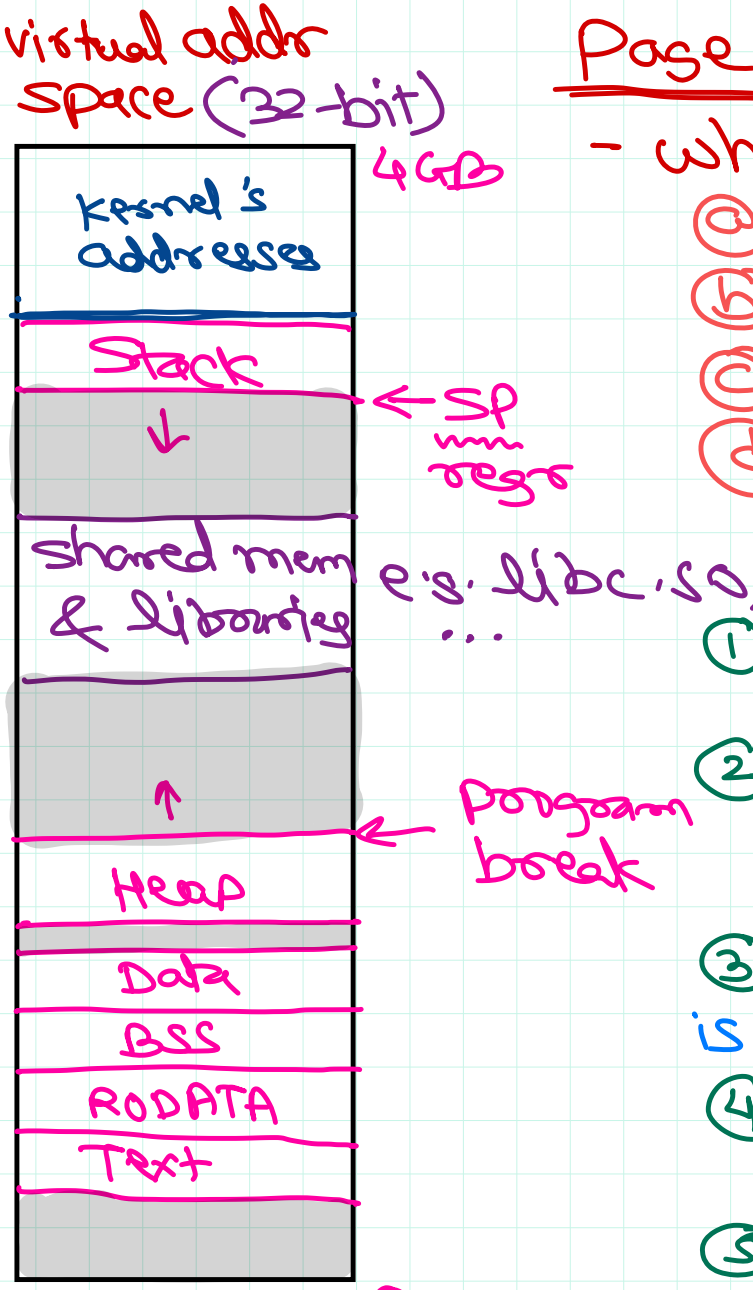
Embedded Operating Systems

Trainer: Nilesh Ghule



Page Table entry → Invalid ?

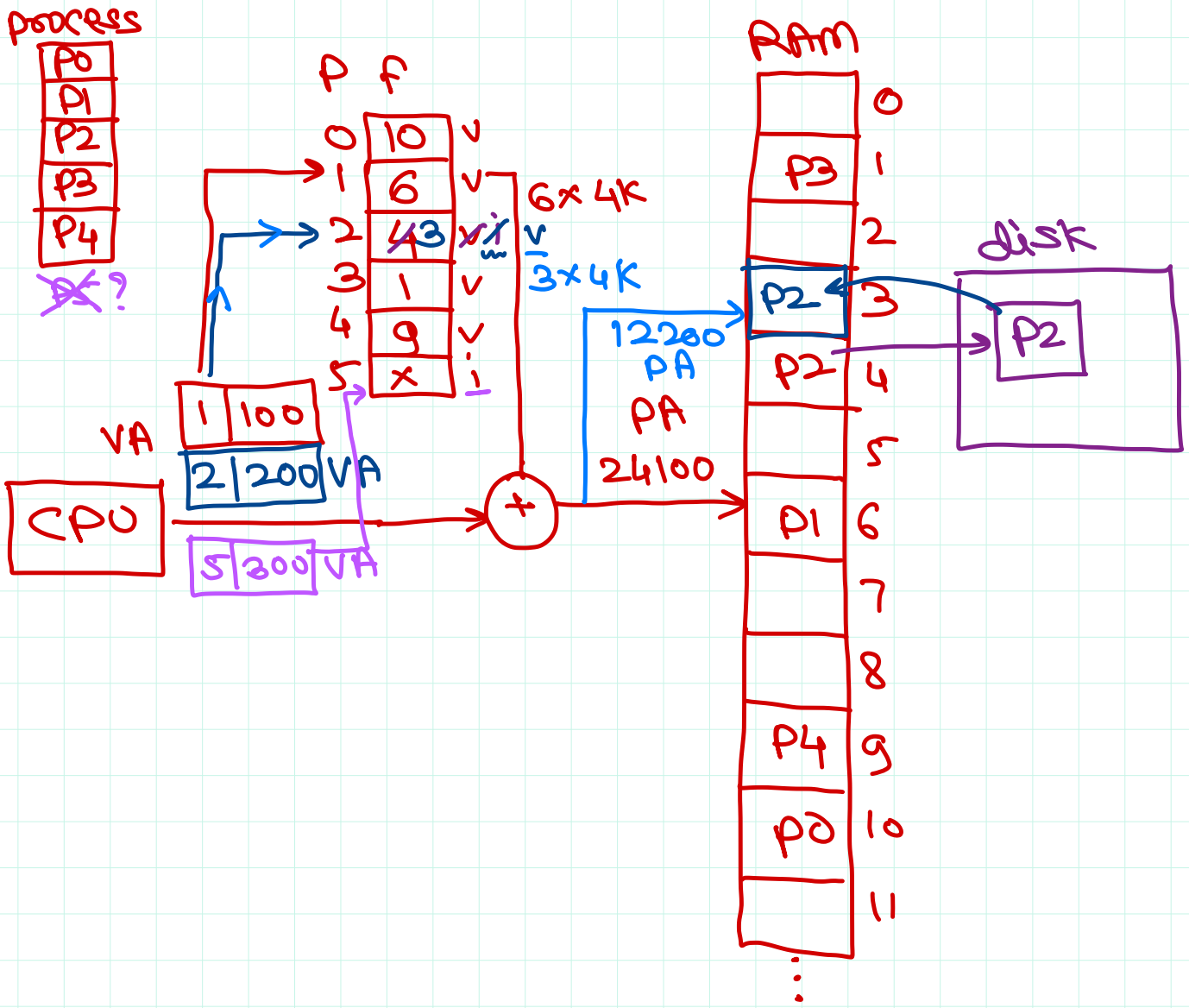
- when page is not present in RAM
  - (a) may be invalid page (dangling ptr).
  - (b) may not be allocated yet.
  - (c) may be in swap area
  - (d) may be in disk (in exe file)

When CPU req a page that is not present in main mem (i.e. PTE is 'invalid'), MMU raise an exception called as "page fault". ↓ OS page fault ex handler.

Page Fault Handling

1. check VA that caused the fault. If the addr is not valid (validity fault), then send SIGSEGV to process.
2. if addr is valid, check the permissions. If 'invalid perm (e.g. WR op on Rdonly page → protection fault), then send SIGSEGV to process.
3. allocate an empty frame for the page. if no frame is empty, swap out some page (of inactive process) & alloc frame.
4. if page is on disk/swap, load it into the allocated frame.
5. update frame addr & validity bit in page table entry.
6. re-start the instruction at which page fault occurs.

Page Fault Handling



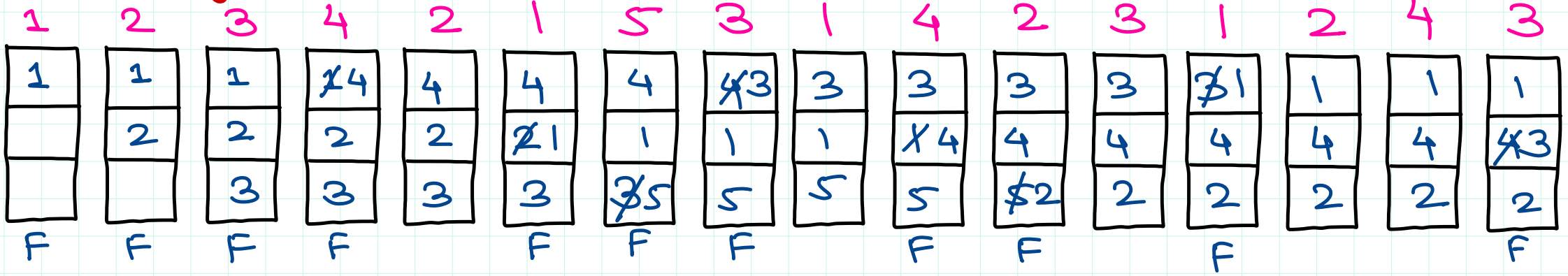
Nilesh Ghule

if there is shortage of memory, then some page(s) should be swapped out. This page is called as "victim page".

To decide the victim page, there are "page replacement algos".

① FIFO ② Optimal ③ LRU

FIFO page replacement algo.



page fault count = 11

if num of frames increase, num of page fault also increase - sometimes → Belady's anomaly.

Optimal Page Replacement → Min num of page faults.

Victim page → which not required in near future

1	2	3	4	2	1	5	3	1	4	2	3	1	2	4	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	X4	4
	2	2	2	2	2	~~2~~5	~~3~~3	3	3	3	3	3	3	3	3
		3	~~3~~4	4	4	4	4	4	4	~~4~~2	2	2	2	2	2
F	F	F	F			F	F			F				F	

page fault count = 8.

LRU Page Replacement

1	2	3	4	2	1	5	3	1	4	2	3	1	2	4	3
1	1	1	X4	4	4	~~4~~5	5	5	~~5~~4	4	4	~~4~~1	1	1	X3
	2	2	2	2	2	2	~~2~~3	3	3	~~3~~2	2	2	2	2	2
		3	3	3	~~3~~1	1	1	1	1	1	~~1~~3	3	3	~~3~~4	4
F	F	F	F		F	F	F		F	F	F	F		F	F

page fault count = 13

LRU impl technique

① stack based approach → 1, 3, 5 → O(n) → LRU, MRU

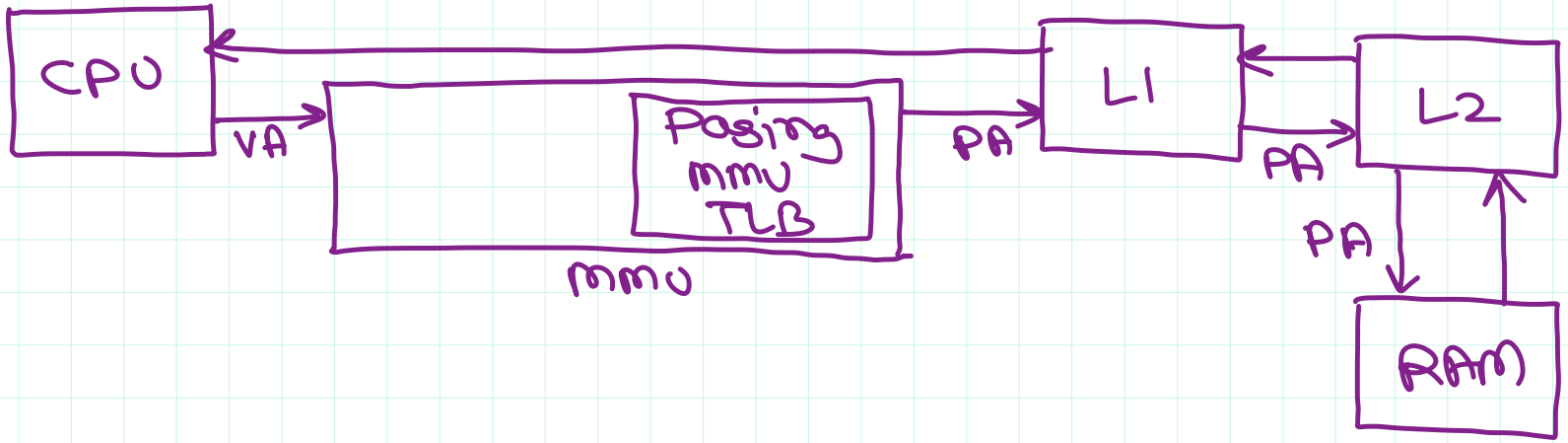
② counter based approach.

LFU, MFU ←

Nilesh Ghule

Second chance LRU

1. LRU gives less num of page faults.
2. However time complexity -> O(n) - slow
3. Approx LRU page replacement algo
   - like LRU (less page faults)
   - faster impl.
4. Second chance LRU - Approx LRU
   - time complexity - O(1).


CPU
VA
Paging
mmu
TLB
mmu
PA
L1
PA
L2
PA
RAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>