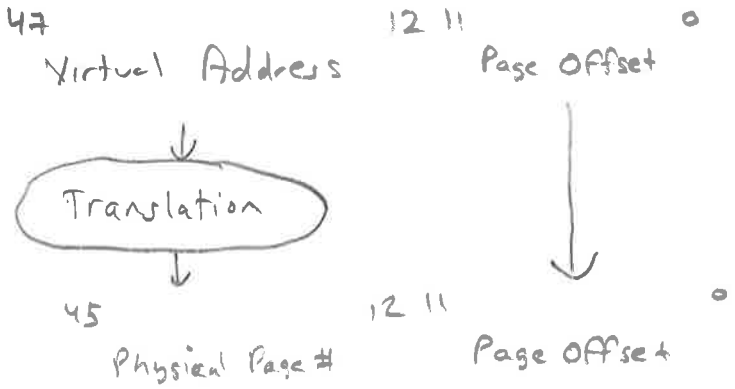
Given a 64-bit x86_64 Architecture where the page size is set at 4096, the number of physical address size in memory is $2^{46}$, and the virtual memory address size is $2^{48}$, derive the number of pages in the disk and draw the representation of the Virtual-to-Physical address translation.

$\log_2(4096) = 12 \Rightarrow 2^{12} = 4096$

VM address size = $48 - 12 = 36$

Physical Address size = $46 - 12 = 34$

$\Rightarrow$ # of Pages $= 2^{34}$



For a single level Page Table with 32-Bit machine with 4-KB pages, define the size of a single Page Table size. Then, what is the smallest amount of page table data we need to keep in memory for n-32 bit applications in a two-level Hierarchical Page Table? And how many applications can be implemented in this two-level Hierarchical Page Table in the equivalent memory space?

4-KB = 4096 $\Rightarrow \log_2(4096) = 12$

Single Page Table

$2^{32}/2^{12} =$ $2^{20}$ pages entries

Each page table entry $\Rightarrow \frac{32}{8} = 4$ bytes

$\Rightarrow 4 \times 2^{20} =$ $2^{22}$ bytes = Single Page Table size

Multi-Level

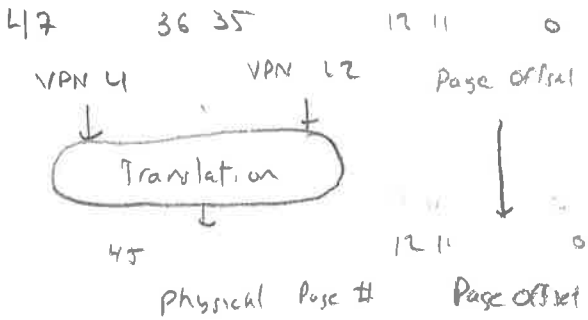
Level 1 - 4 KB

Level 2 - n * 4 KB

Each page table entry is 4 bytes

$\Rightarrow 4 + (4KB + n \cdot 4KB)$

Virtual Address




Number of programs

$2^{27} = 4 * (4KB + n \cdot 4KB)$

$2^{20} = 2^{12} + n \cdot 2^{12}$

$2^8 = 1 + n$

$n = 2^8 - 1 = 255$

$\Rightarrow$ Can map 255 programs using same space as one program before