Real Time Systems
Introduction

Radek Pelánek

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

EVROPSKÁ UNIE esf MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost UNIVERSITAS MASARYKIANA BRUNENSIS

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Organization of the Course

- language
  - materials, written communication – should be in English
  - oral communication – English, Czech
- active lectures
  - exercises during lectures
  - lab sessions (B130)
- evaluation:
  - 4 assignments (50 points)
  - final test (50 points)
  - minimal requirement: at least 50% from each part

Materials

- course content based mainly on books (these are not easily available)
- course web page:
  `http://www.fi.muni.cz/~xpelanek/IA158/`
  - slides (optimized mainly for lecture, not for self-study)
  - references to relevant articles
- $\Rightarrow$ you should attend lectures

Assignments

1. Scheduling (pen and pencil)
2. Programming (C/C++ and POSIX or Java)
3. System construction (Lego Mindstorms)
4. Verification (Uppaal tool)

This is real time course $\Rightarrow$ deadlines are strict.

This Lecture

1. introduction, basic notions
2. examples of real time systems
3. overview of the course
4. puzzles

What are Real Time System?

Definition (Real time system)

A real time system is a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure.

Definition (Real time system)

A real time system is one whose logical correctness is based on both the correctness of the outputs and their timeliness.

Definition (Real time system)

A real time system is any information processing activity or system which has to respond to externally generated input stimuli within a finite and specified period.

Related Notions

reactive system	continuous interaction with the environment (as opposed to information processing)
embedded system	computer system encapsulated in its environment (device it controls), combination of computer hardware and software, dedicated to specific purpose
safety-critical system	a failure may cause injury, loss of lives, significant financial loss

Examples

Are there any examples in this room (building)?
real time system, reactive system, embedded system, safety-critical system

Example from (2010) News

Toyota "sudden acceleration problem"

- 2010 version:
  - sudden accelaration of cars
  - fault in electronic system?
  - related to our concepts – real-time system, reactive system, embedded system, safety-critical system

Example from (2010) News

Toyota "sudden acceleration problem"

- 2010 version:
  - sudden accelaration of cars
  - fault in electronic system?
  - related to our concepts – real-time system, reactive system, embedded system, safety-critical system
- 2011 version:
  - "pedal misapplication" (accelerator, brake)

Embedded Systems

- major application of real time concepts
- important application: it is estimated that 99 % of all processors go into embedded systems
- we will not consider embedded systems per se, but you should have them in mind

Block Diagram of RT System

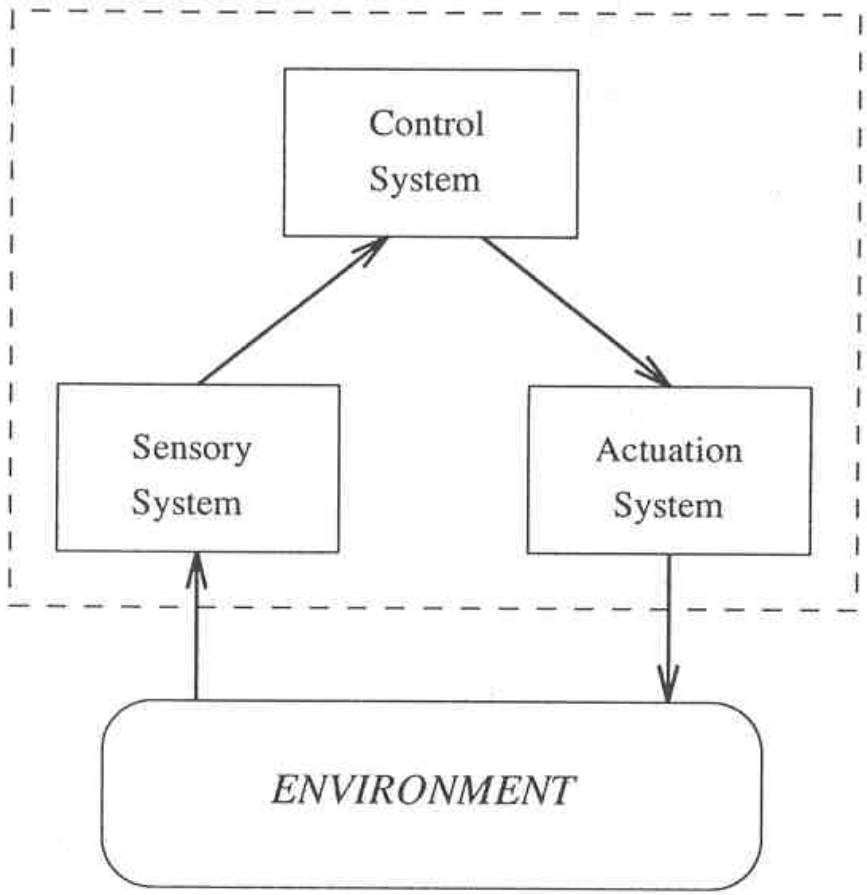


Figure 1.2 Block diagram of a generic real-time control system.

What is Time?

- definitions:
  - The measured or measurable period during which an action, process, or condition exists or continues. (Merriam-Webster)
  - The inevitable passing of events from past to present then future. (Wiktionary)
- measure (second):
  - 1/86400 of a mean solar day
  - duration of 9192631770 periods of the radiation corresponding to the transition between two hyperfine levels of the ground state of the caesium-133 atom

for details visit suitable philosophy or physics course

Real Time vs Fast

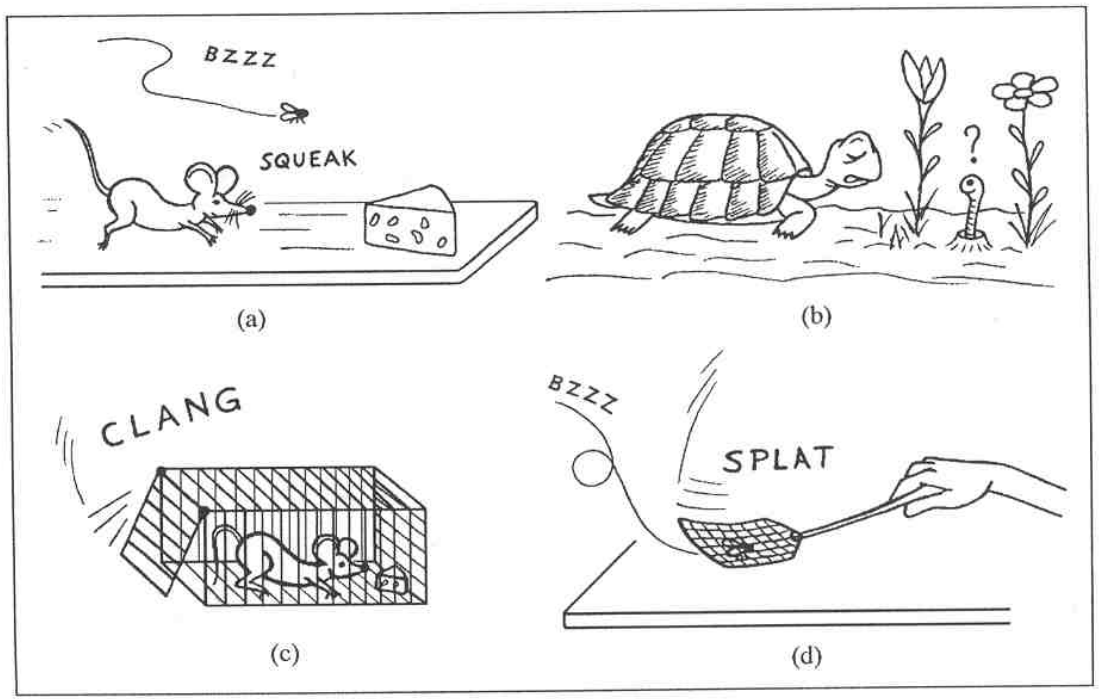


Figure 1.1 Both the mouse (a) and the turtle (b) behave in real time with respect to their natural habitat. Nevertheless, the survival of fast animals such as a mouse or a fly can be jeopardized by events (c and d) quicker than their reactive capabilities.

Time must be considered relatively to the environment.

Real Time vs Fast

There was a man who drowned crossing a stream with an average depth of 15 centimeters.

- fast $\sim$ low average time
- real time $\sim$ predictability, bounded worst case time

Soft and Hard Real Time

deadline – a time within which the task should be completed

hard RT system missing a deadline: failure of the system
aircraft control, nuclear plant control, detection of critical conditions, ...

soft RT system missing a deadline: undesirable for performance reasons
multimedia application, booking system, displaying status information, ...

Soft and Hard Real Time (cont.)

- most systems: combination of both hard and soft deadlines
- firm deadline: missing a deadline makes the task useless (similar to hard deadline), however the deadline may be missed occasionally (similar to soft deadline)
- generalization: cost function associated with missing each deadline

Characteristics of RT Systems

- mixture of hardware and software: use of special purpose hardware and architectures
  (not covered)
- concurrent control of separate system components: devices operate in parallel in the real-world, better to model this parallelism by concurrent entities in the program
  (covered)
- extreme reliability and safety: RT systems are usually safety-critical
  (covered)

Predictability

- predictability is one of the most important
- predictability is one of the most difficult to achieve:
  - cache, DMA, interrupt handling
  - memory management
  - priority inversion
  - difficult to calculate worst-case execution times
  - ...

Examples

- most of the course – abstract models of RT system
- now – several concrete examples

Navigation System

- aircraft navigation system
- inputs:
  - x, y, z accelerometer pulses (5ms rate)
  - roll, pitch, yaw angles (40ms rate)
  - temperature (1s rate)
- output:
  - compute actual velocity (40ms rate)
  - output velocity do display (1s rate)

processes are concurrent and have different rates

Nuclear Plant Monitoring System

- monitoring system for nuclear plant
- event triggered by a signal at various security levels – must respond in 1s
- critical signals (over-temperature of nuclear core) – must respond in 1ms

processes have different priorities, criticality

Airline Reservation System

- reservation of tickets for airlines
- distributed system, several agents may use the system concurrently
- turnaround time less than 15s
- no overbooking

processes share resources

Process Control System

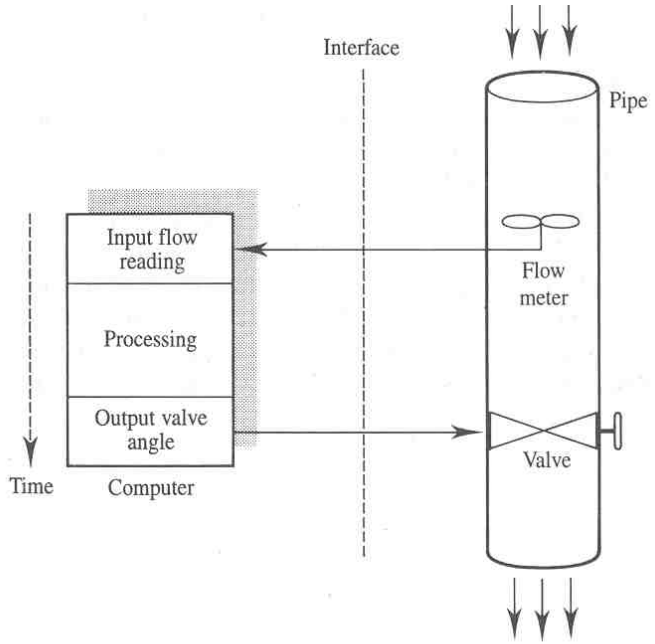


Figure 1.1 A fluid control system.

most of all ... real time!

Process Control System (cont.)

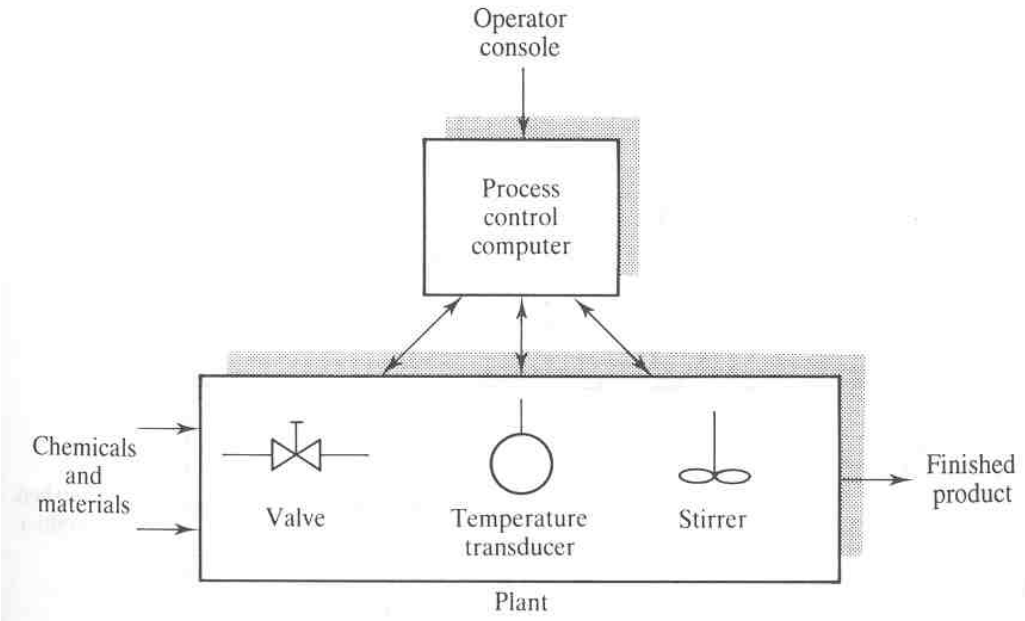


Figure 1.2 A process control system.

real time systems are complex

Production Control System

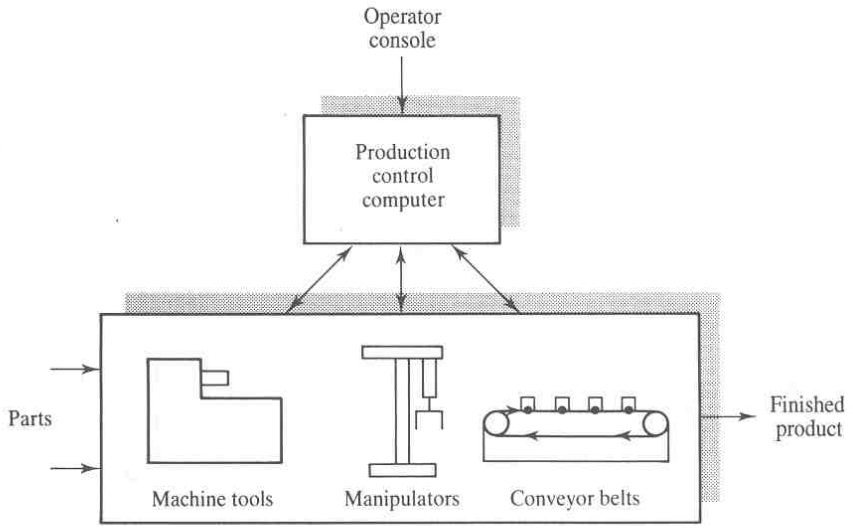


Figure 1.3 A production control system.

and even more complex

Areas of Application

Write down different examples of real-time systems.
Try to formulate ‘areas of application’.

Areas of Application I

vehicle control systems	embedded systems in cars, space missions
transport control systems	railway switching networks, traffic control, air traffic control
plant control	production and manufacturing control, nuclear plants, chemical plants

Areas of Application II

databases	booking systems, telephone switching, radar tracking
home appliances	mobile phones, microwave ovens, washing machines, fridges
image processing	multimedia, mobile phones, digital cameras, industrial inspection systems, medical imaging devices

Infamous Real Time System

- several infamous real time systems
- examples of:
  - what can go wrong
  - significance of consequences
- see also "Collection of Software Bugs"
  `http://www5.in.tum.de/~huckle/bugse.html`

Ariane 5

exploded 40 seconds after start during the first flight (1996)

`http://www.youtube.com/watch?v=kYUrqdUyEpI`

Ariane 5

- disintegration – caused by full nozzle deflection on all engines

Ariane 5

- disintegration – caused by full nozzle deflection on all engines
- nozzle deflections – commanded on basis of data transmitted by inertial reference computer

Ariane 5

- disintegration – caused by full nozzle deflection on all engines
- nozzle deflections – commanded on basis of data transmitted by inertial reference computer
- data – not real data but post-mortem debug information; unhandled floating point exception

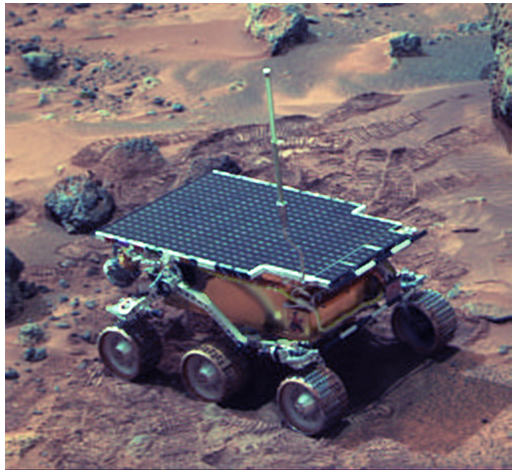
Ariane 5

- disintegration – caused by full nozzle deflection on all engines
- nozzle deflections – commanded on basis of data transmitted by inertial reference computer
- data – not real data but post-mortem debug information; unhandled floating point exception
- exception handling – turned off in order to squeeze CPU utilization

Ariane 5

- disintegration – caused by full nozzle deflection on all engines
- nozzle deflections – commanded on basis of data transmitted by inertial reference computer
- data – not real data but post-mortem debug information; unhandled floating point exception
- exception handling – turned off in order to squeeze CPU utilization
- unexpected value – in a task used for guiding the rocket while still at the launch pad; left running for 40s after lift-off, due to extra time allocated in case of short pauses during countdown

Mars Pathfinder



- unmanned spacecraft, landed on Mars in 1997
- frequent deadlocks $\Rightarrow$ resets, loss of time
- caused by classical priority inversion problem (mutex-protected shared data area)

Apollo 11

- the first landing on the Moon
- software problem during descent – landing nearly aborted
- engineers in charge decided to ignore the problem – later awarded the same medal as astronauts

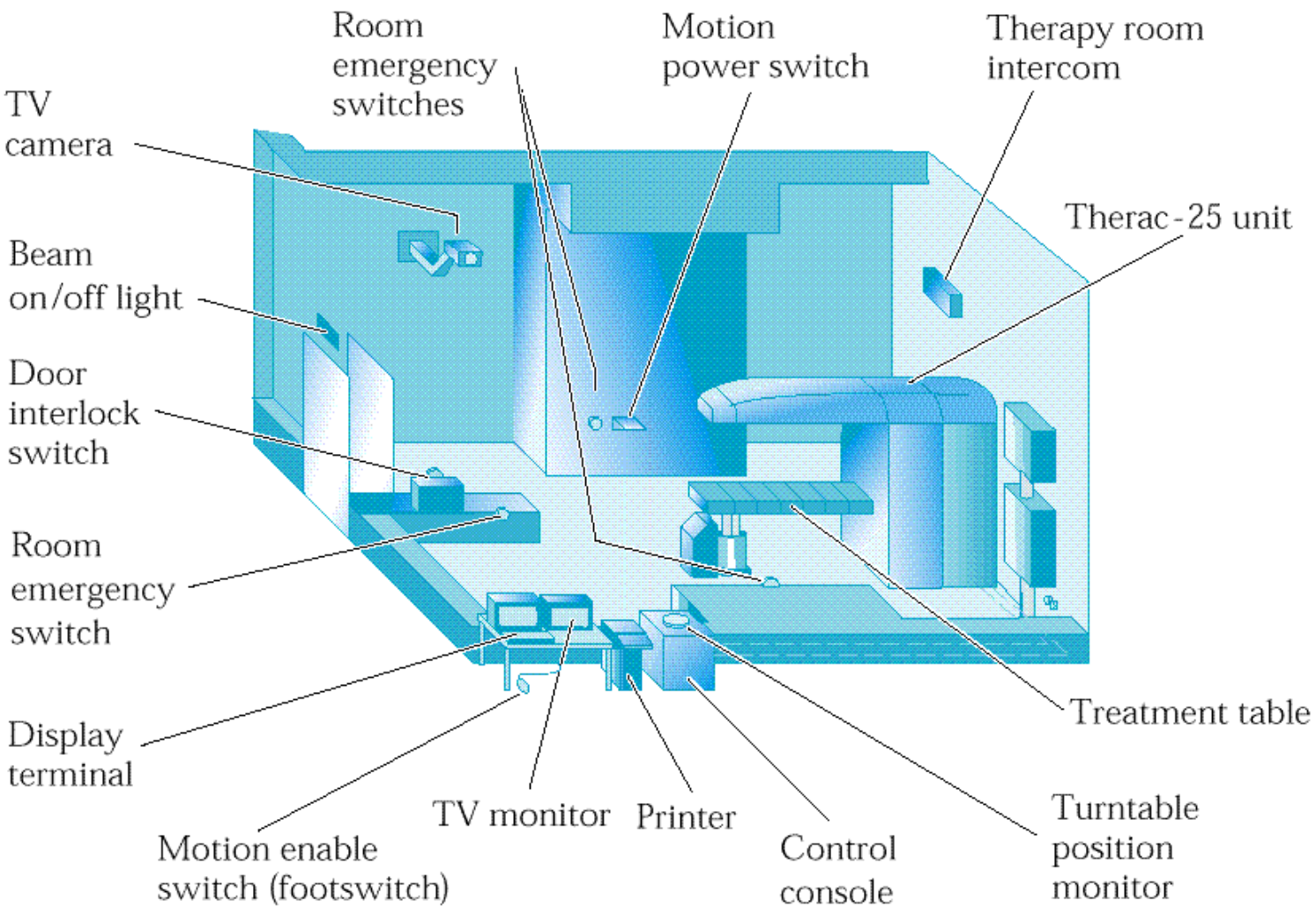
Apollo 11

- spacecraft equipped with a computer for navigation and guidance (programmed in assembler)
- overloaded control system (computer too slow to handle all tasks concurrently) $\rightarrow$ buffer overflow $\rightarrow$ alarm signals
- low-priority jobs were not executed (not critical)

Therac-25

- mid 80', computer controlled therapeutic radiation machine for treatment of tumors
- six deaths and serious injuries due to massive radiation overdoses
- caused by race conditions (wrong mutual exclusion)
- two operation modes: electron mode (low energy), X-ray mode (high energy)

Therac-25


Room emergency switches
Motion power switch
Therapy room intercom
TV camera
Therac-25 unit
Beam on/off light
Door interlock switch
Room emergency switch
Treatment table
Display terminal
TV monitor
Printer
Turntable position monitor
Motion enable switch (footswitch)
Control console

Therac-25: Reconstructed Accident

- operator erroneously enters X-ray mode, realizes the mistake, switches back to electron mode – all within 8 seconds

Therac-25: Reconstructed Accident

- operator erroneously enters X-ray mode, realizes the mistake, switches back to electron mode – all within 8 seconds
- during that time window:
  - treatment phase task is ignoring keyboard input (busy-wait loop)
  - other tasks register the edit

Therac-25: Reconstructed Accident

- operator erroneously enters X-ray mode, realizes the mistake, switches back to electron mode – all within 8 seconds
- during that time window:
  - treatment phase task is ignoring keyboard input (busy-wait loop)
  - other tasks register the edit
- unshielded high energy radiation, no indication to the operator

Patriot Missile Control System

- system used to protect Saudi Arabia during Gulf War
- detects flying objects, performs prediction; trajectory matches prediction $\Rightarrow$ Patriot missile launched
- 25. 2. 1991 - Scud missile hit city of Dhahran, classified as false alarm (no Patriot missile launched)
- software bug: real-time clock accumulating a delay of 57 microseconds per minute; 100 hours $\Rightarrow$ 343 milliseconds

Lessons To Be Learned

- if something can go wrong, it will go wrong
- argument "it works now" has little value for a real time system
- testing can find many errors, but never gives full correctness guarantees
- correctness should be ideally established by a formal verification with clearly stated assumptions and assertions

Therefore this course gives focus on formal treatment and verification.

Objectives of the Course

After the course students should:

- Know specific aspects of real time systems.
- Understand main problems of the design of real time systems and know some solutions.
- Be able to use formal reasoning about real time systems.
- Have a practical experience with a real time system.

Topics

- scheduling
- programming
- verification

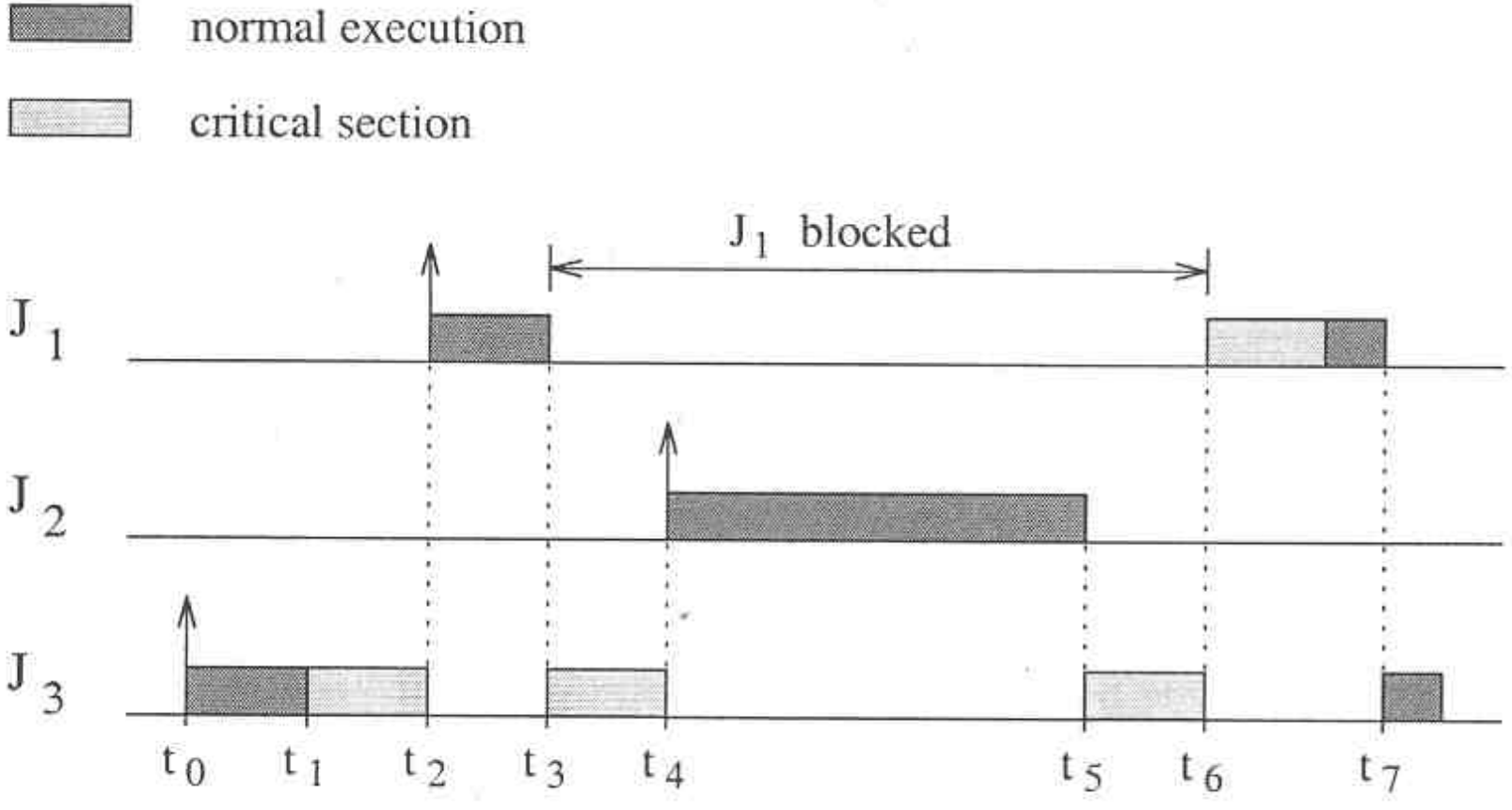
recurring (connecting) theme: mutual exclusion

Scheduling

- input:
  - available processors, resources
  - set of tasks (requirements, deadlines, dependencies ...)
- question: how to assign processor/resources to tasks so that all requirements are met?
- example:
  - 1 processor, jobs are preemptable
  - job 1: release time 0, computation time 1, deadline 2
  - job 2: release time 0, computation time 2, deadline 5
  - job 3: release time 2, computation time 2, deadline 4
  - job 4: release time 3, computation time 2, deadline 10
  - job 5: release time 6, computation time 2, deadline 9

Periodicity, Priorities

- periodic jobs, periodic schedules
- priorities of job (different levels of criticality)
- priority inversion problem, solutions, ...



Resource Access Control

- scheduling with resources
- ensuring exclusive access to resources — mutual exclusion problem
- protocols for mutual exclusion, semaphores, ...

Programming

- concurrency
- general concepts
- overview of programming languages (C + POSIX, Java, Ada)
- programming exercise with C + POSIX

Lego Mindstorms Project

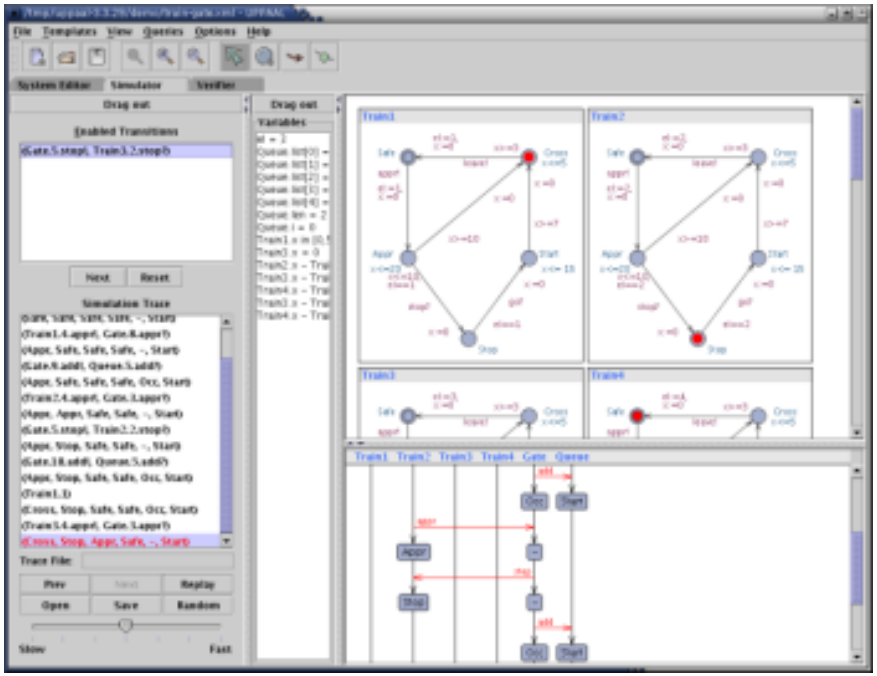
construction and programming of a physical real time system

Verification

- introduction to formal verification
- model checking technique
- basic idea, formal modeling, algorithms
- timed automata formalism

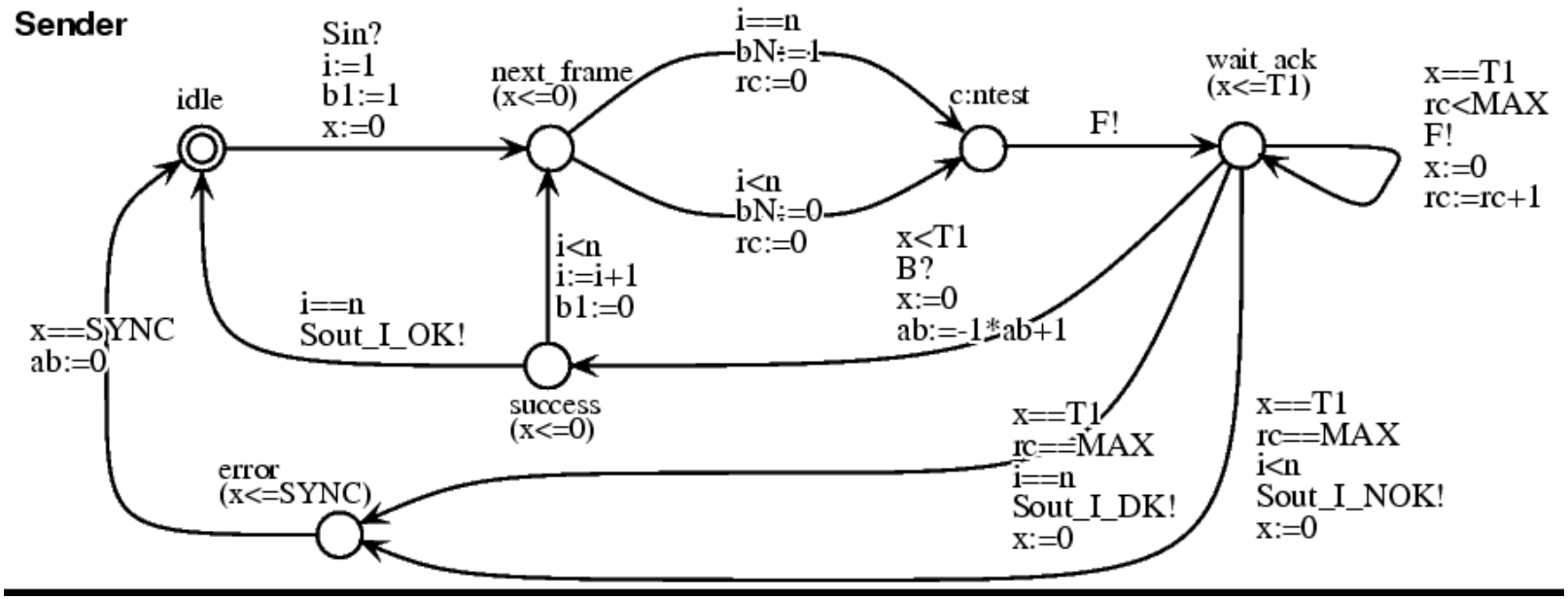
Uppaal

model checking tool for real time systems



Verification Case Studies

Example: Bounded Retransmission Protocol



Puzzles

puzzles illustrating some of the main concepts:

- scheduling
- deadlines
- shared resources, constraints
- concurrency
- prove of infeasibility

Toasts Puzzle

- toast: each side 2 minutes on a pan
- pan: two toasts at a time
- what is the minimum time to make three toasts?
- draw a diagram of an optimal "schedule"

Toasts Puzzle II

- toast both side, one side has to be buttered (after toasting that side)
- time requirements:
  - putting toast on/out/turning: 3 s
  - toasting one side: 30 s
  - buttering: 12 s
- what is the minimum time to make three toasts ?

Bridge Puzzle

- 4 men, river, bridge, night, 1 flashlight
- at most 2 men on a bridge, flashlight necessary
- flashlight cannot be thrown
- wounded men – different time to cross: 5 min, 10 min, 20 min, 25 min
- can they cross in 60 minutes?
- can they cross is less than 60 minutes?

Toasts, Bridge – Concepts

- real time: time to make a toast, time to cross a bridge
- deadline: time to complete the task
- schedule: that's the objective to find
- shared resource (constraint): pan, flashlight
- finding solution – intuition may be sufficient (for a simple puzzle)
- proving optimality (infeasibility of better solution) – formalization necessary, tool support welcomed

Measuring Time

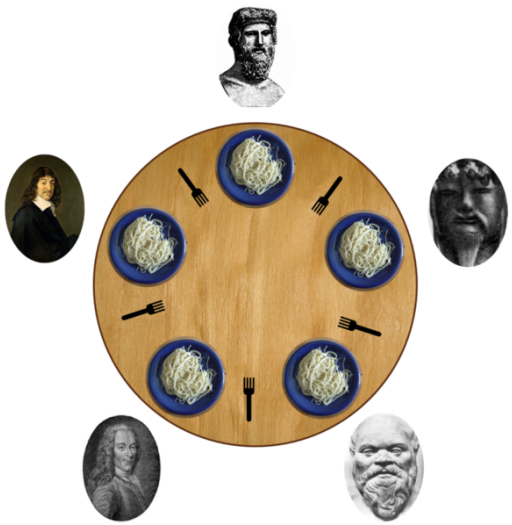
- you have 7 minute and 11 minute hourglasses
- how do you measure 15 minutes? (there are multiple different solutions)
- generalization: a minute and b minute hourglasses, measuring time c

Gossiping Girl Problem

- each girl knows a distinct secret
- girls can talk through phone, during call they exchange all secrets, call takes 1 minute
- communication only in pairs, but calls can be concurrent
- objective: all girls know all secrets
- what is the minimum time to reach the objective (for n girls)
- extension: time dependent on the number of secrets exchanged

Dining Philosophers

- think $\rightarrow$ take left fork $\rightarrow$ take right fork $\rightarrow$ eat $\rightarrow$ drop left fork $\rightarrow$ drop right fork $\rightarrow$ think $\rightarrow$ ...
- possible deadlock
- how to avoid deadlock?



Concurrent Addition Puzzle

$$c := 1, x_1 := 0, x_2 := 0$$

$$\begin{array}{lcl} x_1 := c & & x_2 := c \\ x_1 := x_1 + c & \| & x_2 := x_2 + c \\ c := x_1 & & c := x_2 \end{array}$$

- both processes loop
- arbitrary interleaving
- How can c reach value 5? How can c reach value 13?
- Can c reach any natural value?

Gossip, Philosophers, Addition – Concepts

- concurrency: several "processes" active in parallel
- shared resources: phones, forks, shared variable c
- interleavings: source of complexity

Summary

- course: 4 assignments, active participation quite important
- today: real time system properties – illustrated on sample examples, puzzles
- next: abstract model of real time system, scheduling