

Virtual address

(page #, offset)

Physical address

(frame #, offset)

Page table

page #	frame #
0	
1	
2	
⋮	

Suppose we have a 16-bit address space
 pager = 1KB = 1 frame

Virtual address 4322

4096

226

128

98

64

34

32

2

0001000011100010

offset

0001000011100010

page 4

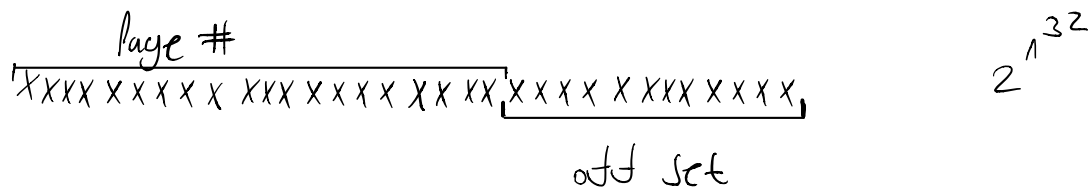
he talked about frames before this.

$$\begin{array}{r}
 3400 \\
 \underline{2048} \\
 1352 \\
 \underline{1024} \\
 328 \\
 \underline{256} \\
 72 \\
 \underline{64} \\
 8
 \end{array}
 = \overbrace{000011}^{\text{Page 3}} \overbrace{0101001000}^{\text{offset}} \text{ Virtual add.}$$

He also mentioned the physical address based on his chart too.

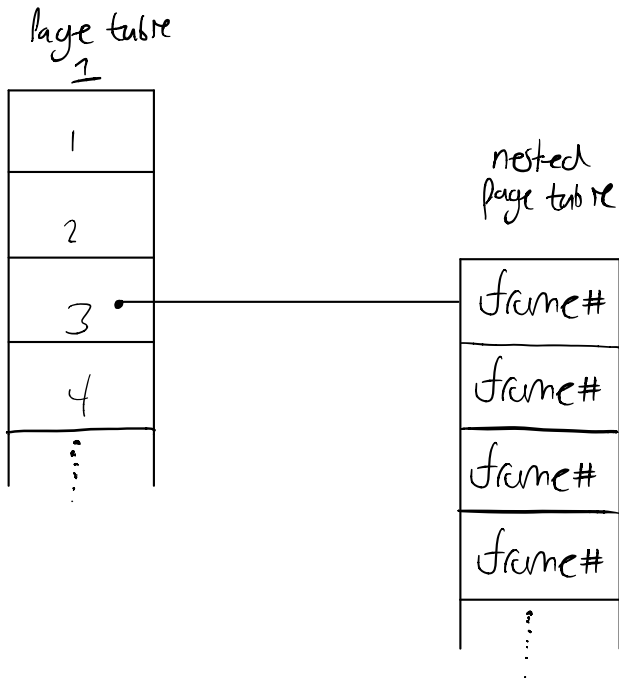
- Internal fragmentation - to much memory given to process
- External fragmentation -

32 bit address space
4KB pages



- you need a page table for each process running.

XXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX



page tables need to be fast.

- need to be in cache
- or a dedicated hardware
ie. translation look aside buffer - TLB

Multi level page tables are nested memory tables

frame physical memory OS-test 2 Wednesday 24th

Suppose we have a 32 bit address space 4kB pages = 2^{12} bytes

2^{20} pages 2^{20} pages

Contiguous space not required.

Inverted page tables we have 32 MB Ram
 $2^5 \cdot 2^{20}$ bytes

$$\frac{2^{25}}{2^{12}} = 2^{13} \text{ frames}$$

1 table for machine

each entry tells

° which process

° virtual page

Virtual address (page #, offset)

frame#

0	ProcessID	Page#
---	-----------	-------

1			
2			
3			

Segmentation

- Segments do not need to be the same size
- Not transparent to a programmer
- Compiler organizes data into segments
 - global variables
 - Subroutines
 - procedure call stack
 - local variables for subroutine
 - growing data structures
 - Shared data
- Virtual address (segment #, offset)
 segment table used to find addresses

Segment table

Segment #	address	Size bytes
0	1000	120
1	10000	250
2	6000	400

Virtual address (1, 46)

physical address (10000 + 46)

buffer overflow attack

Test synchronisation

Deadlock

Memory management,

try and read chapter 7 + problems 7.2, 7.6, 7.12
7.13, 7.14