Sri Lanka Institute of Information Technology

B.Sc. Special Honours Degree/Diploma
in
Information Technology

Final Examination
Year 2, Semester 2 (2019)

IT2060 – Operating System and System Administration

Duration: 2 Hours

Instructions to Candidates:

- This paper contains 6 pages including the cover page.
- This paper has 4 questions with a total of 100 marks.
- Answer all questions in the booklet given.
- Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.

Question 1 (20 Marks)

a) Briefly explain the meaning of the following terms: (2 Marks)

I. Critical section of a program
II. Semaphore

b) Consider the following program which creates two threads. (11 Marks)

```
#include    "unpthread.h"
#define     NLOOP 5000
int         counter;
void  *doit(void *);
int   main(int argc, char **argv)
{
      pthread_t   tidA, tidB;
      Pthread_create(&tidA, NULL, &doit, NULL);
      Pthread_create(&tidB, NULL, &doit, NULL);
      /* wait for both threads to terminate */
      Pthread_join(tidA, NULL);
      Pthread_join(tidB, NULL);
      exit(0);
}
void * doit(void *vptr)
{
      int   i, val;
            for (i = 0; i < NLOOP; i++) {
            val = counter;
            printf("%d: %d\n", pthread_self(), val + 1);
            counter = val + 1;
      }
      return(NULL);
}
```

i. Which function **creates** the threads?

ii. Which **function** is executed by each thread?

iii. What is the **shared variable**?

iv. Find the **critical section** of the code?

v. Solve the above critical section problem using **semaphore**?

vi. What is the purpose of the function *pthread_join()?*

vii. What is the purpose of the function *pthread_self()?*

c) Consider the following code segment for the producer processes: (7 marks)

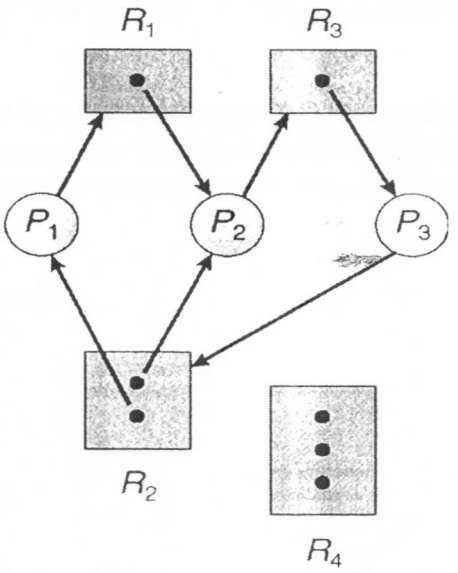
The structure of the producer process

```
do {
        ...
    /* produce an item in next_produced */
        ...
    wait(empty);
    wait(mutex);
        ...
    /* add next produced to the buffer */
        ...
    signal(mutex);
    signal(full);
} while (true);
```

i. List the initial value of the **mutex, empty and full** semaphores.
ii. Complete the consumer code for consumer process using the above semaphore.

Question 2 (20 Marks)

a) "*The following graph is in the **deadlock situation***". Briefly comment on the statement. (2 marks)



Source: Operating systemes by Silberschatz, Galvin and Gagne

b) What are the major differences between **deadlock and starvation**? (2 Marks)

c) List four necessary conditions to arise a **deadlock** in a system.

(2 marks)

d) Describe a method that breaks the **circular wait condition** of a deadlock prevention algorithm.

(2 marks)

e) Consider a system consisting **four similar resources** of the same type that are shared by **three processes**, each of which needs at most two resources. Show that the system is **deadlock free.**

(2 marks)

f) Deadlock recovery can be done by **preempting** some resources from one or more of the deadlocked processes. Briefly discuss at least **two factors** that must be considered when preempting a resource.

(2 marks)

g) For a state described in the following table:

Process	Current Allocation	Maximum Allocation
	R_1	R_1
P_1	2	4
P_2	2	3
P_3	4	10
P_4	3	8

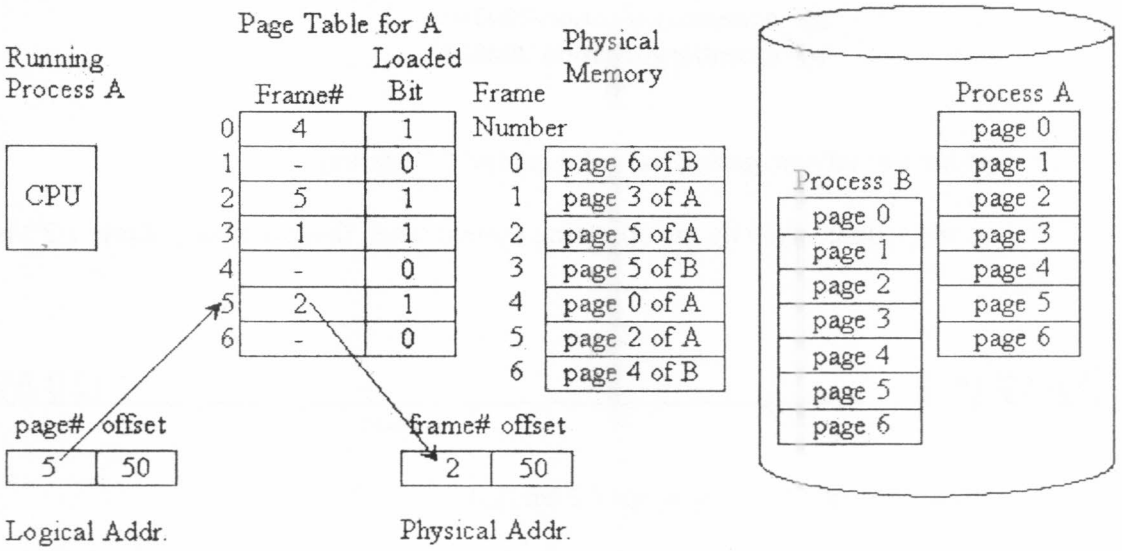
1. Is the **system safe** if the number of resources currently available is 4? Justify your answer.

2. In above table, for a given state and there are 4 resources available. If the Process 3 requests for **three resources** (in addition to the current resource allocation of 4 resources), should the request be granted? Justify your answer.

(8 marks)

Question 3 (20 Marks)

a) Consider the address-translation scheme shown below, where the page table is drawn only for the process A.



i. Where does the page table **store** in the computer system?

ii. Why do we need a **page table**?

iii. Draw the page table content for the process B?

iv. Briefly explain what is done by the CPU for the following conditions:

1) When the CPU generate the **logical address** where the page number is 1 for the process A

2) When the CPU generate the **logical address** where the page number is 2 for the process A

(10 Marks)

b) Assume a system uses **16-bit address** space (0 to 65535) and let a user program be allowed to access only addresses 0 to 20000. Given a page size of 2KB, answer the following.

(6 marks)

1. How many **pages** are there in the system?

2. How many entries of the page table (for that program) will be set as **valid pages** by the operating system?

3. Is there any **internal** fragmentation? Explain your answer. If so, compute the size of the fragmentation.

4. How do you reduce the **internal** fragmentation?

5. What would happened if the corresponding process requests for access to,

   a) a memory location 20479?
   b) a memory location 20480?

c) List the two advantages of **virtual** memory. (2 Marks)

d) List the actions taken by the operating system when there is a **page fault**. (2 Marks)

Question 4 (20 Marks)

a) Briefly explain the following terms (2 Marks)

   i. File Control Block
   ii. Index allocation tchnique

b) In a disk drive, Disk requests are generated for cylinders 7, 25, 3, 18, 6, and 75, in that order. It takes 5 msec per cylinder moved. Assuming the arm is **initially** at cylinder 20 (the **previous** request was at cylinder 5) for a disk with 100 cylinders (1 to 100), how much seek time is needed for:

   i. LOOK?

   ii. C-SCAN?

(8 Marks)

c) Consider a file currently consisting of 200 blocks. Assume that the file control block is already in memory. Calculate how many disk I/O operations are required for linked allocation strategy if:

   I. The **third** block from the beginning is removed.
   II. One block is added at the **beginning**.
   III. The **second** block is removed.

(6 Marks)

d) Briefly explain two **program threats**.

(4 marks)