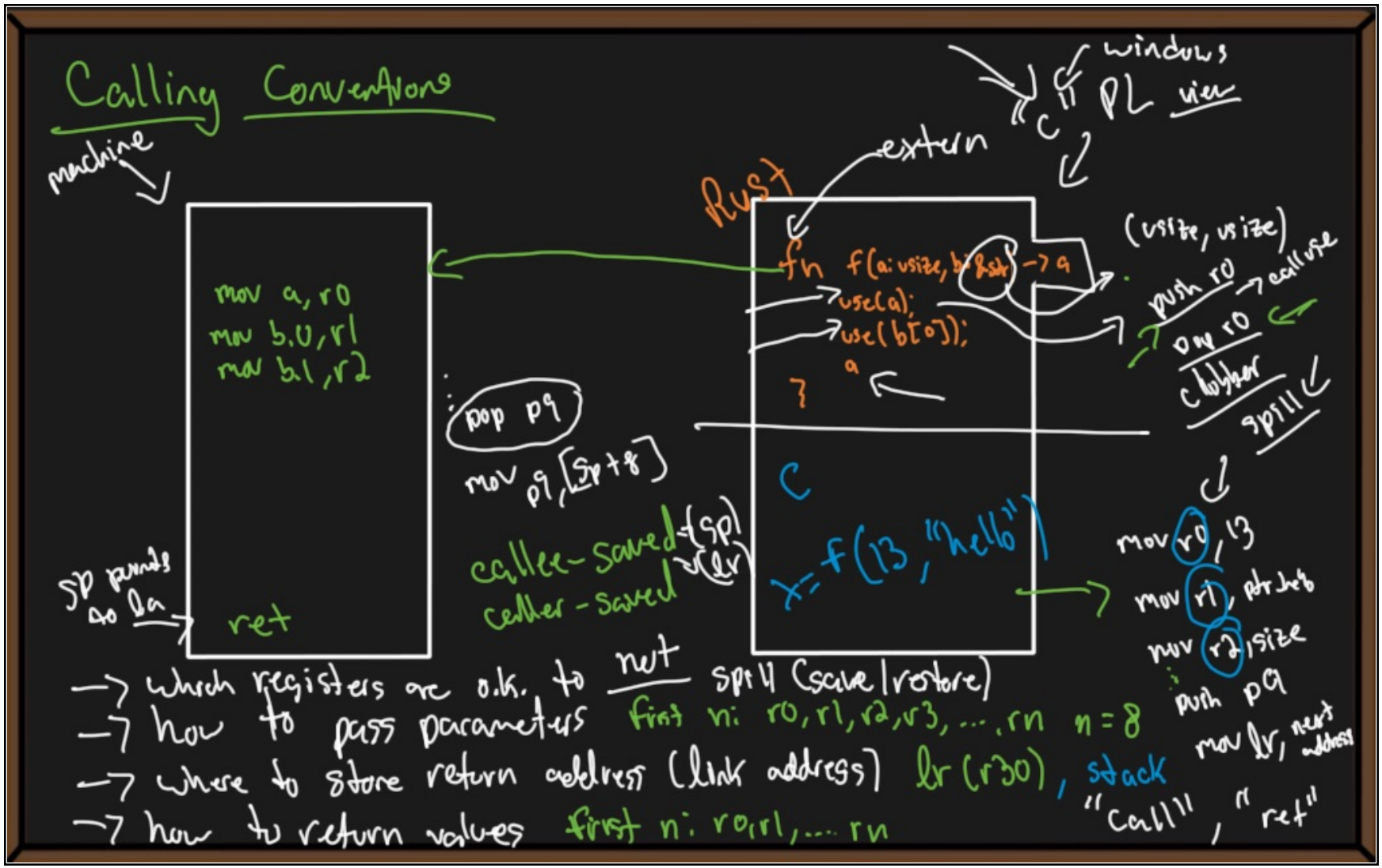
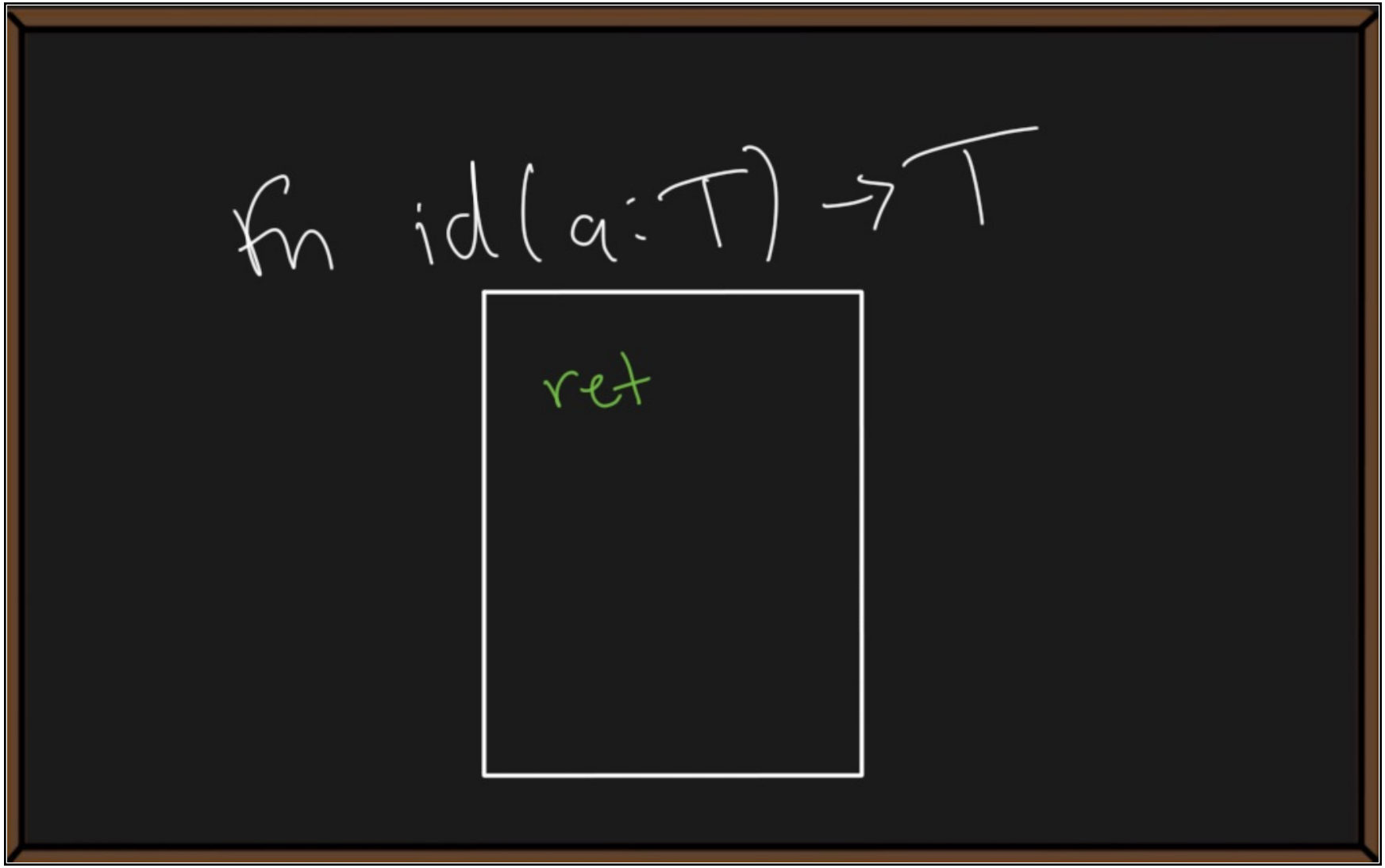
Today

- IPC (inter-process communication)
  - named pipes
  - shared memory
  - signals
- Scheduling
  - 0) First-Come, First Serve
  - 1) Round-Robin
  - 2) Priority + Decay
- System Calls
  - calling conventions
  - software interrupts
  - exception / privilege levels
- The first process
  - bootstrapping + more processes
- Concurrency
  - threads
  - multiprocessor / multicore


Calling Conventions
machine
mov a, r0
mov b.0, r1
mov b.1, r2
pop pc
mov pc, [sp+8]
SP points to ra
ret
Rust
extern
C
windows
PL view
fn f(a: usize, b: &str) -> q
use(a);
use(b[0]);
q
(usize, usize)
push r0
call use
pop r0
C wrapper
spill
C
x = f(13, "hello")
callee-saved
caller-saved
mov r0, 13
mov r1, ptr to hello
mov r2, size
push pc
mov lr, next address
"call!!", "ret"
→ which registers are o.k. to not spill (save/restore)
→ how to pass parameters first n: r0, r1, r2, r3, ... rn n = 8
→ where to store return address (link address) lr (r30), stack
→ how to return values first n: r0, r1, ... rn


fn id(a:T) -> T
ret

System Calls

A request from user-level for the Kernel. (typically numbered)
- typically to perform some priveleged operation on behalf of the user

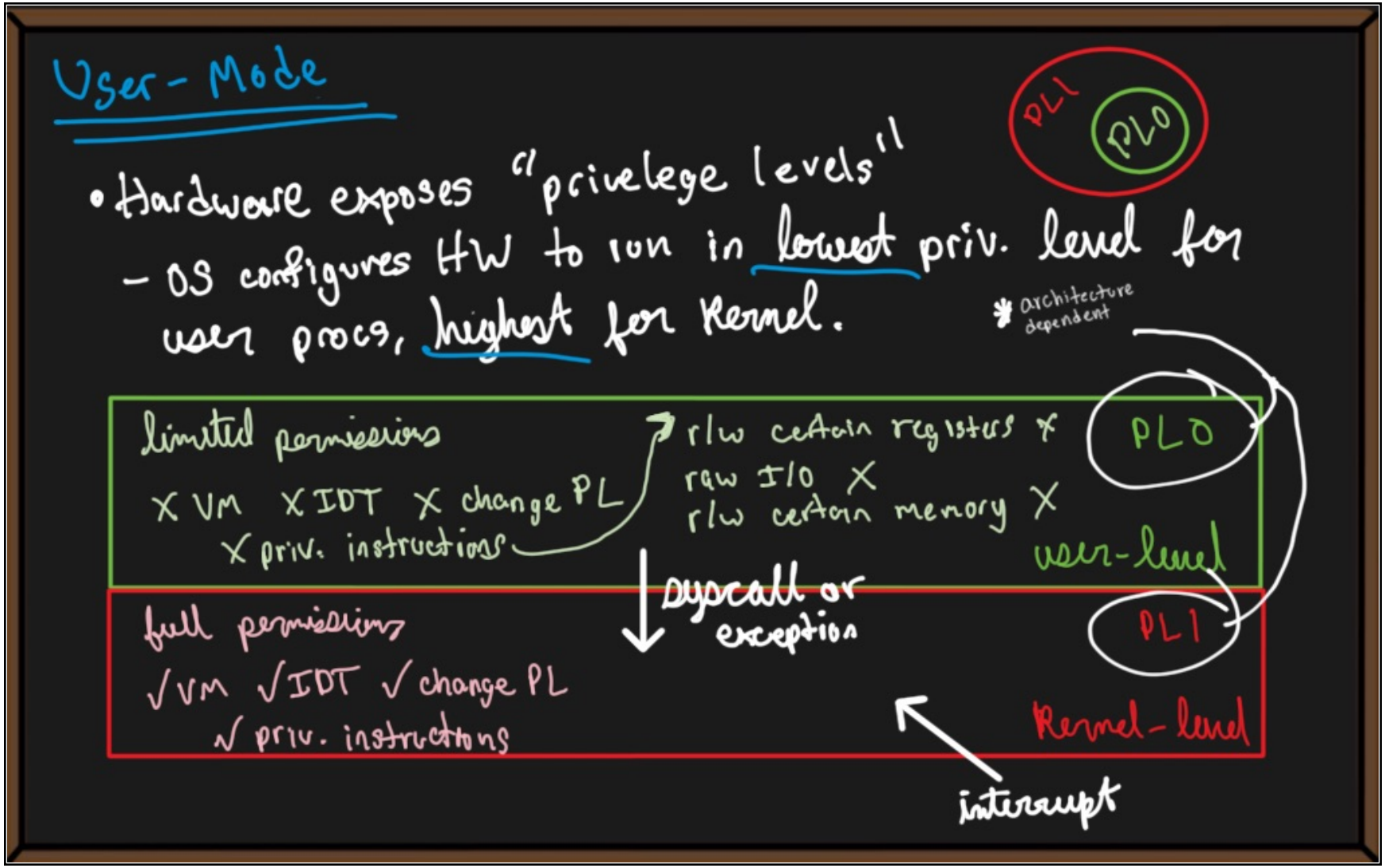
Example: read/write from disk
: allocate memory
: signal a process
: create memory mapping
: create a new process

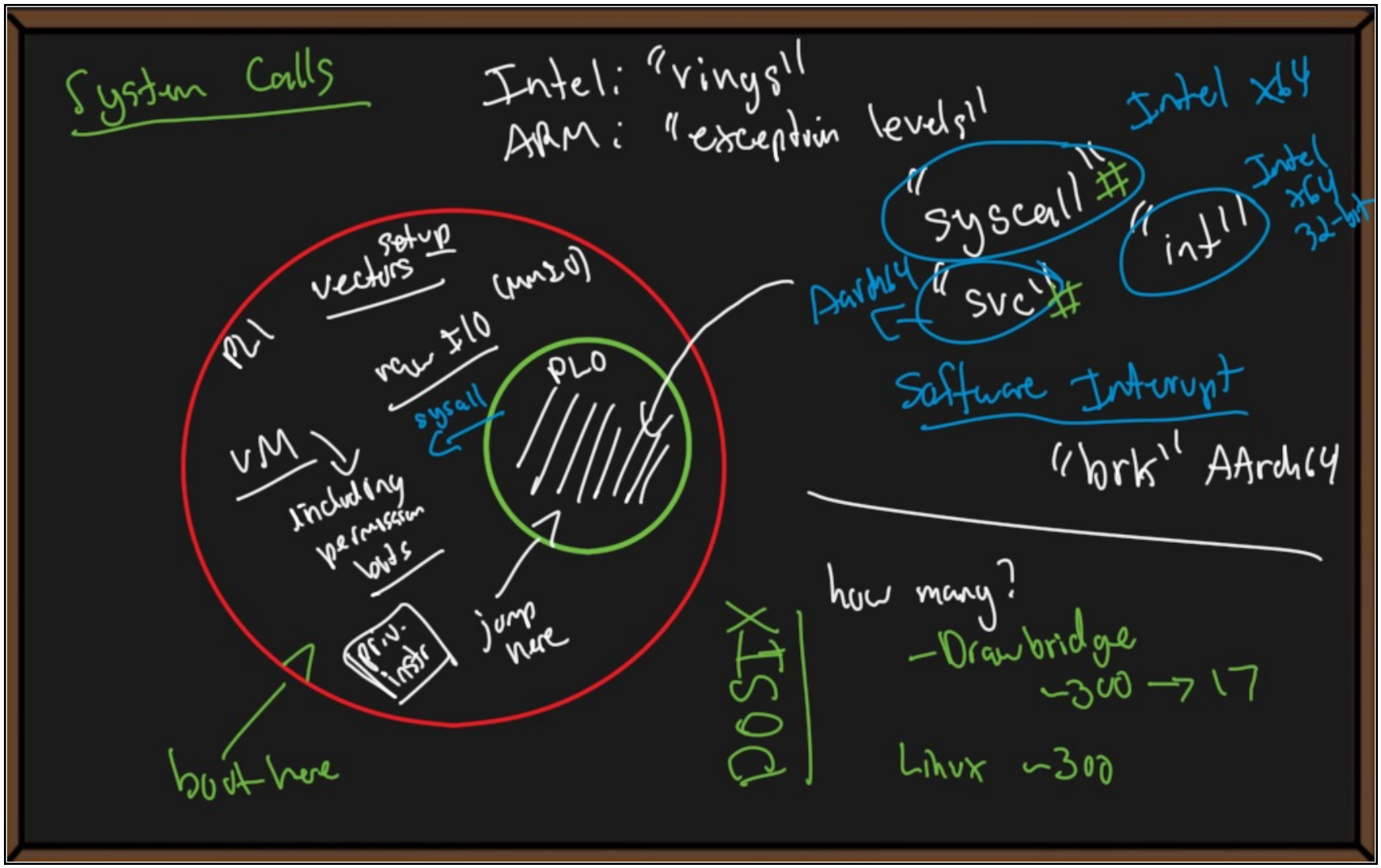
+ 100s more (>300 on Linux)

A protected transfer from fewer to greater priveleges.

Q: Where have we heard this before?

A: Interrupts + Exceptions!


User-Mode
• Hardware exposes "privelege levels"
– OS configures HW to run in lowest priv. level for user procs, highest for Kernel.
* architecture dependent
PL1
PL0
limited permissions
X VM X IDT X change PL
X priv. instructions
r/w certain registers X
raw I/O X
r/w certain memory X
PL0
user-level
syscall or exception
full permissions
√VM √IDT √change PL
√ priv. instructions
PL1
Kernel-level
interrupt


System Calls
Intel: "rings"
ARM: "exception levels"
Intel x64
"syscall" #
"int"
Intel x64 32-bit
Aarch64
"svc" #
Software Interrupt
"brk" AArch64
PL1
vectors
setup
(arm32)
new EL0
PL0
sysall
VM
including permission bits
priv. instr
jump here
bust here
how many?
-Drawbridge ~300 → 17
POSIX
Linux ~300