Subject Summary

What You Would have learned if you didn't skip classes

(True of only a small minority)

2002–2003

Nick Urbanik <nicku(at)nicku.org>

A computing department

Main Topics

- Shell Programming and POSIX commands
- Free Software, Open Standards
- Operating System Structure, Open Standards
- Processes and Threads
- Race Conditions, Locking and Deadlock
- Secure Shell
- Memory Management
- Input and Output
- Systems Integration

What did we Cover from Workshop Notes?

- A burning question from *some* people in group W, and some *specific* people from other groups
- Answer is on the web site, reproduced here:
  - Module 1, Overview
  - Module 2, Basic Shell
  - Module 3, Basic Tools
  - Module 4, More Tools
  - Module 5, Basic Filesystem
  - Module 6, Finding Documentation
  - Module 7, Administering User Accounts and Permissions
  - Module 13, SSH — The Secure Shell

Shell Programming, POSIX commands

- The first seven chapters of Workshop Notes introduced POSIX commands
- The lectures on shell programming used these commands with the shell programming language
- You studied file permissions, including SUID, SGID executables and SGID directories
- Study the lab exercises on shell programming
  - There are solutions on the web site—examine them *closely*
- One exam question relates to these topics
- No need to memorise commands: *appendices to exam* contain lots of information you can refer to, including an excerpt from lecture notes on the `sed` command.

Operating System Structure, Open Standards

- We studied operating system structures:
  - Monolithic kernel (Linux)
  - Microkernel (Mach, Hurd, Windows NT, 2000, XP)
  - Virtual Machine (Mainframes, Java VM, VMware)
  - Layered Architecture (Windows)
- There was one lecture on Free Software and Open Standards
- We studied systems integration in two lectures
- Make sure you understand the lecture on the operating system kernel.
- Most of an exam question relates to these topics

Processes and Threads

- In this long lecture, we covered many topics, including:
  - Comparing processes and threads
    - Make sure you understand the practical effect of the differences between threads and processes
  - Process states
  - POSIX process creation — `fork()`, `exec*()`, `wait()` and `exit()`
    - . . . and using these to create a *simple interactive shell*
- One exam question relates to these topics

Locking, Race Conditions, Deadlock

- The material came from the end of the lecture slides on Processes and Threads, and a separate lecture on Deadlock
  - They really belong together
- We covered *locking* mainly in relation to POSIX threads
- We did a lab exercise on Deadlock
- Understand the *purpose* of locking (understand, don't just memorise the notes)
- One exam question relates to these topics

Input and Output

- This lecture focused on a number of things, including designing and partitioning hard disk systems with redundancy and flexibility.
  - There is a case study involving RAID and a volume manager
- Half an exam question relates to these topics
- For those who where out to lunch,
  - I skipped the section of the notes on installing device drivers

Secure Shell

- We studied the lecture from Module 13 of the Workshop Notes
- We did a workshop on the topic in the laboratory
  - The main issues relate to the proper handling of keys
  - Avoiding security risks
- Half an exam question relates to this topic

Memory Management

- We studied this topic in the lecture theatre
- We did a tutorial exercise on memory management[a]
- One exam question relates to these topics

[a]Except for Group W, who were "out to lunch."

Format of the Exam (2002–2003)

- Has *six* questions
- Select *five* of them
- *All of equal value*, 20%

Advice for the Exam

- *Budget your time* wisely in the exam:
  - Spend a few minutes to *decide which question* you will not attempt
  - Divide remaining time by five
  - Do *not* spend more than this time until you have answered five questions fully
- Show your working
  - A wrong answer with no working gets *zero* marks
  - A wrong answer with some working that is on the right track gets *some* marks
- If you are attempting the supplementary, study the solutions to the main exam on web site :-)
  - but this is not enough to pass :-(

Compared with past papers

- This year's exam is different from past papers
- Teaching focusses on use of *C* and *system calls* much more than previously
  - An appendix to the exam includes function prototypes for some system calls and library functions
- Revising using previous exam papers:
  - I will attempt to provide solutions to previous exams
  - *Definitely not sufficient* for revision of whole course, however.

Watch the Subject Web Site

- Watch the web site for announcements:
- I will write and post *solutions* to problems as soon as I can.
- I made a *new icon* NEW SOLUTIONS to highlight changes to solutions on the site,
  - including solutions to problems I have written.