MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG334 Introduction to Operating Systems – Assignment 2

Date Handed Out	**07 May at 06:00 PM**
Submission Date	**22 of May at 11: 00 PM**

Important: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

1. **In the report to be submitted**: please include your name and student IDs on the first page.
2. **Your code files,** MUST include name and student IDs.
3. We recommend typesetting your submission in Word or any similar tool as you must submit the PDF version of it. Images of handwritten solutions, unreadable reports due to unclarity, or English language typos will not be accepted.
4. As part of the typesetting requirement, all (state) graphs (if any are needed) must be computer-generated (no hand-drawn or stylus-drawn graphs will be accepted). We recommend using Powerpoint/Google Slides or any other tool you prefer to draw any graphs.
5. **Cheating will be punished according to the rules mentioned in the syllabus.**

Task 1 [10 Marks]

Consider a system with five processes and four different types of resources as described below:

Processes	Allocation	Maximum
A	0 0 1 2	0 0 1 2
B	1 0 0 0	1 7 5 0
C	1 3 5 4	2 3 5 6
D	0 6 3 2	0 6 5 2
E	0 0 1 4	0 6 5 6

Available: R1=1, R2=5, R3=2, R4=0

Q1: Consider the Available refers to the remaining resources after the allocation shown in the table. Provide a safe sequence if possible. Please, provide your answer with detailed steps

Task 2 [20 marks]

Consider a system with three processes P1, P2, P3, and P4. There are types of serially reusable resources R1, R2, and R3 with 3, 2, and 2 instances.

Process P1 holds 1 instance of R1 and requests 1 instance of R2.
Process P2 holds 2 instances of R2 and requests 1 instance each of R1 and R3. Process P3 holds 1 instance of R1 and requests 1 instance of R2.
Process P4 holds 2 instances of R3 and requests 1 instance of R1.

Q2: Draw the resource allocation graph of this system. Does the graph illustrate a deadlock or not, please explain your answer in detail.

Task 3 [20 Marks] Processes

Attached to this assignment, there is a file .c called mystery. You are required to:

1) Read carefully the code and run it on a linux based operating system.

2) Explain clearly what the code does and why the output is like that.

To get a full mark you have to explain what the functions **mmap()** and **munmap()** do. To help you with this please find below a reference.

Reference:

- https://www.ibm.com/docs/en/zos/2.4.0?topic=functions-mmap-map-pages-memory

Task 4 [50 Marks] Threads

1) Convert the code you find in the file mystery. c to a code that uses threads and explain what was the difference between your new code and the one in the file mystery. c. [20 Marks]

2) [30 Marks] There are various ways to calculate the value of PI, one of them is explained in the link below:

Reference: https://towardsdatascience.com/estimate-pi-using-random-numbers-8b13a7e8c791

The code to do the calculation is already implemented for you, find it attached to the assignment. The file name is CNG334_Task4. For this task, you are required to:

- Create a constant that defines the total number of threads you are planning to use.
- Add a global variable named circle_count and initialize it to zero.

- Protect your global variable using mutex from Pthread. You can use the following reference to learn how to use Pthread mutex directly.

Reference: https://www.ibm.com/docs/en/i/7.3?topic=ssw_ibm_i_73/apis/users_61.html

- Explain clearly how your code is working before and after the mutex. To get the full mark you have to explain why the mutex is necessary and what each function you used is doing.

NOTE: Initialize your mutex to null.

Bonus Task [20 Marks](No Partial Marks)

This task is optional and the marks gained here will be added to your 1st quiz, 1st question only (not applicable for students who had quiz makeup). Please note, if you had the full mark in question 1 in quiz 1 these points will not be added. In addition, the question in the quiz is graded on a scale from 0 to 50, thus if by adding these points you will exceed the scale the extra point will not be considered.

Process	Arrival time	Total CPU Time	I/O Blocking Time	I/O Interrupt time
P1	0	2	1	1
P2	1	2	1	1
P3	0	3	0	0

Consider that an I/O interrupt occurs only once from the first time the process gets the CPU. Draw the Gantt chart (context switching is ignored here) for the following algorithms considering that the process will be blocked for a time period shown in I/O blocking time, then compute the average process waiting time. Provide the Gantt chart for two of these scheduling algorithms

- **FIFO, SPN, and RR with a time slot equal to 2**

For this assignment, you have to submit:

A) A PDF file that includes:

- Your answer to the questions in tasks 1, 2, 3, and 4.
- Your answer to the bonus question (Optional)
- The code of task 4, question 1