


Week 13 - Friday

Course Review


School of Information Technology and Electrical Engineering
The University of Queensland



Outline

- Today
 - Course Review
 - Exam discussion
- Remember:
 - Assignment 4
 - Due 11pm Friday June 5 (+ any available grace days)
 - No submissions after 11pm Wednesday June 10


2



Review of Course Goals

- Course Aim (from course profile)
 - *"The main goal of this course is to produce students who are competent C systems programmers who have a good understanding of how the underlying operating systems and networks work."*
- 16 learning objectives (from course profile)
 - Some weren't covered in great depth
 - The following slides review the goals and your expected level of knowledge/understanding for the final exam


3



UNIX

1. **Be able to use a UNIX shell, common UNIX shell and utility commands. You should be familiar with UNIX file system protections and other UNIX concepts**
 - This is required knowledge for exam
 - Common utilities: cp, mv, ls, ln, chmod, ...
 - Pipes and redirection: |, >, <
 - Files: links, modes (e.g. 755), l-nodes, ...
2. **Be able to write simple UNIX shell scripts**
 - Not required for the exam


4



C Programs

3. **Be able to write, compile, run and debug C programs in a UNIX environment, including "systems-level" programming**
 - This is the major goal of this course
 - Much of the exam involves C in some manner
 - You will need to be able to write and understand C programs, functions, declarations etc. for the final exam
 - You should be familiar with library functions and system calls talked about in class and used in examples and exercises (no man pages will be supplied in the exam)
 - No compiling/debugging knowledge required for exam (though you should know what a segmentation fault is and what causes it)

5



Operating Systems Basics

4. **Understand the basic purpose and requirements of operating systems and operating system terminology**
 - There are no "definition questions" on the exam (since it's open book) but we will assume you are familiar with all the terminology used in the course

6

COMP2303
COMP7306

File Systems

5. Understand the principles behind file systems and file input/output, including buffering
6. Be able to write programs which access and navigate around file systems and perform file input and output

- You need to understand the principles behind UNIX file systems (but not other types) and how they are structured and used

7

COMP2303
COMP7306

Virtual Memory

7. Understand the principles behind virtual memory

- Paging, page tables, address translation, memory protection etc

8

COMP2303
COMP7306

Dynamic Memory Allocation

8. Understand how dynamic memory allocation works

- Be able to write programs that use malloc(), free() etc.
- [Major change from previous years where students had to write a memory allocator. Past exam questions on this topic are not relevant.]

9

COMP2303
COMP7306

Processes and Threads

9. Understand the concepts of processes and threads and how operating systems manage these
10. Be able to write C programs which create and control processes and threads

- Processes: fork(), exec(), exit(), zombies, reaping, wait(), waitpid(), signals, context switching, etc.
- Threads: programming, mutexes, condition variables, thread safety, ...

10

COMP2303
COMP7306

Interprocess Communication

11. Understand the various methods of inter-process communication and when to choose each particular method

- Signals, pipes, file descriptors, I/O redirection, etc
 - Examinable
- Shared memory, semaphores, critical sections
 - Need to know what they are, but no code questions on these

11

COMP2303
COMP7306

Computer Networks

12. Understand computer network terminology and the possible topologies, functions and architectures of computer networks
13. Understand aspects of the IP, UDP and TCP protocols including IP addresses, port numbers, header fields and when UDP or TCP would be chosen.

- Need to know the basic principles and terminology, but no "definition questions"
- Ethernet level details are examinable
- TCP/UDP/IP details examinable
 - We didn't cover all the header fields in lectures, but those we did are examinable

12

COMP2303
COMP7306

Sockets

14. Understand the sockets API for network programming and be able to write sockets based programs

- Should know sockets interface for TCP and UDP, servers and clients, various styles of servers (multi-process, multi-threaded, using select() etc)
- If asked to write a network program, error handling can be ignored – assume system calls are successful

13

COMP2303
COMP7306

Higher Level Protocols

15. Understand various higher level protocols such as SMTP and HTTP

- All aspects of HTTP covered in lectures are examinable
- Protocol details of DNS, SMTP etc are not examinable but you should understand how/why they use TCP, UDP etc.

14

COMP2303
COMP7306

Other

16. Understand and use the Subversion Revision Control System

- Examinable – you should understand how to do common operations on a repository

15

COMP2303
COMP7306

Exam

- 2 hours + 10 minutes perusal
- 100 marks = 8 questions (10 to 16 marks each)
- Questions involve
 - Writing C (declarations, functions, programs, etc.)
 - Understanding C (expressions, declarations, programs, etc.)
 - Drawing pictures
 - Answering short answer questions
 - Some calculations
 - Calculators permitted
- Answer all questions on the exam question paper

16

COMP2303
COMP7306

Open Book Exam

- Any written/printed material permitted
- Exam is designed to test understanding and doing, not remembering
 - This does not mean you have to know nothing!
 - The less you have to refer to papers & books, the better
 - You should be able to write most C statements, expressions, functions etc without looking at notes or reference material
- **Index** what you do bring in for quick access
- You are advised to bring
 - C Operator precedence table
 - Printouts of code covered in lectures & assignments
 - Any other notes, books etc

17

COMP2303
COMP7306

Preparing

- Recent exam papers available from library
 - Most questions are relevant
 - Ignore those about writing malloc(), free() etc.
 - No official solutions will be provided
 - Remember you can usually check C related answers by compiling/running code
- Ask (and answer) questions via newsgroup
- Review mid-semester exam paper (and sample questions)
 - All questions are relevant
- Write lots of C code, review lecture code and assignment "solutions"

18

COMP2303
COMP7306

Consultation Times

- Times to be advised
- Student input desired... when do you want the sessions to be?

19

COMP2303
COMP7306

Any Questions?

20