RHODES UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

EXAMINATION: NOVEMBER 2020

COMPUTER SCIENCE HONOURS
PAPER 6
DISTRIBUTED AND PARALLEL PROCESSING

Internal Examiner:	Prof GC Wells	**MARKS**: 120 marks
External Examiner:	Prof I Sanders	**DURATION**: 2 hours

GENERAL INSTRUCTIONS TO CANDIDATES

1. This paper consists of **3 pages** and **9 questions**. **Please ensure that you have a complete paper.**
2. Answer ALL questions.
3. The use of calculators is permitted in the examination, however, make sure that you show all workings.
4. The Oxford Concise English Dictionary **may** be used during this examination.

DISTRIBUTED AND PARALLEL PROCESSING [120 MARKS]

QUESTION 1: Hardware Topologies [12+4 = 16 marks]

During the course we discussed several different processor *topologies* (or network topologies, for example, the Simple 2-D Mesh network), and four criteria for assessing the benefits of different topologies (i.e. diameter, bisection width, number of edges and edge length).

a) Explain these four criteria, and state what the ideal characteristics are for each criterion.
b) Name one of the other processor topologies (i.e. **not** the Simple 2-D Mesh), and sketch it.

QUESTION 2: Theory [20 marks]

Discuss Liu's hierarchy of abstractions for *distributed programming*, giving examples of common languages or libraries that implement each level of abstraction. Discuss how the abstraction level impacts on *ease-of-use* and on *performance*.

QUESTION 3: Theory [6 marks]

You have an application where 15% of the computation cannot be parallelised. What is the maximum speedup you might expect with 1000 processors available, compared to a sequential version of the application using one processor (show your working)? State the source/name of the equation you use.

QUESTION 4: Hardware/Theory [5+5 = 10 marks]

a) Discuss the relationship between *processor clock frequency* and *power consumption*, and the implications that this has had for modern processor design and the importance of parallel processing.
b) Discuss the relationship between the *computation : communication ratio* and *grain size* (or *granularity*) of a parallel application.

QUESTION 5: Java Threads [15 marks]

EITHER:

The `java.util.concurrent` package contains several *synchronizers*, including *semaphores*, *barriers* and *latches*. Discuss how each of these mechanisms provides synchronisation, giving a brief example of how they might be used.

OR:

Discuss the *executor* services that are provided by the Java Concurrency Utilities package (`java.util.concurrent`). What is the primary purpose of these services? You should include a discussion of the concept of a "*future*", and its implementation in the Concurrency Utilities package.

QUESTION 6: CSP **[10 marks]**

Explain in detail how a buffer can be introduced between a producer process and a consumer process in CSP. There is a specific communication/coordination problem between the consumer and the buffer that must be solved — what is this, and how is it solved?

QUESTION 7: CSP/JCSP **[4+10 = 14 marks]**

a) What does the following CSP code do?

```
* [  x : integer ; A ? x  →  C ! x * x
   □ x : integer ; B ? x  →  C ! x * x ]
```

b) Convert this code into JCSP (you do not need to give a syntactically perfect answer). Explain any significant differences between the CSP version and the JCSP approach.

QUESTION 8: Remote Procedure/Method Calling **[15 marks]**

Explain in detail how you would use Java's Remote Method Invocation (RMI) in order to implement a simple share price service for a stock exchange. A client program should send a three letter code (a string, representing a company listed on the stock exchange; e.g. "FNB", or "PNP") to the service, which should respond with the current share price in cents (a double value).

QUESTION 9: CSP Metalanguage **[8 + 6 = 14 marks]**

a) Each customer of a bank first opens an account. They then make any number of deposits and withdrawals, and finally terminate their account. Let us initially ignore the amount of each deposit or withdrawal, and not worry whether the account is in credit or debit. The alphabet of the account is therefore:

α ACC = {*open, deposit, withdraw, terminate*}

Construct the process ACC.

b) Add one new process in parallel to your process ACC from part (a), that effects the following change to the bank scenario. The bank shuts at 3.00 p.m. every day, and does not reopen again until 9.00 a.m. the following morning. No transactions on the account are possible during this interval. Introduce two new events {*shut, reopen*} and write a process that prevents any transactions from occurring while the bank is closed. The alphabet of the new parallel process should be:

α ACC ∪ {*shut, reopen*}

END OF EXAMINATION PAPER