CprE 488 – Embedded Systems Design

Exam 2 Review

Phillip Jones
Electrical and Computer Engineering
Iowa State University

www.ece.iastate.edu/~phjones
rcl.ece.iastate.edu

This is the end. My only friend, the end. – The Doors

Announcements

- Exam 2: Tuesday Prep-week, in class
  - **One-side** 8.5x11" page of **hand-written** notes
  - 70 minutes
  - Recommend having a calculator
- Two Parts:
  - Part 1: Canvas (bring an electronic device to use)
  - Part 2: Paper written

Exam Focus

- Main topics:
  - Embedded Operating Systems (Lect-05)
  - Software Optimization (Lect-06)
  - Embedded Control Systems (Lect-07)
  - Hardware Acceleration (Lect-08)

- Questions will cover:
  - Conceptual topics (textbook readings and lecture topics)
  - In-class exercises
  - Understanding of lab experiences

Textbook Readings

- **Processes and Operating Systems**
  - Chapter 6 (3rd & 4th Edition)
- **Program Design and Analysis**
  - Chapter 5 (3rd & 4th Edition)
- **Accelerators and Coprocessors**
  - Sections 8.5-8.7 (3rd Edition)
  - Sections 10.4 – 10.6 (4th Edition)

I just know I left it around here, somewhere…

Relevant Labs

- **MP-2:** Digital Camera Design
  - Color space conversion
  - Image processing pipeline
  - Image framebuffers and VDMA
- **MP-3:** Target Acquisition
  - Embedded Linux bring-up
  - Virtual memory and I/O
  - USB driver development
- **MP-4:** UAV Control
  - Intuition and practicalities for PID control

Lectures: Embedded Operating Systems

- **Embedded Operating Systems**
  - Processes and scheduling

Example: Embedded Operating Systems

1. What is the purpose of a context switch? Using assembly (pseudocode is acceptable), describe the main steps required to perform a context switch.

2. For the periodic processes and deadlines given below:
   - Schedule the processes using RMS
   - Scheule using EDF, and compare the number of context switches require for EDF vs. RMS

Process	Cost (ms)	Period (ms)	Deadline (ms)
P1	2	30	30
P2	5	40	40
P3	7	120	120
P4	5	60	60
P5	1	15	15

Lectures: Software Optimization

- **Software Optimization**
  - Understanding processor performance
  - Early optimizations (redundancy elimination, operator simplification, etc.)
  - Loop restructuring

Example: Software Optimization

- Explain how loop unrolling, loop switching, and loop fusion can be applied by a complier to the C code segment below. Describe the potential benefit, and any possible disadvantages from applying these optimizations aggressively.

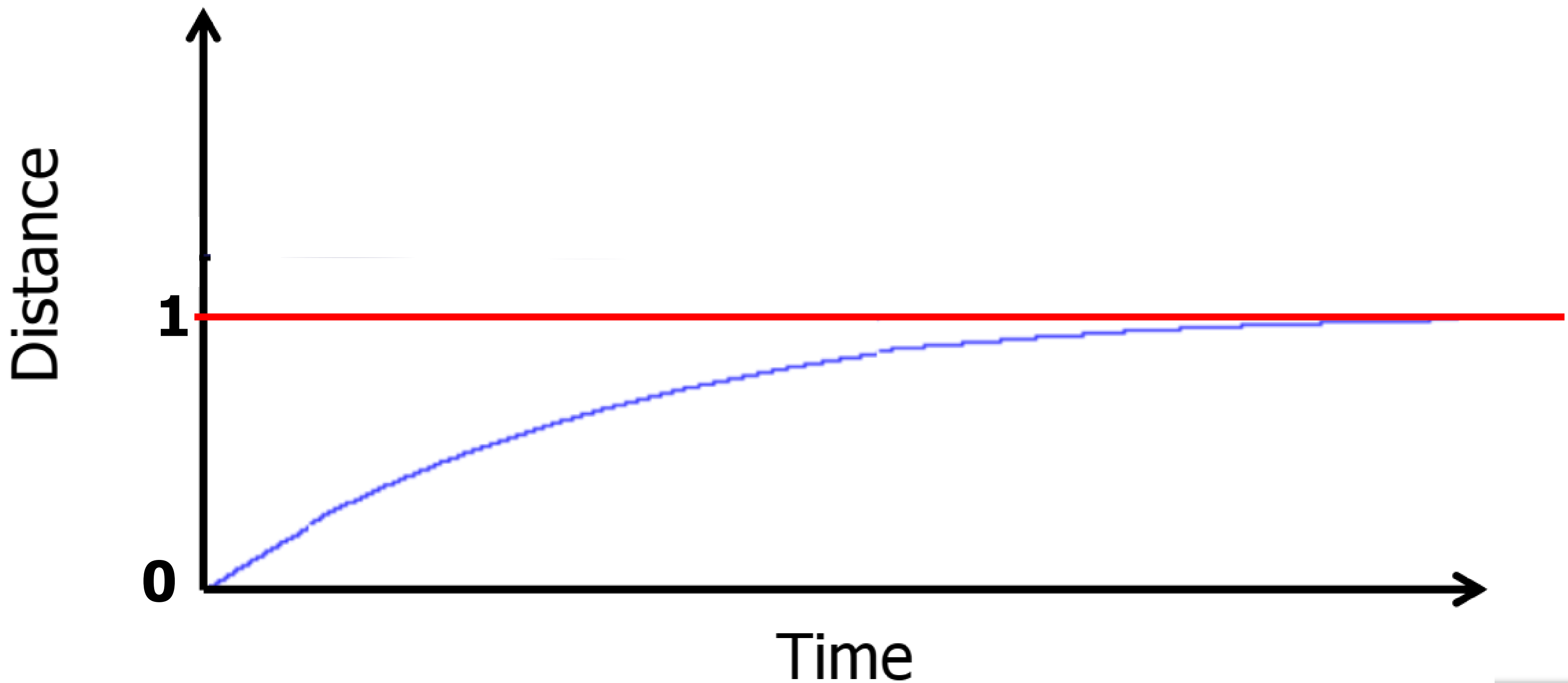
```
for (i = 0; i < N; i++){
   if (j == 23) {
      A[i] = B[i] + 1;
   }
   else {
      A[i] = B[i] + 2;
   }

for (i = 0; i < N; i++) {
   C[i] = A[i] / 2;
}

for (i = 0; i < N; i++) {
   D[i] = C[i+1] * B[i];
}
```

Lectures: Embedded Control Systems

- Embedded Control Systems
  - Open-loop vs. closed-loop
  - PID control (continuous-time and discrete-time form)
  - Understanding P vs I vs D components
  - Nested PID (inverted pendulum example)
  - Model-based control

Example: Embedded Control Systems

- Assume a block is moved up a frictionless ramp from a height of 0m to a height of 1m, and that a PID controller is used to compute the force applied to the block. The plot below shows how the block moves over time. Which PID constant(s) would you changed (and how) to obtain a faster response time?



Lectures: Hardware Acceleration

- Hardware Acceleration
  - Motivation: Moore vs. Dennard vs. Amdahl
  - Coprocessors & Accelerators
  - Performance analysis (**Amdahl's law**)

Example: Hardware Acceleration

- For a randomly generated 32-bit value, how long (in terms of ARM instructions) would the following code take?
- What would a corresponding HW accelerator look like? What would the potential performance savings be?

```
int32_t hamming_distance (uint32_t x, uint32_t y) {
  int32_t dist = 0; uint32_t val;

  val = x ^ y;  // XOR

  // Count the number of bits set to 1
  while (val != 0) {
    if(val & 1)
    {
      dist++;
    }
    val = val >> 1;
  }
  return dist;
}
```