



## **Module Specification**

### **Operating Systems and System Programming [TSI]**

Version: 2023-24, v2.0, 09 Aug 2023

#### **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 and System Programming [TSI]

**Module code:** UFCE4L-12-1

**Level:** Level 4

**For implementation from:** 2023-24

**UWE credit rating:** 12

**ECTS credit rating:** 6

**College:** College of Arts, Technology and Environment

**School:** CATE School of Computing and Creative Technologies

**Partner institutions:** None

**Field:**

**Module type:** Module

**Pre-requisites:** None

**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:** The purpose is to give knowledge about the architecture of modern operating systems, about the types of file systems, the concepts of process, thread, memory; to form skills and abilities in memory management, process management and their interaction, thread management and synchronisation; to give

experience and practical knowledge of programming at the system level in Unix and MS Windows operating systems.

**Outline syllabus:** •Definition of the operating system.

- Operating system functions
- Operating system architecture.
- Classification of operating systems.
- Goals and objectives of system.
- Programming UNIX operating system.
- Introduction to the UNIX operating System.
- Operating system MS Windows.
- Message processing loop and graphical user interface.
- File Systems.
- Low-level file handling in Windows and Unix operating systems.
- Processes and Process Management.
- Threads management and its synchronisation.
- Data exchange between processes, Mechanisms of inter-processor communication.
- Memory management.
- Working with virtual memory.
- Input/output management.
- Introduction to operating systems.
- Classification, Architecture.
- Memory management.

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

**Teaching and learning methods:** The course utilizes various activities to evaluate learning outcomes. These activities include:

1) Automated quizzes: These quizzes are self-administered and serve as a formative assessment. They enable students to continuously evaluate their understanding of the course material. By identifying areas of weakness, students can focus on improving their comprehension.

2) Lectures and Laboratory work: Students engage in practical tasks to apply their theoretical knowledge gained during the lectures. The primary objective is to develop practical skills. Upon completion, students prepare a report that is assessed by the teacher using predefined grading criteria. Feedback is provided to the students, and their ratings and reviews are published on e.tsi.lv for reference.

3) Additional materials: Students are encouraged to study supplementary materials to enhance their understanding of the subject matter. While not explicitly mentioned as an assessment method, this activity likely contributes to the overall learning outcomes.

4) Examination: The course culminates in an exam that evaluates both theoretical knowledge and practical skills acquired throughout the course. This summative assessment assesses the students' overall understanding and application of the course content.

By incorporating these varied assessment methods, the course aims to comprehensively evaluate the students' learning outcomes, including their theoretical understanding, practical proficiency, and continuous self-assessment.

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

**MO1** Describe the functions and relationships between operating systems components.

**MO2** Produce code which efficiently manages operating systems functions: ie processes, threads, memory, files

**MO3** Use the possibilities of system programming to implement software solutions.

**Hours to be allocated:** 120

**Contact hours:**

Independent study/self-guided study = 96 hours

Face-to-face learning = 64 hours

Total = 160

**Reading list:** The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/8DA487E8-3C60-8BBA-F860-B7DECD1410E6.html?lang=en&login=1) via the following link <https://rl.talis.com/3/uwe/lists/8DA487E8-3C60-8BBA-F860-B7DECD1410E6.html?lang=en&login=1>

## Part 4: Assessment

**Assessment strategy:** Students will complete two pieces of assessment:

- 1) Written assignment; students are required to produce and document a software solution (for a given scenario) which utilises and manages OS resources efficiently.
2. Examination; this will test students understanding of the functions and relationship between OS components.

Resits, will be like for like; however, with a different scenario.

### Assessment tasks:

#### Written Assignment (First Sit)

Description: Students are required to produce the software solution, which used OS resources to compete specific scenarios. (max 3000 words)

Weighting: 70 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

#### Examination (First Sit)

Description: Examination (2 hours)

Weighting: 30 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1

**Written Assignment (Resit)**

Description: Students are required to produce the software solution, which used OS resources to compete specific scenarios. (Max 3000 words)

Weighting: 70 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

**Examination (Resit)**

Description: Examination (2 hours)

Weighting: 30 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1

**Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Computer Science and Software Development {Double Degree} {Foundation} [TSI]

BSc (Hons) 2022-23