NATIONAL UNIVERSITY OF SINGAPORE

CS2106 – INTRODUCTION TO OPERATING SYSTEMS

(Semester 1: AY2014/15)

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS

1. This assessment consists of **sixteen (16)** questions and **sixteen (16)** printed pages, including this page.
2. This is a **CLOSED BOOK** assessment. You may use any approved calculators but not any PDA or laptop, especially those capable of external connectivity or communication.
3. Write your answers on this **QUESTION AND ANSWER SCRIPT**. Answer only in the spaces given.

STUDENT NO: _____

Question	Maximum	Earned
1	12	
2	12	
3	12	
4	12	
5	12	
6	12	
7	12	
8	12	

Question	Maximum	Earned
9	12	
10	12	
11	12	
12	12	
13	12	
14	12	
15	12	
16	12	
Total	192	

1. [12 marks] Consider the program shown below, where the a, b, …, h represent some arbitrary processes:

a) Draw the corresponding process flow graph

```
     n1 = 2; n2 = 3;
     fork A;
     fork B;
     quit;
A:   a; join n2, H; quit;
B:   b; fork C; fork D; quit;
C:   c; fork E; fork F; quit;
D:   d; join n1, G; quit;
E:   e; join n1, G; quit;
F:   f; join n2, H; quit;
G:   g; join n2, H; quit;
H:   h; quit;
```

a)

b) The program can be made more efficient by eliminating some of the forks while still preserving the original process flow graph.
How many such forks can be eliminated? _____

Show one possible list of such forks: _____

c) This program cannot be written using cobegin/coend. Modify the graph by moving one process to make it expressible using cobegin/coend. (Preserve all original dependencies. Show only the new graph, not the cobegin/coend program.)

c)

2. [12 marks] In the Dining Philosophers Problem, each of 5 philosophers, p(i), needs to pick up his left fork, f[i], and his right fork, f[i+1 mod 5], in some order. Consider the following 3 strategies for picking up the forks:

   A. Each p(i) picks f[i] first, then f[i+1 mod 5]
   B. Each p(i) picks the lower-numbered fork first, then the higher-numbered one.
   C. Each p(i) picks the odd-numbered fork first, then the even-numbered one. (If both are odd then pick the lower numbered first)

   Answer the following questions for the 3 strategies, A, B, C:
   a. Is deadlock possible?
   b. Is it possible to violate concurrency in that only 1 philosopher is eating while the remaining 4 are blocked?

Strategy	a. (yes/no)	b. (yes/no)
A.		
B.		
C.		

   c. If you answered yes in column b for any of the strategies then show a possible scenario by stating which forks each philosopher is holding and which fork it is requesting at the time of the violation:

	Strategy A	
	Holding	Requesting
p(1)		
p(2)		
p(3)		
p(4)		
p(5)		

Strategy B	
Holding	Requesting

Strategy C	
Holding	Requesting

3. [12 marks] Consider the following monitor, which implements a Critical Section (CS) that more than one process can enter at the same time:

```
monitor m {
        condition c1, c2, c3;

Enter_p1() {
        if (x ≠ 0) c1.wait
        else x = -1; }

Exit_p1() {
        x = 5; }

Enter_p2() {
        if (x ≠ 0) c2.wait
        else x = 2; }

Exit_p2() {
        x = 4;
        c3.signal; }

Enter_p3() {
        if (x ≥ 0) c3.wait; }

Exit_p3() {
        x = -3;
        c1.signal;
        c2.signal; }
}
```

Consider 3 processes, p1, p2, p3. Each process p_i repeatedly (and asynchronously) executes the sequence:

$$\text{Enter\_}p_i\text{(); CS; Exit\_}p_i\text{();}$$

a. Determine which processes can be in CS at the same and what is the value of x at that time:

Processes in CS	yes/no	x =
p1 and p2		
p1 and p3		
p1 and p2 and p3		
p2 and p3		

b. Determine all possible values of x when a given process is in CS by itself (i.e., the only process):

Process in CS	Possible values of x
p1	
p2	
p3	

c. Is deadlock possible with this monitor? _____ (yes/no)

If so, what initial values of x could cause it (give all possible values)? _____

4. [12 marks] Consider the following two implementations of P/V operations on semaphores:

A. When P cannot proceed, it uses busy-waiting
B. When P cannot proceed, it is blocked (not running); a V operation unblocks a process when the semaphore is above 0.

Assume 2 processes start executing the following sequence concurrently (CS is a critical section):

P(s); CS; V(s);

Make the following timing assumptions:

- Executing CS takes c time units
- To block a process takes b time units; to unblock a process also takes b time units
- Time to execute P or V when no blocking/unblocking is involved is negligible (zero).

a. Determine the total time necessary for both processes to execute the sequence P(s); CS; V(s):

Using implementation A: _____ (time units)

Using implementation B: _____ (time units)

b. Determine the total execution time when 3 processes concurrently execute the same sequence P(s); CS; V(s):

Using implementation A: _____ (time units)

Using implementation B: _____ (time units)

5. [12 marks] Consider a system of process and resources analogous to the first project of this course. At the beginning only the Init process is running. The following series of commands is then issued:

```
CR A 2
CR B 1
CR C 2
TO
CR D 1
TO
CR E 2
TO
REQ R1 1
TO
REQ R2 1
TO
REQ R1 1
REQ R2 2
```

Recall that CR stands for create a process, TO stands for timeout, and REQ stands for request a resource. Recall also that resource R1 has one unit and R2 has two units.

a. Determine how many child processes each process has at the end of the above command series.

A:_____ B:_____ C:_____ D:_____ E:_____

b. Which processes are blocked at the end (if any): _____

c. Which process is running at the end: _____

d. Which resource(s) is that process holding (if any): _____

6. [12 marks] Consider three processes, p1, p2, p3, with the following arrival times and total service times (t):

process	arrival	t
p0	0	5
p1	1	6
p2	2	2

a. Compute the average turnaround time for the three processes under:

- FIFO (first in, first out): _____
- SJF (shortest job first): _____
- SRT (shortest remaining time): _____

b. Assume MLF (multi-level feedback)with 5 priority levels and T=1. Complete the following diagram by showing which process is running at which priority level.

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14
priority 5	p0	p1	p2											
priority 4				p0										
priority 3														
priority 2														
priority 1														

c. Assume now that the processes are periodic with the periods: d0 = 15, d1 = 14, d2 = 13.

Is there a feasible schedule under EDF (earliest deadline first)? _____ (yes/no)
Why or why not:

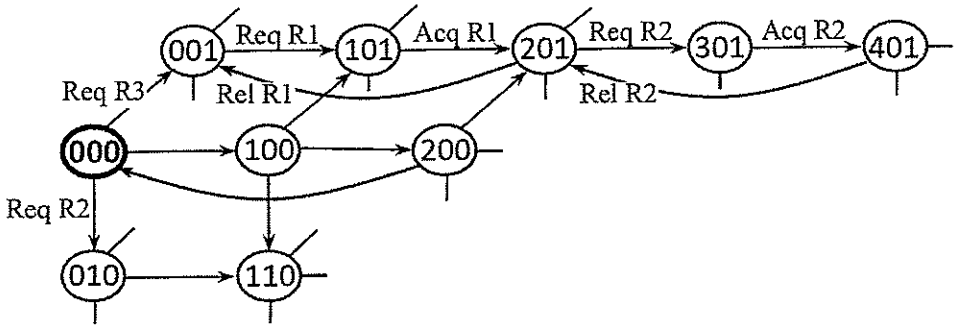
_____

d. Is RM (rate monotonic) likely to produce a feasible schedule? _____(yes/no)
Why or why not?

_____

7. [12 marks] Assume 3 processes, p1, p2, p3, and 3 resources, R1, R2, R3.
   - p1 first requests (and acquires) R1, then R2
   - p2 first requests (and acquires) R2, then R3
   - p3 first requests (and acquires) R3, then R1
   - all processes release their resources in the reverse order of acquisition

   The following figure shows a portion of the state transition diagram of the 3 processes, with 000 as the starting state.



a. In the matrix below, cross out all states that are unreachable:

002	102	202	302	402
012	112	212	312	412
022	122	222	322	422
032	132	232	332	432
042	142	242	342	442

b. In which of the above states is p1 blocked? _____

c. In which of the above states is p2 blocked? _____

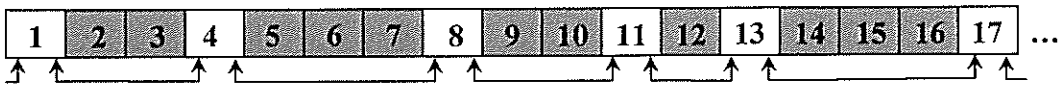
d. In which of the above states is p3 blocked? _____

e. Are any of the above states deadlock states? If so, which ones _____

f. Are any of the states in the entire system (i.e., 000 through 444) deadlock states?

   If so, which ones _____

8. [12 marks] The following diagram shows a portion of main memory where the shaded boxes are occupied blocks and the white boxes are free blocks (holes), connected by a doubly linked list.



Consider the two different implementations discussed in class:

A. Tags and sizes are only at the front of each block.
B. Tags and sizes are replicated at both ends of each block.

a. Assume that block 5 is to be released and answer the questions in the table:

	The tags/sizes of which blocks need to be **examined** to find out if the **right-hand** neighbor is free?	The tags/sizes of which blocks need to be **examined** to find out if the **left-hand** neighbor is free?	The **tags/sizes** of which blocks need to be **modified** to release the block?	The **pointers** of which blocks need to be **modified** to release the block?
A.				
B.				

b. Answer the same questions assuming that block 12 is to be released:

A.				
B.				

9. [12 marks] Three functions, each of length 600 bytes, are linked together into one process and loaded into memory. Consider the following memory management schemes:

   A. paging (no segmentation):
      page size: 1024 bytes
      page table occupies 1 page

   B. segmentation (no paging, each function becomes a separate segment):
      maximum segment size: 2048 bytes
      segment table size: 1024 bytes

   C. segmentation with paging (each function becomes a separate segment):
      page size: 1024 bytes
      page and segment tables occupy 1 page each

   Assume that all three functions as well as all tables are resident in memory. For each of the three systems, determine:

   a. the total occupied memory space (including all tables)
   b. the amount of space wasted due to internal fragmentation
   c. the sizes of s, p, and w (enter n/a if not applicable)

	a. Space occupied (bytes)	b. Space wasted (bytes)	|s| (bits)	|p| (bits)	|w| (bits)
A.					
B.					
C.					

10. [12 marks] Consider the following reference string:

Q R S T Q R U R W

Assuming a memory of 4 page frames, determine which pages will be resident at the end of the string for each of the following policies:

a. FIFO _____

b. LRU _____

c. Second chance algorithm _____

Consider now the Working Set model and the following reference string:

B A A A A B B C A A A B F B G F E E A A E E

What is the largest working set size and the smallest working set size the process will ever have for the different window sizes:

Window size	Largest ws size	Smallest ws size
3		
4		
5		
6		

11. [12 marks] Consider two processes in a purely segmented system. Their segment tables, ST1 and ST2, contain the following entries, where "addr" is the starting address of each segment is physical memory and "len" is its length (in bytes). A dash indicates a non-existent segment:

ST1

	addr	len
0	4000	200
1	6000	100
2	–	–
3	9000	1500
4	2000	100
5	7000	1200
6	–	–

ST2

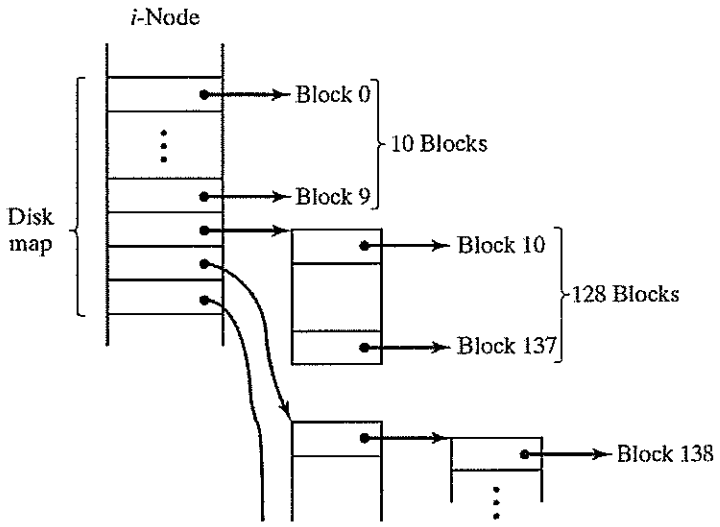
	addr	len
0	2000	100
1	6000	100
2	–	–
3	9000	1500
4	3000	300
5	–	–
6	–	–

a. Which segments are shared? (list their staring addresses, not the segment numbers)

b. Assume that the segments at locations 3000 and 6000 are code segments. Which of the following virtual addresses would be illegal if referenced in that segment? (enter X for each illegal address)

	In segment at 3000	In segment at 6000
0,0		
0,300		
1,0		
1,150		
2,0		
3,1200		
3,2000		
4,0		
4,200		
5,0		
5,2000		
6,0		

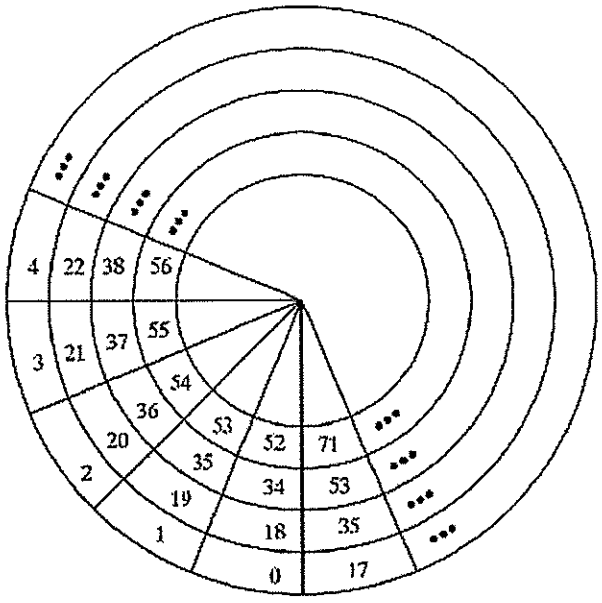
12. [12 marks] Consider the expanding file index of Unix as shown below:



How many disk blocks will be accessed (on average) in order to seek to a random location within a file of the following sizes:

File size	Average # of disk blocks accessed (give as a fraction or decimal number)
2 blocks	
10 blocks	
11 blocks	
50 blocks	

13. [12 marks] Consider the following disk organization:



Assume that all free blocks are linked together individually using the following list:

$$0 \rightarrow 1 \rightarrow 36 \rightarrow 28 \rightarrow 29 \rightarrow 3 \rightarrow 2 \rightarrow 22 \rightarrow 23 \rightarrow 30 \rightarrow 31 \rightarrow 32$$

The read/write head is currently at track 0.

a. Determine how many disk seek operations will be necessary to allocate N blocks from the above list (for each of the values of N from 1 to 9):

N	1	2	3	4	5	6	7	8	9
# of disk seeks									

b. Assume now that the same list of free blocks is sorted in ascending order, i.e.:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 22 \rightarrow 23 \rightarrow 28 \rightarrow 29 \rightarrow 30 \rightarrow 31 \rightarrow 32 \rightarrow 36$$

Determine how many disk seek operations will be necessary to allocate N blocks in this case:

N	1	2	3	4	5	6	7	8	9
# of disk seeks									

14. [12 marks] Assume that a plain text is broken up into blocks of length n and is encoded using the following scheme:

$C_0 = P_0 \oplus K$
$C_i = P_i \oplus C_{i-1}$ (for i>0)

Each P_i is a block of plain text, each C_i is the corresponding block of cypher text, K is a key of length n, and $\oplus$ is the exclusive OR operator.

This is a very weak encoding scheme, because knowing just the length of the key (without knowing its actual value) allows one to decode most of the cypher text.

a. Given the following cypher text and knowing that n=4, underline the portion of C that can be decoded:

C = 0 1 1 0 0 1 0 1 1 1 1 1 0 ...

b. Show the decoded plain text:

P = _____

15. [12 marks] Consider a biometric user authentication system that generates values between 0 and 1 where 1 is a perfect match and 0 perfect mismatch. For valid attempts, the probability of getting a value x is approximated by the following function:

$$g = 4x - 3.2$$

For invalid (imposter) attempts, the probability of getting a value x is approximated by the following function:

$$f = -0.5x + 0.5$$

a. Is it possible to set the threshold value for x such that all genuine attempts are accepted and all

imposter attempts are rejected? _____ (yes/no)

b. What should be the maximal threshold value for x such that all genuine attempts are accepted?

_____

c. What should be the minimal threshold value for x such that all imposter attempts are rejected?

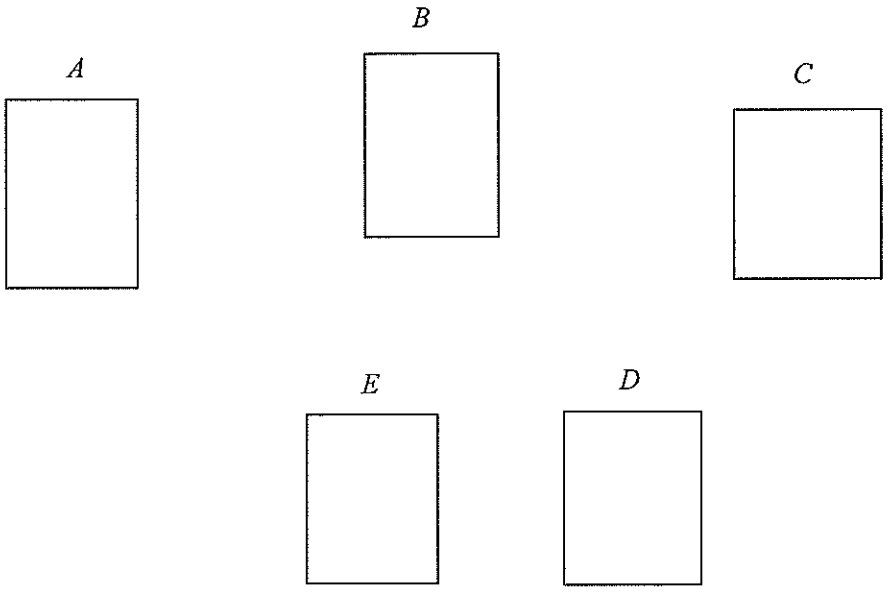
_____

d. Assume the threshold value is set to 0.5. What percentage of genuine attempts will be rejected?

_____ What percentage of imposter attempts will be accepted? _____

16. [12 marks] Consider five objects, *A*, *B*, *C*, *D*, and *E*, in a capability-based system like Hydra.

a. Show the graphical representation of the objects' capabilities (including all rights) such that the following operations may be performed:
- *A* can call *B*;
- *B* can read and write data from/to *C*;
- *B* can give its capability for *C* to another object *D*;
- *A* can read and write the capability list of *C*;
- *A* can read data from *E*;
- *B* can read and write data from/to *E*.



b. Answer the following questions (yes/no) based on your diagram.

- Can data from *E* ever get into *C*? _____
- Can data from *E* ever get into *D*? _____
- Can data from *C* ever get into *A*? _____
- Can data from *D* ever get into *A*? _____

END OF PAPER