University of Colombo, Sri Lanka

University of Colombo School of Computing

Bachelor of Science in Computer Science

Academic Year 2014-2015 — Second Year Examination — Semester II

SCS2106 — Operating Systems I

(2 Hours)

Answer All Questions

Number of Pages = 11 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 **11** 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). Assume that a process transit between the following states: *new*, *ready*, *running*, *waiting* and *terminated.*

i. A process in state S is moved to the *ready* state on receiving a timer interrupt. What is S?

[1 marks]

ii. A process is in the *running* state. What are the possible next states for this proces?

[3 marks]

(b). What is the main difference between a program and process?

[3 marks]

Index Number □□□□□□□□

(c). i. How many X's are printed by the following program?

```
int main()
{

        if (fork())
        {
          printf("X\n");
          if(fork())
              printf("XX\n");
        }
        fork();
        printf("X\n");
        return 0;
}
```

[5 marks]

ii. What is the output of the following program?

```
int main()
{
        int x;
        x=0;
        if (fork())
            x++;
        if(fork())
            x++;
        printf("%d",x);
}
```

[5 marks]

Index Number

(d). Consider the following *C* program.

```
int * f(int x)
{
  int p;
  p=x;
  return &p;
}


int * g(int x)
{
  int y;
  y=x;
  return &y;
}

int main()
{

  int *x,*y;
  x=f(100);
  y=g(2500);
  *x=*x+500;
  printf("%d\n", *y);
  return 0;
}
```

i. What is the output of the above program?

[4 marks]

ii. Assume that the above program is changed by interchanging the lines with `x=f(100);` and `y=g(2500);` as follows?

```
y=g(2500);
x=f(100);
```

What is the output of the above program under this assumption?

[4 marks]

Index Number

2. (a). Draw a resource allocation graph with a cycle, that **does not indicate a deadlock**. Use only two processes and two resource types.

[4 marks]

(b). A System has 11 instances of the resource type R and three processes, P_0, P_1, P_2, P_3, 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	9	5
P_1	9	2
P_2	4	2
P_3	11	0

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

[4 marks]

ii. At time t_1 another process, P_4, is added to the system with maximum need of 10 number of resource R. It has not been allocated any instance of R at t_1. Is the system in a safe state at t_1? If it is in a safe state give a safe sequence. Otherwise justify your answer.

[4 marks]

Index Number

iii. Assume that at time t_2, a single instance of R is given to the P_3. Is the system in a safe state at t_2 under this assumption? If it is in a safe state give a safe sequence. Otherwise justify your answer.

[3 marks]

iv. Assume that there are more resource allocations at time t_3 and the system is not in a safe state at time t_3. Does this also indicate that the the system is deadlocked at t_3? Justify your answer.

[4 marks]

(c). Consider a system with 6 processes ($P_0, P_1, P_2, P_3, P_4, P_5$) and 3 resource types (**A, B, C**). There are 10 instances of **A**, 5 instances of **B**, and 7 instances of **C**. The current resource allocation and the maximum need of processes are given in the following table.

	Allocation			Maximum Need		
Process	A	B	C	A	B	C
P_0	0	1	0	7	5	3
P_1	2	0	0	3	2	2
P_2	3	0	2	9	0	2
P_3	2	1	1	2	2	2
P_4	0	0	0	10	5	7
P_5	0	0	2	4	3	3

Give a safe sequence.

[6 marks]

Index Number

3. (a). A solution to the bounded-waiting mutual exclusion problem that uses `TestAndSet()` is given bellow. This code is given with respect to the process `i`.

```
do{
 waiting[i] = TRUE;
 key = TRUE;
 while(waiting[i] && key)
   key = TestAndSet(&lock);
 waiting[i] = X;

   // critical section

 j = (i + 1) % n;
 while ((j != i) && !waiting[j])
    j = Y;

 if (j == i)
    lock = Z;
 else
    waiting[j] = R;

   // remainder section

 } while (TRUE);
```

i. What should be the initial value of the shared variable `lock`?

[3 marks]

ii. What is `X`?

[3 marks]

iii. What is `Y`?

[3 marks]

Index Number

iv. What is `Z`?

[3 marks]

v. What is `R`?

[3 marks]

(b). The structure of process P_i in Peterson's solution is given bellow.

```
do{
 flag[i] = TRUE;
 turn = j;
 while (flag[L] && turn == M);
    // critical section

 flag[i] = N;
 } while (TRUE);
```

i. What is `L`?

[2 marks]

ii. What is `M`?

[2 marks]

iii. What is `N`?

[2 marks]

Index Number

(c). The structure of process P_0 is given bellow. S and Q are semaphores initialized to 1.

```
wait(S);
wait(Q);

.....
```

Write a code fragment with just two lines for process P_1 using the same semaphores to have a possibility of a deadlock.

[4 marks]

Index Number | | | | | | | | |

4. (a). A system uses memory pages of size s bytes. The memory is byte addressable. The page table entry of the page 0 of the process P in this system is stored at the physical address q. Each page table entry is r bytes long. The function $F()$ takes the address of the page table entry and returns the frame number stored there. For example, $F(q)$ returns the frame number corresponding to the page number 0.

You can assume that the / operator applied to two integers gives the quotient with any fractional part discarded (similar to the integer division in C) and the % gives the remainder of the division.

i. What is the page number of the logical address x?

[3 marks]

ii. What is the offset within its page of the logical address x?

[3 marks]

iii. What the physical address of the page table entry corresponding to the page number of the address x of the process P?

[5 marks]

iv. What is the physical address of the logical address x of the process P?

[5 marks]

Index Number □□□□□□□□

(b). What is external fragmentation with reference to memory management?

[2 marks]

(c). A FAT file system uses blocks of size 4096 bytes. Each FAT table entry is 32 bits long. What is the largest disk that can be handled by this file system?

[4 marks]

(d). Name three file systems other than the FAT file system.

[3 marks]

****