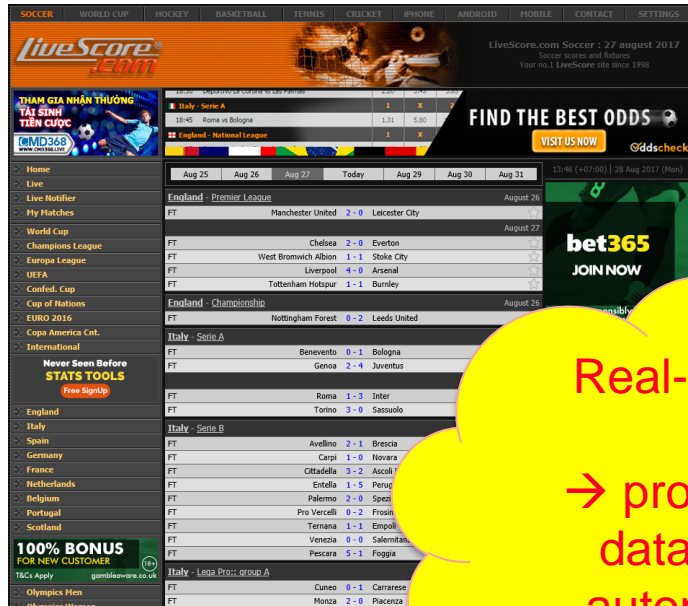

Real-time Systems

Ngo Lam Trung

Dept. of Computer Engineering

What is real-time?



Real-time web
→ provides live
data to user
automatically



Google Search

I'm Feeling Lucky

Google.com.vn offered in: Tiếng Việt Français 中文 (繁體)

What is real-time?



Real-time system

→ provides response within allowed timeframe



Example 1

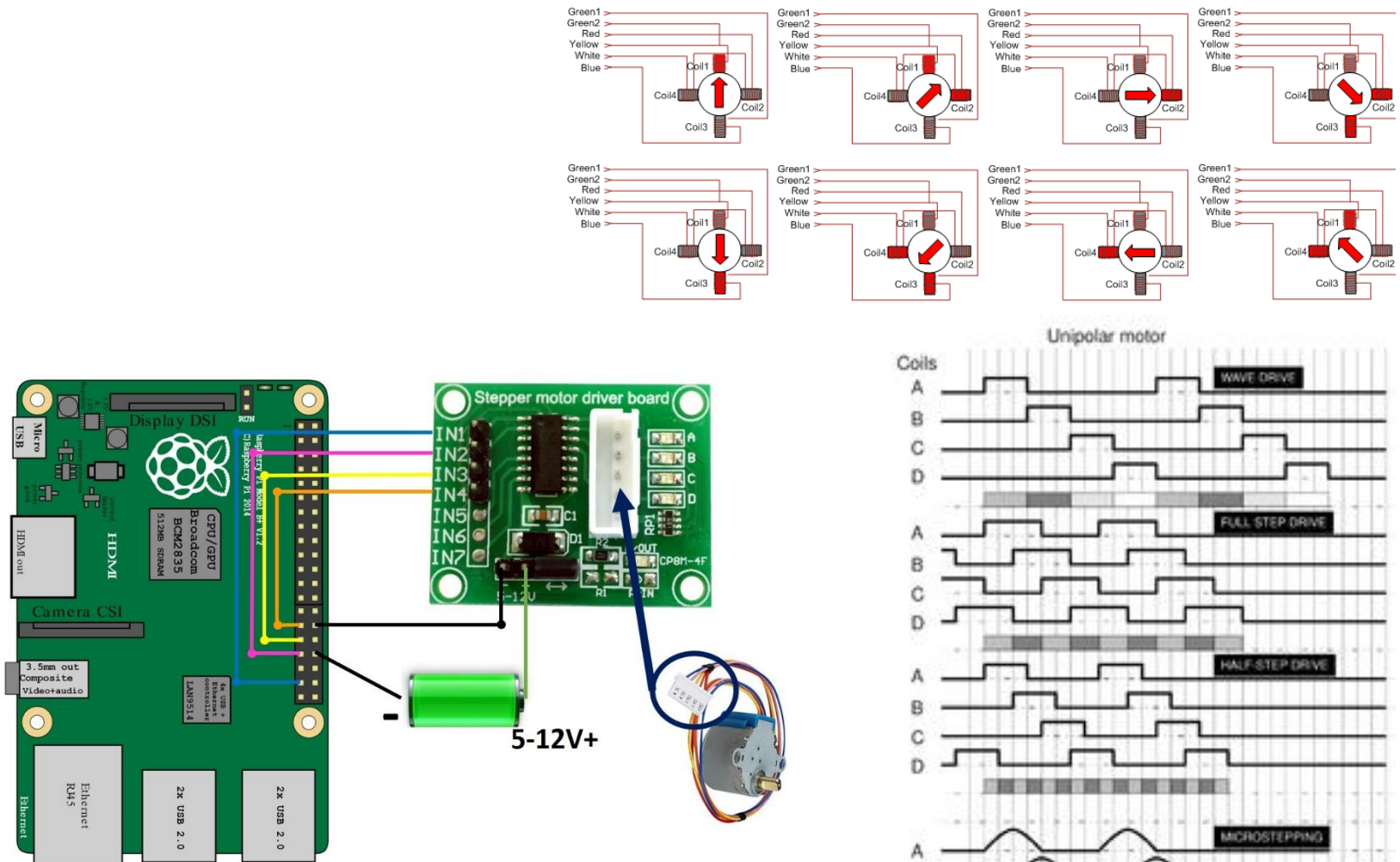
- ❑ Write a program in Windows to blink a LED with frequency of 1 Hz
- ❑ A piece of cake/code, right?

```
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.

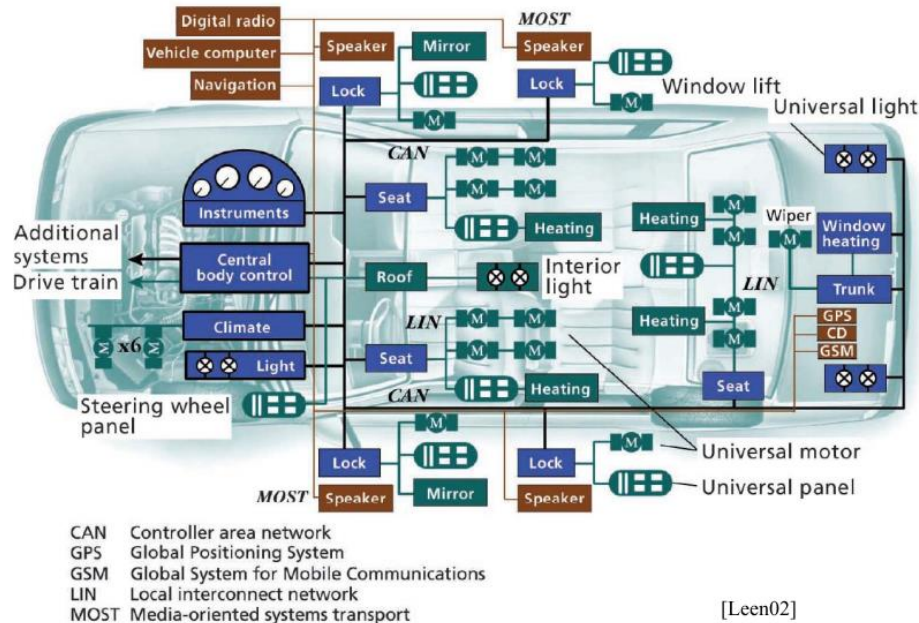
(With some materials from ES, Peter Marwedel)

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



[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 it 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?

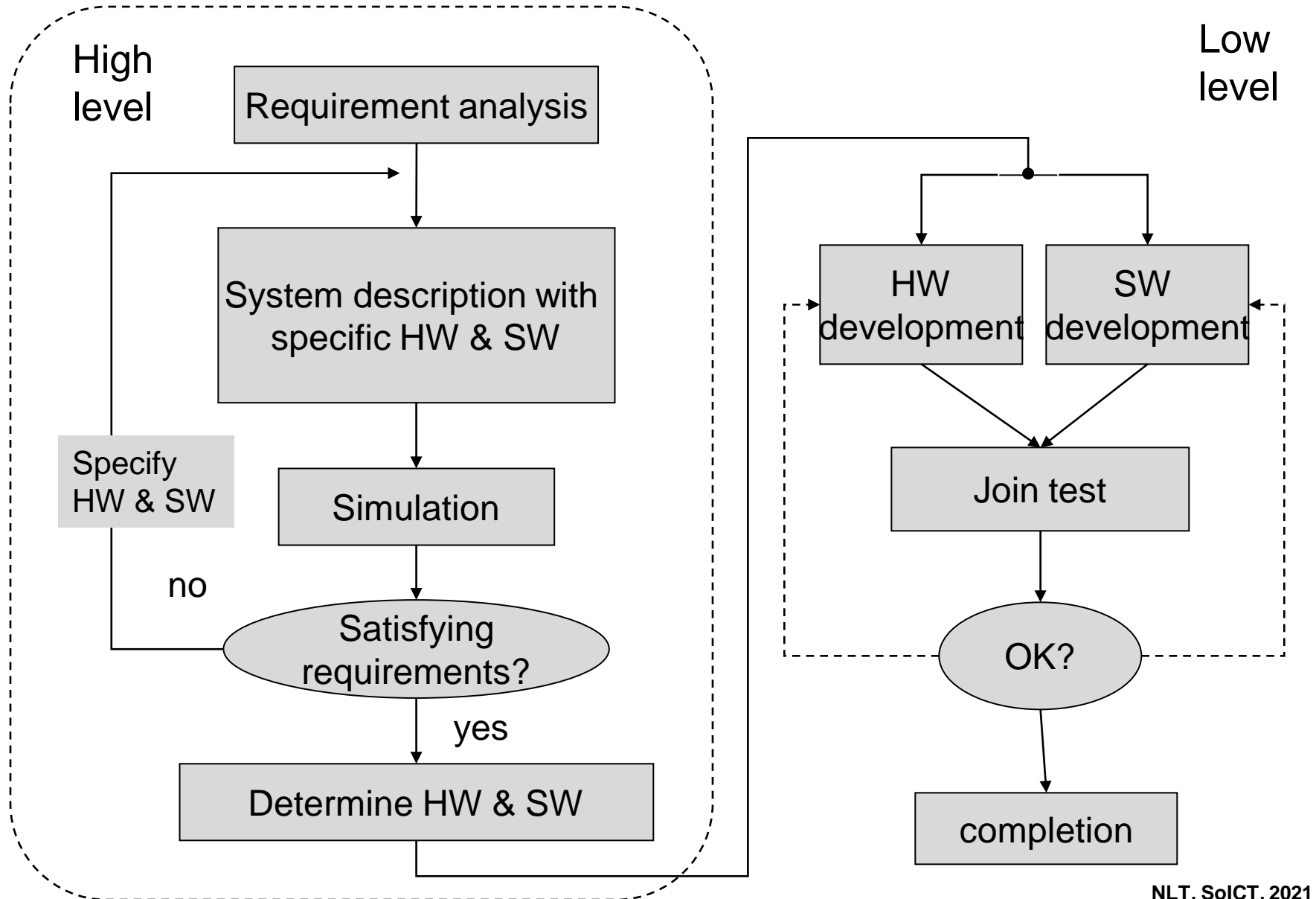
Hardware and software co-design model

- ❑ How to design embedded system?
 - ❑ Hardware or 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**

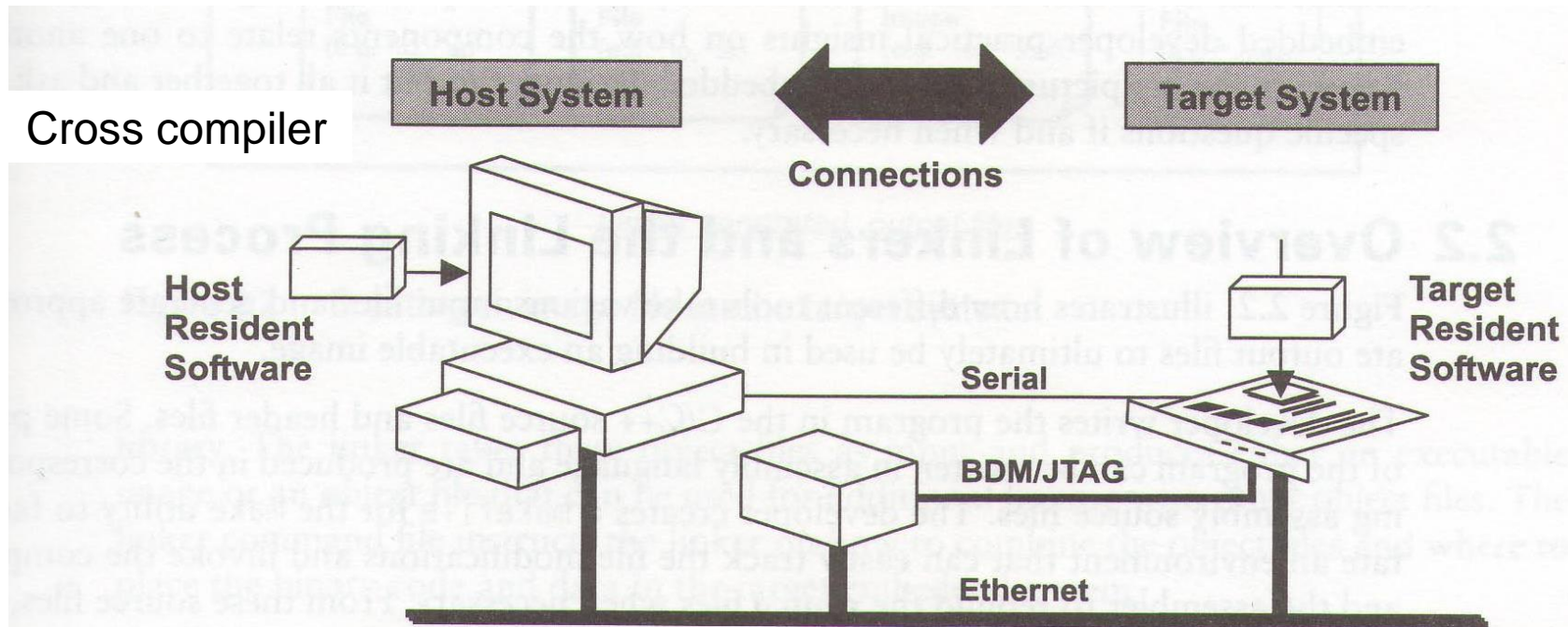
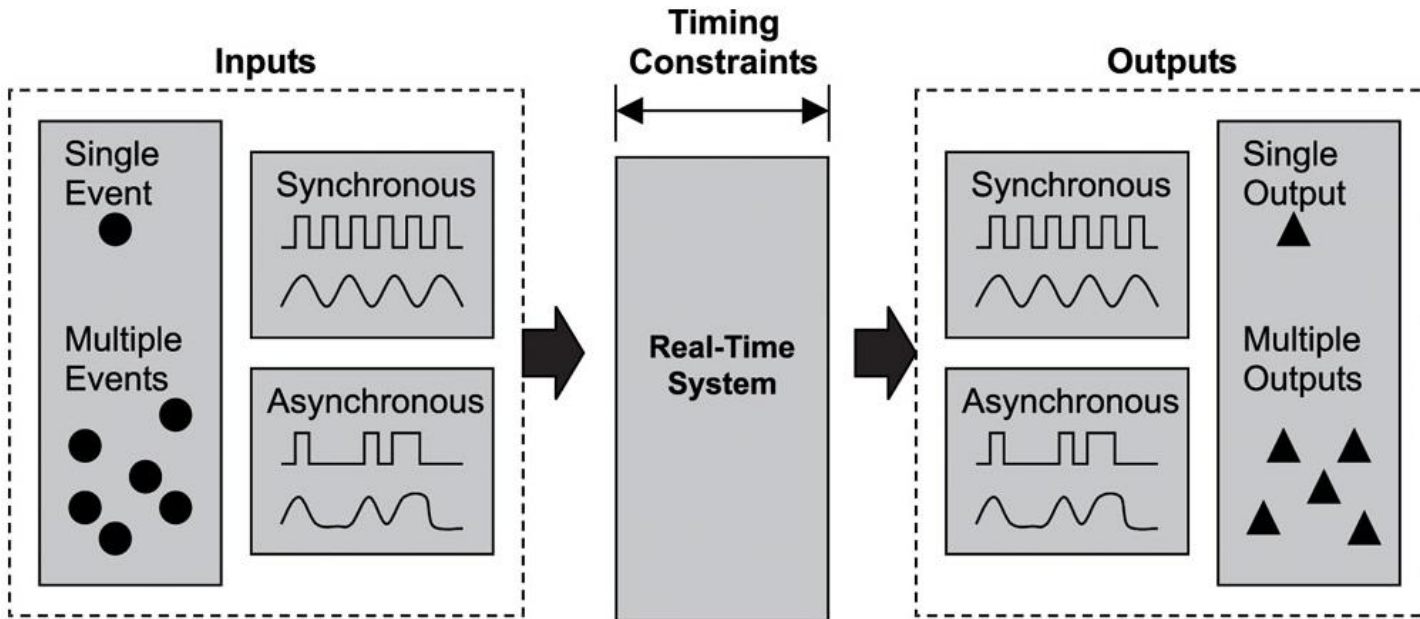


Figure 2.1 Typical cross-platform development environment.

3. Real time systems

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/asperiodic
- ❑ Both of the **timing correctness** and **logical correctness** are important.

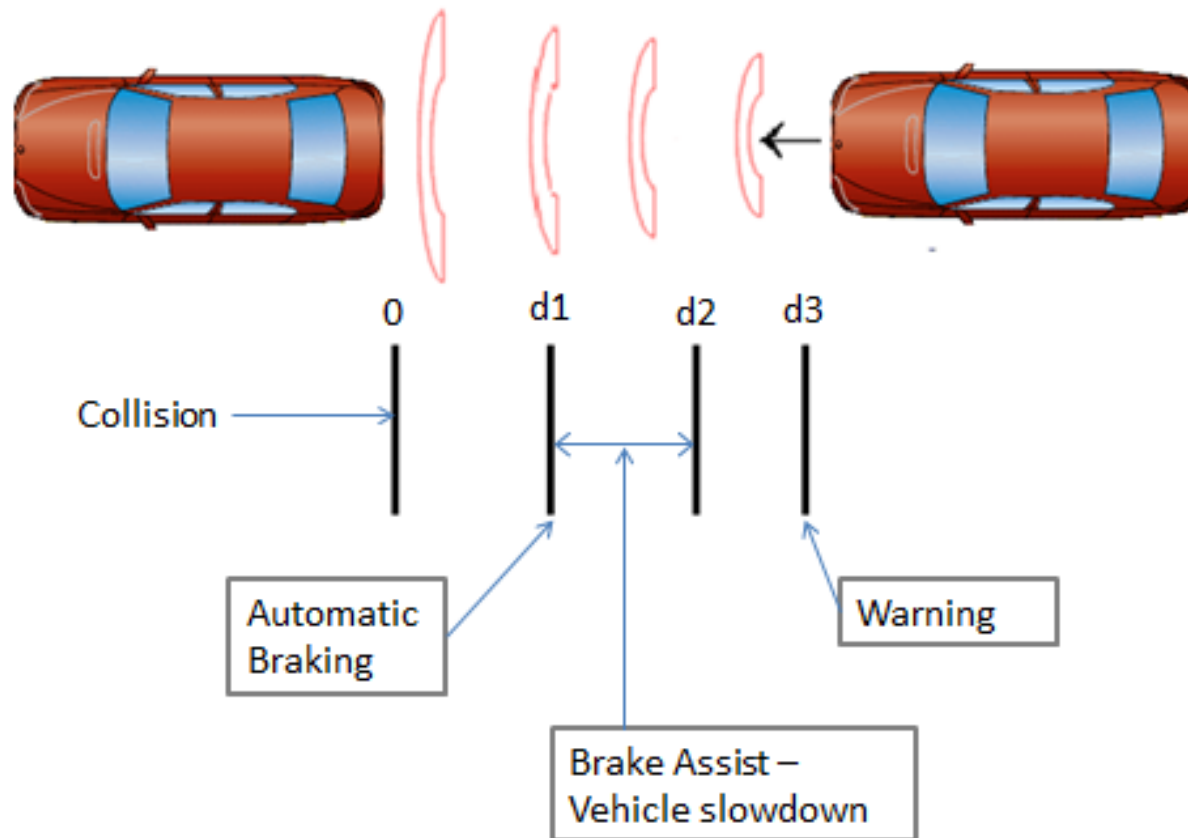
Real-time embedded systems

- ❑ Example: DVD player

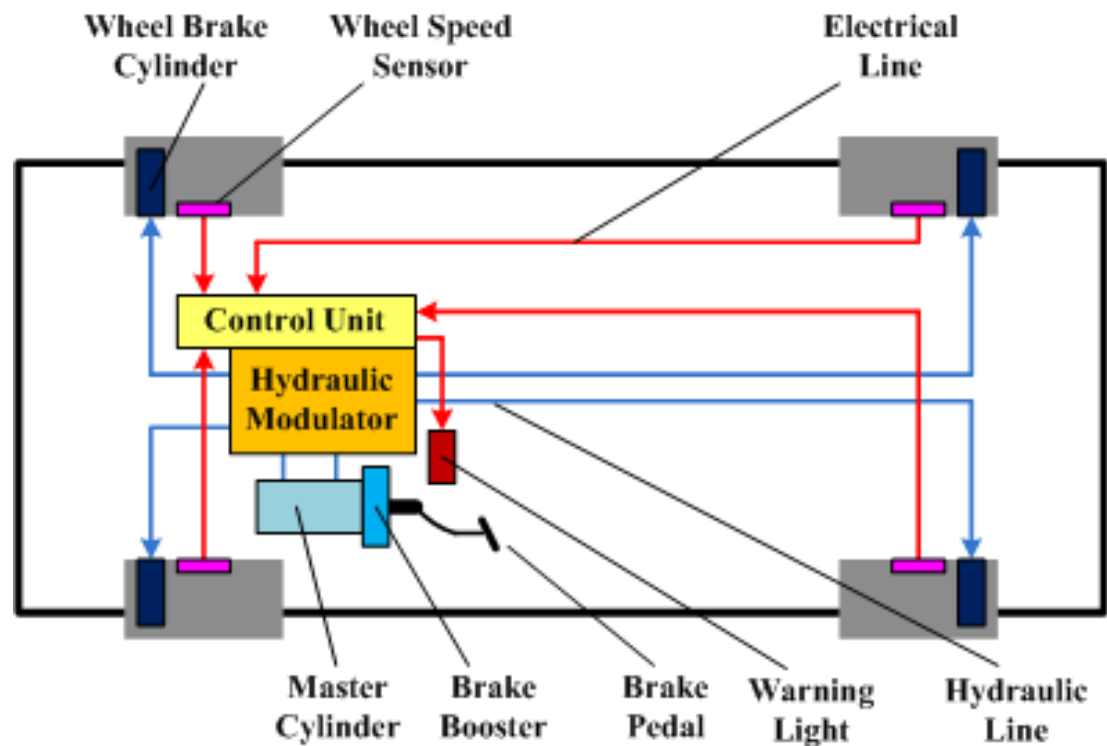


What are real-time requirements?

❑ 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

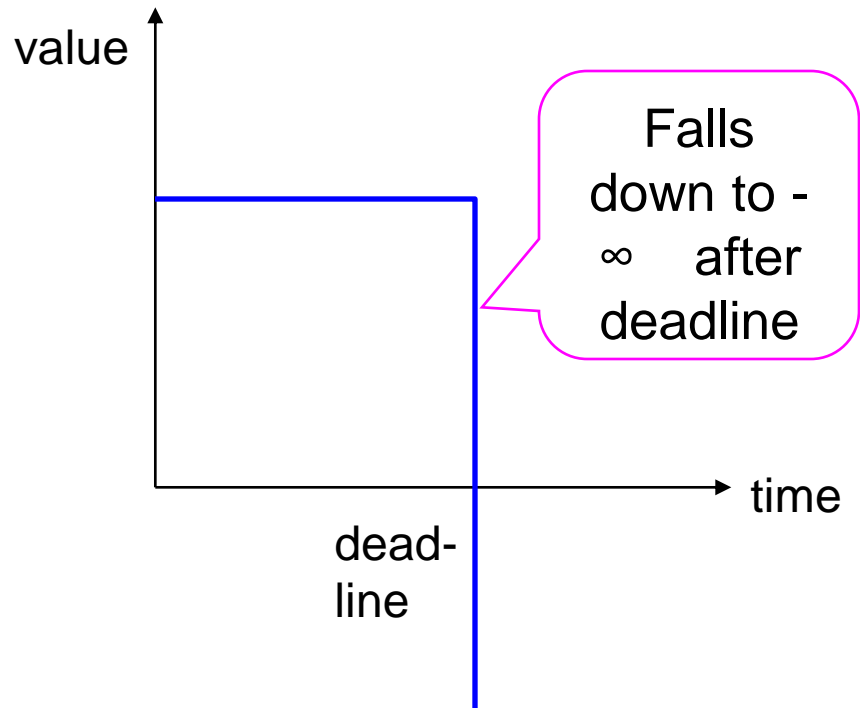
□ **Soft real-time systems**

- 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

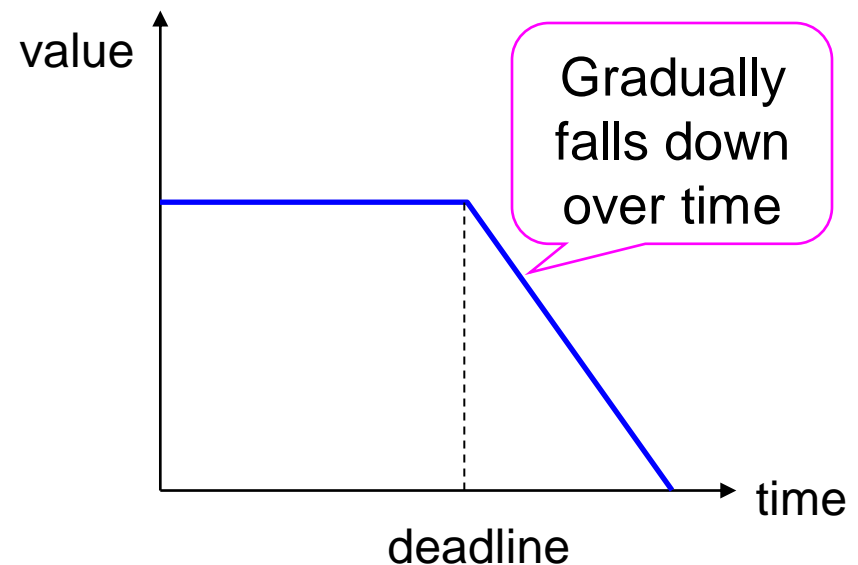
Penalties in real-time systems

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

Hard real-time

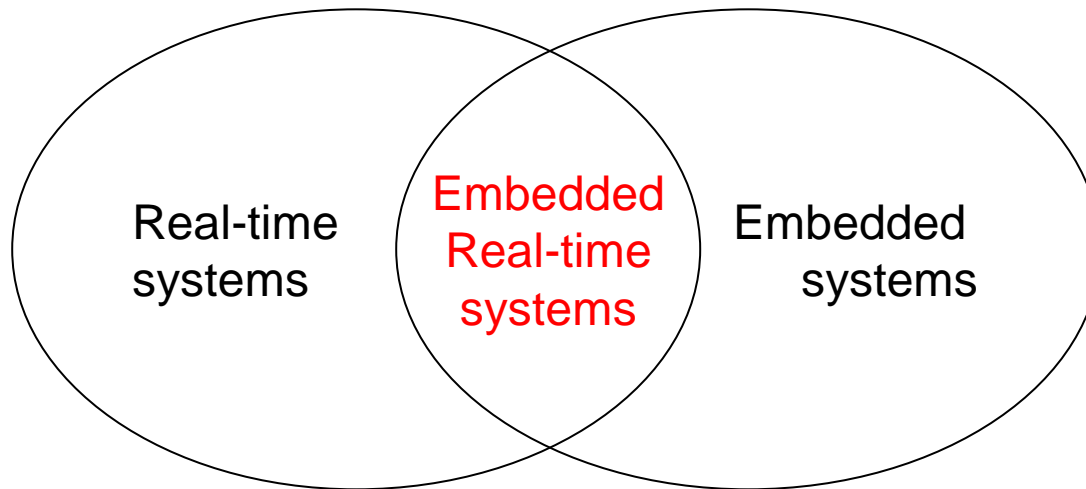


Soft real-time



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

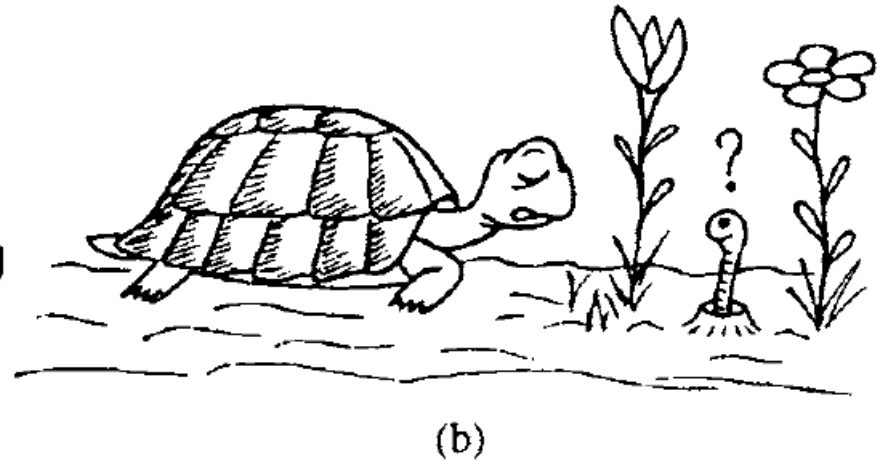
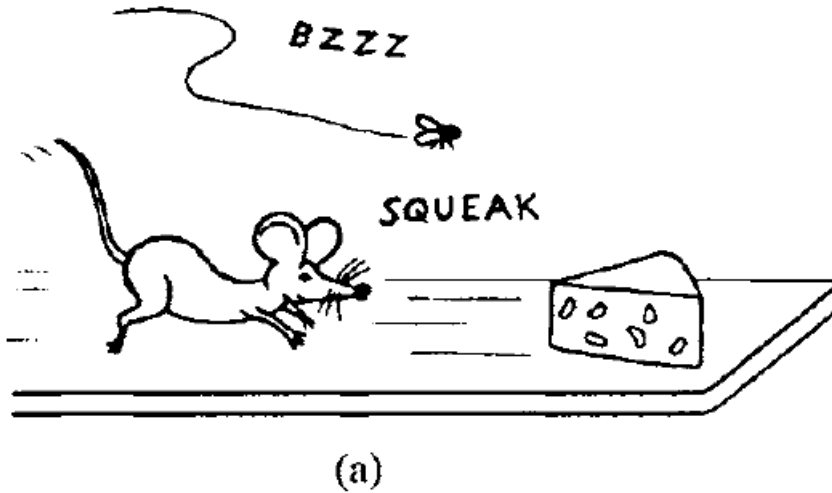
- ❑ Real-time: required timing & function correctness
- ❑ Commutative between embedded systems and real-time systems

❑ Hard/soft real-time systems

- ❑ Determined by penalty in deadline misses

Comparison

❑ Real-time vs real-fast?



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?

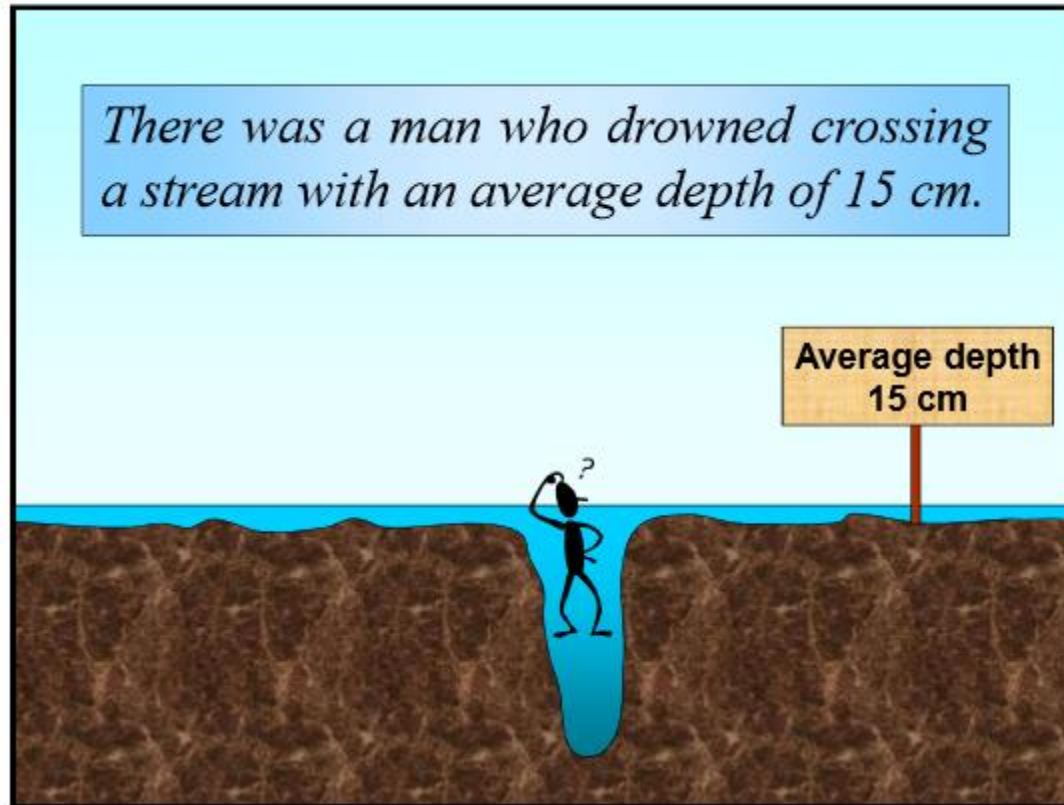
Don't trust average

```
centos@ec2-54-211-215-59.compute-1.amazonaws.com:22 - Bitwise xterm

1 [||||||||||||||||||||| 57.6%] Tasks: 71, 42 thr; 6 running
2 [||||||||||||||||||||| 58.3%] Load average: 0.95 1.83 2.09
Mem[|||||||||||||||||||||799M/3.52G] Uptime: 43 days, 16:43:21
Swp[||||||||||||||||| 0K/0K]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
16105 centos      20   0  119M   2172  1452  R   1.3  0.1   0:00.13 http
7758  nginx       20   0 48372   2608  1048  S   0.7  0.1   1h19:31 nginx: worker process
4507  mysql       20   0 1595M   238M   6944  S   0.7  6.6   9h59:46 /usr/libexec/mysqld --basedir=/usr --
7711  nginx       20   0   558M 24440 14232  S   0.7  0.7   3:06.89 php-fpm: pool www
4517  mysql       20   0 1595M   238M   6944  S   0.7  6.6   2:06.02 /usr/libexec/mysqld --basedir=/usr --
5367  nginx       20   0   561M 25552 14284  S   0.7  0.7   3:08.37 php-fpm: pool www
15800 mysql       20   0 1595M   238M   6944  S   0.0  6.6   0:00.44 /usr/libexec/mysqld --basedir=/usr --
23868 nginx       20   0   557M 24860 15088  S   0.0  0.7   3:09.76 php-fpm: pool www
7763  nginx       20   0   561M 26868 15860  S   0.0  0.7   3:06.59 php-fpm: pool www
23864 nginx       20   0   559M 25280 15668  S   0.0  0.7   3:04.89 php-fpm: pool www
4514  mysql       20   0 1595M   238M   6944  S   0.0  6.6   2:09.42 /usr/libexec/mysqld --basedir=/usr --
23827 nginx       20   0   560M 26180 14408  S   0.0  0.7   3:08.23 php-fpm: pool www
7764  nginx       20   0   559M 24508 14848  S   0.0  0.7   3:01.33 php-fpm: pool www
5427  nginx       20   0   561M 25988 15160  S   0.0  0.7   3:10.29 php-fpm: pool www
5435  nginx       20   0   557M 22944 13552  S   0.0  0.6   3:03.22 php-fpm: pool www
7765  nginx       20   0   558M 23568 13472  S   0.0  0.6   3:01.91 php-fpm: pool www
5330  nginx       20   0   558M 24780 14272  S   0.0  0.7   3:07.31 php-fpm: pool www
7760  nginx       20   0 48440   2700   1044  S   0.0  0.1   1h20:55 nginx: worker process
11067 nginx       20   0   557M 20480 10868  S   0.0  0.6   0:53.14 php-fpm: pool www
23869 nginx       20   0   560M 25208 15128  S   0.0  0.7   3:04.55 php-fpm: pool www
5473  nginx       20   0   558M 24528 13752  S   0.0  0.7   3:07.82 php-fpm: pool www
7759  nginx       20   0 48328   2584   1040  S   0.0  0.1   1h19:51 nginx: worker process
4524  mysql       20   0 1595M   238M   6944  S   0.0  6.6 34:40.11 /usr/libexec/mysqld --basedir=/usr --
5443  nginx       20   0   560M 25100 15072  S   0.0  0.7   3:04.27 php-fpm: pool www
4511  mysql       20   0 1595M   238M   6944  S   0.0  6.6   2:08.50 /usr/libexec/mysqld --basedir=/usr --
4520  mysql       20   0 1595M   238M   6944  S   0.0  6.6   3:42.73 /usr/libexec/mysqld --basedir=/usr --
7574  nginx       20   0   560M 25332 15104  S   0.0  0.7   3:06.14 php-fpm: pool www
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

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>