
# FACULTY OF SCIENCE

## ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

| | |
|---|---|
| **MODULE** | **COMPUTER SCIENCE 3B** CSC3B |
| **CAMPUS** | AUCKLAND PARK CAMPUS (APK) |
| **EXAM** | NOVEMBER 2019 |
| **DATE:** 2019-11-11 | **SESSION:** Morning (08h30 - 11h30) |
| **ASSESSOR(S):** | **DR. J. DU TOIT**<br>**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) **Compare *microkernels*** and ***layered kernels*** in terms of *system services* and *inter-kernel communication*. You may use a diagram to aid your comparison. **[04]**

(b) ***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]**

(c) **Name** the three (3) *groups* of ***system calls***. **[03]**

(d) **Discuss** four (4) points to consider when dealing with a ***cache***. **[04]**

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

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

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

| Process | Priority | Burst Time |
|---|---|---|
| A | 0 | 2 |
| B | 1 | 9 |
| C | 2 | 8 |
| D | 1 | 12 |

Using the ***priority scheduling with round-robin equal priority*** 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: 173.***

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

(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 | 104 | 227 | 0 | 1 |
| B | 68 | 211 | 0 | 0 |
| C | 105 | 168 | 1 | 0 |
| D | 20 | 243 | 1 | 1 |

Answer the following in context of page replacement algorithms.

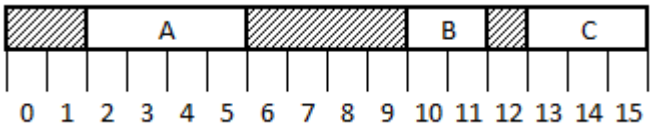
i. Which page will Not Recently Used (NRU) replace? **[01]**

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 bitmap representation of the memory state described below. Order the bits in an 8-bit configurations: **[05]**

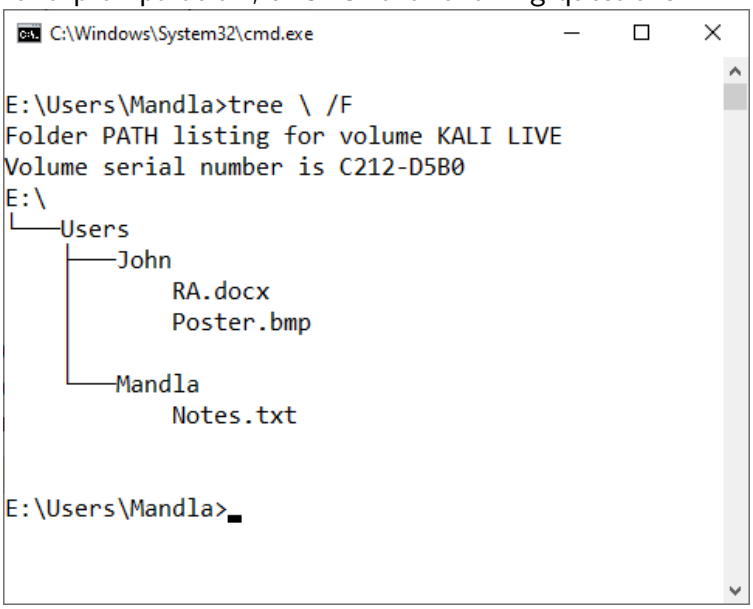



**Total: 15**

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

## QUESTION 4: File Systems

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




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

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

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

| File Name | Starting Block |
|---|---|
| . | 11 |
| .. | 5 |
| File A | 4 |
| File B | 8 |
| File C | 9 |

**Table 1:** Directory

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

**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) **List** and briefly describe the two special file types that are only available on UNIX operating system. **[04]**

**Total: 15**

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

## QUESTION 5: Input/Output

(a) The following assembly code reads a data signal from a keyboard controller **[05]**

```
;Reads data signal from keyboard controller
IN  AL,060h
MOV [KEYBUFF],AL
```

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 prints one character

```
copy_from_user(buffer,p,count);
enable_interrupts();
while(*printer_status_reg != READY);
*printer_data_register = p[0];
scheduler();
```

The second code segment is continuously called, until all the characters are printed.

```
if(count==0) {
  unblock_user();
}
else {
  *printer_data_register = p[i];
  count = count - 1;
  i = i + 1;
}
acknowledge_interrupt();
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) **Name** three properties of a precise interrupt **[03]**

(d) On a 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: ***40,27,10,37,12***. **[02]**

Given these requested cylinders, if the operating system uses the ***elevator algorithm***, **which** order will the cylinders be served in, given the elevator bit indicates ascending numbers?

(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.** ~~

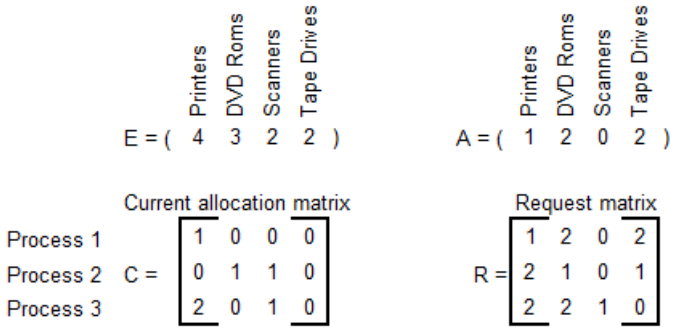
## QUESTION 6: Deadlocks

(a) **Provide** a definition for a deadlock **[02]**

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

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

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




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**.

(d) **List** the four conditions that must hold for a deadlock to occur **[04]**

**Total: 15**

## QUESTION 7: Virtualization and MPS

(a) **Which** assembly instruction, used while exiting a function, is a ***sensitive*** instruction? **[01]**

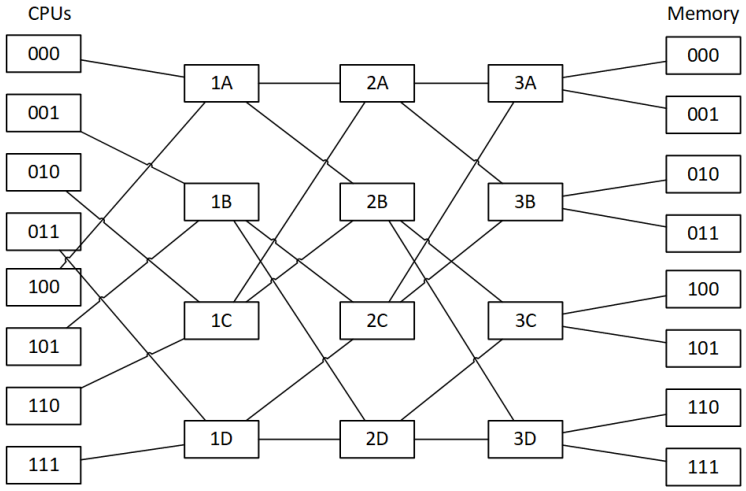
(b) Provide a **definition** of a **sensitive** instruction. **[03]**

(c) With the aid of a diagram, **describe** paravirtualisation. **[05]**

(d) **Briefly describe** the difference between uniform memory access (UMA) and non-uniform memory access (NUMA). **[02]**

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

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




i. **Which** switches will be accessed when CPU 110 needs to access Memory 111. **[01]**

ii. **Which** switches will be accessed when CPU 001 needs to access Memory 100. **[01]**

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

(f) **Discuss** the concept of master-slave multi-processing. Include in your discussion the following aspects:

i. The number of copies of the operating system in memory. **[01]**

ii. The role of the master processor **[01]**

iii. The role of the slave processors **[01]**

iv. One advantage of master-slave multiprocessing. **[01]**

v. One disadvantage of master-slave multiprocessing. **[01]**

**Total: 20**

## QUESTION 8: Security

(a) **List** three (3) security goals used by an operating system designer. **[03]**

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

| | Exam_Notes.doc | Solitaire.exe | BootlegMovie.mpg |
|---|---|---|---|
| **Student1** | Read and Write | | |
| **Student2** | Read | | Read |
| **Student3** | | Read, Write and Execute | Read |

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

**Key:** A -> F **Ciphertext**: wgmirgi

(d) In Asymmetric Cryptography, John wants to encrypt a message so that only Alice can read it. **Whose** key and **which** key must be used? **[02]**

**Total: 10**

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

## QUESTION 9: 80x86 Theory

(a) **Discuss** how ***division*** is handled when using *CPU 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 convert(int* alpha, int beta)
```

(c) **Show** the conversion of $49.1875_{10}$ 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 iterative ***splice*** function that takes the following parameters:

**arrRef** array address
**size** array length

The function will modulus each element in the array by 5. 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)

**Total: 15**

---

~~ **THE END** ~~