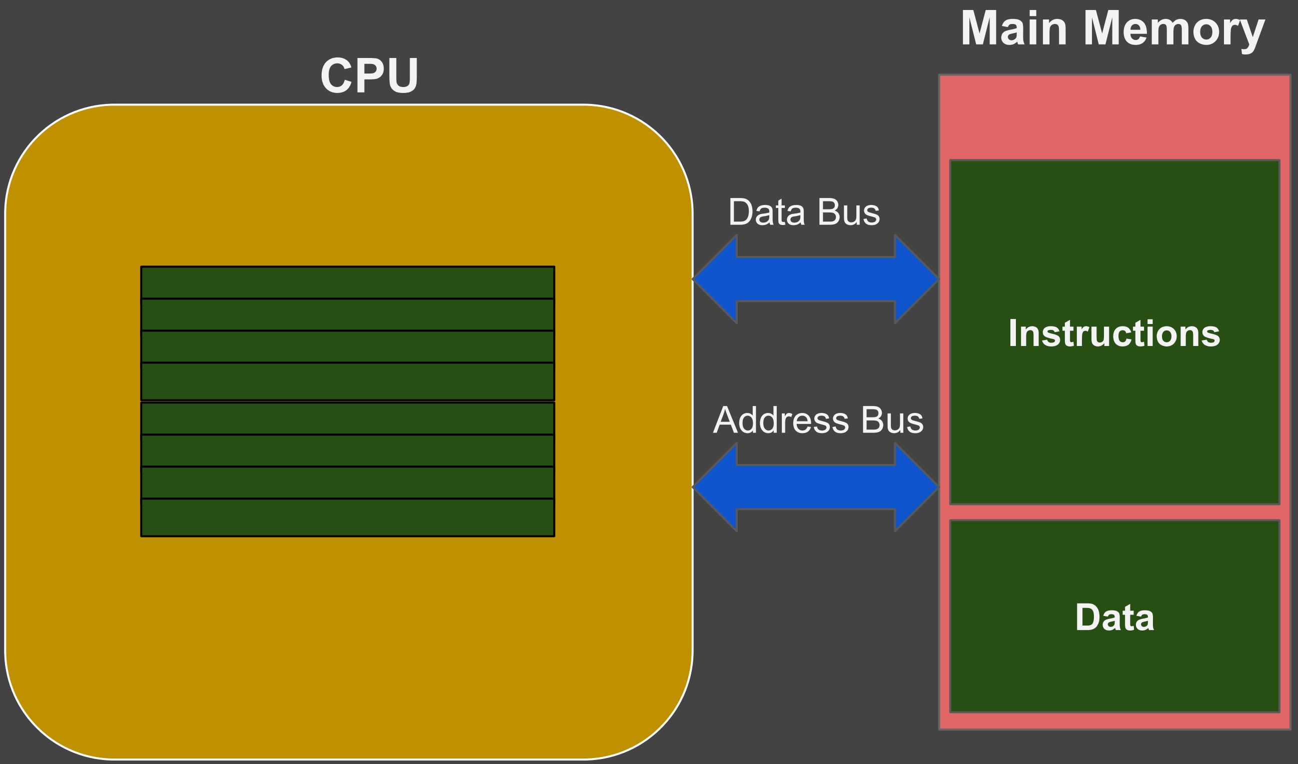
Introduction to x86 Assembly

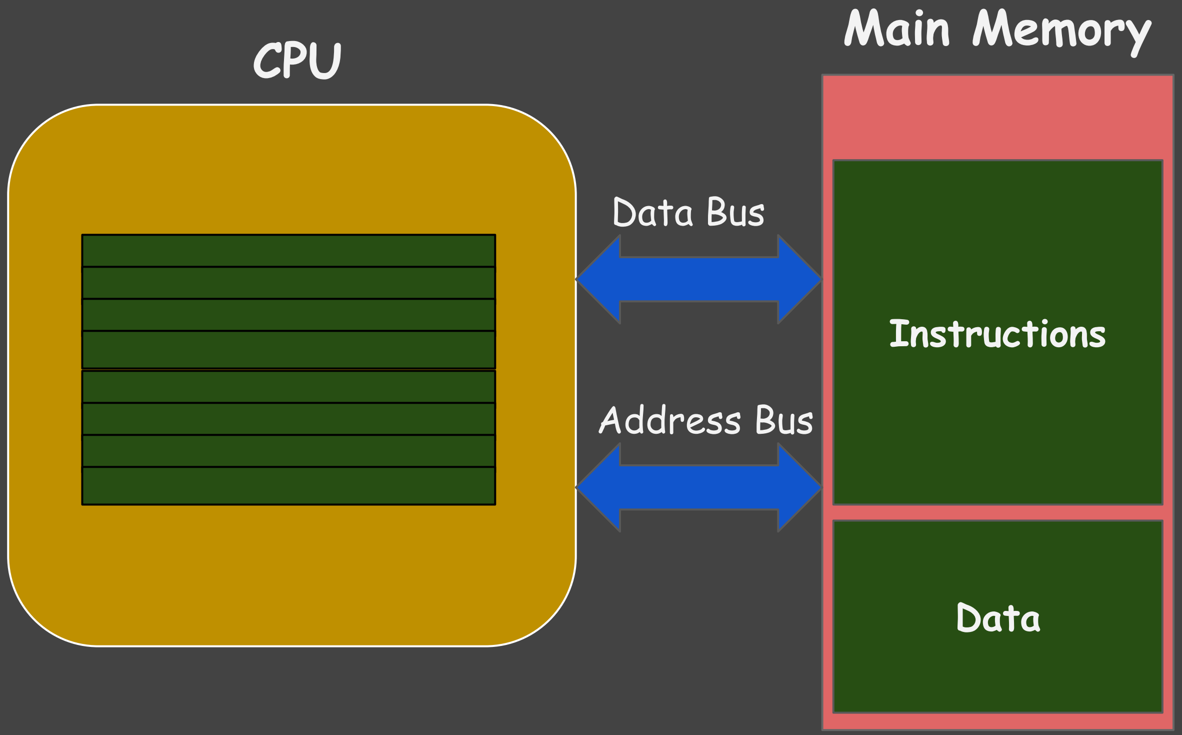
Lecture INT

x86 Interrupts

Remember: fetch-execute cycle



Remember: fetch-execute cycle



Instructions control CPU, or CPU controls instruction?

Remember: fetch-execute cycle

- How the OS can terminate/suspend your application?
- How multiple programs are run on a single CPU?
- How to handle events?
- How to get data from input devices while running?
- Alarm apps?
- real time system?

Remember: fetch-execute cycle

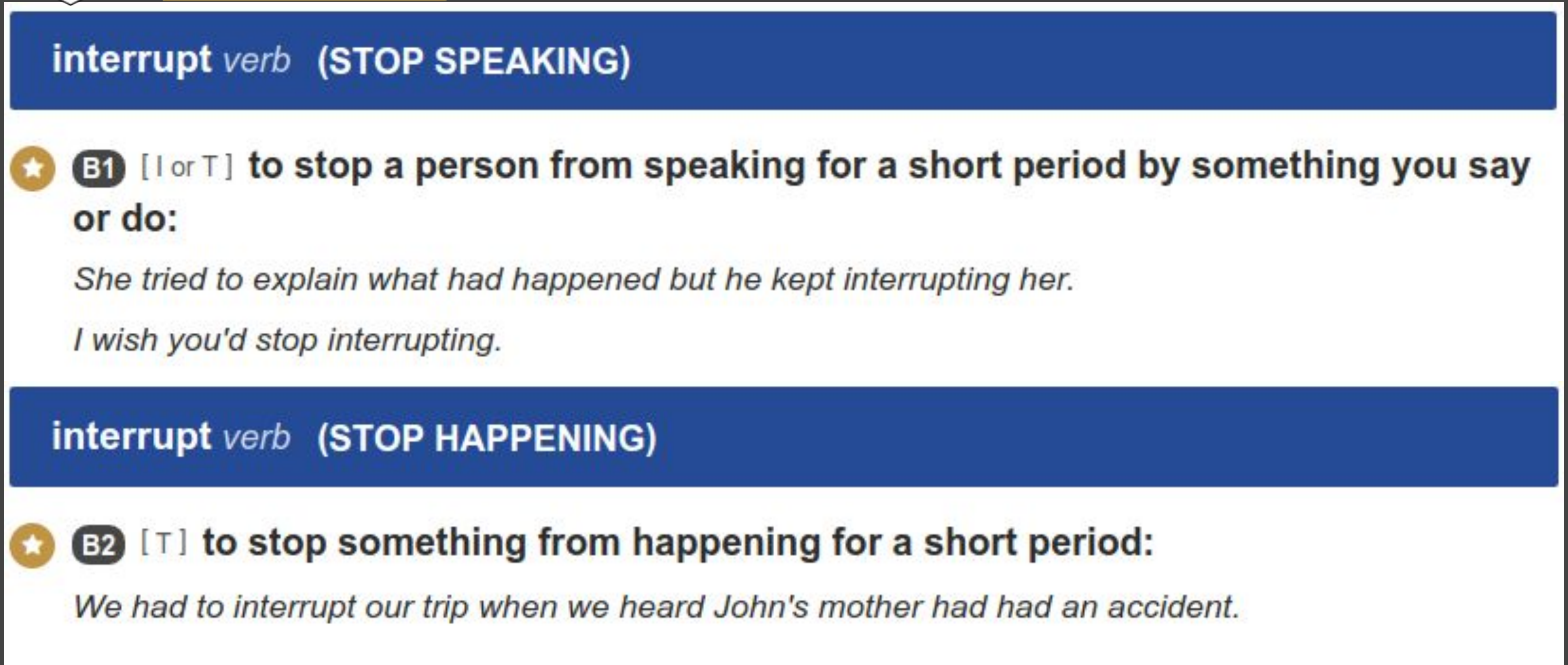
- How the OS can terminate/suspend your application?
- How multiple programs are run on a single CPU?
- How to handle events?
- How to get data from input devices while running?
- Alarm apps?
- real time system?

interrupts

Interrupt

Cambridge Dictionary

K. N. Toosi
University of Technology



interrupt *verb* **(STOP SPEAKING)**

★ **B1** [I or T] **to stop a person from speaking for a short period by something you say or do:**

She tried to explain what had happened but he kept interrupting her.

I wish you'd stop interrupting.

interrupt *verb* **(STOP HAPPENING)**

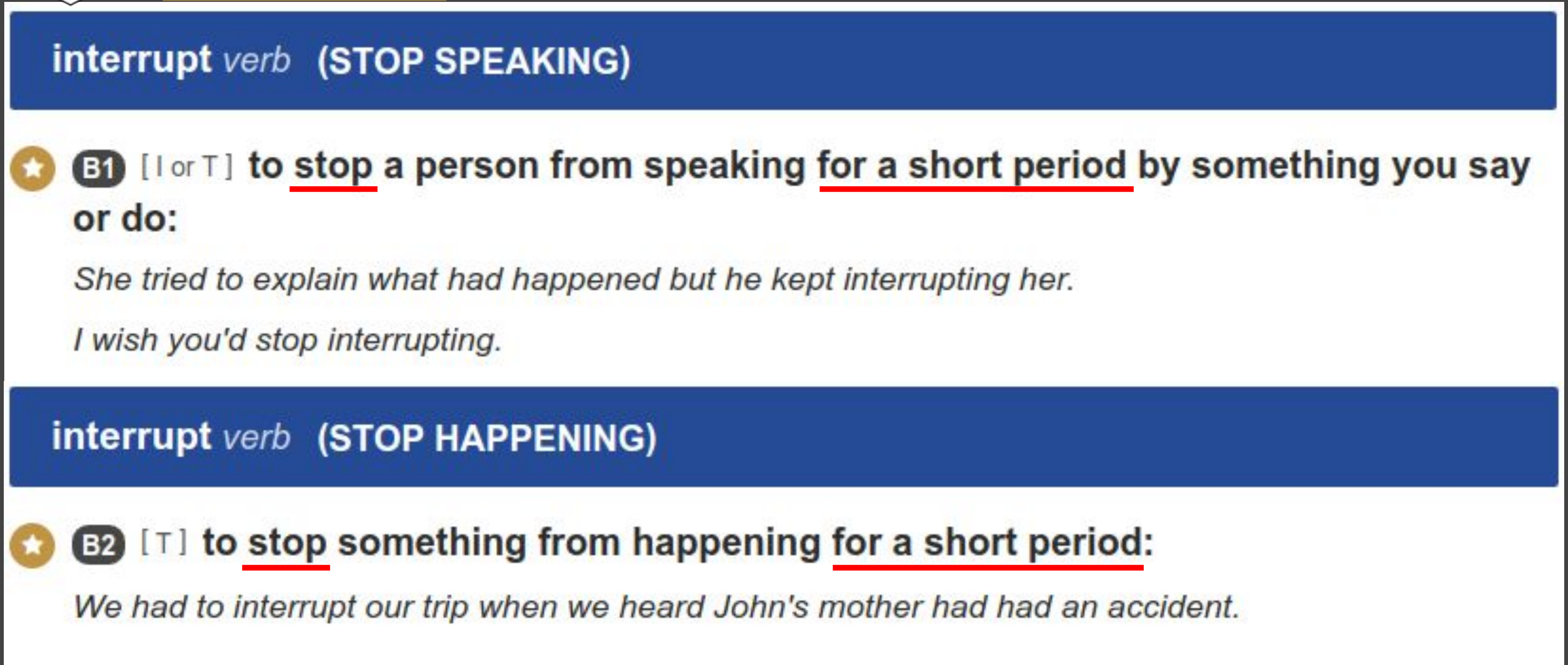
★ **B2** [T] **to stop something from happening for a short period:**

We had to interrupt our trip when we heard John's mother had had an accident.

Interrupt

Cambridge Dictionary

K. N. Toosi
University of Technology



interrupt *verb* **(STOP SPEAKING)**

☆ B1 [I or T] **to stop a person from speaking for a short period by something you say or do:**

She tried to explain what had happened but he kept interrupting her.

I wish you'd stop interrupting.

interrupt *verb* **(STOP HAPPENING)**

☆ B2 [T] **to stop something from happening for a short period:**

We had to interrupt our trip when we heard John's mother had had an accident.

Interrupt

- A signal to Processor
- Immediate attention
- Interrupt handler (interrupt service routine)

Types of Interrupts

- Hardware Interrupts
- Software Interrupts
- Exceptions

Hardware Interrupts

- A signal from hardware
- Input devices (keyboard, mouse, scanner, network interface,)
- Interrupt Request (IRQ)
- Asynchronous

Software Interrupts

- `int` instruction
- `INT` immed8
    - push EFLAGS
    - push (far) return address (CS + EIP)

Exceptions (traps)

- exceptional conditions in execution
  - divide by zero
  - access a protected memory area.
  - write on read-only memory area

Interrupt

- Hardware/Software Signals CPU
- CPU suspends current activity
- CPU saves its state (registers, flags, etc.)
- CPU calls an Interrupt Service Routine
- CPU retrieves its state
- CPU resumes previous activity

Real Mode: Interrupt Vector Table (IVT)

- 256 4-byte vectors, Address 0x0 to 0x3F of memory
- each vector CS+IP
- when interrupt happens:
  - FLAGS, CS and IP are pushed

Software Interrupts

Software Interrupts

Interrupt service routine (ISR)

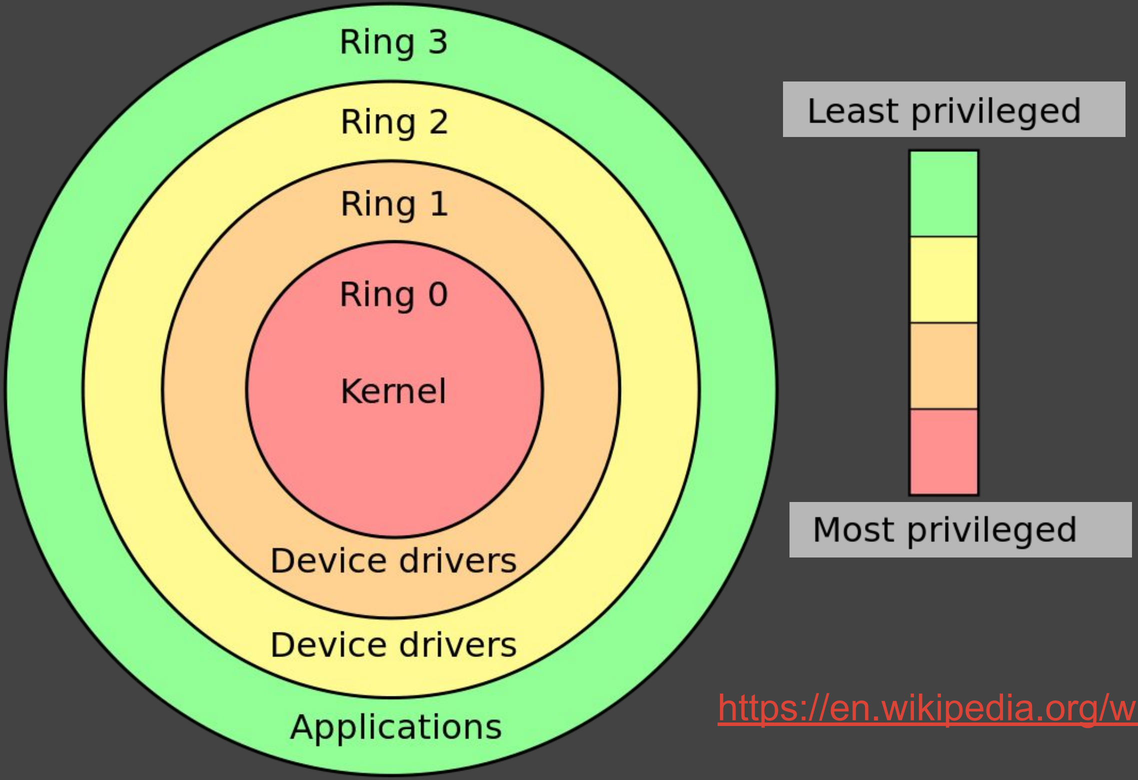
- AKA Interrupt handler
-

Software Interrupts

- IRET (NASM)

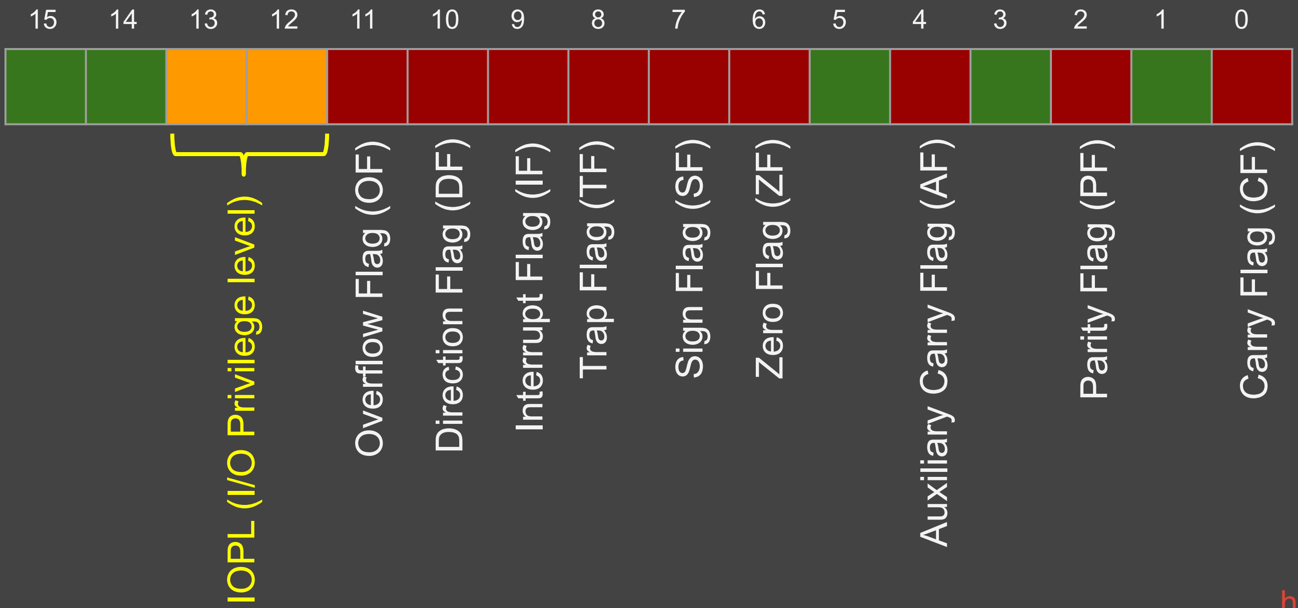
- Some other assemblers
  - IRET
  - IRETD
  - IRETQ

Protection ring

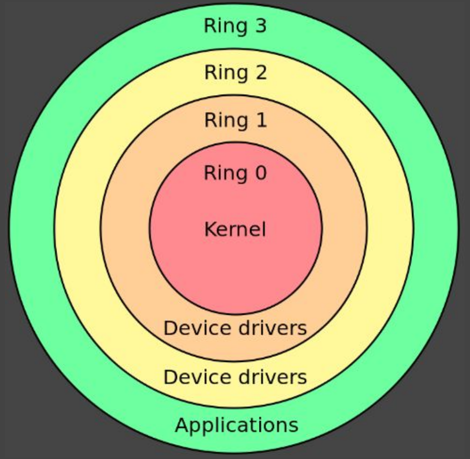


https://en.wikipedia.org/wiki/Protection_ring

The IPOL flag



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
🟩	🟩	🟧	🟧	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟩	🟥
		IOPL (I/O Privilege level)		Overflow Flag (OF)	Direction Flag (DF)	Interrupt Flag (IF)	Trap Flag (TF)	Sign Flag (SF)	Zero Flag (ZF)		Auxiliary Carry Flag (AF)		Parity Flag (PF)		Carry Flag (CF)



https://en.wikipedia.org/wiki/Protection_ring

Software Interrupts

- INT immed8
- INT3
- INT1
- INT0 (32-bit mode only)
- Most software interrupts cannot be called from user space

User-space programs

- User-space applications are run at privilege 0
- Cannot call most software interrupts
- Use OS API's instead

References

- https://en.wikibooks.org/wiki/X86_Assembly
- https://wiki.osdev.org/Interrupts