MODULE SPECIFICATION

Part 1: Information			
Module Title	Operating Systems		
Module Code	UFCFWK-15-2	Level	Level 5
For implementation from	2019-20		
UWE Credit Rating	15	ECTS Credit Rating	7.5
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Module type:	Standard		
Pre-requisites	Computer and Network Systems 2019-20		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description

Educational Aims: See Learning Outcomes.

Outline Syllabus: The syllabus includes:

Operating System Organization models and structures

History and implications of using Open Source code in operating systems. Licensing issues and their legal implications.

Process and Object Management kernel services, interrupt handlers, scheduling. Inter-process Communication event handling, message passing, synchronous/asynchronous, shared memory.

Concurrency and Synchronization semaphores, critical regions, monitors, message passing, multi-threaded processes.

Memory Management Organization algorithms and policies, Virtual Memory Management.

Security Models for secure computing, access control, capability based systems, access control lists.

Virtualization. History of VMs. Variety of virtualization – full, partial, para. Emulators, simulators and virtualization. VM in languages – Java VM.

I/O Management Device driver design, Buffering and interrupt handling. File and Persistent Object Management File organization, directories and naming, index nodes, disk block management.

Network and distributed file systems Protection and Security Models for secure computing, access control, capability based systems, access control lists.

Embedded and mobile OS. Background to embedded and mobile os. Hardware and software requirements for embedded/mobile. Embedded/mobile OS.

Teaching and Learning Methods: Laboratory exercises will allow the student to gain familiarisation with the tools and techniques required for the implementation and verification of operating systems.

Students will be expected to demonstrate self-direction and originality in their learning which will be facilitated through student directed tutorials.

Scheduled learning: in the form of tutorials, demonstrations and practical classes will comprise 1/3 of the total study time for this module.

Independent learning: will constitute the remaining study time with an expectation that approximately 46 hours will be spent on self-directed study, a further 40 hours in support of the coursework and 16 hours preparation for the presentation.

Contact Hours:

Activity:
Contact: 48 hours
Assimilation and skill development: 42 hours
Undertaking coursework: 40 hours
Exam preparation: 20 hours
Total: 150 hours

Part 3: Assessment

Summative assessment is achieved through the demonstration of an innovative solution to a design problem along with submission of a logbook.

Formative assessment will be provided as oral feedback throughout the laboratory sessions particularly with respect to the design development and the log-book entries.

Final summative assessment, for more theoretical aspects of material, will be by exam.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Presentation - Component B		50 %	Logbook and demonstration of final product
Examination - Component A	✓	50 %	Exam (120 minutes)

Resit Components	Final Assessment	Element weighting	Description
Presentation - Component B		50 %	Logbook and demonstration of final product
Examination - Component A	✓	50 %	Exam (120 minutes)

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will achieve the following learning outcomes:

Module Learning Outcomes	Reference
Show a detailed knowledge and understanding of the design, structure and implementation of modern operating systems (OS) as well as the data structures and interfaces of a OS	MO1
Write small utility programs, in both script and compiler level languages, that interface to the system primitives	MO2
Build and modify a OS, with particular application to user/system interface and memory sub-systems	MO3
Understand the security problems and solutions in an OS	MO4

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	102
Total Independent Study Hours:	102
Scheduled Learning and Teaching Hours:	
Face-to-face learning	48
Total Scheduled Learning and Teaching Hours:	48
Hours to be allocated	150
Allocated Hours	150

Reading List

The reading list for this module can be accessed via the following link:

https://uwe.rl.talis.com/modules/ufcfwk-15-2.html

Part 5: Contributes Towards
This module contributes towards the following programmes of study: Computing [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19 Forensic Computing and Security [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19 Forensic Computing and Security {Dual} [Mar][FT][Taylors][3yrs] BSc (Hons) 2018-19 Forensic Computing and Security {Dual} [Aug][FT][Taylors][3yrs] BSc (Hons) 2018-19 Computing [Sep][FT][Frenchay][3yrs] BSc (Hons) 2018-19 Forensic Computing and Security [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19 Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19 Computer Science [May][FT][Villa][3yrs] BSc (Hons) 2018-19 Computer Science [Jan][FT][Villa][3yrs] BSc (Hons) 2018-19 Computer Science [Sep][FT][Villa][3yrs] BSc (Hons) 2018-19 Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2018-19