UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Academic Year 2016/2017 – Second Year Examination – Semester II – 2018

SCS2106 – Operating Systems I – (Part B)

TWO (2) HOURS (For both parts A & B)

To be completed by the candidate

Examination Index No: .....

Important Instructions to candidates:

1. The medium of instruction and question is **English.**
2. If a page or a part of this question paper is not printed, please inform the supervisor immediately.
3. Note that questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
4. Write your index number on each and every page of the Question paper.
5. This part of the paper has **02** questions in **09** pages.
6. Answer **ALL** questions. All questions carry equal marks (25 marks).
7. **This paper consists of two parts, Part A (Question No 1 and Question No 2) and Part B (Question No 3 and Question No 4) and submit separately.**
8. Any electronic device capable of storing and retrieving text including electronic dictionaries and mobile phones are **not allowed.**
9. **Non-Programmable** calculators are **allowed.**

For Examiner's use only	
Question No	**Marks**
3	
4	
Total	

Index No: .....

3. (a) All operating systems are using threads for different purposes. Write down four (4) reasons for the use of threads in operating systems.

[4 marks]

(b) Threads, single-threaded processes and finite-state machines are 3 methods of constructing a typical server. Write down the characteristics of each method.

[3 marks]

Using threads

Using single-threaded processes

Using finite-state machines

(c) A virus-guard is a software program. A lecturer has decided to assign you to implement a virus guard as a take-home assignment. You can implement it as a single-threading environment or as a multi-threading environment. What is the environment that you will choose for the implementation? Briefly explain the reasons to justify your selection.

[3 marks]

The environment that you are going to select for the implementation

The reasons

(d) What is the difference between user-level threads and kernel-level threads?

[2 marks]

(e) Long-term information storage is an essential requirement in operating systems. To fulfil that requirement, different file system implementations are introduced. One such example is an i-node implementation.

Explain how the path name */cygwin64/lib/gcc/x86_64-pc-cygwin* is searched in an i-node based system.

[13 marks]

Assumptions

root directory is in block 7
cygwin64 directory is in block 133
lib directory is in block 190
gcc directory is in block 220

cygwin64's i-node – 10
lib's i-node – 16
gcc's i-node – 70
x86_64-pc-cygwin's i-node – 82

Index No: .....

Index No: .....

4. (a) Peterson's solution is coming up under the solutions for mutual exclusion. Implementation of Peterson's solution is as follows:

```
#define FALSE 0
#define TRUE 1
#define N 2 /* number of processes */

int turn;
int interested[N];
void enter region(int process) {
   int other = A;
   interested[process] = TRUE;
   turn = process;
   while (B && C);
}

void leave region(int process) {
   D;
}
```

(i) What is A?

[3 marks]

(ii) What is B?

[3 marks]

(iii) What is C?

[3 marks]

(iv) What is D?

[3 marks]

(b) Readers-Writers problem is one of the famous classical inter-process communication (IPC) problems. Using semaphores, this problem can be solved. The implementation of the solution using semaphores is as follows:

Variable declaration and initialization

```
typedef int semaphore; /* use your imagination */
semaphore mutex = 1; /* controls access to rc */
semaphore db = 1; /* controls access to the database */
int rc = 0; /* # of processes reading or wanting to */
```

Function for Reader

```
void reader(void) {
     while (TRUE) {
          E;
          rc = F;
          if (rc == 1) down(&db);
          G;
          read data base( );
          H;
          rc = rc - 1;
          if (rc == 0) up(&db);
          I;
          use data read( );
     }
}
```

Function for Writer

```
void writer(void) {
     while (J) {
          think up data( );
          down(K);
          write data base( );
          up(L);
     }
}
```

(i) What is E?

[2 marks]

(ii) What is F?

[2 marks]

(iii) What is G?

[2 marks]

(iv) What is H?

[2 marks]

(v) What is I?

[2 marks]

(vi) What is J?

[1 marks]

(vii) What is K?

[1 marks]

(viii) What is L?

[1 marks]

****