FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**COMPUTER SCIENCE 3B** CSC3B
CAMPUS	AUCKLAND PARK CAMPUS (APK)
SPECIAL EXAM	JANUARY 2021 - **MEMO**
DATE: 2021-01	**SESSION:** Special
ASSESSOR(S):	**DR J. DU TOIT** **MR. A. MAGANLAL** **PROF D VAN DER HAAR**
MODERATOR:	**EXTERNAL: MR. J. PRINSLOO (NWU)**
DURATION: 180 MINUTES	**MARKS:** 150

Please read the following instructions carefully:

1. You are not allowed to assist or gain assistance from anyone during the assessment.
2. Complete the **Honesty Declaration: Online Assessment** and submit it.
3. If you **do not have access to a computer** then you can do a pen and paper submission. Write legibly and use CamScanner to create a PDF.
4. No communication concering this test is permissible during the assessment session except with **ACSSE** staff members.
5. Answer each of the five main questions in its own pdf document.
6. Upload each pdf document seperately to eve.
7. This paper consists of **12** pages (**10** questions).

QUESTION 1: Operating Systems - General

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

Solution:
(✓ for name, ✓ for description)

- Direct Memory Access (DMA): This is a special kind of hardware that controls the flow of data between memory and some device controller without the constant intervention of the CPU.
- Busy waiting: Driver sits in a tight loop, continually polling the device.

(b) The CPU ***processes*** instructions in *three (3) phases*. **Name** the *three phases* and **provide** a description of each. **[06]**

Solution:
(✓ for name, ✓ for description)

Fetch - an instruction is fetched from primary memory.
Decode - the instruction (and any opcodes) are processed from the item taken from memory
Execute - the instruction is executed (run on the CPU) and the instruction pointer is incremented.

(c) **Name** one (1) type of ***operating system structure***. **[01]**

Solution:
(✓ each, any one)

- Monolithic
- Micro-kernel
- Layered
- Client-Server
- Virtual Machine

(d) **State** whether the following instructions should be run in ***user mode*** or ***kernel mode***.

i. Disable all interrupts **Solution:**Kernel **[01]**
ii. Read the time of day clock **Solution:**User **[01]**
iii. Set the time of day clock **Solution:**Kernel **[01]**
iv. Change the memory map **Solution:**Kernel **[01]**

Total: 15

QUESTION 2: Processes and Threads

(a) **Provide** a *definition* of ***daemon process***. **Provide** an example of this type of process. **[04]**

Solution:

(✓✓ each)

- A daemon is a process that runs in the background and has a specific function. Daemons usually start when the operating system is booted.
- Any valid example of a background task (no user interaction) (accept incoming email, listen to incoming web requests, print services, etc...)

(b) **List** two (2) *events* that result in ***process creation*** **[02]**

Solution:

(✓ each, any 2)

- System initialization.
- Execution of a process-creation system call by a running process.
- A user request to create a new process.
- Initiation of a batch job.

(c) **Describe** the role of the ***scheduler*** in an operating system and how it *executes* this role. **[04]**

Solution:

(✓✓ each)

- The scheduler is responsible to ensure each process or thread gets CPU time.
- It accomplishes this goal by ensuring each process gets assigned a certain amount of processing time, after which it is interrupted to ensure another process can run.

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

Process	Priority	Burst Time (msec)
A	1	12
B	0	2
C	1	10
D	2	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	2	5	5	5	5	2	2
Process	B	A	C	A	C	D	A
Priority when run	0	1	1	2	2	2	3

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: 208.

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

Convert:208 to binary: 1101 0000 (1 mark)

Lookup: 110 (6) (2 marks)

Result: 01 (Page frame) (1 marks)

Physical address: 011 0000 (1 mark)

Decimal: 48 (1 mark)

(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	282	210	0	1
B	160	261	0	0
C	134	323	1	1
D	141	363	1	0

Answer the following in context of page replacement algorithms.

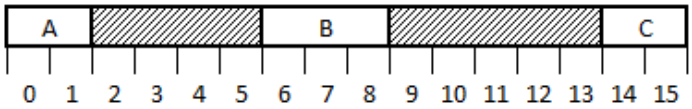
i. Which page will Not Recently Used (NRU) replace? **Solution:**B **[01]**

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

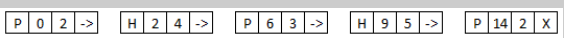
iii. Which page will Least Recently Used (LRU) replace? **Solution:**A **[01]**

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

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



Solution:



- (2) Mark for correct starting location of each item
- (2) Mark for correct length for each item
- (1) Mark for correct P or H indication

Total: 15

QUESTION 4: File Systems

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

```
E:\Inventory>tree \ /F
Folder PATH listing for volume SYSRCD603
Volume serial number is 269A-47DE
E:\
├───Inventory
│       HWList.xlsx
│
└───Finances
    ├───2020
    └───2019
            IT3B.pdf
```

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

ii. **Provide** the *relative path name* for the file called `IT3B.pdf` from the current working directory **[02]**

Solution:

- `E:\Inventory`(1 mark if it is an absolute path, but incorrect path) (2 marks if it is the correct path)
- `..\Finances\2019\IT3B.pdf`(1 mark if is a relative path, but incorrect) (2 marks if correct path)

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

File Name	Starting Block
.	17
..	19
File A	14
File B	5
File C	18

Table 1: Directory

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

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]**

Solution:

- 17
- 19
- 14,8,12,7,10 (2 marks)
- Should be a table with the top entry ”Attributes” and then, 5,15,9,13 (1 Mark for the table, 1 Mark for Attributes, 1 Mark for data blocks)

(c) **Name and briefly discuss** two methods that increases file system performance. **[04]**

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.

Total: 15

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]**

```
1  void write_string( int colour, const char *string )
2  {
3      const int VIDEO_MEMORY = 0xB8000
4      volatile char *video = (volatile char*)VIDEO_MEMORY;
5      while( *string != 0 )
6      {
7          *video++ = *string++;
8          *video++ = colour;
9      }
10 }
```

Discuss this method used in the code used by the software to interface with the device controller's registers and buffers.
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.

Solution:

- Memory Mapped (1)
- IO devices are mapped to memory address ranges (1)
- Simple high level programming instructions can be used to interface with the device (1)
- Special care should be taken to ensure that control registers are not cached (1)
- 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. (1)
- Special care must be taken to ensure that some memory addresses be interfaced on the IO peripheral bus and not the memory bus. (1)

(b) **Describe** the different steps that may occur when a fingerprint image scanner is used to scan a fingerprint into the computer. **[06]**
The computer architecture makes use of a ***Direct Memory Access (DMA)*** controller.
Your description must apply to this specific scenario.

Solution:
- The CPU programs the DMA controller.(1)
- The DMA controller requests a memory transfer from the scanner controller.(1)
- The scanner controller reads the data from the scanner and transfers the data to memory.(1)
- As soon as the fingerprint image has been transferred to memory, the DMA controller sends and ACK to the interrupt controller. (2)
- The interrupt controller sends an interrupt to the CPU to continue the scanner driver.(1)

(c) **Discuss *Uniform interfacing*** as a strategy for device independent software. Include in your discussion the following aspects: **[04]**
- The need for uniform interfacing.
- How this is achieved.

Solution:
Uniform interfacing is necessary (2 marks):
- Without uniform interfacing the operating system developer will have to modify the kernel for each device driver (2).
- Driver writers know what is expected of them.

How this is achieved (2 marks)
- Devices are grouped into categories or classes.(1)
- A standard set of functions for each device class is defined by the OS designers.(1)

(d) On an imaginary disk with 40 cylinders a request comes in to read cylinder 19. While the hard disk is busy servicing the request on cylinder 19, requests to the following cylinders come in: ***19,20,33,8,16.23***. **[02]**
Given these cylinders, if the operating system uses the ***elevator*** algorithm, **write** the order in which the cylinders will be serviced. The directional bit for the elevator algorithm is currently set to **1**, which indicates an **upwards** (ascending) direction.
Write only the cylinder numbers in order of service.

Solution:
19,20,23,33,16,8

(e) **Briefly describe** what a mickey is in mouse software and why this is important for mouse software. **[03]**

Solution:
- Mickey: The minimum value (movement) a mouse has to detect before it is sent to the computer. (2)
- The changes in the X and Y values (greater than a mickey) determine the movement of the mouse. (1)

Total: 20

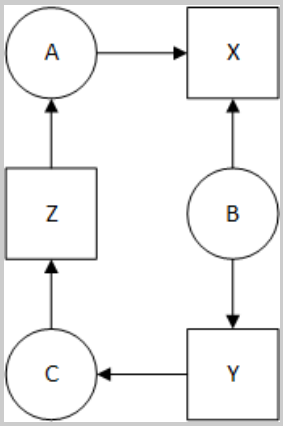
QUESTION 6: Deadlocks

(a) **Draw** a ***resource allocation graph*** for the state provided below, **and** specify whether the system is in a deadlock. **[04]**

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

Solution:

- Processes as circles (1 mark)
- Resources as squares (1 mark)
- Correct arrows (1 mark)
- System is not deadlocked (1 mark)



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

	Printers	CD-ROMS	Scanner	Feeder
E =	1	5	7	3

	Printers	CD-ROMS	Scanner	Feeder
A =	0	0	5	0

Current allocation matrix

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

Request matrix

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

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**.

Solution:

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

This scenario is not deadlocked (1 mark)

(c) You are employed as a system administrator for a company. The organisation wrote a Customer Booking system that requires a number of resources in the computer. Each customer that logs in, creates a small instance of a customer process. Each instance may require a number of core database tables, which must be locked when updated. **[06]**

Beta testing has shown that multiple customer instances can cause the system to hang, because each instance may lock separate tables that may be requested by another process instance.

You are currently contemplating redesigning the system to either avoid deadlocks or prevent deadlocks. **Which** one of these two solutions would you implement and motivate why.

Solution:

The student can choose any of the two solutions, but valid arguments must be provided.

2 marks are to be awarded for the argument.
4 marks are to be awarded for valid statements.

Avoiding

- There are really great algorithms that helps to avoid the problem.
- One such example is the banker's algorithm.
- The problem is that most systems do not know ahead of time how many resources they may require and
- Resources are also very dynamic. Users may continue to log in or log out of a system, changing the existing resource requirements.
- It is really difficult to redesign the system to avoid deadlocks

Preventing

- There are four potential approaches here:
- Attack (i) mutual exclusion, (ii) hold and wait, (iii) non-preemptive resource and (iv) circular wait.
- It may be possible to redesign the system to implement one of the four approaches.
- It may require a whole redesign, but is a bit more valid than trying to avoid.

Total: 15

QUESTION 7: Virtualisation and MPS

(a) **Discuss *Type-1 hypervisors***. **[06]**

Include in your discussion the following aspects:

- The location of the hypervisor.
- The location of the guest virtual user and virtual kernel modes.
- How privileged instructions in the guest operating system is captured by the hypervisor

Solution:

(6 marks allocated across the following aspects)

- The hypervisor runs on the CPU's kernel mode.
- The hypervisor runs above the hardware.

- The guest kernel and user modes run in the CPUs user mode
- When the guest os executes a kernel privilege instruction, it traps to the hypervisor.
- The hypervisor captures the trapped instruction and translate and executes it

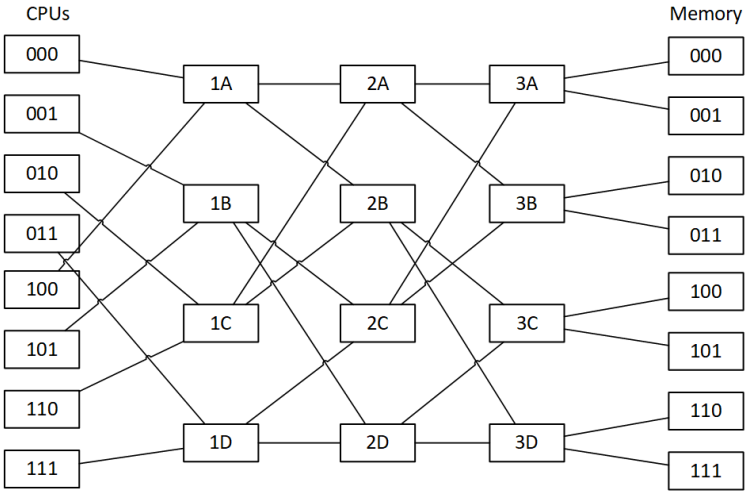
(b) **Briefly describe** how memory may be organised on UMA multiprocessors with a bus-based architecture **[05]**

Solution:

There are three general models:

- Without caching, and shared memory (1)
- With caching and shared memory(1)
- With caching (1) and each CPU also has private memory (1) modules but also shared memory (1).

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



i. **Which** switches will be accessed when CPU 011 needs to access Memory 010. **[01]**

Solution:
1D 2C 3B

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

Solution:
1A 2B 3∈D

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

Solution:
Yes (1 mark), No switches are shared. (only 1 mark)

Total: 15

QUESTION 8: Security

(a) **Describe** how a hash value can provide a certain level of integrity of a document. **[03]**

Solution:

- The original document's hash is stored.
- To determine if the document has been modified, the document is again hashed.
- The hash just calculated is compared to the hash that is stored. If they are the same then it may not have been modified.

(b) Given the following protection matrix. **List** the ***Access Control List (ACL)*** for the files described in the matrix. **[03]**

	marks.xls	exam.doc	assignment.doc
Student1			Read
Student2		Read and Write	
Student3	Read and Write	Read and Write	Read

Solution:

- marks.xls: Student3:RW
- exam.doc: Student2:RW;Student3:RW
- assignment.doc: Student1:R;Student3:R

(c) Given the following mono alphabetic substitution cipher, and cipher text. **Provide** the plain text for the cipher text. **[02]**

Key: A -> E **Ciphertext**: ibeqw **Solution:**exams

(d) In Asymmetric Cryptography, John signed a message. Alice would like to confirm that John really sent the message. **Whose** key and **what** key must be used to confirm the origin? **[02]**

Solution:

John's (1)
Public (1)

Total: 10

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]**

Solution:

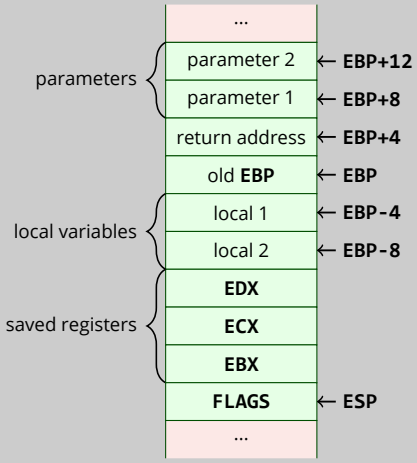
(✓ each)

- FPU needs to be initialised **FINIT**.
- **FDIV** used on FPU.
- Values need to be loaded onto the FPU stack (ST0 and ST1)
- Answer remains on FPU stack and needs to be stored or popped off (ST0).
- Only IEEEFloats are involved on FPU itself.

(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 testFast(int* source, int value)
2 {
3   int ret = 5;
4   int ter = 87;
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 -56.375_{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: $-56.375_{10} = 111000.0110_2$ ✓

Scientific notation: $1.110000110_2 \times 2^5$ ✓

S bit = 1 (negative number) ✓
E bits = $5 + 127 = 10000100_2$ ✓
F bits = 110000110_2 padded with 0

Binary : 1100 0010 0110 0001 1000 0000 0000 0000
Hex : 0xC2618000 ✓

Total: 15

QUESTION 10: 80x86 Cold code

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

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

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

The function will add each element in the array by 7. The function operates recursively. Perform ***tros*** only in the value at `arr[n]` and call the ***tros*** 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)

Solution:

- **Entry code**
  - Setup stack frame ✓
  - Push registers and flags ✓
- **Base case**
  - Test condition using **CMP** ✓
  - Jump to end of function ✓
- **Perform add**
  - Get correct address of element ✓✓
  - Modify value ✓
  - Save value in correct address ✓
- **Recursive call**
  - Push parameters in reverse order ✓✓ (if order is incorrect then (✓))
  - **CALL *tros*** ✓
- **Exit code**
  - Pop registers and flags reverse order ✓
  - Destroy stack frame ✓
- **Return**
  - **RET** ✓
  - Correct operand for **RET** ✓

Total: 15

~~ THE END ~~