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
- Operating System Structure
- Booting an Operating System
- Processes and Threads and Inter Process Communication (IPC)
- 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
- You have done an assignment using this information, integrating what we covered
- One exam question relates to these topics
- No need to memorise commands: *appendices to exam* contain lots of information you can refer to

Operating System Structure, Booting

- We studied operating system stuctures:
  - Monolithic kernel (Linux)
  - Microkernel (Mach, Hurd, Windows NT, 2000, XP)
  - Virtual Machine (Mainframes, Java VM, VMware)
  - Layered Architecture (Windows)
- We studied bootloaders at length
- Most of an exam question relates to these topics

Processes and Threads, IPC

- In this long lecture, we covered many topics, including:
  - Comparing processes and threads
  - Process states
  - POSIX process creation — `fork()`, `exec*()`, `wait()` and `exit()`
    - . . . and using these to create a simple interactive shell
  - We covered the basics of Inter Process Communication (IPC) in lectures,
    - . . . and in more detail in lab
    - Don't miss that lab!
- 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
  - I will move them into the same file when I rewrite the lecture in LaTeX instead of MS PowerPoint
- We covered *locking* mainly in relation to POSIX threads
- We did a lab exercise on Deadlock
- One exam question relates to these topics

Input and Output

- This lecture mainly focussed on two main topics:
  - DMA and buffering
    - Single buffering, and
    - Double buffering — when it is necessary
  - 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.”

Systems Integration

- We studied systems integration in two lectures
- Involves getting systems from many manufacturers to work together nicely, such as Windows, Unix, Linux and Macintosh
- We mentioned LDAP
- We studied Samba in a workshop session, where we created a primary domain controller using Samba, adding Windows 2000 Professional to the domain
- Part of one exam question relates to these topics

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

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
  - 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 will make a *new icon* to highlight changes on the site,
  - including solutions to problems as I write them.