Course Information

- Meeting time and room: TTh 12.30 - 1.45, deBartolo 207
- Instructor: Christian Poellabauer
  354 Fitzpatrick, cpoellab@cse.nd.edu
  Office hours:
  - Monday 9am-10am
  - Tuesday 11am-noon
  - by appointment (email works best) or come in when door open
- http://www.cse.nd.edu/~cpoellab/teaching/cse40463/
- TA: Pramita Mitra
  214 Cushing, pmitra@cse.nd.edu
  Office hours:
  - Tuesday/Thursday 4pm-5pm
  - by appointment

Important Dates

- Midterm break: October 23/25
- Thanksgiving: November 22
- One-page project proposal due: October 2
- Two-page project proposal due: October 16
- Progress report for project: November 20
- Term project report due: December 14
- Term project presentations and demos: TBD

Course Structure

- 1st half:
  - lectures (slides, papers will be provided)
  - around 3 homeworks
  - small programming assignments
  - participation
  - brainstorming for course project
- 2nd half:
  - seminar (each student 2 papers)
  - paper reviews
  - course project
  - participation
  - 0-2 homeworks

Grading Policy

- Programming assignments: 25% (collab: no!)
  - around 3 coding/measurement challenges
- Homeworks: 25% (collab: no!)
  - 3-4 (each 1 week), mostly in 1st half of semester
- Course project: 25% (collab: yes)
  - 1-2 students/team, 1st half: problem finding, 2nd half: execution
- Participation: 25% (collab: yes)
  - questions, discussions, paper reviews, progress reports, paper presentations, project presentation, etc.

What is a Real-Time System?

- A real-time system (defined by IEEE) is a system whose correctness includes its response time as well as its functional correctness. In other words, in a real-time system, it not only matters that the answers are correct, but it matters when the answers are produced. Note that by this definition, systems requiring a defined Quality of Service are usually real-time systems, although they might not use those words to describe themselves.

Introduction: Key Aspects

- RTS interact with environment: I/O
  - size/complexity of system
  - ranging from interrupts sent to custom hardware (embedded system), that reacts immediately
  - to larger systems based on Real-Time Operating Systems that support hard real-time constraints
  - to general-purpose systems that support soft constraints.
- Response time constraints: Scheduling
  - small embedded systems: one single task with well-defined (static) execution cycles
  - large systems with OS and many tasks: scheduling based on some kind of priority
Definitions

- **System**: A system is a mapping of a set of inputs into a set of outputs.
- **Response time**: The time between the presentation of a set of inputs to a system (stimulus) and the realization of the required behavior (response), including the availability of all associated outputs, is called the response time.
- **Real-time system**: A real-time system (RTS) is a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure.
- **Failure**: A failed system is a system that cannot satisfy one or more of the requirements stipulated in the formal system specification.

**Soft RTS**: A soft RTS is one in which performance is degraded but not destroyed by failure to meet response-time constraints.

**Hard RTS**: A hard RTS is one in which failure to meet a single deadline may lead to complete and catastrophic system failure.

**Firm RTS**: A firm RTS is one in which a few missed deadlines will not lead to a total failure, but missing more than a few may lead to complete and catastrophic failure.

---

More Definitions

- **Event versus State**: the door is 30º open, the valve is 45% open, the tank is 20% full, the speed is 35mph, the temperature is 140ºF
  - door opens by 5º, the valve closes by 2%, the tank is filled to 40%, the car slows down to 20mph, the temperature sinks by 5ºF
- **Release time**: The release time is the time at which an instance of a scheduled task is ready to run.
- **Periodic/aperiodic/sporadic**: The rate at which events take place can be periodic, aperiodic, and sporadic.
- **Determinism**: A system is deterministic if, for each possible state and each set of inputs, a unique set of outputs and next state of the system can be determined.

**Utilization**: The (CPU) utilization or time-loading factor, $U$, is a measure of the percentage of non-idle processing.

$$U = \sum e_i/p_i \ (i=1..n)$$
Real-Time Embedded Systems

```
(r(t) \rightarrow A/D \rightarrow r_k \rightarrow Controller \rightarrow D/A \rightarrow y_k \rightarrow y(t) \rightarrow y(t) \rightarrow u(t) \rightarrow u_k \rightarrow A/D \rightarrow A/D)
```

Sensor \rightarrow Plant \rightarrow Actuator

“Embedded Systems”
- PDA, cell phone
- Digital camera
- iPod
- Television, game console
- household appliances
- ATM
- Satellite
- Car, Train, Airplane
- Air traffic control
- More than 95% of all microprocessors are used in real-time embedded systems!

Example: Car
- Operating environment: road conditions, other cars
- Controlling system:
  - human driver (sensors?)
  - computer (sensors?)
- Controls:
  - accelerator, brake pedal, steering wheel
- Actuators:
  - wheels, engines, brakes
- Cruise control:
  - regulates speed of car by adjusting throttle
  - measures speed through device connected to drive shaft
  - hard real-time: drive shaft revolution events
  - soft real-time: driver inputs, throttle adjustments

“Ubiquitous Computing”
- Mark Weiser (1991 article)
- Goal: get computers out of the way of everyday activities
- “Our computers should be like an invisible foundation that is quickly forgotten but always with us, and effortlessly used throughout our lives.” (Mark Weiser)
- “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” (Mark Weiser)
- Origin: Xerox Palo Alto Research Center (PARC) 1987
- Initial goals: solve problems of personal computer:
  - too complex and hard to use
  - too isolating
  - too demanding of attention
  - too dominating

Ubiquitous Computing
- Challenges:
  - create new relationship between people and computer
  - get the computer out of the way
- Approach taken:
  - defined and created new computing artifacts
- ParcTab
- ScratchPad
- LiveBoard

MediaCup
- everywhere
- integrated
- interconnected
- invisible
Enabling Technologies

- Today technologies are catching up with the vision
- off-the-shelf products
- decent performance/accuracy
- good quality and durability

Examples:
- Nike+iPod
- PIC, Timi
- PDA, smartphone, watches
- WiFi, Bluetooth, GPRS, GSM
- Sensors: location (GPS, ultrasonic, IR, cell-based), environmental (sound, light, pressure), ....

Some Real-Time Applications

Some Real-Time Examples

Some Real-Time Examples

Some Real-Time Examples

Some Real-Time Examples
Some Real-Time Examples

<table>
<thead>
<tr>
<th>Date</th>
<th>Close</th>
<th>Open</th>
<th>High</th>
<th>Low</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>100</td>
<td>98</td>
<td>102</td>
<td>96</td>
<td>500</td>
</tr>
<tr>
<td>1/2</td>
<td>103</td>
<td>100</td>
<td>105</td>
<td>101</td>
<td>600</td>
</tr>
<tr>
<td>1/3</td>
<td>105</td>
<td>103</td>
<td>107</td>
<td>102</td>
<td>700</td>
</tr>
</tbody>
</table>

![Diagram of a graph](image-url)