Bangladesh University of Engineering and Technology
CSE 314: Operating Systems Sessional
IPC Assignment

Science Project

1. Scenario

Suppose, there are N students in a class with student IDs 1, 2, ….,N. They are working in groups for science projects. Each group has a size of M (where N = cM, c is a positive integer). Groups are formed based on their students' IDs. For example, if M= 3, students with IDs 1, 2, 3 form Group #1, students with IDs 4, 5, 6 form Group #2, and so on.

Each group has to submit a written project report to the Library. Let us see a 3-phase workflow for one group G. The members of G are denoted as S_1, S_2, …, S_M. The group leader is S_M.

> *Printing Phase*: S_1 independently uses a print station and prints his/her part of writing. Then, s/he hands over the printed copy to S_M. Similarly, S_2, S_3, …S_M follow the same procedure.
> *Binding Phase*: When S_M receives the printed copy from each of S_1, S_2, ……S_M, s/he goes to a binding station to bind the report.
> *Submission Phase*: S_M goes to the Library and submit the report. To complete the submission, S_M needs to sign in an entry book.

2. Concurrency and Mutual Exclusion

Submission of a report is not as simple as it seems. Resource limitation and sharing make the task challenging.

Challenge #1: There are 4 printing stations: PS_1,...,PS_4. The rule to assign a student to a printer is (student ID mod 4 + 1). For example, student with ID = 4 goes to PS_1, student with ID = 1 goes to PS_2, and so on. Not necessarily, all students arrive at the printing stations simultaneously. If any printing station is already in use, one needs to wait.

Challenge #2: There are 2 binding stations. Therefore, two groups can be served simultaneously.

Challenge #3: There exists a single entry-book in the library. Only one group can write in that book at a time. However, more than one person can read the entry information simultaneously while no one is writing in that book.

3. Tasks

Task #1: When a student arrives at the assigned printing station and finds that the station is occupied, s/he waits in the waiting room. When a student S finishes using a station PS, s/he sends a text to everyone who is waiting on that station to tell them about the availability of PS. S always wants his/her group-mates to get the station next. Therefore, S first sends a text to his/her group mates (if any) who might be waiting on PS. Then, S sends a text to other students who might be waiting on PS. (S always sends one text per person)
Your task is to implement this part using the logical construct of the classical dining-philosopher problem.
Task #2: The group leader from each group performs the task of binding. The leader waits until it receives printed copies from all group members. There are no assignments for the binding station. A leader can choose to go to any of the machines and wait if neither of the stations is available. Your job is to ensure *concurrency.* You can not use any pre-assigned group-to-binding station mapping. Also you are not allowed to introduce busy-waiting.
Task #3: The group leader from each group goes to the library after the binding is done. To complete submission s/he needs to write in the entry-book. S/he can only write if one else is reading/ writing.
There are 2 staff members in the library who keep track of the submission. One person is in charge of reporting to the science teacher the periodic update of the book entries. Another person is in charge of showing the period updates in the digital display. Your task is to implement this part using the logical construct of reader-writer problem where a writer waits until all readers finish reading.

4. Implementation Guideline

- There **MUST NOT** be any busy waiting anywhere in the implementation.
- You can generate all students at once and populate student IDs. However, to achieve randomness, you should add random delays before a student steps into the printing phase.
- Use Poisson distribution to generate random numbers. Use a suitable inter-arrival rate.
- The timing of the operations should be implemented using sleep. The relative time for each operation is given.
- The record in the entry book is basically the total number of submissions (initially 0) which you can implement using a shared variable. Use different random intervals for the two readers of the entry book. For example, Staff 1 can read at time 1, Staff 2 starts reading at time 3 while Staff 1 is already reading, then Staff 1 at time 8 and so on. Readers can stay in the system from the beginning and use some randomness to show concurrency
- Print every move of a student in detail (with timing info). Samples are given just to show the output format.

Operation Name	Relative Time Unit

Printing	w
Binding	x
Reading/Writing	y

- Input/Output:
  - You will take input from a file and give output in an output file
  - The format of the input file is:

    N M

    w x y

    Where N = Number of students

    M = Size of each group

    w, x, y = relative time units for the operations

5. Sample Input/Output

Input	**Output (complete for passenger 1)**
15 5 **10 8 3**	Student 4 has arrived at the print station at time 4 Student 2 has arrived at the print station at time 5 Student 1 has arrived at the print station at time 7 Student 3 has arrived at the print station at time 8 Student 6 has arrived at the print station at time 10 Student 7 has arrived at the print station at time 10 Student 5 has arrived at the print station at time 12 Student 8 has arrived at the print station at time 14 …. Staff 1 has started reading the entry book at time 13. No. of submission = 0 Student 4 has finished printing at time 14 Staff 2 has started reading the entry book at time 17. No. of submission = 0 ….. Group 1 has finished printing at time 60 …… Staff 2 has started reading the entry book at time 78. No. of submission = 0 Group 2 has started binding at time 80 Staff 1 has started reading the entry book at time 82. No. of submission = 0 Group 2 has finished binding at time 88 …..

	Group 2 has submitted the report at time 95 Staff 2 has started reading the entry book at time 98. No. of submission = 1

The timestamps and the relative order of the print statements are imaginary here. In your program you will calculate the times based on the interaction between the system and the passengers.

6. Marks Distribution

Tasks	Subtask	Marks
Randomness in student arrival at printing stations		**10**
Task 1	Determining availability of printer without busy waiting	**15**
	Promoting own group-mate to get the printer	**10**
Task 2	Implementation of group leaders wait for the group mates to finish printing without busy waiting	**15**
	Implementation of binding process ensuring concurrency	**20**
Task 3	Implementation of reader-writer problem with readers having higher priority	**10**
Printing moves of a student/group		**5**
Printing moves of a student/group with timing information		**10**
Introduce some random waits in code using sleep to mimic a real scenario in your output		**5**

7. Submission

- Create a directory by your 7 digit student name (1905XXX).
- Put your source code inside.
- Zip the folder, rename it to 1905XXX.zip.
- Submit the zipped folder.

Deadline: August 07, 2023. 11:55 PM.

Plagiarism Policy

- –100% marks will be deducted for plagiarism.
- Work on the problem on your own.