

Operating Systems!

An Overview of Computer Systems (Part I)

Prof. Travis Peters

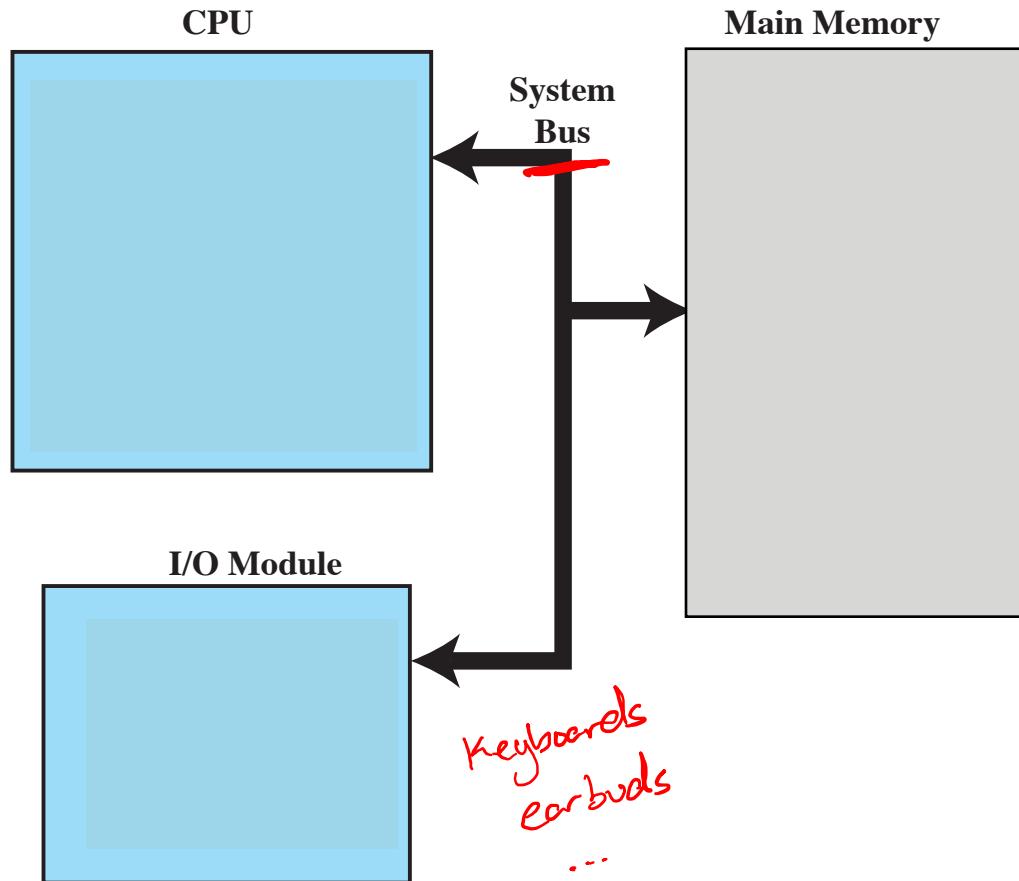
Montana State University

CS 460 - Operating Systems

Fall 2020

<https://www.cs.montana.edu/cs460>

Today



Announcements

- Sign up for Slack!
- Silly Easter Egg (*Zoom Invite!*)
+ D2L...

Learning Objectives

- Review basic elements of **computer systems** and their interrelationships
- Review important system concepts (e.g., processor execution, interrupts, memory, I/O)

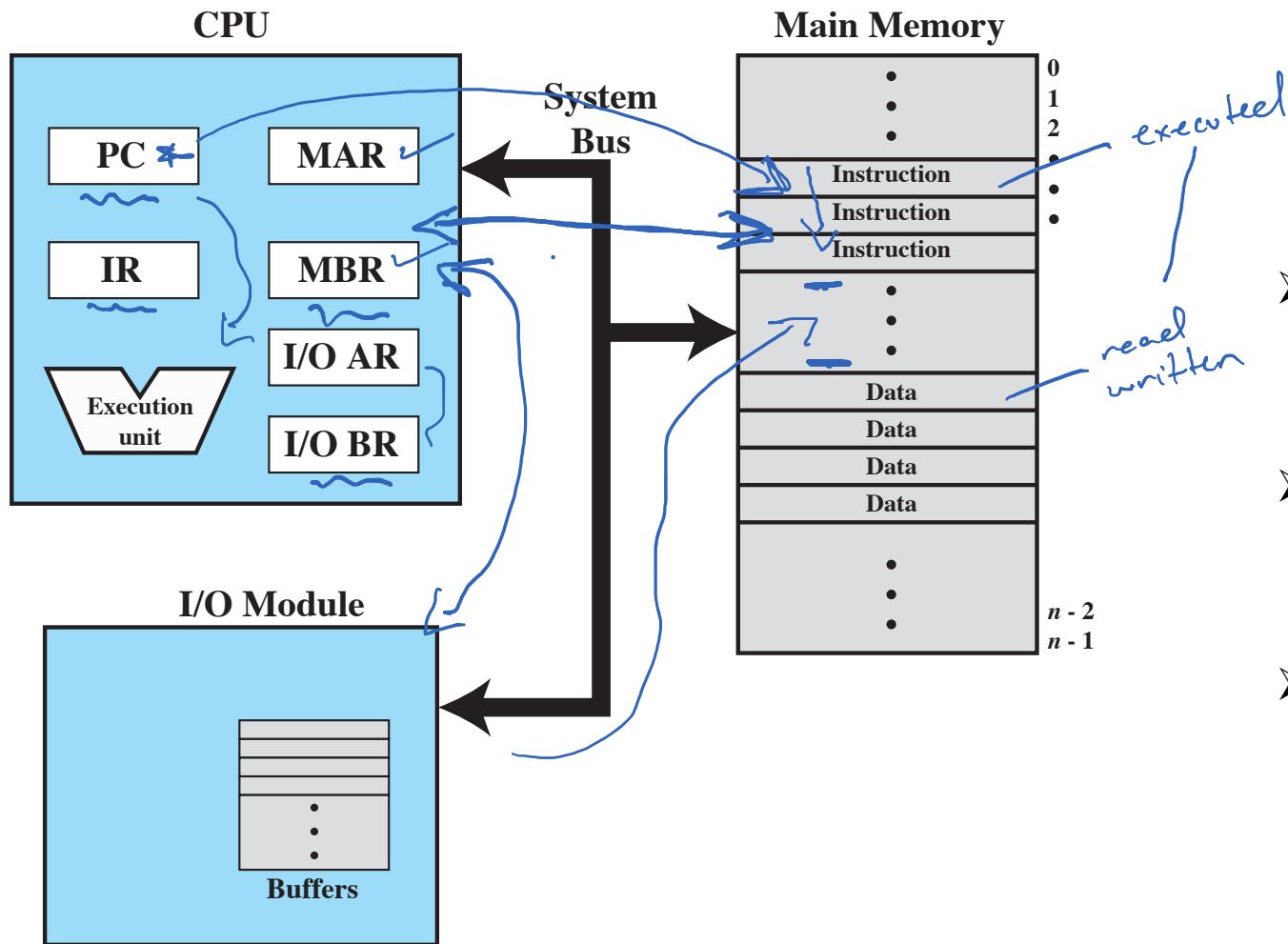
Leftovers From Last Time...

What Does “Operating System” Make You Think?

Breakout Questions!

- OK... So what is an OS?
 - makes a computer more convenient to use
 - intermediary
 - coordinator / schedule / resource manager
- Different types of systems & OSs?
 - specialized systems
 - scientific computing
 - security *
 - constrained systems *
 - ⋮

Computer System: A High-Level View



- addresses specify where to read/write main memory and I/O devices
- registers hold addresses, results, etc. close to the processor
- buffers hold read/write data

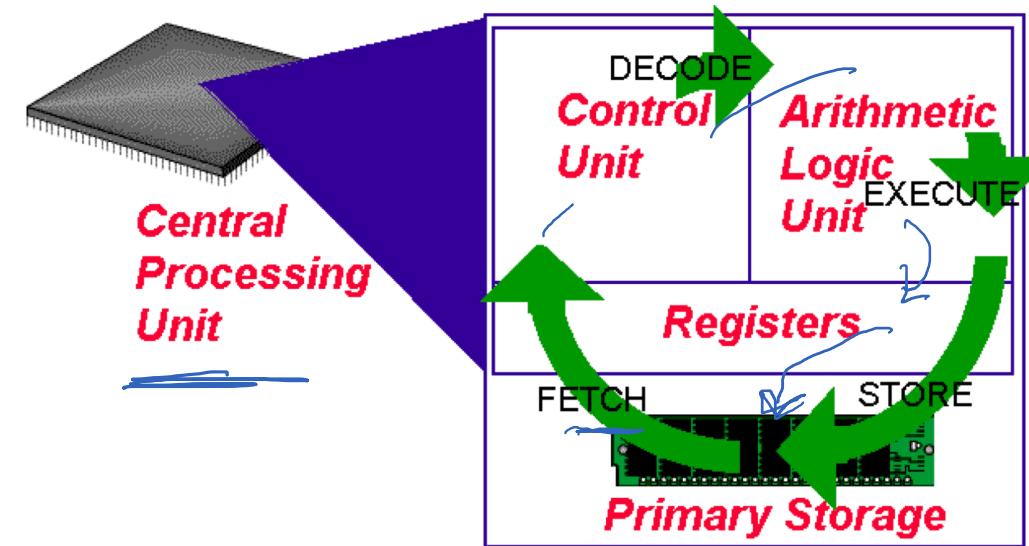
Processors

*What follows are the big ideas,
which gives us context for the role and purpose of the OS.*

Processors In a Nutshell

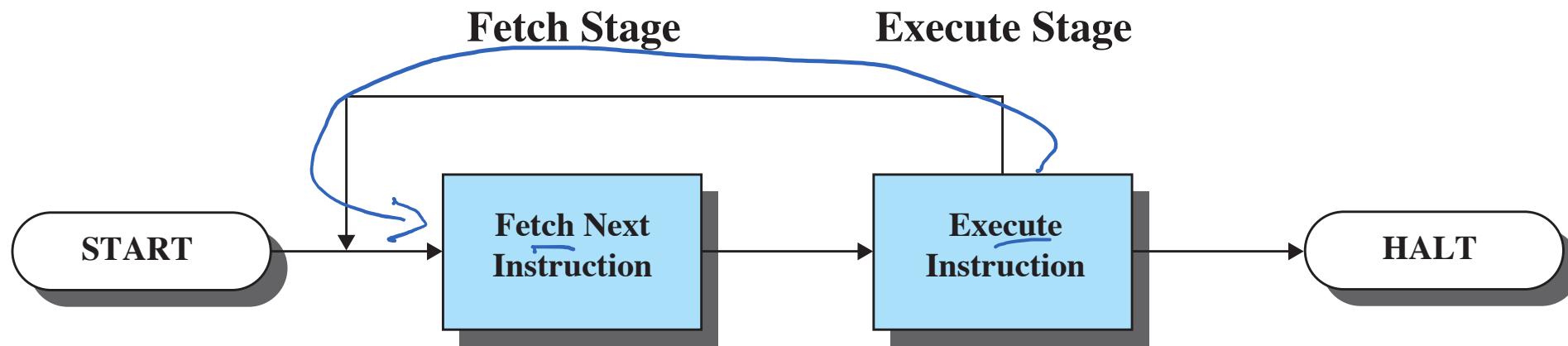
- Processors control the operation of the computer
- A program is just a set of *instructions* & *data* stored in memory
- Processor continuously runs “instruction cycle”

PLC FETCH the next instruction,
DECODE it,
EXECUTE it, and
STORE the result



*Most instructions just move data around,
do basic math or logic operations, or change
the control flow.*

Processors In a Nutshell



The book uses a nice flow chart...

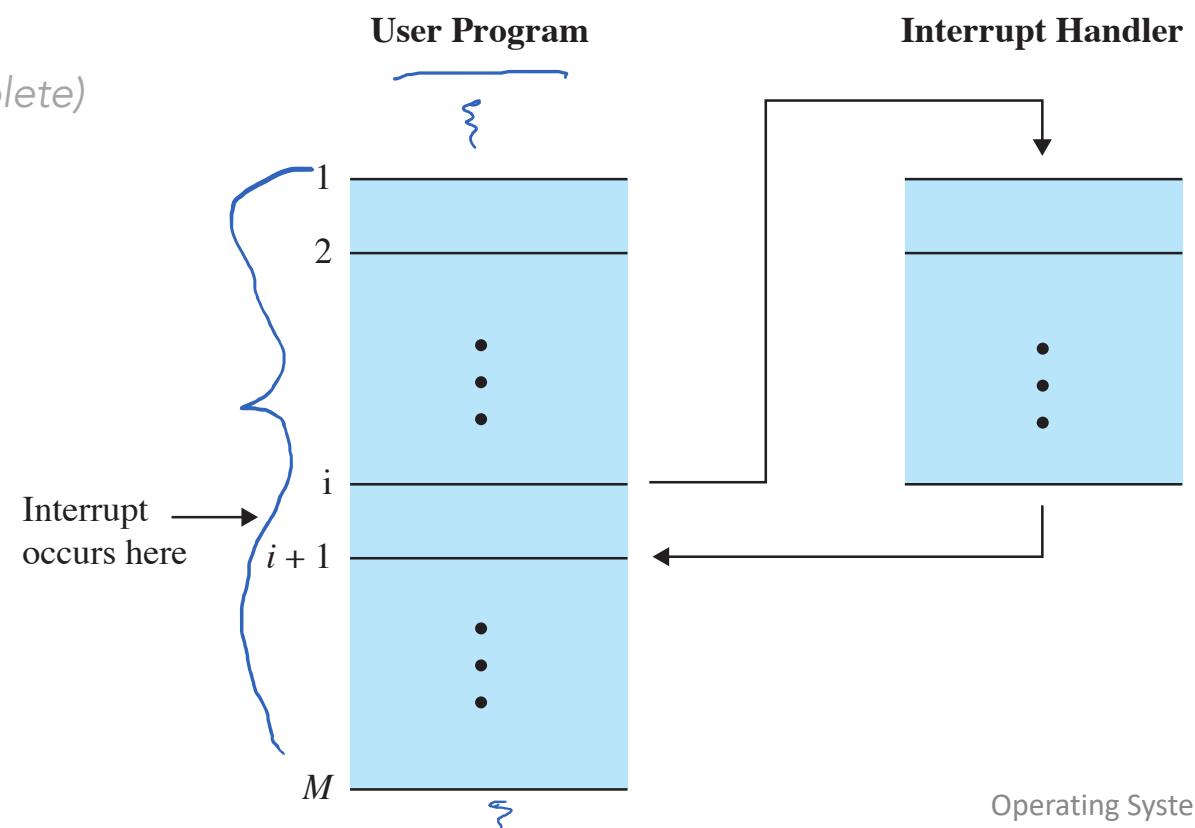
Sharing/Yielding the Processor with *Interrupts*

- Interrupts... “interrupt” the normal sequencing of the processor to improve processor utilization.

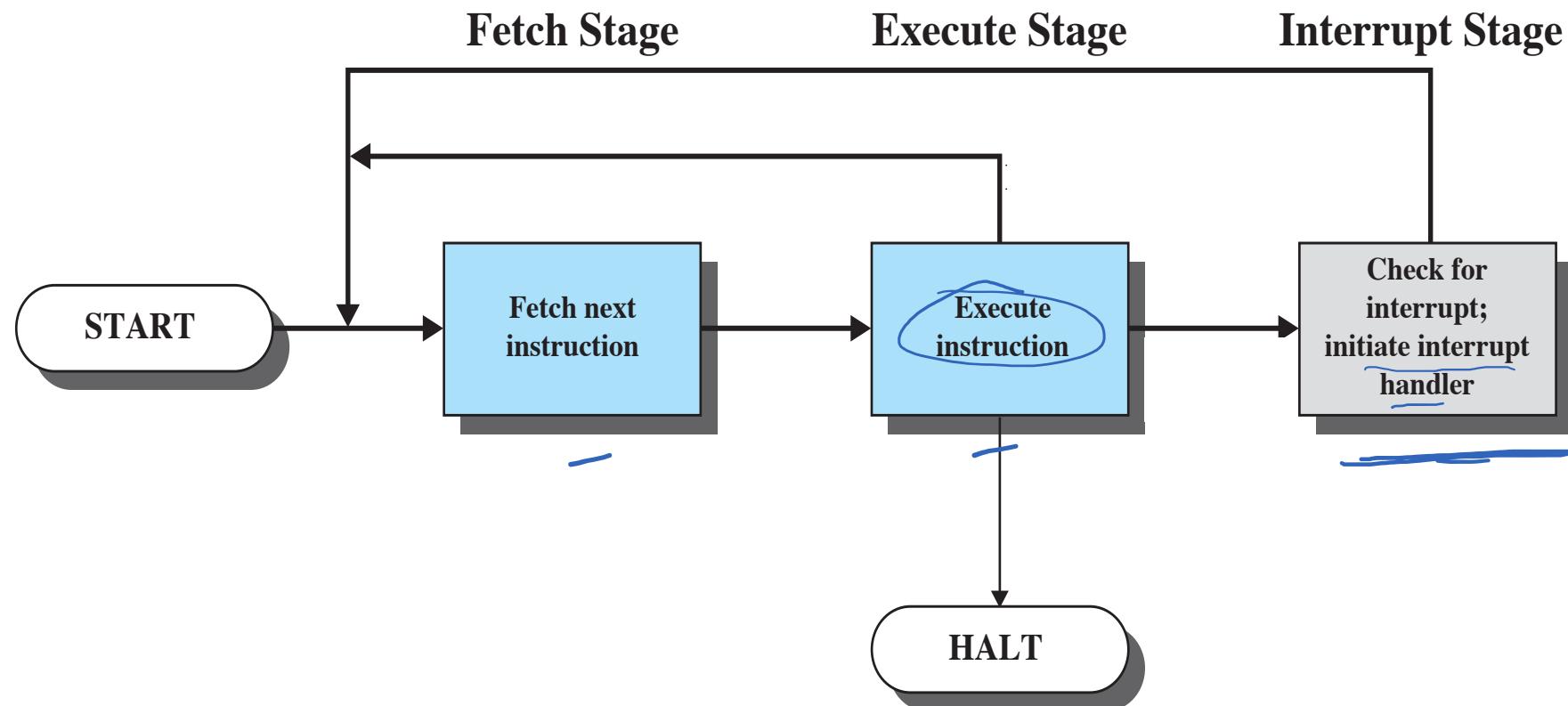
- program interrupt (e.g., illegal instruction, out-of-bounds access)
- timer interrupt (e.g., interrupt every X cycles / Y μ s)
- I/O interrupt (e.g., packet arrived, disk transfer complete)
- hardware failure (e.g., connection dropped)

→ run a “handler”

Why? *Sharing is caring; Most I/O devices (and people) are slowwwww (compared to the processor); enables OS to regain control, ...*



Sharing/Yielding the Processor with *Interrupts*



Sharing/Yielding the Processor with *Interrupts*

Breakout Question!

What do you do if another *interrupt* occurs while you are in the middle of processing an *interrupt*?

- pros/cons of your approach?

② Logic: decide which interrupt is more important.

- dependency
- privilege
- time sensitive

