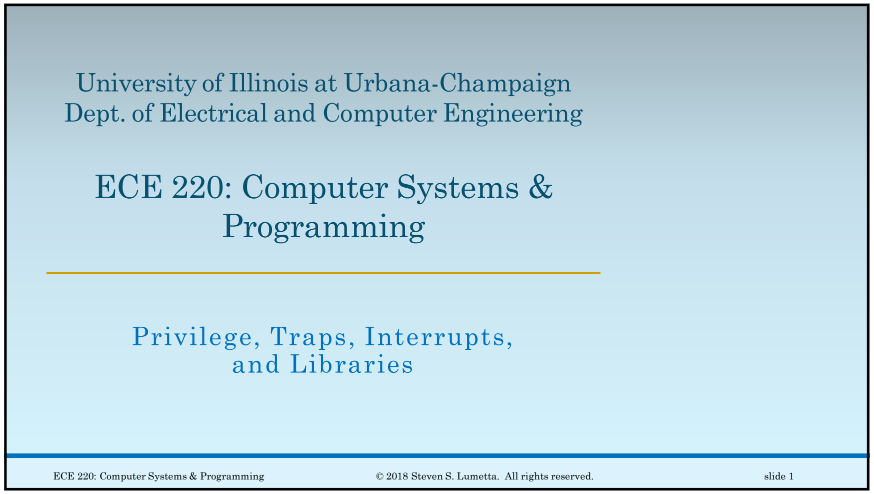
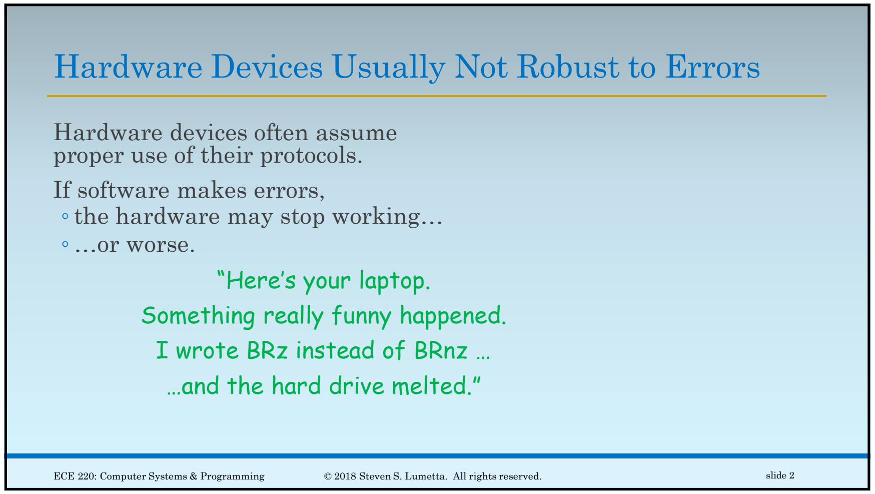


University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 220: Computer Systems & Programming

Privilege, Traps, Interrupts, and Libraries

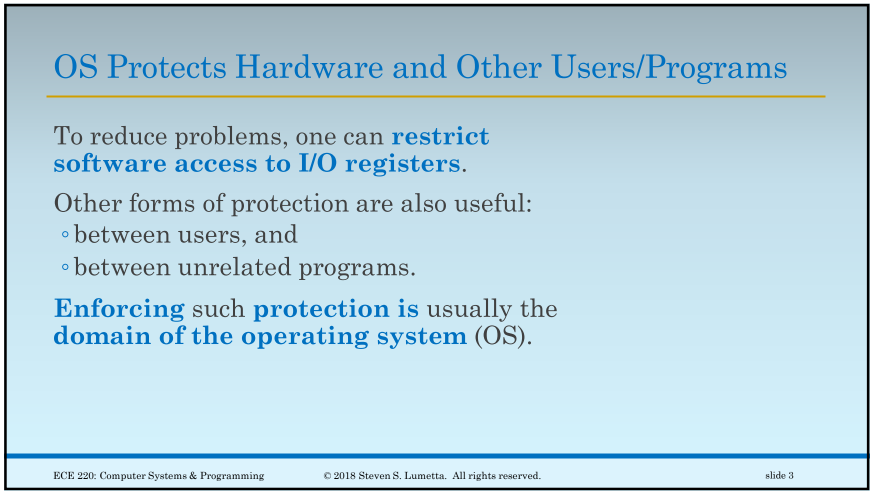


Hardware Devices Usually Not Robust to Errors

Hardware devices often assume proper use of their protocols.

If software makes errors,
- the hardware may stop working…
- …or worse.

"Here's your laptop.
Something really funny happened.
I wrote BRz instead of BRnz …
…and the hard drive melted."

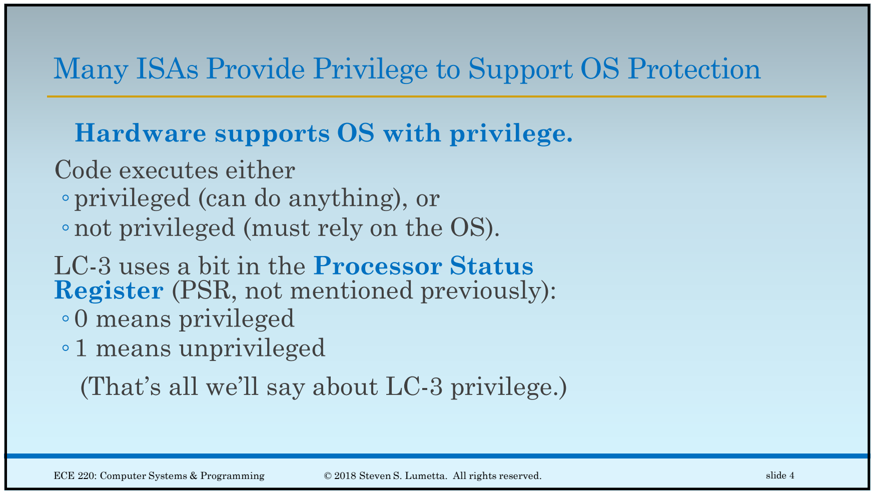


OS Protects Hardware and Other Users/Programs

To reduce problems, one can **restrict software access to I/O registers**.

Other forms of protection are also useful:
- between users, and
- between unrelated programs.

Enforcing such **protection is** usually the **domain of the operating system** (OS).



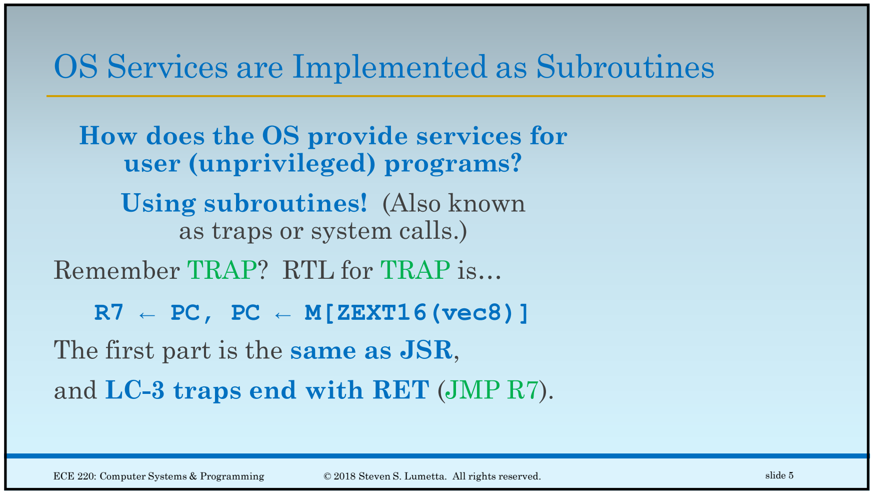
Many ISAs Provide Privilege to Support OS Protection

Hardware supports OS with privilege.

Code executes either
- privileged (can do anything), or
- not privileged (must rely on the OS).

LC-3 uses a bit in the **Processor Status Register** (PSR, not mentioned previously):
- 0 means privileged
- 1 means unprivileged

(That's all we'll say about LC-3 privilege.)



OS Services are Implemented as Subroutines

How does the OS provide services for user (unprivileged) programs?

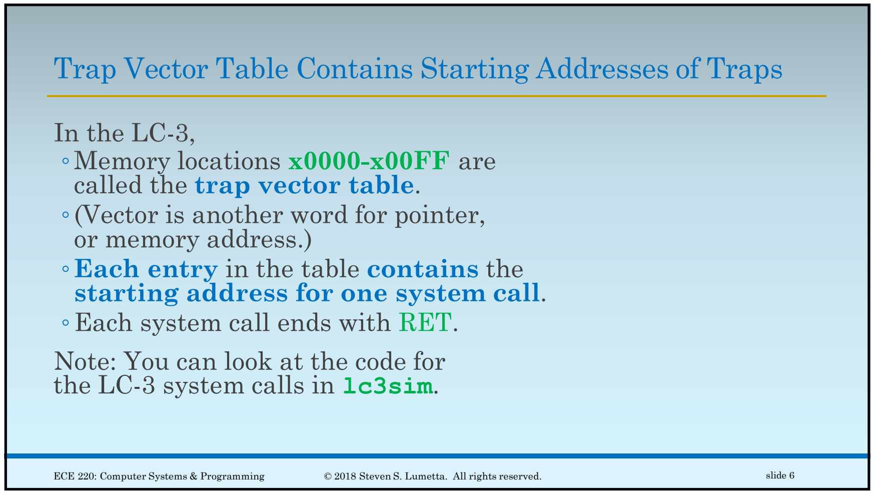
Using subroutines! (Also known as traps or system calls.)

Remember TRAP? RTL for TRAP is…

R7 ← PC, PC ← M[ZEXT16(vec8)]

The first part is the **same as JSR**,

and **LC-3 traps end with RET** (JMP R7).

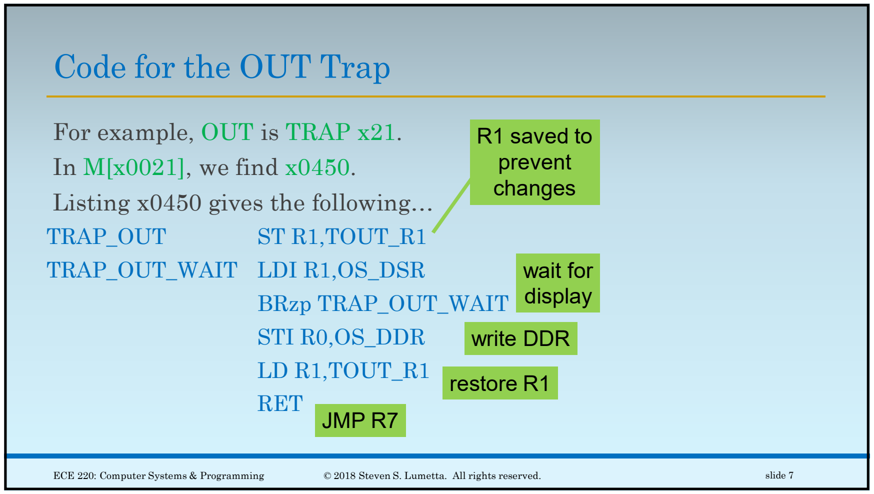


Trap Vector Table Contains Starting Addresses of Traps

In the LC-3,

- Memory locations **x0000-x00FF** are called the **trap vector table**.
- (Vector is another word for pointer, or memory address.)
- **Each entry** in the table **contains** the **starting address for one system call**.
- Each system call ends with RET.

Note: You can look at the code for the LC-3 system calls in **lc3sim**.



Code for the OUT Trap

For example, OUT is TRAP x21.

In M[x0021], we find x0450.

Listing x0450 gives the following…

```
TRAP_OUT        ST R1,TOUT_R1
TRAP_OUT_WAIT   LDI R1,OS_DSR
                BRzp TRAP_OUT_WAIT
                STI R0,OS_DDR
                LD R1,TOUT_R1
                RET
```

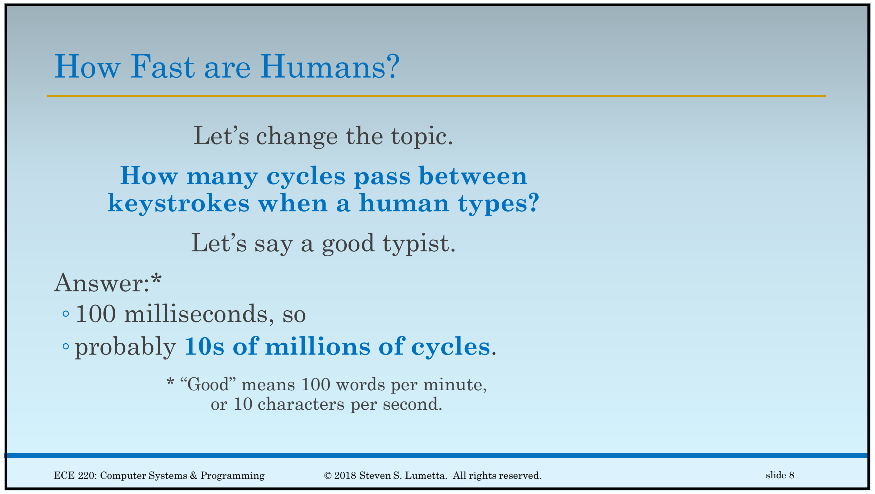
R1 saved to prevent changes

wait for display

write DDR

restore R1

JMP R7



How Fast are Humans?

Let's change the topic.

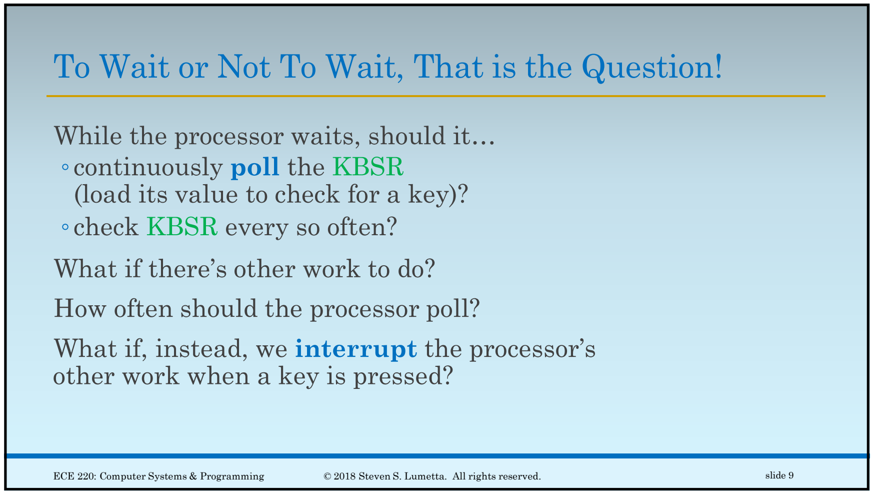
How many cycles pass between keystrokes when a human types?

Let's say a good typist.

Answer:*

- 100 milliseconds, so
- probably **10s of millions of cycles**.

* "Good" means 100 words per minute, or 10 characters per second.



To Wait or Not To Wait, That is the Question!

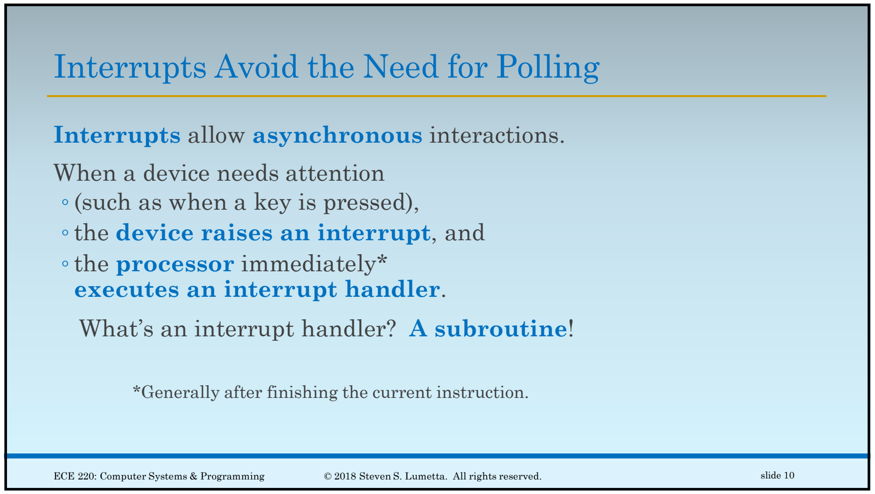
While the processor waits, should it…

- continuously **poll** the KBSR (load its value to check for a key)?
- check KBSR every so often?

What if there's other work to do?

How often should the processor poll?

What if, instead, we **interrupt** the processor's other work when a key is pressed?



Interrupts Avoid the Need for Polling

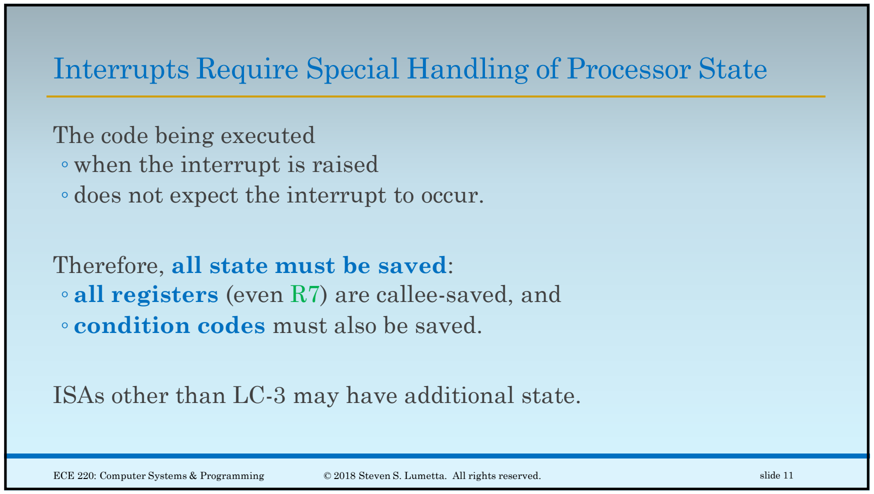
Interrupts allow **asynchronous** interactions.

When a device needs attention

- (such as when a key is pressed),
- the **device raises an interrupt**, and
- the **processor** immediately* **executes an interrupt handler**.

What's an interrupt handler? **A subroutine**!

*Generally after finishing the current instruction.



Interrupts Require Special Handling of Processor State

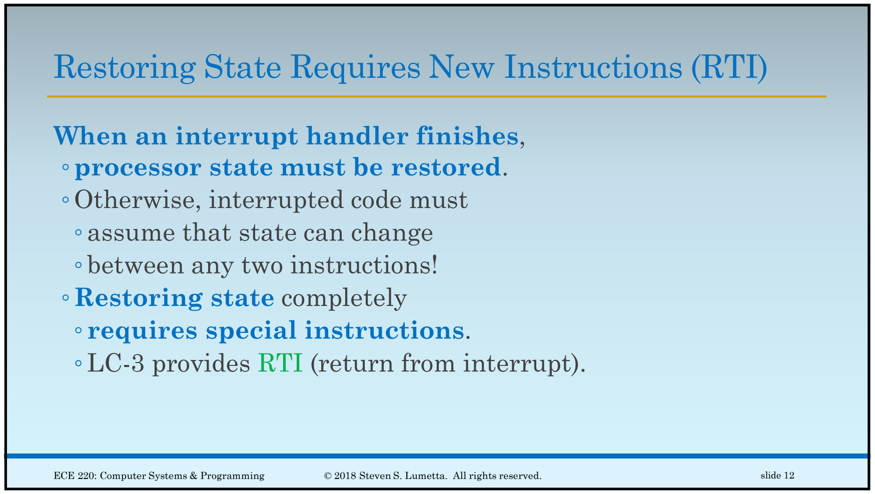
The code being executed

- when the interrupt is raised
- does not expect the interrupt to occur.

Therefore, **all state must be saved**:

- **all registers** (even R7) are callee-saved, and
- **condition codes** must also be saved.

ISAs other than LC-3 may have additional state.



Restoring State Requires New Instructions (RTI)

When an interrupt handler finishes,

- **processor state must be restored**.
- Otherwise, interrupted code must
  - assume that state can change
  - between any two instructions!
- **Restoring state** completely
  - **requires special instructions**.
  - LC-3 provides RTI (return from interrupt).

Can You Do Calculations?

I need your help again.

But … let me check your background first.

Without a calculator, how many of you can …
- do long division?
- calculate a square root?
- calculate transcendental and hypertranscendental functions (sin, cos, tanh, Γ, ...)?
- use a library to find out?

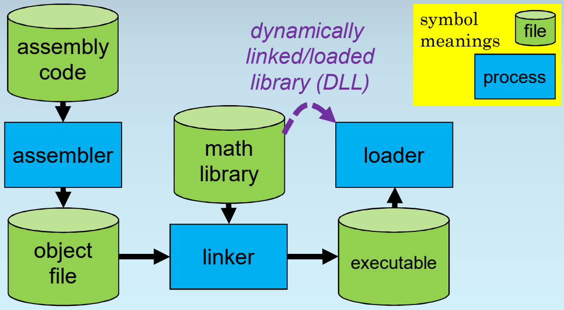
(The last skill is important!)

What is a Library?

In programming, a **library** is…
- a **body of subroutines** for common tasks
- typically **written in advance**
- **by someone else**, and
- **incorporated** into a program **by a linker**.

Examples from C include…
- the standard I/O library
- the math library

Libraries are Incorporated by the Linker

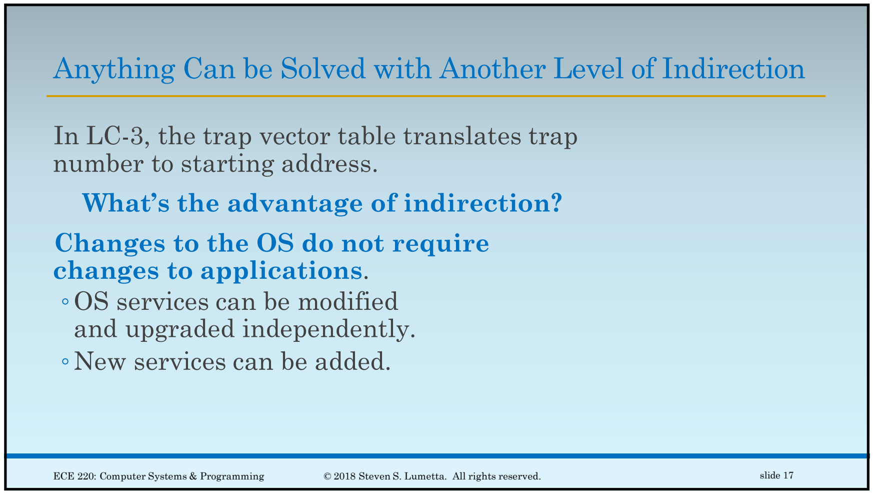


System Calls / Traps are a Library, Too

But the system calls provided by an OS are also a body of subroutines…

System Calls/Traps are (usually)
- **a set of library routines**
- usually **executed with privilege***
- **preloaded into the computer** (sometimes in ROM, as with BIOS)
- **accessed indirectly** (by number, not address)

*But not in the LC-3 ISA.



Anything Can be Solved with Another Level of Indirection

In LC-3, the trap vector table translates trap number to starting address.

What's the advantage of indirection?

Changes to the OS do not require changes to applications.

- OS services can be modified and upgraded independently.
- New services can be added.