Draft commentary on the syllabus for *Operating Systems and System Integration*

This commentary is preliminary. It is an evolving document that will be developed further as I build the teaching package for this subject. It is mainly intended as a tool for designing the teaching plan and laboratory work. As I build the teaching package, I will expand some areas and reduce others. I may have missed the intention of the original syllabus design in some areas; if you notice this, please tell me. In other areas I have deliberately introduced a practical interpretation.

Nick Urbanik <nicku@vtc.edu.hk>

Module Title:	Operating Systems and System Integration	
Module Code:	CMT3321	
Module value:	1.5	
Duration:	30 weeks	
Class Contact Hours:	Lecture	30 hours
	Laboratory	30 hours
	Tutorial	30 hours

Tutorial uses group discussion, short talk, practice, short talk, practice

Assessment Scheme:	Continuous Assessment:	50%
	Examination:	50%

Module Rationale/Aims

- to provide more insight into the resource management of common operating systems;

  This is from a practical and comparative point of view. For example with memory management: not examining many of the internal details of memory management, but focusing on the practical consequences: "Do I have enough memory?" "Am I in danger of running out of swap space?" "Which processes are using so much memory?"
- to introduce the major technical issues associated with the design and use of operating systems;

  We are not teaching students how to design an operating system. Rather, what criteria to use when selecting a suitable operating system for a particular application.
- to introduce the concepts of system integration;

  System integration means integrating the hardware and software of an individual machine to provide a functional service, and also to integrate a system of computers into a useful whole. This second point involves making various computer systems interoperate both within one computer and also over a network.

- to introduce the hardware and software tools used to integrate various operating systems in a heterogeneous environment;

  One of the main tools is Samba. Others are tools that run on Windows machines to help them integrate with other systems.
- to give students practical experience of systems integration activities.

  That's a significant part of the laboratory exercises, which will involve getting Windows, Linux/UNIX and Novell machines working peacefully and productively with each other.

Learning Objectives

Students will be able to:

- identify and describe the *resource management techniques* used in modern operating systems;
- evaluate and select the *appropriate operating system* for some specific applications;
- demonstrate the use of techniques in *system integration*;
- select and use the *appropriate software* and *hardware tools* for system integration;
- design a usable and *effective operating environment* for both stand-alone and *networked computers*.

Syllabus Keywords

computer components, memory, configuration, performance, I/O devices, secondary storage, PC systems, operating systems, process management, multitasking, scheduling, memory management, virtual memory, paging, I/O management, buffering, spooling, file management, systems integration, cross platform, remote system, performance, network, user authentication, file and printer sharing.

Textbooks/References

Stallings, W., *Computer Organisation and Architecture*, 4th edition, Prentice Hall, 1996.
Cook, B and White, N., *Computer Peripherals*, Edward Arnold, 1995.
Williams, G. R., *Windows NT & UNIX: Administration, Coexistence, Integration and Migration*, Addison Wesley, 1998.

Important is *Using Samba*, a book that is freely available over the Internet (can also be purchased as a hard copy).

The book *Operating System Concepts* by Silbershatz and Galvin is a useful reference, but is a university level book, and is less practical than this subject. Similarly with William Stallings, *Operating Systems: Internals and Design Principles*.

Key Content Area

This subject could easily be very theoretical, and mainly involve a detailed study of the internals of operating systems.

While it is helpful to understand some aspects of the internal operation of an OS, this subject will be *practical* in nature. It will only provide sufficient theoretical background for the student to understand what they are doing in practical system management and configuration. The laboratory exercises will centre on practical application of techniques that will be directly useful in employment.

While Linux will be used to illustrate most aspects of the operating system, I will make a great effort to ensure that this is a comparative study. For each technique that the students investigate in Linux, they will investigate the corresponding technique in Windows.

The laboratory work will use standard tools wherever possible, rather than expensive or unusual products that the employer is unlikely to provide.

Here: AAA means "Ask Albert and Alex to show me how it works in Windows".

	lecture	tut/lab
1. Operating system structures	2	2

This is a short overview of the subject. It introduces the students to what they will be doing here, and covers the basics of what an operating system is, what types there are, and a brief comparison of what's available now.

(a) Monolithic Systems

(b) Layered Systems

(c) Virtual Machines

Tutorial: include a quick demo of VMware.

(d) Characteristics of modern operating systems

2. Multiprogramming operating systems	8	13

(a) Process management

This involves:

- What is a process
- Concept of ownership of a process
- Kernel and user processes
- Determining the status of a process (in Linux, using `ps`, `top` and `gtop`; in Windows, AAA (asking Albert and Alex to show me))
- changing the status of the process by killing it or changing its priority (In Linux with `kill`, `killall`, `nice`, `snice`, `renice`; in Windows: AAA)
- Starting and stopping services.

(b) Memory management

- What is virtual memory
- What is memory management
- determining memory usage (both overall and by individual processes) with `top`, `gtop` and `free`; in Windows, AAA
- determining swap space usage and appropriate swap size with `top`, `gtop` and `free`; in Windows, AAA

(c) Input/Output management

- What is I/O

- What is Input/Output management (network, disk, serial, USB, video, IDE, SCSI,...)
- device drivers and how to install and configure them (practical sessions with Linux and Win98/NT/2000)
- interrupts and the kernel (configuration and monitoring)
- monitoring I/O performance (with Linux and Windows)
- realtime and non-realtime requirements (briefly mention rtLinux)

(d) File management

- What is a file system? Examples: FAT, ext2, NTFS,...
- partitioning and partitioning systems (practical experience with partitioning hard disks: primary, extended and logical partitions; BIOS limitations and how to work around them)
- Mounting a file system
- file hierarchy and organisation
- file permissions and ownership: changing permissions and ownership
- Brief introduction to the security implications of file permissions and ownership
- Emergency repair of partition table and filesystems with recovery disks such as `tomsrtbt`.
- [*Only if time allows*]: file undeletion in ext2 and NTFS (AAA).

3. Computer System Components 5 10

This is an opportunity for students to get some practical experience with installing and configuring computer hardware. This will also include a practical troubleshooting laboratory session.

(a) General Issues to be addressed for each component

One general issue to be solved is device conflict, how to recognise and solve it.

(b) Specific components and issues:

In how much detail did the previous subjects cover this hardware? If very little, will cover the basics of how each of these work, as well as the configuration options available for them.

- Video displays
- Sound and voice
- Hard copy input and output
- Secondary storage

4. Systems Integration 8 20

(a) Planning and implementation issues

Appropriateness of operating systems for various purposes. Issues of costing and maintenance.
How to keep the client going: Symantic's *Ghost* or Innovative Software Ltd's *ImageCast IC3*, and Novell's *ZENworks*, or network computers (NCs) and thin clients.

(b) Cross platform technologies for software application use

Introduce Samba, VMware, `dosemu`. Samba is incredibly important and needs to be covered in practical detail.
Windows: AAA.

(c) Remote system management tools

Secure shell (openssh) is *the* tool for remote system administration for every platform except (?) Windows: AAA.
Lab exercises in how to install, configure with public and private keys, interoperation with Windows (`putty`, `ttssh`), automatic X11 forwarding.
In lab: include backup over network using `ssh`.
VNC for remote administration of NT. Security implications of using VNC.
Using VNC with `ssh`.

(d) Performance issues

Best covered as discuss each topic, rather than treated at length in isolation.

5. Network Integration 7 15

(a) Configuring and using network applications

This is about Samba; most other useful network applications are file and print sharing, or Internet applications.

(b) User Authentication

Authentication methods: PAM (pluggable authentication modules, used in Linux and now Solaris), Win98, system used in NT, 2000 (AAA), basic authentication (web servers)
Briefly mention directory systems, such as LDAP, NDS, and provide some laboratory practice with LDAP.

(c) File and printer sharing

with Samba, lpd, perhaps lprng, SMB, NFS, Novell file and printer sharing.

(d) Internet application integration

Apache: setting up and configuring on both Linux and NT/2000
introduction to cross platform Internet application development environments such as Zope, PHP (and possibly Midguard), Perl (and `mod_perl`), and CGI.
XML, DSSL, SOAP.
Setting up an ftp server that allows remote execution of selected commands (such as `site exec locate`), automatic on-the-fly compression of files and directories. Security implications. Integrating it with web server.

total: 30 60

Topic not mentioned but that I would like to cover

I think that *installation* of operating systems is an important practical topic. Students will get experience with installing Linux and NT, and choosing disk partitioning strategies.

Prerequisites/Corequisites

None/none

All students used a UNIX environment for their programming module in year 1; according to some sources, the students enjoyed this subject. This means that the students are at least familiar with the basics of using a UNIX system, such as Linux.

Exemption Criteria:

None

Teaching & Learning Strategies

General strategies on teaching and learning should be referred to Section I.