Module 6: Operating System Design

Stage	1
Semester	2
Module Title	Operating System Design
Module Number/Reference	6
Module Status (Mandatory/Elective)	Mandatory
Module ECTS credit	5
Module NFQ level (only if applicable)	8
Pre-requisite Module Titles	None
Co-requisite Module Titles	None
Is this a capstone module? (Yes or No)	No
List of Module Teaching Personnel	Dr Faheem Bukhatwa Mr Kevin McNally

Contact Hours				**Non-contact Hours**			**Total Effort (Hours)**
Lecture	**Practical**	**Tutorial**	**Seminar**	**Assignment**	**Placement**	**Independent work**	
24		12		32		32	100

Allocation of Marks (Within the Module)

	Continuous Assessment	**Project**	**Practical**	**Final Examination**	**Total**
Percentage contribution	40%			60%	**100%**

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. discuss the nature, origin and function of operating systems at a generic level and analyse issues of resource management, especially of processes and memory

2. analyse the role of operating systems in non-traditional environments such as embedded systems, concurrent systems and real-time systems
3. install various operating systems; evaluate and compare the interface, functionality and suitability of different operating systems
4. demonstrate an ability to manipulate an operating system using systems programming
5. explain how the operating system relates to the underlying hardware and the software that runs on it

Module Objectives

This module focuses on the understanding of the concepts involved in designing an operating system; and understanding of the complexity and the many requirements of an operating system. This module introduces the learner to some fundamental algorithms used in operating systems. It introduces the concept of concurrency in an OS; explores the concept of multiprocessing and distributed operating systems; and it provides a practical knowledge of operating systems.

Module Curriculum

Introduction to operating systems

- Introduction: What is an Operating System;
- Types of OS: Batch, Time-Sharing, Personal Computer Systems, Parallel Systems, Distributed Systems; Operating System Structures: System Components, Operating System services, System calls, system programs, system structure.
- System Design and Implementation;
- Introduction to DOS, Linux and another commercial desktop operating system; emerging and experimental operating systems to be used for practical work.

Processor management

- Definition of a process; process control block and process image; process states and life cycle.
- scheduling algorithms, thread scheduling,
- I/O bound jobs and CPU bound jobs
- Turnaround time, waiting time, execution time.
- CPU Scheduling: basic concepts and scheduling criteria,
- Scheduling algorithms: FCFS, Shortest job first, Priority, Shortest remaining
- priority scheduling works.
- Round robin, Spooling, Quantum

Process management

- Threads - overview and benefits; user and kernel threads.
- Processes: Process concept, operations on processes, cooperating
- processes, inter-process communication;

- Deadlocks: System model, deadlock characterisation, detection, methods for handling deadlocks, recovery from deadlock, deadlock prevention and avoidance.
- Synchronisation: critical section problem, Two task solution,
- synchronisation in hardware, Semaphores, classical
- synchronisation problems, Dining philosopher problem.

Memory management

- Memory management: fixed, dynamic and reloadable partitions memory management systems. contiguous memory allocation, paging and segmentation;
- Virtual memory: demand paging, page swapping, allocation of frames, thrashing.
- Internal and external fragmentation in different systems.
- Contiguous Memory Allocation
- Page strategies: FIFO, Most recently used, least recently used. Optimal page replacement
- Security aspect of memory management

File management

- File Systems: File concept, access methods, directory structure, protection, file system structure, allocation methods, free space management, directory implementation, efficiency and performance.
- Fixed and variable record files, contiguous files.
- Sequential, direct and indexed access files.

Device management

- Dedicated and shared devices. Sequential and direct storage media devices. Inter record gap, Inter-block gap, Plates, cylinders, tracks and sectors.
- Fragmentation, Addressing on a disk.
- Calculating capacity and speed.
- Blocking and buffering.
- Seek, search and transfer times.
- Seek policies: seek strategies, including FCFS, SSTF, SCAN/LOOK, C-SCAN/C-LOOK, and how they compare

Protection and security

- Protection: Goals of protection, domain of protection, access
- matrix, implementation of access matrix;
- Security: the security problem, authentication, program threads, system threats, threat monitoring, encryption, computer security classifications.

Specialised operating systems

- Distributed Operating Systems;
- Embedded Operating Systems;
- Real-time Operating Systems.

Reading lists and other learning materials

Silberschatz, A., Galvin, P. B. & Gagne, G. *Operating System Concepts with Java (7th Edition)*, Wiley, 2006

Tanenbaum, A. S. *Modern Operating Systems (3rd Edition),* Prentice-Hall, 2007

Secondary reading

Holcombe, C. & Holcombe, J. *Survey of Operating Systems*, Career Education, 2005

Sobell, M. G. *A Practical Guide to Linux(R) Commands, Editors and Shell Programming*, Prentice-Hall, 2005
Stallings, W. Operating Systems: Internals and Design Principles (6th Edition),

Module Learning Environment

Accommodation

Lectures are carried out in class rooms / lecture halls in the College. Tutorials are partially carried out in class and partial are at labs throughout the Campus.

Library

All learners have access to an extensive range of physical and electronic (remotely accessible) library resources. The library monitors and updates its resources on an on-going basis, in line with the College's Library Acquisition Policy. Lecturers update reading lists for this course on an annual basis as is the norm with all courses run by Griffith College.

Module Teaching and Learning Strategy

The module is delivered through a combination of lectures, tutorials and practical lab programming sessions. The learners complete a series of worksheets throughout the module which build on the learning in lectures.

Module Assessment Strategy

The module assessment consists of 1) continuous work and 2) a final examination.

Element No	Weighting	Type	Description	Learning Outcome assessed
1	10%	Weekly Tutorial Submission	Work material aims at enhancing the understanding of material covered.	1,2,3,5
	10%	Assignment	Programming or report writing on a particular topic of operating systems	1,2,4
	20%	Mid-term test	Evaluation of the learner attainment of material covered	1,2,3,5
2	60%	Closed Book Examination	End of module Examination	1, 2, 3, 5