2021 EXAMINATIONS

Part II

COMPUTING AND COMMUNICATIONS – On-line Assessment

Available Time **[23 Hours]**

Recommended Completion Time **[3 Hours]**

SCC.211 OPERATING SYSTEMS

---

*Candidates are asked to answer **THREE** questions from **FOUR**; each question is worth a total of 20 marks.*

Question 1

1.a

i. Briefly explain why concurrent programming can be particularly difficult to debug.

[2 marks]

ii. How and in which conditions does concurrency improve the 'performance' of software?

[2 marks]

iii. What is meant by a livelock, why are these a problem, how is it different from a deadlock, and what synchronization primitive can be leveraged to avoid their occurrence?

[4 marks]

iv. It is theoretically possible to spawn billions of threads for a given process – provide two reasons why achieving this many active threads within standard PC hardware is not feasible.

[2 marks]

1.b

i. Define what is meant by a race condition **and** a lost update problem, making sure that these definitions are clearly distinguished. Provide a concise example scenario within computers where both of these concepts may occur.

[4 marks]

ii. Provide a concise segment of pseudo code which depicts the occurrence of a race condition and lost update problem. Explicitly declare relevant variables and thread creation necessary. Clearly identify where the critical section occurs within this code.

[4 marks]

iii. In the context of your pseudo code provided in 1.b (ii), briefly describe how the code can be made thread safe.

[2 marks]

[Total 20 marks]

Question 2

2.a

i. Explain the differences between a process and a thread. Give **one** reason why these concepts are commonly conflated. **[3 marks]**

ii. Concisely describe one scenario where it is beneficial for a computer to **not** create and leverage multiple concurrent processes.

[2 marks]

iii. Briefly compare and contrast between a UNIX process and a user-level thread.

[3 marks]

iv. Discuss **one** advantage and **one** disadvantage of using kernel threads over user threads. Explain why a programmer might favour using one thread type over the other.

[4 marks]

2.b

A programmer has been tasked to create a program that upon pressing any key on the keyboard, the program will create 10 threads which begin independently iterating from 0 to 100.

After a random time interval, a calling thread will read the current value of all 10 threads, and output the final summed value. It is possible that this calling thread may arrive before the 10 counting threads have each completed counting to 100.

Write an implementation for this scenario, so that the final output is 1000 no matter when the calling thread arrives: You are free to use either Java-based or C-based pseudo-code and any concurrent programming primitives as you see fit. A precise description of the semantics of the concurrent primitives should be provided, which argues how they ensure non-determinism. Code should be as precise and complete as possible.

[8 marks]

[Total 20 marks]

Question 3

3.a A system has the following processes:

Process	1	2	3	4	5
Arrival Time (ms)	0	5	15	18	40
Burst Time (ms)	10	20	5	15	20

Based on these processes and **showing all working**…

i. Rank the following schedulers in terms of average turnaround and average waiting times

a. First Come First Served
b. Round Robin with 10ms quantum
c. Round Robin with 15ms quantum

[10 marks]

ii. Justifying your answer – one sentence for each, which of the scheduling schemes would be best for

a. A system where the processes are expected to be highly interactive
b. A system running payroll type jobs with no user interaction

[2 marks]

iii. What would be the Average Turnaround Time and Average Waiting Time if the processes are scheduled with a fixed priority, Round Robin scheduler with a 10ms quantum, process 4 is set to high-priority, and all other process are low-priority.

[3 marks]

Question 3 continues on next page…

Question 3 continued.

3.b The following shows the queue of track numbers corresponding to incoming and outstanding disk requests. Complete a copy of the diagram below for each of the SSTF and SCAN disk scheduling algorithms, and assuming it takes one unit of time to move from one track to the next, calculate the total seek time for requests under each scheme.

One request is serviced, and one new request arrives during each scheduling interval, so in the first interval, a request for track 22 arrives and it is immediately serviced. In the second interval, a request for track 18 arrives, and a request for track 24 is serviced.

22		**18**		**35**		**25**		**42**		**17**		**30**		**No requests**		
↓		↓		↓		↓		↓		↓		↓				
32		**18**														
24		**32**														
~~**22**~~		~~**24**~~														
↓		↓		↓		↓		↓		↓		↓		↓		↓
22		**24**														

[5 marks]

[Total 20 marks]

Question 4

4.a. The table below represents a set of page table mappings. Identify the physical addresses accessed for each of the following virtual addresses: 152, 278, 604, 321, 798, and 357.
You should assume:

- Addresses are 11 bits in length
- Values are given in hexadecimal
- The free frame list contains
  HEAD → 0 → 2 → 3 → Ø
- Frames are always taken
  from the head of the free list
- No current entry is indicated by: --

Page Table	
0	--
1	1
2	5
3	--
4	6
5	--
6	7
7	4

[4 marks]

4.b. A buddy allocator has the allocations below. Show the state of the memory after each of the following requests. The allocator prefers to allocate the lowest address, i.e. to the left.

Free C; Free B; E = Allocate(16K); Free A; F = Allocate(5K);
Free D; G = Allocate(24K); Free F; H = Allocate(8K); I = Allocate(10K)

A = 8K	**B = 4K**	**4K Free**	**C = 8K**	**D = 8K**

[8 marks]

Question 4 continues on next page…

Question 4 continued.

4.c. Given the page reference string { 2, 4, 3, 5, 4, 2, 8, 2, 6, 7, 3, 8 } and an initially empty set of four frames, determine the number of page faults that would be seen with both the FIFO and LRU schemes. You must show the mappings for all four frames after each memory access.

[8 marks]

[Total 20 marks]

--- End of Paper ---