COMP30640 Operating Systems: Quiz 2

Exercise 1
Why is memory protection important in an operating system? What is the main memory protection mechanism?

Sample Solution 1
OS and user processes exist in the same physical memory: Some mechanism must protect the kernel (primary) memory from being accessed by user processes (and also the memory allocated to a process from being accessed by other processes)

Simplest scheme is to use base and bound registers: these registers are loaded by the OS before starting the execution of any program in user mode: base ≤ address < base + bound; otherwise exception raised.

Exercise 2 & 3
A computer uses a memory protection scheme based on the base and bound registers. If a program is 7,000 bytes long and it is loaded at address 35,000, what is the content of these registers?

Sample Solution 2 & 3
base = 35,000, bound = 7,000

Exercise 4
What is the purpose of system calls?

Sample Solution 4
System calls allow user-level processes to request services of the operating system.

Exercise 5
Define the concept of application programming interface (API) and why it is useful to use this concept for the interaction between the kernel mode and user mode.

Sample Solution 5
An API is a "contract" or agreement between some services (already implemented by the API) and an application. The API describes the services given and how to use them (e.g., parameters: inputs and output, protocols etc.). Those services are the building blocks of the application – the application does not need to know the details of the implementation of the services but can compose them freely.

Most system calls are accessed through an API in Operating Systems (e.g., POSIX for Unix systems). This gives flexibility to the user mode applications (they can use all the services provided at the kernel level by the API), transparence (the user mode application do not need to know the details of the implementation) and makes the kernel more secured (the user mode applications do not access directly the kernel level and do not need to know how it is implemented).

Exercise 6
Describe the CPU protection mechanism.

Sample Solution 6

We must ensure that the OS always maintains control: a user program might get stuck into an infinite loop and never return control to the OS. Timer: Generates an interrupt after a fixed or variable amount of execution time; OS may choose to treat the interrupt as a fatal error (and stop program execution) or allocate more execution time.