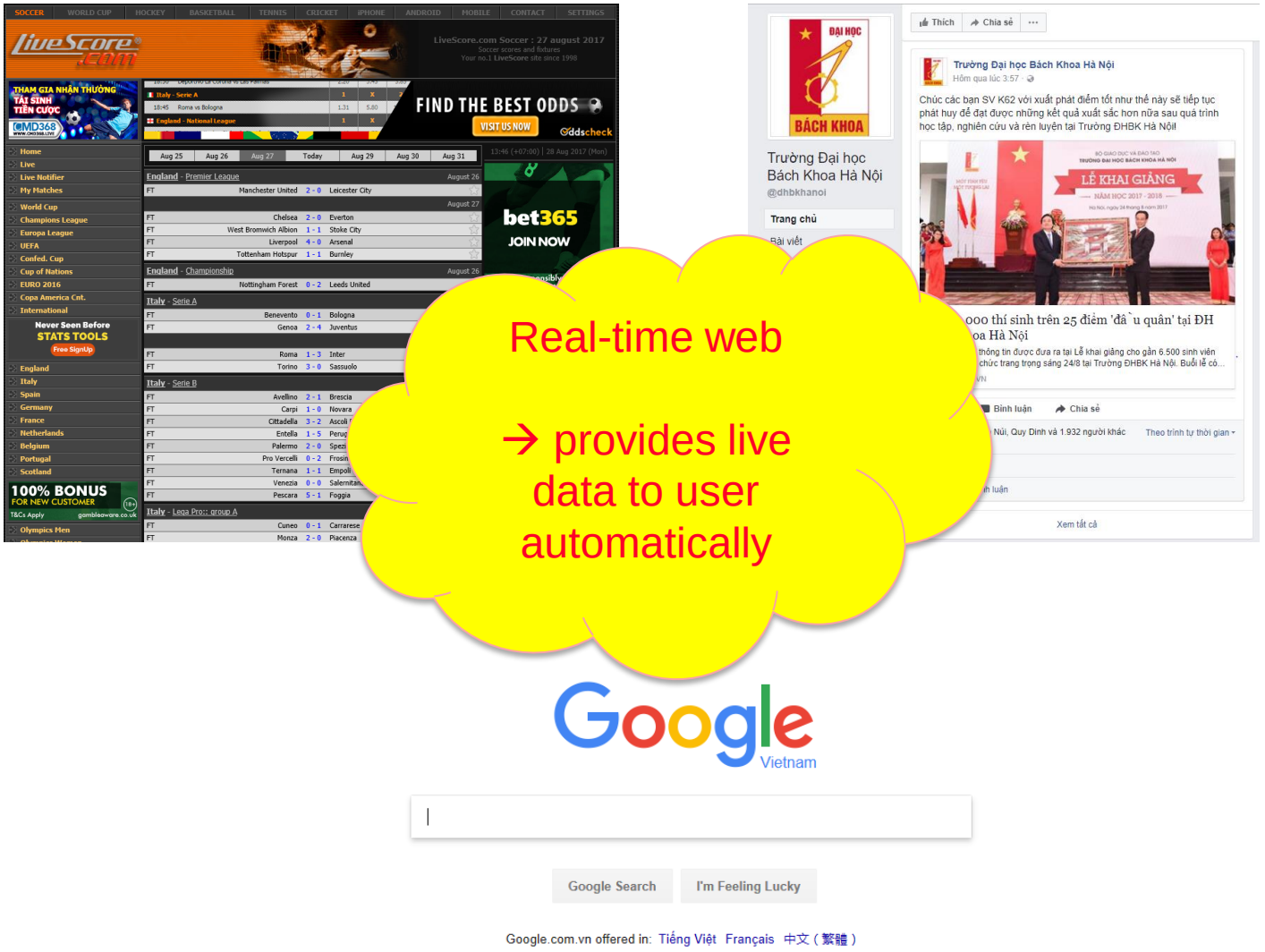
# Real-time Systems

Ngo Lam Trung

Dept. of Computer Engineering

# What is real-time?



# What is real-time?



Real-time system

→ provides response within allowed timeframe

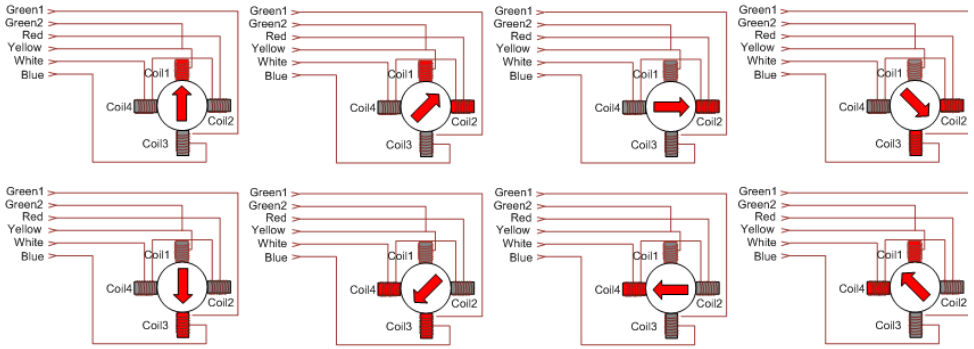

# Example 1

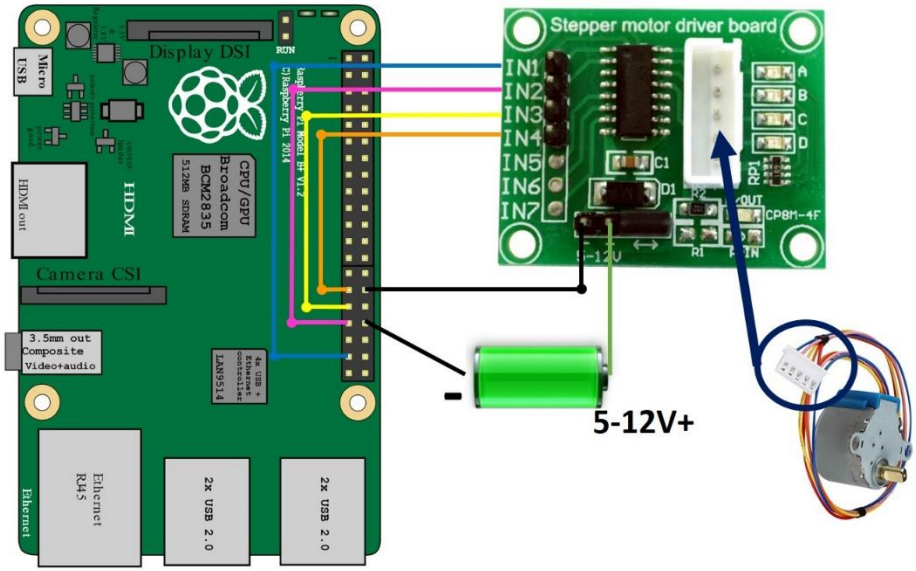
- Write a program in Windows to blink a LED with frequency of 1 Hz
- A piece of cake/code, right?

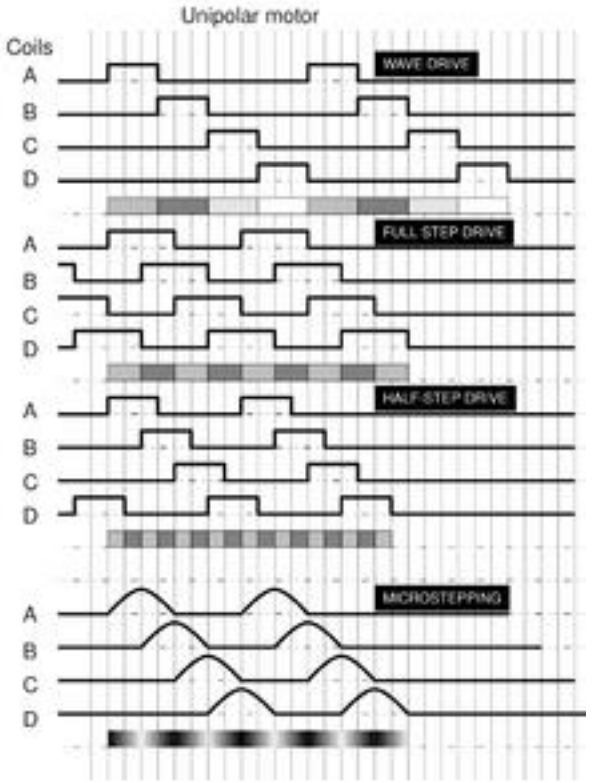
```
DispatcherTimer timer = new DispatcherTimer();
timer.Interval = TimeSpan.FromMilliseconds(500);
timer.Tick += Timer_Tick;
timer.Start();

private void Timer_Tick(object sender, object e)
{
        now = DateTime.Now;
        TimeSpan span = now - prev;
        prev = now;
        txtTime.Text = span.TotalMilliseconds.ToString();
}
```

- ***Not really!!!***

# Example 2: Linux is better than Windows?




Smooth controlling of motor requires accurate timing

# Example 2

- With RT_PREMPT Linux seems better, but not sufficient

Video

- In contrast, controlling a stepper motor with a low-cost Arduino is not that hard!!!

# Course content

- Basic concepts of real-time systems.
- Hard real-time systems and soft real-time systems.
- **Tasks scheduling algorithms in real time system.**
- **Schedulability analysis**

# References


- Qing Li and Carolyn Yao, Real-Time Concepts for Embedded Systems, 2003.
- Giorgio C. Buttazzo, Hard Real-time Computing Systems Predictable Scheduling Algorithms and Applications, 1997.

# Class administration

- Instructor: Ngo Lam Trung
- Class time: Wednesday AM
- Mid-term & Final exam: TBA

# Chapter 1: Introduction

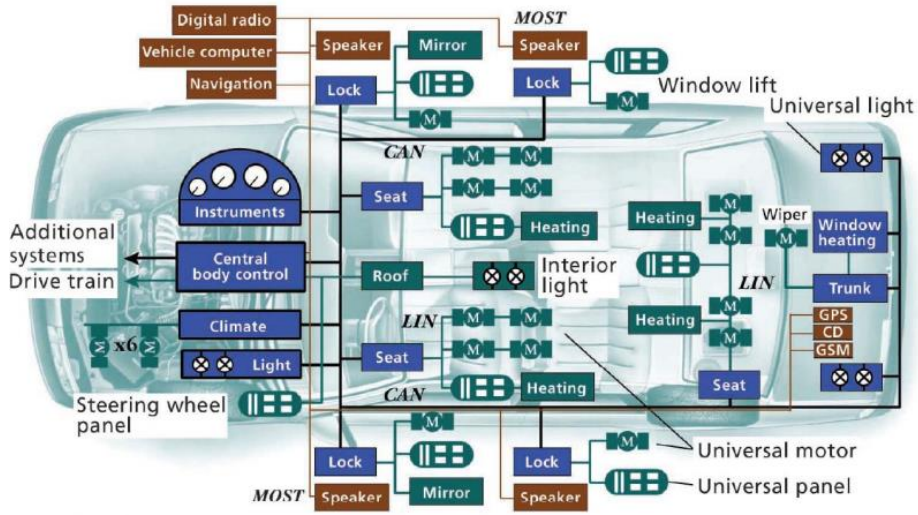
- Introduction of embedded system
- Characteristics of embedded system
- Real time system and real time embedded systems
- Hard real time vs soft real time.

*(With some materials from ES, Peter Marwedel)*

# 1. Embedded system

- Definition from Textbook 1:
  - Computing systems with tightly coupled hardware and software integration, that are designed to perform a dedicated function

Systems within systems




CAN Controller area network
GPS Global Positioning System
GSM Global System for Mobile Communications
LIN Local interconnect network
MOST Media-oriented systems transport

[Leen02]

Standalone system

# Other definitions of Embedded Systems

"Dortmund" Definition: [Peter Marwedel]

**Embedded systems are information processing systems embedded into a larger product**

Berkeley: [Edward A. Lee]*:*
**Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems.**

Wikipedia:
An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.

# Examples of embedded systems

**Sonicare Elite toothbrush**

**Microprocessor: 8-bit**

**Has a programmable speed control, timer, and charge gauge**

# Examples of embedded systems


**Product: Microsoft's Smart Personal Object Technology (SPOT) watch (discontinued in 2008).**

**Microprocessor: 32-bit ARM with FM Radio Chip**

**Downloads data using extra bandwidth on FM radio stations in major cities**

**Big idea but also a failure!**

# Examples of embedded systems

- Domestic robots

Roomba



Kuka youBot

# Examples of embedded systems

**S class Mercedes**

**Control system contains around 100 embedded processors**

# 2. Characteristics of embedded systems

# Dependability

- Dependability is the most important characteristic
  - Reliability *R(t)* = probability of system working correctly provided that is was working at *t*=0
  - Maintainability *M(d)* = probability of system working correctly *d* time units after error occurred.
  - Availability *A*(*t*): probability of system working at time *t*
  - Safety: no harm to be caused
  - Security: confidential and authentic communication
- System dependability depends on the estimation of working/runtime condition in design time.
- Incorrect/insufficient estimation → good system will fail.
- Dependability must be considered very early in design time

# Efficiency

- Embedded system must be efficient
  - Code-size efficient: (especially for systems on a chip)
  - Run-time efficient
  - Weight efficient
  - Cost efficient
  - Energy efficient

# Efficiency and application awareness

- Which CPU is faster, why?
  - CPU in spacecraft
  - CPU in Boeing 777
  - CPU in your laptop
- Which is better for a mobile phone: a quad-core 2.2GHz or single-core 1GHz CPU?

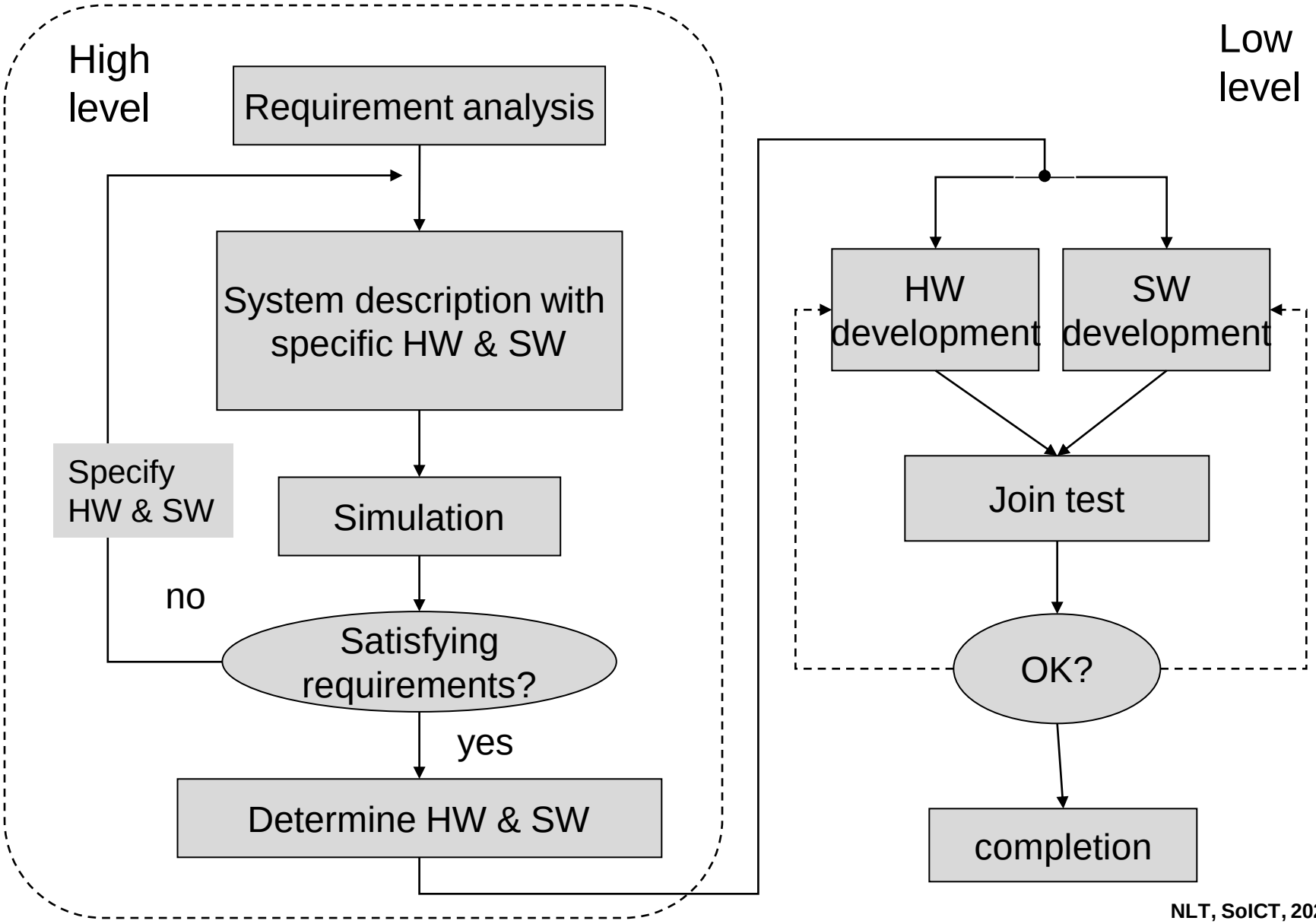
# Hardware and software co-design model

- How to design embedded system?
  - Hardware of software first?
  - How to optimize system design and performance?

- Hardware and software co-design
  - Parallel development of HW & SW of an embedded system
  - Beneficial in an embedded system with custom hardware and software

- Software component can use special hardware features.
- Hardware component can simplify module design if functionality can be achieved in software.

# Hardware and software co-design model



# Cross-platform development

- Target system: limited hardware resource → cannot be used as development environment
- How to develop software to run on target system?
  - → Use a different platform as development environment
- Platform: hardware, OS, and development tools

# Cross-platform development

- Cross-platform development
  - Platform: HW + OS + SW development tools
  - Software development where developing platform and running platform are separating

Cross compiler

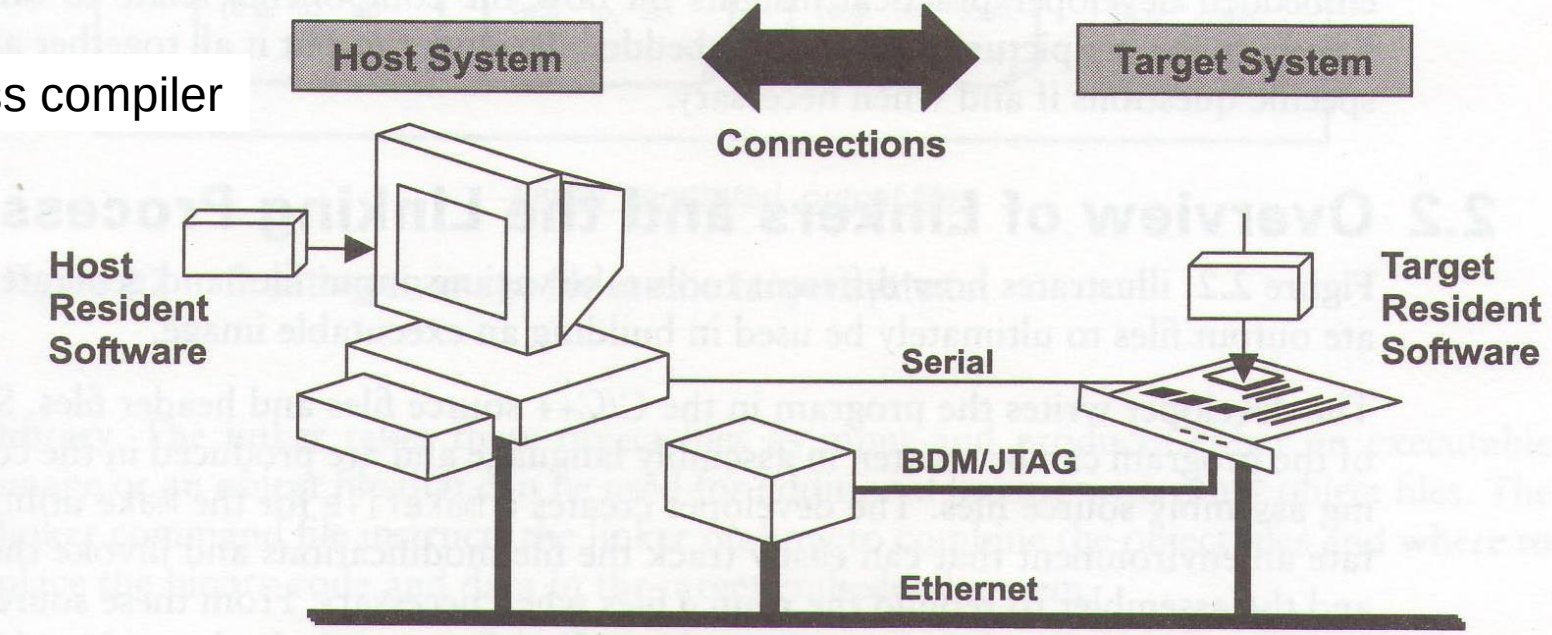


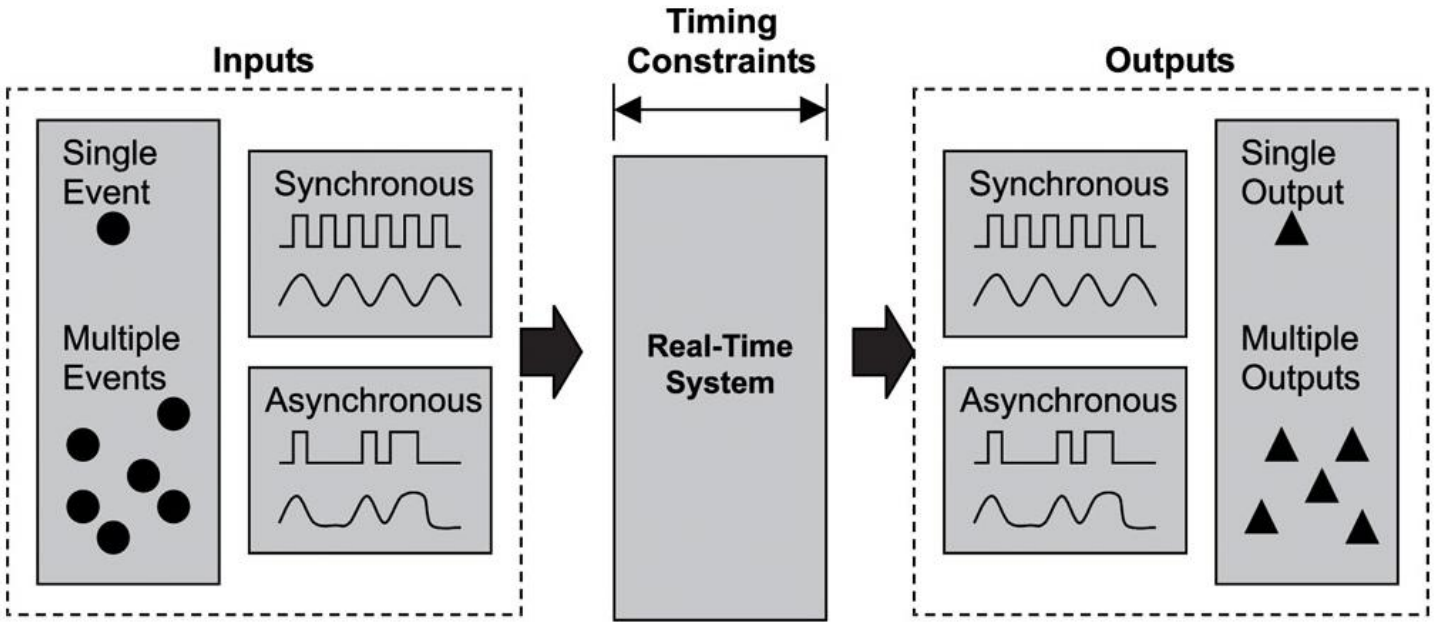

Figure 2.1 Typical cross-platform development environment.

# 3. Real time systems

A real-time system means that the system is subjected to real-time, i.e., the response should be guaranteed within a specified timing constraint or the system should meet the specified deadline. For example flight control systems, real-time monitors, etc.

Types of real-time systems based on timing constraints:
Hard real-time system: This type of system can never miss its deadline. Missing the deadline may have disastrous consequences. The usefulness of results produced by a hard real-time system decreases abruptly and may become negative if tardiness increases. Tardiness means how late a real-time system completes its task with respect to its deadline. Example: Flight controller system.
Soft real-time system: This type of system can miss its deadline occasionally with some acceptably low probability. Missing the deadline have no disastrous consequences. The usefulness of results produced by a soft real-time system decreases gradually with an increase in tardiness. Example: Telephone switches.

# Real-time systems



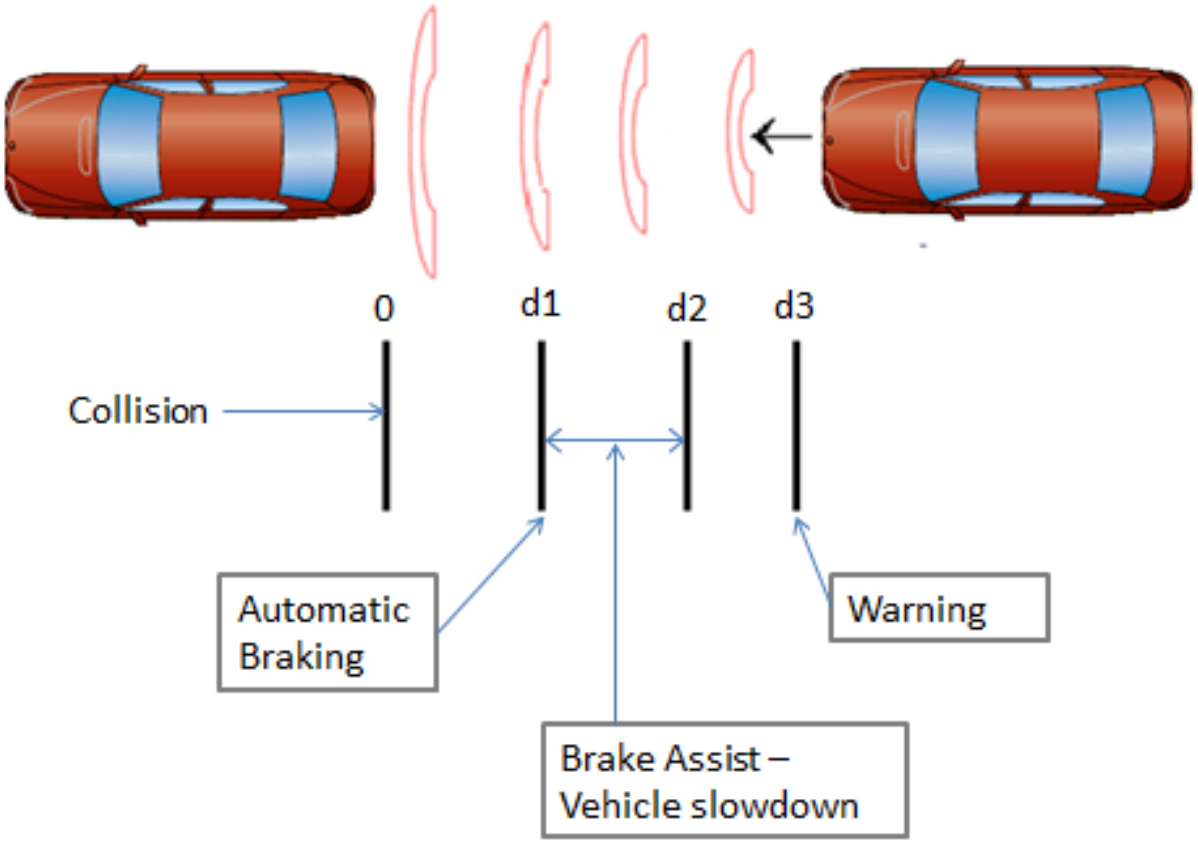

- Real-time systems:
  - Those systems that respond to external events with guaranteed timing constraints
  - Timing constraints: start time, finished time
  - External events: periodic/aperiodic
  - Both of the timing correctness and logical correctness are important.

# Real-time embedded systems

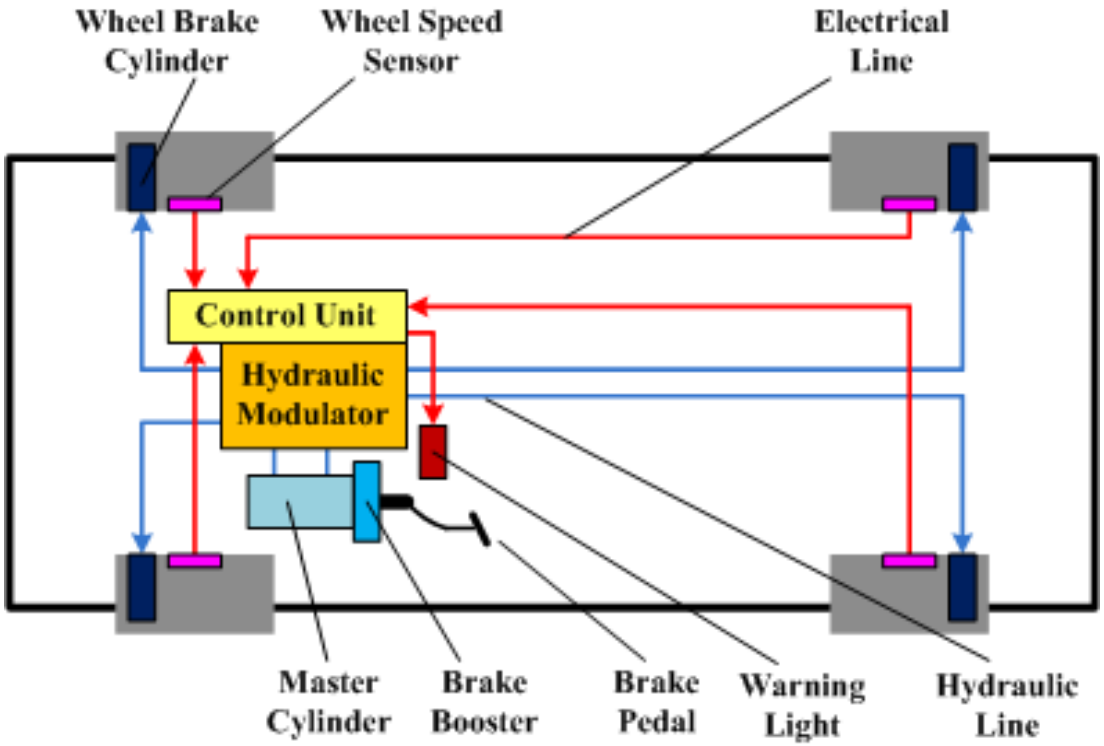
- Example: DVD player


What are real-time requirements?

- Autonomous Emergency Breaking System



# Anti-lock Breaking System



# Hard and soft real-time systems

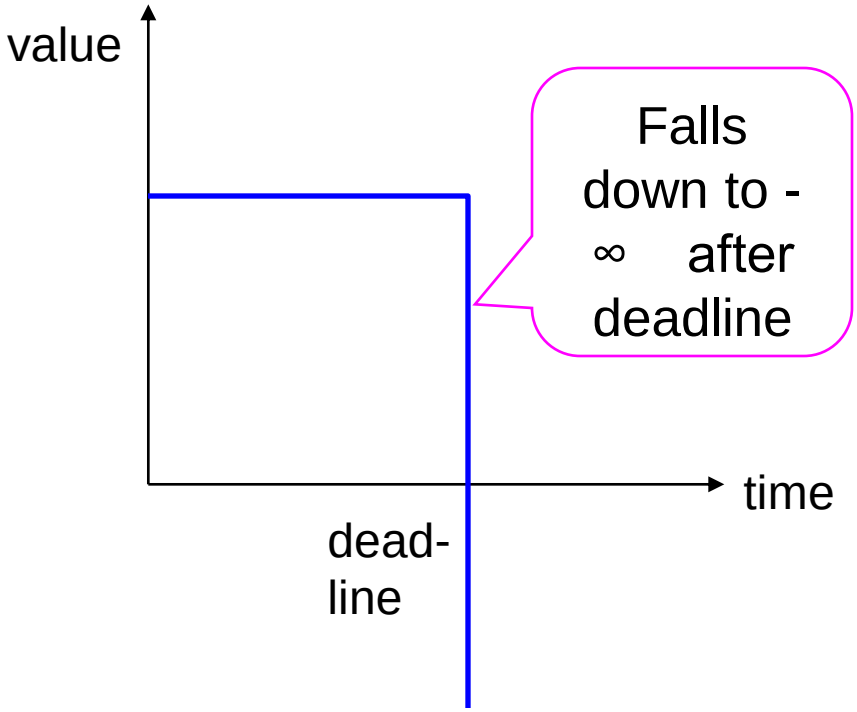
- **Hard real-time systems**
  - Must meet deadlines with a non-zero degree of flexibility
  - Missing deadlines derives catastrophes
  - Ex: car ABS, aviation systems, missile guidance

- **Soft real-time systems**
  - Must meet deadlines but with a degree of flexibility
  - Missing deadlines decreases the value of the computed results. Decrement of the value is proportion to the delay.
  - Ex: DVD player

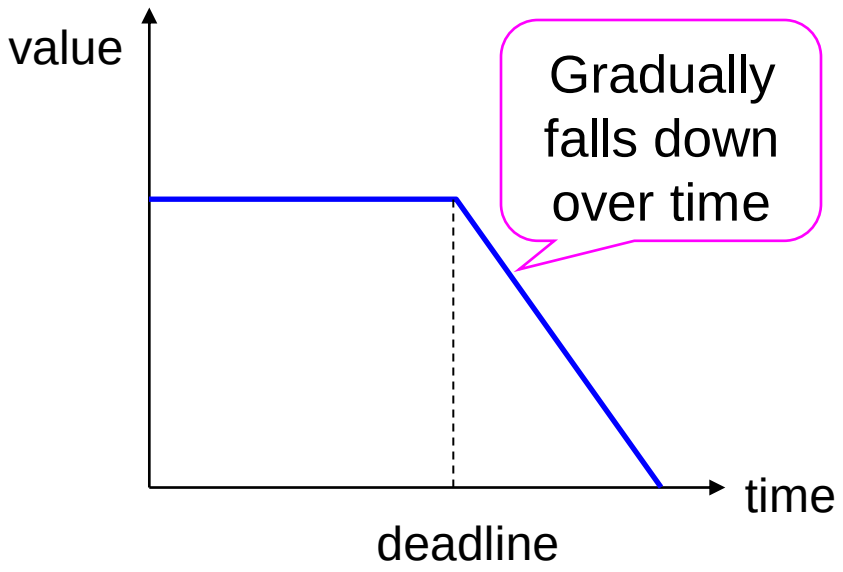
# Penalties in real-time systems

Variations of values of execution results with respect to the finished time

Hard real-time

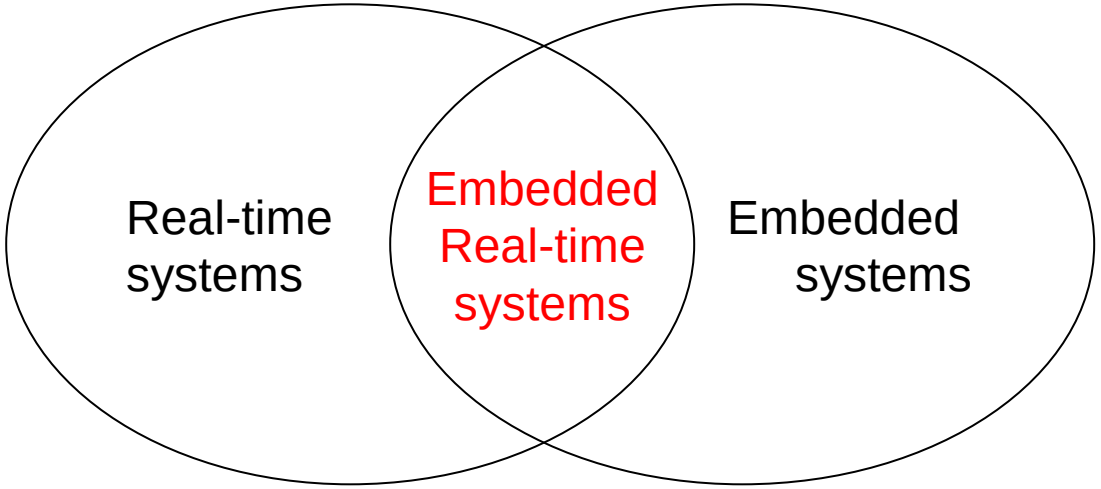



Soft real-time



# Real-time embedded systems

Large overlap of real-time systems and embedded systems

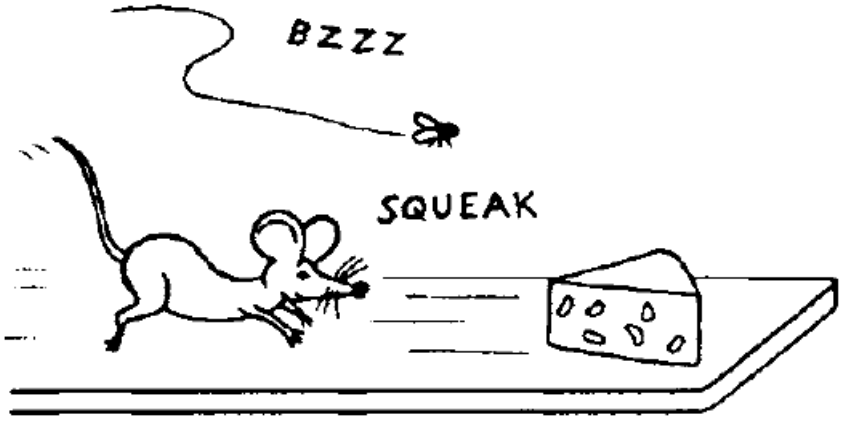


# Points to remember

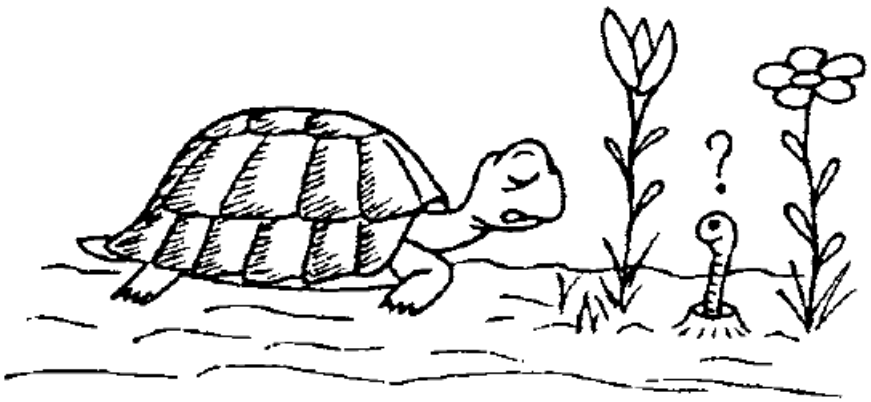
- Embedded systems:
  - Computing systems with tightly coupled hardware and software integration, that are designed to perform a dedicated function
- Real-time embedded systems
  - Real-time: required timing & function correctness
  - Commutative between embedded systems and real-time systems
- Hard/soft real-time systems
  - Determined by penalty in deadline misses

# Comparison

- Real-time vs real-fast?




(a)




(b)

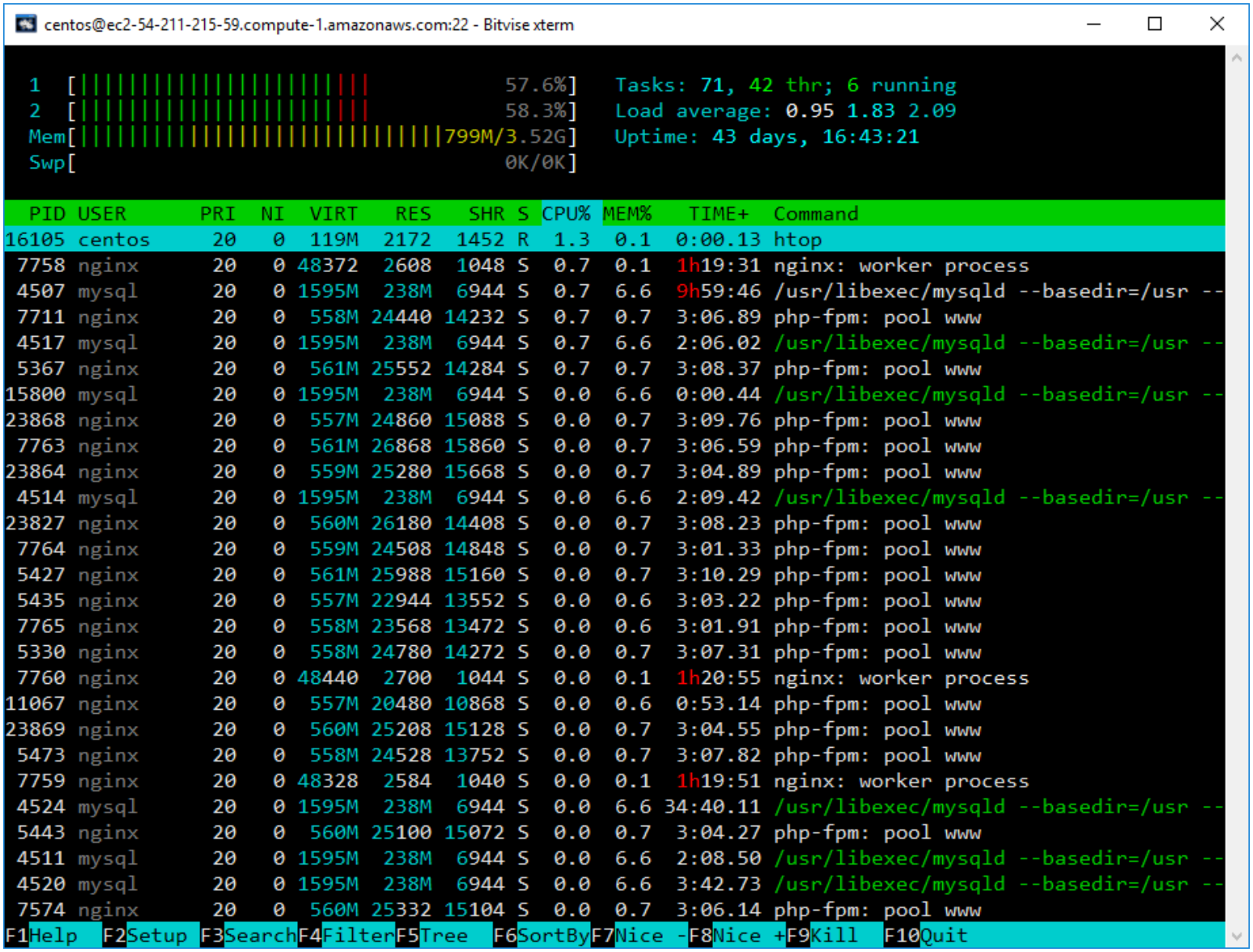
La Fontaine's The Hare and the Tortoise
"it's no use running, it's better to leave early…"

# Comparison

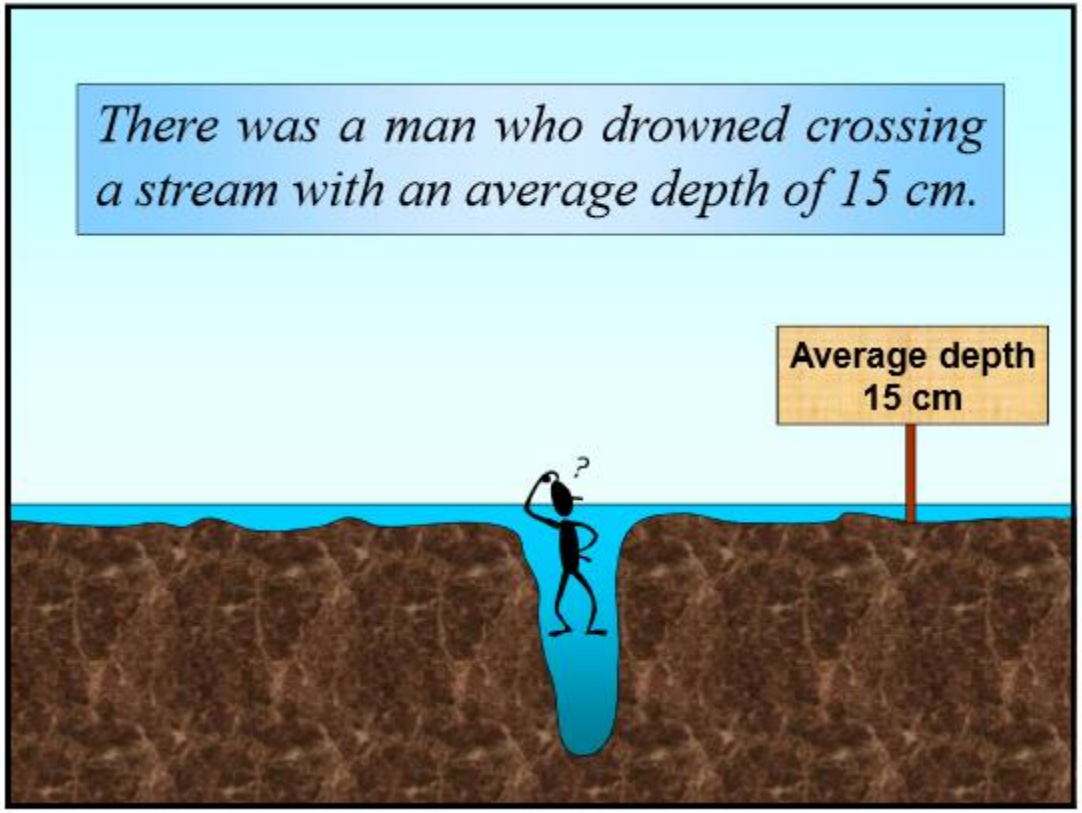
- The objective of a real-time system is to guarantee the timing behavior of each individual task.
- The objective of a fast system is to minimize the average response time of a task set.

- Which is more difficult?
- A high performance server is running with 1% CPU usage. Is it a real-time system?

# Don’t trust average



# Don’t trust average



# Discussion

- What is the most important part of a real-time system?
  - Hardware?
  - Software?
- Homework: do some research on computing system on spacecraft

  http://www.cpushack.com/space-craft-cpu.html