Notes for 10/28/09

On Pentium, page table addresses are 10 | 10 | 12 bits: first level | second level | page offset.

What does an entry in the second level page table look like?
20 bits to identify page | valid bit | dirty bit | reference bit | supervisor bit

Reference bit is used so that the OS can tell approximately how recently something was used.
Supervisor bit is used to distinguish kernel mode versus user mode.

TLB – translation lookaside buffer – a cache for virtual → physical mappings
Can typically hold 16-512 entries.
It is highly associative.
It takes 0.5 to 1 clock cycles to look up.
Misses require 10-100 cycles to look up.
Typically only 2% of lookups miss.

When you context switch in between processes, then you have to flush the TLB to preserve memory protection. Otherwise, you will have virtual → physical mappings from the previous process after a context switch, which could be used to access that process's memory.

Some TLB's have thread id's, which can be used to distinguish between mappings from different processes. In this case, you don't need to flush, as long as each process has a different thread id.

How does a page lookup typically work?

Hardware checks in TLB.
If miss, hardware uses first 10 bits to index the first level page table.
Then use hardware uses second 10 bits to index the second level page table.
If either of the page table levels miss (if the valid bit is not set on either level), then the hardware raises a page fault exception, and the kernel is trapped.
The OS tries to fix the problem. For instance, maybe the page exists, but the OS has not put the entry in the table yet. Or the page has been swapped to the hard disk. The OS maintains its own data structures to figure out whether the page has been swapped or not. If the OS fixes the problem, then the process is unaware of the page fault.
If the OS cannot fix the problem, then it will terminate the offending process such as by sending a SIGSEGV signal.