University of KwaZulu-Natal
School of Engineering
Discipline of Computer Engineering
Internal Examiner: Mr. Raleigh Reddy
External Examiner: Dr. Melvin Ferreira

UNIVERSITY OF KWAZULU-NATAL
INYUVESI YAKWAZULU-NATALI

ENEL4OS
Operating Systems for Engineers
Date: 01 June 2026
Time: 120 minutes
Total marks: 100 marks

Main Examination

Instructions:

- All algorithms must be implemented in pseudocode.
- Mark allocations are indicated in brackets.
- The question paper consists of 3 pages, please make sure you have them all.

Question 1 (10 marks)

1.1. Explain the difference between absolute and relative paths. (2 marks)

1.2. Explain the difference between batch multiprogramming and time sharing systems in terms of their principal objective. (2 marks)

1.3. State 6 attributes that files typically have. (3 marks)

1.4. State 6 basic operations that are required to allow users to work with and on files besides the opening and closing of them. (3 marks)

Question 2 (20 marks)

2.1. A system is composed of 6 processes, (P1, P2, P3, P4, P5 & P6), and 3 types of serially reusable resources, (R1, R2 & R3). The system has 2 instances of R1, 2 instances of R2 and 2 instances of R3.

Suppose that:

- P1 holds 1 instance of R1 and requests 1 instance of R3.
- P2 holds 1 instance of R2 and requests 1 instance of R1.
- P3 holds 1 instance of R1 and requests 1 instance of R2.
- P4 holds 1 instance of R2 and requests 1 instance of R1.
- P5 holds 1 instance of R3.
- P6 requests 1 instance of R3.

2.1.1 Draw the resource-allocation graph (RAG) for this system state. (3 marks)

2.1.2 Which, if any, of these processes are in deadlock? Justify your answer. (1 mark)

2.2. Consider a system with 4 processes, P0 through to P3 and 4 resource types A, B, C and D. Resource type A has 7 instances, resource type B has 6 instances, resource type C has 4 instances and resource type D has 5 instances. Suppose that, at time T0, the following snapshot of the system has been taken:

Max

	A	*B*	*C*	*D*
P0	4	2	1	3
P1	1	2	2	1
P2	1	1	2	2
P3	3	1	3	5

Need

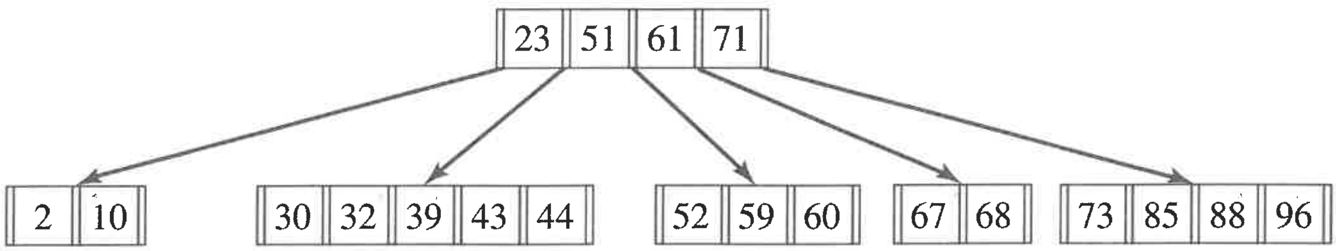
	A	*B*	*C*	*D*
P0	0	1	1	2
P1	1	0	2	1
P2	1	1	0	1
P3	2	0	1	3

Available

A	B	C	D
2	2	0	1

Use the banker's algorithm to determine if the system is in a safe state. (5 marks)

2.3. Consider the B-tree given below:



2.3.1. Show the new B-tree that is formed when the key = 90 is inserted. (1 marks)

2.3.2. Using the B-tree formed in 2.3.1, show the new B-tree that would be formed from it when the key = 45 is inserted. (10 marks)

Question 3 (30 marks)

3.1. Consider a single-processor system that is using the round robin (RR) scheduling algorithm with the time quantum being 5 milliseconds. The following set of 5 processes below arrive at the ready queue for the times given along with their respective CPU burst times also provided in milliseconds.

Name	Arrival Time	Burst Time (ms)
P1	0	15
P2	10	8
P3	15	4
P4	20	12
P5	27	7

3.1.1. Provide a GANNT chart of the RR algorithm for this set of tasks. (5 marks)

3.1.2. What is the average waiting time for the RR algorithm? (2 marks)

3.2. Consider a single-processor system that is using the preemptive shortest job first (SJF) scheduling algorithm. The following set of 7 processes below arrive at the ready queue for the times given along with their respective CPU burst times also provided in milliseconds.

Name	Arrival Time	Burst Time (ms)
P1	0	9
P2	5	6
P3	9	8
P4	11	11
P5	18	9
P6	30	4
P7	37	2

3.2.1. Provide a GANNT chart of the preemptive SJF algorithm for this set of tasks. (7 marks)

3.2.2. What is the average waiting time for the preemptive SJF algorithm? (2 marks)

3.3. Consider a single-processor system that is using the preemptive priority scheduling (PS) scheduling algorithm with first-come, first-served (FCFS) as the tie-breaking rule. The following set of 5 processes below arrive at the ready queue for the times given along with their respective CPU burst times also provided in milliseconds. The priority values for each process is also shown with lower values having a higher priority.

Name	Arrival Time	Burst Time (ms)	Priority
P1	0	10	5
P2	5	12	3
P3	15	10	1
P4	20	14	2
P5	25	11	4

3.3.1. Provide a GANNT chart of the preemptive PS algorithm for this set of tasks. (5 marks)

3.3.2. What is the average waiting time for the preemptive PS algorithm? (2 marks)

3.4. List 3 different storage levels that the NTFS (New Technology File System) uses. (3 marks)

3.5. Describe the 4 main objects found in the virtual filesystem architecture of Linux. (4 marks)

Question 4 (40 marks)

4.1. The Reader/Writer problem is one of the classic problems of synchronization wherein some processes want to only read from a resource like a database and are referred to as the Readers. The other processes want to write and update the database and are referred to as the Writers. The solution discussed can be described as readers-preference because no reader is kept waiting to use the resource unless a writer is currently writing to it. That is because once any reader locks the resource, it is not given up until all readers are finished with it and hence any writers may wait a long time.

Provide a pseudocode solution via semaphores inclusive of comments to the Reader/Writer synchronization problem in which the readers are given preference. As such, no reader is kept waiting to use the resource unless a writer is currently writing to it and once any reader locks the resource, it is not given up until all readers are finished with it. (18 marks)

The pseudocode will have to use the following shared variables:

1. **readcount**: an integer variable (initialized to 0) that keeps track of how many processes are currently reading the object.
2. **mutex:** a semaphore (initialized to 1) which is used to ensure mutual exclusion when readcount is updated i.e. when any reader enters or exit from the critical section.
3. **wrt**: a semaphore (initialized to 1) common to both reader and writer processes.

4.2. The Dining-Philosophers problem is another one of the classic problems of synchronization wherein 5 philosophers are seated around a circular table either thinking or eating. When a philosopher gets hungry he/she tries to pick up 2 chopsticks that are closest to him/her in order to eat. He/she cannot pick up a chopstick that is already in the hand of a neighbor. Only when a hungry philosopher has both chopsticks at the same time then would he/she begin eating without releasing the chopsticks. He/she would then put down both of the chopsticks once he/she has finished eating and continue thinking again.

Provide a pseudocode solution via semaphores inclusive of comments to the Dining-Philosophers synchronization problem which will have to use the following shared variable: (10 marks)

chopstick[5]: a semaphore array where all elements are initialized to 1.

4.3. Describe the size characteristics of embedded systems. (3 marks)

4.4. Describe the 4 necessary conditions for deadlock to occur. (4 marks)

4.5. List and briefly describe the 5 states of a process. (5 marks)

END OF PAPER