2018 EXAMINATIONS

Lancaster University

Part II

COMPUTING AND COMMUNICATIONS [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 separate answer book for each question.

Question 1

1.a Define what we mean by *determinism* and *non-determinism* in the context of concurrency, and describe the significance of non-determinism within concurrent programming. Provide an example where non-determinism is beneficial for multi-threaded software.

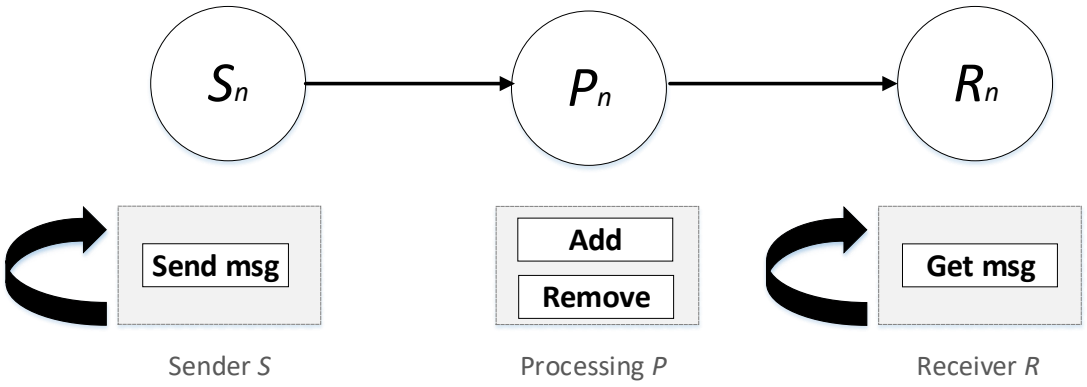
[4 marks]

1.b Explain the terms *thread* and *process*, and provide one difference between them.

[3 marks]

1.c

i. Consider a scenario as shown in the figure below where a Sender thread *Sn* sends a message *msg* to process thread P*n.* At the same time, Receiver thread *Rn* retrieves the message from the associated processing threads. If the receiver attempts to retrieve a message that does not exist within a processing thread, the program will crash. The execution order of receivers and senders is unknown.



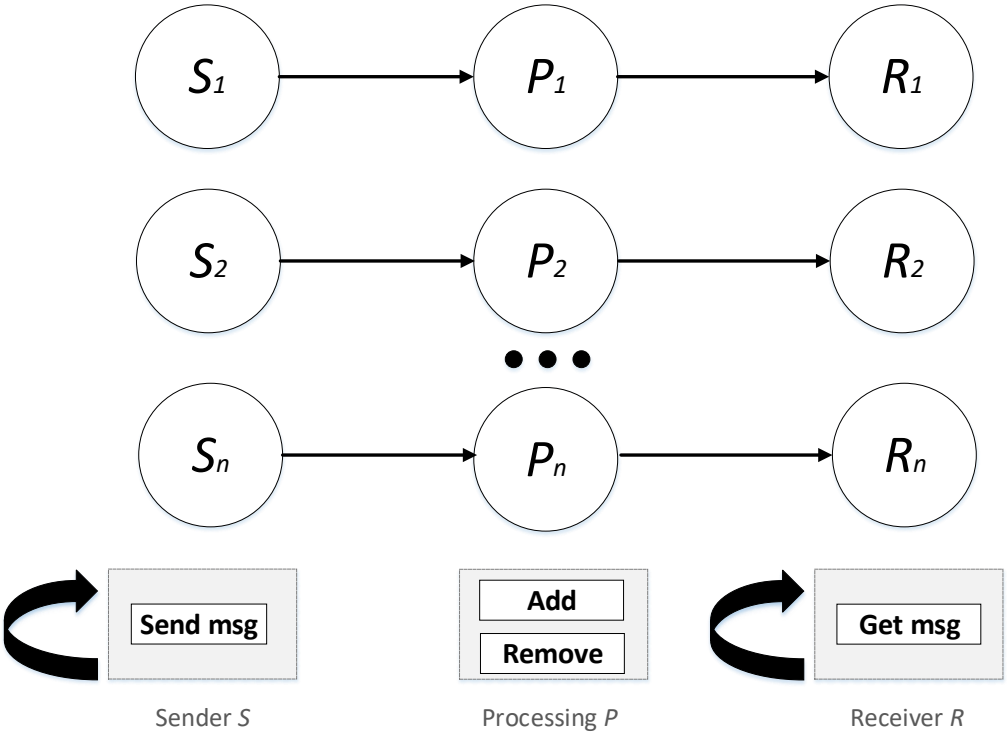
Write an implementation for this scenario that is thread-safe. You must include at least three methods Sender, Receiver, and Process, and any additional methods deemed necessary to ensure concurrency. You are free to use either Java-based or C-based pseudo code and any concurrent programming mechanisms as you see fit. The implementation and a very concise description of the semantics of these concurrency mechanisms should be provided, and code should be as clear as possible.

[10 marks]

Question 1 continues on next page…

Question 1 continued…

ii. Provide a description of how you might extend the implementation so it can allow *n* senders, *n* processes and *n* receivers as demonstrated in the diagram below.



[3 marks]

Total 20 marks

Question 2

2.a Define what is meant by Moore's Law and Amdahl's law, and how they are related to concurrency.

[4 marks]

2.b Define what is meant by competition and cooperation within the context of concurrent programming. Give examples of the type of concurrency mechanisms used in order to facilitate such behaviour within a program.

[4 marks]

2.c Describe a scenario where it is more advantageous to use spinlocks instead of blocking locks, and explain why.

[3 marks]

2.d A programmer has created a class named manipulation that contains three methods that are involved in producing a value for function f(x). The programmer desires that the class be capable of allowing *n* threads to access its contents simultaneously, and has begun creating skeleton code as shown below:

```
class manipulation implements Runnable {
      int x;
      int fX;

      magnify (int x){
            print("Magnifying…");
            x = x * 2;
            transform(x);
      }

      transform (int x){
            print("Transforming…");
            //fX = ??;       //Needs completing
      }

      public void run (){
            int x = num;     //Start of thread execution
            magnify(x);
            transform(x);
            print(fX);       //End of thread execution
      }
}
```

Question 2 continues on next page…

Question 2 continued…

Within run(), int x = num is defined, and then calls magnify(x) and transform(x). **Magnify( ) will double the value of x and also call transform(x) within itself**. transform( ) will calculate f(x) using the function $f(x) = (x + 2)^2$. After this is completed, the thread will call print(fX) and exit as shown in the comments.

Write an implementation for this scenario that is thread-safe: You are free to use either Java-based or C-based pseudo code and any concurrent programming mechanisms as you see fit. A concise description justifying the use of any concurrency mechanisms should be given.

[6 marks]

2.e Define what we mean by the '*Thundering Herd*' problem, outline the implications of this behaviour for tech companies that operate very large computing systems, and how could they mitigate its effect?

[3 marks]

Total 20 marks

Question 3

3.a A 32-bit computer system using paged memory has the following address structure, where the last two bits allow reference to the bytes forming the word being read or written:

10 bits	10 bits	10 bits	2 bits

i. Draw a diagram that illustrates the key values and data structures that the system must maintain for each process to operate such a paged system.

[3 marks]

ii. Justifying your answer, given any virtual/process address referencing an in-memory page, how many memory accesses would be required to retrieve the data held at that address?

[2 marks]

iii. What additional system or hardware is typically provided to reduce the cost of accessing memory in this type of paged system and what function does it perform?

[2 marks]

3.b Modern operating systems tend to implement page based memory allocation schemes. Explaining the role and need for each of buddy, slab, and zone schemes. Outline why operating systems typically provide such additional support beyond page based allocation.

[6 marks]

3.c Draw a series of diagrams that clearly illustrates how a buddy allocator would service the following sequence of requests from a 32K memory region. You should assume that the buddy implementation, given a choice, always returns the lowest available memory address.

[7 marks]

```
A = Allocate ( 3K )
B = Allocate ( 6K )
C = Allocate ( 4K )
D = Allocate ( 4K )
Free ( B )
Free ( C )
E = Allocate ( 10K )
Free ( A )
F = Allocate ( 13K )
```

Total 20 marks

Question 4

4.a Filesystems store a number of attributes for each file. Identify six common types of attribute likely to be found in a filesystem such as ext2.

[3 marks]

4.b Explain in detail how, within an operating system such as Linux or Windows, a call to a function such as read( ) along with its associated parameters would pass from a user space application to a call within the kernel to the matching filesystem function, and how the return value would be passed back to the application. Your answer must explain moves between user and system/kernel modes and how the correct filesystem driver is identified and called.

[7 marks]

4.c A traditional Unix type filesystem similar to ext2 uses 2048 byte disk blocks and four byte block pointers. Assuming this filesystem has inodes containing ten direct block pointers and one pointer for each of single, double and triple indirection, what is the maximum size of file that can be stored? Note: there is no requirement for you to give a numeric answer; you must show all working and should give your answer as an expression built from the relevant values given above.

[6 marks]

4.d A number of modern filesystems have moved away from traditional allocation to extent based schemes with delayed allocation, sometimes known as allocate on flush. Provide a brief description of the *extents* and *delayed-allocation* schemes, and the advantages the adoption of these can have over traditional allocation schemes within filesystems.

[4 marks]

Total 20 marks

--- End of Paper ---