Date of Examination: 20/05/2019

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Semester Final Examination: Fall 2018

Year: 3rd Semester: 2nd

Course Number: **CSE3213**

Course Name: **Operating System**

Time: 3 (Three) hours Full Marks: 70

*[There are seven questions carrying a total of 14 marks each. Answer **any five** questions. Marks allotted are indicated in the right margin.]*

1. a) What are the basic functionalities of an Operating System? 'I/O devices and the CPU can execute concurrently'. Explain. [4]

   b) What is the importance of a bootstrap program to an Operating System? Explain the Microkernel based Operating System architecture. [5]

   c) What is a daemon process? Explain the Type-1 and Type-2 hypervisor. [5]

2. a) What is a process-control-block? How can an Operating System calculate CPU utilization with respect to the number of processes in the ready queue? [4]

   b) Provide a race-condition free and non-semaphore based solution for the producer-consumer problem. What do you understand by Synchronous Multiprocessor Scheduling? [5]

   c) How do two or more processes sharing a common memory area can get into a race-condition? Using the semaphore data structure of an Operating System, provide a race-condition free solution for the Sleeping Barber problem. If there are three barbers in the system and four customers in the waiting chairs, what will be the state of the system? [5]

3. a) Explain the 'Multi-level Feedback Queue' process scheduling algorithm with an example. [4]

   b) Explain the procedure to approximate CPU burst-time of batch processes in the case of 'Shortest Job First' process scheduling algorithm. How can an Operating System perform load balancing of jobs in a multiple-processor system? [5]

   c) Given the following process information, find the average turn-around time using the *Round-Robin* process scheduling algorithm. Assume that, time quantum **q=3.** [5]

Process	Arrival time	CPU burst time
P1	7	12
P2	5	11
P3	6	9
P5	0	17
P4	14	13

4. a) What is meant by a safe-state? Show the deadlock detection mechanism with any arbitrary Resource Allocation graph having at-least two deadlock situation. [4]

b) Explain the inode data structure used in a Unix-style file-system. Assuming a 4KB data block and a 4 byte pointer size, what will be the maximum allowable file size? [5]

c) How can a system recover from a deadlock situation? Explain the 'File Allocation Table' file-system architecture. [5]

5. a) Requests for read operations from the disk blocks residing in the following cylinders are given below. [4]

12 6 13 1 6 61 19 18 47 60 2 120 112 13 7 212

Compare the following disk-arm scheduling algorithms in terms of the total number of cylinder movement.

i) Shortest Seek Time First.
ii) SCAN.

Assume that, the cylinders are numbered from 0 to 400 and the disk head is currently positioned over the cylinder number 22 and moving towards the lower numbered cylinders.

b) Why is a thread called a Light Weight Process? Explain the Kernel-level thread model with necessary figures. [5]

c) What do you understand by cylinder skew in a disk management system? How does an Operating System deal with a disk bad sector? [5]

6. a) Explain the two level memory-page-table construction using the following data. [4]
Page/Frame size: 2048 byte; Number of pages: 32; Number of memory frames: 64.

b) Compare the 'Least Recently Used' and 'Optimal' memory page replacement algorithms in terms of the number of page faults using the following page requests by a process P1. Assume that, there are only three page frames in the memory. [5]

Page requests: 10, 15, 8, 12, 0, 1, 7, 11, 10, 4, 7, 9, 5, 11, 7, 11, 2, 1, 10, 7, 9.

c) What are the different address binding techniques used by the memory management module of an Operating System? What are the differences between global and local page replacement technique? [5]

7. a) What are the different ways an Operating System can keep track of the free /occupied memory spaces? [4]

b) Define Internal and External memory fragmentation using examples. Explain the memory address translation process using Memory Segmentation. [5]

c) Explain the use of Translation Look-aside Buffer during the logical to physical address translation process. Draw the diagram for paging hardware with TLB and virtual memory. [5]

Date of Examination : 06/09/2018

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering

Semester Final Examination: Spring 2018

Year: 3rd Semester: 2nd

Course Number: CSE3213

Course Name: Operating System

Time: 3 (Three) hours Full Marks: 70

[There are seven questions carrying a total of 14 marks each. Answer any five questions.
Marks allotted are indicated in the right margin.]

1. a) What are major functions of an Operating System? What are the necessities of System calls? 4
   b) What is the advantage of modular-monolithic Operating System architecture? Distinguish the differences between Type-1 and Type-2 hypervisors in a virtual machine Operating System. 5
   c) Explain the Semaphore data structure and its operation with any example. How do two or more processes sharing a common memory area can get into a race condition situation? 5

2. a) Explain the lifecycle of a process. What is a pre-emptive process scheduling? 4
   b) What should be scheduling criteria for batch, interactive, and real-time systems? What is the difference between symmetric and asymmetric multiprocessing used in a multi-processor process scheduling system? 5
   c) Find the average turnaround time for the following problem using Round Robin process scheduling algorithm with time quantum $q = 4$ in a single microprocessor based system 5

Process	Arrival Time	CPU Time
P2	0	9
P0	18	15
P1	8	7
P3	4	18
P4	16	5
P5	12	5
P6	22	7

3. a) Without using the Semaphore data structure, provide a race-condition free solution for the standard producer-consumer problem with n number of producers and m number of consumers. What could be the possible faults in your solution? 4
   b) Explain the Shortest remaining time CPU Scheduling algorithm with any example. How does an Operating System estimate the CPU burst time for the above algorithm? 5
   c) What are the advantages of using threaded applications? Discuss the Kernel level thread model with necessary figures. 5

4. a) Requests for read operations from the disk blocks residing in the cylinders are: 4
   12 16 3 1 6 9 89 75 37 62 12 11 10 13 61 220.
   Show the sequence of requests served using the SCAN and LOOK disk scheduling algorithms with the total number of cylinder movements. Assume that the cylinders are numbered from 0 to 300 and the disk head is currently positioned over the cylinder number 40. Moreover, the disk head is currently moving towards the higher numbered cylinders.

b) Explain the deadlock detection algorithm from a Resource Allocation Graph considering single copy of each type of available resources. How can a deadlock be recovered by an Operating System? 5

c) Explain the safe and unsafe state. Using the following information, will it be safe to serve the request < 2, 1, 3 > by the process P5? 5

Resources= {A 23, C 20, D 19}

	Allocation			Max		
	A	C	D	A	C	D
P1	1	2	1	7	5	3
P2	3	5	3	5	8	5
P3	3	0	7	9	2	8
P4	4	2	0	8	8	6
P5	5	1	0	10	4	4
P6	1	4	1	4	6	3
P7	1	1	1	2	2	2

5. a) Compare the following memory allocation algorithms in terms of external fragmentation for the variable length contiguous memory allocation. 4

i) Best fit. ii) First fit. iii) Worst fit.

Use the following data for the comparison.

Memory holes: 20, 61, 56, 77, 44, 190, 10; Memory allocation requests : 55, 9, 19, 22, 100.

b) How does an Operating System restrict invalid memory access with memory abstraction? Explain the data structure of the page table used by a paging supported Operating System. 5

c) What is the difference between a virtual and a physical memory address? Translate the following address issued by a process to its corresponding physical memory address with the following information. Assume any physical location in the RAM for the corresponding virtual page. The symbols have their usual meaning. 5

Page frame size: 16 KB
RAM size: 8 MB
Address space size of the process: 4 MB
The process requests data from it 300th page with the offset being 21611.

6. a) What is thrashing in a paging supported system? Explain the Enhanced Second Chance page replacement algorithm. 4

b) What is a Translation Look-aside Buffer (TLB)? How does an Operating System perform address translation with a TLB and virtual memory? 5

c) What happens when a page is replaced with another page due to insufficient space in RAM. Given the following page requests by a process P, allocate them in the memory using the Least Recently Used page replacement algorithm. Show the memory status using a stack based implementation. Assume that, there are only three page frames in the memory. 5

Memory page requests: 10, 5, 4, 5, 0, 10, 4, 9, 10, 3, 5, 8, 7, 0, 7, 1, 5, 4, 8, 1.

7. a) How does an Operating System calculate CPU utilization in terms of number of processes in the ready queue? How much performance gain can be achieved while a single threaded application is run in a multi-core system compared to a single core system with exactly same CPU clock speed? 4

b) What is the difference between a program and a process? Why is a Process Control Block needed for each process in the system? Explain. 5

c) What do you understand by cylinder skew? How can an Operating System detect and replace a disk bad sector? 5

Date: 25/09/17

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

3rd Year, 2nd Semester, Final Examination, Spring 2017

Course No: **CSE3213** Course Title: **Operating System**

Time: 3 hours Full Marks : 70

[There are seven questions carrying a total of 14 marks each. Answer any five questions.
Marks allotted are indicated in the right margin.]

1. a) Show with necessary diagrams the method of accessing hardware resources by a user process. 4

   b) Explain the *Type-1* and *Type-2 hypervisors* in a virtual machine Operating System. 4

   c) Discuss the *Monolithic* and *Microkernel* Operating System architecture with necessary diagrams. 6

2. a) What are the ways an Operating System can deal with a *process-deadlock* situation? 4

   b) What is the difference between a *user-level* and a *kernel-level thread*? Discuss the kernel-level thread model with necessary diagrams. 4

   c) You are given the following information. Use the *Banker's algorithm* to determine whether the system is currently in *safe state*. The symbols have their usual meaning. 6

	X	Y	Z
Resources=	20,	19,	16}

	Allocation			Max		
	X	Y	Z	X	Y	Z
P1	1	1	1	7	5	6
P2	2	3	3	5	8	5
P3	3	0	5	9	2	8
P4	2	2	0	8	8	6
P5	3	1	0	14	4	5
P6	1	2	1	4	6	3
P7	1	1	1	2	9	2

Now, Process P1 requests <X=2, Y=1, Z=1>; Will it be safe to serve this request ?

3. a) What do you understand by *Dynamic Loading* and *Dynamic Linking*? Show the virtual address and its corresponding physical memory address with the following information. The symbols have their usual meaning. 4

Page frame size : 16 KB

RAM size: 10MB

Address space size of the process : 15 MB

The process requests data from it 100th page with the offset being 15300.

b) What are the differences between *global* and *local page replacement* technique? 4

c) Compare the *Least Recently Used* and *FIFO* memory page replacement algorithms in terms of the number of page faults using the following page requests by a process P1. Assume that, there are only three page frames in the memory. 6

Page requests: 0, 5, 4, 2, 0, 1, 7, 9, 11, 5, 0, 4, 7, 0, 5, 11, 3, 11, 2, 1.

4. a) Draw the diagram for an address translation process in any virtual memory system with a Translation Look-aside Buffer. 4

b) Explain the two level memory-page-table construction using any example. 4

c) What are the different ways an Operating System can keep track of the free memory spaces? Using the following data, calculate the total amount of external fragmentation separately for the *first fit* and *best fit* memory allocation techniques. Assume that, space allocation is variable length and contiguous. 6

Memory holes : 10 5 2 22 11 7 8

Memory requests : 5 6 11 6 6 10

5. a) Explain the *File Allocation Table* structure used by an Operating System. 4

b) With an 8KB data sector size, what could be the maximum file size using a *Linux inode data structure*? Assume that, each data pointer takes 8 bytes of space. 4

c) What do you understand by *sector interleaving*? Requests for read operations from the data blocks residing in the following cylinders are given below. 6

2 6 3 1 52 19 80 70 37 62 12 29 10 390 79 310,

Show the sequence of requests served using the *Shortest Seek time first* and *C-Look* disk scheduling algorithms and calculate the total number of cylinder movements. Assume that, the cylinders are numbered from 0 to 400 and the disk head is currently positioned over the cylinder 51. The disk head is moving towards the higher numbered cylinders.

In the case of *C-Look* algorithm, assume that, the disk head serves any cylinder request while moving from higher to lower numbered cylinders only.

6. a) What is the function of the *dispatcher module* of an Operating System? Explain the *process state transition* diagram. 4

b) What is meant by processor affinity? Explain *symmetric and asymmetric multiprocessing* used in a multi-processor process scheduling system. 4

c) Provide a race-condition free solution for the classical *reader-writer* problem with the following modification. Assume that, there can be *n*-number of reader processes and *m*-number of writer processes in the system. Any number of reader processes can read the system simultaneously. No writer process is allowed into the system until all the reader processes leave the system. If there is no reader process in the system, any number of writer processes can try to access the system but only one writer process can write into the system at a time. All the reader processes trying to read the system while there are writer processes in the system have to wait until all the writer processes leave the system. Explain your solution using any situation that reflects the mentioned criteria. 6

7. a) Rewrite the *Peterson's solution* for handling race-condition among n-number of processes accessing a shared resource. You have to write the pseudo-code for only one process. 4

b) Using the standard *producer-consumer* problem, show the disadvantage of using wakeup and sleep commands used for synchronizing the producer and consumer processes. 4

c) Given the following process information, find the average turn-around time using the *Round-Robin* (RR) and *Shortest Remaining Time Next* process scheduling algorithms. In the case of RR algorithm, assume that the time quantum **q=4.** 6

Process	**Arrival time**	**CPU burst time**
P1	7	12
P2	5	11
P3	6	9
P4	0	17
P6	14	13

Date: 25/09/17

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

3rd Year, 2nd Semester, Final Examination, Spring 2017

Course No: **CSE3213** Course Title: **Operating System**

Full Marks : 70

Time: 3 hours

[There are seven questions carrying a total of 14 marks each. Answer any five questions.
Marks allotted are indicated in the right margin.]

1. a) Show with necessary diagrams the method of accessing hardware resources by a user process. 4

   b) Explain the *Type-1* and *Type-2 hypervisors* in a virtual machine Operating System. 4

   c) Discuss the *Monolithic* and *Microkernel* Operating System architecture with necessary diagrams. 6

2. a) What are the ways an Operating System can deal with a *process-deadlock* situation? 4

   b) What is the difference between a *user-level* and a *kernel-level thread*? Discuss the kernel-level thread model with necessary diagrams. 4

   c) You are given the following information. Use the *Banker's algorithm* to determine whether the system is currently in *safe state*. The symbols have their usual meaning. 6

		X	Y	Z
Resources=	{20,	19,	16}	

	Allocation			Max		
	X	Y	Z	X	Y	Z
P1	1	1	1	7	5	6
P2	2	3	3	5	8	5
P3	3	0	5	9	2	8
P4	2	2	0	8	8	6
P5	3	1	0	14	4	5
P6	1	2	1	4	6	3
P7	1	1	1	2	9	2

Now, Process P1 requests <X=2, Y=1, Z=1>; Will it be safe to serve this request ?

3. a) What do you understand by *Dynamic Loading* and *Dynamic Linking*? Show the virtual address and its corresponding physical memory address with the following information. The symbols have their usual meaning. 4

Page frame size : 16 KB

RAM size: 10MB

Address space size of the process : 15 MB

The process requests data from it 100th page with the offset being 15300.

b) What are the differences between *global* and *local page replacement* technique? 4

c) Compare the *Least Recently Used* and *FIFO* memory page replacement algorithms in terms of the number of page faults using the following page requests by a process P1. Assume that, there are only three page frames in the memory. 6

Page requests: 0, 5, 4, 2, 0, 1, 7, 9, 11, 5, 0, 4, 7, 0, 5, 11, 3, 11, 2, 1.

4. a) Draw the diagram for an address translation process in any virtual memory system with a Translation Look-aside Buffer. 4

b) Explain the two level memory-page-table construction using any example. 4

c) What are the different ways an Operating System can keep track of the free memory spaces? Using the following data, calculate the total amount of external fragmentation separately for the *first fit* and *best fit* memory allocation techniques. Assume that, space allocation is variable length and contiguous. 6

Memory holes : 10 5 2 22 11 7 8

Memory requests : 5 6 11 6 6 10

5. a) Explain the *File Allocation Table* structure used by an Operating System. 4

b) With an 8KB data sector size, what could be the maximum file size using a *Linux inode data structure*? Assume that, each data pointer takes 8 bytes of space. 4

c) What do you understand by *sector interleaving*? Requests for read operations from the data blocks residing in the following cylinders are given below. 6

2 6 3 1 52 19 80 70 37 62 12 29 10 390 79 310

Show the sequence of requests served using the *Shortest Seek time first* and *C-Look* disk scheduling algorithms and calculate the total number of cylinder movements. Assume that, the cylinders are numbered from 0 to 400 and the disk head is currently positioned over the cylinder 51. The disk head is moving towards the higher numbered cylinders.

In the case of *C-Look* algorithm, assume that, the disk head serves any cylinder request while moving from higher to lower numbered cylinders only.

6. a) What is the function of the *dispatcher module* of an Operating System? Explain the *process state transition* diagram. 4

b) What is meant by processor affinity? Explain *symmetric and asymmetric multiprocessing* used in a multi-processor process scheduling system. 4

c) Provide a race-condition free solution for the classical *reader-writer* problem with the following modification. Assume that, there can be *n*-number of reader processes and *m*-number of writer processes in the system. Any number of reader processes can read the system simultaneously. No writer process is allowed into the system until all the reader processes leave the system. If there is no reader process in the system, any number of writer processes can try to access the system but only one writer process can write into the system at a time. All the reader processes trying to read the system while there are writer processes in the system have to wait until all the writer processes leave the system. Explain your solution using any situation that reflects the mentioned criteria. 6

7. a) Rewrite the *Peterson's solution* for handling race-condition among n-number of processes accessing a shared resource. You have to write the pseudo-code for only one process. 4

b) Using the standard *producer-consumer* problem, show the disadvantage of using wakeup and sleep commands used for synchronizing the producer and consumer processes. 4

c) Given the following process information, find the average turn-around time using the *Round-Robin* (RR) and *Shortest Remaining Time Next* process scheduling algorithms. In the case of RR algorithm, assume that the time quantum **q=4.** 6

Process	Arrival time	CPU burst time
P1	7	12
P2	5	11
P3	6	9
P4	0	17
P6	14	13

Date: 28th September, 2016

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Third Year, Second Semester **Final Examination, Spring 2016**

Course No: **CSE 3213** Course Title: **Operating System**

Time: 3 Hours Full Marks: 70

[Answer any 5(Five) sets from 7(Seven) sets.]
[Marks allotted are indicated in the right margin within '[ ]'.]

1.a) Suppose someone from MIT has designed a scheduling algorithm. In his design he said that the process with completion time/CPU burst time "x" should be given priority, $p(x) = x^2$. Do you agree that, this algorithm will work better? Justify your answer. [3]
[Explanation: Suppose you have 4 processes with completion time 2, 1, 3, 5. The priority of the processes would be 4, 1, 9 and 25. That means process no. 4 will run first, then process 3, then process 1 and lastly process 2]

b) How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system? [3]

c) Describe the hardware implementation of the page table. Show an example where in the logical address, page size n=2 and m=4. The physical address space is having 32 Byte size. [5]

d) Consider the following segment table: [3]

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. 1, 10
iii. 2, 500
iv. 3, 400
v. 4, 112

2.a) How can we prevent the occurrence of a deadlock? Describe all the protocols. [4]

b) What system calls have to be executed by a command interpreter or shell in order to start a new process? [2]

c) A single-lane bridge connects the two Vermont village of North Tunbridge and South Tunbridge. Farmers in the two villages use this bridge to deliver their products to their neighboring town. The bridge can become deadlocked if both a northbound and a southbound farmer get on the bridge at the same time (Vermont farmers are stubborn and are unable to back up). Using semaphores, design an algorithm that prevents deadlock. Do not be concerned about starvation (the situation in which northbound farmers prevent southbound farmers from using the bridge or vice versa). [3]

d) The general structure of a typical process P_i in critical-section problem is given below. [5]

```
do{
    Entry section
        Critical section
    Exit section
        Reminder section
}while(TRUE)
```

i) Now write the codes for the entry section and exit section using Peterson's solution to the critical-section problem. Also describe the data structures that you have used to solve the problem.

ii) What are the requirements that must be satisfied by a solution to the critical-section problem?

iii) Now prove that Peterson's solution satisfies all the requirements that must be satisfied by a solution to the critical-section problem.

3.a) Draw and describe the queuing-diagram representation of process scheduling. [4]

b) Differentiate between the direct communication link and the indirect communication link. [4]

c) Compare the main memory organization schemes of contiguous memory allocation, paging, and segmentation with respect to the following issues: [2]

i. External fragmentation

ii. Internal fragmentation

d) What is Network Attached Storage (NAS) and Storage Area Network (SAN)? Could a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organization (with nonredundant striping of data)? If so, how? [4]

4.a) In Shortest Job First scheduling algorithm, we process the job with shortest CPU burst time. Say you have n processes to schedule. The CPU burst time for each process can be at most m and at least 1. The total completion time is defined as the summation of individual CPU burst time of each process. Can you find the maximum and minimum total completion time? [2]
[Explanation: say n=4 and m=8. say the CPU burst time of processes are 2, 4, 5, 1 (all are less than 8 and greater than or equal 1). Now the total completion time is: 1+2+4+5 = 12. In this question you are asked to maximize and minimize this summation value.]

b) What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other? [3]

c) What is a Wait-for graph? Why is it used for detecting deadlocks instead of resource-allocation? [4]

d) Explain the dining-philosophers problem. Write down the solution of this problem. [5]

5.a) Servers can be designed to limit the number of open connections. For example, a server may wish to have only *N* socket connections at any point in time. As soon as *N* connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections. [3]

b) Although the SJF algorithm is optimal, in that it gives the minimum average waiting time for a given set of processes, why it cannot be implemented at the level of short-term CPU scheduling? How can we solve this problem? [2]

c) What is race condition? Now assume two operations A(counter++) and B(counter- -): [4]

A: register1 = counter
register1 = register1 + 1
counter = register1

B: register2 = counter
register2 = register2 – 1
counter = register2

Show a computation sequence to illustrate how race condition may happen.

d) Suppose you have been given a multilevel queue scheduler with two levels shown below. [5]

Round Robin (Quantum =4)
Shortest Job First

Consider the following data:

Process	Level	Burst Time	Arrival Time
P1	1	11	3
P2	2	13	0
P3	2	6	8
P4	1	5	11
P5	1	7	17

Now, show the Gantt chart, Waiting Time, and Turn Around Time for each of the process.

6.a) Consider the following set of process, with the length of CPU burst given in milliseconds. [6]

Process	***Burst Time***	***Priority***
P_1	9	4
P_2	13	2
P_3	7	3
P_4	15	1
P_5	9	0

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.

i. Draw four Gantt chart that illustrate the execution of these processes using the following scheduling algorithms: non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 5).
ii. What are the waiting time and turnaround time of each process for each of the scheduling algorithms in part (i)?

b) Several types of scheduling policies were discussed in class e.g. first-come-first-served (FCFS), round robin (RR), shortest job first (SJF), multilevel feedback queues (MLFQ). Suppose you want to optimize your scheduler for certain types of workloads. For each type, state and briefly justify which type of scheduler you would use: [2]

i. Multiuser workloads in which no individual user should be favored.
ii. Workloads with many mixed CPU and I/O jobs.
iii. Workloads with frequent I/O bound jobs and some very long-running, CPU-heavy jobs.

c) Write the Banker's algorithm. [4]

d) What is multilevel feedback queue scheduling? Describe with an example. [2]

7.a) Describe Amdahl's law. If an application is 25% parallel and 25% serial and the application is moving from single core, to dual core what will be the speedup ratio? [2]

b) Consider the following snapshot of a system: [6]

Process	***Allocation***	***Max***	***Available***
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm:

i. What is the content of the matrix Need?
ii. Is the system in a safe state?
iii. If a request from the process P1 arrives for (0 4 2 0), can the request be granted immediately?

c) Consider two concurrently running processes P_1 with a statement S_1 and P_2 with a statement S_2. Suppose we require that S_2 be executed only after S_1 has completed. How can we solve this synchronization problem using semaphores? [2]

d) Describe the options for breaking a deadlock after detecting a deadlock in the system. [4]

Date: 17th October, 2015

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Third Year, Second Semester **Final Examination, Spring 2015**

Course No: **CSE 3213** Course Title: **Operating System**

Time: 3 Hours Full Marks: 70

[Answer any 5(Five) sets from 7(Seven) sets.]

[Marks allotted are indicated in the right margin within '[ ]'.]

1.a) Consider the various definitions of Operating System. Consider whether the operating system should include applications such as Web browsers and mail programs. Argue both that it should and that it should not, and support your answer. **[3]**

b) How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system? **[2]**

c) Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in the logical address table and the physical address table? **[2]**

d) Consider the following segment table: **[3]**

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. 1, 10

iii. 2, 500

iv. 3, 400

v. 4, 112

e) Differentiate between internal and external fragmentation. Why valid-invalid bit is used in the page table? Give an example. **[4]**

2.a) What are the five major activities of an operating system with regard to process management? [2]

b) What system calls have to be executed by a command interpreter or shell in order to start a new process? [2]

c) Race conditions are possible in many computer systems. Consider a banking system that maintains an account balance with two functions: deposit(amount) and withdraw(amount). These two functions are passed the amount that is to be deposited or withdrawn from the bank account balance. Assume that a husband and wife share a bank account. Concurrently, the husband calls withdraw( ) function and the wife calls deposit( ). Describe how a race condition is possible and what might be done to prevent the race condition from occurring. [2]

d) The general structure of a typical process P_i in critical-section problem is given below. [6]

```
do{
    Entry section
        Critical section
    Exit section
        Reminder section
}while(TRUE)
```

i) Now write the codes for the entry section and exit section using Peterson's solution to the critical-section problem. Also describe the data structures that you have used to solve the problem.

ii) What are the requirements that must be satisfied by a solution to the critical-section problem?

iii) Now prove that Peterson's solution satisfies all the requirements that must be satisfied by a solution to the critical-section problem.

e) Could you simulate a multilevel directory structure with a single-level directory structure in which arbitrarily long names can be used? If your answer is yes, explain how you can do so, and contrast this scheme with the multilevel directory scheme. If your answer is no, explain what prevents your simulation's success. [2]

3.a) Draw and describe the queuing-diagram representation of process scheduling. **[3]**

b) Consider a system where the processes are communicating using indirect communication link. Now suppose that processes p1, p2, and p3 all share mailbox A. Process p1 sends a message to mailbox A, while both p2 and p3 execute a receive( ) from mailbox A. Which process will receive the message sent by p1? Write all the methods to give the answer of this question. **[3]**

c) Consider a system consisting of m resources of the same type being shared by n processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock free if the following two conditions hold: **[2]**

i. The maximum need of each process is between 1 and m resources.

ii. The sum of all maximum need is less than m + n.

d) What is Network Attached Storage (NAS) and Storage Area Network (SAN)? Could a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organization (with nonredundant striping of data)? If so, how? **[4]**

e) Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is: **[2]**

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

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. FCFS

ii. SSTF

4.a) What resources are used when a thread is created? How do they differ from those used when a process is created? **[3]**

b) What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other? **[3]**

c) What is Sun Network File System? Explain. **[2]**

d) How VFS layer allows an operating system to support multiple types of file systems? **[2]**

e) Suppose you have given a multilevel queue scheduler with two levels shown below. [4]

Round Robin (Quantum=3)
Shortest Job First

Consider the following data:

Process	*Level*	*Burst Time*	*Arrival Time*
P1	1	10	3
P2	2	8	0
P3	2	9	8
P4	1	5	11
P5	1	7	17

Now show the Gantt chart using these data.

5.a) Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections. [3]

b) 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: [2]

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. Now assume that recent CPU usage for process P_1 is 40, process P_2 is 18, and 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?

c) Explain dining-philosophers problem. Write down the solution of this problem [5]

d) Show that, if the wait( ) and signal( ) semaphore operations are not executed atomically, then mutual exclusion may be violated. [2]

e) Explain why spinlocks are not appropriate for single-processor systems, yet are often used in multiprocessor systems. [2]

6.a) Consider the following set of process, with the length of CPU burst given in milliseconds. [8]

Process	Burst Time	Priority
P_1	9	4
P_2	13	2
P_3	7	3
P_4	15	1
P_5	9	0

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.

i. Draw four Gantt chart that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 5).

ii. What are the waiting time and turnaround time of each process for each of the scheduling algorithms in part (i)?

b) Several types of scheduling policies were discussed in class e.g. first-come-first-served (FCFS), round robin (RR), shortest job first (SJF), multilevel feedback queues (MLFQ). Suppose you want to optimize your scheduler for certain types of workloads. For each type, state and briefly justify which type of scheduler you would use: [2]

i. Multiuser workloads in which no individual user should be favored.

ii. Workloads with many mixed CPU and I/O jobs.

iii. Workloads with frequent I/O bound jobs and some very long-running, CPU-heavy jobs.

c) Describe the options for breaking a deadlock after detecting a deadlock in the system. [2]

d) What is Wait-for graph? Why is it used for detecting deadlocks instead of resource-allocation? [2]

7.a) What is multilevel feedback queue scheduling? Describe with an example. [2]

b) Consider the following snapshot of a system: [6]

Process	***Allocation***	***Max***	***Available***
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm:

i. What is the content of the matrix Need?

ii. Is the system in a safe state?

iii. If a request from the process P1 arrives for (0 4 2 0), can the request be granted immediately?

c) Consider two concurrently running processes: P_1 with a statement S_1 and P_2 with a statement S_2. Suppose we require that S_2 be executed only after S_1 has completed. How can we solve this synchronization problem using semaphores? [2]

d) In a computer system, neither the resources available nor the demands of processes for resources are consistent over long periods. Resources break or are replaced, new processes come and go, new resources are bought and added to the system. If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances? [4]

i. Increase *Available* (new resources added).

ii. Decrease *Available* (resource permanently removed from system).

iii. Increase *Max* for one process (the process needs more resources than allowed, it may want more).

iv. Decrease *Max* for one process (the process does not need that many resources).

v. Increase the number of processes.

vi. Decrease the number of processes.