

~~But this is not the case~~

- MOTIVATION: KEYBOARD READING

There are many details we have to worry about:

- STATUS REQS
 - DATA REQS
 - DATA FORMAT
 - DATA PROTOCOL
 - ... ETC.
- asynchronicity of KB INPUT!

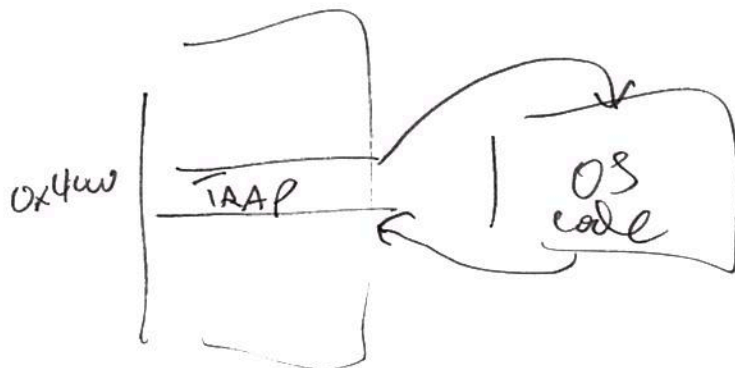
- ANNOYING! PLUS, DIFFERENT DEVICES = DIFFERENT PROTOCOLS ETC.

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

- WE NEED TO INTRODUCE PRIVILEGE

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

- WE ACCOMPLISH THIS WITH TRAP INSTRUCTIONS.

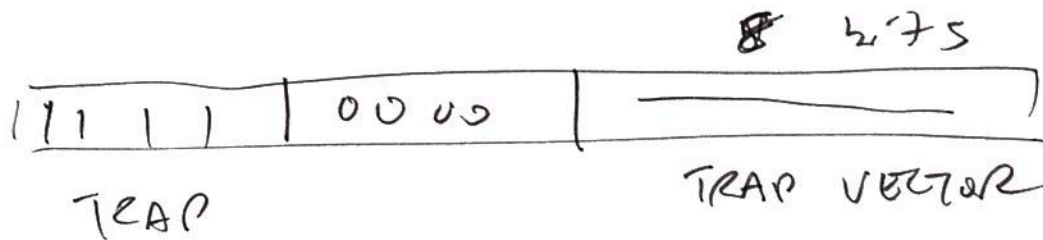


How does it work? NEED 4 THINGS

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

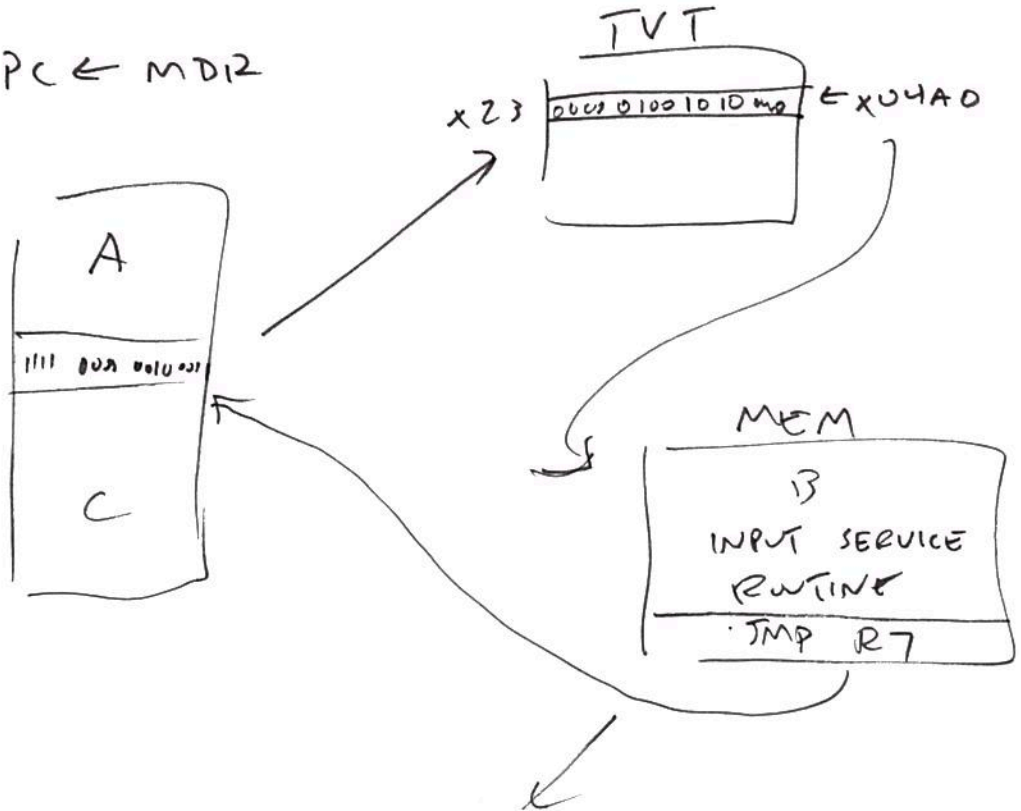


Full Mechanism:

TRAP TABLE LIVES @ 0x0 - 0xFF

IN EXEC. PHASE,

- 1) TRAP VECTOR EXTENDED TO 16 bits, loaded INTO MAR.
- 2) CONTENTS OF MEM @ VECTOR LOADED INTO MDR.
- 3) R7 IS LOADED WITH (INDEX) PC.
- 4) $PC \leftarrow MDR$



ABS. JUMP w BASE REG!

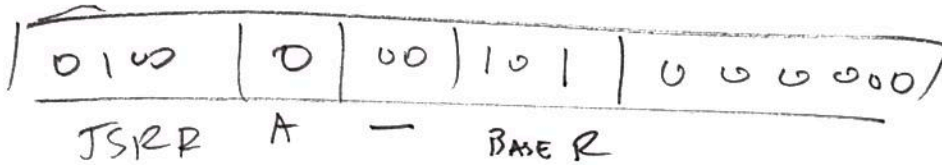
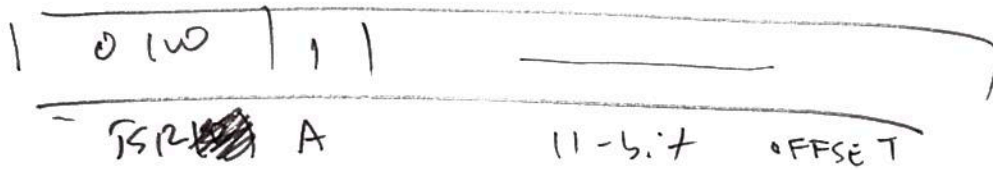
ALSO CALLED "RET"

[CODE EXAMPLE]

SUBROUTINES



- SIMILAR PROBLEM, BUT DON'T NEED PRIVILEGE
- WE'RE JUST TRYING TO NOT REPEAT OURSELVES
↓
GENERAL PRINCIPLE OF SOFTWARE ENGR. (DRY)
- ~~WE NEED~~ JSR (JUMP SUBROUTINE)



- WE NEED A "CALLING CONVENTION"
 - WHICH REGS CAN A SUBROUTINE USE?
 - WHICH CANT IT?
 - WHERE DO WE PUT ARGS?
 - WHO SAVES WHAT? (CALLER OR CALLEE?)
- PART OF AN "ABI" DECIDED BY ARCH. CREATORS. COMPILERS MUST FOLLOW THIS.