

Jeremiah Webb

CS420

Professor Al Amin

Exercise Questions:

1.1.1a

Abstraction can be used without virtualization. For example, an operation that needs to read a certain amount of bytes from a file, this can be implemented in low-level instructions, thus the new operation is not a virtual version of any physical object.

1.1.1b

Virtualization creates the illusion of a completely new object, however, this new object can have the same characteristics of a previous physical existing object. For example, virtual printing allows users to connect to multiple virtual printer clients, but these can point to a single physical printer. So yes, virtualization can be used without abstraction.

1.1.1c

Due to abstraction, a virtual object is a simpler version regarding features and a higher-level interface of an underlying physical object. For example, a shared network drive acts as a local storage device that is implemented in a separate remote network server.

1.1.2a

Multiprogramming enables multiple programs to be active in memory at the same time. While one is waiting for an I/O operation to be complete, another program can be run on the CPU. Time-sharing guarantees that each computation gets a specific turn to run on the CPU on time, that is most efficient.

1.2.1a

Traps & interrupts both suspend the operation of a current operation or instruction sequence and then divert control to a dedicated area within the OS kernel. An interrupt is caused by an external event to the current operation/execution. Interrupts are thus asynchronous to the current execution in that the interruption may occur unpredictably between any two instructions.

A trap, on the other hand, is a special case of interruption caused by an event internal to the current execution. Traps are synchronous to the current execution, the interruption is caused by the current execution. Interruption can be both error and intentional.

1.2.2a

Multiprogramming can function without interrupts. A context switch can occur when a computation terminates or blocks to wait for an I/O operation.

1.2.2b

Time-sharing requires a periodic interrupt to prevent long-running operations and computations from monopolizing a CPU and thus delaying the other needed computations for an unpredictable amount of time.