- [x] 1. Last time
- [x] 2. Page faults: costs
- [x] 3. Page replacement policies
- [x] 4. Thrashing
- [ ] 5. mmap()

---

2. Page faults: costs

look at AMAT (avg. memory access time)

AMAT = (1-p) * (mem access time) + p * (page fault time)

t_M t_D

p is probability (or frequency of a page fault)

mem access time ~ 100ns t_M

disk access time ~ 10ms = 10^7 ns t_D

QUESTION: what is p such that s.t. paging hurts performance by less than 10%?

$1.1 \cdot t_M \geq (1-p)\, t_M + p \cdot t_D$

$$1.1 \cdot t_M \geq t_M - p \cdot t_M + p \cdot t_D$$

$$.1 t_M \geq p(t_D - t_M)$$

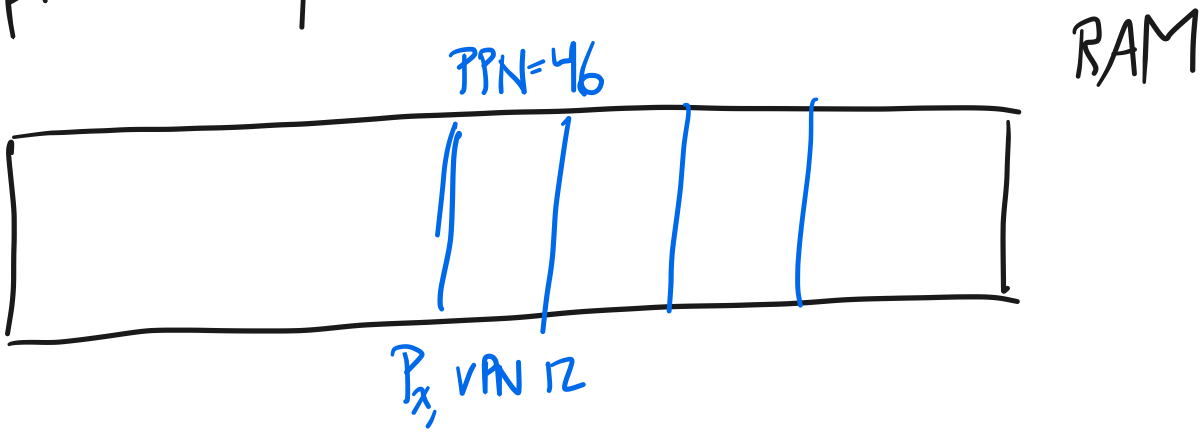
$$p \leq \frac{t_M}{10 \cdot (t_D - t_M)} \approx 10^{-1} \cdot \frac{t_M}{t_D} =$$

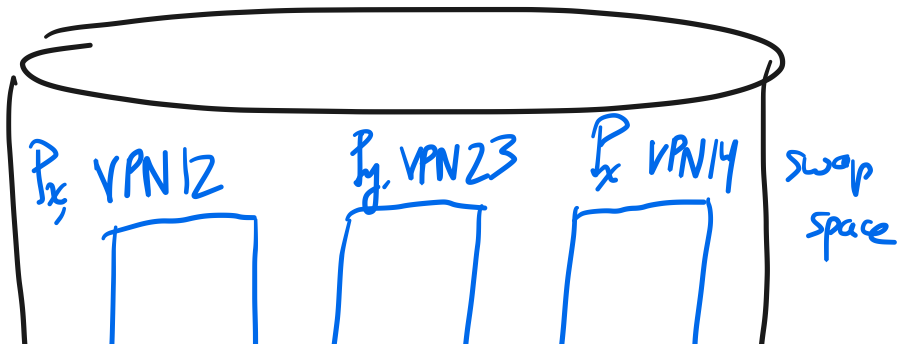
$$= \frac{t_M}{10 \cdot t_D} = \frac{100\,ns}{10 \cdot 10^7\,ns}$$

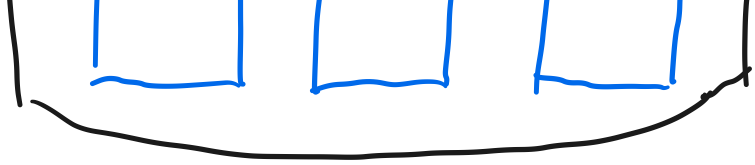
$$= \frac{100ns}{10^8\,ns} = 10^{-6}$$

---

3. Page replacement policies







- FIFO: eject oldest
- MIN (OPT): eject entry that won't be referenced for the longest time

input:
reference string
cache size

output:
number of evictions, or more generally misses

FIFO

A B C A B D A D B C B

phys_slot

S1 A h D h C

S2 B h A B h

S3 C

7 misses/swaps/evictions, 4 hits

OPTIMAL

	A	B	C	A	B	D	A	D	B	C	B
phys_slot											
S1	A			h		h			h	C	h
S2		B			h						
S3			C			D		h			

5 misses/swaps/evictions, 6 hits

LRU

	A	B	C	A	B	D	A	D	B	C	B
phys_slot											
S1	A			h		h			h	C	h
S2		B			h						
S3			C			D		h			

5 misses/s/e, 6 hits

LRU

	A	B	C	D	A	B	C	D	A	B	C	D
phys_slot												
S1	A				h				h			
S2		B				h				B		
S3			C				h				C	
S4				D				h				D

4 misses/s/e, 8 hits

S1 A D C D C

S2 B A D C

S3 C B A D

12 misses 0 hits

---

back to FIFO

A B C D A B E A B C D E

S1 A D E h C

S2 B A h D

S3 C B h

3 hits

A B C D A B E A B C D E

S1 A h E D

S2 B h A E

S3 C B

S4 D C

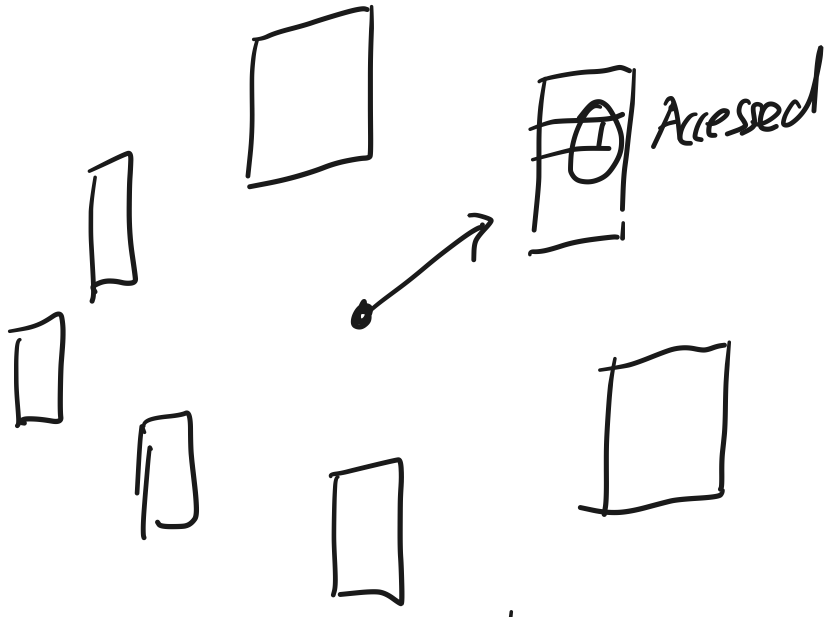
2 hits WORSE

BELADY'S ANOMALY

- OPT minimizes misses/swaps/evictions
  - but can't be implemented in general.

- LRU: approximates OPT (assuming what?)

- approximate LRU with CLOCK



H/W sets Accessed + Dirty bits

OS consumes these bits and clears them.

- Generalization of CLOCK: N^{th} Chance (see notes).

4. Thrashing

ex: program touches 50 pages, equiprobably
but only 40 phys. frames (or slots)

Thrashing: processes demand more memory for active use than the system has.

3 reasons:
(a) process has no temporal locality, or
(b) " " temporal locality but not enough memory; or
(c) individually all processes fit, but there's not enough memory.