- MOTIVATION: KEYBOARD READING

There are many details we have to worry about:

- STATUS REqs ASYNCHRONICITY
- DATA REqs of KB
- DATA FORMAT
- DATA PROTOCOL
  
- ANNOYING! PLUS, DIFFERENT DEVICES = DIFFERENT PROTOCOLS, ETC.

- EVEN WORSE, WHAT IF WE BORK THE KB? OTHER DEVICES CAN'T USE IT

- WE NEED TO INTRODUCE PRIVILEGE

  ONLY "TRUSTED" CODE CAN INTERACT WITH THE DEVICE DIRECTLY.

- WE ACCOMPLISH THIS WITH TRAP INSTRUCTIONS.
How does it work? NEED 4 THINGS

1. OS has a set of "service routines". THIS IS JUST CODE THAT LIVES SOMEWHERE ELSE IN MEMORY, (ALSO CALLED "SYSTEM CALLS"

2. A TABLE OF POINTERS TO SERVICE ROUTINES (TRAP VECTOR TABLE, SYSTEM CALL TABLE, ETC.)

3. TRAP INSTRUCTION (on x86, "SYSCALL")

4. "LINKAGE" BACK TO "USER" PROGRAM. (MECHANISM TO RETURN TO PC OF OUR CODE AFTER TRAP COMPLETES.

WHEN TRAP INSTR. EXECUTES IT:
- CHANGES PC TO ADDR OF SERVICE ROUTINE
- SETS UP LINKAGE

\[ 1111 | 0000 \]

TRAP \hspace{8} TRAP VECTOR
**FULL MECHANISM:**

**Trap Table** lives @ 0x0000 - 0x0FFF

1. Trap vector ZEXTed to 16 bits, loaded into MAR.
2. Contents of MEM @ vector loaded into MDR.
3. R7 is loaded with (INCD) PC.
4. PC < MDR.

**Diagram:**

![Diagram of memory and register interactions]

ABS. JUMP is raise req!

ALSO CALLED "ZET"

[CODE EXAMPLE]
- Similar Problem, but don't need privilege.
  - We're just trying to not repeat ourselves. General principle of SFAE Eng. (DRY).

- JSR (Jump subroutine)

  1010

  0100 11

  JSR A  11-bit offset

  0100 000101 00000000

  JSRR A ← BASE

- We need a "calling convention."
  * Which regs can a subroutine use?
  * Which can't it?
  * Where do we put args?
  * Who saves what? (caller or callee?)

- Part of an "ABI" decided by arch. creators. Compilers must follow this.