FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**COMPUTER SCIENCE 3B** CSC03B3
CAMPUS	AUCKLAND PARK CAMPUS (APK)
EXAM	NOVEMBER 2021 - **MEMO**
DATE: 2021-11-03	**SESSION:** Normal
ASSESSOR(S):	**DR J. DU TOIT**
	MR. A. MAGANLAL
MODERATOR:	**EXTERNAL: MR. P. JOOSTE (NWU)**
DURATION: 180 MINUTES	**MARKS:** 150

Please read the following instructions carefully:

1. You must complete the assessment **by yourself** within the prescribed time limits.
2. No communication concerning the assessment is permissible during the assessment session *except* with **ACSSE** staff members.
3. You are *bound* by **all** university regulations including, but not limited, to assessment, plagiarism, and ethical conduct.
4. You *may not* directly take any text or source code from any reference, including your own previous submissions. All text and source code *must* be written by **yourself** *during* the assessment.
5. You are allowed to submit either a *typed submission* or a *pen and paper submission*.
6. If you are submitting a *pen and paper submission* then:
    - Write *cleanly* and *legibly*.
    - Make use of CamScanner (or equivalent) app to create a single PDF from your written work.
7. **All answers** must be in a *single* **PDF** file. Make sure your details appear at the top of the first page of the PDF file. The name of the file must be in the following format:
   `SURNAME_INITIALS_STUDENTNUMBER_CSC3B_2021_EXAM.pdf`
8. Upload the *single* **PDF** file of your answers to both Blackboard and EVE **BEFORE** the timer expires. Failing to submit to Blackboard means you did not hand in on time and will not earn you any marks.
9. Complete the Honesty Declaration and upload it to the relevant practical on EVE. The completed **Honesty Declaration** is required for a submission to be eligible to be marked.
10. Additional time for submission is allowed for as per the posted deadlines on EVE.
11. This paper contains **9** question(s).
12. This paper consists of **14** page(s) excluding the cover page.

QUESTION 1: Operating Systems - General

(a) An operating system ***multiplexes resources*** in *time* and in *space*. **Provide** one example of a ***time multiplexed resource*** and one example of a ***space multiplexed resource***. **[04]**

Solution:

(✓✓ each)

- Time Multiplexing: Any computer component that can be time shared, like the CPU.
- Space Multiplexing: Memory or hard disk is space multiplexing.

(b) **Compare** and **contrast** ***microkernel*** based *operating systems* and ***monolithic kernel*** based *operating system*. You may provide a diagram to aid your comparison **[08]**

Solution:

(✓ per point, max 4 per kernel)

- **Microkernel**
  - Split the operating system up into small well defined modules.
  - Only one of the module need to run in the kernel.
  - Most of the operating system runs in user mode.
  - The microkernel handles interrupts, processes, scheduling, IPCS.
- **Monolithic**
  - Runs as a single program in kernel mode.
  - Runs as a collection of procedures linked together into a larger executable.
  - Very efficient Inter-Process Communication
  - A crash in one of the procedures can cause the entire operating system to go down.

(c) **Describe** how an operating system *abstracts* the ***concept of a file***. **[03]**

Solution:

A file allows a user to store data on some type of storage. The data and the way it is abstracted is presented as a file which is part of a file system. The user and applications do not have to worry about the complexity of interfacing with the actual hardware to write data to and from the storage system. It also allows the use of different storage types, but exposes the files as a coherent data structure.

Total: 15

QUESTION 2: Processes and Threads

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

i. Error exit **Solution:** Voluntary **[01]**

ii. Killed by another process **Solution:** Non-voluntary **[01]**

(b) **Compare *monitors*** and ***locking variables*** as *methods of synchronisation*. **[04]**

Solution:

(✓ for concept, ✓ per discussion)

	Concept	Discussion
Monitors	Compiler driven	Notifications are provided by the programmer
Locking Variables	Busy waiting	Threads are in a spin lock while waiting for the lock to be released

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

Solution:

(✓ for name, ✓ for performance)

- User-level threads - switching between threads requires minimal resources.
- Kernel-level threads - switching between threads requires full context switch.

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

Table modified for memo, single table in normal paper

Process	Priority	Burst Time (msec)	Process	Priority	Burst Time (msec)
A	0	12	C	0	2
B	1	6	D	1	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):

Solution:

Time Spent	5	2	5	2	5	2	1
Process	A	C	A	D	B	A	B
Priority when run	0	0	1	1	1	2	2

The order of the processes maybe different within the same prioritylevel

Total: 15

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: 240.

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

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

Solution:

The marker should see if marks can be awarded to the rest of the steps taking into account the earlier mistake

- Convert: 240 to binary: 1111 0000 ✓
- Lookup: 111 (7) ✓✓
- Result: 10 (Page frame) ✓
- Physical address: 101 0000 ✓
- Decimal: 80 ✓

(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	205	246	0	0
B	337	340	1	1
C	144	352	1	0
D	239	241	0	1

Answer the following in context of page replacement algorithms.

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

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

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

iv. Which page will second chance replace? **Solution:** A **[02]**

(c) The following diagram shows a page table with three entries. The diagram further describes a sequence of requests for pages. **Redraw** the diagram on your answer sheet and indicate how pages are paged into and out of the page table given the ***Optimal*** page replacement algorithm for each request in the sequence. **[05]**

Clearly indicate the number of page faults and where they occur.

Solution:

Page Requests	2	2	7	1	6	1	1	8	7	1	1
Page Table	2	2	2	2	6	6	6	8	8	8	8
			7	7	7	7	7	7	7	7	7
				1	1	1	1	1	1	1	1
Page Fault	1		1	1	1			1			

(d) Consider a 16-bit virtual address space of 64 virtual pages.

Each page has an address space of 1024 addresses.

The physical memory consists of 32 page frames.

Answer the following question related to this virtual and physica memory arrangement.

Show the relevant calculation for each fo the answers.

i. How many bits are used in the offset for each of the virtual and page frames? **[02]**
Solution: 10 bits✓ 2^{10} = 1024 addresses

ii. How many bits are necessary to represent the virtual page number? **[01]**
Solution: 6 bits✓ 2^{6} = 64 virtual pages

iii. How many bits are necessary to represent the physical page number? **[01]**
Solution: 5 bits✓ 2^{5} = 32 page frames

Total: 20

QUESTION 4: File Systems

(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]**

Solution:
`E:\Project\Marketing`
((✓) if it is an absolute path, but incorrect path)
(✓✓ if it is the correct path)

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

Solution:
`..\Sales\customers.txt`
((✓) if is a relative path, but incorrect)
(✓✓ if correct path)

(b) Given the directory and file allocation table (FAT) below, answer the questions that follow.

File Name	Starting Block
.	10
..	14
File A	8
File B	12
File C	18

Directory

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

File Allocation Table

i. **Name** the block number of the current working directory. **Solution:** 10✓ **[01]**

ii. **Name** the block number of the parent directory. **Solution:** 14✓ **[01]**

iii. **List** the blocks that stores the content of File A. **Solution:** 8,7,5,17,9 ✓✓ **[02]**

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

Solution:

Attributes
18
16
0
2
19

- Attributes at top ✓
- 18, 16, 0, 2, 19 ✓
  (if 18 is omitted (✓) instead)
- Data blocks ✓

(c) **Name and briefly discuss** three methods that increases file system performance. **[06]**

Solution:

- Caching. ✓ Using RAM to store frequently opened disk blocks or files.✓
- Block read ahead. ✓ Instead of just reading the required blocks, read the blocks ahead of time. Reduces the IO time.✓
- Reducing disk arm motion.✓ Creating cylinder groups with their own i-nodes.✓

(d) **List** three advantages of implementing a file system using linked list table in memory **[03]**

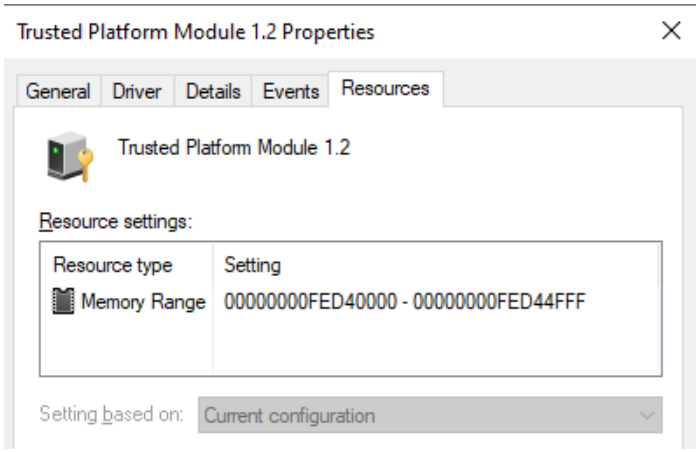
Solution:

- **(adv)** No disk space is lost to disk fragmentation✓
- **(adv)** The directory entry only has to store the first block✓
- **(adv)** Random access of files are much faster than normal linked allocation✓

Total: 20

QUESTION 5: Input/Output

(a) The following picture describes the resources used by the Trusted Platform Module (TPM) built into a Dell laptop. Answer the following questions related to this resource



i. **Name** the approach the CPU can use to interface with the control registers and data buffers of the TPM. **[01]**

Solution: Memory mapped ✓

ii. **Briefly discuss** two advantages using this method of interfacing with the hardware. **[02]**

Solution:
- Simple high level programming instructions can be used to interface with the device ✓
- Memory address ranges belonging to the device can easily be excluded from normal user processes to ensure it does not take part of the virtual to physical translation. ✓

iii. **Briefly discuss** two disadvantages of the methods depicted in this scenario **[02]**

Solution:
- Special care should be taken to ensure that control registers are not cached ✓
- Special care must be taken to ensure that some memory addresses be interfaced on the IO peripheral bus and not the memory bus.✓

(b) You have been asked to write a device driver for a 3D printer. The 3D printer controller does **not** have any DMA capabilities, but include interrupt capabilities. Each 3D printed design makes use of a set of vectors. A vector is made up of two coordinates. The 3D printer reads one vector at a time, prints a layer of plastic and reads the next vector. **[05]**

Write a high-level step-by-step explanation performed by the interrupt service procedure of the device driver. (The step-by-step instructions can be written in descriptive english instead of code).

Solution:

(✓ for each)
- Unblock the user process if there are no more vectors to print and continue acknowledging the interrupt
- Copy the next vector into the controller buffer.
- Decrement the total number of vectors.
- Acknowledge the interrupt.
- Return from the interrupt.

(c) Discuss device drivers. In your discussion include the following aspects **[05]**

- Where in the operating system layers the device drivers normally execute?
- Who is normally responsible for writing device drivers?
- Will the same copy of a device driver work for all operating systems?

Solution:

- Device drivers are normally located in kernel mode ✓✓
- Device manufacturers normally write device drivers ✓✓
- Device drivers are operating system specific ✓

(d) On a disk with 40 cylinders a request comes in to read cylinder 34. While the hard disk is busy servicing the request on cylinder 34, requests to the following cylinders come in: ***38,8,35,11***. **[03]**

Given these requested cylinders, if the operating system uses the ***shortest seek first***, **which** order will the cylinders be served in?

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

Solution:

{34}, 35, 38, 11, 8 (34 maybe be left out) ✓✓✓

(e) **Briefly describe** what a mickey is in mouse software. **[02]**

Solution:

Mickey - The minimum value (movement) a mouse has to detect before it is sent to the computer. ✓✓

Total: 20

QUESTION 6: Deadlocks

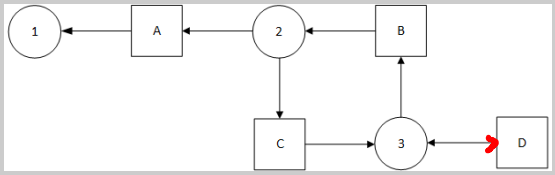
(a) **Draw** a ***resource allocation graph*** for the following states **and** specify whether the system is in a deadlock: **[04]**

- Process 1 holds A
- Process 2 holds B and requests A and C
- Process 3 holds C and requests B and D

Solution:

(✓ each)

- Processes as circles
- Resources as squares
- Correct arrows
- System is deadlocked



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

	Printers	DVD Roms	Scanners	Tape Drives
E = (	5	7	6	5)

	Printers	DVD Roms	Scanners	Tape Drives
A = (	3	4	2	4)

Current allocation matrix

Process 1	1	1	1	1
Process 2	0	1	2	0
Process 3	1	1	1	0

Request matrix

R =	3	2	2	3
	5	2	4	2
	1	0	3	0

Use the deadlock detection algorithm to determine if the current state is in a deadlock. For each round of the algorithm give the process that ran as well as the available resource vector (A vector).

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

Solution:

The student is only penalised for an immediate mistake, One mark may be awarded to final state if interpreted correctly.

- Process 1: A = (4 5 3 5) ✓
- Process 3: A = (5 6 4 5) ✓
- Process 2: A = (5 7 6 5) ✓

This scenario is NOT deadlocked scenario. ✓✓

(c) In the last two weeks, two processes running on a booking server have gone into a deadlock on a daily basis. Both processes use two separate tables that are necessary for the process to complete. **[06]**

Briefly describe two techniques that you will consider as a technique to recover from the deadlock.

Solution:

Only two solutions are required. Maximum 3 ✓ per solution

Recovery through preemption ✓✓✓

- Suspend one the processes.
- Reassign the database table to the other process. Reassign the resource when the process is complete.✓

Recovery through rollback. ✓✓✓

- Rollback the process up to the point where it does not request the resources.
- Checkpoints are required to be written to a file, so that when the process restarts it restarts from the previous checkpoint.
- Checkpoints typically contain the memory image of the process at that time, but also all of the resources assigned to the process at that time.

Recovery through killing process. ✓✓✓

- Can be done only if it is safe.
- Process updating a database by adding numbers is not safe, but just updating is safe.

Total: 15

QUESTION 7: Virtualization and MPS

(a) **Critically compare *Type-1*** against ***Type-2*** hypervisors. **[06]**

Solution:

3 ✓ for Type-1 and 3 ✓ for Type-2.
We allow more differences to be mentioned. Mark using discretion, as long as there is a comparison.
Type-1:

- Runs directly on the CPU without any host O/S
- Requires the CPU to support Type-1 hypervisors (special processor extensions)
- Guest operating systems make hyper-calls into the hypervisors

Type-2:

- Requires a host O/S.
- Does not require processor extensions.
- Guest operating systems undergoes binary translation so that the sensitive system calls are interpreted by the hosted hypervisor.

(b) You have just been employed at a cloud service provider. A colleague of yours notices that some of the servers mentions ***NUMA*** and some don't. **[05]**

Discuss *NUMA* and ***UMA*** architectures.

Include in your discussion the following aspects:

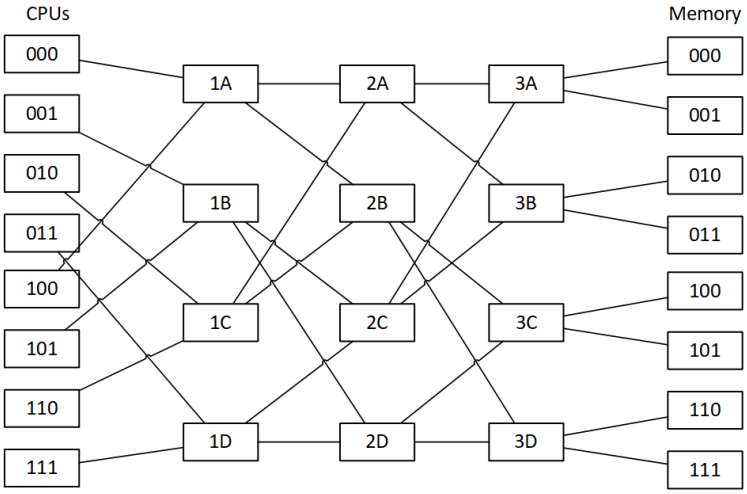
- **A brief explanation** of what NUMA and UMA is
- **Explain** why NUMA is required for some computers and why other computers do not require NUMA.
- **Describe** what characteristics you can visually see on a motherboard that may indicate whether they are using either NUMA or UMA.

Solution:

1 ✓ for an attempt to describe what NUMA is, it may be included in the description of NUMA and UMA later. 2 ✓ for why servers may require NUMA architecture. 2 ✓ for showing they realise that server hardware that has complex memory switches usually require NUMA, while server hardware that uses more localised memory architectures does not make use of NUMA.

- NUMA is an architecture that describes how multiple memory modules and multiple CPUs are interconnected, with the focus on the consistent speed.
- Servers require NUMA architecture when the CPU and Memory modules does not have consistent speed between all of them.
- Some memory modules can be accessed faster than other memory modules.
- UMA: Access between the processors and memory are uniform.
- NUMA: Motherboard usually has a number of CPU slots, with a number of Memory Banks, each working through separate memory controllers.
- UMA: The Motherboard may have more than one CPU slot, but typically not more than two. The memory banks are also physically closer together on the motherboard.

(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]**
Solution: 1A 2B 3D✓

ii. **Which** switches will be accessed when CPU 011 needs to access Memory 101. **[01]**
Solution: 1D 2D 3C✓

iii. Can the request in (i) and (ii) be simultaneously processed? Justify your answer. **[02]**
Solution: Yes✓, No switches are shared.✓

Total: 15

QUESTION 8: 80x86 Theory

(a) **Discuss** the concept of a ***stack frame***. Your discussion must include how the stack frame is *created/destroyed* and the *use* of the stack frame. **[05]**

Solution:

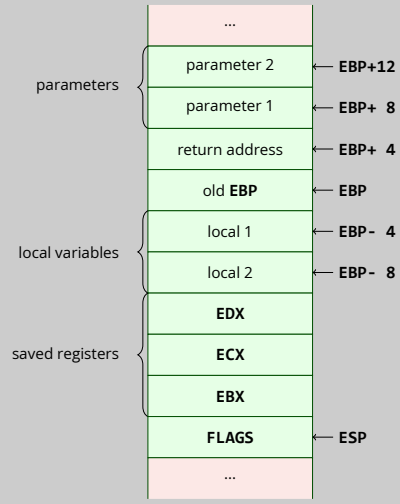
(✓ each, exact wording not required.)

- Stack frame is used to store data for the local context (local variables)
- Creation of a new stack frame involves pushing **EBP** and initialising it to **ESP**
- Reserving space in the stack frame is done by moving **ESP**
- Local variables are referenced relative to **EBP**
- Destruction of a stack frame involves resetting **ESP** and popping the old **EBP**.

(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 local variables. **[05]**

```
1  int flex(int* values, int length)
2  {
3    int fitness = 77;
4    int bait = 2;
5  }
```

Solution:



Cells are 4 bytes (**DWORD**)

- ✓ for correct parameter order and types.
- ✓ for correct local variables.
- ✓ for return address
- ✓ for old base pointer
- ✓ for registers and flags (no **EAX**)

(c) **Show** the conversion of -51.6875_{10} into ***IEEE Single-Precision Representation***. Show *all the steps of your calculation* and show the final result as a *hexadecimal number*. **[05]**

Solution:

Convert to binary: $-51.6875_{10} = 110011.1011_2$ ✓

Scientific notation: $1.100111011_2 \times 2^5$ ✓

S bit = 1 (negative number) ✓

E bits = 5 + 127 = 10000100_2 ✓

F bits = 100111011_2 padded with 0

- Binary: 1100 0010 0100 1110 1100 0000 0000 0000
- Hex: 0xC24EC000 ✓

Total: 15

QUESTION 9: 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    ; plif function code here
9  _start:
10   ; code omitted
11 PUBLIC start
12 END
```

A iterative ***plif*** function that takes the following parameters:

`arrRef` array address
`size` array length

The function will logical left shift each element in the array by 3. The function operates iteratively.

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

Solution:

- **Entry code**
  - Setup stack frame ✓
  - Push registers and flags ✓
- **Loop condition test**
  - Test condition using **CMP**/**JECXZ** ✓
  - Jump to end of function ✓
- **Perform logical left shift**
  - Get correct address of element ✓✓
  - Modify value ✓
  - Save value in correct address ✓
- **Loop increment**
  - Move to next iteration ✓✓
  - **JMP** or **LOOP** to start of loop ✓
- **Exit code**
  - Pop registers and flags reverse order ✓
  - Destroy stack frame ✓
- **Return**
  - **RET** ✓
  - Correct operand for **RET** ✓

Total: 15

~~ THE END ~~