Lecture II: Memory management

- Simple base/limit MMU
- Address translation
- Paging / TLB
- Swapping
Dead state is safe if an order in which processes can be run competely (assuming they acquire min 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.

P3 Requests resource C.
If granted:

P1 | P2 | P3 | 1 1 1
---|----|----|----
F   | 0 0 0

Not safe block.

P2 Requests C
If granted:

P1 | P2 | P3 | 1 1 1
---|----|----|----
F   | 0 0 0

Safe: run P2, P1, P3
CPU
  base
  limit

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

Problems

- Allocation: ensure processes don't overlap
  - fragmenting enough free memory to use, but not contiguous

- Procs need to adapt to their location

Now actually problems

- how do we know base & limit are?
**Address translation**

- **Processes work with logical addresses**
  - Ex: 0x0234

- **Memory Management Unit (MMU)**
  - Translates logical address to physical address
  - E.g.: 0x3234
Swapping

create P3?

Memory

OS

0x3000

0x2000

0x1000

P0

P1

P2

P3

Disk (huge, slow)

P1

P1

0x3000
Fragmentation

external fragmentation:
- enough free space, but not contiguous.
- Could make all processes same size: no external fragmentation.

internal fragmentation:
- wasted space inside process.
Paging: split process's address space up into fixed-sized "pages", put pages into different "frames" of physical memory.
TLB: translation lookaside buffer

log. address

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.