Virtualization

1. For each of the following virtualization categories, explain what interfaces are virtualized?
   A. Process virtual machine,
   B. System virtual machine,
   C. Containers

2. Explain briefly with examples (1) Process virtual machine, (2) System virtual machine, (3) Emulator, (4) Binary optimizer, (5) High-level Language Virtual Machine.

3. Which interface does a Process VM virtualize? Which interface does a System VM virtualize?

4. (a) How do Interpreters differ from Dynamic Binary Translators? (b) How do Binary Optimizers differ from Emulators?

5. What are the advantages and disadvantages of Classical System VMs compared to Para-virtualized VMs?

6. What is a co-designed virtual machine? Briefly describe and give an example.

7. What type of virtual machine (VM) is each of the following and why? Be as specific as possible. (a) Java Virtual Machine (JVM) (b) VMWare (c) Xen (d) Digital FX!32 (e) VirtualPC (f) (e) Transmeta Crusoe (Code Morphing)

8. Explain the difference between the concepts of full-virtualization and para-virtualization, giving at least one example of both virtualization techniques.

9. When you have to design a system that does emulation, under what circumstances would you opt for Interpretation and when would you opt for Binary Translation? Justify your answer.

10. Let’s say that you are asked to modify MacOS so that programs and libraries compiled on Windows OS could run natively on MacOS, meaning they should be executed as normal processes (i.e. without using a system virtual machine and Windows guest OS). What would be your high-level approach?
    OR
    Let’s say that you are asked to modify the Linux OS so that programs and libraries compiled on Windows OS could run natively on Linux, meaning they should be executed as normal programs (i.e. without using a system virtual machine and Windows guest OS). What would be your high-level approach?

11. Hypervisors often do everything that traditional operating systems do, such memory management, CPU scheduling, etc. So, what is the one key aspect by which a hypervisor differs from a traditional Operating Systems?

12. What is the difference between a Type-1 hypervisor and a Type-2 hypervisor? Give examples

13. For system virtual machines, explain how virtual memory addresses are translated to physical addresses when (a) hardware supports EPT/NPT (extended/nested page tables) and (b) hardware only supports traditional (non-nested) page tables.

14. How does Intel VTx extend the traditional CPU execution privilege levels to support system virtual machines?

15. Compare different approaches for virtualizing I/O devices for virtual machines.

16. Explain the key hardware-level virtualization support provided by Intel for
    (a) Memory translation for VMs
    (b) CPU privilege levels for guest OS execution
    (c) Direct I/O device access by VMs

17. Explain how the "problem" instructions (i.e. sensitive instructions that do not generate a trap in user mode) are handled in hypervisors that provide (a) full virtualization and (b) para-virtualization.

18. (a) What is a shadow page table? How is it constructed and/or updated by the hypervisor? (b) What type of hardware virtualization support is needed to avoid constructing shadow page-tables in full-virtualization?