CSE 330: Operating Systems

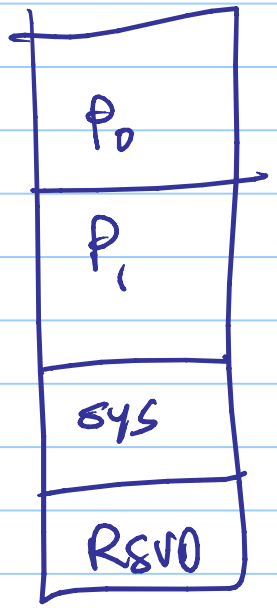
Class: 19 **Date:** 3/22

Spring 2018

Note Title

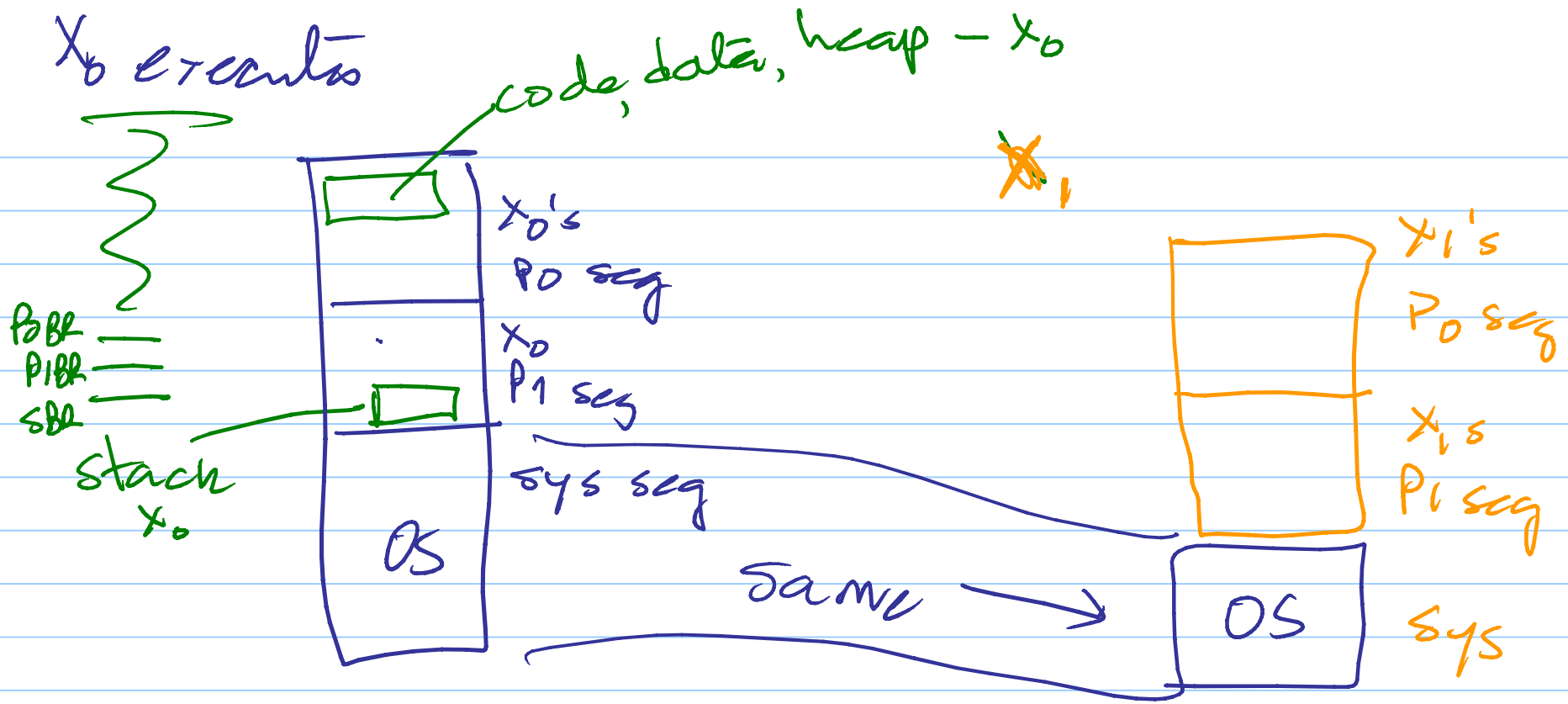
VAX virtual mem

P0 — 0 - 1 GB
P1 — 1 - 2 GB
SYS — 2 - 3 GB
RESVD — 3 - 4 GB

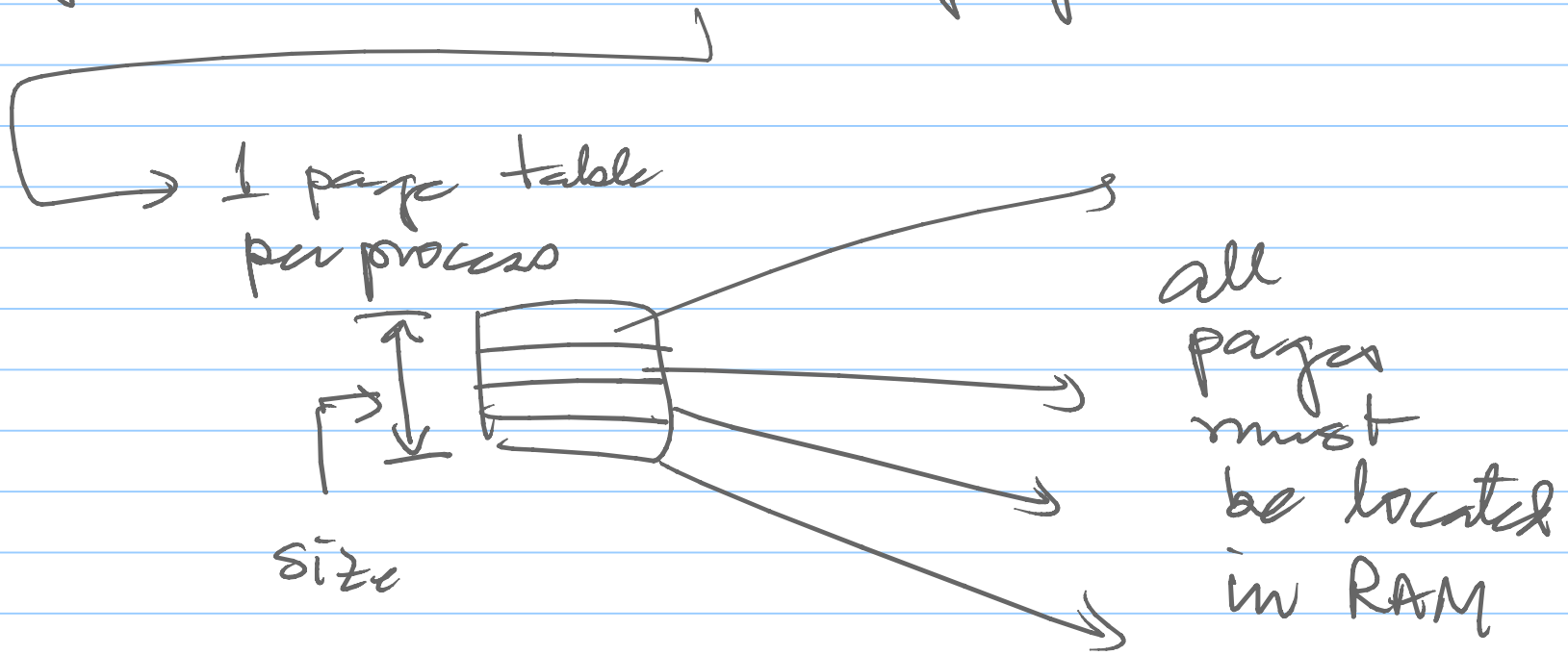


Segment table 3 entries

P0BR —
P1BR —
SBR —


X_0 executes
code, data, heap – X_0
P0BR
P1BR
SBR
stack
X_0
X_0's
P0 seg
X_0
P1 seg
sys seg
OS
save
OS
sys
X_1
X_1's
P0 seg
X_1's
P1 seg

paging systems → pure paging



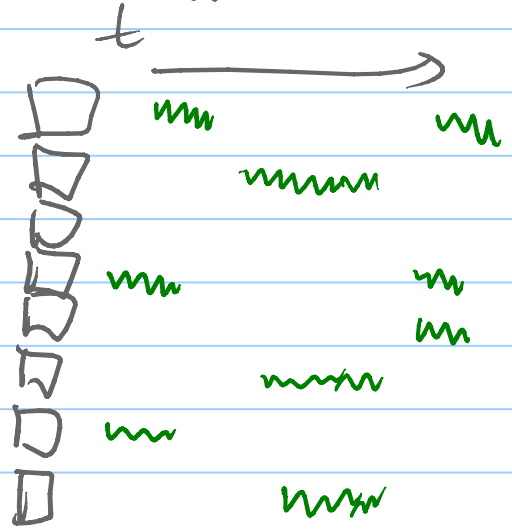
Virtual Memory

— amount of memory used is larger than available physical RAM

— Even 1 process may occupy more mem than all of RAM

Programs have the locality property

- amount of mem actively used is << total memory allocated
- working set

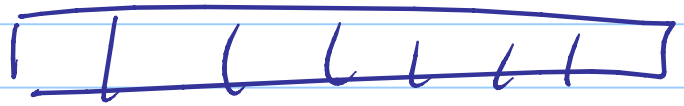


Implementing Virtual mem

- several methods
- Demand paging
  - ↳ keep a few pages in mem
    rest on disk
    - ↳ swap space

process loading

create mem map



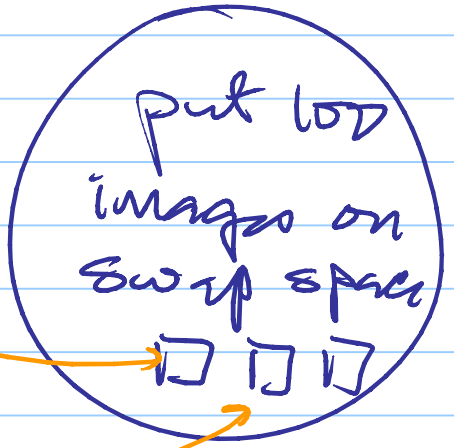
page table

VALID BIT

1 → in RAM

0 → not in RAM

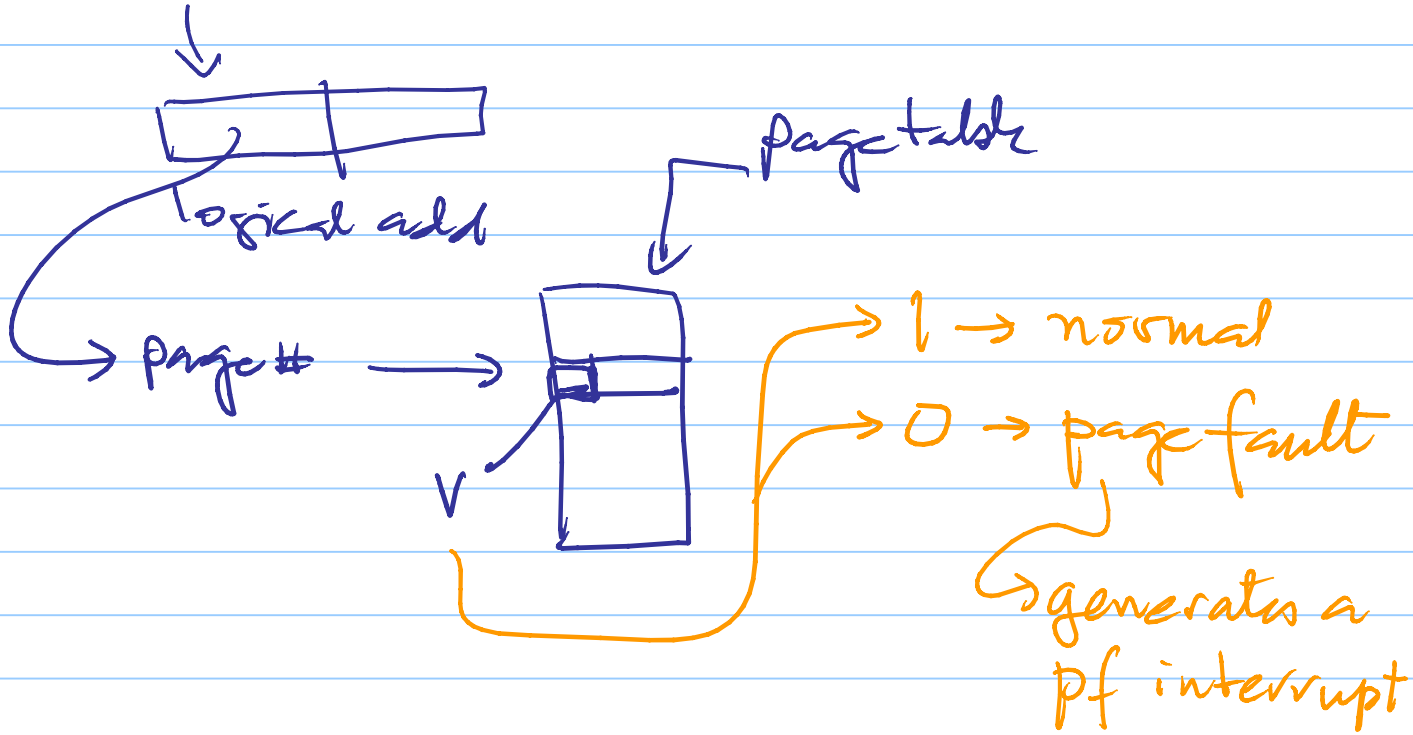
put LOD images on swap space

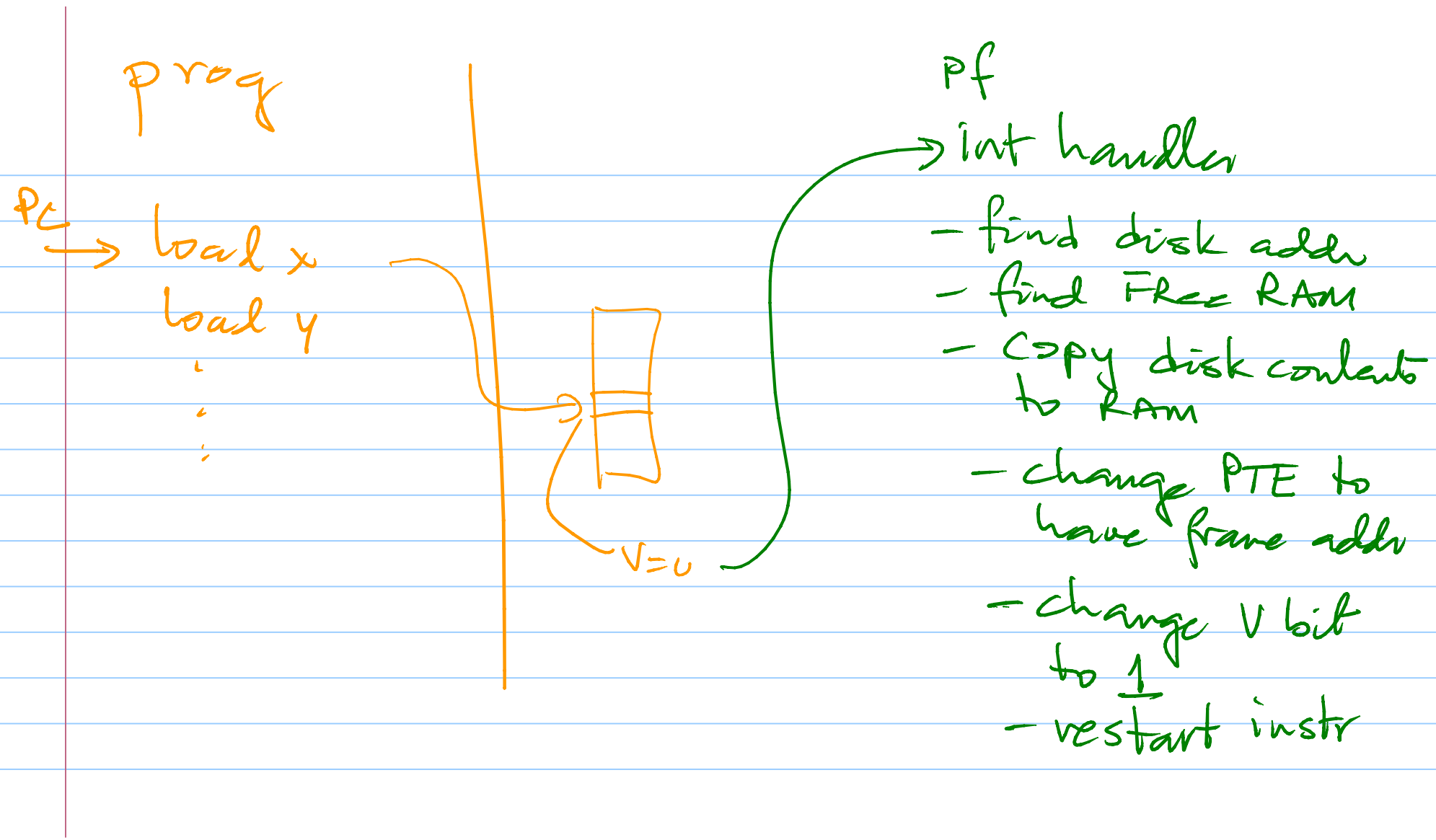


LOD pages

when a process executes

- fetch/load or STORE




prog
PC
load x
load y
V=0
pf
int handler
- find disk addr
- find FREE RAM
- copy disk contents to RAM
- change PTE to have frame addr
- change V bit to 1
- restart instr

Start a process — no pages

working set loaded — some pages

move pf — move pages

→ need to free up mem

↑ page replacement

page replacement policies → page out & page in

- FIFO → first page in, first out
- OPT → Best policy
- LRU → Least recently used

↳ choose victim to page out

FIFO

→ ① the old page may still be heavily used

② suffers from Belady's anomaly

③ Not a stack policy

page reference string

	1	2	3	4	1	2	5	1	2	3	4	5	
3 pages →	1	2	3	4	1	2	5	–	–	3	4		(9)
4 pages →	1	2	3	4	–	–	5	1	2	3	4	5	(10)
5 pages →	–	1	2	3	4	–	–	5	–	–	–	–	(5)

BELADY's Anomaly