College of Arts, Technology and Environment
ACADEMIC YEAR 2023/24

Assessment Brief

Submission and feedback dates

Submission deadline: Before 14:00 on 04/01/2024

Is eligible for 48-hour late submission window.

Marks and Feedback due on: 24/01/2024

N.B. all times are 24-hour clock, current local time (at time of submission) in the UK.

Submission details

Module title and code: Advanced Systems Programming UFCFWR-15-3

Assessment type: Practical coursework examination with regular signoffs

Assessment title: Small group project - Component B

Assessment weighting: 50% of total module mark

Size or length of assessment: N/A

Module learning outcomes assessed by this task:

1. Develop modern low-level system programs using an appropriate programming language.
2. To discuss the challenges of secure low-level programming and write secure code in a modern systems programming language to perform systems programming.
3. Critically review and demonstrate the advantages and disadvantages of integrating automatic memory management with the operating system/runtime.
4. Review and evaluate the role of different system programming languages, such as C, C++, and Rust.

Completing your assessment

This is a group assignment, where groups can contain 1 or 2 members.

Larger groups are not permitted.

You must submit work individually, highlighting in the submitted readme who you worked with.

What am I required to do on this assessment?

For this component you are required to complete a group project, which is made up of multiple tasks. The main body of work will be to implement a C++ library that supports concurrent execution of co-routines, providing unit tests, examples of use, and documentation.

All work will be written in C++, developed on the csctcloud.uwe.ac.uk, and contained with a Gitlab repo.

The assignment details are on Blackboard, under Learning Material/Assignment, and submission is a Gitlab repo link, which is submitted via Blackboard.

Where should I start?

Before progressing to the assignment you need to complete the setup for the remote development server, csctcloud.uwe.ac.uk.

- Accessing CSCT Cloud using Azure CLI and SSH Keys
- Setting up remote development for VSCode

For the most part this should be straightforward as you continue to use the remote server as per last year and needed to be completed for component A assessment.

Once you have are again connected to the remote server you can use the following links to access the worksheets:

- Assignment

Worksheet 1 must be completed first, and has an earlier deadline, before moving onto worksheet 2.

What do I need to do to pass?

The assignment contains a marking scheme for each task, where each task is worth a percentage of the overall mark. You must get a total minimum of 40% to pass this component.

The following marking scheme is how each individual task will be marked.

IMPORTANT: Additionally, note that all work must be included in a Gitlab repo, with the link to this repo submitted on Blackboard. Failure to provide a Gitlab repo link will result in a mark of zero. It is not valid to submitted directly via Blackboard or via email.

Percentage	100-86 Outstanding	85-70 Excellent	69-60 Very Good	59-50 Good
	• **Impressive** demonstration of programming	• **Excellent** demonstration of programming	• **Very good** demonstration of programming	• **Good** demonstration of programming

	and software development skills. • Demonstrates **outstanding insight** into the technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Outstanding reflection** on the tests' results.	and software development skills. • Demonstrates **excellent insight** into the technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Excellent reflection** on the tests' results.	and software development skills. • Demonstrates **very good insight** into the technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Very good reflection** on the tests' results.	and software development skills. • Demonstrates **good insight** into the technologies employed. • Appropriate software testing; tests **most** aspects of the product in **detail**. **Good reflection** on the tests' results.

	49-40 Adequate	39-30 Poor / Inadequate	29-0 Very Poor	
	• **Some** demonstration of programming and software development skills. • Demonstrates **some insight** into the technologies employed. • **Some or Poor** software testing; tests **some** aspects of the product. **Some reflection** on the tests' results.	• **Little or no** demonstration of programming and software development skills. • Demonstrates **some insight** into the technologies employed. • Poor software testing; tests **some** aspects of the product. **Limited reflection** on the tests' results.	• **Little or no** demonstration of programming skills. • Demonstrates **little or no insight** into the technologies employed. • **Little or no** software testing. **Limited or no reflection** on the tests' results.	

How do I achieve high marks in this assessment?

Completing all the tasks in the assignment will get you a good mark, but you need to add some additionally features, ones not covered in the specification, to get a mark above 72. We will discuss suitable examples of these in the practicals as the module progresses.

A key requirement for this assessment is a high-quality README.md, documenting your work, how to use it, and demonstration of it running. This is README is written in Markdown, see Blackboard for resources, but also the cheat sheet is useful.

How does the learning and teaching relate to the assessment?

Lectures 1-8 will cover martial that will be used to complete this assessment.

See the Blackboard schedule for details of what topics will be covered when.

What additional resources may help me complete this assessment?

- It is critical for success on this module to attend the weekly lecture and your practical.
- Markdown cheat sheet.
- Nathan Renney and Benedict Gaster have office hours, please see Blackboard for times and how to book a slot.

What do I do if I am concerned about completing this assessment?

UWE Bristol offer a range of Assessment Support Options that you can explore through this link, and both Academic Support and Wellbeing Support are available.

For further information, please see the Academic Survival Guide.

How do I avoid an Assessment Offence on this module? [2]

Use the support above if you feel unable to submit your own work for this module.

Marks and Feedback

Your assessment will be marked according to the following marking criteria.

Percentage	100-86 Outstanding	85-70 Excellent	69-60 Very Good	59-50 Good
	• **Impressive** demonstration of programming and software development skills. • Demonstrates **outstanding insight** into the	• **Excellent** demonstration of programming and software development skills. • Demonstrates **excellent insight** into the	• **Very good** demonstration of programming and software development skills. • Demonstrates **very good insight** into the	• **Good** demonstration of programming and software development skills. • Demonstrates **good insight** into the technologies employed.

	technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Outstanding reflection** on the tests' results.	technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Excellent reflection** on the tests' results.	technologies employed. • Appropriate software testing; tests all aspects of the product in **great detail**. **Very good reflection** on the tests' results.	• Appropriate software testing; tests **most** aspects of the product in **detail**. **Good reflection** on the tests' results.

You can use these to evaluate your own work before you submit.

1. In line with UWE Bristol's Assessment Content Limit Policy (formerly the Word Count Policy), word count includes all text, including (but not limited to): the main body of text (including headings), all citations (both in and out of brackets), text boxes, tables and graphs, figures and diagrams, quotes, lists.
2. UWE Bristol's UWE's Assessment Offences Policy requires that you submit work that is entirely your own and reflects your own learning, so it is important to:
    - Ensure you reference all sources used, using the UWE Harvard/OSCOLA system and the guidance available on UWE's Study Skills referencing pages.
    - Avoid copying and pasting any work into this assessment, including your own previous assessments, work from other students or internet sources
    - Develop your own style, arguments and wording, so avoid copying sources and changing individual words but keeping, essentially, the same sentences and/or structures from other sources
    - Never give your work to others who may copy it
    - If an individual assessment, develop your own work and preparation, and do not allow anyone to make amends on your work (including proof-readers, who may highlight issues but not edit the work) and

    When submitting your work, you will be required to confirm that the work is your own, and text-matching software and other methods are routinely used to check submissions against other submissions to the university and internet sources. Details of what constitutes plagiarism and how to avoid it can be found on UWE's Study Skills pages about avoiding plagiarism.