UNIVERSITY OF LONDON

BSc EXAMINATION 2024

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2850: Operating Systems
CS2850R: Operating Systems – for FIRSTSIT/RESIT CANDIDATES

Time Allowed: **TWO hours**

Please answer **ALL** questions

Calculators are not permitted

1. **Theory Questions (25 marks).**

   (a) Briefly describe I/O handling based on interrupts. Give one advantage of this method over busy waiting. [5 marks]

   (b) This question is about the the Round Robin scheduling algorithm for interactive systems:

      i. Briefly explain how the algorithm works. [3 marks]

      ii. Consider four processes A, B, C, and D. Suppose their running times are:

	running time (ms)
A	3
B	7
C	4
D	5

      Indicate how these processes are scheduled by the Round Robin scheduling algorithm until all finish their execution, assuming a quantum length of 4ms. The initial list of runnable processes is A, B, C, D (A is at the head). Make sure to show all your workings. [6 marks]

   (c) Explain in which scenario context switching is faster for threads than for processes. [3 marks]

   (d) Consider a program that consists of the following two concurrent processes sharing the variable `x`:

```
P1:  x = 4;                    P2:  x = 2;
     if (x > 3) {                    x = x * 2;
       x = x + 5;
     }
     else {
       x = x * 3;
     }
```

   How many different values may the variable `x` have when the program terminates? Give a possible execution sequence for each case. [8 marks]

2. **Theory Questions (25 marks).**

(a) In the context of Inter-Process Communication, explain how the `sleep` and `wakeup` primitives work. Give one advantage of semaphores over these primitives. [5 marks]

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

State X	Has	Max
A	1	8
B	3	9
C	1	2
D	1	5

State Y	Has	Max
A	2	9
B	2	10
C	3	5
D	1	6

Each row shows the information for a process. The columns show the number of resources held, and the maximum number of resources needed by each process during its lifetime. In both cases, there are a total of **10** resources in the system (some of which are being held as shown in the tables).

For each of the states (X and Y), indicate whether it is *safe* or *unsafe* , and justify your classification. [6 marks]

(c) Ext4 i-nodes are 256 bytes in size. Suppose there are 2,000 files on the disk, 200 of which are currently open by processes. What is the total main memory usage of i-nodes in this scenario? Justify your answer. [4 marks]

(d) This question is about the Working Set page replacement algorithm. The following table shows the time of last use, and the R and M bits for each memory page (the times are in ms).

Page	Time of Last Use (ms)	R	M
1	90	1	0
2	100	0	0
3	30	1	1
4	50	0	1

Assuming a page fault occurs at virtual time 200ms, and the Working Set algorithm is used with τ = 100ms:

i. For each page, indicate whether it is in the working set or not, and justify your conclusion. [4 marks]

ii. Indicate which page is selected for eviction, and describe how the information in the table above is updated by the algorithm. [6 marks]

3. **Swap two characters (25 marks)**

(a) Write a function, `void swapChar(char *c1, char *c2)`, that swaps the values of two character variables. [5 marks]

(b) Write a function, `char * initializeString(char *buf)`, that

- takes a string constant as input,
- dynamically allocates a string of the same length as the input,
- copies the content of the input into the dynamic string,
- returns a pointer to the dynamic string.

Hint: the function should consist of two `while` loops, one for determining the length of the input and the other for copying the content of the input into the newly allocated string. Do not forget to null-terminate the string at the end. [10 marks]

(c) Complete the following main function,

```
int main() {
    char *buf = "CS2850";
    char *s = initializeString(...);
    swapChar(..., ...);
    printf("s=...\n", ...);
    ...
}
```

The program should print

```
s=SC2850
```

and it should not leak memory when it runs. You can assume the function definitions and the `#include` statements are appended on top of the file. [10 marks]

4. **A linked list of integers (25 marks)**

The program below uses dynamic memory allocation to create a linked list of integers.

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int label;
    float value;
    struct node *next;
};
int main() {
    struct node* head = NULL;
    int i = 0;
    char c = '\0';
    while ((c = getchar()) != '\n' && c != EOF) {
        if (c <= '9' && c>= '0'){
            struct node *new = malloc(sizeof(struct node));
            new->label = i;
            new->value = c - '0';
            i++;
            new->next = head;
            head = new;
        }
    }
    while (head) {
        printf("label=%d, ", head->label);
        printf("value=%f\n", head->value);
        struct node *temp = head;
        head = head->next;
        i = i + temp->value;
        free(temp);
    }
    printf("i=%d\n", i);
}
```

(a) What is the size of the structure? Why is it always better to use `sizeof` instead of computing the sum of the member sizes? [5 marks]

(b) What is the difference between `'\0'` and `'0'`? Why do you need to subtract `'0'` in `new->value = c - '0'`? [5 marks]

(c) Why are the nodes printed from the last to the first? Why does the second `while` loop end after the program prints the node with the label `0`? [5 marks]

(d) How many dynamic allocations does the program perform? How many times does it call `free`? [5 marks]

(e) Can you predict the output of the last call of `printf` if you assume the user enters `1423\n` on the terminal after the program has started? Is the argument of `printf` correct? [5 marks]

END