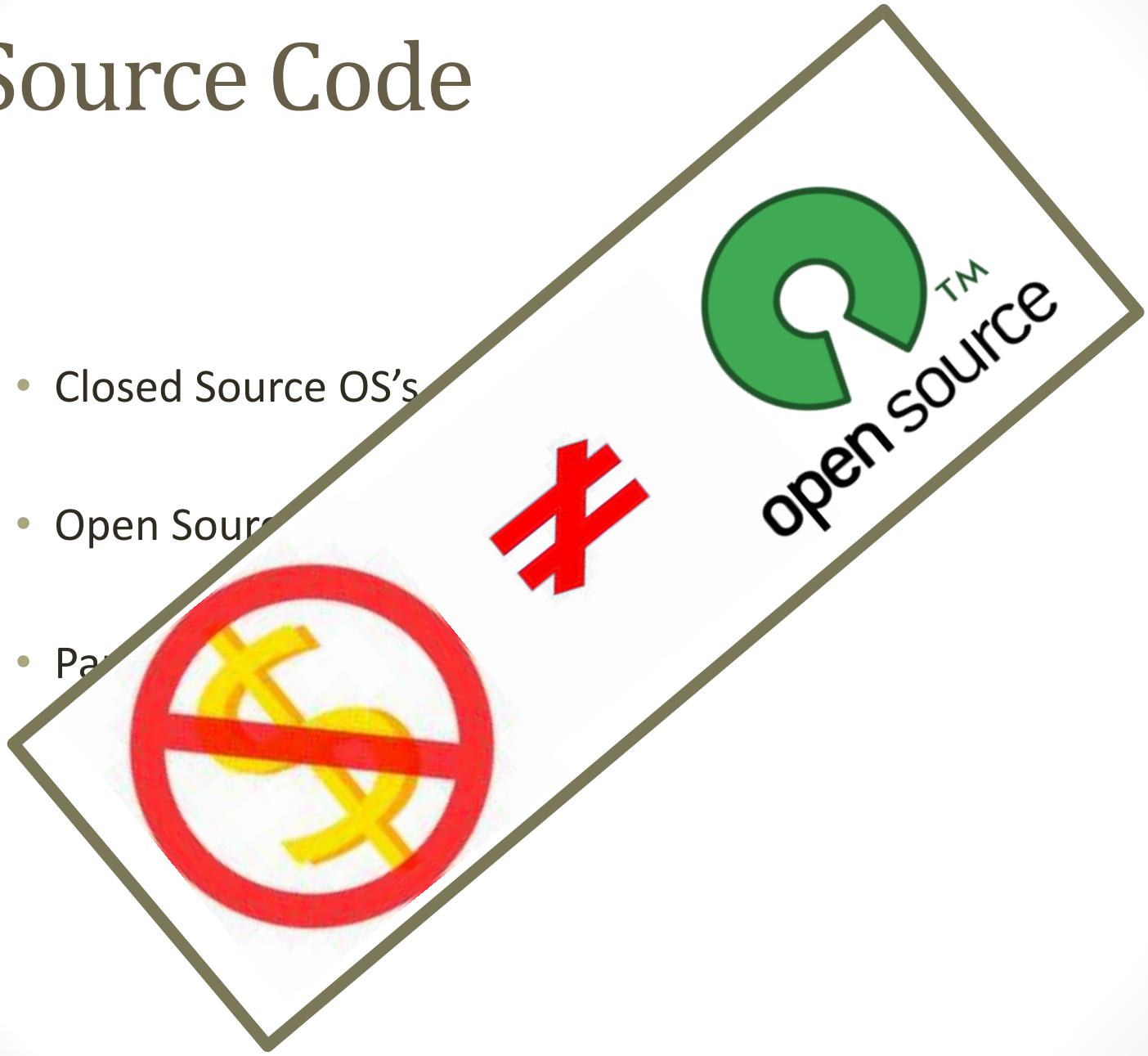
- One important aspect of choosing an OS is the level of support for that OS
- Is it only offered by the vendor ?
- Do we have strong community support ?
- Availability of Documentation

Source Code

- Closed Source OS's (Proprietary)
- Open Sources OS's
- Partially Open Source OS's

Source Code

- Closed Source OS's
- Open Sour
- Pa



Summary

Linux 4 Embedded Systems

http://Linux4EmbeddedSystems.com