University of Rajshahi

Department of Computer Science and Engineering

B. Sc. (Engg.) Part-III Even Semester Examination 2020

Course: CSE-3241 (Operating System and System Programming)

Full Marks: 52.5 Duration: 3(Three) Hours

Answer 06(Six) questions taking any 03(Three) questions from each part

Section-A

1. (a) What is kernel of an operating system? Explain the state transition diagram of process, 3.5

   (b) In the UNIX environment what happens in the following three cases as shown in Fig. 1. 5.25

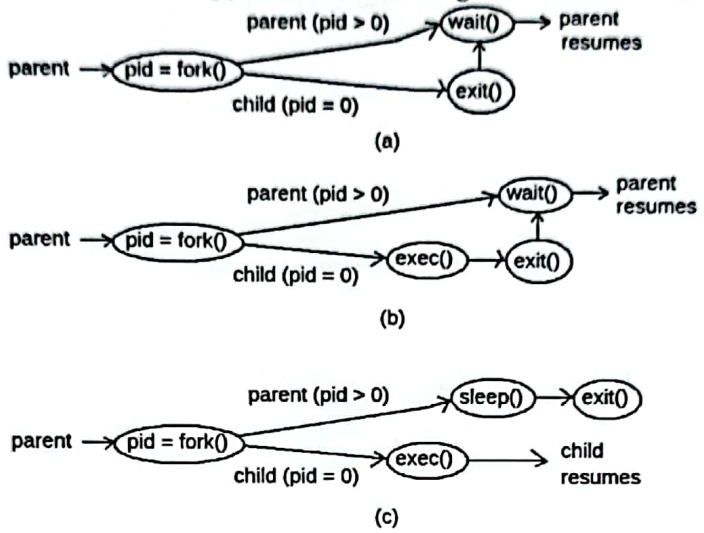


Fig 1

2. (a) From the resource allocation graphs in Fig. 2m, figure out in which case deadlock can occurs. 2.75
   Give proper explanation. Assume that:
   - Ri → Pj : Process Pj is holding an instance of resource type Ri.
   - Pj → Ri : Process Pj is waiting for an instance of resource type Ri.

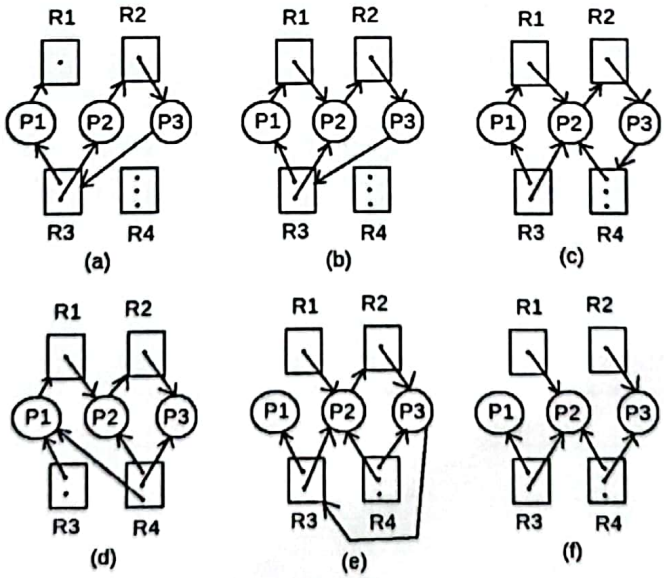


Fig. 2m

   (b) As process executes, it changes state. How many processes can be in the ***New***, ***Running***, ***Waiting***, ***Ready*** and ***Terminated*** state in a system having six cores per socket and two threads per core? Why? 2

(c) When the processor is switched from one process to another process in a system with a single processor: 4

i. Why does an operating system need to access process control block (PCB) ?
ii. Mark the place in Fig. 2n when Process P0, P1 and operating system are in the idle state?
iii. Will the processor be ever in the idle state? Why?
iv. Is it true that the idle time of the user process will increase if the number of user processes increases? Why?

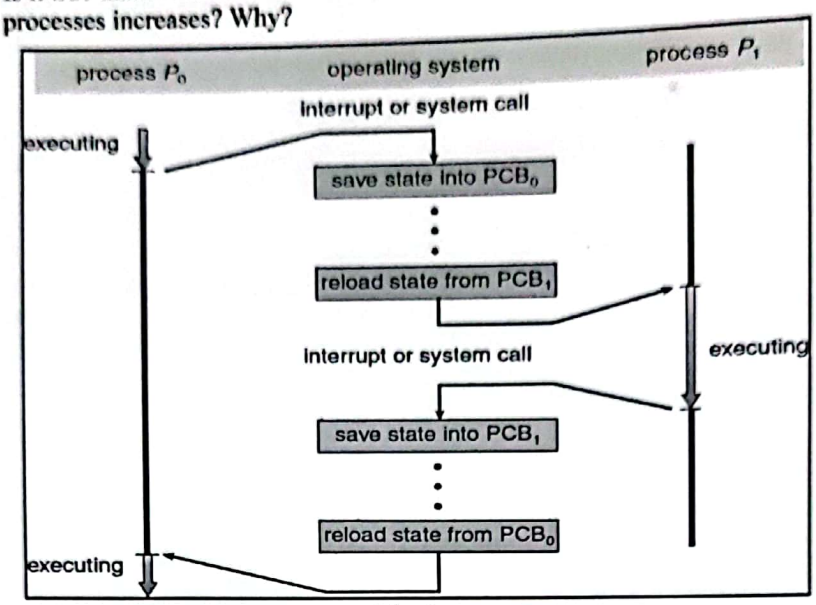


Fig. 2n

3. (a) How can a multi-threaded process be benefited by a multi-core system? 1

(b) Can a multi-threaded process with a global variable be in trouble? If 'yes', how can we solve it? 2

(c) Define thread. What do you know about the fork and exec system call in a multithreaded program? 2.75

(d) Say, a web server accepts requests for web pages, images, music, and so forth from multiple clients (perhaps thousands). Which of the architecture in Fig. 3 is better and why? 3

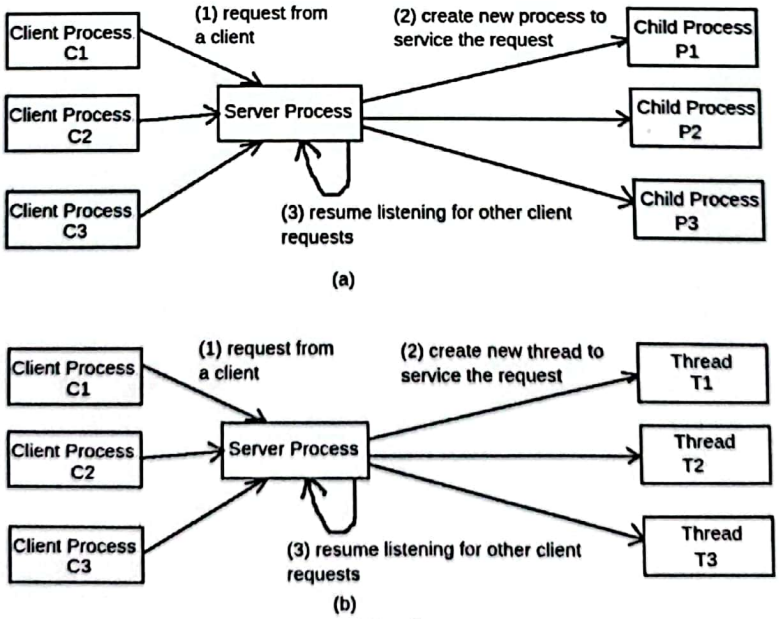


Fig. 3

4. (a) Which kind of CPU scheduling algorithm would you like to be implemented in your operating system? Why? 2.75

(b) How can an operating system avoid 'starvation' of a user process? 1

(c) How can two processes communicate with each other if 2
- there is an parent-child relationship between two processes?
- no parent-child relationship exists between two processes?

Defend your answer.

(d) Briefly discuss message passing techniques of inter process communication. 3

Section-B

5. (a) What are the pros-and-cons of the following two ways of allocating disk space for a file named 'Cat.jpg'? 4

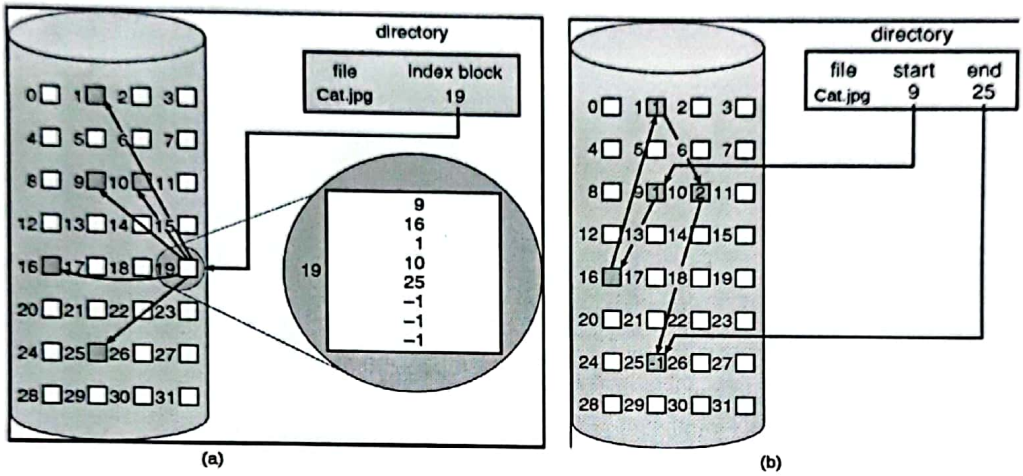


Fig. 5

(b) In Fig. 5(a), how large the index block should be? Defend your answer. 1.25

(c) The 'first-fit', 'best-fit' and 'worst-fit' strategies are the ones most commonly used to select a free hole from the set of available holes. Write pros and cons of these strategies. 2

(d) Say, there is a disk with 32 blocks among which block no. 2, 3, 4, 5, 16, 17, 18, 19, 20, 25, and 31 are free. Now can you allocate continuous blocks to three files, named 'Cat1.jpg', 'Cat2.jpg' and 'Cat3.jpg' which need 4, 5 and 2blocks? Why? 1.5

6. (a) Define virtual memory. What are the advantages of it? 3

(b) Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6. How many page fault would occur for the following page replacement algorithms: 3
- LRU replacement
- FIFO replacement
- Optimal replacement

Assume that there is a memory with three frames and all frames are initially empty so the first unique pages will cost one fault each.

(c) Say that four programs, named ProcessA, ProcessB, ProcessC and ProcessD, are written, so that when they will execute: 1.75
- ProcessA will create three named pipes (say, Pipe1, Pipe2 and Pipe3).
- ProcessB will write to Pipe1.
- ProcessC will write to Pipe2 at first and then read from Pipe1.
- ProcessD will write to Pipe3 at first and then read from Pipe2.

What would happen if you run processes by opening terminals in the following order: ProcessA at first, then ProcessB, then ProcessC and at last ProcessD.

(d) What is the difficulty of implementing optimal page replacement algorithm? 1

7. (a) Explain graphically what happens when we modify and delete one file which is softly linked to another file. 1.25

(b) What will be the size of a page table in a 3

- 32-bit operating system having 4KB page
- 32-bit operating system having 8KB page
- 64-bit operating system having 4KB page

(c) Does a two-level page table occupy less space than a single-level page table? Defend your answer. 2

(d) Say three documents are opened for editing in an office software. How many processes are running for the three documents? Will they be benefited by paging techniques? Defend your answer. 2.5

8. (a) Distinguish between: 4

- virtual memory and physical memory
- preemptive and non-preemptive scheduling
- swap-in and swap out
- process ID and thread ID

(b) Assume that a process is trying to access a page that is not brought into memory. Write down the steps written inside the circles of Fig. 8 for handling the page fault. 4.75

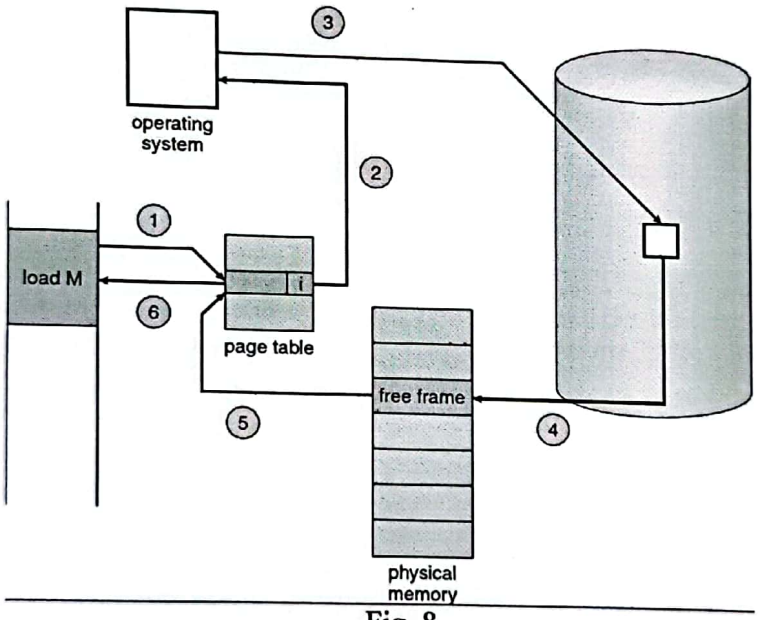


Fig. 8

University of Rajshahi
Dept. of Computer Science and Engineering
B.Sc.(Engg.), Part-III (Even Semester), Examination 2018
Course: CSE3241 (Operating System and System Programming)
Full Marks: 52.50 **Time: 3 hours**
[Answer any six questions taking three from each section]

Section A

1. a) What is an OS? Mention the advantages of parallel system and distributed system? 3.00
   b) What is an Application Programming Interface (API)? Discuss. 2.75
   c) Discuss different types of system calls. 3.00

2. a) The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: the higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function: 3.00

```
Priority = (recent CPU usage / 2) + base
```

where `base = 60` and `recent CPU usage` refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage for process *P*1 is 40, for process *P*2 is 18, and for process *P*3 is 10. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU-bound process?

   b) The following processes are being scheduled using a preemptive, priority-based, round-robin scheduling algorithm. 5.75

Process	Priority	Burst	Arrival
*P*1	8	15	0
*P*2	2	20	0
*P*3	4	20	20
*P*4	4	20	25
*P*5	5	5	45
*P*6	5	15	55

Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. The scheduler will execute the highest priority process. For processes with the same priority, a round-robin scheduler will be used with a time quantum of 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

   i. Show the scheduling order of the processes using a Gantt chart.
   ii. What is the turnaround time for each process?
   iii. What is the waiting time for each process?

3. a) What do you mean by critical section problem? 1.00
   b) Consider the following snapshot of a system: 5.00

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P_0	0 0 1 2	0 0 1 2	1 5 2 0
P_1	1 0 0 0	1 7 5 0	
P_2	1 3 5 4	2 3 5 6	
P_3	0 6 3 2	0 6 5 2	
P_4	0 0 1 4	0 6 5 6	

We claim that the system is currently in a safe state in the sequence (P_0, P_2, P_3, P_4, P_1). If a request for process $P_1 = (0, 4, 2, 0)$ arrives, can the request be granted immediately.

   c) Briefly explain the conditions to have deadlock in a system. 2.75

4. a) Differentiate between logical and physical address space. 2.00
   b) Explain the difference between external and internal fragmentations. 2.75
   c) What is segmentation? Briefly explain segmentation hardware. 4.00

Section B

5. a) The entire program is not needed to be in memory. Explain. 1.75

   b) What are the sequences to occur as a result of page fault? 3.00

   c) What is Bélády's anomaly? Discuss LRU page replacement algorithm with example. 4.00

6. a) Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150 and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is: 4.00

   2,069; 1,212; 2,296; 2,800; 544; 1,618; 356; 1,523; 4,965; 3,681

   Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

   i. SCAN
   ii. C-SCAN

   b) Consider the following I/O scenarios on a single-user PC: 4.75

   i. A mouse used with a graphical user interface
   ii. A disk drive containing user files
   iii. A graphics card with direct bus connection, accessible through memory-mapped I/O

   For each of these scenarios, would you design the operating system to use buffering, spooling, caching, or a combination? Would you use polled I/O or interrupt-driven I/O? Give reasons for your choices.

7. a) An operating system supports a paged virtual memory. The central processor has a cycle time of 1 microsecond. It costs an additional 1 microsecond to access a page other than the current one. Pages have 1,000 words, and the paging device is a drum that rotates at 3,000 revolutions per minute and transfers 1 million words per second. The following statistical measurements were obtained from the system: 4.00

   - One per cent of all instructions executed accessed a page other than the current page.
   - Of the instructions that accessed another page, 80 percent accessed a page already in memory.
   - When a new page was required, the replaced page was modified 50 percent of the time.

   Calculate the effective instruction time on this system, assuming that the system is running one process only and that the processor is idle during drum transfers.

   b) Consider the two-dimensional array A: 4.75

   ```
   int A[][]= new int[100][100];
   ```

   Where `A[0][0]` is at location 200 in a paged memory system with pages of size 200. A small process that manipulates the matrix resides in page 0 (locations 0 to 199). Thus, every instruction fetch will be from page 0.

   For three page frames, how many page faults are generated by the following array-initialization loops? Use LRU replacement, and assume that page frame 1 contains the process and the other two are initially empty.

   i.
   ```
   for(int j=0;j<100;j++)
       for (int i = 0; i < 100; i++)
           A[i][j] = 0;
   ```

   ii.
   ```
   for(int i=0;i<100;i++)
       for (int j = 0; j < 100; j++)
           A[i][j] = 0;
   ```

8. a) What is kernel? Is Linux a kernel or an operating system? 2.00

   b) Briefly discuss message passing technique of IPC 4.00

   c) Differentiate between user level thread and kernel level thread. 2.75

University of Rajshahi
Dept. of Computer Science and Engineering
B. Sc. Engg., Part-III (Even Semester), Examination 2017
CSE3241: Operating System and System Programming
Full Marks: 52.5 Time: 3 Hours

Answer three questions from each section

Section-A

1 (a) Define operating system. What are the services of operating system? 3.75

(b) What is Process control Block (PCB)? Discuss the information that a PCB contains with a specific process. 4

(c) What is kernel? 1

2 (a) What is the difference between process and program? 1.5

(b) Define thread. What do you know about the fork and exec system call in a multithreaded program? 2.5

(c) What do you mean by process states? Discuss the process states and transitions with diagram. 3

(d) Distinguish between I/O bound and CPU bound process. 1.75

3 (a) 2

```
#include <stdio.h>
#include <unistd.h>
int main(){
        fork();
        fork();
        return 0;
        fork();
        fork();
        return 0;
}
```

Including the parent process, how many processes will be created by the above program? Explain by drawing a process tree.

(b) 6.75

```
#include <stdio.h>
#include <unistd.h>
int main(){
        int a = 5;
        pid_t CID, myPID;
        myPID = getpid();
        CID = fork();
        if(CID == 0){
                printf("My PID: %d and my parent's PID: %d\n", myPID, getppid());
                a = a – 5;
                printf("a = %d\n", a);
        }
        else{
                printf("My PID: %d and my parent's PID: %d\n", myPID, getppid());
                a = a + 5;
                printf("a = %d\n", a);
                while(1);
        }
```

```
    return 0;
}
```

Answer the following questions with proper explanations:

1. What will be the output of the above program?
2. What would happen, if:
   1. ***while(1);*** statement is replaced by ***wait(NULL);*** in the ***else*** block?
   2. ***while(1);*** statement is in the ***if*** block after the last ***printf()*** statement, instead of in the ***else*** block?
   3. there is no ***while(1);*** statement in any block?
   4. ***while(1);*** statement is before ***return 0;*** statement instead of in the ***else*** block?

4 (a) Assume that two named pipe, **pipe1** and **pipe2**, are created so that two processes, **writer1** and **writer2**, can exchange their process ID via pipes. Process **writer1** prefers to access **pipe1** at first, then **pipe2**, whereas Process **writer2** prefers to access pipe2 at first, then pipe1. 5.75

1. Is it problematic to run two processes of the following two programs concurrently? If yes, how can you solve it?
2. If both processes access pipes in the same order (e.g., **pipe1** at first, then **pipe2**), what would happen?

---------- writer1.c ----------

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

#define BUF_SZ 50
int main(){
        int fd;
        char msg[BUF_SZ];
        fd = open("pipe1", O_RDONLY);
        read(fd, msg, BUF_SZ);
        write(1, msg, strlen(msg) + 1);
        close(fd);
        sprintf(msg, "Writer1: %d\n", getpid());
        fd = open("pipe2", O_WRONLY);
        write(fd, msg, strlen(msg) + 1);
        close(fd);
        return 0;
}
```

---------- writer2.c ----------

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

#define BUF_SZ 50
int main(){
        int fd;
        char msg[BUF_SZ];
        fd = open("pipe2", O_RDONLY);
        read(fd, msg, BUF_SZ);
        write(1, msg, strlen(msg) + 1);
        close(fd);
        sprintf(msg, "Writer2: %d\n",
getpid());
        fd = open("pipe1", O_WRONLY);
        write(fd, msg, strlen(msg) + 1);
        close(fd);
        return 0;
}
```

(b) What would be the output if the process of the following program runs? 3

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]){
        int x = 5;
        printf("Hello, I am Process-%d\n----------\n", getpid());
        execlp("/bin/cat", "/bin/cat", "execlpExample.c", NULL);
        printf("%d + 2 : %d\n", x, x + 2);
        return 0;
}
```

Section-B

5 (a) Consider the following page reference string: 6

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

How many page faults would occur for the following page replacement algorithms, assuming three and seven frames? Remember that all frames are initially empty, so the first unique pages will cost one page fault.

- LRU replacement
- FIFO replacement
- Optimal replacement

(b) Consider a 100-by-100 matrix, A, where A[0][0] is at location 200 in a paged memory system with pages of size 200. A small process that manipulates this matrix resides in page 0 (locations 0 to 199). Thus every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array-initialization loops, using LRU replacement and assuming that page frame, i.e., Frame-0, contains the process and the other two page frames (i.e. Frame-1 and Frame-2) are initially empty? 2.75

(a)
```
for (int j = 0; j < 100; j++)
        for (int i = 0; i < 100; i++)
                A[i][j] = 0;
```

(b)
```
for (int i = 0; i < 100; i++)
        for (int j = 0; j < 100; j++)
                A[i][j] = 0;
```

6 (a) Given the following five memory partitions: 3.25

100 KB, 500 KB, 200 KB, 300 KB and 600 KB (in order).

i. How would the first-fit, best-fit and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)?

ii. Which algorithm makes the most efficient use of memory?

(b) Consider the following segment table: 2.5

Segment	Base	Length
0	219	600
1	2300	14
2		
90	100	
3	1327	580
4	1952	96

What are the physical addresses of the following logical addresses, in {s, d} form where s means segment number and d means offset?

(a) {0, 430}, (b) 1,10, (c) {2, 500}, (d) {3, 400}, and (e) {4, 112}

(c) Suppose there is 64 bytes physical memory in a system, each process occupies 16 bytes in logical space and the frame/page size is 4 bytes. If processes P1, P2 and P3 occupy {5, 6, 1, 2}, {7, 0, 3, 4} and {15,14, 10, 11} frames respectively, then draw the page table(s) and inverted page table for those three processes. 3

7 (a) Consider the following snapshot of a system: 4.75

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	2	1	5	2	0
P2	1	0	0	0	1	7	5	0				
P3	1	3	5	4	2	3	5	6				
P4	0	6	3	2	0	6	5	2				
P5	0	0	1	4	0	6	5	6				

Draw the resource-allocation graph. Answer the following questions using the Banker's algorithm:

i. Is the system in a safe state?

ii. If a request from P1 arrives for (0, 4, 2, 0), can the request be granted immediately? If yes, will the system be still in the safe state?

(b) Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system? 2

(c) Consider a disk has 32 blocks each of which is 4 KB in size. If blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18, 25, 27 are free and the rest of blocks are allocated, then: 2

i. What would be the free space bit map vector?

ii. In order to store this bit map, how much space do we need in memory?

8 (a) Consider the following set of processes with the length of the CPU burst given in milliseconds: 7

Process	Burst Time	Niceness
P_1	10	5
P_2	1	-5
P_3	2	5
P_4	1	15
P_5	5	0

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0. The CPU scheduler enforces an inverse relationship between niceness value and priority.

i. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms:

- FCFS
- preemptive SJF
- non-preemptive SJF
- non-preemptive priority
- preemptive priority
- RR(quantum = 4)

ii. What is the total waiting time of each processes for each of these scheduling algorithms?

iii. Which of the algorithms results in the minimum average waiting time (over all processes)?

(b) What happens when a user application requests a service from the operating system? 1.75

University of Rajshahi
Department of Computer Science and Engineering
B. Sc. Engg. Part3 Even Semester, Examination-2016
Course: CSE-3241 (Operating System and System Programming)

Full Marks: 52.5 **Time: 3 Hours**

[Answer three questions from each part]

Part A

1. a) What types of information does a Process Control Block hold? Describe. 3
   b) What is 'kernel 'of an operating system? Explain the state transition diagram of process. 3
   c) Explain with a queuing diagram how a process migrates from one queue to another queue during its life cycle. 2.75

2. a) If a law is passed in the parliament as " when two trains approach each other at a crossing, both shall come to a full stop and neither shall start up again until the other has gone." What will happen? Explain the situation in the context of using system resources by processes in a multiprogramming environment. 2
   b) In the resource request algorithm, if $Request_i > Need_i$, an error condition is raised. Why? 1
   c) If a system has a multiple numbers of instances of a resource type, explain how you could detect the system whether it is deadlocked or not. 4
   d) What factors should be considered to select a victim for resource preemption? 1.75

3. a) Briefly discuss the use of job queue, ready queue and device queue. 3
   b) Consider the following processes and their corresponding CPU burst time in ms: 5.75

Process	Burst Time
P_1	9
P_2	12
P_3	4
P_4	13
P_5	7

and quantum=10ms. The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4 and P_5 all at time 0. Draw the Gantt chart and find out the waiting time and turnaround time for SJF and RR algorithm.

4. a) What do you meant by 'busy waiting'? Explain how it occurs. 2
   b) 'Race condition' produces uncertain outcomes. Explain. 2
   c) Discuss the implementation of semaphore. 4.75

Part B

5. a) What do you mean by external and internal fragmentation? 2
   b) When do we need hierarchical paging? Explain the address translation for two level 32-bit paging architecture. 3.25
   c) Explain how the paging environment can be used for memory protection. 3.5

6. a) Define virtual memory. What are the advantages of it? 1.75
   b) When does the system require a page to be replaced? 1

c) What is the difficulty of implementing optimal page replacement algorithm? 2
d) Explain the following LRU approximation algorithms for page replacement. 4
   (i). Additional-reference-bit algorithm
   (ii). Second-chance algorithm

7. a) What is a file? What are the common attributes maintained for a file? 3
   b) Describe the sequential access and direct access methods for file. 3.75
   c) What is the limitation of single level directory structure? Explain. 2

8. a) What is the importance of partitioning the disk? 2
   b) Write short notes on the following (any three): 6.75
   (i). Time sharing system.
   (ii). Real-time system.
   (iii). Segmentation.
   (iv). Swapping.
   (v). Binary Semaphore.
   (vi). File System mounting.

Rajshahi University
Department of Computer Science and Engineering
B.Sc(Engg) Part-III, Even Semester Examination-2015
Course: CSE-3241 (Operating System and System Programming)
Full Marks: 52.5 Time: 3 hours

Answer any Six (06) questions taking at least three from both the groups.

PART-A

1. a) Define Operating system. 'Operating system acts as the government'- explain. 4.5
   b) What is context switch? Explain how the CPU switches between the processes while context switching. 3
   c) What is the difference between a ready and a waiting process? 1.25

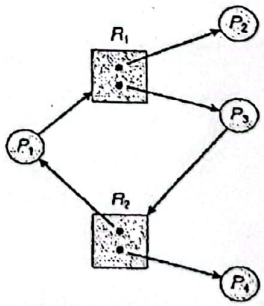
2. a) Define IO-bound process and CPU-bound process. Explain why it is important to make a good processes mix to maximize the CPU utilization. 4
   b) Explain how the message passing system can be used for interprocess communication. 4.75

3. a) Differentiate preemptive and non-preemptive scheduling techniques. 1.5
   b) Illustrate the criteria used for comparing CPU scheduling algorithms. 2
   c) Consider the following processes with CPU burst P1=10ms, P2=29ms, P3=3ms, P4=7ms, P5=12ms and quantum=10ms. Draw the Gantt chart; find the average waiting time and average turnaround time for FCPS, SJF and RR scheduling algorithms. 5.25

4. a) What is the difference between asymmetric and symmetric multiprocessing techniques? 2
   b) What do you mean by 'busy waiting'? Give an example explaining it. 3
   c) *'Race condition'* produces uncertain outcomes. Explain with a suitable example. 3.75

PART-B

5. a) Define dead lock. Give a real life example to explain dead lock. 2
   b) *"A cycle in the resource allocation graph is necessary, but not sufficient condition for happening deadlock."* Explain it from the adjacent graph. 3



   c) Define safe sequence. Write down an algorithm to find whether a sequence is safe or not. 3.75

6.	a)	Define logical and physical addresses.	2
	b)	Discuss the relative advantages and disadvantages of 'first-fit', 'best-fit' and 'worst-fit' algorithms.	3
	c)	*"In paging memory management scheme, external fragmentation problem is solved, but internal fragmentation still exists."* Explain with examples.	2
	d)	What is TLB? What is the benefit of using TLB before going to the page table?	1.75
7.	a)	'The entire program is not needed to be in main memory'-explain.	2
	b)	Mention the steps in handling a page-fault.	3.75
	c)	Discuss Least Recently Used (LRU) page replacement algorithm.	3
8.	a)	Write the purposes of the followings: i) Boot control block ii) Volume control block	2
	b)	What is the difference between a raw disk and a formatted disk?	2
	c)	Describe the indexed allocation method for directory implementation.	2
	d)	Explain how the bit vector can be used to manage free-space onto the disks. How you can compute the block number of the disk?	2.75

University of Rajshahi
Department of Computer Science and Engineering
B.Sc(Engg.) Part-III, Even semester Examination-2014
Course: CSE-3241(Operating System and System Programming)
Full Marks: 52.5 Time: 3 hours

Answer any Six (06) questions taking at least Three (03) from each part.

PART-A

1.(a) What is 'Kernel' of an operating system? Explain the state transition diagram of process. 4
(b) What are the main functions of an operating system? How does it govern the entire system? 3
(c) Differentiate CPU-bound process and I/O-bound process. 1.75

2. (a) With the help of a state transition diagram, explain various states of a process. 2
(b) Explain the Round Robin CPU scheduling algorithm with an example. 2.75
(c) Explain why real-time systems require a pre-emptive scheduling policy. 1
(d) Consider the following set of processes, with the length of the CPU-burst time in given ms 3

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4 and P5 all at time 0.
a. Draw four Gants charts illustrating the execution of these processes using FCFS, SJF and RR (quantum=1) scheduling.
b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

3. (a) What is the difference between asymmetric and symmetric multiprocessing techniques? 2
(b) Explain the different migration techniques for load balancing between different processors in multiprocessing environment. 3
(c) What do you mean by 'race condition'? Explain how it occurs in case of cooperating multiprocessing environment. 2.5
(d) What are the main drawbacks of hardware solutions for synchronization problems? 1.25

4. (a) If a law is passed in the parliament as *"When two trains approach each other at a crossing, both shall come to a full stop and neither shall start up again until the other has gone."* What will happen? Explain the situation in the context of using system resources by processes in a multiprogramming environment. 2
(b) *What do you mean by safe sequence in resource allocation state? Apply the banker's algorithm to find out the safe sequence from the snapshot of a system as given below-* 5

Process↓	Allocation	Max	Available
	A B C D	A B C D	A B C D
P_0	0 0 1 2	0 0 1 2	1 5 2 0
P_1	1 0 0 0	1 7 5 0	
P_2	1 3 5 4	2 3 5 6	
P_3	0 6 3 2	0 6 5 2	
P_4	0 0 1 4	0 6 5 6	

Where A, B, C and D are system resources, all others carry their conventional meanings.

(c) What are advantages of aborting one process at a time until the deadlock cycle is eliminated to recover a system from deadlock situation? 1.75

PART-B

5. (a) Explain the difference between logical and physical addresses. 2
(b) What is external fragmentation? Explain any one solution to this problem? 2
(c) Describe the segmentation hardware with diagrams. 3
(d) Consider the following segment table: 1.75

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?
(i) 0, 430 (ii) 2, 500 (ii) 3, 400

6. (a) Define virtual memory. What are the advantages of it? 2
(b) What is page fault? Explain how the system handles a page fault. 3
(c) Explain the 'second chance' and 'enhanced second chance' algorithms as the approximations of LRU page replacement algorithm. 3
(d) Discuss the advantages of page buffering technique. 0.75

7. (a) What is a file? What are the common attributes maintained for a file? 3
(b) Describe the sequential access and direct access methods for file. 4
(c) What is the limitation of single level directory structure? Explain. 1.75

8. (a) Define 'boot control block' and 'volume control block'. 2
(b) What is FCB? Explain its importance and uses for implementing a file system. 2.25
(c) Explain the different approaches for implementing indexed allocation technique for storage allocation. 3
(d) Explain how a bit vector can be used to manage free space in storage. 1.5