

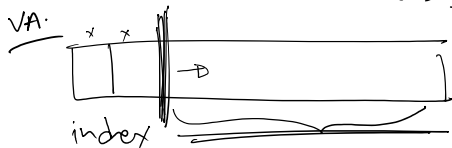
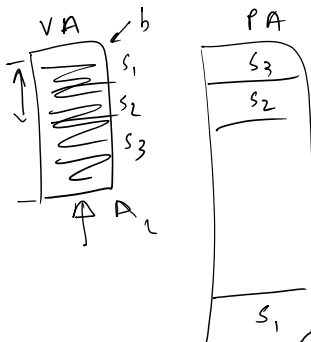
memory < virtualization mgmt.

Lecture 17.

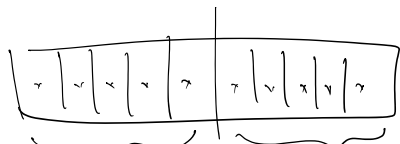
- segmentation

+ logical continuous VA regions
base + bound regs. for translation.

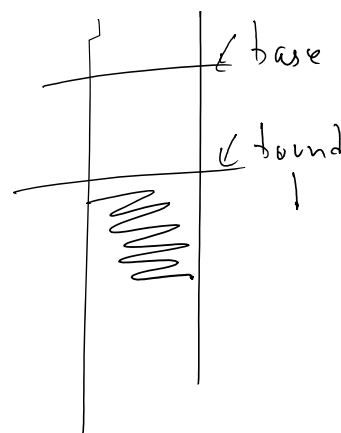
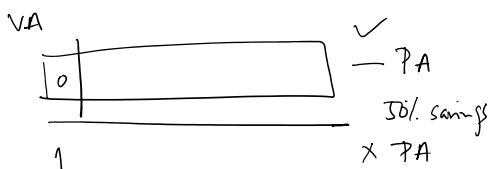
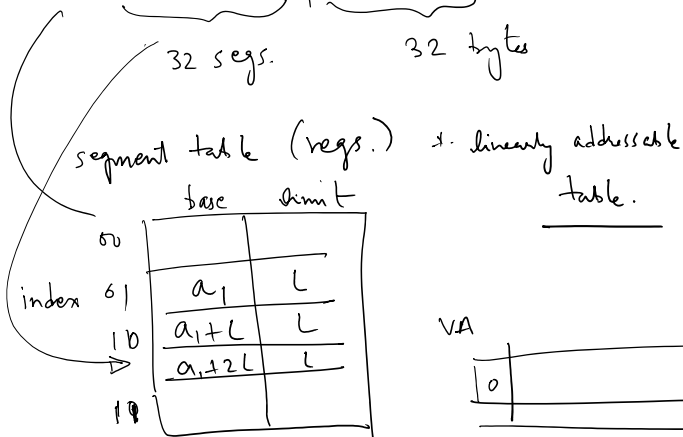
- per segment protection
- non-contiguous allocation.
- segment-level on-demand allocation
- variable sized segments
- segment-level sharing (read-only)
- fine vs. coarse grained segment sizes.



4 segments
+ regs. of base + bound



00 → base | limit
01 → y
10 → a
11 → b



~ 1 KB - segm. size
4 GB - addr. space
segment table entry 4 bytes

$$\text{size of seg. table} = \frac{4 \times 10^9}{10^3} = 4 \times 10^6 \times 4 = 16 \text{ MB}$$

issues / -ves w/ segmentation.
translation table overheads.

coarse ↔ finer

issues / -ves w segmentation

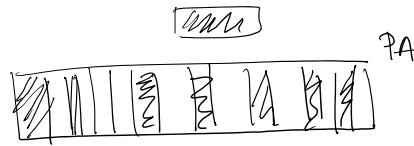
- segment table o/hds.

- utility of memory w/ large segments can be very low.

- (external) fragmentation

coarse $\longleftrightarrow$ finer

tradeoff

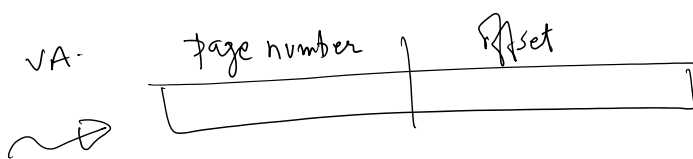


design 3: paging.

~ all allocations to a process are of fixed size granularities — a page.

~ process memory — collection of pages. (segment!)

translated & accessed via a page table.



single level
paging solⁿ.
 $\Updownarrow$
segmentation.

o/hds. $\leftarrow$ $\bullet$ $\rightarrow$ finer-level
of table control
'lookup' of memory