

# Module Guide

## *Undergraduate Programme Academic Year 2004/2005*

<b>Module:</b>	Programming For Communications and Networks V	
<b>Web-site:</b>	<a href="http://copsewood.net/tic/pcn5">http://copsewood.net/tic/pcn5</a>	
<b>Division:</b>	Electronics, Communications and Software	
<b>Module Co-ordinator:</b>	Richard Kay	
<b>Lecturers:</b>	Richard Kay (RK), Michael Boyd (MB)	
<b>Laboratory Tutors:</b>	Richard Kay, Michael Boyd	
<b>Appointments:</b>	G0 reception: RK 13:15 - 15:15 Thurs. MB see TIC intranet	
<b>Brief Descriptions of the TWO Items of Assessment:</b>  <b>You will be expected to complete TWO Assessments.</b>	<ol style="list-style-type: none"> <li>1. An in class test including a number of short questions will be held during week 8.</li> <li>2. A series of coursework excercises will be demonstrated during tutorial sessions and written up.</li> </ol>	
<b>Assessment Weighting:</b>	Assessment 1: 50%, Assessment 2: 50%	
<p><b>Individual assignments. The work you submit shall be your own and not the product of collaboration with anyone else. Plagiarism will be penalised.</b></p> <p><b>In-course assessments shall be submitted through the Coursework Collection System, to the module co-ordinator.</b></p>		
<b>Contents of Guide:</b>	see below	

## Syllabus and supporting information

### **PURPOSE:**

The module aims to provide an understanding of inter-process communication, and the principles and techniques used in writing programs that need to communicate with each other.

### **LEARNING OUTCOMES:**

On completing the module the student should be able to develop programs that communicate with other programs using the operating system's facilities for inter-process communication and synchronization. Outline topics will be covered by lectures and practical laboratory-sessions, augmented by coursework assignments, tutor-guided study and self-study.

### **INDICATIVE CONTENT:**

**Operating systems overview: Introduction - historical perspective: evolution of operating systems; structure and design of modern operating systems; concepts of multiprogramming and multiprocessor systems; distributed systems. Basics of computer systems: architecture, interrupt servicing, processor execution modes**

**Operating systems basics: System components; processes; memory; I/O system; file-system; user-interface; program development environment; networking.**

**Process management: Comparative scheduling policies. inter-process communication. resource deadlock and techniques for its avoidance.**

**System calls: High and low level input and output, inter-process communication; system resources, file-management, process management.**

**Interprocess communication: Signals in UNIX. Process communication and synchronisation: concurrency, co-operating processes, concurrency control, critical-sections and solutions. Hardware support. Semaphores: integer and counting (busy waiting). Use for process synchronisation. Classic synchronisation problems: bounded-buffer, readers/writers, dining philosophers. Monitors. Distributed co-ordination: event-ordering, mutual exclusion - central and fully-distributed. Pipes, FIFOs, shared memory, semaphores, sockets. Deadlocks: prevention, avoidance, detection.**

### **READING LIST**

Peek, Todino-Gonguet and Strang (2001) *Learning the Unix Operating System, 5e*, O'Reilly

Newham and Rosenblatt. (1998) *Learning the Bash Shell 2e*, O'Reilly

Kernighan and Ritchie (1989) *The C Programming Language*,

Prentice-Hall. Rochkind, Marc J. (1985) *Advanced UNIX Programming*, Prentice-Hall.

Haviland, Keith and Salama, Ben (1987) *UNIX System Programming*, Addison-Wesley.

Silbershatz, Abraham, Galvin, Peter and Gagne, G. (2000) *Applied Operating System Concepts*, John Wiley & Sons .

---

## Teaching Schedule for module: **Prog. Coms & Nets V**

Module Co-ordinator: R Kay

Wk No	Date W/C-Mon	Lecturer	Lecture Topic	Tutorial Topic	Assignment	
					Set	Due In
1	27 Sep04	RK	Introduction to Linux and Unix	Logging into Linux, editing files		
2	4 Oct 04	RK	Pipes and I/O plumbing	Simple shell commands, redirection.	2	
3	11 Oct 04	RK	Scripting on Linux	Simple Bash shell scripts		
4	18 Oct 04	RK	'C' programming on Linux	Interfacing 'C' programs and Shell Scripts		
5	25 Oct 04	RK	Introduction to process management	Use of ps, kill, fork(), the exec() family.		
6	1 Nov 04	RK	Networking utilities	Ping, dig, telnet, ssh, ftp, sendmail		
7	8 Nov 03		Assignment completion			
8	15 Nov04	test	In-class test	Coursework briefing/demonstrations	1	1
9	22 Nov04	MB	Signals in Unix, process communication and synchronisation.	Process Control in UNIX 1 (using fork(), exec() and wait())		
10	29 Nov04	MB	Semaphores: integer and counting (busy waiting).	Process synchronisation. Handling signals.		
11	6 Dec 04	MB	Interprocess communication: pipes, FIFOs, shared memory, semaphores	Interprocess communication: pipes; FIFOs		
12	13 Dec04	MB	Monitors. Distributed co-ordination: event-ordering, mutual exclusion	Interprocess communication: semaphores, messages		
			Vacation (3 weeks)			
13	10 Jan 05	MB	Deadlocks: prevention, avoidance, detection	System information and services. Shell commands for IPC		
14	17 Jan 05	MB	Review	Record locking: deadlocks. Interprocess comms: shared memory		
15	24 Jan 05		Examination week			
16	31 Jan 05		Examination week			2

## Assignment Schedule for: **Module Name**

	Assignment Title	Date Issued (week commencing)	Date Due In	Weighting
1	In class test	15/11/04	15/11/04	50.00%
2	Coursework	04/10/04	03/02/04	50.00%