27/3/18

Segmentation

Q. On a system using simple segmentation, compute the physical address for each of the logical addresses given the following segment table if the addresses generate a segmentation fault indicates so.

Segment	Base	Length (limit register address)
0	330	124
1	576	211
2	111	99
3	498	302

Logical addresses : a) 0, 99 b) 2,78 c) 1,265 ∄
d) 3, 222 e) 0,111

⇒ Logical address : 0, 99
segment no ↵ ↳ offset

- offset checked with limit register value.
- check :- offset < limit

a) 99 < 124 ∴ can map ∴ 330 + 99 = 429.
phy. addr.

b) 78 < 99 ∴ can map ∴ 111 + 78 = 189

c) 265 ≮ 211 ∴ cannot map

d) 222 < 302 ∴ can map ∴ 498 + 222 = 720

(e) 111 < 124 ∴ can map ∴ 330 + 111 = 441

Q. On a system using paging and segmentation both, the virtual address space (logical address space) consists of 8 segments. where each segment can be upto 2^{29} bytes long. The hardware pages each segment into 256 byte pages. How many bits in the virtual address specify the

(a) segment no

(b) page no.

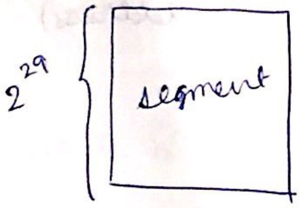
(c) offset within page

(d) entire virtual address.

Ans. : 8 segments - each of 2^{29} bytes

∴ No. of segments = 8

→ ∴ 3 bits to uniquely identify the segmen



2^{29} bytes long

page size = 256 = 2^8

(page size is offset size)

(c) offset within page = 8 bits

(a) segment = 3 bits

(b) $2^{29}/2^8$ = 21 pages

d) 3 bits + 21 bits + 8 bits = 32 bits

VIRTUAL MEMORY

- separation of user logical memory from physical memory
- Virtual memory can be implemented via

demand pageing

demand segmentation

Valid - Invalid bit

memory access time = 200 ns.

Avg. page fault = 8 msec.

EAT = ((-p) × 200 + p (8)

= 200 + p × 7999800

If one access out of 1000 causes a page fault, then EAT = 8.2 µs.
This is a slowdown by a factor of 40.
To keep slowdown to 10% $p < 0.0000025$

8. On a system using demand page memory, it gets 120 nsec to satisfy the memory request if the page is in memory. If the page is not in memory, the request takes 5 msec. What would be the page fault rate to ~~be~~ achieved an effective access time ~~time~~ of 1 µsec?

4/4/18

COW – copy On Write – allows both parent And child processes to initially share the same pages in memory.