Introduction to Computer Security (Part II)

Professor Travis Peters
CSCI 476 Computer Security
Spring 2020

Today

Announcements

- We need a note taker for the class! ➡ ***Contact ODS if interested***
- Lab 00 ➡ ***It's up!***
- Bring your laptops (especially on Thursdays—in general, these will be more hands-on days!)

Goals & Learning Objectives

- Review some basics
  - Models/layout of a computer & a program
  - Basic C programming
  - Basic command line usage
  - Linux & Basic Linux Security

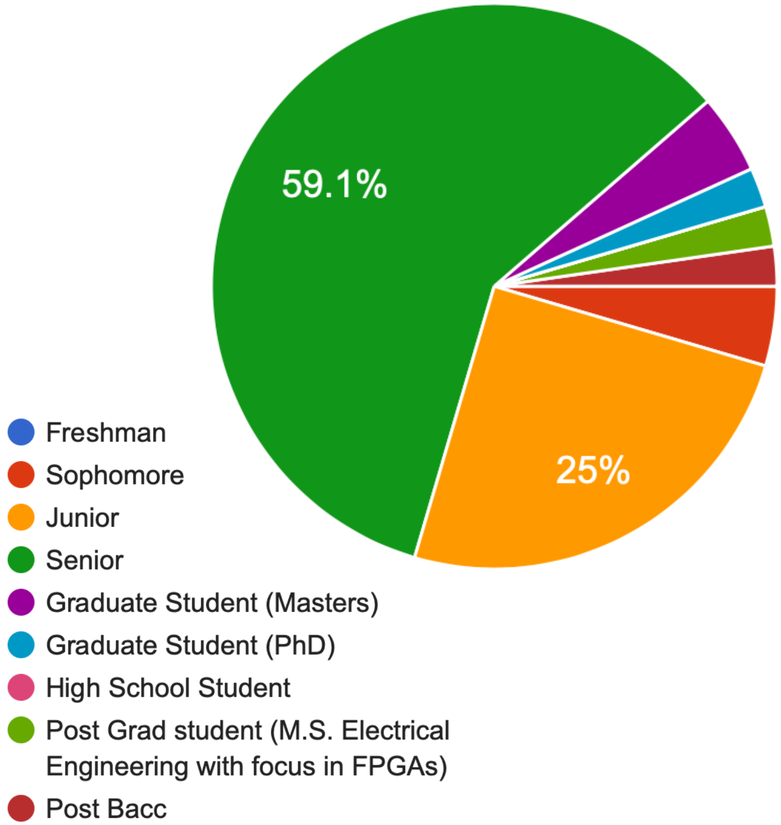
BUT FIRST, Some Insights From the Questionnaire!

The results are in! (mostly…)

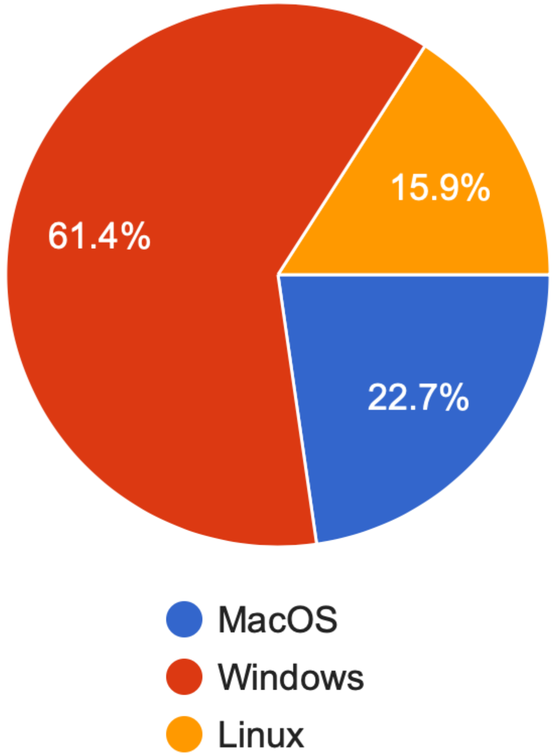
First, Some Insights From the Questionnaire!

Some insights into who we are, our coursework backgrounds, and what types of OS/machines we use

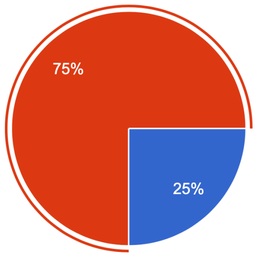
Who do we have in the class?



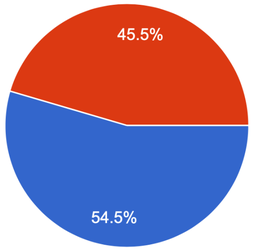
What OS/machine are people using?



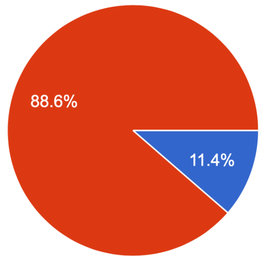
Have you taken OS?



Have you taken Networks?

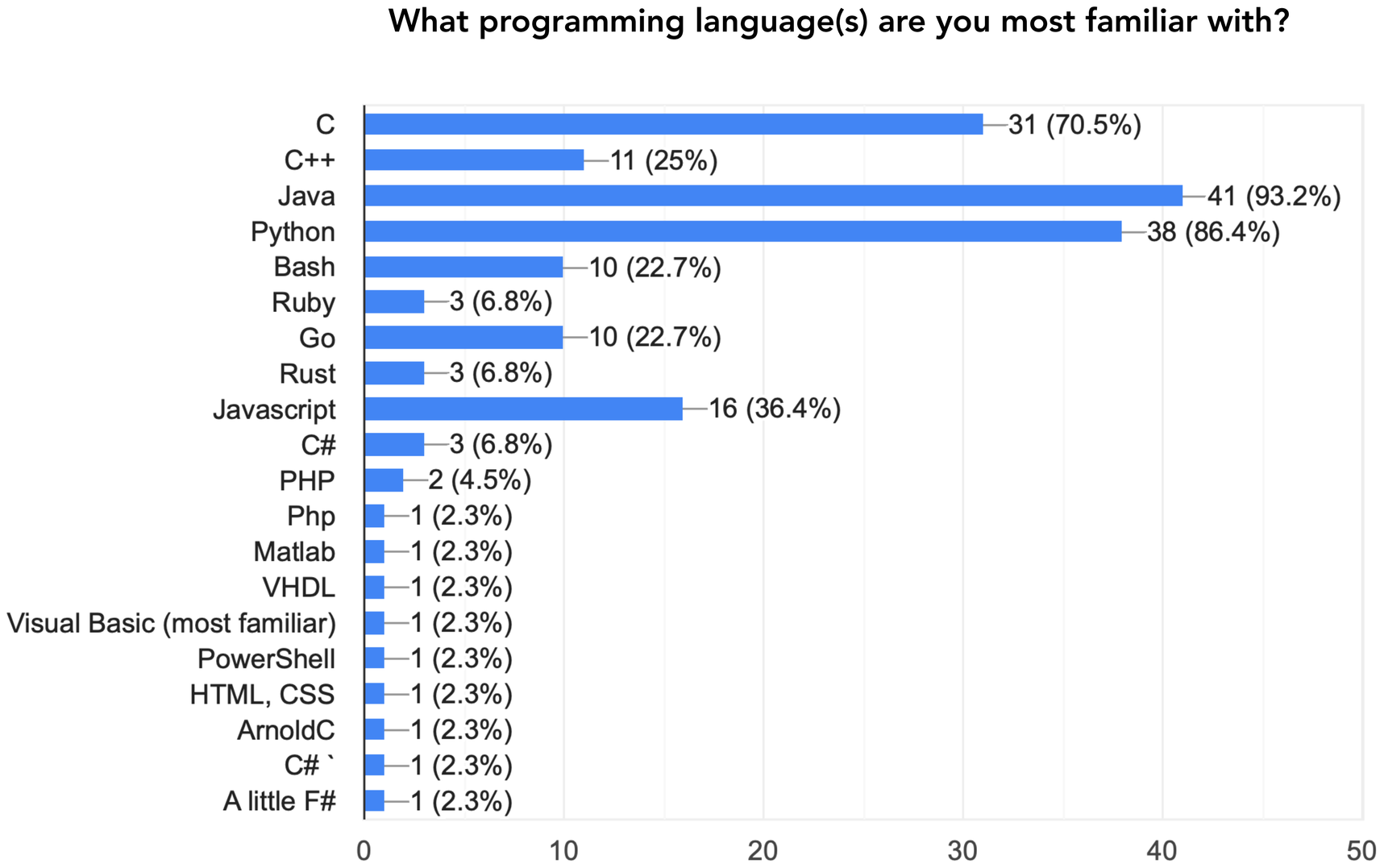


Have you taken Security?



First, Some Insights From the Questionnaire!

Some insights into what programming languages we use



First, Some Insights From the Questionnaire!

Some comments from the class

So I can understand what threats actually exist in the real world to reduce the exposure of the attacks within the systems I build.

I want to do Infosec as a career

I'd like to explore options for a career involving computer security but I don't yet have a security background.

I want to learn how to protect information.

Why are you taking CSCI 476?
What do you really hope to learn?

it's a 400 level class

need the credit, sounded really interesting

Some people also said nice things about me…

People also seemed to be interested in…

Web Security (e.g., JWT), Crypto, Internet Security, VPNs, Cybercrime, ML in Security, DNS Vulns, CPU Vulns, Best Practices, …
How to be the best hacker ever
How to break all the things

…we won't cover all of these topics, but it should be a good starting point for you!

First, Some Insights From the Questionnaire!

Some comments from the class (cont.)

I generally like the option to work alone on projects and assignments.

I work full-time/ on the weekends/ part-time as an intern

I really like to read textbooks. I prefer an introduction to the topic in class so that I can further read about it in my own time. I wouldn't mind occasional lab periods, but I prefer going to your office hours for specific questions.

I love programming projects that let me apply what I'm learning.

Anything else I should know about you?

Who's honestly good on exams?

I hate and love computers.

I would love to do research in this area, if you have any ideas or advice please let me know!

C > Python, Linux > Windows, Ford > Chevy

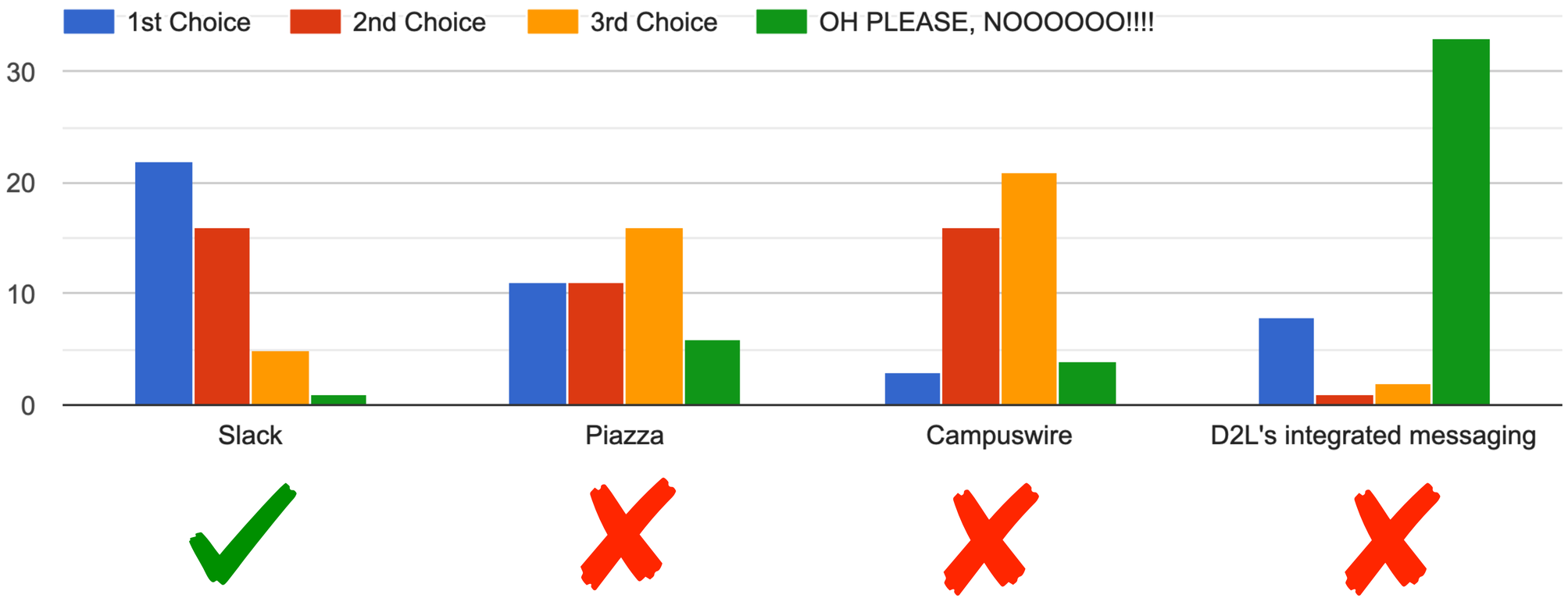
Office hours definitely scare me because I never feel like I know what to ask

This class intimidates the heck out of me. Answering your questions about OS and networks is concerning me even more because I don't know really anything about either of those.

First, Some Insights From the Questionnaire!

Some insights into our communication preferences

If we were to use ONE TOOL for COURSE COMMUNICATIONS, what is your preference?



First, Some Insights From the Questionnaire!

Some insights into our communication preferences

*Some in class exploration — let's take a quick peek at **Slack***

— https://media.tenor.co

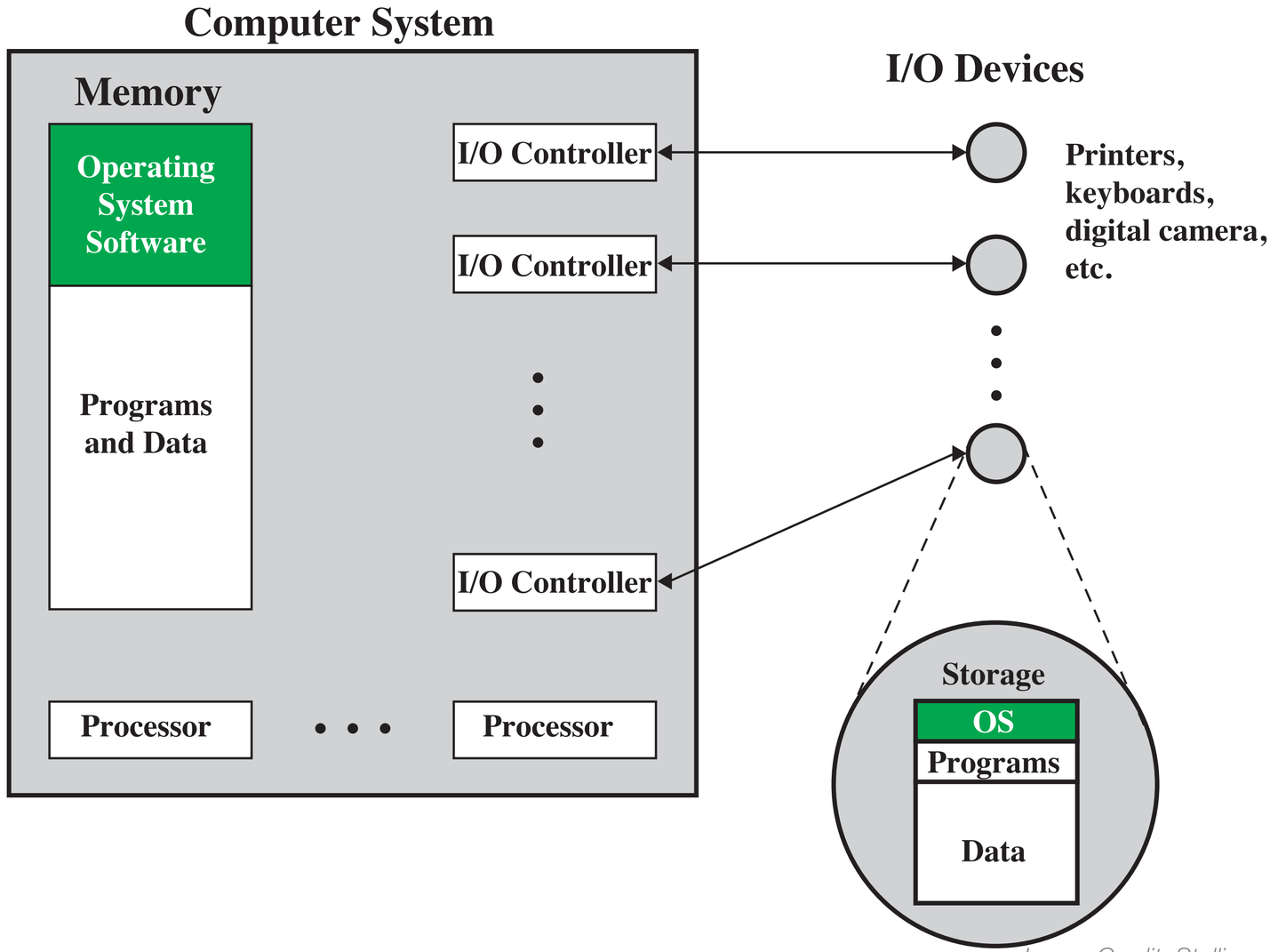
Some Review

Many of the following concepts are specific to the UNIX family (specially Linux), which is most relevant for this course.

A lot of the ideas, however, are universally relevant.

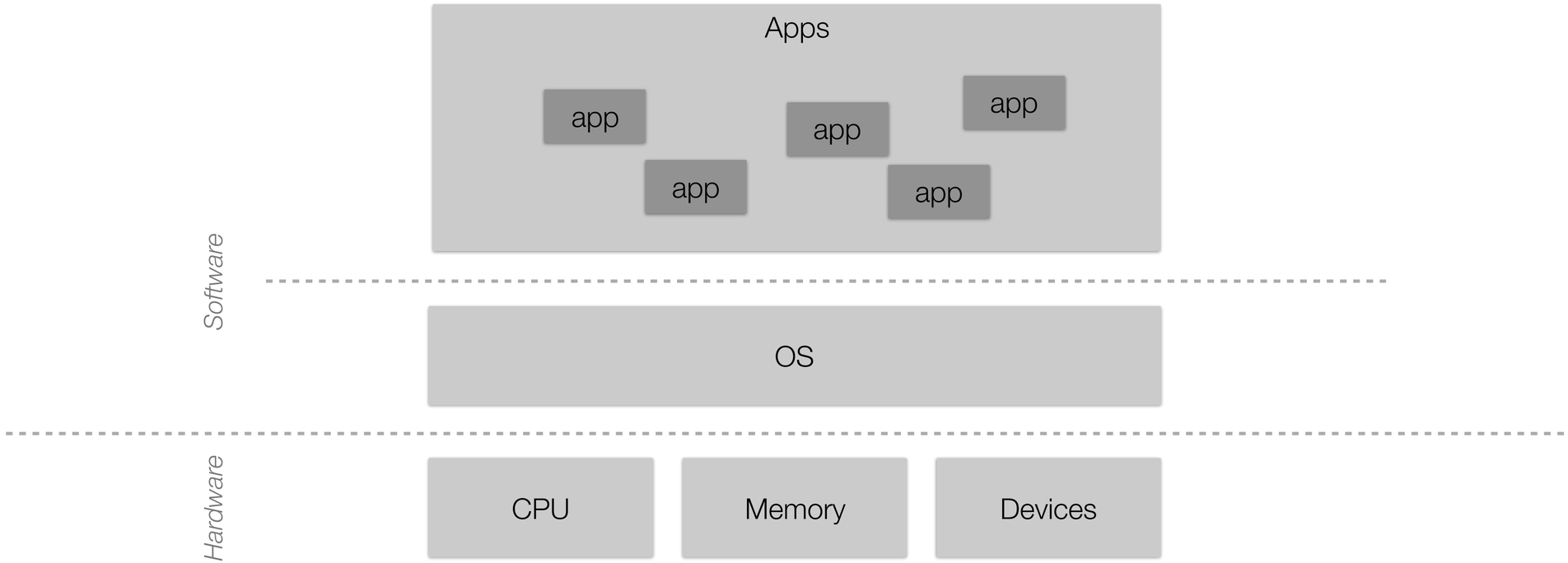
Background: A Computer in a Nutshell

A computer, is a computer, is a computer, …

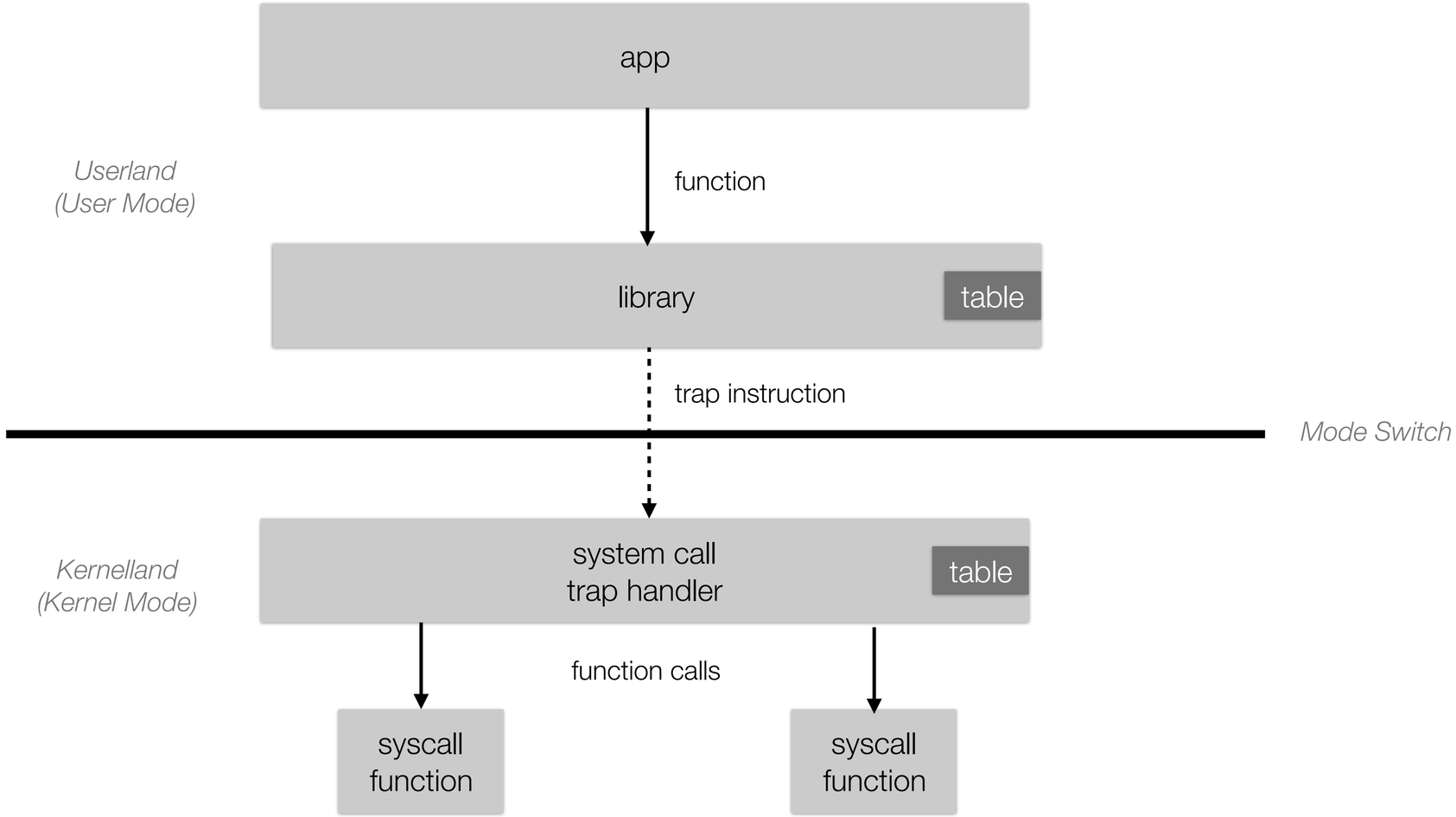


—*Image Credit: Stallings*

Background: Typical Layers of a Computer



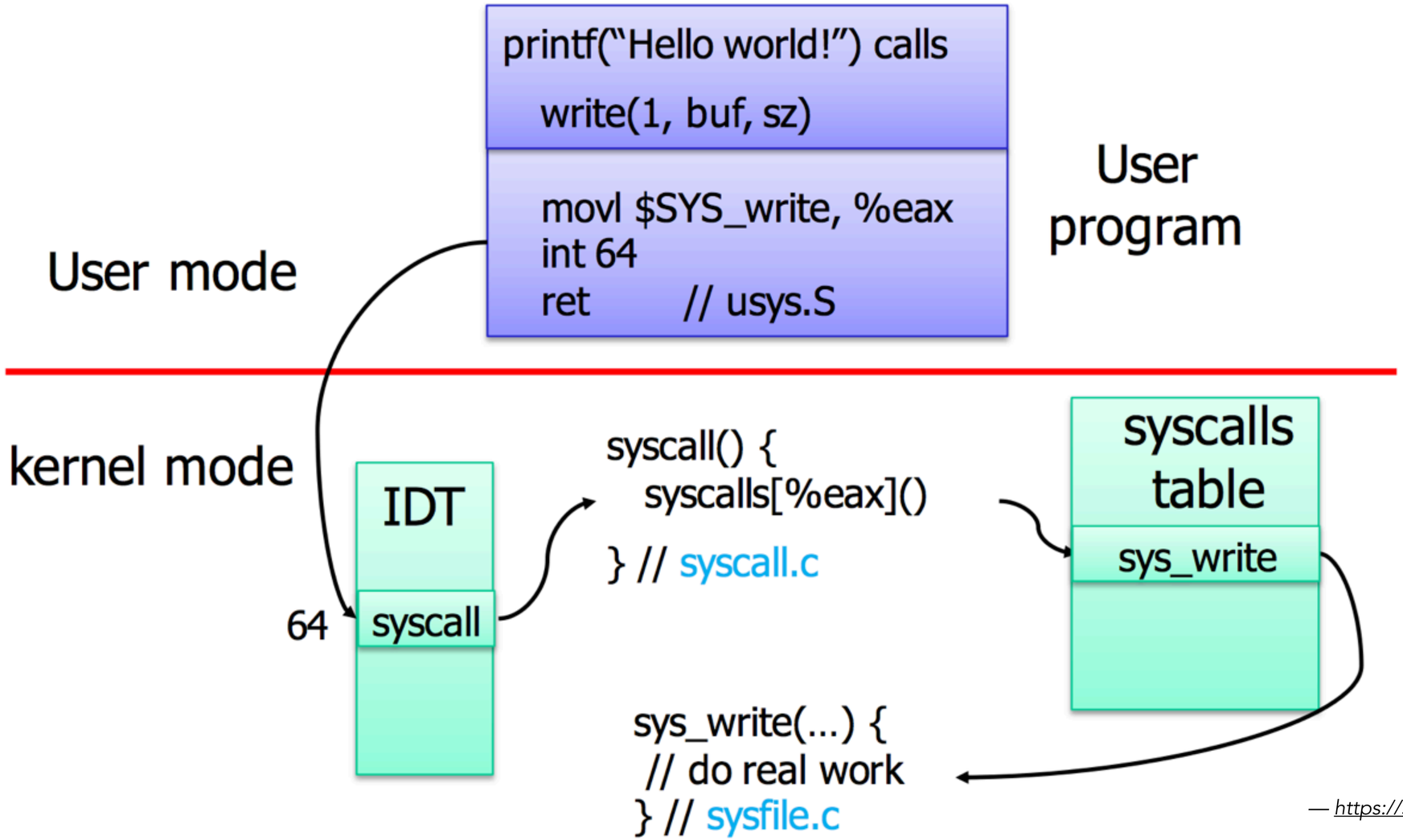
Background: How Apps Use System Resources



Inspired by Figure 4.4 from The Craft of System Security - 1st Edition. Sean Smith (2007).

Background: How Apps Use System Resources

An example



— *https://stackoverflow.com/a/29658740*

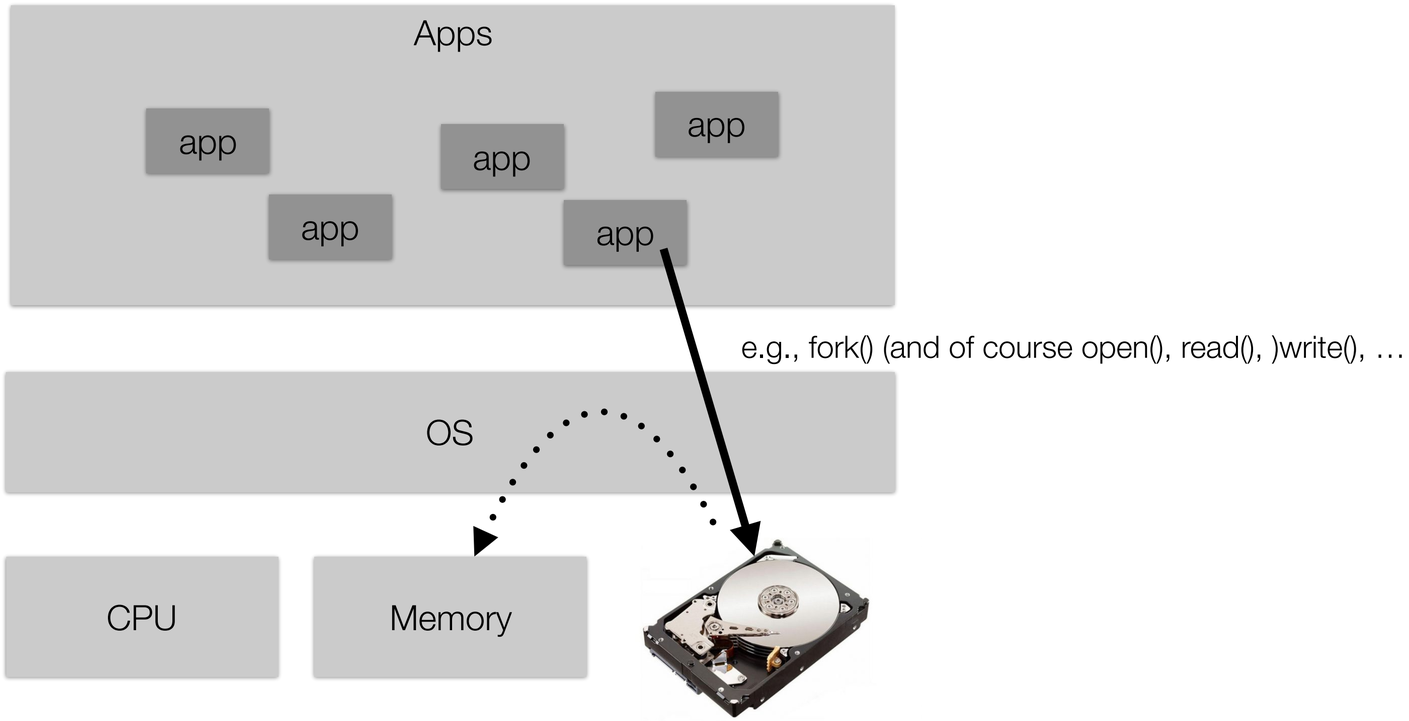
Inspired by Figure 4.4 from The Craft of System Security - 1st Edition. Sean Smith (2007).

Background: An App's Layout in Memory

- How does a program (file) get loaded?

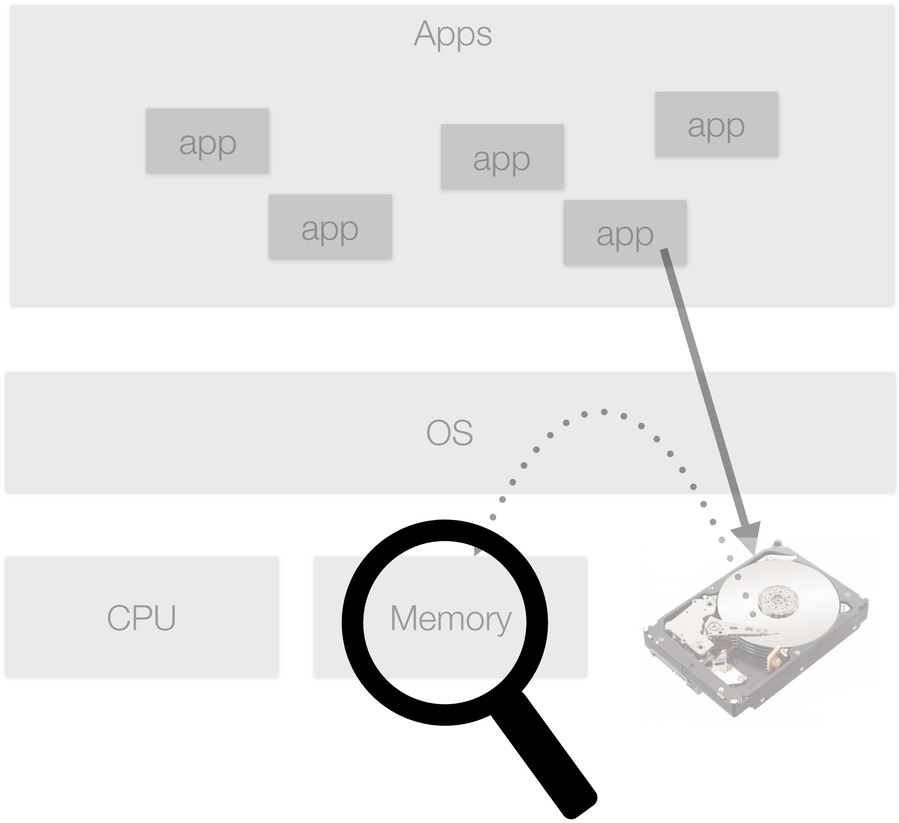
Background: An App's Layout in Memory

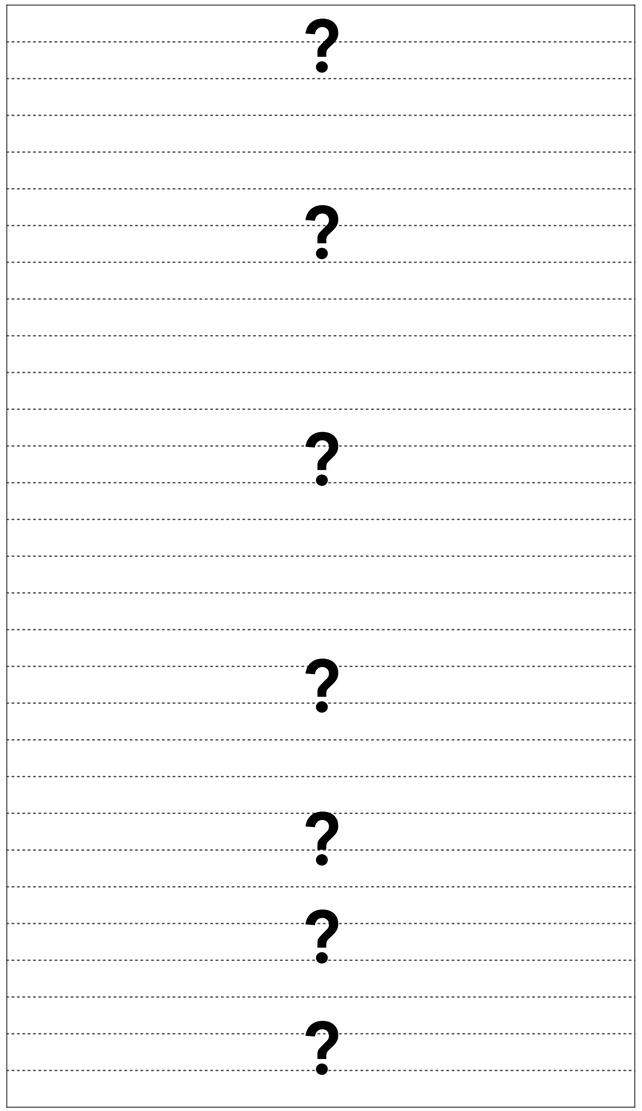
- How does a ***program*** (file) get loaded?



Background: An App's Layout in Memory

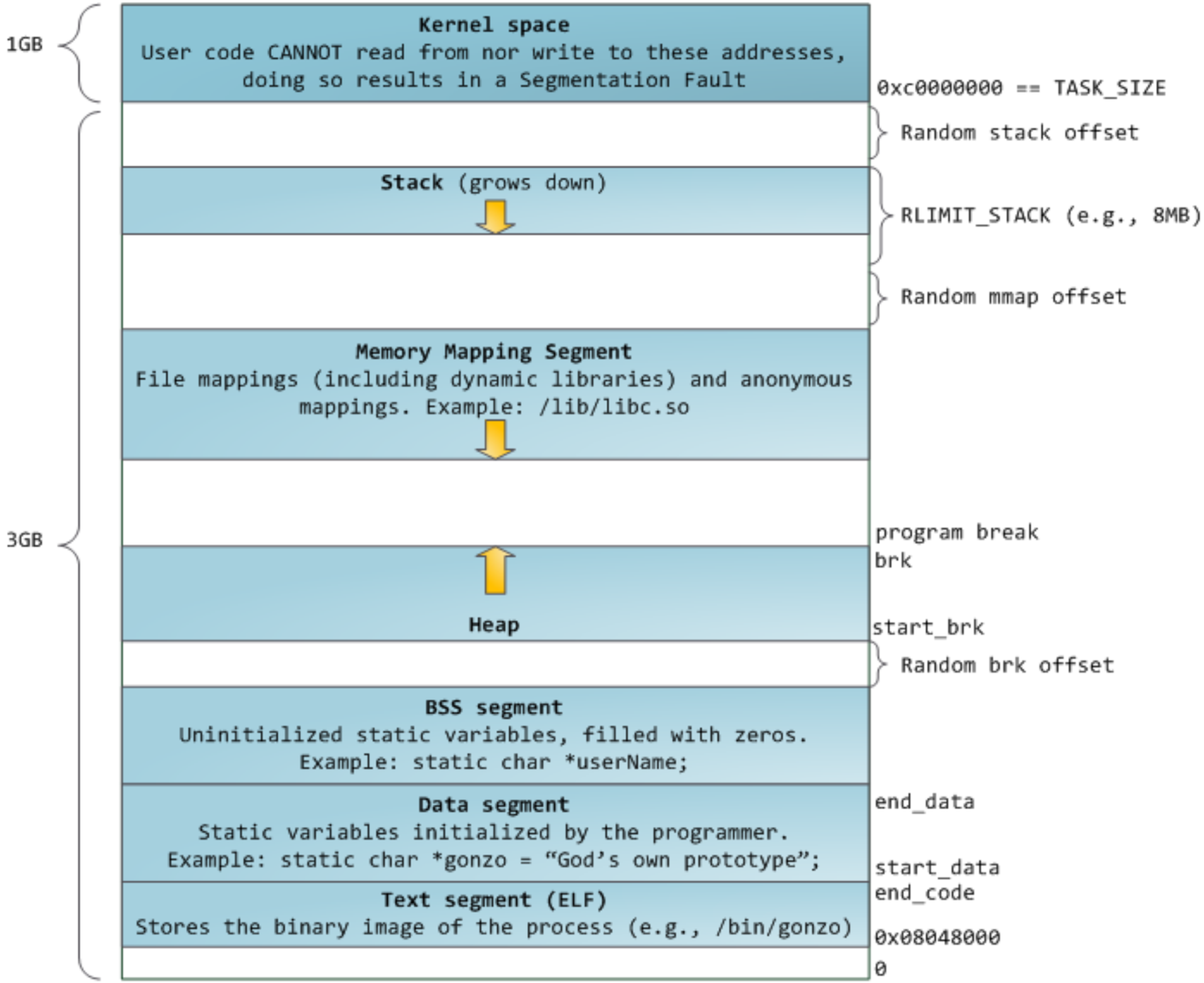
- When a ***program*** is loaded into memory, what does it look like?





Background: An App's Layout in Memory

- When a ***program*** is loaded into memory, what does it look like?



Background: A C Program to Verify Our Thoughts…

Some in class exploration — see ***probe.c***

Some Linux Basics: Users, Groups, Files—*oh my!*

How would you protect your computer & its resources?

How would you protect your computer & its resources?

Ideas?!

How would you protect your computer & its resources?

Modeling and managing system security the UNIX-y way: access control

who can do **what** to **whom**

users/groups

Need notions of "identity"

A human?

Groups of humans?

A service?

An administrator? (e.g., "root")

objects

Usually *things* on a *filesystem*

(e.g., programs, data)

permissions (read/write/execute)

*OK, I know the who/whom—what are **you** permitted to do?*

Read the file?

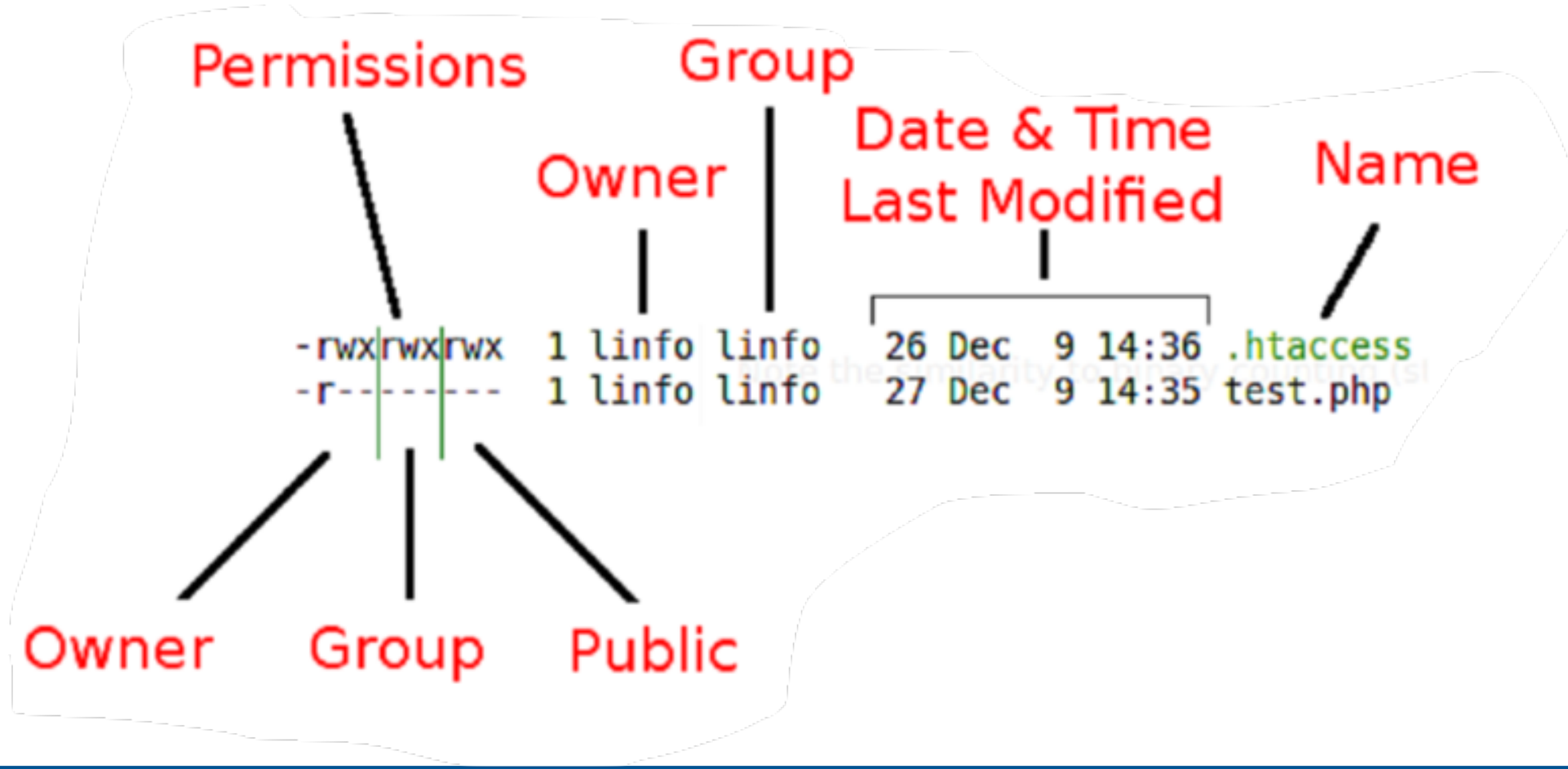
Write to it/modify it?

Run it?

How would you protect your computer & its resources?

Modeling and managing system security the UNIX-y way: access control

Every file has…



A Typical **who** can do **what** to **whom** Flow

If **user A** asks to perform **operation O** on a **file object F**, the OS checks:

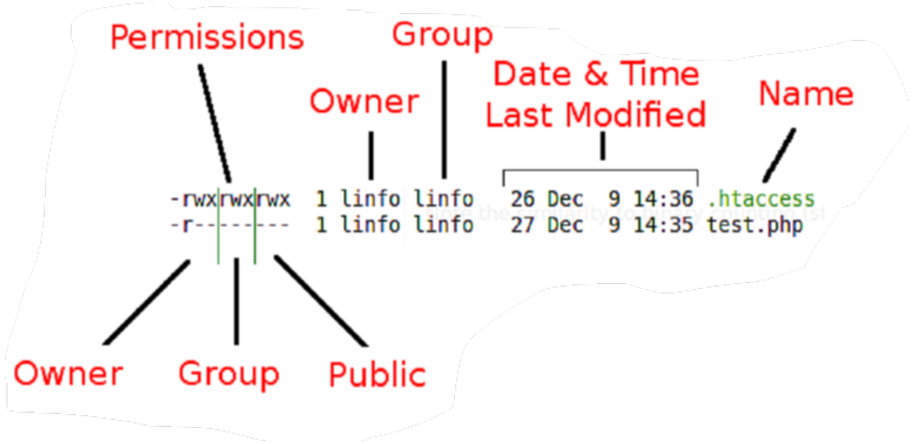
1. Is **A** the owner of **F**? >>> use **owner permissions** to decide whether A can do operation O.

   A is not F's owner

2. Is **A** a member of **F's** group? >>> use **group permissions** to decide…

   A is not F's owner or a member of F's group

3. >>> use the **"everyone else" / "others" permissions** to decide…



Some in class exploration — let's take a quick look at these ideas in a VM.

Except….

*…for this interesting thing known as **Set-UID** and **Set-GID***

- UNIX mechanisms for changing user/group identity
  - **setuid** = set user ID
  - **setgid** = set group ID
  - Enables users to run an executable with the permissions of the executable's owner or group, respectively
- Created to deal with inflexibilities of UNIX access control
  - *Why might this be useful?*
- Also the source of endless security problems…
  - *Why might this be a bad idea?*