Date of Examination: *24.05.2021*

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering
Program: B.Sc. in Computer Science and Engineering
Semester Final Examination: Spring 2020
Year: 3rd Semester: 2nd
Course Number: CSE3213
Course Name: Operating System

Time: 3 (Three) Hours **Full Marks: 60**

Use single answer script

Instructions:	i)	Answer script should be hand written and should be written in A4 white paper. You must submit the hard copy of this answer script to the Department when the university reopens.
	ii)	You must write the following information at the top page of each answer script: **Department:** **Program:** **Course no:** **Course Title:** **Examination:** **Semester (Session):** **Student ID:** **Signature and Date:**
	iii)	Write down Student ID, Course number and put your signature on top of every single page of the answer 3script.
	iv)	Write down page number at the bottom of every page of the answer script.
	v)	Upload the scan copy of your answer script in PDF format through provided **google form** at the respective course site (i.e., **google classroom)** using institutional email within the allocated time. Uploading clear and readable scan copy (uncorrupted) is your responsibility and you must cover all the pages of your answer script. However, for clear and readable scan copy of the answer script student should use only one side of a page for answering the questions.
	vi)	You must avoid **plagiarism,** maintain **academic integrity, and ethics**. You are not allowed to take any help from another individual and if taken so can result in stern disciplinary actions from the university authority.
	vii)	Marks allotted are indicated in the **right margin.**
	viii)	Necessary **charts/tables** are attached at the end of the question paper. You may use graph papers where necessary.
	ix)	Assume any reasonable data if needed.
	x)	Symbols and characters have their usual meaning.
	xi)	Before uploading, rename the PDF file as **CourseNo_StudentID.pdf** e.g., CSE3213_180104001.pdf

The answer script **(one single PDF file)** must be uploaded at designated location in the provided **Google Form link** available in the Google classroom.

There are 7 (Seven) Questions. Answer any 5(Five).

Question 1. [Marks: 12]

a)	Why is it necessary to have an Operating System service named *system call*? Explain with an example the execution process of a system call.	[1+3]
b)	What are the ways an Operating System has to deal with deadlocks in the system? Create a directed *Resource Allocation Graph (RAG)* with at least ten processes, ten resources and three deadlocks. Now show using a cycle detection algorithm the process of detection of all the individual deadlocks in your designed RAG.	[1+3]
c)	Why is a thread called *Light Weight process*? Explain the kernel-level thread architecture.	[1+3]

Question 2. [Marks: 12]

a)	Explain the importance of *CPU utilization* from the Operating System's point of view. 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 with only one writer process allowed to 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.	[1+3]
b)	What is the function of the *dispatcher* module of an Operating System? Explain the *state-transition* diagram of a process.	[2+2]
c)	Given the following process information, compare the *Round-Robin (RR)* and *Shortest Remaining Time Next* process scheduling algorithms in terms of average response and waiting time. In the case of the RR algorithm, assume that the time quantum q = 4.	[4]

Process	Arrival time	CPU burst time
P1	5	12
P2	0	11
P3	4	9
P4	12	17
P6	35	13

Question 3. [Marks: 12]

a)	Requests for read operations from the data blocks residing in the following cylinders are given below. 12 6 3 121 52 9 80 70 37 62 2 60 10 310 9 31 21 5 13 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 500 and the disk head is currently positioned over cylinder 41. The disk head is moving towards the higher numbered cylinders. In the case of the *C-look* algorithm, assume that the disk head serves any cylinder request while moving from lower to higher numbered cylinders only.	[4]

b)	Explain the procedure of storing a file using the *inode* data structure in a Unix-based Operating System with necessary diagrams. Assume a 4-KB block size and a 4-byte pointer size. Also, show the file locating mechanism of the file '/etc/config/hosts.abc'	[4]
c)	Explain the *File Allocation Table* architecture with necessary figures. How can an OS handle a bad sector in a disk drive?	[2+2]
Question 4. [Marks: 12]		
a)	How can an Operating System estimate CPU time for a process? Explain race-condition with examples.	[2+2]
b)	Why is a *busy-waiting-based* solution towards the race-condition problem not preferable? Explain the Semaphore data structure and its usage.	[2+2]
c)	What is the difference between a process and a program? Compare the Peterson's and the strict alteration algorithms for handling the race-condition problem	[4]
Question 5. [Marks: 12]		
a)	How can an OS detect a *memory page-fault*? Show the virtual address and its corresponding physical memory address with the following information. Assume that the requested data is available in the 28^{th} memory frame. Page frame size: 16 KB RAM size: 16MB The Address space size of the process: 32 MB The process requests data from its 200th page with the offset being 15200.	[1+3]
b)	Using the following page requests, compare the *Least Recently Used* and *Optimal page replacement* algorithms. Assume that, there are only three page-frames in the memory. Page requests: 0, 15, 4, 2, 12, 0, 1, 7, 9, 7, 15, 0, 4, 7, 10, 5, 21, 3, 11, 5, 1, 21	[4]
c)	Draw the diagram for an address translation process in any virtual memory system with a *Translation Look-aside Buffer*.	[4]
Question 6. [Marks: 12]		
a)	Explain *internal* and *external memory fragmentation* with examples. Show the application of *best* and *worst fit* memory allocation algorithm using some dummy data.	[2+2]
b)	Show the memory protection mechanism that an OS employs in the case of a contiguous memory allocation scheme. Explain the process of keeping track of free and allocated memory (variable-sized memory partition) using linked-list	[2+2]
c)	Why is *virtual memory* needed? What are the differences between *global* and *local page replacement* techniques?	[1+3]
Question 7. [Marks: 12]		
a)	Present a situation with an explanation where a thread creation is preferable instead of replicating the whole process.	[4]
b)	Explain the *Type-1* and *Type-2 hypervisors* in a virtual machine Operating System.	[4]
c)	Explain the options that an OS has to recover from a deadlock.	[4]