


Solution

National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Operating Systems	Course Code:	CS 2006
	Degree Program:	BS (CS/ SE/ DS)	Semester:	Fall 2023
	Exam Duration:	60 Minutes	Total Marks:	30
	Paper Date:	11-Nov-2023	Weight	15%
	Section:	ALL	Page(s):	5
	Exam Type:	Mid-2		

Student : Name: \_\_\_\_\_ Roll No. \_\_\_\_\_ Section: \_\_\_\_\_

Instruction/Notes: Avoid unnecessary explanation. Kindly write your information in the above mentioned space.  
Read ALL questions carefully and answer accordingly.

Question 1: What will be the output of the given code?

(5 Marks)

```
#include <pthread.h>
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void *printme(void *ip)
{
    int *i;
    i = (int *) ip;
    printf("Hi. I'm thread %d\n", *i);
    pid_t pid = fork();
    printf("Hello with %d\n", *i);
    return NULL;
}

int main()
{
    int i, vals[4];
    pthread_t tids[4];
    void *retval;

    for (i = 0; i < 4; i++) {
        vals[i] = i;
        pthread_create(&tids[i], NULL, printme, &vals[i]);
        pthread_join(tids[i], &retval);
        printf("Joined with tid %d\n", i);
    }
    return 0;
}
```

Answer:

Hi. I'm thread 0  
Hello with 0  
Hello with 0  
joined with tid 0  
Hi. I'm thread 1  
Hello with 1  
Hello with 1  
joined with tid 1  
Hi. I'm thread 2  
Hello with 2  
Hello with 2  
joined with tid 2  
Hi. I'm thread 3  
Hello with 3  
Hello with 3  
joined with tid 3

0295

## Question 2:

(10 Marks)

**Late-Night Cake.** A group of students are studying for an exam. The students can study only while eating cake. Each student executes the following loop:

```
while (true) {
    pick up a piece of cake;
    study while eating the cake;
}
```

If a student finds that the cake is gone, the student goes to sleep until another cake arrives. The first student to discover that the group is out of cake phones Layers at Johar town to order another cake before going to sleep. Each cake has  $C$  slices. Write code to synchronize the student threads and the cake delivery thread (Your solution should avoid deadlock and phone Layers (i.e., wake up the delivery thread) exactly once each time a cake is exhausted. No piece of cake may be consumed by more than one student.)

Also write the initial values of any variable and semaphore used in the code.

Hint: You need to write only two functions. One is for students and other is for Layers

10

$\text{min} = 2$   
 $\text{Avg} = 5$   
 $\text{max} = 10$

$\text{mutex} = 1$  3 binary semaphore  
 $\text{order-pizza} = 0$   
 $\text{deliver-pizza} = 0$   
 $\text{bool first} = \text{True}$ ,  $\text{have-pizza} = \text{False}$

$\text{cheerious}()$  layer  
 $\{$   
 $\text{do}$   
 $\{$   
 $\text{wait}(\text{mutex});$   
 $\text{wait}(\text{order-pizza});$   
 $\text{makePizza}();$   
 $\text{slices} = C;$   
 $\text{first} = \text{true};$   
 $\text{signal}(\text{deliver-pizza});$   
 $\text{signal}(\text{mutex});$   
 $\}$   
 $\text{while}(1);$   
 $\}$

$\text{int slices} = 5$  4  $(C=5)$   
 $\text{Students}()$   
 $\{$   
 $\text{do}$   
 $\{$   
 $\text{wait}(\text{mutex});$   
 $\text{while}(!\text{have-pizza})$   
 $\{$   
 $\text{if}(\text{slices} > 0)$   
 $\{$   
 $\text{slices} --;$   
 $\text{have-pizza} = \text{True};$   
 $\}$   
 $\text{else}$   
 $\{$   
 $\text{if}(\text{First})$   
 $\{$   
 $\text{signal}(\text{order-pizza});$   
 $\text{first} = \text{false};$   
 $\}$   
 $\text{wait}(\text{deliver-pizza});$   
 $\text{signal}(\text{mutex});$   
 $\text{study}(); \text{sleep}();$   
 $\text{have-pizza} = \text{false};$   
 $\}$   
 $\}$   
 $\}$

context switch

0295



Question 3:

(10 Marks)

Assume that there are 5 processes, P0 through P4, and 4 types of resources. At T0 we have the following system state:

Total resources (A=3 , B=17 , C=16 , D=12)

Process	Max	Allocation
	A, B, C, D	A, B, C, D
P0	0 2 1 0	0 1 1 0
P1	1 6 5 2	1 2 3 1
P2	2 3 6 6	1 3 6 5
P3	0 6 5 2	0 6 3 2
P4	0 6 5 6	0 0 1 4

Suppose the system is in a safe state, can the following requests be granted by using Banker's Algorithm, why or why not? Calculate any matrix or vector if required. Please also run the safety algorithm on each request if necessary.

(If any of the request is granted the next request will be checked on the **new state of the system**)

a. P1 requests (2,1,1,0)

b. P1 requests (0,2,1,0)

c. P3 requests (1,0,2,0)

① P1 request (2,1,1,0)

⇒ Is request<sub>1</sub> ≤ Need<sub>1</sub>??

(2,1,1,0) ≤ (0,4,2,1) false denied <sup>request</sup>

② P1 request (0,2,1,0)

⇒ Is request<sub>1</sub> ≤ Need<sub>1</sub>??

(0,2,1,0) ≤ (0,4,2,1) True

⇒ Is request<sub>1</sub> ≤ Available??

(0,2,1,0) ≤ (1,5,2,0) True.

Now apply safety Algorithm to find safe sequence.

3

0295

Processes	Max				Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	2	1	0	0	1	1	0	0	1	0	0	1	5	2	0
P1	1	6	5	2	1	2	3	1	0	2	1	0	6	3	0	0
P2	2	3	6	6	1	3	6	5	1	0	0	1	1	12	6	2
P3	0	6	5	2	0	6	3	2	0	8	2	0	1	12	7	6
P4	0	6	5	6	0	0	1	4	0	6	4	0	2	14	10	7

P0, P3, P4, P1, P2

(2, 1, 1, 0)

(P0, P1, P2, P3, P4)

safe sequence can be many.

(C) P3 requests (1, 0, 2, 0)

$\Rightarrow$  is  $request_3 \leq Need_3$ ??

$(1, 0, 2, 0) \leq (0, 0, 2, 0)$  false

request denied.

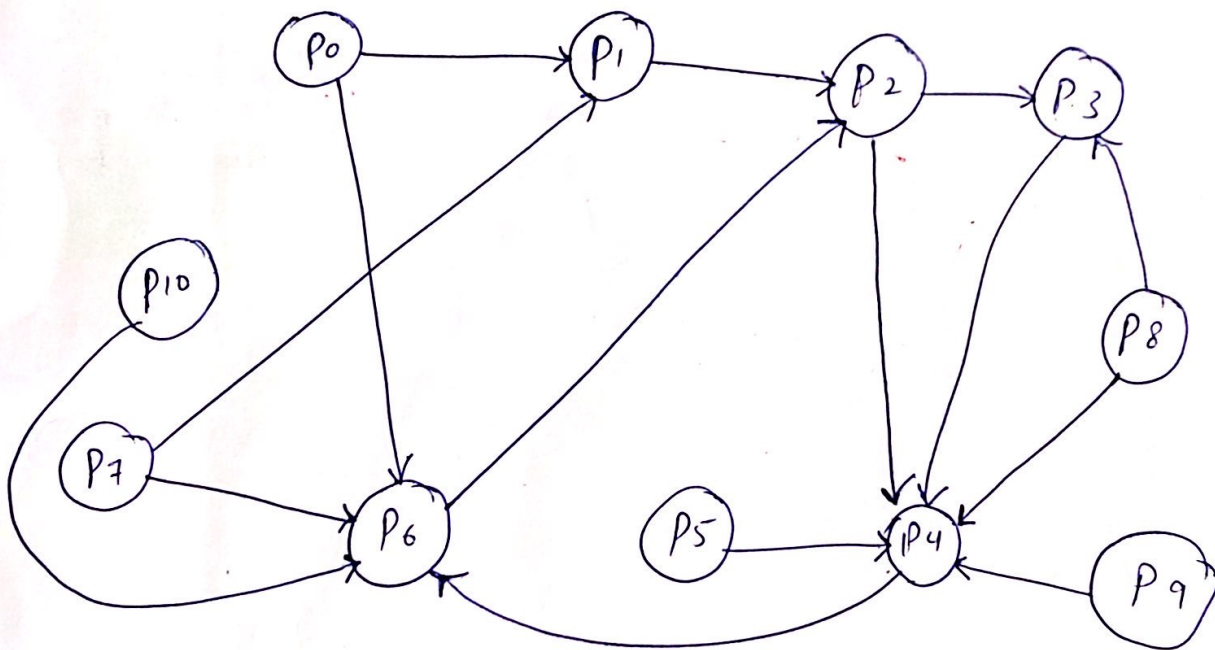
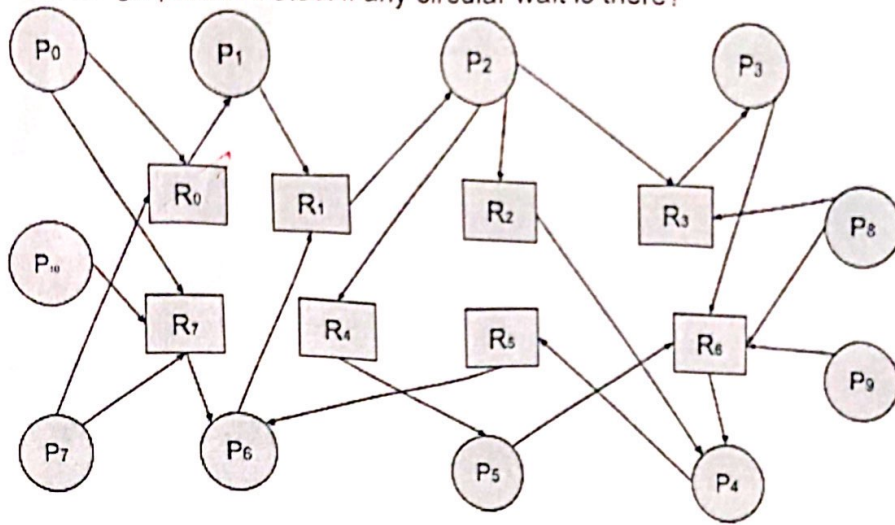
But there exist a safe sequence after running the Banker's algorithm.

(P0, P3, P4, P1, P2)

**Question 4:**

Create a wait-for-graph and detect if any circular wait is there?

(5 Marks)



Yes circular wait is there.