Date of Examination: *24.10.2021*

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

Time: 02 (Two) Hours **Full Marks: 50**

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 script.
	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_190104001.pdf
	xii)	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 06 (Six) Questions. Answer any 4(Four).

Question 1. [Marks: 12.5]

a) Why is an OS called a Kernel? With an example explain the steps of accessing hardware resources by an application program. [1.5+6]

b) How can an OS ensure better CPU Utilization? Show with examples. Explain the kernel-level thread architecture. [2+3]

Question 2. [Marks: 12.5]

a) Provide the solution (in pseudo-code form) for the standard Dining Philosopher problem. The algorithm must use a lock-variable based solution to deal with the race-condition problem ensuring guaranteed mutual exclusion. Which means, for the mutual exclusion, you cannot use the semaphore data structure. Show the effectiveness of your algorithm for the situation where philosopher 0 is currently in 'eating' state and both the philosopher 1 and philosopher 2 are trying to change their state to 'eating'. [5.5]

b) Based on the following data, provide a design for a Multilevel Feedback Queue with proper justification. Comment on the efficiency of your design based on the average waiting time for the processes. [7]

Process	Arrival time	CPU burst time
P1	0	72
P2	6	14
P3	9	8
P4	12	7
P5	20	3
P6	28	80
P7	68	14
P8	25	13

Question 3. [Marks: 12.5]

a) What is a page-fault from an OS's point of view? Show the virtual address and its corresponding physical memory address with the following information. Assume that the requested data is available in the 318^{th} memory frame. [1.5+4]
Page frame size: 128 KB
RAM size: 1GB
The Address space size of the process: 256 MB
The process requests data from its 210^{th} page with the offset being 31×2^{10}.

b) Discuss the fundamental concept of using Translation lookaside buffer**.** Using the following page requests, compare the *Least Recently Used* and *Optimal page replacement* algorithms. Assume that, there are five page-frames in the memory. [2+5]

Page requests: 14, 4, 90, 10, 15, 14, 2, 12, 0, 41, 7, 9, 7, 15, 0, 14, 7, 10, 5, 21, 3, 11, 5, 1, 41, 10, 90

Question 4. [Marks: 12.5]

a) Explain the causes where a set of processes can be trapped into a deadlock situation. How can a deadlock be prevented? [2.5+2.5]

b) Explain the following terms with necessary figure(s) with respect to a multiprocessor based scheduling system. [7.5]

i) Asymmetric multiprocessor scheduling
ii) Symmetric multiprocessor scheduling
iii) Pull and Push migration during load balancing

Question 5. [Marks: 12.5]

a) Calculate the maximum allowable file size using the *inode* data structure in a Unix-based Operating System with necessary diagrams. Assume an 8-KB block size and a 2-byte pointer size. Also, show the file locating mechanism of the file '/etc/passwd' [3.5+1]

b) Requests for read operations from the data blocks residing in the following cylinders are given below. [8]

2 16 3 9 80 70 37 32 416 10 210 19 240 15 33

Assume that the cylinders are numbered from 0 to 600 and the disk head is currently positioned over the cylinder 111. The disk head is moving towards the higher numbered cylinders. In the case of the *C-look/C-SCAN* algorithm, assume that the disk head serves any cylinder request while moving from lower to higher numbered cylinders only. Carefully think and propose the best disk scheduling algorithm (as per the algorithms covered in the class) based on the above data with necessary comparison with other methods.

Question 6. [Marks: 12.5]

a) Compare the Microkernel and Monolithic modular architecture and point out which one may you choose while designing an Operating System. [4.5]

b) What are the probable options for recovering from a deadlock? [4]

c) Show the calculation steps of the equation that an OS uses to estimate CPU time for a process. [4]