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

THE UNIVERSITY
of LIVERPOOL

SUMMER 2001 EXAMINATIONS

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

COMPUTER SYSTEMS AND THEIR IMPLEMENTATION

(Special Resit Paper – 1999-2000 Syllabus)

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer **eight** questions from Section A and **two** questions from Section B.

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).

SECTION A

Give brief and relevant answers to any **eight** of the following ten questions:
[5 marks each]

(1) On a typical multi-user computer, why might an increase in the amount of main memory result in improved system performance?

(2) What is meant by a context switch?

(3) Explain carefully the difference between the C expressions `*p++` and `(*p)++`

(4) What is the reason for passing back a status value when a program terminates, and how should this be done?

(5) A UNIX file's permissions are encoded as the octal number 0751. What does this mean in terms of the permissions that are set for this file, and what difference would it make if the file was a directory?

(6) A link to a program simply gives that program an additional name. How would the program be written so as to be able to perform different actions according to the name used to execute it?

(7) Describe in detail the actions performed by a terminal device driver when a user types several characters at the keyboard, followed by a backspace, and finally the return key.

(8) What should a command that normally expects an input file as its argument do if the filename is not present on the command line? Explain the reason for this, and describe how it can be achieved in the code.

(9) Describe the steps involved in creating two processes connected by a pipe.

(10) What is meant by a compiler-compiler, and why may the term be misleading?

SECTION B

This section calls for fuller solutions than those of the previous section.

Answer any **two** of the following three questions:

[30 marks each]

B1

(a) Explain how a UNIX process can send a signal to another process, and describe the three options that a process has for specifying how it will react on receipt of a signal. [9 marks]
Which signal will not respond to any such request to alter its action, and why is it provided? [4 marks]

(b) A call to the *alarm( )* routine requests the operating system to send the SIGALRM signal to the process at some specified number of seconds in the future. Using this, show how you might develop a program which gives an indication of the computer's relative speed of execution. (Do not worry about getting the C syntax exactly right). [11 marks]
Would your program be affected if it was run as a background process? If so, how would you detect and deal with this? [6 marks]

B2

The head of a software company asks you to write a high-level I/O library. You propose to make use of software buffering in your routines.

(a) Explain how your buffering mechanism would work, and describe its advantages. [6 marks]

(b) Now describe the potential difficulties that it may cause, with particular regard to the following [5 marks each]:

(i) debugging of program code via the insertion of output statements
(ii) filestore inconsistency problems
(iii) system calls that replace one executing program with another
(iv) system calls that spawn child processes

For each of the above, suggest a simple remedy. [4 marks]

B3

The writer of a compiler intends to make use of a lexical analyser generator, but needs help with specifying the regular expressions defining the language tokens.

(a) Define regular expressions for each of the following lexemes, explaining how you derive them. [3 marks each]

(i) reserved words consisting of upper case letters only, e.g. INT, WHILE
(ii) identifiers consisting of a single lower case letter followed by any number of lower case letters, digits and underscores; e.g. fred7, t_o_m_
(iii) hexadecimal constants, specified by using a leading 0x or 0X; e.g. 0x9, 0X3A, 0x3aBc
(iv) unsigned floating point numbers, which must have at least one digit before and after the decimal point; e.g. 0.2, 3.14159

(b) Now define a regular expression for the *#include* directive of the C pre-processor. The directive takes the form

#include <file.h>

Assume that white space may not appear between the # and the *include*, or anywhere inside the angle brackets, but it may appear elsewhere.
Your expression should also specify that the directive appears by itself on a line, and that the filename must end in the ".h" characters; the only other characters allowed in the filename are lower case letters and digits.
Carefully explain how you derive your regular expression. [9 marks]

(c) Briefly describe the phases of compilation that take place after lexical analysis. [9 marks]