1. a) False

   b) True

   c)

   Notes

   - **User Mode**
     - Is restricted
     - Executing code has no ability to *directly* access hardware or reference memory [1]
     - Crashes are always recoverable [1]
     - Is where most of the code on our computer / applications are executed [3]
   - **Kernel Mode**
     - Is previleged (non-restricted)
     - Executing code has complete and unrestricted access to the underlying hardware [3]
     - Is generally reserved for the lowest-level, most trusted functions of the operating system [1]
     - Is fatal to crash; it will halt the entire PC (i.e the blue screen of death) [3]
   - **Interrupt**
     - Are signals sent to the CPU by external devices, normally I/O devices. [2]
     - Tells the CPU to stop its current activities and execute the appropriate part of the operating system (**Interrupt Handler**). [2]
     - Has three different types [2]

       1) **Hardware Interupts**
          * Are generated by hardware devices to signal that they need some attention from the OS.
          * May be due to receiving some data

            Examples

            · Keystrokes on the keyboard
            · Receiving data on the ethernet card

          * May be due to completing a task which the operating system previous requested

            Examples

            Transfering data between the hard drive and memory

2) **Software Interupts**

   * Are generated by programs when a system call is requested

3) **Traps**

   * Are generated by the CPU itself
   * Indicate that some error or condition occured for which assistance from the operating system is needed

- **Content Switch**
  - Is switching from running a user level process to the OS kernel and often to other user processes before the current process is resumed
  - Happens during a timer interrupt or system call
  - Saves the following states for a process during a context switch
    * Stack Pointer
    * Program Counter
    * User Registers
    * Kernel State
  - May hinder performance
- **System Call**

  Example
  - `yield()`
    * Is a system call
    * Causes the calling thread to relinquish the CPU
    * Places the current thread at the end of the run queue
    * Schedules another thread to run
- **Threads**
  -

References

1) Coding Horror, Understanding User and Kernel Mode, link
2) Kansas State University, Basics of How Operating Systems Work, link
3) Kansas State University, Glossary, link