CSE 330: Operating Systems

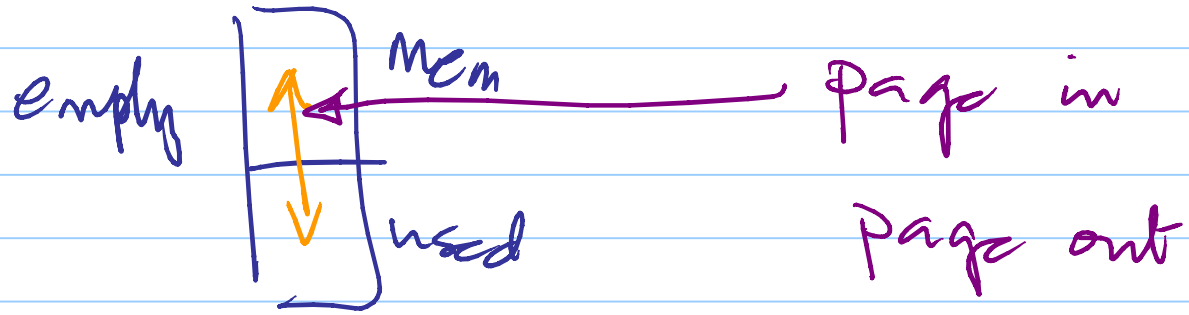
Class: 20 **Date:** 3/27

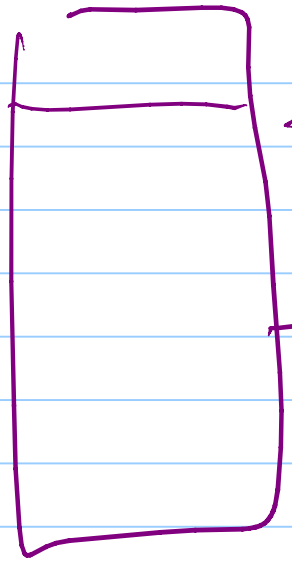
Spring 2018

Note Title

Page replacement

- memory gets full (or close to)
- due to demand paging





high water mark ↑ start page out

low water mark ↓ stop page out

→ which page to page out (victim)

page replacement

① FIFO (victim = oldest page)

each page is marked with the time it was paged-in (in PT)

→ suffers from Belady's Anomaly

→ old page may be valuable

→ not a stack policy

Stack policy → set of pages in memory for n pages is a subset of pages in mem for n+1 pages

the OPT (optimal) policy

[set of pages]

victim → the page that will not be used for the longest time (in future)

OPT policy replaces page P_i

another policy replaces P_j

P_j will be needed before P_i

↳ higher page fault or equal rate

reverse of OPT

- replace page that has not been used for the longest time

↓

LRU least recently used

- good? → past predicts the future
- stack? - yes

LRU is not implementable

"use a page" → mem reference

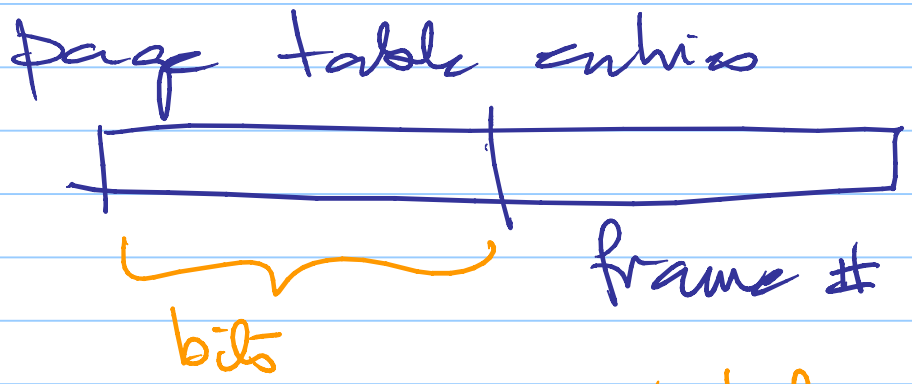
→ happens every ?

nanoseconds ?

→ timestamps → too much overhead.

[LRU approximations] → kind of like LRU

Additional reference bits algorithm

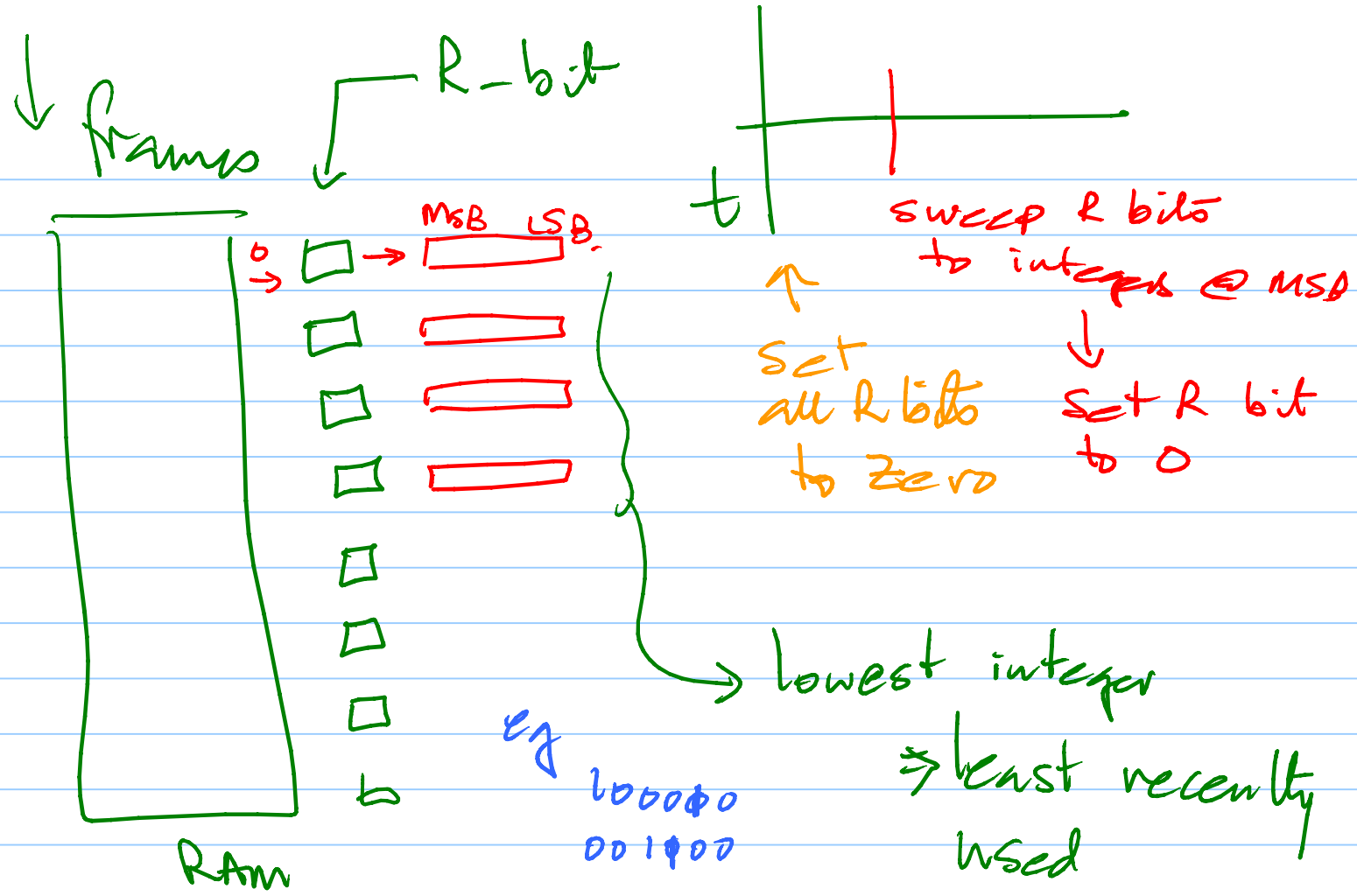


→ valid bit

r, w, x bits → permission

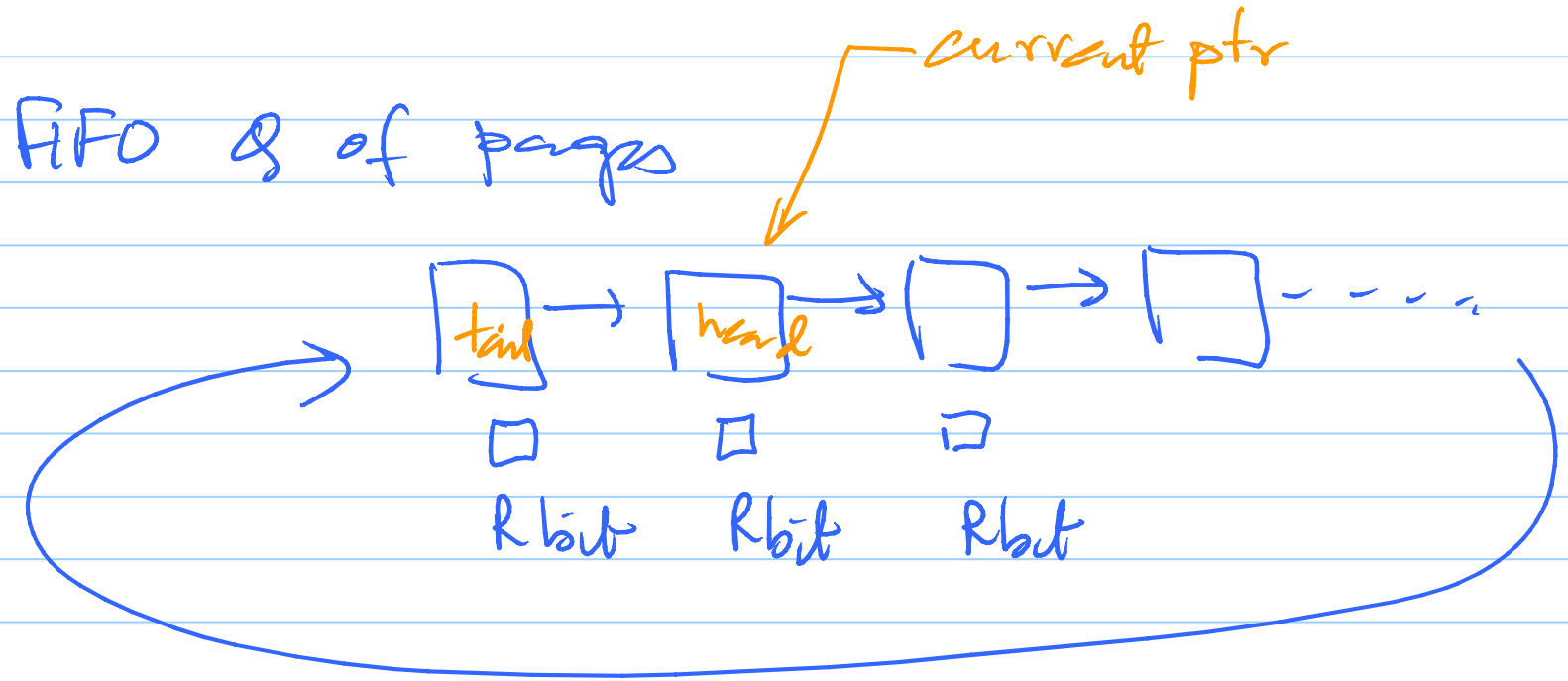
modify bit — has been modified

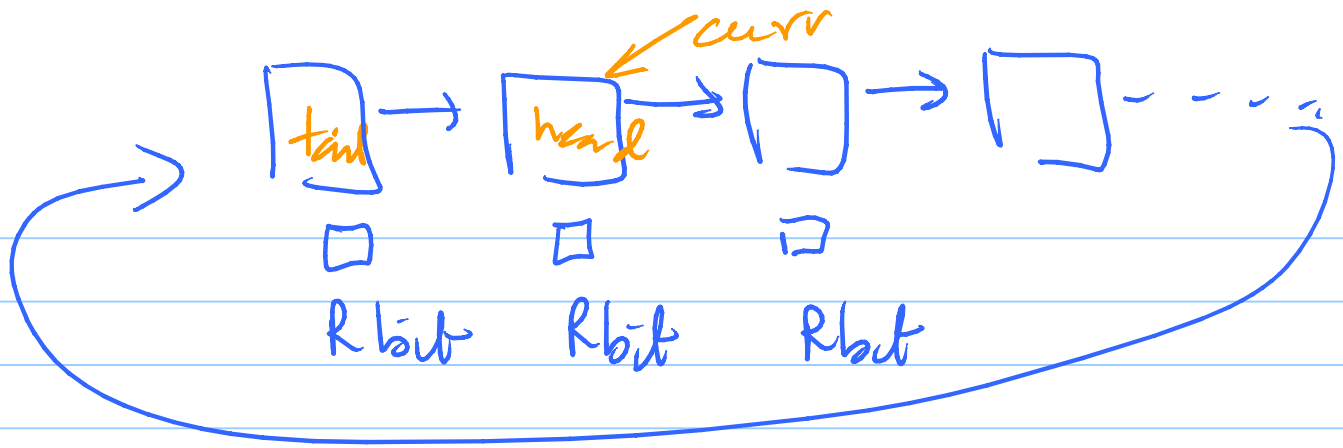
reference bit — " " referenced


frame
R-bit
MSB
LSB
0
RAM
t
sweep R bits to integers @ MSB
set all R bits to zero
set R bit to 0
lowest integer
⇒ least recently used
eg
100010
001100

Second chance replacement

↳ modify FIFO to act like LRU





Victim → page @ current ptr.

- check R bit
- if zero → victim
- if 1 change to 0 and move to next page

Second chance

DEADLOCKS

$x, y \leftarrow 1$

T_1

P(x)

P(y)

V(x)

V(y)

T_2

P(y)

P(x)

V(x)

V(y)

→ deadlock may happen

Deadlocks can/may/ happen

- deadlock prone
- depends on programs

When a deadlock happens

↳ forever

(stable property)

System model for resource allocation

1. Acquire a resource
2. Use the resource
3. release the resource (explicitly)

Necessary Conditions for deadlocks

(not sufficient) [4 4 are necessary]

- mutual exclusion
- no preemption
- hold & wait
- circular wait