

# TTIT62 Real-time Process Control

## Lecture 1: Introduction

Simin Nadjm-Tehrani

Real-time Systems Laboratory  
Department of Computer and Information Science  
Linköping university



## What is real-time systems?

*Real-time systems: The time that the result of the computation is delivered is as important as the result itself*

- A control system should be fast enough relative developments in its environment
- Predictability!

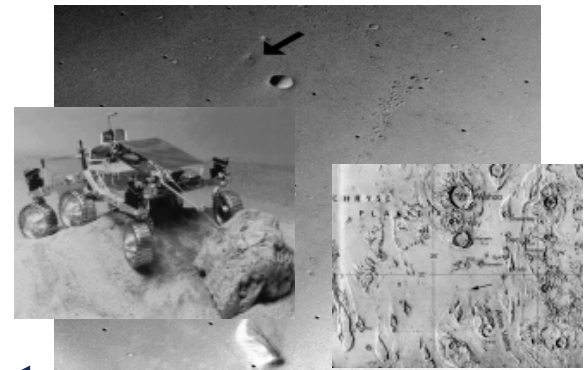


## Predictability...

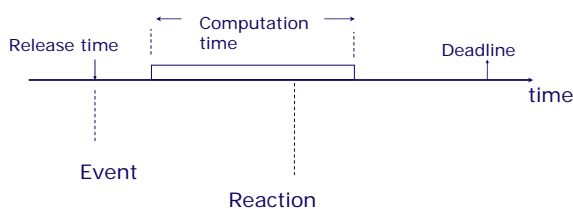
The film...



## Really good example



## Events and computations



## Predictability and Criticality

- Hard RT
  - Not meeting a deadline leads to system failure
- Soft RT
  - Deadlines should be met, but system functions correctly if they are *occasionally* missed
    - How often? By how much?
- Firm RT
  - OK for deadlines to be *occasionally* missed, but after the deadline the result is of no use



## Hard real-time

- If the sensor detects a crash, the air bag in the car should be activated within a given time
- If the signal is red, the train must be stopped within a given time
- If the liquid level has reached the max. value, the valve must be opened within a given time



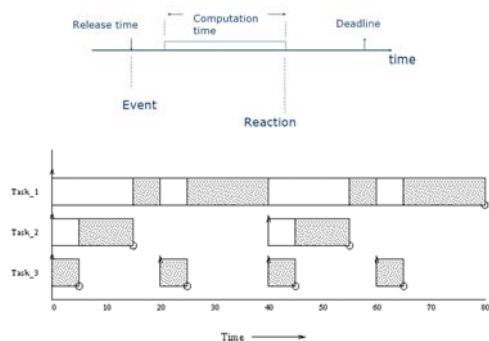
## Scheduling

... is about allocating resources, specially the CPU time, among all the processes such that the timeliness requirements are met.

The scheduler decides which process is to be run next.



## Multiple tasks & parameters



## Sharing the brain

- Labs based on football playing robots
  - computational processes given
  - parameters given
  - how to allocate CPU so that the team member that needs the ball most gets to play it?
- Implement three schedulers and test in a simulated environment



## Sharing Resources

- Resource sharing
  - Mutex, Locks
  - Deadlocks, starvation
- Real-time systems + resource sharing
  - When more than one resource type is shared
  - Will the deadline be met for all tasks?



## Dependability

- Systems cannot act predictably *in time* if they do not deliver their services at all!
- How can we produce systems that do their job, and how *to measure* how well they do their jobs?
  - How do things go wrong and why?
  - What can we do about it?

