15

University of Colombo, Sri Lanka

University of Colombo School of Computing

Bachelor of Science in Computer Science

Academic Year 2017-2018 — First Year Examination — Semester II

SCS1214 — Operating Systems I

(2 Hours)

Answer All Questions

Number of Pages = 14 Number of Questions = 4

To be completed by the candidate

Index Number | | | | | | | | |

Important Instructions

- The duration of the paper is **2** Hours.
- The medium of instructions and questions is English.
- This paper has **4** questions on **14** pages.
- Answer **all** the **4** questions.
- **Write your answers on and only on the space provided** on this question paper.
- Do not tear off any part of this answer book. Under no circumstances may this book (or any part of this book), used or unused, be removed from the Examination Hall by a candidate.
- Questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
- Any electronic device capable of storing and retrieving text, including electronic dictionaries and mobile phones, are **not allowed**.
- Non-programmable Calculators may be used.

To be completed by the examiners

1	
2	
3	
4	
Total	

Index Number | | | | | | | | |

1. (a). The following, incomplete, code segment is written in the assembly instructions of the virtual machine used for the assignments of the SCS1214 course.

```
0 movv sp 90
1 movv a 120
2 movv b 180
3 call 20
4 add a b
5 out acc
6 halt

20 push b
21
22 push acc
23 movv a 25
24 movv b 125
25 add a b
26 out acc
27
28 pop a
29 pop b
30 ret
```

i. What is the assembly instruction that should be in the memory location 21?

[3 marks]

ii. What is the assembly instruction that should be in the memory location 27?

[3 marks]

iii. Assume that the completed program has executed on the virtual machine. What is the output?

[3 marks]

Index Number | | | | | | | | |

iv. What is the content of the memory at address 91 just before the instruction at the memory address 30 is executed? Justify your answer.

[5 marks]

(b). Assume that a process transits between the following states: *new*, *ready*, *running*, *waiting* and *terminated*. Some of the state transitions of a process has been observed and given bellow.

$$\ldots X \rightarrow \ldots X \rightarrow Y \rightarrow Z \rightarrow terminated$$

The states X, Y and Z are different states. The unobserved states are given as $\ldots$.

i. What is X?

[2 marks]

ii. What is Y?

[2 marks]

iii. What is Z?

[2 marks]

Index Number | | | | | | | | |

(c). Following program is compiled and executed on an $x86$ machine running Linux.

```
int main()
{
  int x;

  x=fork();
  x=fork();

  if(!x) printf("%d\n", !x+2);

}
```

What is the output of this program?

[5 marks]

Index Number | | | | | | | | |

2. (a). A program P has the statement X. Multiple processes executes the program P concurrently. At most 10 processes are allowed to execute the statement X concurrently. Write a code segment of P, in C-like pseudo code, to use semaphores to ensure that at most 10 processes are allowed to execute the statement X concurrently.

[5 marks]

(b). A System has 12 instances of the resource type R and three processes, P_0, P_1, P_2, that require R. The maximum requirements of R for each process and the current allocations at time t_0 are given in the following table.

	Maximum Need	Current Allocation
P_0	10	6
P_1	4	2
P_2	9	2

The system is in a safe state at t_0. Give a safe sequence.

[4 marks]

Index Number | | | | | | | | |

(c). P_0 and P_1 are two concurrent processes and S and Q are semaphores.

P_0

```
wait(S);
wait(Q);
// Critical Section
signal(Q);
signal(S);
```

P_1

```
wait(Q);
wait(S);
// Critical Section
signal(S);
signal(Q);
```

Draw a resource allocation graph to depict a possible deadlock involving P_0 and P_1.

[5 marks]

(d). The structure of the producer process of the bounded-buffer problem is given below.

```
do{
  // Produce an item
  wait(empty);
  wait(X);

  // add the item to the buffer

  signal(mutex);
  signal(full);
}while (TRUE);
```

The buffer used by the producer and the consumer has 100 slots.

Index Number | | | | | | | | |

i. What is X?

[2 marks]

ii. What is the initial value of `empty` before any process started producing or consuming items?

[2 marks]

iii. What is the initial value of `full` before any process started producing or consuming items?

[2 marks]

iv. Give the code for the consumer process.

[5 marks]

Index Number | | | | | | | | |

3. (a). Explain why Operating Systems have to manage the main memory.

[4 marks]

(b). What are the different types of memory that can be found in a typical computer? Compare their storage capacities and access speeds.

[4 marks]

Index Number | | | | | | | | |

(c). Explain the main reason for introducing cache memory to computer systems.

[4 marks]

(d). Explain four main functions an Operating System is expected to perform in managing the main memory.

[4 marks]

Index Number | | | | | | | | |

(e). Explain two different techniques used by the Operating Systems to manage free memory.

[5 marks]

Index Number | | | | | | | | |

(f). What is the memory management mechanism used by the IBM 360 machines to keep multiple programs in the main memory at the same time?

[4 marks]

Index Number | | | | | | | | |

4. (a). If the memory is byte-addressable and a process is given 2GB virtual memory, what is the maximum number of pages of size 16KB a process can have in its virtual memory? Justify your answer.

[4 marks]

(b). Explain how a virtual address is mapped to a physical address in an operating system that provides a virtual memory.

[5 marks]

Index Number | | | | | | | | |

(c). Let a process generates the following page reference string during its execution.

$$7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 7, 0, 1, 7, 0, 1$$

If the process is allowed 3-page frames how many page faults occurs if Demand Paging Optimal Page Replacement algorithm is used? Justify your answer.

[8 marks]

Index Number | | | | | | | | |

(d). Give four (4) main system calls provided by an operating system to handle files. Explain the main actions to be carried out by the operating system for each system call you have specified.

[8 marks]

****