Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers

UNIVERSITY OF
BIRMINGHAM

School of Computer Science

LI Operating Systems and Systems Programming

Mock Exam January 2023

Time allowed: 2 hours

[Answer all questions]

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.

```
int main()
{
    int *A = (int *) malloc(sizeof(int));
    scanf("%d", A);
    int *B;
    B = A;
    free(B);
    return 0;
}
```

[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.

```
int* randomArray(void)
{
    int array[10], i;
    for (i = 0; i < 10; i++)
       array[i] = rand();
    return &array[0];
}
```

[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.

Non-alpha only

```
int sum2Darray1(int a[N][M])
{
   int i, j, sum=0;
   for(i=0;i<M;i++)
     for(j=0;j<N;j++)
       sum =sum + a[j][i];
   return sum;
}
```

```
int sum2Darray2(int a[N][M])
{
   int i, j, sum=0;
   for(i=0;i<N;i++)
     for(j=0;j<M;j++)
       sum =sum + a[i][j];
   return sum;
}
```

[7 marks]

Question 2

(a) The question is about Main and Virtual Memory. Provide a brief answer.

(i) Why does a processor have a set of registers in addition to a large main memory? **[1 mark]**

(ii) A scheduler controls the degree of multi-programming in an Operating System. The scheduler can send a process to which state(s)? **[1 mark]**

(iii) Does adding more frames during Page Replacement always lead to improved performance? **[1 mark]**

(iv) A system is running with following measure behaviour: CPU utilization 10%; Paging disk 95% Other I/O devices 3%. Explain which of the following actions will improve CPU utilization and why?

i. Install more main memory

ii. Install a faster disk

iii. Changing the degree of multi programming **[3 marks]**

(b) Briefly describe what the possible consequences are of a buffer overflow in the kernel. **[6 marks]**

(c) Consider the following piece of kernel code. The intention is that whenever data is written to a proc-file, this data is written to a device. The device provides two functions: **start_transfer** starts the transfer of count bytes to the device and returns immediately, and **transfer_finished**, which is called by the device when the data transfer is finished. The function **kernelWrite** should return the number of bytes transferred to the device.

```
int total_transferred = 0;
/* total number of bytes transferred since module loaded */
int transferred = 0;
/* bytes transferred to device in single transaction */

/* called by device when transfer finished */
/* called in interrupt mode */
void transfer_finished(int count) {
  transferred = transferred + count;
  /* wakeup waiting process */
}

/* called every time data is transferred to kern
    ,as a result of writing to proc-file  */
int kernelWrite(char * buffer, int count) {
  /* buffer is pointer to user space */
  start_transfer(buffer, count);
  /* go to sleep until woken up in transfer finish */
  transferred = transferred + count;
  return transferred;
}

void init module(void) {
  /* set up proc-structure - code omitted */
  proc->write_proc = kernelWrite;
}
}
```

This kernel code compiles correctly but does not work as intended. Identify these errors and suggest remedies. If you think critical sections are required, it is sufficient to indicate begin and end of a critical section, and whether you would use semaphores or spinlocks to protect the critical section.

[8 marks]

Non-alpha only

Question 3

(a) Four processes running on a single-core processor (Table 1). Among all the Scheduling Algorithms, briefly explain which one you will prefer and which one you would like to avoid for the given scenario? **[5 marks]**

Table 1: List of Processes

Process	Type	Arrival time	Burst time
P1	CPU Bound	0	50
P2	I/O Bound	1	3
P3	I/O Bound	2	4
P4	CPU Bound	3	20

(b) Predict all possible outputs that the following C program will print to the console and briefly explain your answer. What will be the state of parent process? Briefly explain the behaviour of the program if we comment out the line number 16.

```
 1 #include <sys/types.h>
 2 #include <sys/wait.h>
 3 #include <unistd.h>
 4 #include <stdio.h>
 5 #include <stdlib.h>
 6 int main() {
 7    pid_t   wpid, child_pid;
 8    int status = 0;
 9    int i;
10    for(i = 0; i < 2; i++) {
11        if ((child_pid = fork()) == 0) {
12           printf("process \n");
13           exit(0);
14        }
15    }
16    while ((wpid = wait(&status)) > 0);
17    return 0;
18 }
```

[5 marks]

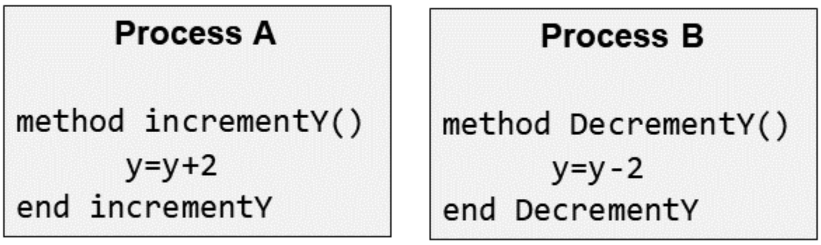
(c) Your computer system uses Round Robin scheduler and is not very responsive and so you decide to change the scheduling time quantum from 50 msec to 1 msec. Now the performance is even worse. Why is this happening? **[4 marks]**

(d) Consider a concurrent system with two processes A and B (Figure 1). Assume y is a shared variable with value of 5. Describe how a race condition is possible and provide a solution to prevent the race condition from occurring. **[6 marks]**

End of Paper

Non-alpha only

Figure 1: Concurrent processes A and B



Process A

```
method incrementY()
       y=y+2
end incrementY
```

Process B

```
method DecrementY()
       y=y-2
end DecrementY
```

End of Paper

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.