RHODES UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

EXAMINATION: NOVEMBER 2019

COMPUTER SCIENCE HONOURS
PAPER 1
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 **4 pages** and **10 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 [15 marks]

During the course we discussed five different hardware approaches used in parallel and distributed processing: *multicore processors*, *symmetric multiprocessors*, *heterogeneous processors*, *clusters*, and *supercomputers*. Describe and discuss any **two** of these five approaches. In terms of the Candidate Type Architecture (CTA) what is the critical performance factor that distinguishes these systems.

QUESTION 2: Theory [12 marks]

During the course we discussed a number of parallel programming patterns, including the *Divide and Conquer Pattern*, the *Pipeline Pattern* and the *Replicated-Worker Pattern*. Describe and discuss any **two** of these three patterns.

QUESTION 3: Theory [6 marks]

You have an application where 25% of the application cannot be parallelised. What is the maximum speedup you might expect with 200 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: Theory [10 marks]

Outline the differences between distributed and parallel processing. Explain any terms you use, and mention examples.

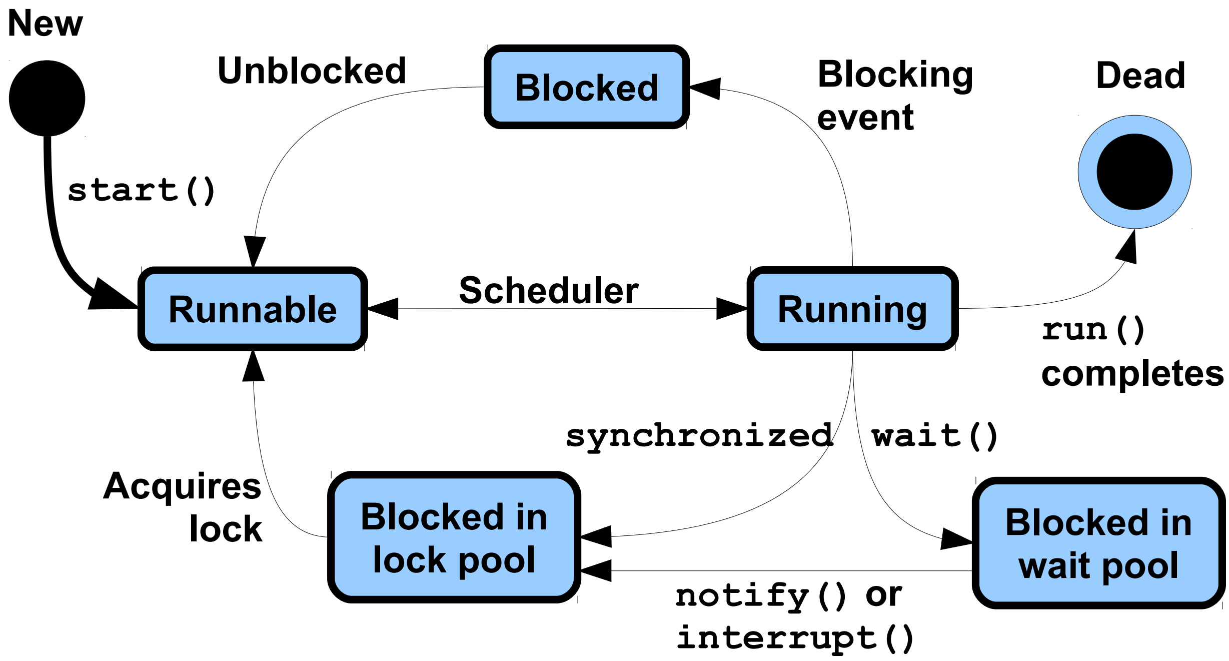
QUESTION 5: Java Threads [20 marks]

EITHER:

The `java.util.concurrent` package contains several different categories of classes useful in writing multi-threaded programs. Discuss these categories, giving examples of the classes in each category and their intended uses.

OR:

Threads in Java follow the life-cycle depicted in the diagram below. Discuss the life-cycle of a Java thread in detail. In your answer you should discuss the kinds of events that can lead to transitions between the states, and pay particular attention to the method of synchronisation between threads (you may wish to use short code examples to illustrate your answer).



QUESTION 6: CSP/occam/JCSP **[10 marks]**

Discuss the relationship between CSP, occam and JCSP, including the points of difference. What are some of the advantages of using JCSP rather than the standard Java thread facilities?

QUESTION 7: OpenMP **[3+8+2+3 = 16 marks]**

a) What is the effect of executing the following code fragment using OpenMP (i.e. what value is it calculating)?

```
int t = 0;
#pragma omp parallel for reduction(+:t)
for (int k = 1; k <= N; k++)
  { t += k;
  }
```

b) Explain the bold line in the code above in detail.

c) How many threads will be created to execute the above code fragment? How is this number determined?

d) What are the advantages of using OpenMP compared to other forms of multithreading parallelism?

QUESTION 8: Remote Procedure/Method Calling **[10+2 = 12 marks]**

a) Explain in detail how *remote procedure/method calls* are implemented for distributed systems.

b) What is the role of the *registry* in Java's RMI?

QUESTION 9: CSP **[7 marks]**

Explain what the following CSP code does, and what the various features of the CSP syntax used here mean.

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

QUESTION 10: CSP **[6+6 = 12 marks]**

Each customer of a bank first opens an account. He or she then makes any number of deposits and withdrawals, and finally terminates his/her 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}

a) Construct the process ACC.

b) The bank shuts at 3.00p.m. every day, and does not reopen again until 9.00a.m. the following morning. Only deposits are possible during the interval. Introduce two new events {shut, reopen}, and write a process that, when added in parallel with ACC, prevents any other events from happening between these two. The alphabet of the new process should be: α ACC ∪ {shut, reopen}

END OF THE EXAMINATION