HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Benjamín Aage Birgisson

Exercise 1.1.1:

(a) Abstraction can be used without virtualization.
- *True, abstraction emphasizes on simplification by combining objects.*

(b) Virtualization can be used without abstraction.
- *False, virtualization uses (relies on) abstraction but emphasizes on diversification and replication.*

(c) Abstraction and virtualization can be used together.
- *Yes, they are sometimes used interchangeably and/or bundled together.*

Exercise 1.1.2:

(a)

- *They have that in common to execute code stored in memory, reading data from an input device, and output results to a printer or disk.*

- *Differences are that the time-sharing is an extension of multiprogramming (multiprogramming, which keeps several programs active in memory, maximizing the use of CPU and other resources), where the CPU is switched periodically among all active computations to guarantee acceptable response times to each user.*

Exercise 1.2.1:

(a)

- *They both stop an execution of current program and transfers control to the kernel.*

- *An interrupt may be triggered by a signal sent to the CPU from an external device (completion of an I/O operation, or implemented time-sharing), or the currently executing instruction.*

A trap is however an interrupt triggered by an error, or a supervisor call instruction (to the kernel).

Exercise 1.2.2:

(a)

- Yes, multiprogramming is possible without interrupts.

(b)

- No, it is one of the two most common uses of externally generated interrupts. For example, when an application (1) is waiting for the I/O device to complete, the kernel selects another application (2) to run. When the data transfer completes for application (1), the device issues an interrupt to the CPU, which stops application (2). Application (1) is then completed, and when the function terminates, execution can transfer back to the interrupted application (2) and continue.