EE445M Midterm Study Guide (Spring 2016):

Instructions:

- Open book and open notes.
- No calculators or any electronic devices (turn cell phones off).
- Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space (boxes) provided.
- *Anything outside the boxes will be ignored in grading.*
- For all questions, unless otherwise stated, find the most efficient (time, resources) solution.

Lecture notes Lectures 1 through 6
Book Chapters 1-4 (except Section 3.2 Memory Manager)

Week	Notes	Reading	Topics
1 **1/18**	Lecture 1	1.1-1.6, 2.3-2.9, 2.12	Introduction - ARM architecture, instruction set, stack, µVision4 compiler, GPIO, timer, UART, device drivers
2 **1/25**	Lecture 2	2.1, 3.4 3.1, 3.7	Software design - Modular programming, call & data flow graphs, flowcharts, I/O synchronization Debugging - Lab environment, intrusiveness, monitor, output to scope, simulator
3 **2/1**	Lecture 3	3.3, 4.1	RTOS - Multi-threading/-tasking, OS architecture OS kernel - Interrupt servicing, operating modes, context switching
4 **2/8**	Lecture 4	4.2, 3.5-3.6	Threads - TCB, cooperative & preemptive multitasking, round-robin scheduler Thread communication & synchronization - Critical sections, reentrance, FIFO, mailbox
5 **2/15**	Lecture 5	4.3-4.5, 4.7.2	Semaphores - Spinlock & blocking semaphores, monitors, deadlock Debugging - Testing, path expressions, performance measures (response time, jitter, throughput)
6 **2/22**	Lecture 6	4.1, 4.6	Scheduling - Real-time scheduling, priority scheduler, scheduling anomalies, fixed-rate scheduler, process networks

Lab Important Topics
Lab 1: Interrupts, Cortex M4 architecture, FIFO queues, UART, ADC
Lab 2: Real time OS, semaphores, critical sections, synchronization, communication
Lab 3: Debugging, priority, scheduling, blocking semaphores

Architecture (Chapters 1, 2, 3 are a review of EE445L) (except 3.2 malloc and free)

- Registers, buses, ports, stack,
- Interrupts NVIC, tail chain, late arriving,
  - SysTick, PendSV, edge triggered interrupts,
  - arm, enable, latency, jitter, hardware FIFO,
- ARM assembly code,
  - subroutine linkage (AAPCS),
  - parameter passing (registers and stack), l
  - local variables (registers and stack),
  - interrupt linkage.
- Data flow graph, call graph, flowcharts.
- HAL, device driver.

Debugging (Section 3.7)

- intrusiveness,
- profile,
- dump,
- control, observability,
- coverage
- white box, black box

Data structures

- FIFO,
- statically allocated linked lists,
- dynamically allocated linked lists,

OS stuff

- latency, real- time, interrupt priority,
- reentrancy, critical sections, race condition,
- sleeping,
- scheduling
  - preemptive/nonpreemptive=cooperative,
  - round robin,
  - priority,
  - rate monotonic,
  - earliest deadline first,
  - least slack-time first),
- semaphore implementation
  - spinlock,
  - cooperative spinlock,
  - blocking),
- semaphore applications
- OS Concepts
  - kernel,
  - bounded waiting,
  - priority inversion, priority inheritance, aging, starvation,

mutual exclusion,
certification,
hooks,
slack time,
lateness,
rate monotonic scheduler, rate monotonic theorem,
protection,
CPU utilization,
semaphore application (study the book examples),
deadlocks
- necessary conditions,
- detection,
- prevention,
- avoidance),

resource allocation graph, monitors.
See homework questions at the end of Chapter 4).

Real time OS, semaphores, critical sections, synchronization, communication
Spring 2001,Quiz2, Question 2, Sleep primitive
Fall 2001,Quiz2, Question 4, Priority scheduler, deadlock
Spring 2002, Quiz1, Question 3, Dynamic thread allocation, thread Kill
Fall 2002, Quiz2, Question 2, application of semaphores
Fall 2002, Final, Question 4, use of semaphores
Fall 2002, Final, Bonus questions 1,2,6, assembly language used in OS programming
Fall 2003, Quiz1, Question 2, use of semaphores
Fall 2003, Quiz1, Question 3, changing the TCB
Fall 2003, Quiz1, Question 4, definition of time jitter
Fall 2003, Quiz1, Question 5, implementation of OS_Wait
Fall 2003, Final, Question 14, definitions of OS concepts/terms
Fall 2004, Quiz2, Question 2, Three thread rendezvous
Fall 2004, Quiz2, Question 3, Binary semaphore
Fall 2004, Final, Question 9, Path expression
Fall 2005, Quiz2, Question 4, Reader/writer problem
Fall 2005, Quiz2, Question 5, Cooperative thread scheduler
Fall 2006, Quiz2, Question 9, Fork
Fall 2006, Quiz2, Question 5, Resource allocation graph
Fall 2006, Final, Question 5, Exponential Queue or multi-level feedback queue scheduling
Spring 2008, Quiz2, Question 4, use of semaphores
Spring 2008, Final, Question 2, Effect of OS on time-jitter while sampling an ADC
Spring 2008, Final, Question 5, Critical section, design new instruction
Spring 2009, Quiz 2, Question 4, Critical section
Spring 2009, Quiz 2, Question 5, Fork and join
Spring 2009, Final, Question 5, kill threads that finish executing
Spring 2010, Quiz 1, Question 2, word bank
Spring 2010, Quiz 1, Question 4, alternate words for signal and wait
Spring 2010, Quiz 1, Question 5, what happens if an ISR calls OS_Wait

Spring 2010, Quiz 1, Question 6, implementing mutual exclusion
Spring 2010, Quiz 1, Question 7, application of semaphores
Spring 2011, Quiz 1, Question 4, definitions
Spring 2011, Quiz 1, Question 5, application of semaphores
Spring 2011, Quiz 1, Question 6, new implementation of semaphores
Spring 2011, Quiz 1, Question 7, priority scheduler (the 2011 class did horrible on this question because they parroted their lab solution without reading the question)
Spring 2010 Final, Question 5, definitions d, i, j
Spring 2011 Final, Question 8, bounded waiting
Spring 2011 Final, Question 9, real time OS, minimizing latency
Spring 2011 Final, Question 11, FIFO with semaphores
Spring 2011 Final, Question 12, implementing semaphores in a Dual core processor
Spring 2011 Final, Question 16, implementing a thread scheduler one a 16-core processor
Spring 2012 Quiz 1, Question 4, Two SPs.
Spring 2012 Quiz 1, Question 5, OS definitions.
Spring 2012 Quiz 1, Question 7, Monitor and deadlocks.
Spring 2012 Quiz 1, Question 8, OS_AddThread and OS_Kill.
Spring 2012 Quiz 1, Question 9, Use OS to debounce a switch.
Spring 2013 Quiz 1, Question 1, Priority.
Spring 2013 Quiz 1, Question 3, OS definitions.
Spring 2013 Quiz 1, Question 5, using semaphores.
Spring 2013 Quiz 1, Question 6, Assembly language thread switch.

General questions
Fall 2004, Quiz2, Question 4, Time-jitter
Fall 2004, Quiz2, Question 5, Definitions and a word bank
Fall 2005, Quiz2, Question 6, Time-jitter
Fall 2006, Final, Question 4, Critical section
Spring 2009, Quiz 2, Question 3, FIFO implementation
Spring 2011, Quiz 1, Question 1, time jitter
Spring 2011, Quiz 1, Question 2, reentrant, parameter passing, LR
Spring 2011, Quiz 1, Question 3, bit-banded I/O eliminates critical section, which registers are pushed on the stack during an interrupt context switch, what is LR during an ISR
Spring 2010 Final, Question 1, Cortex M3 interrupt context switch (answer for LM3S8962)
Spring 2011 Final, Question 2, Cortex M3 interrupt context switch
Spring 2012 Quiz 1, Question 3, Harvard architecture.
Spring 2012 Quiz 1, Question 6, Reentrancy.
Spring 2013 Quiz 1, Question 2, Control and observability.
Spring 2013 Quiz 1, Question 4, Critical section