PAPER CODE NO.	EXAMINER : **Dr D Jackson**
COMP204	DEPARTMENT : Computer Science Tel. No. 43678

THE UNIVERSITY
of LIVERPOOL

SUMMER 2002 EXAMINATIONS

Bachelor of Arts : Year 2
Bachelor of Engineering : Year 2
Bachelor of Science : Year 1
Bachelor of Science : Year 2

COMPUTER SYSTEMS AND THEIR IMPLEMENTATION

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer ***FOUR*** questions

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY
of LIVERPOOL

1.
(a) What is meant by a *context switch*? What actions occur during a context switch? (6 marks)

(b) With the aid of suitable diagrams, describe the three scenarios for program execution under UNIX. For each scenario, describe the system calls used to implement it.
(12 marks)

(c) Which of the three scenarios you have just described correspond to the execution of the following UNIX commands? Explain your answers.
(i) javac prog1.java
(ii) javac prog1.java &
(7 marks)

2.
(a) A Java object called *account* is used to represent a bank account. The balance of the account is currently £100, represented as an integer variable that is private to the object. The object also has a method called *deposit*, used to add money to the account.
Suppose the customers of the bank are represented as Java threads. Thread *Customer1* wishes to make a deposit of £25 with the following method call:

```
account.deposit(25);
```

while the thread *Customer2* makes a similar call:

```
account.deposit(50);
```

What will be the balance of the account after these calls have been made? Explain your answer carefully. (7 marks)

(b) What is meant by a *semaphore*, and how are they used?
Explain how semaphores could have been employed in the programming situation given in part (a).
(8 marks)

(c) The following two scenarios show a semaphore being used by two threads T1 and T2. Explain what is wrong with each of the two attempts, and give the possible consequences of using them in this manner.
(i)

```
      T1                          T2
P(s);                       P(s);
  critical region             critical region
P(s);                       V(s);
```

(ii)

```
      T1                          T2
V(s);                       P(s);
  critical region             critical region
P(s);                       V(s);
```

(10 marks)

3.
(a) One of the problems associated with a simple store management system, in which programs are allocated to contiguous partitions in order, is that it does not cater well for processes wishing to expand their storage needs. Give two reasons why a process may need to expand in this way. Describe two other problems of simple management schemes like this.
(6 marks)

(b) Describe three policies for selecting an empty partition into which to load a waiting program. Suppose a new program requires 200K of memory, and the following empty partitions are Available:
95K, 400K, 210K, 150K, 5K, 800K, 320K
Which partition would you use for each of your selection strategies?

(6 marks)

(c) Explain what is meant by *segmentation*, and describe how it is used on a typical system. What are the advantages of segmentation over the simple linear system? What information should be used in deciding how to segment a program, and what can happen if the segments are poorly chosen?

(8 marks)

(d) Briefly, what are the main differences between segmentation and *paging*?

(5 marks)

4.
Below is the grammar for a hypothetical programming language:

1.	<program>	::=	begin <stat-list> end
2.	<stat-list>	::=	<statement> { <statement> }
3.	<statement>	::=	id := <expr> ;
4.	<statement>	::=	read (<id-list>) ;
5.	<statement>	::=	write (<expr-list>) ;
6.	<id-list>	::=	id { , id }
7.	<expr-list>	::=	<expr> { , <expr> }
8.	<expr>	::=	<primary> { <addop> <primary> }
9.	<primary>	::=	(<expr>)
10.	<primary>	::=	id
11.	<primary>	::=	intliteral
12.	<addop>	::=	+
13.	<addop>	::=	-

(a) By drawing the appropriate derivation tree (parse tree), show how you could apply the productions above to parse the following program:

```
begin n := (x - 1) + (y - 1 + z); end
```

(9 marks)

(b) What is meant by an Abstract Syntax Tree (AST), and what is its role?
Convert your derivation tree from (a) into an AST.

(8 marks)

(c) Describe the general approach to generating code from an AST. Show how it applies to your own AST, and give the code generated from your program in the form of *tuples*.

(8 marks)

5.

(a) Explain what is meant by a *network protocol*, and say why they are needed. (5 marks)

(b) Draw a diagram of the ISO 7-layer protocol stack for network communication. Describe the role of each layer of the ISO stack in sending a message from one process to another.

(15 marks)

(c) Name a common alternative to the ISO model, and explain why it is more widely used in real network implementations. (5 marks)