Flat page table

- 1 entry per page in virtual address space even if the address space is not used
- Entry contains frame # + bits that tell us if the page is accessable (in memory or not, dirty etc).
  large enough to select a frame.

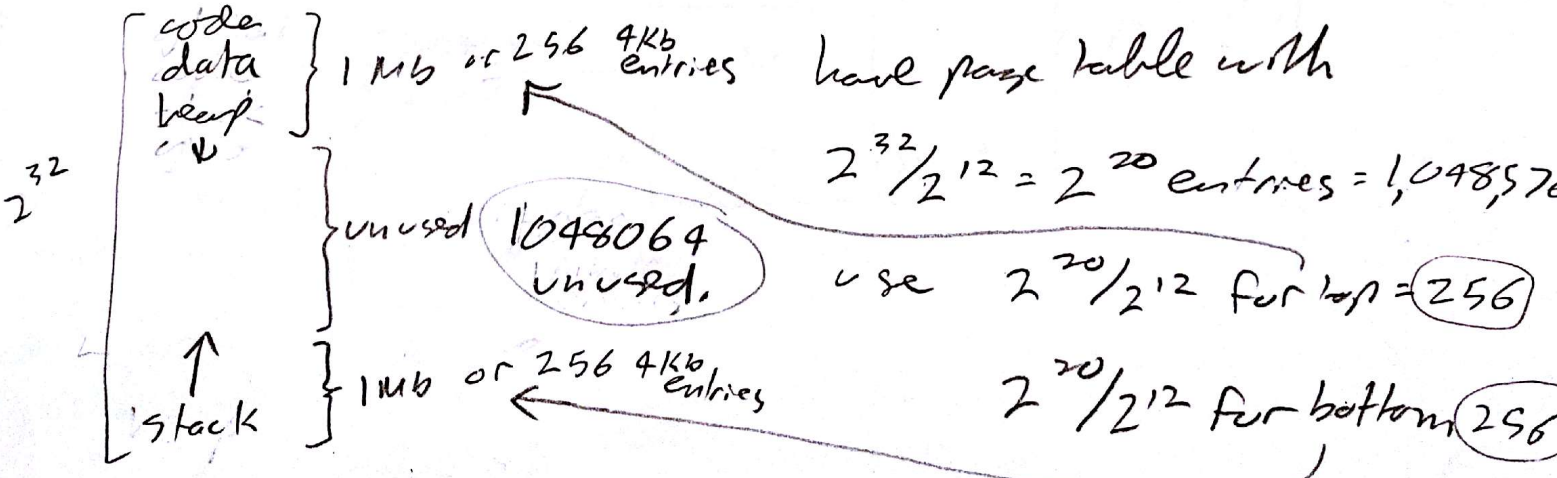
ex. overall size = $\frac{\text{virtual memory}}{\text{page size}}$ × size of entry (4 bytes = 32 bits)

4Gb = 2^{32}

typical 4 Kb = 2^{12}

$$= \frac{2^{32}}{2^{12}} * 4 \text{ bytes} = 2^{20} * 4 = 4 \text{ MByte}$$

Problem is page table is large even if process uses very little memory



so our 2 Mb app needs a page table of 4 Mbytes

what about 64 bit address space?

$$\frac{\text{virtual memory}}{\text{page size}} \quad \frac{2^{64}}{2^{12}} * 8 \text{ bytes (64 bits)} = 2^{55} =$$

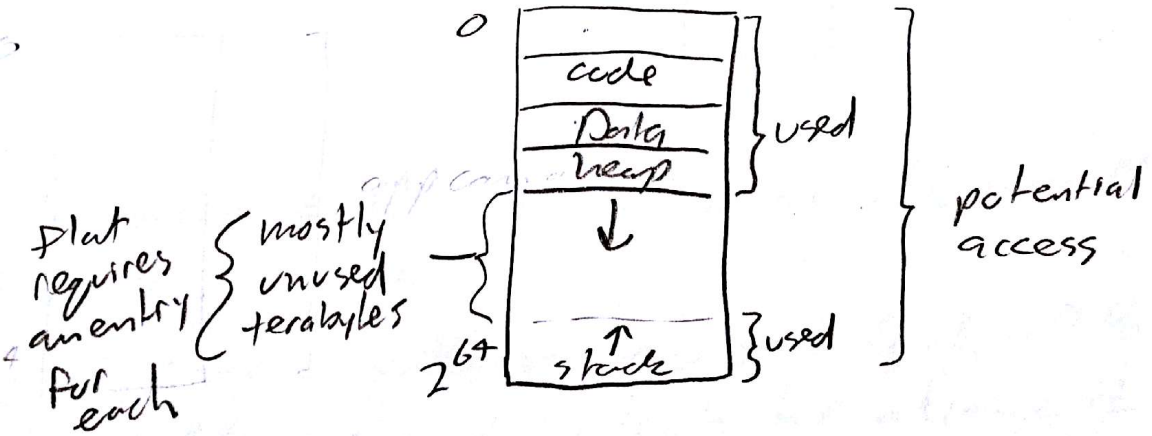
$= 3.6 \times 10^{16}$ bytes ⇒ bit more than we have (Smithburg)
$= 33$ million gigabytes

Multi level page table

- reduce the size of flat page tables
- size proportional to how much mem application can address, not how much its using!

32 bit ≒ 4-8 Mbytes
64 bit ≒ too big

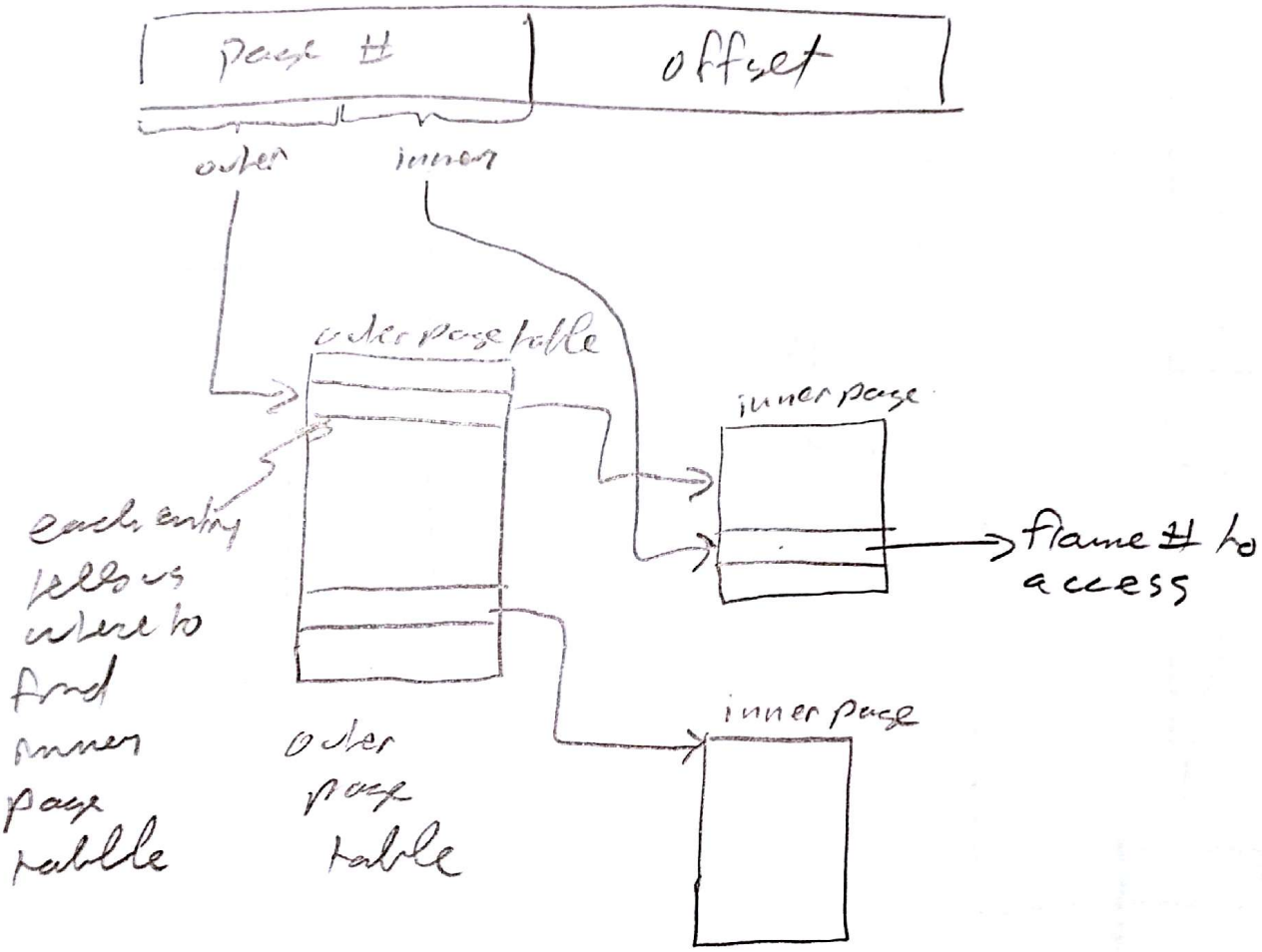
how to solve



Multi level

use bits to index tables

avoid entries for unused entries

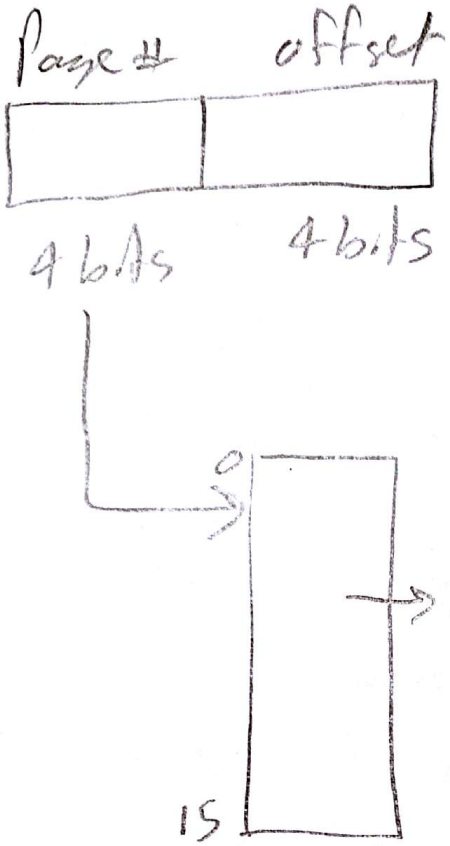


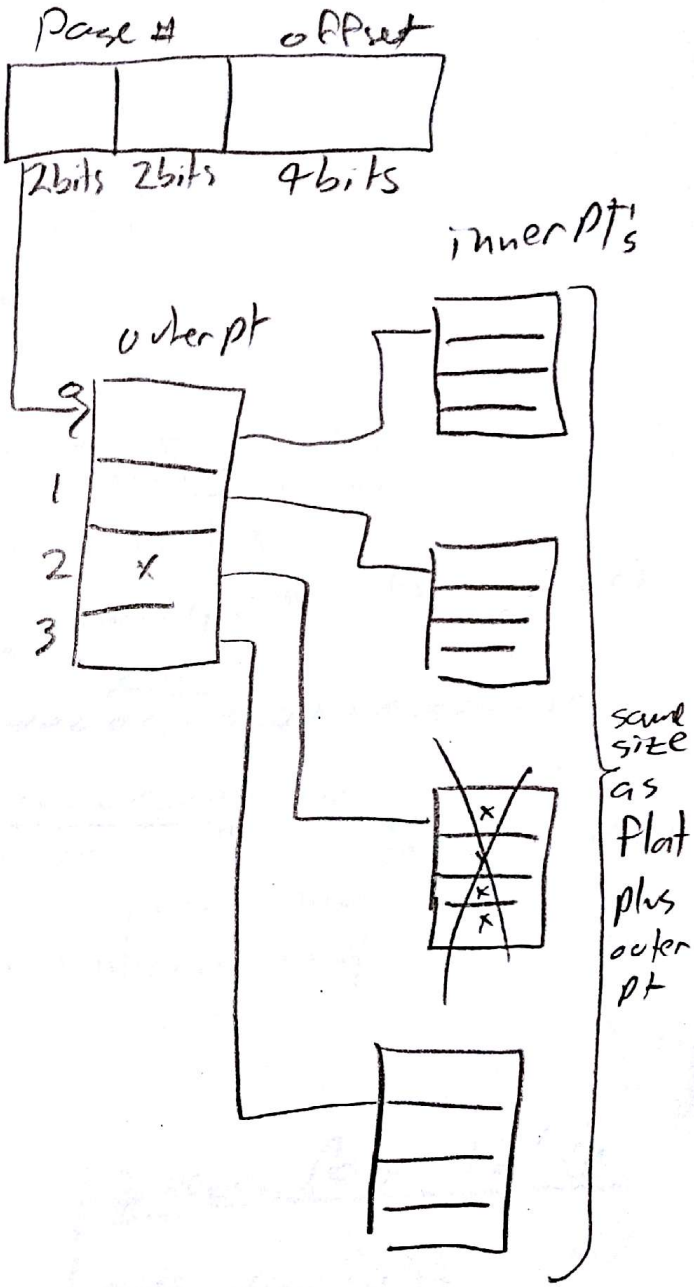
outer page # tells us which of the small inner page #'s to use.

inner page # tells us where in each of the inner page tables we need to look for a frame #

seems like we gained nothing have just as many page tables (plus the outer page table) as flat table

Two level page table simple example

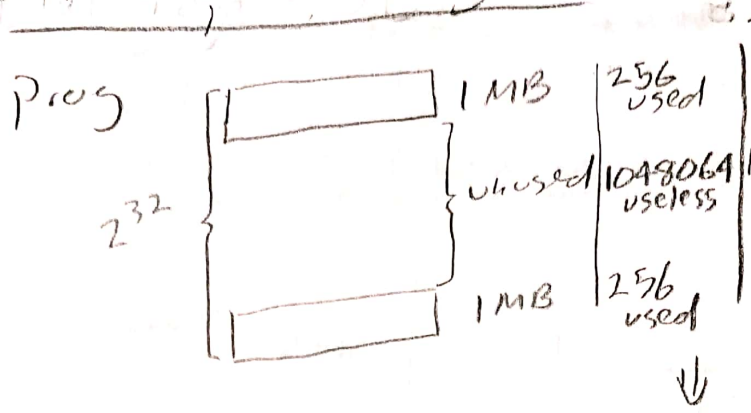




savings? when outer page table points to inner page tables that have no entries don't need to have those inner page tables

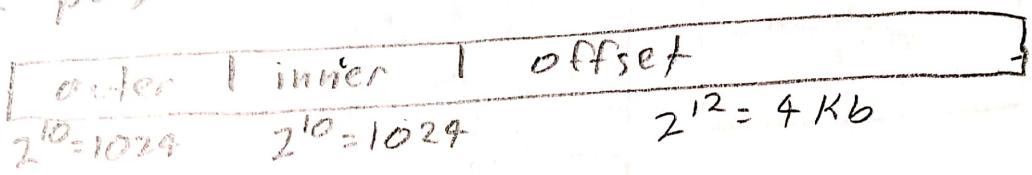
have 1 outer page table & some inner page tables

Where are Savings?



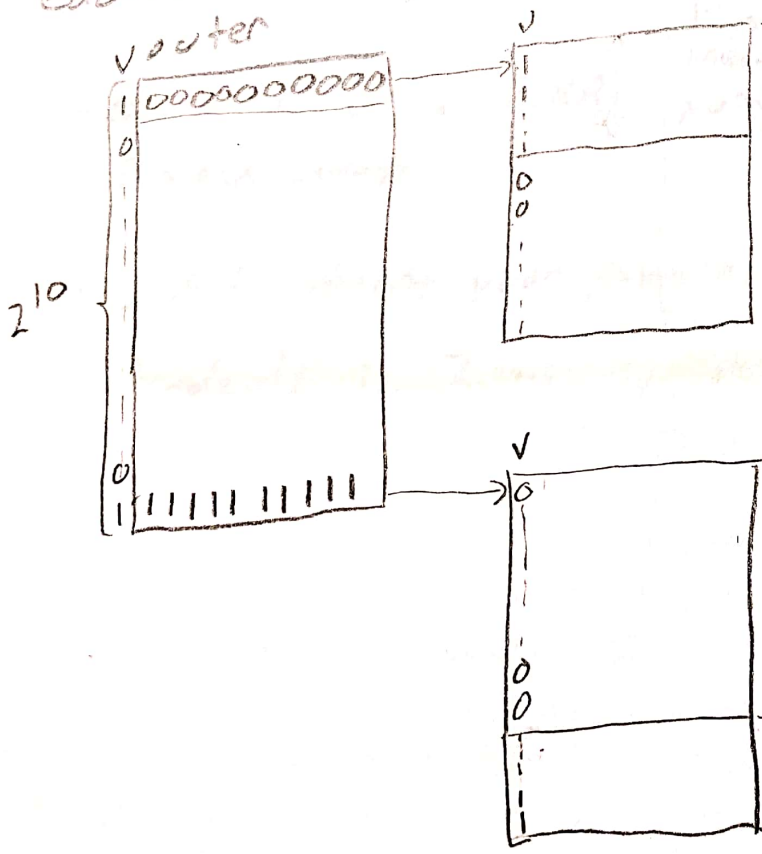
32 bit address space
4 Kb physical page (12 bits)
Flat PT = $2^{32}/2^{12} = 2^{20}$ entries

2 level page table



inner = outer entries = 1024

each outer entry can address $2^{22} = 4$ Mb



size

outer pt = $2^{10} \times 4 = 4$ Kb

inner pt's = $2 * 2^{10} \times 4 = 8$ Kb

total = 12 Kb
verses 4 Mb

This is why multilevel page tables are used!