Notes

Virtual memory - divided up into equal sized frames (power of 2)

Application
Typically, a frame has an offset of 12 bites (least significant), the rest of the most significant is for the page number
The application's memory is divided into frames of the same size

Memory
Frame number - most significant
Offset - least significant

Virtual memory is about how to swap from the application's point of view of memory to the actual memory
Application starts at address 0, the first page number is 0.
Application --> Memory doesn't always match 1 to 1
Application - logical address; Memory - physical address
From the offset of the logical address, that does map to the physical address offset
OS keeps track of all frames that are empty

Virtual memory works much like a cache
Still need replacement algorithms

Thrashing - working set is too large to be contained in memory or the program doesn't exhibit locality of reference
Results in a speed equaling to the slower memory
Solutions are to get more memory, reduce the number of running processes, better page replacement algorithms, or restructure programs that don't follow the locality of reference

Inverted page table - contains what's in physical memory rather than the application's memory
IPT - mapping of physical memory to virtual memory, needs a hash table along with the chain to figure out what the address is

Translation Lookaside Buffer - a special cache for the page table
A physical hardware to support the OS to find a cache page table

Multi-level page tables - improves performance in the x86 architecture

x86 Page Table

Fast Feedback Form

What is one topic that made the least sense to you during today's lecture?

The inverted page table. From what I've understood..

Page # --> hash --> linked list --> frame #
Hash is used to narrow down where the page may be located in the page table.
If the page is not contained within the linked list, a miss has occurred and the missing page must be loaded from virtual memory.

What is the one topic discussed today that you think will be the most pertinent to you in the future? Identify the one you think was the most important to you.

Virtual memory in general. I work with virtualization in my everyday job and virtual memory would be best to avoid to increase the virtual machine performance. In the current virtualization setup I work with, virtual disks are stored on a SAN and the physical hosts themselves have no hard disks, thus virtual memory increases the overhead of not only disk and CPU processing, but the network performance overhead as well.

Write two questions that you would like to see on our quiz next class period.

1. Data that is structured close together is following the rule of..? *Locality of reference*
2. When converting an address from virtual memory to physical memory, the offset portion of the address is changed. *False*