UNIVERSITY OF LONDON

BSc EXAMINATION 2018

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2850: Operating Systems

Time Allowed: **TWO hours**

Answer ALL questions
Calculators are NOT permitted

1. (a) Give a description of two of the main components in a computer architecture. [3 marks]

   (b) Briefly describe the Worst Fit memory management algorithm, and explain its rationale. [4 marks]

   (c) Explain what is virtual memory and what is its main purpose. [6 marks]

   (d) Processes and threads are amongst the key concepts of operating systems.

   i. Briefly describe what pseudo-parallelism is. [3 marks]

   ii. What is a thread? Explain why it is better to use multiple threads instead of multiple processes in certain circumstances. [3 marks]

   (e) Consider the concurrent program that consists of the following two processes that share the variable `x`:

```
P1:   x = 2;
      x = x * 2;
```

```
P2:   x = 1;
      if(x == 1)
        x = x + 5;
      else
        x = x * 2 + 4;
```

   How many different values may the variable `x` have when the program terminates? Give a possible execution sequence for each case. [6 marks]

2. (a) There are four properties that are required for any correct algorithm to solve the mutual exclusion problem. Explain the importance of the property "No process running outside its critical region may block any process." in this context. [6 marks]

   (b) Consider the following proposed solution to the mutual exclusion problem for a number of threads that share the same code, where variable `x` is initially set to `0`:

```
while(1) {
    if(x == 0) {
        x = 1;
        critical_region();
        x = 0;
        non_critical_region();
    }
}
```

   i. Briefly describe the operation of the algorithm, giving particular emphasis to any dependencies between the threads. State whether this algorithm enforces strict alternation or not and justify your answer. [5 marks]

   ii. Give one example of execution order that shows the above algorithm does not fully comply with the conditions required for solving the mutual exclusion problem. [4 marks]

   iii. Write a revised version of the algorithm using a semaphore, whilst correctly guaranteeing mutual exclusion. (Use pseudo-code for semaphore-related calls.) Briefly describe why your solution works. [6 marks]

   (c) Describe how the Test and Set Lock (TSL) instruction works, and how it can be used to correctly implement mutual exclusion. You can use pseudo-code in your explanation. [4 marks]

3. (a) Describe the goal of the barrier synchronisation mechanism, giving emphasis to how it compares to semaphores. [4 marks]

   (b) File systems can use a contiguous layout or linked lists structures to store files in disks. Briefly describe these two methods, and compare them. [6 marks]

   (c) In the context of deadlock avoidance algorithms, consider the following states, assuming there is only one type of resource.

State X	Has	Max
A	2	8
B	2	6
C	1	6
D	2	3

State Y	Has	Max
A	1	7
B	0	5
C	2	8
D	1	5

   Each row shows the information for a process. The columns show the number of resources held, and the maximum number of resources simultaneously required by each process. In both cases, there are a total of **8** resources in the system (some of which are being held as shown in the tables).

   For each of the states (X and Y), indicate if it is a safe or unsafe state, and justify your classification. [4 marks]

   (d) Explain how to prevent deadlocks by breaking the necessary condition 'hold and wait' and discuss the effectiveness and feasibility of the proposed solution. [5 marks]

   (e) In the context of virtualisation, hypervisors should score well in three dimensions: safety, fidelity and efficiency. Give a brief definition for each of these. [6 marks]

4. (a) Consider the following C program.

```
#include <stdio.h>

int main() {
  int x, y;
  int *p, *q;

  x = 7;
  y = 3;
  printf("%d %d\n", x, y); // i
  p = &y;
  *p = x;
  printf("%d\n", y);       // ii
  q = p;
  *p = -3;
  *q = 42;
  printf("%d\n", *p);      // iii

  x = 13;
  y = 17;
  p = &x;
  q = &y;
  y = *p;
  x = *q;
  printf("%d %d\n", x, y); // iv
  p = &x;
  printf("%d\n", x == *p); // v
}
```

Write down what each of the statements labelled i – v is going to print when the program is run. [10 marks]

(b) Consider the following definition of a node of a linked list of integers:

```
struct node {
  struct node *next;
  int v;
}
```

Write a C function `int sum_all(struct node *nd)` that sums up all integers of the list starting at node `nd` and returns the sum as its result. [5 marks]

(c) A programmer writes the following program to execute two functions "job1" and "job2" in parallel, and then indicate to the user that both jobs are done.

```
int job1() { ... }
int job2() { ... }

int main(int argc, char *argv[]) {
    if (fork() == 0) {
        return job1();
    }
    if (fork() == 0) {
        return job2();
    }
    printf("Jobs done!\n");
    return 0;
}
```

Upon testing the program, the programmer finds that "Jobs done!" is output before the jobs have finished. Explain what the problem is, and modify the above code so that it behaves correctly. [5 marks]

(d) Write a C function `void chomp(char *s)` that removes any spaces or newline characters from the end of the string `s`.

For instance, the string "`Hello World \n`" (where `\n` is the newline character) becomes "`Hello World`". You should modify the string in place, the old string does not have to be preserved. [5 marks]

END