)006

0109

00059

THE OPEN UNIVERSITY OF SRI LANKA
BACHELOR OF SOFTWARE ENGINEERING - LEVEL 5
ECX5263 COMPUTER ORGANIZATION AND OPERATING SYSTEMS
FINAL EXAMINATION 2014
DURATION : THREE HOURS

DATE : 19th September 2015 **TIME : 0930 - 1230 HOURS**

Answer any *five* questions. All questions carry equal marks.

1.

(i) Add the following numbers as fixed-point integers. Your calculations must be shown by using binary numbers in two's complement. Choose appropriate word length.

(a) -1.25 + (-21.5)

(b) -2.625 + 4.125

(ii) Represent all the above numbers as floating point numbers of the following format using biased exponent.

S (1 bit)	Exponent (8-bit)	Mantissa (23-bit)

(iii) In a source code a programmer declared an array of integers. All elements of the array are signed 16-bit integers. Your task is to hide a text in English using this integer array and all characters of the text are in extended ASCII codes (8 bits). An integer number represents two characters and they are written in consecutive bytes of the element.

(a) Show that after hiding a text in English the array contains only positive numbers.

(b) What will be decimal numbers of the array if you hide the text OUSL? The ASCII code of the each letter is 79, 85, 83 and 76 in decimal respectively.

(c) Draw a diagram to show how the given text is stored in the memory. Assume that the memory is byte addressable.

2. An enhancement in a computer system improves only some part of the system. Accordingly improvement of the performance depends on the impact of the enhanced part. The f denotes the fraction of the computation time in the old system that can be improved with the enhancement made; S_e is the achievable speedup only if the enhanced part of the system is used.

(i) If the old time of the system (without improvement) is T_{old} formulate the new time T_{new} of the system after the enhancement.

(ii) The speedup of the new system (after the improvement) is

$$S_{new} = \frac{T_{old}}{T_{new}}.$$

Accordingly derive an equation for S_{new} in terms of f, S_e, which is Amdahl's law.

(iii) The MIPS rating of a processor is 1000. However the processor requires at least one memory access per instruction. The memory latency of the system is 10 ns. If you are able to double the MIPS rating of the processor what is the achievable overall speedup

of the system. Assume that there is no change in memory latency. You have to show how the Amdahl's law can be used for solving the problem.

3.

(i) Draw typical virtual machine schematic for Memory-Memory Architecture and briefly describe the functionality of it.

(ii) Compare and contrast Memory-Memory architecture with Accumulator architecture considering performance, size of the processor and size of the code of a program.

(iii) Write a program using the ISA (given in the Appendix) of the Accumulator architecture of the Students' Experimental Processor (SEP) to find the total (X) of all negative values of an array, i.e.

$$X = \sum_{i=1}^{n} A_i \text{ ; where } A_i < 0.$$

Assume that all elements of the array and the value n are stored in the memory.

4.

(i) What are the necessary conditions needed to exist a deadlock.

(ii) What are the main objectives of a resource manager?

(iii) Explain how you prevent the occurrence of a deadlock.

(iv) Draw a Resource Allocation Graph for the following.
There are 3 Processes (P1, P2, and P3) and 4 different types of Resources (R1, R2, R3, and R4). Instances for each resource are 1, 1, 2, and 3 respectively.

- P1 requests an instance of R1 and is holding an instance of R3.
- P2 requests an instance of R3 and is holding one instance of R1 and one instance of R2.
- P3 requests an instance of R3 and R2, and is holding an instance of R4.

(a) Is there any chance to occur a deadlock? Show how it could happen.

(b) Propose a method to avoid the deadlock at the current situation.

5.

(i) Describe the two types of data rate: *media data rate* and *interface data rate*.

(ii) Consider specifications of a typical disk as follows. The advertised average seek time is 9ms, the transfer rate is 4MB/s, rotates at 7200 rpm, and the controller overhead is 1ms. Assume the disk is idle so that there is no queuing delay.

(a) What is the average time to read or write a 512-byte sector?

(b) If the measured seek time is 33% of the calculated average, what is the average time to read or write a 512-byte sector?

(iii) Briefly describe three methods which improve the average disk accessing time.

6.

(i) Name three organizations of cache memory and describe them briefly.

(ii) Briefly describe the tasks of an I/O system of a computer.

(iii) Write policies are needed for data caches when an instruction writes data to the cache. Briefly describe the Write Back and Write Through policies.

(iv) A colour display has maximum resolution of 640x480 pixels. It displays 256 different colours simultaneously. The display should be refreshed normally at the rate of 60 times per second.

(a) How many bits does it need to save information of colours per pixel?

(b) What is the minimum size of the display buffer memory?

(c) What would be the data rate that the display works properly?

7.

(i) Describe the life cycle of a process.

(ii) Describe the technique, Context Switching.

(iii) Consider the following program segments for two different processes executing concurrently

P1	**P2**
For A:=1 to 3 do	For B:=1 to 3 do
x:= x + 1;	x:= x + 1;

In the above processes A and B are not shared variables, but x is a shared variable and starts at zero.

(a) If the processes P1 and P2 execute only once at any speed, what are the possible values of x? Explain your answers.

(b) Give suitable data structures of Process Control Blocks (PCB) for each process considering only the information provided.

8.

(i) Distinguish between *Fixed Partition Multiprogramming* and *Dynamic Partition Multiprogramming*.

(ii) A computer system has a memory with a capacity of 1 Mbytes. The system loads the operating system into the memory starting from the lowest memory address. The job queue of the system is full of jobs that require 180, 340, 100, 200, and 50 Kbytes. Operating system needs only 50 Kbytes of memory.

(a) Show how jobs will be allocated in the memory for the *Dynamic Partition Multiprogramming*.

(b) The jobs in the size of 100 and 50 Kbytes have finished their jobs and freed their allocated memory. How can a new job needing 150 Kbytes be allocated in the memory.

(c) If *Paged Memory Allocation* is deployed in the computer system show how the jobs are allocated in the memory (assume page size as 64 Kbytes).

(d) What is your answer for the part (b) when *Paged Memory Allocation* is deployed?

(e) Calculate the amount of unusable memory due to internal fragmentation.

Appendix

ISA of Accumulator Architecture

Arithmetic instructions		
ADD	Addition	Acc ← Acc + op
SUB	Subtraction	Acc ← Acc - op
MUL	Signed multiplication	Acc (16 bit) ← Acc(8 LSBs) * op(8 LSBs)
DIV	Unsigned division	Acc ← Acc / op2
INC	Increment by 1	Acc ← Acc + 1
Logical instructions		
AND	Bit-wise And	Acc ← Acc AND op
OR	Bit-wise OR	Acc ← Acc OR op
XOR	Bit-wise XOR	Acc ← Acc XOR op
SHL	Shift left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & 0
SHR	Shift right by 1-bit	CF ← Acc (LSB), Acc ← 0 & Acc (15 down to 1)
ROL	rotate left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & CF
ROR	rotate right by 1-bit	CF ← Acc (LSB), Acc ← CF & Acc (15 down to 1)
NOT	One's compliment negation	op1 ← NOT Acc
Control Transfer instructions		
Conditional Branches		
JC	Jump if carry	If CF =1 then IP ← IP + Operand
JOF	Jump if over-flow	If OF =1 then IP ← IP + Operand
JS	Jump if Sign	If SF =1 then IP ← IP + Operand
JP	Jump if parity	If PF =1 then IP ← IP + Operand
JZ	Jump if result is zero	If ZF =1 then IP ← IP + Operand
Unconditional branch		
JUMP	Jump	IP ← IP + Operand
Loops		
LOOZ	Loop until zero	Count ← Count – 1 IF Count =0; Loop termination ELSE; IP ← IP + operand
Calls and Returns		
CALL	Procedure call	implied return address ← IP IP ← Immediate address
RETURN	Return from procedure	IP ← Contents of implied return address
Miscellaneous instructions		
NOP	No operation	
Data Movement instructions		
LOADacc	Copy the operand to the accumulator	Immediate: Acc ← op Direct: Acc ← memory (op)
STOREacc	Copies the accumulator to the memory address	Direct: Memory (op) ← Acc Indirect : Memory {memory (op)} ← Acc