Operating Systems

Team Project Assignment (Worth 25%)

A Study of the Synchronisation and Concurrency Issues in the Dining Philosophers' Problem completed using the ThreadMentor Visualisation Tool

Dr. Kevin Farrell
eMail: kevin.farrell@tudublin.ie

January 2025

Table of Contents

1. Changes to this document

Version 1: 28 January 2025 – First version.

Version 2: 6 March 2025 – Updated the dates of the Mid-Semester Project Progress Presentations, and updated date of final presentations.

2. Objectives

You are required to work in teams of **four** students (a team of **three** students is permissible depending on numbers in the Lab Group. **Please note**: all team members **must** be in the same Lab Group.

The project problem is to study, using the ThreadMentor software tool, the synchronisation and concurrency issues arising in a programmed solution to the Dining Philosophers problem. The **ThreadMentor e-book website** is available at:

https://www.cs.mtu.edu/~shene/NSF-3/e-Book/

Put simply, the project problem is to study a running program, called the Dining Philosophers problem, which has multiple threads (multiple subprocesses), and to understand how those threads are programmed so that they take it in turns to access shared resources (chopsticks).

3. Summary of Tasks Required

The following is a summary of the tasks required by you and your team. Please note that the detailed deliverables are given in the section 4. One of your main tasks is to learn how to accomplish the following tasks as you progress through the semester. These are given in the *approximate* order in which you should undertake them. You are **not** expected to know how to do them at the start:

1. Learn about the Dining Philosophers problem – lots of resources online and in OS books.
2. Learn about ThreadMentor and how it deals with Threads – see the ThreadMentor e-book website (above).
3. Study and understand the programmed solutions to the Dining Philosophers' Problem provided on the ThreadMentor e-book website. You'll need to learn a little about C++ in the process. **Note**: you won't be asked to write C++ programs.
4. There are **three** categories of *solution* to the Dining Philosophers problem presented on the ThreadMentor e-book website. However, for this project, I ask you to learn about **two** of those categories. You might be wondering what we mean by a "solution" to the problem: A solution is a set of C++ files (with some ThreadMentor

C++ code in them) which allow you to create an executable program. When you run this program, it allows you to visualise (thanks to ThreadMentor) the execution of the multiple threads and the interactions between them; in particular, how the Philosophers are programmed to take it turns to access the shared resources (chopsticks). There are *several* different solutions provided in one category. You should read about **ALL** of these solutions, and then choose **ONE** to study. The categories I ask you to study are as follows:

a) Mutual Exclusion Locks (Two solutions)

b) Semaphores (One solution)

5. Set up ThreadMentor on your Linux VM: We'll be doing this in a practical session.
6. Download the Dining Philosophers C++ code given on the ThreadMentor e-book website – there are several files to download for each solution: We'll be doing this in a practical session.
7. Create a Makefile, and then compile the code with the aid of it: We'll be doing this in a practical session.
8. Run the resulting program using the ThreadMentor visualisation tool, analysing the running threads **and** the corresponding lines of code at various times in its execution: We'll be doing *some* of this in a practical session.
9. For your chosen solution, study it, give a mid-semester progress presentation, write a final report, give a final presentation/demonstration discussing the synchronisation and concurrency issues in the execution of the programmed solution, and answer questions from your Lecturer. As part of your discussion, your report **and** presentation **must** contain a description of sequences of screenshots of the visualised execution, accompanied by corresponding source code fragments. Therefore, when running the program(s), it is important to take appropriate ThreadMentor screenshots and code snapshots, so that they can be included in your report. You can use the **spectacle** screenshot tool on Mageia Linux for this.
10. **Notes**:

    a) To maximise your grade, it is absolutely essential that you provide screenshots with corresponding code snapshots in both your presentations and report.

b) When you click on a Tag in the ThreadMentor application, it opens a window with the corresponding code highlighted.

c) The sequence of screenshots you finally use in your presentation and report **must** be obtained during a single run of the program, since the program behaviour is different each time you run it.

4. Deliverables

There are **three** deliverables for this project, given below. This project will be awarded a total of **25%** of the marks for the Operating Systems module divided as follows:

Description	**% points**
Mid-Semester Progress Presentation	5%
End of Semester Presentation/Demonstration and Team Report Marks are *generally* divided equally between the presentation/demonstration and the Team report.	20%
Project Total	**25%**

Grading of team members will be equal, except in circumstances where it becomes apparent that some team members are not contributing adequately. **In particular, see the notes below.**

4.1 Mid-Semester Progress Presentation: Practical Session in Week beginning 17 March for Gerome's Groups, and Week Beginning 24 March for Kevin's Groups

Give a mid-semester team powerpoint-style presentation on the progress of your project of **5 minutes maximum**. At this point in the semester, I am expecting your team to be able to:

- Explain what the Dining Philosophers Problem is.
- Compile and run the code of your chosen solution to the Dining Philosophers Problem.
- Outline your Project work for the remainder of the semester, including providing a *legible* Gantt chart.
- Answer questions on *some* aspects of your chosen solution.

The following is the indicative marking scheme for the Mid-Semester Progress Presentation:

GRADING CRITERION	**MAX. MARKS (%)**
Description of progress so far, including explaining the Dining Philosophers Problem (could include reading – requires evidence of understanding)	Max = 15
Have the students compiled and run a solution for the Dining Philosophers' Problem? (Not looking for detailed explanation of what's happening during the execution – that's end of semester presentation)	Successful Compile + Run = 30 OR Attempted Compile (+ *stated* errors) = 15 OR 0
Gantt chart with a brief description	10
Answering Questions *(Students should demonstrate a basic knowledge of their chosen solution. Detailed knowledge not required at this point).*	Max = 30 OR If no questions can be asked due to lack of progress = 0
Style & Presentation	15
TOTAL	**100**

4.2 Team Report

Write a Team report, with the following properties:

1. Is 5,000 words maximum. This is not the word count target – it is the limit!
2. It must contain a list of **every member** on your team (student number and full name) on the title page. Only those members who contributed to the project and report should be listed. As a team member, you should be fully familiar with the entire report and all aspects of the project – not just the sections you wrote/worked on!
3. It **must** contain the following declaration immediately after the Title Page:
   "We are aware of the University policy on plagiarism in assignments and examinations (3AS08). We understand that plagiarism, collusion, and copying are grave and serious offences in the University and we will accept the penalties that

could be imposed if we engage in any such activity. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study. We declare that this material, which we now submit for assessment, is entirely of our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work."

Note: failure to include the declaration will result in a **zero** mark for the report.

4. It **must** be a document, in PDF format, and be named in accordance with the following naming rule:

   Surname1_Surname2_Surname3_Surname4_OS2025.pdf

5. It must contain proper referencing/citation. All sources used for both text and diagrams **must** be referenced/cited **within the body** of your report, with the list of references provided at the end of the document; this also includes websites. If you don't understand what this means, it is your responsibility to ask me. The library website also has useful information on how to reference/cite correctly. Please **do not** use Wikipedia as a reference; instead, if you do use Wikipedia as an *information source*, please go to the original references cited within those Wikipedia pages and read them, and cite the ones that you use. **Note**: the inclusion, without proper citation, whether intentionally or unintentionally, of material belonging to others is plagiarism. Including large blocks of text from other sources, even if cited/referenced, is also plagiarism. Short quotations are fine. Reports that do not cite correctly may attract a **zero or near-zero mark** depending on the severity of the plagiarism. In cases, where a team of students allows their work to be plagiarised by another team, both teams may a receive a **zero mark for the Project**. If you are repeating the module, you are **not** permitted to resubmit work that you previously submitted. This also constitutes plagiarism. Cases of plagiarism **will** be processed through the University's Plagiarism procedures, with a record being kept by the School of Informatics and Cybersecurity of the incident, sanctions and of the identities of the students involved.

6. It must be typed with a spacing of at least 1.5. Each student is advised to retain an additional hard-copy for future study.

7. Reports should be uploaded by **one member only** of each team to BrightSpace (the link will be provided nearer the deadline). We currently use the Turnitin antiplagiarism system. So you are permitted a **one-time only** upload. You will **not** be permitted to upload a "draft" version and then a "final" version later. This will be strictly enforced. So, please make sure that all members of your team are happy with your report before you upload it. **The deadline for uploads is the day of your practical session in Week 12; i.e. your practical session in the week beginning 28th April 2025 - see notes below.**

The following is the *indicative* marking scheme for the Team Report:

GRADING CRITERION	MAX. MARKS (%)
Introduction and Background	10
Evidence and discussion of Compilation	15
Discussion of results with screenshots of History Graph + Tags and Corresponding Code and Thread Status Graphs. Should include a demonstration of understanding of the synchronisation issues.	40
Proper citation	20
Style	5
Conclusions	10
TOTAL	**100**

4.3 Project Presentation and Demonstration (Practical Session in the week beginning 28 April)

Give a **10 minute** presentation on the project in your practical session in Week 12; to include a demonstration and verbal description of the running program if requested by me.

Please DO NOT spend time in your final presentation giving a general explanation of what the Dining Philosophers problem is. We will all know what it is by that stage! You will lose marks if you do this.

A copy of your team's presentation must also be uploaded by **one member only** to a link on BrightSpace prior to the practical session – **see notes below.**

The following is the indicative marking scheme for the Presentation:

GRADING CRITERION	**MAX. MARKS (%)**
Presentation Overview slide	5
Background to your solution; relevant general synchronisation methodologies	5
How your solution works	10
Demonstrating how your solution works: **legible** screenshots of the history graph and other ThreadMentor windows; understanding ThreadMentor tags in context of your solution	15
Demonstrating how it works: related **legible** code fragments, and how these relate to each history graph, etc.	15
Answering Questions	45
Timing, Style and Presentation	5
TOTAL	100

4.4 Notes on Mid-Semester and Final Presentations

1. Please make sure your images are big enough/legible in your presentation. It is often better to put one image on each slide and use multiple slides, rather than put multiple images on one slide, where they are then too small to understand.
2. This also applies to images of code: it is usually better to list *only* the fragments of code that you want to discuss, rather than an entire class or method where the text is too small to read. Bear in mind that your audience can only read and understand a small amount of code on a slide in the short time that is available.
3. Each team will also be asked questions on their project at the end of their Mid-semester and Final presentation. This is an ***essential*** part of the assessment. These questions will generally relate to the content of your team's presentation and project *and* your team's and individuals' wider knowledge and understanding of OS concepts related to the project. The duration of questioning is dependent on the quality of the presentation. As a rule of thumb, presentations which are of a high standard usually

involve a shorter questioning period. **Please note**: the purpose of asking you questions is not to 'catch you out' or to make you feel uncomfortable. It is to try to assess your level of involvement and work contributed to the project by you and your team so that you can be awarded a fair grade.

4. The Project Presentations and Report may be graded on an individual basis, where it becomes clear that some members of the team did not contribute fully to the project.
5. It is a ***requirement*** of the Project that your team take part in the presentation/demonstration at the end of the semester. This is so that I can verify that the project work is your own work – this is where the questioning comes in. Unless there is a really good reason, I will not accept Project Reports where the team has not undertaken the presentation/demonstration, and your team will consequently score 0/15 for that part of the project.
6. Similarly, if any individual fails to take part in the presentation/demonstration without good reason (for e.g.: certified medical reasons, family illness or bereavement, etc.), he/she will score 0/15 for ***both*** the presentation/demonstration ***and*** the report. This is for the same reasons as above; i.e. the presentation and subsequent questioning is used to help verify that your individual contribution to the project work is your own work. The remaining members of the team will be graded as normal.

5. What am I looking for?

The following are ***guidelines*** of what I am looking for from students in their presentations, report and in answers to questions:

- Research: demonstrating that you have engaged in a reasonably broad and in-depth research of the project subject for a second year Computing student.
- Understanding and discussion of topic: demonstrating in your presentation and report that you understand the subject, ***and*** that you are able to discuss it using appropriate ***technical*** language.
- Originality: showing an interesting original approach.
- Professional style in both presentation and report: Use of title page, tables of contents and figures, consistent fonts, heading styles, page numbers, etc.

- Adherence to word limit in the report: being brief is just as much a skill as being able to waffle!
- Inclusion and proper use of citation/references in both report and presentation: You must provide citations ***within the body*** of your report and presentation. Please ***do not*** use Wikipedia as a reference; instead, if you do use Wikipedia as an information source, please go to the original references cited within those Wikipedia pages, read them, and then cite them in your document.

6. Team Difficulties: Interpersonal Issues, Members not Working, etc.

Sometimes there can be difficulties with members in a team not getting along or not pulling their weight. Please let me know by email ***as early as possible*** in the project so that we can meet to try to resolve the issues. In my experience, early discussion around issues often solves them. Please ***DO NOT*** wait until the week before the project report and presentation are due, to ask me for assistance as there will be little I can do at that stage to help resolve matters. E-mail: kevin.farrell@tudublin.ie

7. Questions?

If you have questions or if some aspect of the project is unclear, please post your question to the Q&A Forum, email me or ask me in class.