Week12 Assignment

Please complete a report in English in English in English in English and upload the corresponding codes.

The files should be uploaded directly without compression without compression without compression without compression

The files to be submitted for this assignment are:

1. report.pdf
2. phy1.c
3. phy2.c
4. milk.c

1. Deadlock [30 pts]

Given four processes P1, P2, P3, P4 and four resource types in the operating systems: A, B, C, D. The total number of instances of these resource types are (2, 4, 3, 3). The following matrices show a snapshot of the resource allocation at time T0.

	Allocation	**Max**
	A B C D	A B C D
P1	0 2 1 0	2 3 1 0
P2	0 1 0 1	0 1 2 2
P3	0 0 1 0	1 0 1 1
P4	1 1 0 0	1 2 1 1

(1) Is the operating system in a safe state? Why? [10 pts]

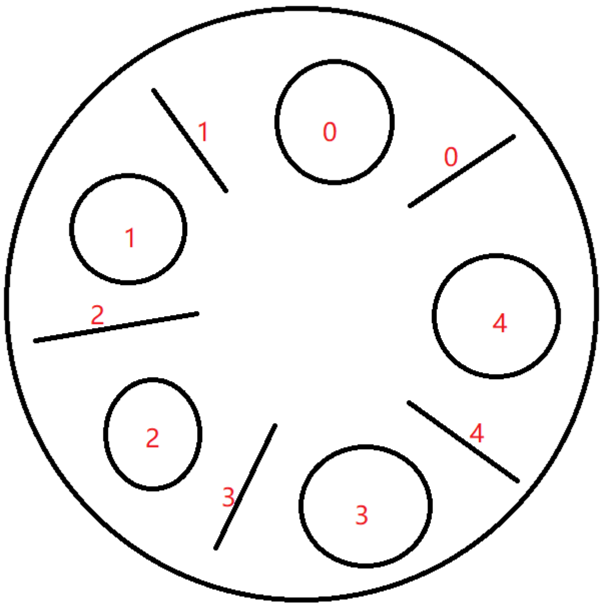
(2) If P4 requests (0,0,1,1), please run the Banker's algorithm to determine if the request should be granted. [10 pts]

(3) Let's assume P4's request was granted anyway (regardless of the answer to question 2). If then the processes request additional resources as follows, is the system in a deadlock state? Why? [10 pts]

	Request
	A B C D
P1	2 1 0 0
P2	0 0 1 0
P3	1 0 0 0
P4	0 1 0 0

2. Dining philosophers problem [40 pts]

Five silent philosophers sit at a round table with bowls of spaghetti. Forks are placed between each pair of adjacent philosophers.



Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when they have both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After an individual philosopher finishes eating, they need to put down both forks so that the forks become available to others. A philosopher can take the fork on their right or the one on their left as they become available, but cannot start eating before getting both forks.

Eating is not limited by the remaining amounts of spaghetti or stomach space; an infinite supply and an infinite demand are assumed.

Design a discipline of behavior (a concurrent algorithm) such that no philosopher will starve; i.e., each can forever continue to alternate between eating and thinking, assuming that no philosopher can know when others may want to eat or think.

At present, phy.c cannot always run to `100% done`. You need to modify the code in the specified area in phy.c(line 77-97) to make the code run to `100% done` .

Please give **two** different ideas to solve this problem.

Explain your design idea, and include screenshot of your code and the running result(run 5 times) in your report.

Referece:

[Dining philosophers problem](#)

[Docs (feishu.cn)](#)

3. The too much milk problem. [30 pts]

Now, your family bought a new fridge, it has space to store 2 bottles of milk. There are three people in your family who buy milk, dad, mom and grandfather. A person buys one bottle of milk at one time. You are responsible for drinking milk, one bottle at a time.

There are some rules:

1. Everyone doesn't know what everyone else is going to do.
2. When the refrigerator is empty, two people can buy milk at the same time
3. When the amount of milk in the refrigerator is more than two bottles, the refrigerator will stop working. (The program enters an infinite loop)
4. When there is no milk in the refrigerator, you can not open the refrigerator or the program enters an infinite loop.

At present, milk.c cannot run to completion. You need to use semaphore to solve this problem.

Explain your design idea, and include screenshot of your code and the running result in your report.