- [x] 1. Last time
- [x] 2. Midterm logistics
- [x] 3. Class overview
- [x] 4. Your questions

---

2. Logistics
   - 75-minute exam
     - closed book
     - turn in your exam at minute x, $x < 65 \vee 75 \leq x < 78$
   - One two-sided sheet of notes, with the specifications on the class web page

3. Class overview (not comprehensive, not guaranteed to be necessary or sufficient for exam)

   Material
   - readings
   - labs
   - HWs
   - lectures/classes

- Lectures/classes

Lectures/classes:

- OSes: what are they?
  - goals, purpose
- Processes:
  - process's view of memory, registers
  - stack frames
  - OS's view of processes
- System calls
- Process/OS control transfers
- Process birth: fork()/exec()
- Shell
- File descriptors
- Redirection, pipelines
- threads
- concurrency
  - hard to deal with! abstractions help (but not fully)

- critical sections
- mutexes
- spinlocks
- condition variables
- monitors

- lots of things can go wrong: safety problems, liveness problems, etc.
  - lack of sequential consistency makes the problem worse
- safety: build primitives that get help from H/W (xchg)
- liveness: various problems, including deadlock
- tradeoffs
  - for example, performance vs. complexity
- "advice"

- software safety (Therac-25)
- scheduling
  - when scheduling happens, which metrics, what costs

- specific disciplines
- lessons + conclusions

- virtual memory
    - paging
    - virtual memory on the x86-64

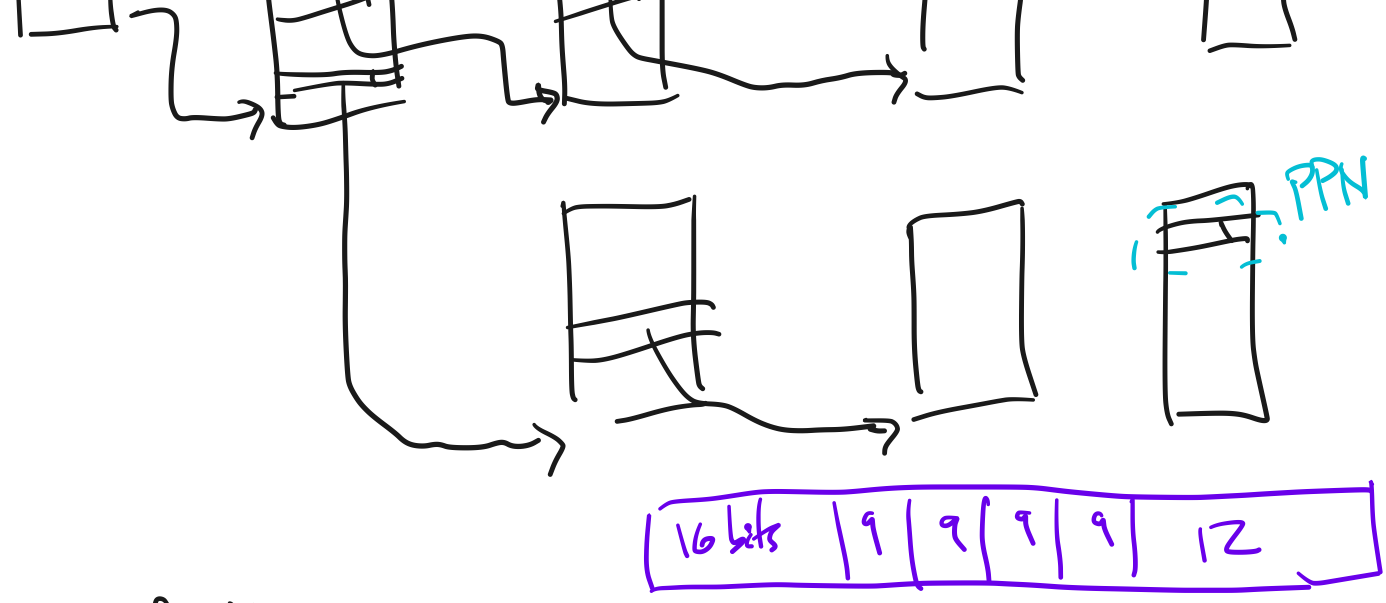
      VA: [16 bits | 36 bits | 12 bits]
      - 16 bits: unused (0 - - - - 0 or 1 - - - - 1)
      - 36 bits: VPN
      - 12 bits: offset

    - entry in L1...L4 page tables:

      [... 40 bits ... 3 bits]
      - 40 bits: PPN
      - 3 bits: user/kernel | W/R | present/not

      protection → user/kernel | W/R | present/not

    - TLB: <VPN, (PPN, perms)>



- Page faults
  - mechanics
  - costs
  - uses

- Page replacement policies
  FIFO, LRU, CLOCK, OPT

---

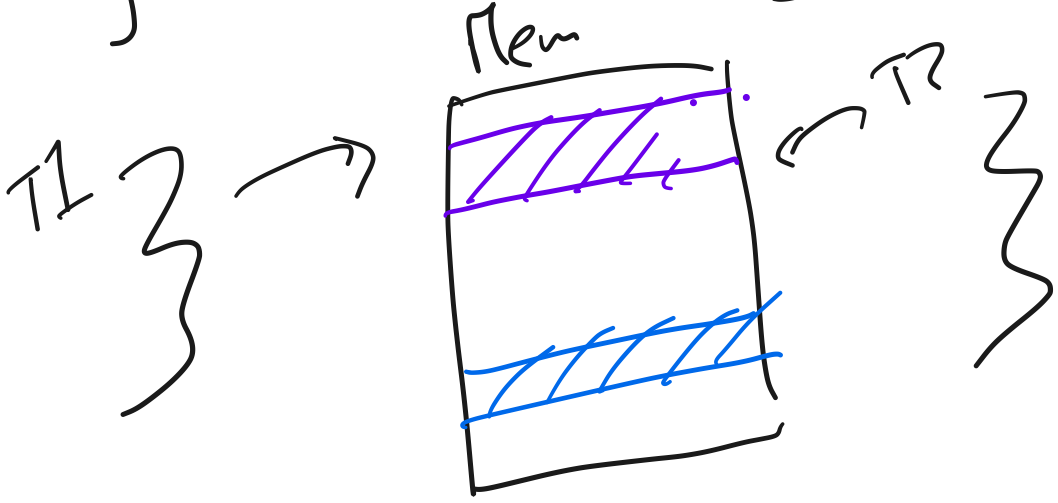
4. Your questions

Concurrency:

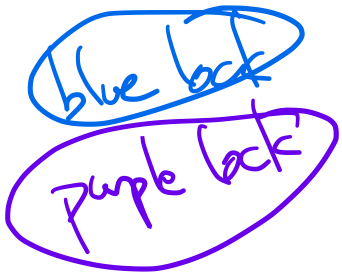
... CVs?

- How many

Fine-grained vs. Coarse-grained locking



Coarse: 1 lock



Weighting

1, 1, 1

CFS

3, 1, 1

- # of Virt. Pages

Calc #of virt. pages

alloc 12KB

main ( )

{

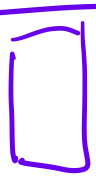
mutex . destroy

thread_exit exit

}

event-driven programming

TLB misses, pg faults



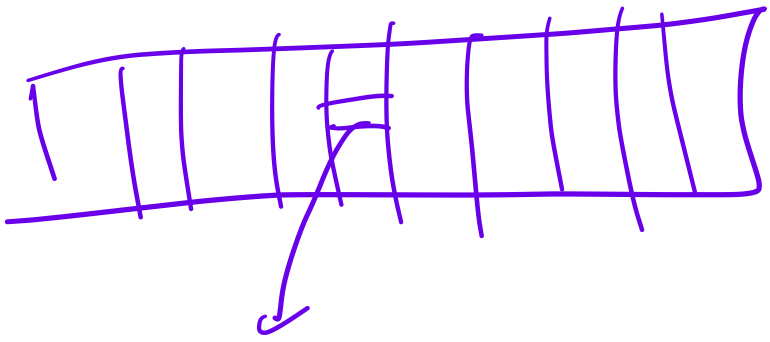
1 TLB miss
↳ 0x500 movq 0x200000, %rax → 2 TLB misses, 1 pg fault

0x5040 incq %rax, 1

0x5080 movq %rax, 0x300000 → 2 TLB misses, 1 pg fault

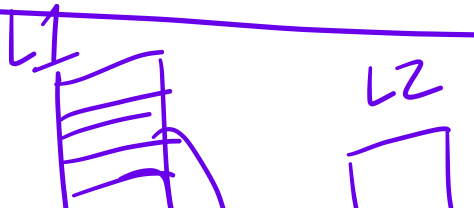
0x200,000
0x200,fff
} same page

IDT

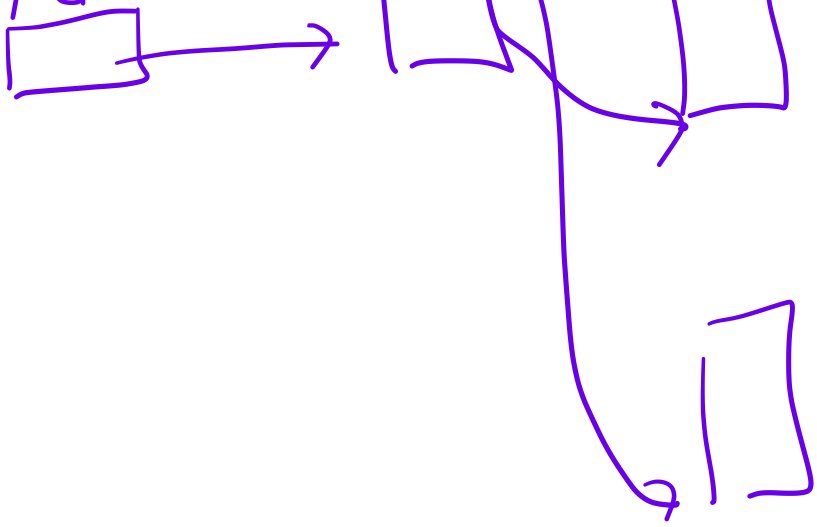


boot.S

.S



i.c3



Monitor

```
smutex_lock(m);
m.acquire();
```

acquire

```
m.P();
```