# C211 – Operating Systems

## Tutorial: Virtual Machines
### – *Answers* –

1. Can a hypervisor virtualise the x86 architecture on top of any hardware architecture (e.g, ARM)?

   *No, because virtualization relies on the ability to execute most instructions natively. While it would be possible to "run" x86 code on top of other architectures (e.g., by translating each CISC instruction into several RISC instructions), this is referred to as simulation not virtualization.*

2. Consider a Type 1 hypervisor that can support up to $n$ virtual machines at the same time. PCs usually have a maximum of four disk primary partitions. Can $n$ be larger than 4? If so, where can the data be stored?

   *Virtual machines have nothing to do with disk partitions. The hypervisor can take a single disk partition and divide it up into subpartitions and give each virtual machine one of them. In principle, there can be hundreds of them. It can either statically partition the disk into n pieces or do this on demand as blocks are requested.*

3. VMWare does binary translation one basic block at a time, then it executes the block and starts translating the next one. Could it translate the entire program in advance and then execute it? If so, what are the advantages and disadvantages of each technique?

   *It could translate the entire program in advance. The reason for not doing that is that many programs have large pieces of code that are never executed. By translating basic blocks on demand, no unused code is ever translated. A potential disadvantage of on-demand translation is that it might be slightly less efficient to keep starting and stopping the translator, but this effect is probably small.*

4. Does binary translation slow down user-level function calls?

   *No, as user-level code is not translated during binary rewriting.*

5. Does it make sense to paravirtualize an operating system if the source code is available? What if it is not?

   *Paravirtualization requires source code availability. Linux has been paravirtualized precisely because the source code is available. Windows has not been paravirtualized because the source code is not available.*

6. PCs differ in minor ways at the very lowest level, things like how timers are managed, how interrupts are handled, and some of the details of DMA. Do these differences mean that virtualized appliances are not going to work well in practice? Explain your answer.

   *No, those differences might mean that porting the hypervisor from platform to platform requires a little bit of tweaking, but the machine emulated is the same in all cases, so the virtual appliance should work everywhere.*