First question

Why is the Intel x86 architecture especially hard to virtualize?

- Because its instruction set includes instructions that produce different results when executed in user mode and in privileged mode.
- (Virtualization is easier when all privileged instructions are caught by a trap when one tries to execute them in user mode.)
First question

- How does VMWare solve this problem?

  - VMWare uses dynamic "binary translation" when direct execution of code would not work
First question

- How does Xen solve this problem?
  
  - Xen exports a virtual machine abstraction that is “similar but not identical to the underlying hardware”
    - Paravirtualization
    - Requires some modifications to the guest OS
Second question

- What does Xen do to minimize *TLB flush overhead*?
  - The top 64MB region of each address space is reserved to Xen
    - Can execute Xen code without changing the page map and flushing the TLB
Third question

- Why does FAWN use a log-structured organization for its datastores?

  - Because it uses flash memory for its datastores and small random writes are very expensive on flash
Fourth question

- FAWN only stores a small fragment of the key of each item in its **in-memory hash tables**.
  - What is the main advantage of this solution?
  - It reduces the memory footprint of the hash tables thus reducing the node main memory requirements
Fourth question

- FAWN only stores a small fragment of the key of each item in its *in-memory hash tables*.
  - What is the resulting performance penalty?
    - It results in additional accesses to the secondary store
    - With the 15-bit key fragment, only one in 32,768 retrievals from the flash will require fetching an additional record.
Fifth question

- How does FAWN *delete* datalog entries?
  - It marks hash table entry invalid and adds a delete entry to the log (for durability)
Sixth question

Why does GFS not use data caching?

- Most applications stream through huge files
- It would be ineffective
Seventh question

How does a GFS recover \textit{after a crash}:
(one-line answers)

- \textit{Its file to chunk mappings}?
  - By replaying its operation log

- \textit{The locations of chunk replicas}?
  - By polling its chunk servers