- [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
- 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
  - intro
  - paging

---

condition variables

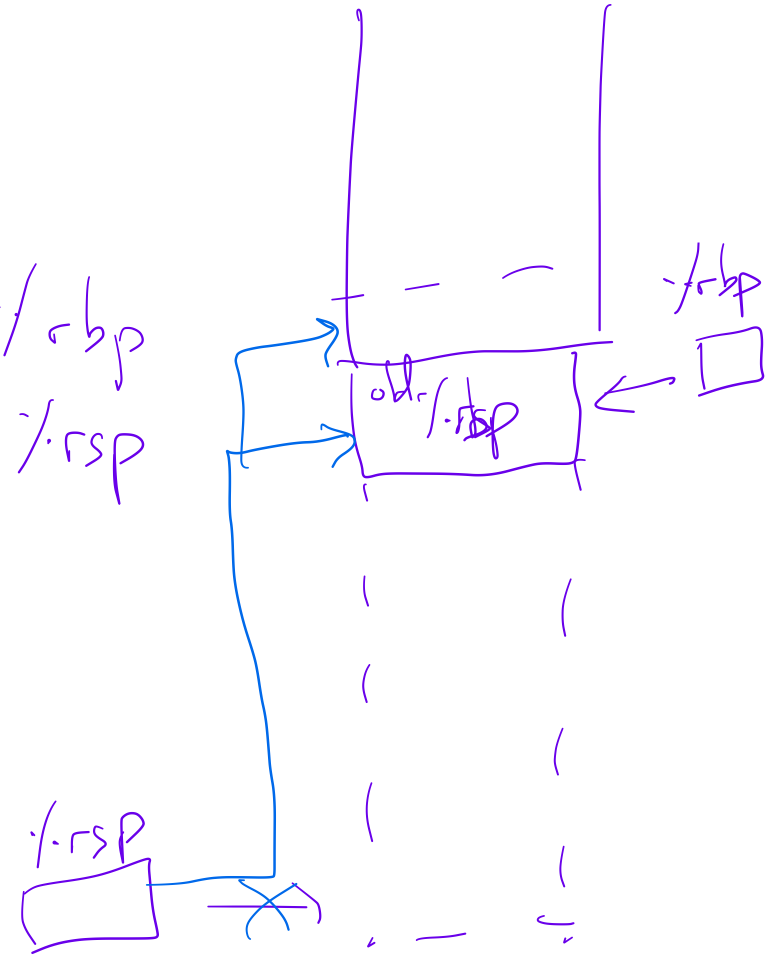
```
wait(&cv, &mutex);
signal(&cv, &m);
bcast(&cv, &m);
```

acquire(

```
a = 1;
b = 1;
```

	pass	curr	
P1 (Mike)	10	0	20
P2 (Sam)	5	0	5
P3 (Zach)	1	0	15
P4			

```
f:
proby:  pushq %rbp
        movq %rsp, %rbp
        subq $0x20, %rsp
```



movq %rbp, %rsp

epilog { popq %rbp
retq

T1

acq(&mA);
acq(&mB);

T2:

acq(&mB);
acq(&mA);

- Og