Module Specification

Advanced Systems Programming

Version: 2023-24, v2.0, 06 Jul 2023

Contents

Part 1: Information

Module title: Advanced Systems Programming

Module code: UFCFWR-15-3

Level: Level 6

For implementation from: 2023-24

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

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: Systems Development Group Project 2022-23

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Computing has changed so much in the last decade that desktop computer become largely irrelevant. Instead heterogeneous, multicore, mobile, and real-time systems - smart mobile phones, netbooks, and laptops - are now ubiquitous. With all this we might have expected a move away from programming in C.

This course will introduce the student to modern system development and operating systems that ease the challenges of systems programming.

Features: Not applicable

Educational aims: This module aims to advance the students' understanding of programming concepts, in particular systems programming, with application to a wide set of applications, including embedded systems, games and audio programming, and high performance computing.

Outline syllabus: This module looks at the features offered by modern C++ and Mozilla's Rust programming language. Emphasis will be placed on system correctness and secure programming, to ensure the resulting systems are safe to use in an adversarial environment.

Part 3: Teaching and learning methods

Teaching and learning methods: Laboratory exercises will allow students to gain familiarisation with the tools and techniques required for the implementation and verification of safe embedded systems.

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

The lecture series will be supported by weekly practical sessions in which the students have the opportunity to apply some of the concepts discussed during the lecture series. The practicals will allow the students to explore and debug mobile devices and/or device simulations using a range of tools.

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

MO1 Develop modern low-level system programs using an appropriate programming language; (A/B)

MO2 To discuss the challenges of secure low-level programming and write secure code in a modern systems programming language to perform systems programming. (A/B)

MO3 Critically review and demonstrate the advantages and disadvantages of integrating automatic memory management with the operating system/runtime; (B)

MO4 Review and evaluate the role of different system programming languages, such as C, C++, and Rust. (B)

Hours to be allocated: 150

Contact hours:

Independent study/self-guided study = 114 hours

Face-to-face learning = 36 hours

Total = 150

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://eur01.safelinks.protection.outlook.com/?url=https%3A%2F%2Frl.talis.com%2F3%2Fuwe%2Flists%2FED9BB685-9800-1F2F-B8DD-38959339BBA2.html%3Flang%3Den-GB%26login%3D1&data=02%7C01%7CElias.Pimenidis%40uwe.ac.uk%7Ccd913533534e403ebc0908d7afb4b1ad%7C07ef1208413c4b5e9cdd64ef305754f0%7C0%7C0%7C637171062922471397&sdata=iQ%2B5V%2FCyLvLZntFB%2Fg37%2BI4Mt97UaKWfG92tOp%2F8AbU%3D&reserved=0

Part 4: Assessment

Assessment strategy: Students will work in small groups to complete a practical project related to Systems Programming. The work and its documentation will be assessed through a practical skills assessment. Completion of the project is supported throughout the module via practical exercises that are set and completed during practical sessions. Students will be demonstrate their solutions to the worksheets during class time. The demonstrations are signed-off the students receive formative feedback. The sign off sheet will be handed in as evidence of their work and this forms the portfolio part of the assessment.

Through the use of practical work students will also be assessed in their effective use and understanding of the tools and technologies that they utilise.

The referral coursework is the same as the first sit except that the student will be required to provide evidence of their achievements on the practical worksheets rather than an in-person demonstration.

Assessment tasks:

Portfolio (First Sit)
Description: Evidence of completed practical worksheets.
Weighting: 50 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2

Practical Skills Assessment (First Sit)
Description: Small group project with demonstration.
Weighting: 50 %
Final assessment: Yes
Group work: Yes
Learning outcomes tested: MO1, MO2, MO3, MO4

Portfolio (Resit)
Description: Evidence of completed practical worksheets.
Weighting: 50 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2

Practical Skills Assessment (Resit)

Description: Small group project with demonstration.

Weighting: 50 %

Final assessment: Yes

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2021-22

Computer Science [Sep][FT][Villa][3yrs] BSc (Hons) 2021-22

Computer Science [Jan][FT][Villa][3yrs] BSc (Hons) 2021-22

Computer Science [May][FT][Villa][3yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Sep][FT][Frenchay][4yrs] BSc (Hons) 2020-21

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