NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

ASSESSMENT FOR
Semester 2 AY2015/2016

CS2106 Introduction to Operating Systems

May/June 2016 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains **THREE (3)** questions and comprises **THIRTEEN (13)** printed pages, including this page.
2. Answer **ALL** questions within the spaces provided.
3. PLEASE WRITE NEATLY AND CLEARLY.
4. This is an Open Book assessment.
5. Please write your Student Number below.

STUDENT NO: _____

This portion is for examiner's use only

Question	Marks	Remarks
Q1	/30	
Q2	/20	
Q3	/20	
Total	/70	

Question 1 (30 MARKS)

We have a round-robin time slicing system with a jiffy time of 1 ms, with 20 levels of priority ranging from 0 to 19. The time quantum Q(t) in milliseconds given to a task t with priority P(t) is given by:

$$Q(t) = 400 - 20 \times P(t)$$

The priority P(t) for a task t is in turn given by:

$$P(t) = base(t) + decay(t, i) + nice(t), 0 \leq P(t) \leq 19$$

Where base(t) is the base priority level, nice(t) is the current nice level (assumed initially to be 0), and decay(t, i) is a decay function given by:

$$decay(t, i) = 2i - 1, i \geq 1$$
$$decay(t, i) = 0, i = 0$$

Where i is the elapsed CPU time for the task t in hours.

a. Given the set of tasks shown in the table below, complete the priority levels and time quantum for each task using the information given above. All tasks start at the same time. (8 marks)

Priority Levels Over Time, nice(t) = 0

Task t	base(t)	i=0	i=1	i=2	i=3	i=4	i=5
A	5						
B	4						
C	2						
D	8						

Time Quantum in Milliseconds Over Time, nice(t) = 0

Task t	base(t)	i=0	i=1	i=2	i=3	i=4	i=5
A	5						
B	4						
C	2						
D	8						

b. Task A renders video onto a screen. It takes 10ms to render a single frame of video, and to get smooth videos the task must render at a minimum rate of 30 frames per second. Given the set of tasks in part a, at which values of i does A produce smooth videos? (12 marks)

c. Assuming that you have superuser privileges, what is a suitable nice value that will allow Task A to always render smooth videos for all values of i? (10 marks)

Question 2 (20 MARKS)

We have the following program:

```
int data; // 32 bit integer

FILE *fpIn = fopen("inFile.txt", "r");

// fpIn points to a valid input file,
for(i=0; i<8192; i++)
{
      fscanf(fpIn, "%ld", &data);
}

fclose(fpIn);
```

This program is run on a computer with the following disk drive specification, connected through a 100 megabytes per second bus.

Specification	**Value**
Seek Time to Adjacent Track	2 ms
Maximum Seek Time	25 ms
Average Seek Time	12 ms
Rotational Speed	3600 rpm
Head Switching Speed	Negligible
Maximum Throughput	200 megabytes per second
Blocks per Track	16
Bytes per Block	128
Number of platters	2
Number of sides	4
Number of cylinders	1024

The disk drive is almost fully empty. The directory and inode blocks are stored in various parts of the drive and may not be adjacent to each other.

a. Assuming that accessing inFile.txt nvolves one directory block read and one inode block read, what is the total number of blocks read by this program? (4 marks)

Total number of blocks read for inFile.txt	

b. Assume that there is no caching or buffering, and that the disk blocks are not skewed. What is the total time in milliseconds that this program spends reading the file? Include timing to read the directory and inode blocks. (10 marks)

(PAGE IS INTENTIONALLY LEFT BLANK)

c. Once again assuming that there is no caching or buffering, repeat part b. assuming that the blocks are skewed such that rotational delay is completely eliminated when the head moves to the next cylinder to continue reading. Once again what is the total time in milliseconds that this program spends reading the file? Include timing to read the directory and inode blocks. (6 marks)

Question 3 (20 MARKS)

In this question the p=calloc(n, b) returns a pointer to n consecutive units of memory each b bytes long, in the form of an array. So call like p=calloc(5, 6) will return a pointer to a block of memory that is 30 bytes long. free(p) frees the memory pointed to by p, and sizeof(x) returns the size of variable x in bytes.

A char variable is 1 byte long, and an int variable is 4 bytes long.

We have a memory system that consists of 256 bytes in total and **memory is allocated in units of 4 bytes (i.e. requesting for 1 byte will still return a 4-bytes of memory)**, starting at address 0. Consider the following program fragment:

```
int *x[5];

for(int i=0; i<3; i++)
{
     x[i] = calloc(5, sizeof(char));
}

free(x[0]);
free(x[2]);
x[0] = calloc(6, sizeof(int));
```

a. We are using linked-lists to manage free memory. Sketch the linked-lists after each memory operation for a first fit policy. Indicate FAIL if you are unable to perform the operation shown, and explain why the operation fails. For your convenience, the memory operations are unrolled and repeated below for you: (12 marks)

```
x[0] = calloc(5, sizeof(char));
```

```
x[1] = calloc(5, sizeof(char));
```

```
x[2] = calloc(5, sizeof(char));
```

```
free(x[0]);
```

```
free(x[2]);
```

```
x[0] = calloc(6, sizeof(int));
```

b. What is the percentage of memory that is wasted as internal fragmentation before the first free statement? Express your percentage with respect to total memory ***requested by the program, not allocated.*** (4 marks)

c. What would be the percentage of memory that is wasted due to internal fragmentation if we allocated in units of 1 byte instead of 4? (2 marks)

d. Based on your answers in b and c, comment on the trade-offs of using allocation units of multiple bytes versus allocation units of 1 byte, particularly in systems with large amounts of memory. (2 marks)

END OF PAPER