



## **Module Specification**

### **Operating Systems**

Version: 2021-22, v1.0, 05 Feb 2020

#### **Contents**

<b>Module Specification .....</b>	<b>1</b>
<b>Part 1: Information .....</b>	<b>2</b>
<b>Part 2: Description .....</b>	<b>2</b>
<b>Part 3: Teaching and learning methods .....</b>	<b>3</b>
<b>Part 4: Assessment.....</b>	<b>5</b>
<b>Part 5: Contributes towards .....</b>	<b>6</b>

## **Part 1: Information**

**Module title:** Operating Systems

**Module code:** UFCFWK-15-2

**Level:** Level 5

**For implementation from:** 2021-22

**UWE credit rating:** 15

**ECTS credit rating:** 7.5

**Faculty:** Faculty of Environment & Technology

**Department:** FET Dept of Computer Sci & Creative Tech

**Partner institutions:** None

**Delivery locations:** Frenchay Campus, Villa College

**Field:** Computer Science and Creative Technologies

**Module type:** Standard

**Pre-requisites:** Computer and Network Systems 2021-22

**Excluded combinations:** None

**Co-requisites:** None

**Continuing professional development:** No

**Professional, statutory or regulatory body requirements:** None

## **Part 2: Description**

**Overview:** Not applicable

**Features:** Not applicable

**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.

### **Part 3: Teaching and learning methods**

**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

**Module Learning outcomes:** On successful completion of this module students will achieve the following learning outcomes.

**MO1** 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 On successful completion of this module students will achieve the following learning outcomes.

**MO2** Write small utility programs, in both script and compiler level languages, that interface to the system primitives On successful completion of this module students will achieve the following learning outcomes.

**MO3** Build and modify a OS, with particular application to user/system interface and memory sub-systems On successful completion of this module students will achieve the following learning outcomes.

**MO4** Understand the security problems and solutions in an OS

**Hours to be allocated:** 150

**Contact hours:**

Independent study/self-guided study = 102 hours

Face-to-face learning = 48 hours

Total = 150

**Reading list:** The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://uwe.rl.talis.com/modules/ufcfwk-15-2.html) via the following link <https://uwe.rl.talis.com/modules/ufcfwk-15-2.html>

## **Part 4: Assessment**

**Assessment strategy:** 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.

**Assessment components:**

**Examination (Online) - Component A (First Sit)**

Description: Online examination

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4

**Final Project - Component B (First Sit)**

Description: Logbook and demonstration of final product

Weighting: 50 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

**Examination (Online) - Component A (Resit)**

Description: Online Examination

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested:

**Final Project - Component B (Resit)**

Description: Logbook and demonstration of final product

Weighting: 50 %

Final assessment: No

Group work: Yes

Learning outcomes tested:

**Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21

Computing [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21

Computing [Sep][SW][Frenchay][4yrs] BSc (Hons) 2020-21

Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2020-21

Forensic Computing and Security [Sep][SW][Frenchay][4yrs] BSc (Hons) 2020-21

Forensic Computing and Security [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21

Cyber Security and Digital Forensics [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21

Cyber Security and Digital Forensics [Sep][SW][Frenchay][4yrs] BSc (Hons) 2020-21

Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2019-20

Computing {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2019-20

Forensic Computing and Security {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2019-20

Forensic Computing and Security {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2019-20

Computer Science [Jan][FT][Villa][3yrs] BSc (Hons) 2020-21

Computer Science [May][FT][Villa][3yrs] BSc (Hons) 2020-21

Computer Science [Sep][FT][Villa][3yrs] BSc (Hons) 2020-21

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20

Computer Science {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2019-20

Computer Science {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2019-20

Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19

Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19