Virtual Memory

1. What are 3 specific benefits of using virtual memory?

   1. Bridges memory and disk in memory hierarchy.
   2. Simulates full address space for each process.
   3. Enforces protection between processes.

2. What should happen to the TLB when a new value is loaded into the page table address register?

   The valid bits of the TLB should all be set to 0. The page table entries in the TLB correspond to the old page table, so none of them are valid once the page table address register points to a different page table.

3. x86 has an "accessed" bit in each page table entry, which is like the dirty bit but set whenever a page is used (load or store). Why is this helpful when using memory as a cache for disk?

   It allows smarter replacements. We naturally want fewer misses (page faults), so if possible, we would want to replace a page table entry that hasn't been used. The "accessed" bit is one way of giving us enough information to implement this.

4. Fill out the table!

Virtual Address Bits	Physical Address Bits	Page Size	VPN Bits	PPN Bits	Bits per row of PT
32	32	16 KiB	18	18	22
32	26	8 KiB	19	13	17
36	32	32 KiB	21	17	21
40	36	32 KiB	25	21	25
64	40	64 KiB	48	24	28

Hamming ECC

Recall the basic structure of a Hamming code. Given bits $1, \ldots, m$, the bit at position 2^n is parity for all the bits with a 1 in position n. For example, the first bit is chosen such that the sum of all odd-numbered bits is even.

Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	P1	P2	D1	P4	D2	D3	D4	P8	D5	D6	D7	D8	D9	D10	D11
P1	X		X		X		X		X		X		X		X
P2		X	X			X	X			X	X			X	X
P4				X	X	X	X					X	X	X	X
P8								X	X	X	X	X	X	X	X

1. How many bits do we need to add to 0011_2 to allow single error correction?

   Parity Bits: 3

2. Which locations in 0011_2 would parity bits be included?

   Using P for parity bits: $PP0P011_2$

3. Which bits does each parity bit cover in 0011_2?

   Parity bit #1: 1, 3, 5, 7
   Parity bit #2: 2, 3, 6, 7
   Parity bit #3: 4, 5, 6, 7

4. Write the completed coded representation for 0011_2 to enable single error correction.

$1\underline{0}0\underline{0}011_2$

5. How can we enable an additional double error detection on top of this?

   Add an additional parity bit over the entire sequence.

6. Find the original bits given the following SEC Hamming Code: 0110111_2

   Parity group 1: error
   Parity group 2: okay
   Parity group 4: error
   Incorrect bit: $1 + 4 = 5$, change bit 5 from 1 to 0: 0110011_2
   $0\underline{1}1\underline{0}011_2 \rightarrow 1011_2$

7. Find the original bits given the following SEC Hamming Code: 1001000_2

   Parity group 1: error
   Parity group 2: okay
   Parity group 4: error
   Incorrect bit: $1 + 4 = 5$, change bit 5 from 1 to 0: 1001100_2
   $1\underline{0}0\underline{1}100_2 \rightarrow 0100_2$

8. Find the original bits given the following SEC Hamming Code: $0100110100001 10_2$

   Parity group 1: okay
   Parity group 2: error
   Parity group 4: okay
   Parity group 8: error
   Incorrect bit: $2 + 8 = 10$, change bit 10 from 0 to 1: $0100110101001 10_2$
   $0\underline{1}0\underline{0}110\underline{1}0100110_2 \rightarrow 01100100110_2$