

MTE 241

Computer Structures and Real Time Systems

F'11

MTE 241

□ **Instructor**

- Rudolph E. Seviora
- DC2528, ext 32850, seviora <at> uwaterloo.ca

■ **Lab Instructor**

- Irene Y. Huang

■ **Teaching Assistants**

- Drawil, Nabil
- El-Rayes, Karim
- Kashif, Hany
- Kaleghi, Bahador
- See the Course Syllabus (on ACE) for email addresses, location, phone, office hours

MTE 241

- **Lectures:** Tue, Thu 8:30-10:00
CPH-2387
- **Tutorial 1:** Wed 11:30-12:20; RCH 207
- **Tutorial 2:** Fri 11:30-12:20; RCH 204
- **Makeup Lectures** for Midterm Week:
 - Sept. 21, Wed ; 16:30 - 17:20; MC 1085
 - Oct. 12, Wed ; 16:30 - 17:20; MC 1085
 - Nov. 9, Wed ; 16:30 - 17:20; MC 1085
- **Midterm:** Monday, Oct. 24, time TBD

Course Web Site

UW ACE [uwace.uwaterloo.ca]

- additional lecture material
- project information
- old exams
- assignments + solutions
- plus much more

Lecture/Reference Material

- Course text
 - Gary Nutt, *Operating Systems*, Third Edition, Addison Wesley
- MTE 241 Lecture Transparencies
 - available at U of W Bookstore
- RTX overview document, project specs
 - available for download from course ACE

Course Objectives

Exposure to/familiarization with:

- ❑ Fundamentals of concurrency
- ❑ Standard operating system structures
- ❑ Real-time considerations
- ❑ Common OS [Unix, Windows XX, QNX]

Hands-on experience

- ❑ Design, implementation of a small RTOS
- ❑ Software development for embedded, real-time on a Unix platform
- ❑ Team work

Course Contents

- ❑ Introduction
- ❑ Using the Operating System
- ❑ Operating System Organization
- ❑ Real-Time Executive Design [not in text]
- ❑ Computer Organization
- ❑ Device Management
- ❑ Processes, threads and resources

Course Contents

- ❑ Scheduling
- ❑ Basic Synchronization Techniques
- ❑ High-level synchronization and interprocess communication
- ❑ Memory Management
- ❑ Virtual Memory
- ❑ File Management

Real-Time Operating Systems

- ❑ concept of process and threads
- ❑ internal OS structures
- ❑ scheduling (preemptive – nonpreemptive)
- ❑ interprocess communication (IPC)
- ❑ interprocess synchronization
- ❑ shared resource management
- ❑ memory management
- ❑ interrupt handling framework
- ❑ timing services
- ❑ RTOS initialization

Course Project

design, implement, demonstrate operation of a small Real-Time Executive:

- 12 OS primitives, 4 system processes

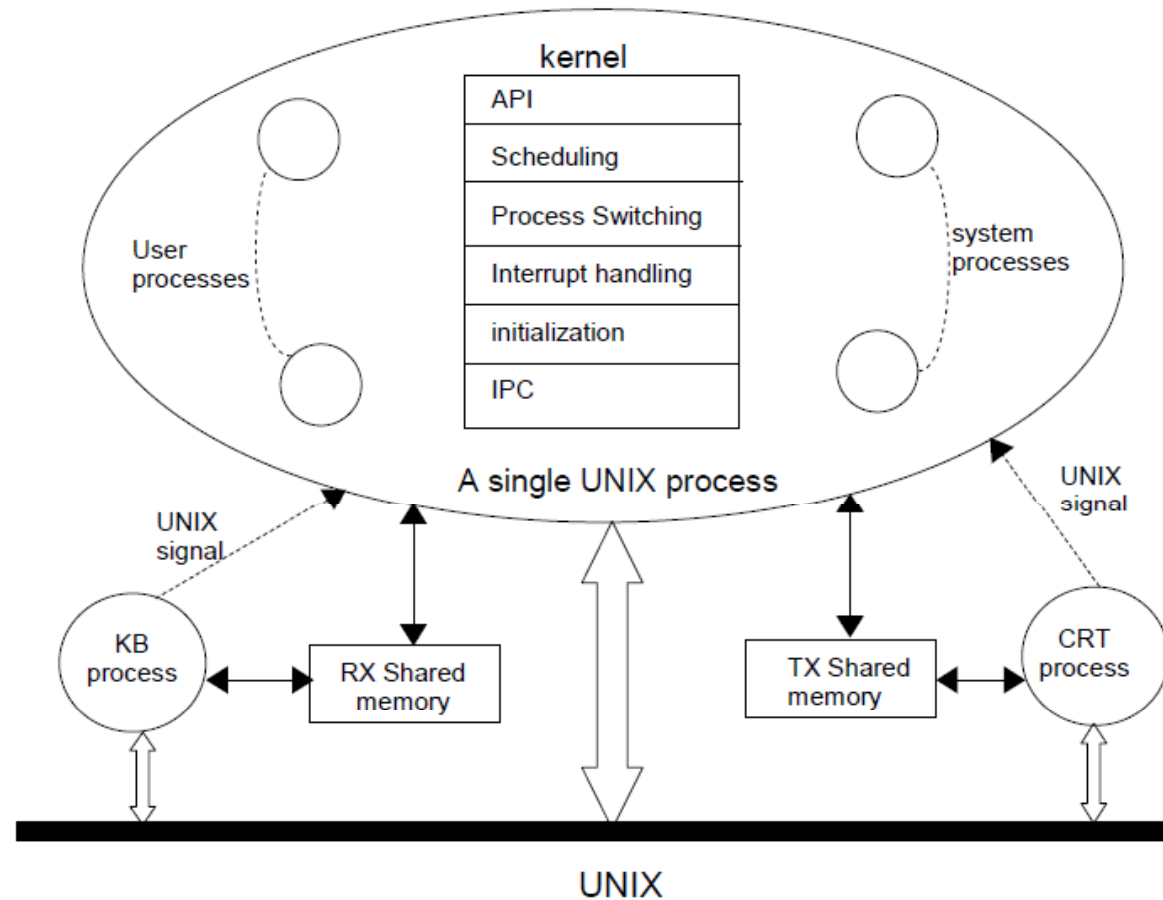
platform: Linux/Unix workstation

deliverables:

- SW Design Description
- partial implementation (C,C++)
- full implementation, demo of operation
- report (brief)

groups of 4: register by Sep 21 [ECE Course Book]

Project Overview



Project Deliverables and Weights

	weight	due
1. RTX Design Description	40%	see next slide
2. RTX Partial Implementation	10%	see next slide
2. RTX Full Implementation	40%	see next slide
3. Demo of Operation	incl abv	'at' demo
4. Final Report	10%	demo+2D

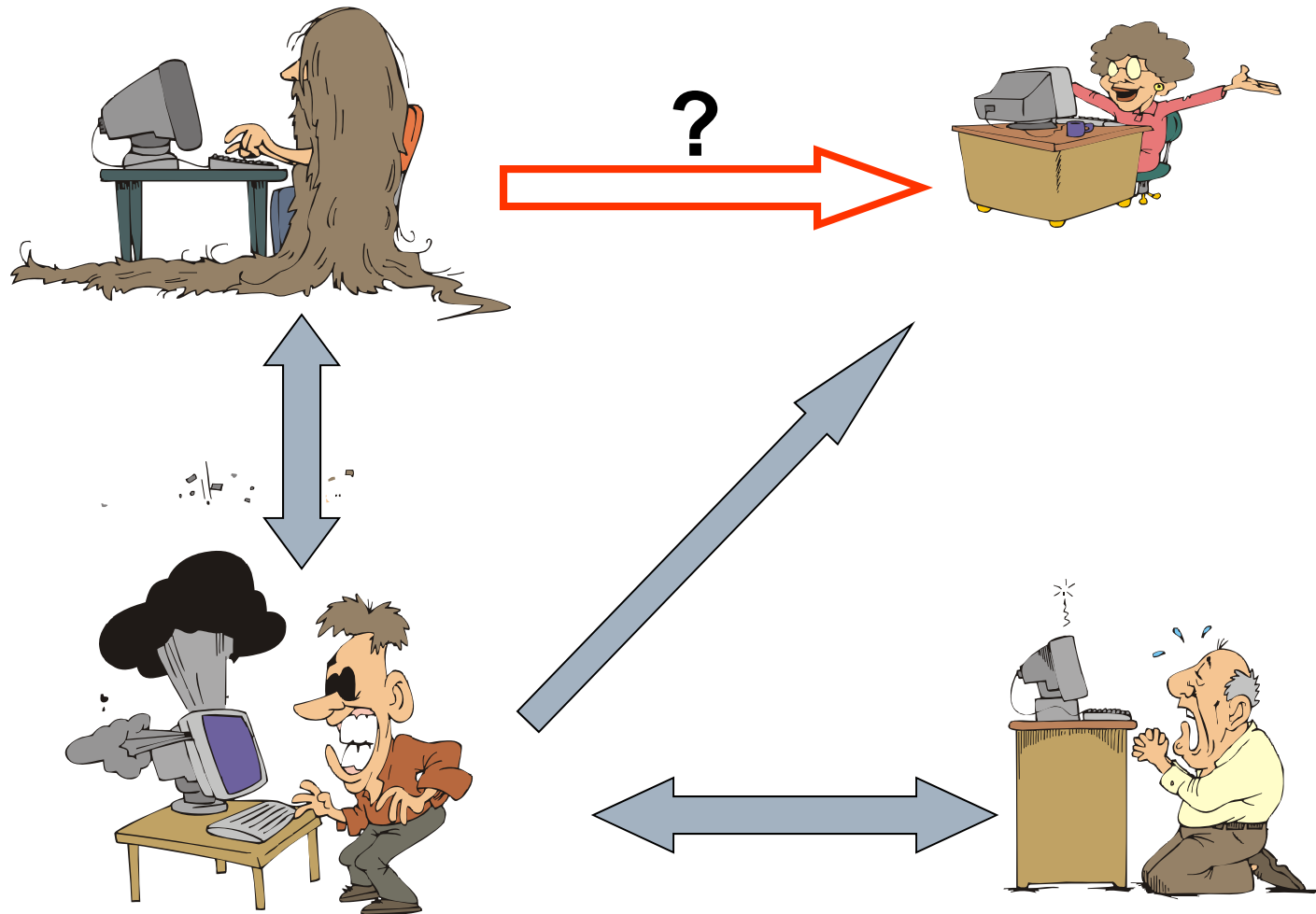
Project Deliverables: Dates

- ❑ The SDD deliverable
 - Upload to course ACE by 23:59, Thu Oct 20
 - Hand in hard copy in tutorial Fri Oct 21
- ❑ Partial RTX Implementation
 - Upload source to course ACE by 23:59, Nov 12
- ❑ Full RTX Implementation
 - Upload source to course ACE by 23:59, Nov 28
 - Demos begin Nov 30
- ❑ Final Report, upload within 48 hrs of demo
- ❑ See Project Description for details

Project Support

- ❑ software development
C++/C compiler [gcc, (gdb, ddd debuggers)]
- ❑ platform: ECE Linux network (ecelinux)
- ❑ implementation in C or C++
- ❑ no scheduled lab hours
- ❑ do most test & debug on your PC/Unix/Linux platform?

MTE 241 Project Stages



Marking Scheme

- Best of two alternatives:

Project	30%	30%
Midterm Exam *	20%	10%
Final Exam *	50%	60%
Total	100%	100%

***To pass course, your weighted exam average must be $\geq 50\%$**