Lecture 11: Memory management

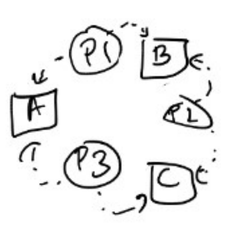
- simple base/limit MMU
- address translation
- paging / TLB
- swapping

Def: state is **safe** if $\exists$ an order in which processes can be run completely (assuming they acquire max allocation, then release all resources)

Bankers algorithm: maintain invariant that state is safe by blocking when resources are requested.

- when process requests resource, block until granting request would be safe.

	current A B C	max A B C
P1	1	1 1
P2	1	1 1
P3		1 1 1
Free	1 1	1 1 1



Can't run P1 fully
Can't run P3 fully
could do P2

⇒

	A B C
P1	1
~~P2~~	
P3	
F	1 1

then P1
then P3.

P3 requests 1 resource C.

if granted:

	A B C
P1	1
P2	1
P3	1
F	0 0 0

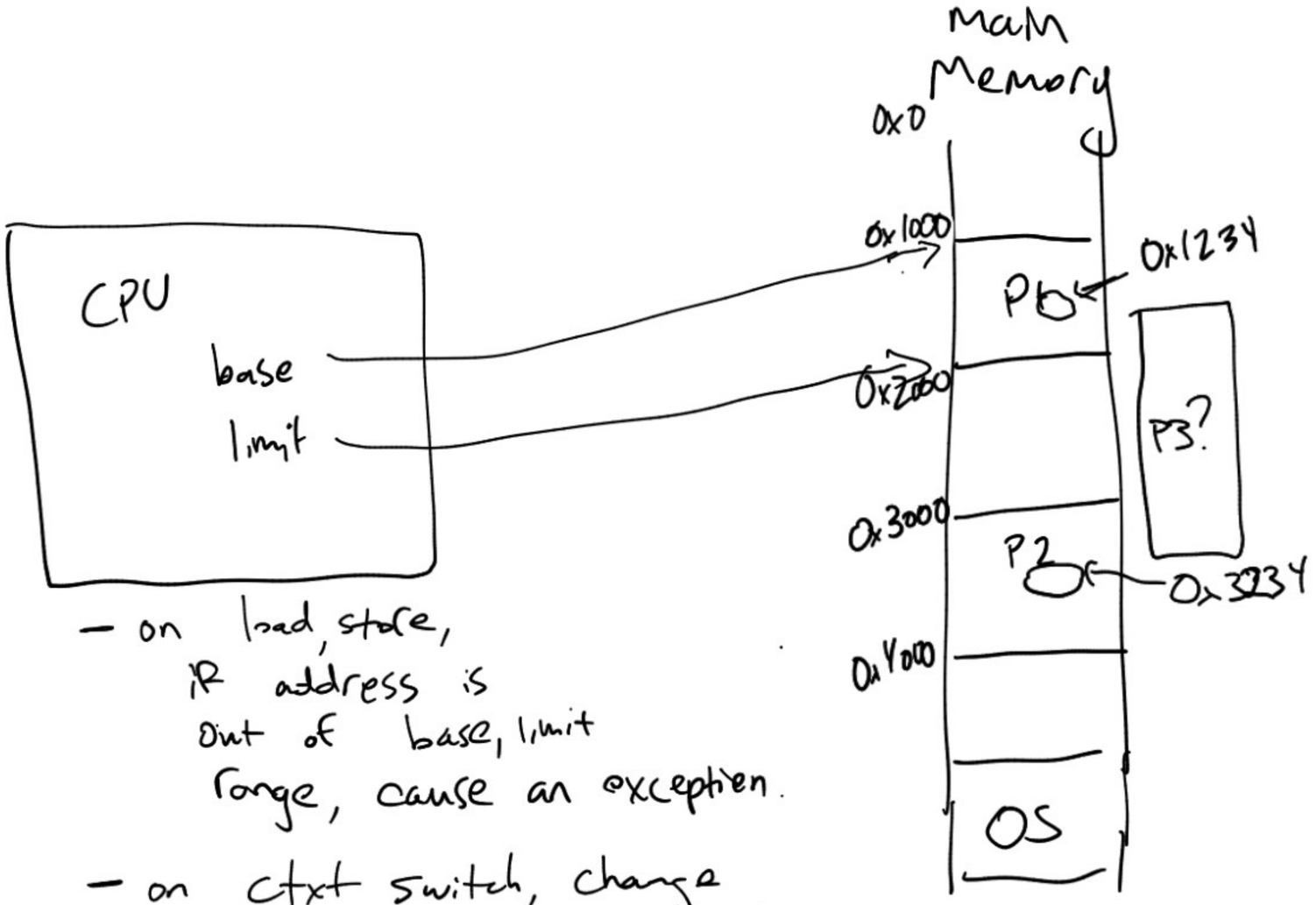
not safe
block.

P2 requests C

if granted:

	A B C
P1	1
P2	1 1
P3	
F	0 0 0

safe:
run P2, P1, P3



- on load, store, IR address is out of base, limit range, cause an exception.
- on ctxt switch, change base & limit to point to new proc.

Problems

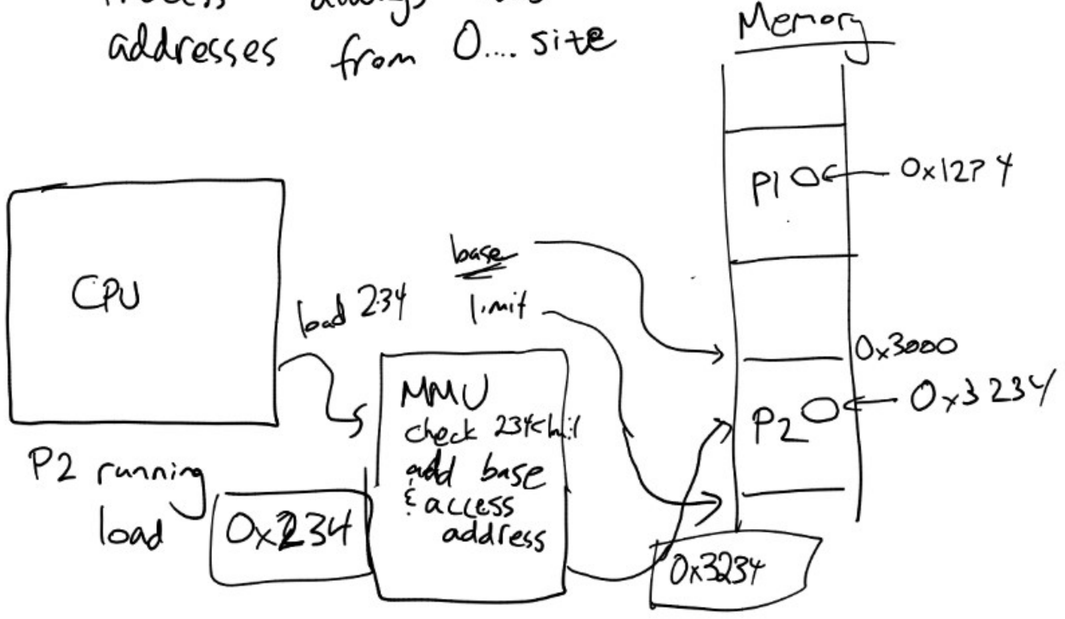
- Allocation: ensure processes don't overlap.
  - fragmentation: enough free memory to use, but not contiguous.
- Procs need to adapt to their location.

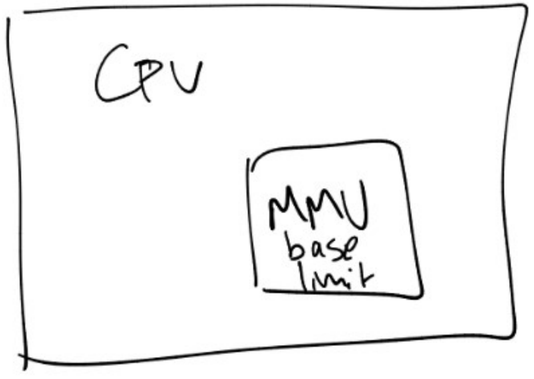
Not actually problems

- how do we know base & limit are?

Strawman: Address translation

Process always uses addresses from 0.... site

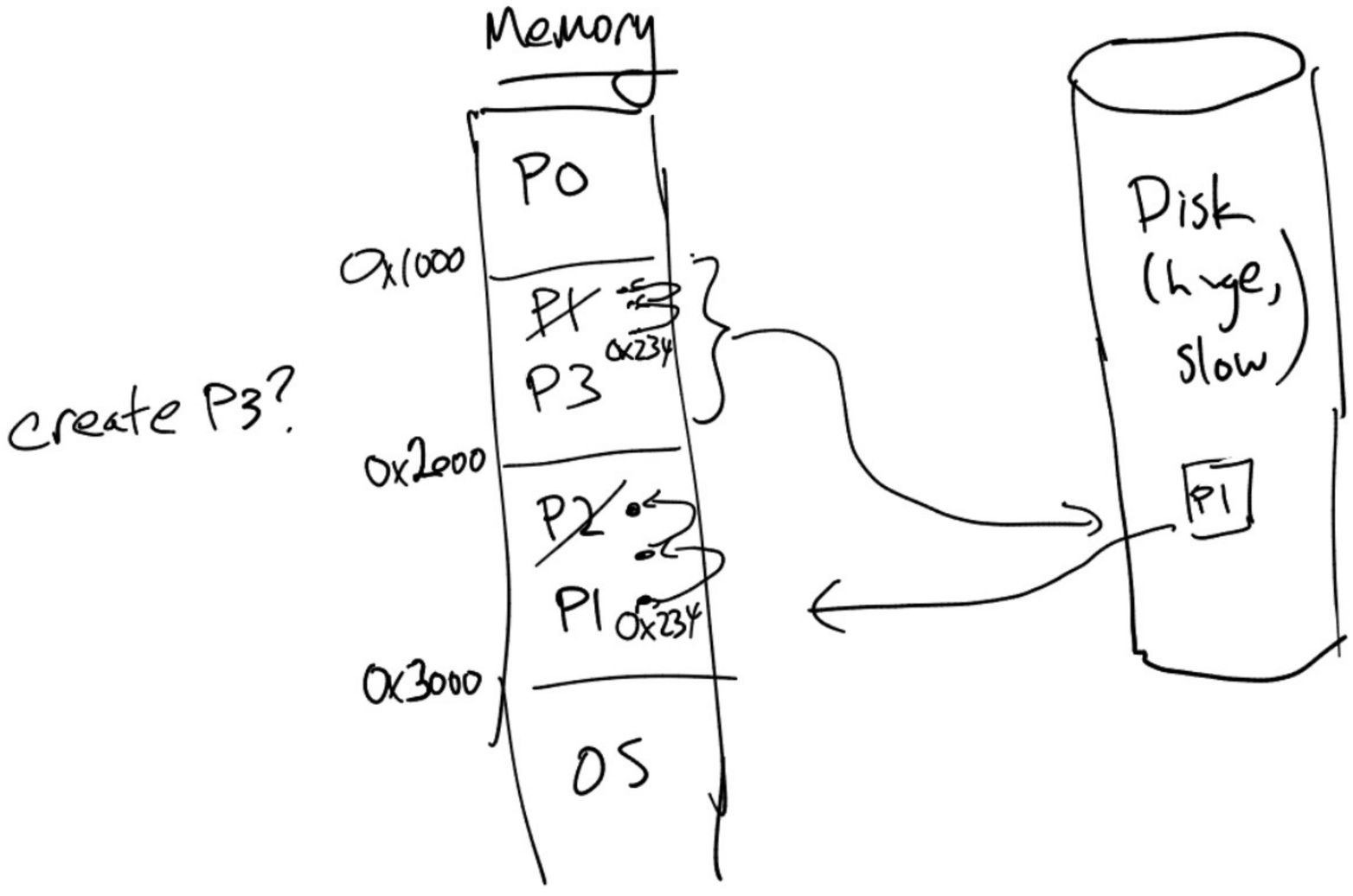




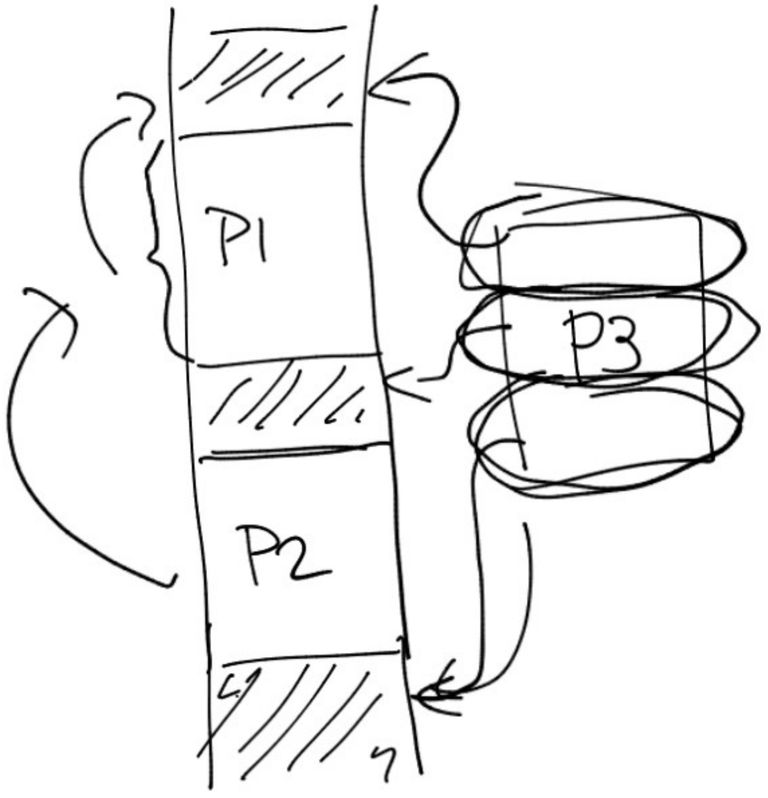
address translation

- processes work with "logical" addresses
  ex: 0x0234
- memory management unit (MMU) translates logical addresr to physical addr.
  e.x: 0x3234

Swapping


Memory
P0
0x1000
P1
P3
0x234
0x2000
P2
P1
0x234
0x3000
OS
Disk
(huge,
slow)
P1
create P3?

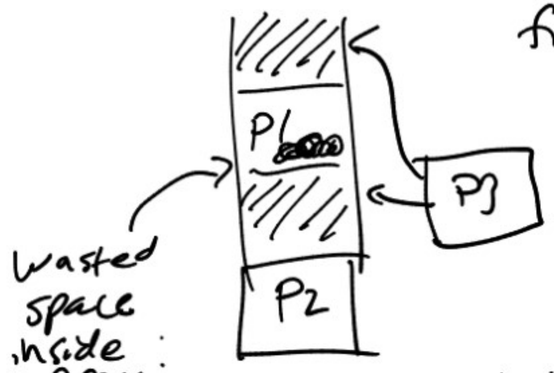
Fragmentation



external fragmentation:

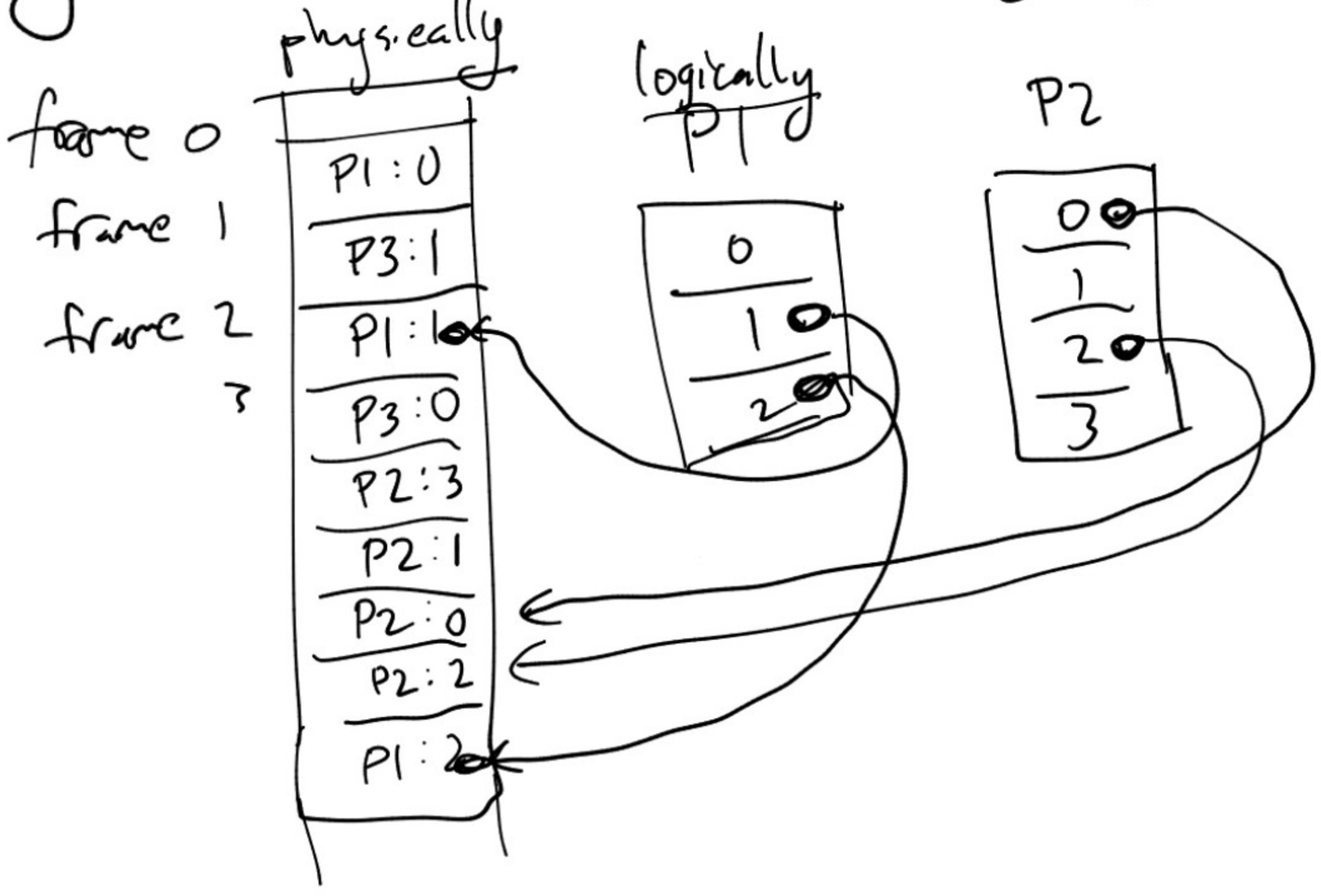
enough free space, but not contiguous.

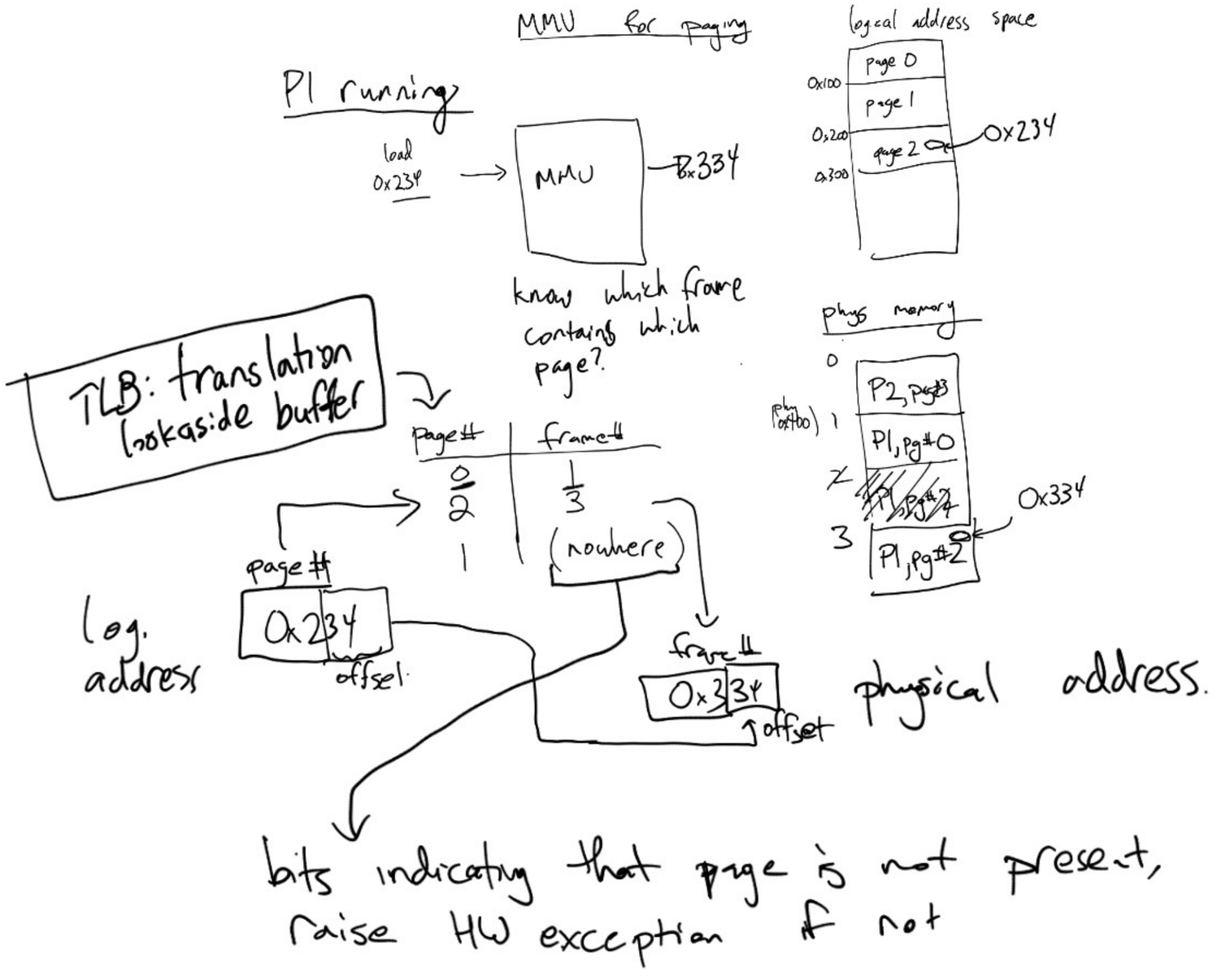
Could make all processes same size: No external frag.



internal fragmentation

Paging: split process's address space up into fixed-sized "pages", put pages into different "frames" of phys. memory




MMU for paging
P1 running
load 0x234
MMU
0x334
know which frame contains which page?
logical address space
page 0
page 1
page 2
0x100
0x200
0x300
0x234
phys memory
P2, pg#3
P1, pg#0
P1, pg#1
P1, pg#2
0x334
TLB: translation lookaside buffer
page#
frame#
0
2
1
3
(nowhere)
page#
0x234
offset
log. address
frame#
0x334
offset
physical address.
bits indicating that page is not present, raise HW exception if not

Segmentation

- processes tell OS that certain regions of log. address space are protected (ex: read only (constants) executable, not writable (code))
- OS crashes process if program misbehaves
- TLB has 3 permission bits for each entry (r, w, x), causes a HW exception on access to a page w/o permissions.

logical address space