2023 EXAMINATIONS

Lancaster University

Part II

COMPUTING AND COMMUNICATIONS – Written Exam [2 hours]

SCC.211 Operating Systems

---

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

Use a <u>separate</u> answer book for <u>each</u> question.

Question 1

1a. Write a program in Java-like pseudocode that launches 100 threads. Each of the threads increments a shared variable *count* by 1 (assume that *count* is initially 0). The program prints the value of count before exiting. Your program must ensure that the value of *count* printed is 100.

For synchronization, you are limited to using the *join* method on threads and the *synchronized* keyword. Other synchronization techniques are strictly disallowed.

Your program must be brief, to the point, and not overly complicated. You can omit exception handling from your code.

[20 marks]

[Total 20 marks]

Question 2

2a. Consider two threads with IDs 0 and 1, each with a critical section in which they access a shared resource. Say they are using Peterson's algorithm (reproduced below from the lecture notes) for mutual exclusion.

```
//Peterson's algorithm
int tiebreak = 0;
bool[] interested = {FALSE, FALSE};

void get_lock() {
      int self = thread_getid();
      int other = 1-self;
      interested[self] = TRUE;
      tiebreak = other;
      while(interested[other] && tiebreak == other) ;
}

void release_lock() {
      int self = thread_getid()
      interested[self] = FALSE;
}
```

i. The following is an interleaved execution of the threads, where the prefix 0: and 1: indicate whether the instruction is from thread 0 or thread 1.

```
0: interested[0] = TRUE
1: interested[1] = TRUE
0: tiebreak = 1
1: tiebreak = 0
```

After these four instructions, one of the two threads is able to enter its critical section. Which thread is it that is able to do so?

[2 marks]

ii. Give all the shared variables in Peterson's algorithm. **[2 marks]**

Question 2 continues on next page…

Question 2 continued.

2b. Consider the protocol Purchase given below.

```
Purchase {
   role B, S
   parameter out ID key, out item, out price

   B->S: request[out ID key, out item]
   B->S: buyerOffer[in ID key, out price]
   S->B: sellerOffer[in ID key, out price]
}
```

Which of the following statements are true?

i. A binding of the parameter `ID` identifies an enactment of `Purchase`.
ii. Every BSPL protocol must have a parameter named `ID`.
iii. It is possible for an enactment of `Purchase` to complete with multiple bindings for the parameter `price`.
iv. It is possible for an enactment of `Purchase` to complete with a unique binding for the parameter `price`.

[8 marks]

2c. Consider the protocol `Purchase` given above. Give two distinct and complete enactments of the protocol as UML interaction diagrams.

[8 marks]

[Total 20 marks]

Question 3

3.a Given a 64K memory area with 32K of remaining free space, as illustrated below, **produce a set of diagrams** that clearly show, **step-by-step**, how memory is allocated under the following schemes given an ordered set of requests for memory: 4K, 9K, 7K, and 4K.

- Buddy allocation
- Best Fit allocation
- Worst Fit allocation

8K	8K Free	**A** : 8K	8K Free	16K Allocated	16K Free

Shaded areas represent memory already allocated.

[12 marks]

3.b **Justifying your answer in one sentence for each** of case *a* and *b*, which, if any, of the final allocation states would benefit from de-fragmentation if the memory was:

i. RAM
ii. Disk storage

[2 marks]

3.c Assuming the block identified as *A* in the diagram above is freed immediately after servicing the 7K request; for each algorithm, given this state, what would be the largest amount of memory that could be returned for a single request? **Your answer must include diagrams** to show which memory is returned in each case.

[3 marks]

3.d For each of the above allocation schemes, identify the form of fragmentation typically seen.

[3 marks]

[Total 20 marks]

Question 4

A single priority round-robin scheduler is presented with the following processes:

Process	1	2	3	4
Arrival (ms)	0	5	25	37
Burst (ms)	25	10	15	20

4.a **Showing ALL working** and being careful to **justify your answer** in terms of total waiting and turnaround times, would it be better to have a scheduling quantum of 10 or 15ms?

Your answer **MUST show the schedule** of process execution for each quantum.

[10 marks]

4.b **Justifying your answer** and **showing ALL working**, would performance improve with a priority round-robin system if process one was scheduled with a higher priority than the other processes and a 10ms quantum used? You should assume that all other processes run with the same priority.

[6 marks]

4.c **Justifying your answer** and **showing ALL working**, would the system be better with a simple First Come First Served schedule?

[4 marks]

[Total 20 marks]

---End of Paper---