FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**COMPUTER SCIENCE 3B** CSC3B
CAMPUS	AUCKLAND PARK CAMPUS (APK)
SUPPLEMENTARY EXAM	JANUARY 2020

DATE: 2020-01 **SESSION:** Normal

ASSESSOR(S): **DR. J. DU TOIT**
MR. A. MAGANLAL

MODERATOR: **EXTERNAL: MR. J. PRINSLOO (NWU)**

DURATION: 180 MINUTES **MARKS:** 150

Please read the following instructions carefully:

1. Answer **all** the questions.
2. Answer questions in order.
3. Answer only in the examination books provided.
4. The use of calculators is *not* permitted.
5. Write ***cleanly* and *legibly***.
6. This paper contains **10** questions.
7. This paper consists of **7** pages excluding the cover page.

QUESTION 1: Operating Systems - General

(a) **Discuss** the ***common components*** in a computer that an operating system *manages*. **[04]**

(b) ***Handling I/O*** can be done in three (3) different ways. One of the ways is through ***busy waiting***. **Discuss** the other two (2) ways of ***handling I/O***. **[04]**

(c) **Discuss** how ***multithreaded*** *processors* differ from ***multi-core*** *processors*? **[04]**

(d) i. **When** does an *operating system* make ***use of a system call***. **[02]**

ii. **Provide** one example of a ***system call***. **[01]**

Total: 15

QUESTION 2: Processes and Threads

(a) **State** if the following ***process termination*** *conditions* are voluntary or non-voluntary.

i. Normal exit **[01]**

ii. Fatal error **[01]**

(b) **List** any four (4) *goals* that must be achieved for a *scheduling algorithm* for ***interactive systems***. **[04]**

(c) **Name** the two (2) ***types of threads***. **Discuss** performance considerations, with respect to scheduling, for each of these thread types. **[04]**

(d) Consider the following processes in a *preemtive* system (Highest priority = 0): **[05]**

Process	Priority	Burst Time
A	2	5
B	3	9
C	4	6
D	5	2

Using the ***priority scheduling with priority decrease*** algorithm with a `5 msec` quanta provide the order execution in the following format (copy and complete the table into your answer sheet):

Time Spent	...	...
Process	...	...
Priority when run	...	...

Total: 15

~~ **Assessment continues on the next page.** ~~

QUESTION 3: Memory Management

(a) Given a fictional CPU. **Determine the 7-bit physical memory address in decimal** for the following 8-bit virtual address, given the following page table. **[06]**

Virtual address: 40.

Index	Page Frame	Present
7	01	1
6	00	0
5	00	1
4	00	0
3	00	0
2	10	1
1	11	1
0	00	0

Show all the steps from converting from decimal to binary and then from looking up the address to converting back from binary to decimal.

(b) A computer has four page frames. The time of loading, time of last access and the R and M bits for each page are shown below:

Pages	Loaded	Last ref.	R	M
A	16	142	1	0
B	46	191	0	1
C	38	199	0	0
D	132	201	1	0

Answer the following in context of page replacement algorithms.

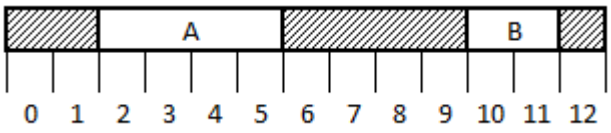
i. Which page will Not Recently Used (NRU) replace? **[01]**

ii. Which page will First In First Out (FIFO) replace? **[01]**

iii. Which page will Least Recently Used (LRU) replace? **[01]**

iv. Which page will second chance replace? **[01]**

(c) **Provide** a linked list representation of the memory state described below: **[05]**



Total: 15

~~ **Assessment continues on the next page.** ~~

QUESTION 4: File System

(a) Given the command prompt below, **answer** the following questions:

```
E:\Project\Marketing>tree \ /F
Folder PATH listing for volume KALI LIVE
Volume serial number is C212-D5B0
E:\
└───Project
    │   Index.txt
    │
    ├───Marketing
    │       poster.pub
    │       promo.mpg
    │
    └───Sales
            customers.txt
```

i. **Provide** the *absolute path* for the ***working directory***. **[02]**

ii. **Provide** the *relative path name* for the file called `Index.txt` from the current working directory **[02]**

(b) Answer the following questions, given the following directory and file allocation table (FAT).

File Name	Starting Block
.	6
..	11
File A	3
File B	12
File C	19

Table 1: Directory

0	EOF	11	EOF
1	FREE	12	21
2	FREE	13	EOF
3	4	14	FREE
4	10	15	16
5	FREE	16	20
6	EOF	17	FREE
7	EOF	18	13
8	FREE	19	15
9	EOF	20	EOF
10	7	21	18

Table 2: File Allocation Table

i. **Name** the block number of the current working directory. **[01]**

ii. **Name** the block number of the parent directory. **[01]**

iii. **List** the blocks that stores the content of File A. **[02]**

iv. **Draw** and i-node representation for File B **[03]**

(c) **Discuss** contiguous allocation as a solution implementing files. Include in your discussion the following aspects: **[04]**

- How data is stored.
- At least one advantages and one disadvantages.
- Provide at least one example of where this scheme is implemented

Total: 15

~~ **Assessment continues on the next page.** ~~

QUESTION 5: Input/Output

(a) The following assembly code prints a black and white, upper case letter 'A' to the top left corner of a console screen. **[05]**

```
;Print character 'A'
MOV [B8000h],0F41h
```

This method is the only method available to OS programmers to interface with device controllers. **Discuss** this method.
Include in your discussion the following aspects:

- The name of the method.
- A description of the method.
- One advantage of using this method.
- One disadvantage of using this method.

(b) The next two code segments describes how some text is sent to a printer. **[05]**

The first code segment prepares everything for printing

```
copy_from_user(buffer,p,count);
set_up_DMA_controller();
scheduler();
```

The second code segment is called after the characters have printed.

```
acknowledge_interrupt();
unblock_user();
return_from_interrupt();
```

Discuss the fundamental method used by the code to perform I/O. Include in your discussion the following aspects:

- The name of the fundamental method.
- The CPUs involvement in the method.
- One advantage of the method.
- One disadvantage of the method.

(c) **List** three (3) of the I/O software layers. **[03]**

(d) On an imaginary disk with 40 cylinders a request comes in to read cylinder 20. While the hard disk is busy servicing the request on cylinder 20, requests to the following cylinders come in: ***39,22,31,21,15***. Given these cylinders, if the operating system uses the ***shortest seek first*** algorithm, **[02]**

write the order in which the cylinders will be serviced.

Write only the cylinder numbers in order of service.

(Example if you think it will be cylinder 1 then 2 then 3 etc, write 1 2 3).

Total: 15

~~ **Assessment continues on the next page.** ~~

QUESTION 6: Deadlocks

(a) **List** three (3) strategies that can be taken for dealing with deadlocks **[03]**

(b) **Draw** a ***resource allocation graph*** for the following states **and** specify whether the system is in a deadlock: **[04]**

- Process W requests A
- Process X holds A and requests B and C
- Process Y requests A

(c) Consider the following resource matrices and vectors (E - existing resources, A - available resources): **[05]**

	Interrupt Table	PCB	S Queue	Thread Table
E = (	1	7	5	7)

	Interrupt Table	PCB	S Queue	Thread Table
A = (	0	2	3	1)

Current allocation matrix

Process 1	0	2	0	1
Process 2 C =	1	2	0	0
Process 3	0	1	2	5

Request matrix

R =	1	0	0	2
	0	3	3	6
	0	2	3	1

Use the deadlock detection algorithm to determine if the current state is in a deadlock. For each round of the algorithm provide the available resource vector (A vector).

After the final round of the algorithm state whether system is **deadlocked or not**.

(d) Name an approach that can be taken for each of the following conditions to prevent deadlocks:

i. Mutual exclusion. **[01]**

ii. Hold and wait. **[01]**

iii. Circular wait. **[01]**

Total: 15

QUESTION 7: Virtualization and MPS

(a) **Name and briefly describe** the requirements for virtualisation **[06]**

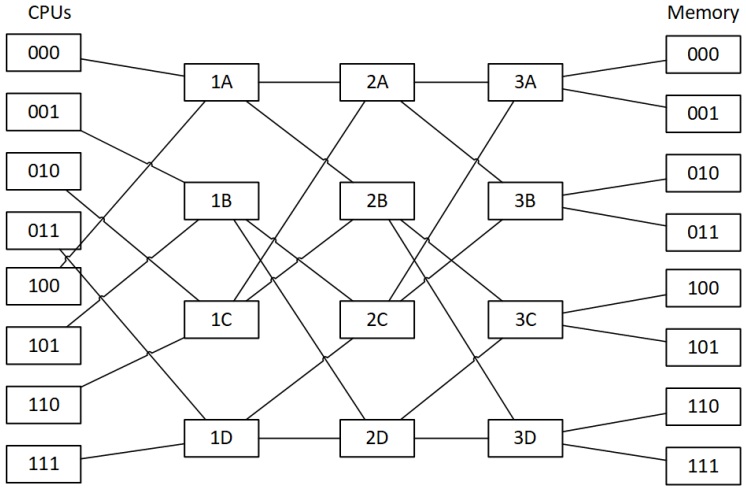
(b) Statement: Microsoft Windows can run as a para-virtualised guest.
Discuss the above statement by referring to the following aspects:

i. State whether the above statement is **True** or **False**. **[01]**

ii. **Motivate** your answer in (i) **[03]**

~~ **Assessment continues on the next page.** ~~

(c) Given the following ***omega switching network*** answer the questions which follow:



i. **Which** switches will be accessed when CPU 000 needs to access Memory 111. **[01]**

ii. **Which** switches will be accessed when CPU 100 needs to access Memory 100. **[01]**

iii. Can the request in (i) and (ii) be simultaneously processed? Justify your answer. **[02]**

(d) **Briefly discuss** time sharing and space sharing in context of multi processing scheduling. **[02]**

(e) Complete the missing nodes (i - iv) highlighted in the following table that summarises the differences between a multprocessor system and a distributed systems **[04]**

Item	Multiprocessor	Distributed System
Node configuration	(i)	Complete computer
Node peripherals	All shared	(ii)
Location	(iii)	Possibly worldwide
Operating systems	One, shared	(iv)

Total: 20

QUESTION 8: Security

(a) **With the aid of a diagram** describe how basic encryption and decryption works. **[05]**

(b) Given the following protection matrix. **List** the ***capability lists (C-List)*** for users on the databases. **[03]**

	Banking_DB	Media_DB	Movies_DB
Student1			Read
Student2		Read and Write	
Student3	Read and Write	Read	

(c) Given the following monoalphabetic substitution cipher, and ciphertext. **Provide** the plaintext for the following ciphertext. **[02]**

Key: A -> G **Ciphertext**: vgyy

Total: 10

~~ **Assessment continues on the next page.** ~~

QUESTION 9: 80x86 Theory

(a) **Discuss** how ***division*** is handled when using *FPU arithmetic* in 80x86 assembly. Your discussion must include instructions used, the registers that are affected, the data types involved and procedure followed. **[05]**

(b) **Draw** the stack as it will exist after the following function in the **C** programming language is called (after the stack frame is set up). The function contains no local variables. **[05]**

```
1 void decimate(int* psi, int* phi)
```

(c) **Show** the conversion of 63.625_{10} into ***IEEE Single-Precision Representation***. Show *all the steps of your calculation* and show the final result as a *hexadecimal number*. **[05]**

Total: 15

QUESTION 10: 80x86 Cold code

Write an 80x86 assembly program that contains the following function:

```
1  .386
2  .MODEL flat
3  .STACK 4096
4  ExitProcess PROTO NEAR32 stdcall, dwExitCode : DWORD
5  .DATA
6    ; code omitted
7  .CODE
8    ; function code here
9  _start:
10   ; code omitted
11 PUBLIC start
12 END
```

A recursive ***map*** function that takes the following parameters:

`arrRef` array address
`n` array index
`size` array size

The function will modulus each element in the array by 3. The function operates recursively. Perform ***map*** only in the value at `arr[n]` and call the ***map*** function with a higher `n` value. If the value of `n` is greater or equal to the `size` parameter then the function will return.

Note: The function must make use of recursion. (If you provide a solution that does not use recursion you will not be eligible for the full allocation of marks)

Total: 15

~~ **THE END** ~~