****

School of Computing Science

**CSC1007 – Operating Systems**

**Assessed Coursework**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Name** | **CSC1007 – Operating Systems** | | | | |
| **Coursework No.** | **Assignment** | | | | |
| **Deadline** | **Time** | **23:59** | **Date** | **2 April 2021** | |
| **% Contribution to final course mark** | | | **25%** | | |
| **Solo or Group** | **Solo** |  | **Group** | | **√** |
| **Anticipated Hours** | **10 hours per group member** | | | | |
| **Submission Instructions** | **Submit your assignment report to the Dropbox of LMS Xsite** | | | | |
| **Please Note: This Coursework cannot be Re-Done** | | | | | |

**Code of Assessment Rules for Coursework Submission**

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline

a. the work will be assessed in the usual way;

b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause to Admin-In-Charge

**Penalty for non-adherence to Submission Instructions is 2 bands**

# Objectives

The objectives of the assignment are to reinforce the knowledge learnt in CSC1007 – Operating Systems, particularly on the process deadlocks avoidance and memory managements. We will use C programs to simulate the algorithms performed by the operating systems.

You can write your C program and execute the codes on your laptop using Visual Studio Code. But you can also choose the run the C programs on your Raspberry Pi, depending on your own decision.

# Group assignment questions

Please answer all questions, using a report in the Microsoft Word format, with the C programming codes copied in the report.

Please also attach the separate C programming codes .c files, .h files, and supporting files in the submission together with the WORD report.

**Question 1: (50 Marks)**

We have learnt the Banker’s algorithm for process deadlock avoidance in Lecture 6. In the first exercise, write a C programming which is to simulate the functions of the OS to implement the Banker’s algorithm, to detect if the system is in Safe state or unsafe state.

If it is in the safe state, your Banker’s algorithm need print out (display) the execution sequences in which these processes may complete on the screen.

If it is in the unsafe state, your Banker’s algorithm need display which process it fails at by showing the available resources at this point of time, and the processes needed resources at this point of time.

There are six processes to be allocated. There are four resources types (A, B, C, D). Your Banker’s program need read separated 2D array values in the C program:

* allocation[6][4]: the values of the allocation matrix of the processes,
* Max[6][4]: the values of the Max matrix of the processes.

An example allocation matrix and Max matrix are shown as follows. You can change the values of these two matrices if you want.

*Allocation Max Available*

*A B C D A B C D A B C D*

*P*0 2 1 3 3 7 3 4 5 ? ? ? ?

*P*1 2 3 1 2 8 6 2 1

*P*2 3 3 3 1 9 5 5 6

*P*3 2 1 3 4 6 4 6 3

*P*4  3 2 2 5 8 3 2 4

*P*5  2 1 2 3 8 3 2 3

**Displayed Information for Users:**

Your C program need first display the minimum accepted number of instances of resource types A, B, C, D. For example:

* Type A need be larger than 46.
* Type B need be larger than 24.
* Type C need be larger than 21.
* Type D need be larger than 22.

Hint: how to calculate the minimum number of instances of each resource type? According to the Max matrix values in your C program, you need calculate the summation of each resource type of all processes. The input number of instances need be larger than the summation of the corresponding resource type.

**Inputs:**

Your C program need take four inputs as follows:

1. Input 1: the number of total instances of resource type A. (e.g., entering 50, means there are 50 instances of resource type A).
2. Input 2: the number of total instances of resource type B.
3. Input 3: the number of total instances of resource type C.
4. Input 4: the number of total instances of resource type D.

**Output:**

* Your C program need display the execution sequences in which these processes may complete on the screen, if it is in the safe state.
* You C program need display which process it fails at by showing the available resources at this point of time, and the processes needed resources at this point of time, if it is in the unsafe state.

In your C programming codes, please add clear and sufficient comments for every line of the C codes, to help readers better understand your codes.

**Question 2: (50 Marks)**

We have learnt the dynamic memory allocation problem with three approaches: First-fit, Best-fit, and Worse-fit in Lecture 8. In the second exercise, write a C programming which is to simulate the functions of OS to implement these three dynamic memory allocation approaches, and compare the results.

Given 12 memory partitions available (holes): 160 KB, 350 KB, 650 KB, 80 KB, 410 KB, 50 KB, 720 KB, 905 KB, 570 KB, 130 KB, 260 KB, and 830 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms allocate 10 input processes with different sizes (in order)?

You can hardcode the 12 memory partitions’ sizes in an array in your C program. But the sizes of 10 input processes will be keyed in by users using the keyboard to your C program. Whenever users key in 10 sizes of processes from the keyboard, your C program need perform the allocations into 12 memory partitions using three different algorithms:

1. What will be the resultant memory partitions after allocating these 10 input processes using First-fit algorithm, and print out (display) on the screen?
2. What will be the resultant memory partitions after allocating these 10 input processes using Best-fit algorithm, and print out (display) on the screen?
3. What will be the resultant memory partitions after allocating these 10 input processes using Worse-fit algorithm, and print out (display) on the screen?

**Displayed Information for Users:**

Your C program need first display the maximum accepted size of input processes: Input size will be smaller than the largest value in the array of your memory partitions. In this case, the maximum value of the memory partition is 905 KB.

**Inputs:**

Your C program need take any 10 inputs which are the sizes (KB) of processes, separated by commas. These input values are keyed in by users at run time, so your C program cannot hardcode them.

For example: 123, 456, 78, 901, 234, 567, 89, 600, 200, 345.

**Output:**

Your C program need display three types of information:

* Display the sizes of the remaining memory partitions after allocating these 10 input processes using the First-fit algorithm.
* Display the sizes of the remaining memory partitions after allocating these 10 input processes using the Best-fit algorithm.
* Display the sizes of the remaining memory partitions after allocating these 10 input processes using the Worse-fit algorithm.

If any input processes cannot be allocated into any memory partition, your C program need display the message. For example, the input process with size 600 KB cannot fit into any remaining memory partition. It has to wait for other processes’ completion.

In your C programming codes, please add clear and sufficient comments for every line of the C codes, to help readers better understand your codes.

# Assessment criteria

Assignments will be assessed according to the criteria for the Group Assessment and the Individual Assessment. Note that the criteria are subject to change depending on the progress throughout the trimester. Following timely submission, the exercise will be given a numerical mark between 0 (no submission) and 100 (perfect in every way). The numerical marks will then be converted to a grade.

Your mark for this assignment will be computed as follows:

**Each group member grade = Group assessment \* (weighted peer review score by each group member and his/her presentation performance in presentation video)**

*Table 1. Group assessment\**

\*Group assessment will be weighted by peer review

|  |  |
| --- | --- |
| **Criteria** | **Weight** |
| Codes implementation:   * Code quality * Code structure * Functionalities * Neat variable and function names | 30 |
| Quality of comments and documentation in the source codes, to help readers better understand your source codes. | 20 |
| Solutions that produces the correct results. | 20 |
| Report and PowerPoint presentation video:   * Professional writing of the report with a clear logic and structure of your solutions. * A clear and smooth demonstration and presentation of your solutions in the 4 minutes presentation video. * Every group member must give the speech on his/her respective portion in the assignment. * Highlight the strength and limitation of your solutions. * List down every group member’s individual workloads and contributions in the report and presentation. * Reflection from each group member regarding their learning journey throughout the assignment. | 30 |

*Table 2. Individual contribution assessment*

|  |
| --- |
| Each individual team member is required to implement a portion of the programming codes, documentation and report. Every group member must give the speech on his/her respective portion in the assignment. Therefore, a clear task distribution is required. Peer evaluation score will be used as part of the individual contribution assessment factor. |

# Timeline and deliverables

Final submission: **Friday, 2 April 2021, 23:59 pm**.

Each group is to submit a Px-Groupxx-CSC1007.zip file to your group’s LMS xSITE Dropbox (‘CSC1007 - OS Group Assignment’). Px means labs P1, P2, or P3. Groupxx means the group number in the corresponding Px. One submission per group only.

The Px-Groupxx-CSC1007.zip file should contain files as follows:

* the **WORD format final report** named Px-Groupxx-report-CSC1007.doc,
* all the individual source codes (.c, .h files),
* a group power point presentation video Px-Groupxx-CSC1007.mp4, maximum 4 minutes length where all group members must give their speech on their respective portion of works in this group assignment.