

# UNIVERSITY OF BIRMINGHAM

School of Computer Science

Second Year Undergraduate Papers

January Examinations 2021

## Contents

30203 LI Systems Programming in C and C++ . . . . .	3
---	---

# 30203 LI Systems Programming in C and C++

## Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

## Question 1

The question is about pointers and memory management in C.

- (a) Will there be any memory leakage in the following program? Explain your answer.

```
1 int main()
2 {
3     int *A = (int *) malloc(sizeof(int));
4     scanf("%d", A);
5     int *B;
6     B = A;
7     free(B);
8     return 0;
9 }
```

**[6 marks]**

- (b) A programmer has written the following function with the aim to return a pointer to an array of 10 random integers (`int`) to a caller function. There is a serious problem with this code. Explain what is wrong, why it is a problem, and how it can be fixed. Use this to write a correct version of the function without changing the function-signature. Assume that the caller function of `randomArray` is responsible for freeing any memory occupied by the array.

```
1 int* randomArray(void)
2 {
3     int array[10], i;
4     for (i = 0; i < 10; i++)
5         array[i] = rand();
6     return &array[0];
7 }
```

**[7 marks]**

- (c) Consider the following two C functions `sum2Darray1` and `sum2Darray2`. Both of them compute the sum of all the elements of an input 2-dimensional matrix. Which one of them will be able to exploit memory hierarchy and thus achieve faster computation time? Explain your answer.

```

1 int sum2Darray1(int a[N][M])
2 {
3     int i, j, sum=0;
4     for(i=0; i<M; i++)
5         for(j=0; j<N; j++)
6             sum =sum + a[j][i];
7     return sum;
8 }

```

```

1 int sum2Darray2(int a[N][M])
2 {
3     int i, j, sum=0;
4     for(i=0; i<N; i++)
5         for(j=0; j<M; j++)
6             sum =sum + a[i][j];
7     return sum;
8 }

```

**[7 marks]**

## Question 2

The question is about concurrent programming.

- (a) Consider a concurrent system with three processes P1, P2 and P3.

<pre> //Process P1 void P1(void){     Statement A } </pre>	<pre> //Process P2 void P2(void){     Statement B } </pre>	<pre> //Process P3 void P3(void){     Statement C } </pre>
--	--	--

Provide a (semaphore based) solution to synchronize P1, P2 and P3 such that the following constraints on execution order is satisfied:

- Statement B before Statement A,
- Statement A before Statement C

Provide a possible (deadlock free) trace of your solution.

**[7 marks]**

- (b) In the following program, the main thread creates four peer threads and passes a pointer to the loop variable to each one. Each peer thread prints a message containing the loop variable.

```
1 #include<stdio.h>
2 #include<pthread.h>
3
4 void *foo(void *arg){
5     int *myid = (int *) arg;
6     printf("Hello from thread %d\n", *myid);
7     return NULL;
8 }
9
10 int main(){
11     pthread_t tid[4];
12     int i;
13
14     for(i=0; i<4; i++)
15         pthread_create(&tid[i], NULL, foo, &i);
16
17     for(i=0; i<4; i++)
18         pthread_join(tid[i], NULL);
19
20     return 0;
21 }
```

We might expect that the program will print all the four values of *i*, but when the program is executed, we see the following incorrect result containing repetitions:

```
Hello from thread 1
Hello from thread 3
Hello from thread 3
Hello from thread 3
```

What causes this behavior? Explain your answer.

**[7 marks]**

- (c) [Continuation of Question 2b above] Rectify *only* the main() function such that the concurrent peer threads print unique values, i.e., the first thread prints 0, the second thread prints 1, the third thread prints 2 and the final thread prints 3. We don't expect the threads will print "in order" (we expect that they just print the correct value per thread). Explain your answer.

**[6 marks]**

### Question 3

This question is about Processes, Memory management & Scheduling.

- (a) The C program shown below is compiled and run on a UNIX machine. Predict all possible outputs that this program will print to the console and explain your answer.

```
1 #include <sys/types.h>
2 #include <sys/wait.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 int main() {
6     int x = 1;
7     pid_t pid = fork();
8     if (pid == 0) {
9         x = x * 2;
10    } else if (pid > 0) {
11        wait(NULL);
12        x = 3;
13    }
14    printf("%d\n",x);
15 }
```

**[4 marks]**

- (b) Schedule the processes (given in Table 1) using round robin scheduling with quantum 10. Also, calculate the total turnaround time.

**[5 marks]**

Table 1: Process table

ID	Arrival Time	Duration
P0	3	20
P1	15	22
P2	0	32
P3	5	3
P4	49	29

- (c) What are the physical memory addresses for each of the following logical addresses for the segment table (Table 2)? Note any that are invalid. The logical addresses are:
- (i) 3 , 15
  - (ii) 0 , 512
  - (iii) 1 , 4096
  - (iv) 0 , 1

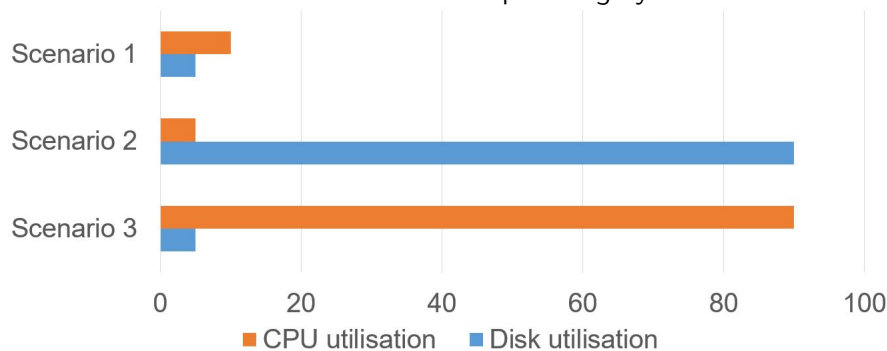
**[4 marks]**

Table 2: Segment table

Segment	Base Address	Length
0	16384	400
1	4096	4096
2	32768	810
3	20480	1024

- (d) Consider a system with four frames of memory, and the following sequence of page accesses: 0,1,2,3,4,2,3,0,1,4,2. When do page faults occur using FIFO and LRU replacement algorithms? Briefly justify your answer.
- (e) Consider a demand-paged computer system which was recently measured to determine the utilisation of CPU and the paging disk to decide the degree of multiprogramming. The results are shown in the figure below. Explain what is happening in each scenario and what actions an operating system can take.

**[4 marks]**



**[3 marks]**