

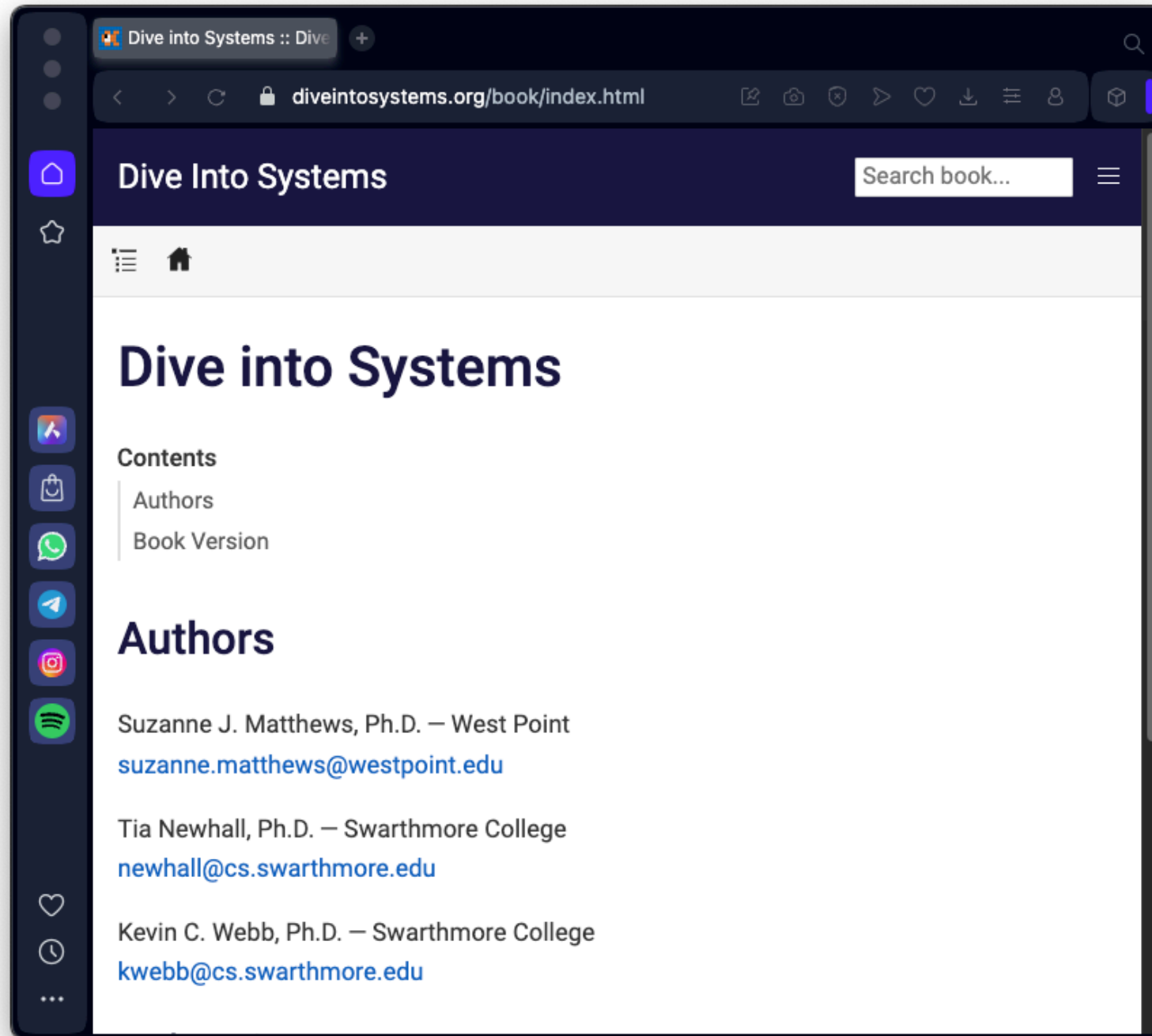
0. Introduction

For COMSC 142

This course covers

- How a C program executes on a computer
- How programs translate into binary
- How circuits execute binary encoding
- How operating systems manage programs
- Assembly language
 - 32-bit x86
 - 64-bit x86
 - ARMv8

Free online textbook



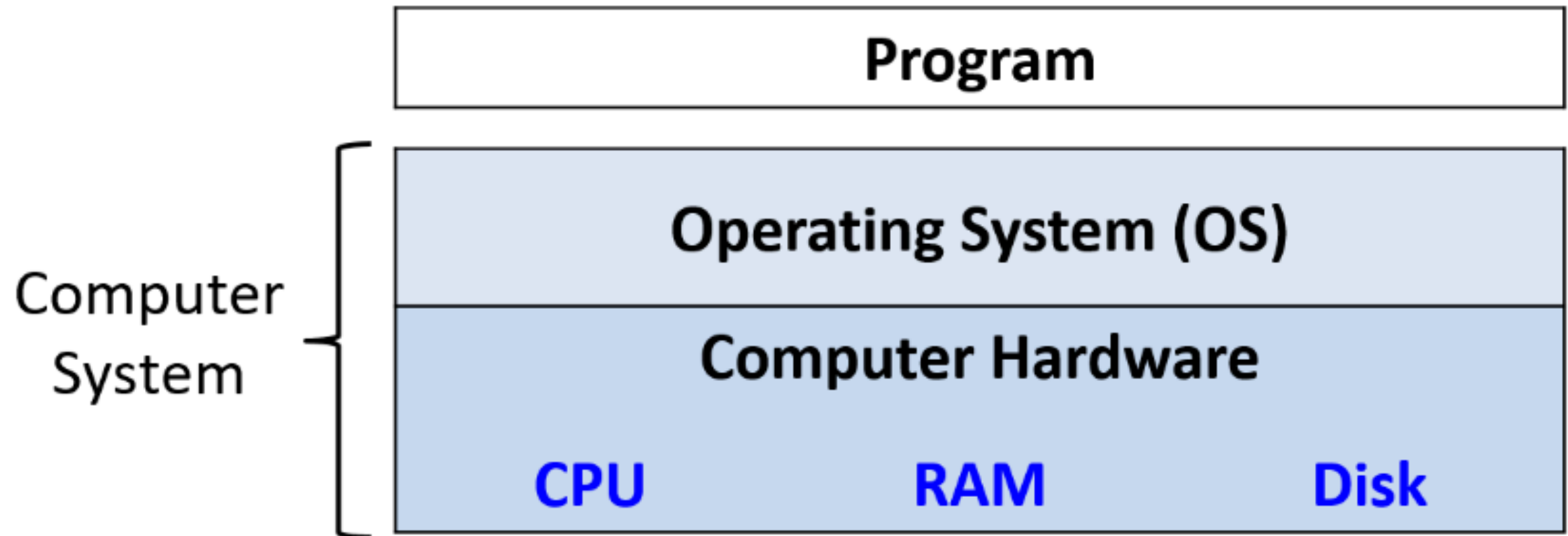
- <https://diveintosystems.org/book/index.html>

Topics

- Computer System Components
- What You Will Learn
- Linux, C, and the GNU Compiler

Computer System Components

Computer System Components



Hardware

- Input/output (IO) ports
 - Takes data from outside and displays it back to the user
- Central processing unit (CPU)
 - Runs instructions on data and memory addresses

Hardware

- Random Access Memory (RAM)
 - Stores data and instructions
 - Loses data on power-off
- Secondary storage devices
 - Hard disks and SSDs
 - Store data even with power is off

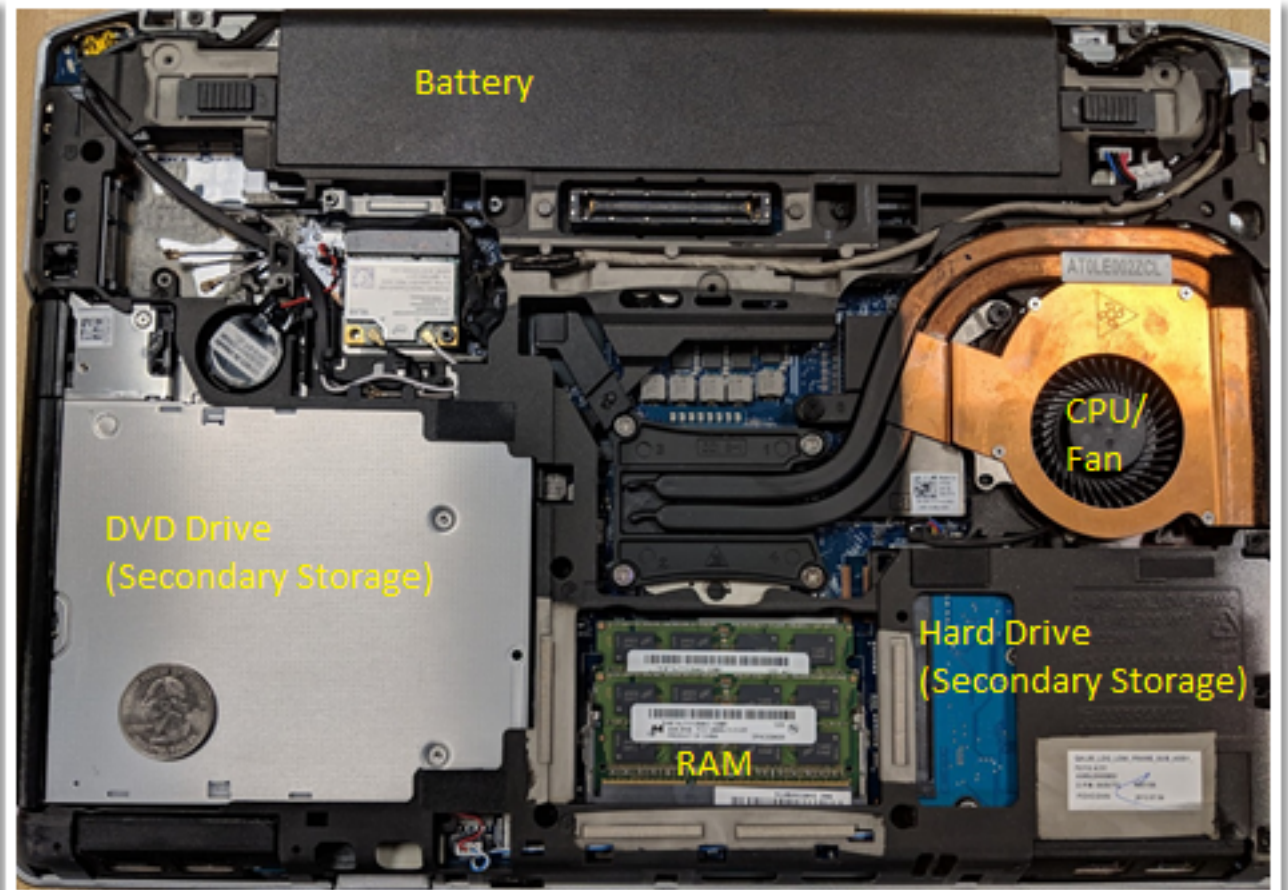
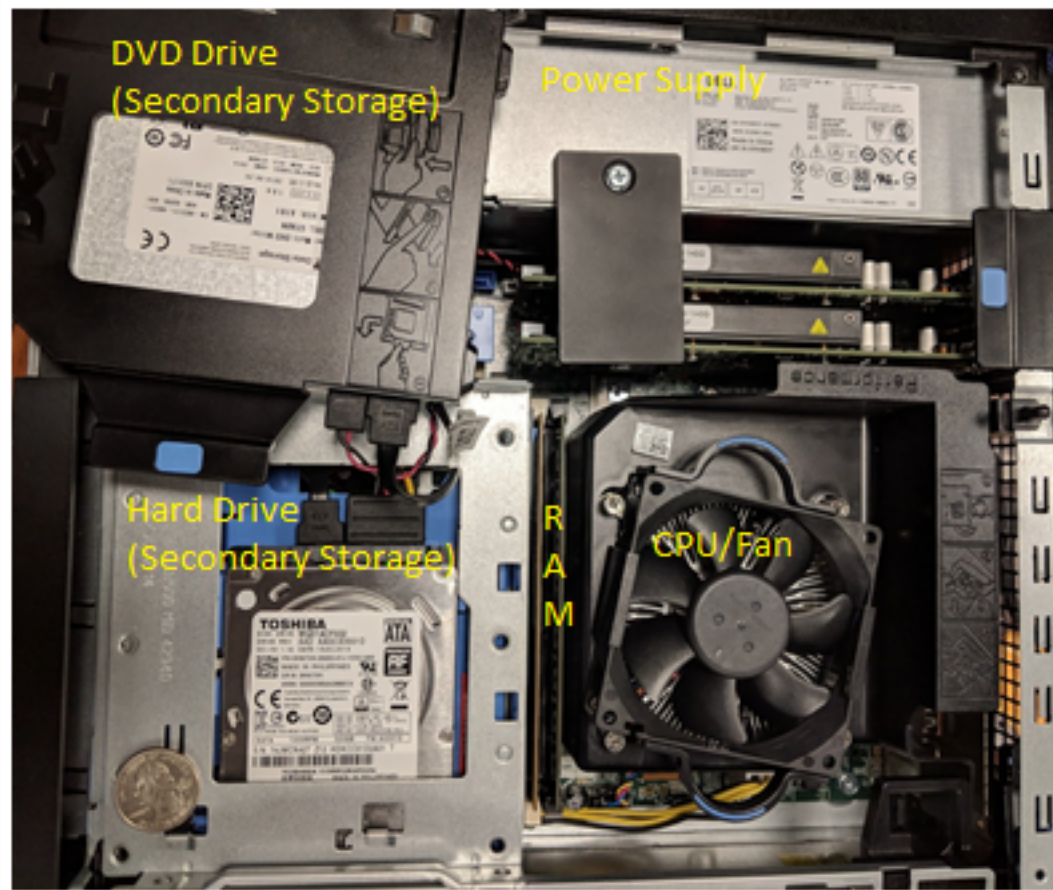
Operating System (OS)

- Software layer between hardware and the software that a user runs
- Implements programming abstractions like processes and the file system
 - So developers don't need to understand hardware details
- Manages hardware resources
- Controls how and when programs execute
- Allows multiple programs to run simultaneously

Types of Computers

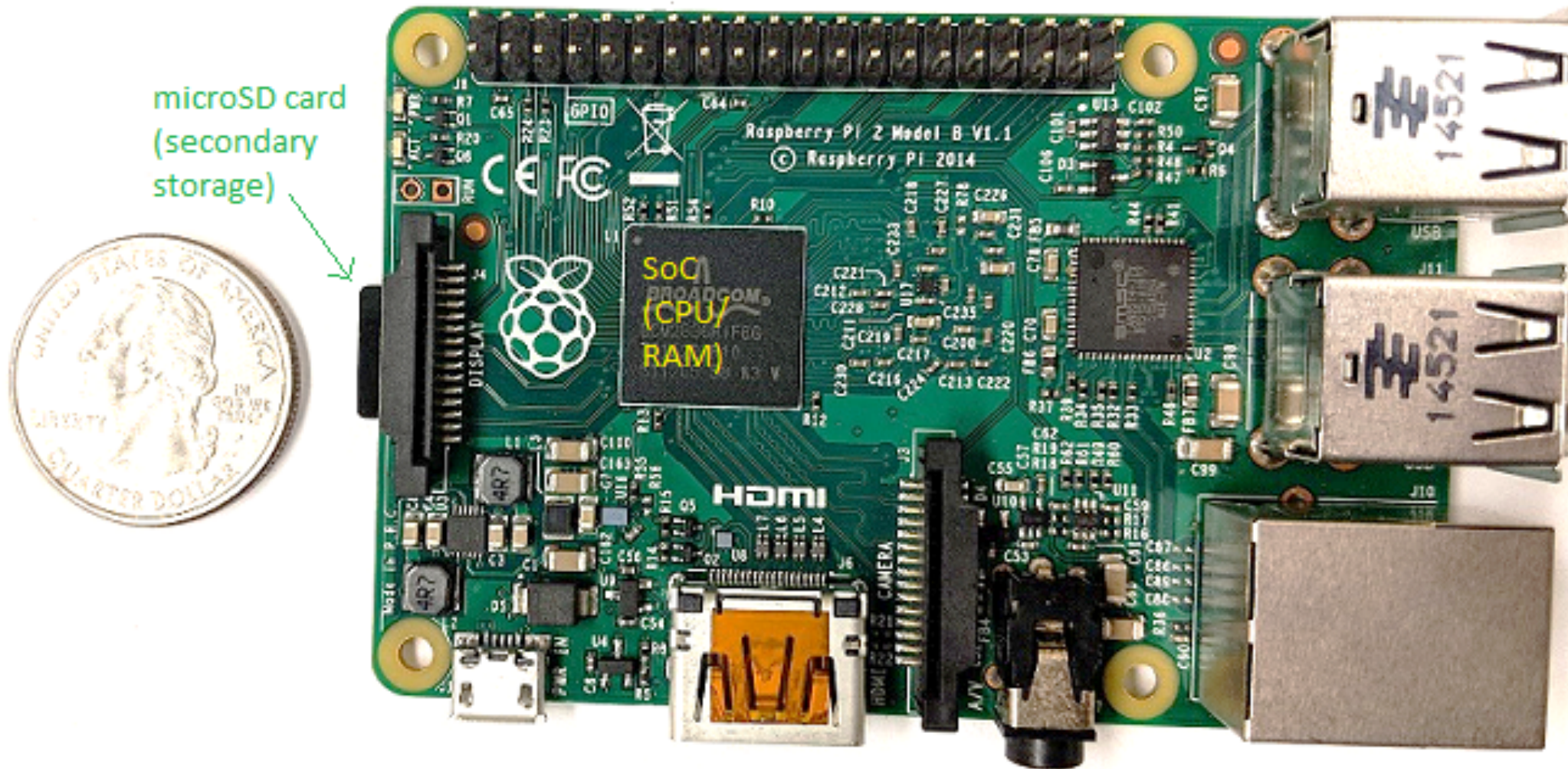
- **General purpose and reprogrammable**
 - Most common type
- **Calculators**
 - Not general purpose, don't generally have an OS
- **Microcontrollers**
 - Embedded in devices, run only one program
 - Have no OS

Desktop and Laptop



Single-Board

- Raspberry Pi
- System-on-a-Chip (SoC)



Multicore Processors

- Can execute multiple programs simultaneously
 - Parallel execution
- A feature of desktops, laptops, Raspberry Pis and smartphones

Operating Systems

- MacOS
- Windows
- Linux/Unix

What You Will Learn

How a Computer Runs a Program

- Encoding a program into binary
- Compiler, assembly and binary machine code
- CPU executes binary instructions on binary data
- OS
 - Interfaces with user
 - Controls program execution
 - Manages system resources

System Costs and Performance

- Algorithm choice
- How programs use system resources
 - Memory hierarchy
 - Code optimization
- Using parallel computing
 - Thread abstraction
 - Parallel programming models

Linux, C, and the GNU Compiler

The C Language

- High-level language
- Less abstracted from hardware than other high-level languages
- More control over hardware

GNU C Compiler (GCC)

- Runs on Linux
- Free and open-source
- Dominant on supercomputing systems
 - Although not on consumer PCs

Kahoot!

Ch 0