



JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

EN.540.635 Software Carpentry

Lecture 1 What is a Computer?

Weekly Programming Challenges:

- Weekly challenges will be available after class on Tuesdays and are normally due by class start time the following Tuesday (Turn in by Thursday midnight for up to 75% credit).
- You are welcome to work together, but turn in your own work.
- Challenges will start off easy and progress as the semester continues.
- You may not have learned everything prior to a challenge – hence the word “Challenge”.

Labs:

- These assignments are to be completed during the class period and will not be handed in for a grade (with a few exceptions).
- Some of the later lab assignments can be handed in for extra credit (this is subject to change).

- **Lazor Project:**

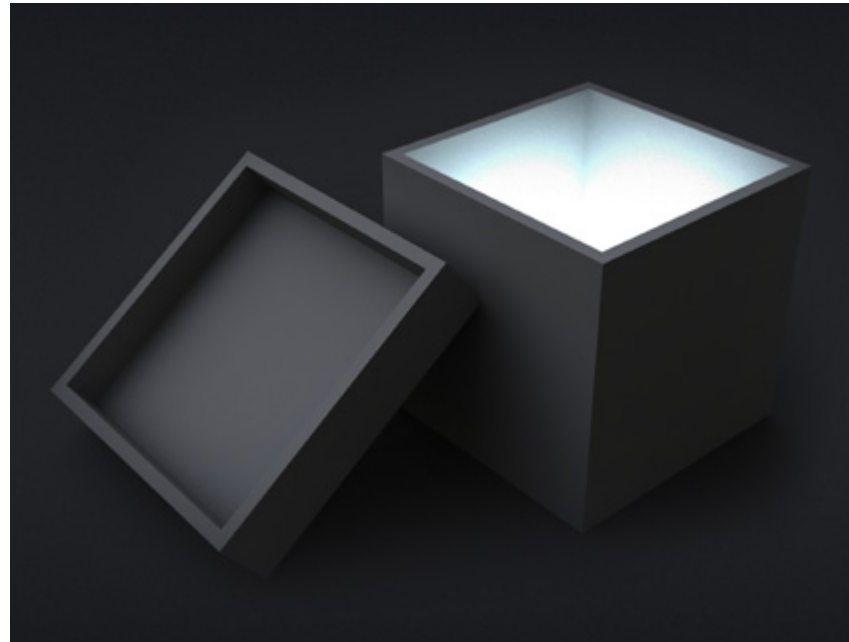
- Group project that covers a lot of the topics we cover in this course.
- Involves the use of Git (version control).
- Learning how to work on a programming project with others.

- **Final Project:**

- Apply what you have learned to a program of your choosing.
- Need to get prior approval from the instructors.
- Should be more robust (require more effort) than the Lazor project.
- Final presentation near the end of the semester.
- Important to make sure that your work is your own!

What is a Computer?

- A magical box.
- A machine that uses ones and zeros to do things.
- It lets me use the internet.



- We count in decimal: 0
- But what if we only knew the numbers 0 and 1?
- Binary System!
- Computers count in this: 0

How do you read 10,353?

$$\begin{aligned} &\text{Ten Thousand, Three Hundred and Fifty Three} \\ &= 1 * 10,000 + 0 * 1,000 + 3 * 100 + 5 * 10 + 3 * 1 \\ &= 1 * 10^4 + 0 * 10^3 + 3 * 10^2 + 5 * 10^1 + 3 * 10^0 \end{aligned}$$

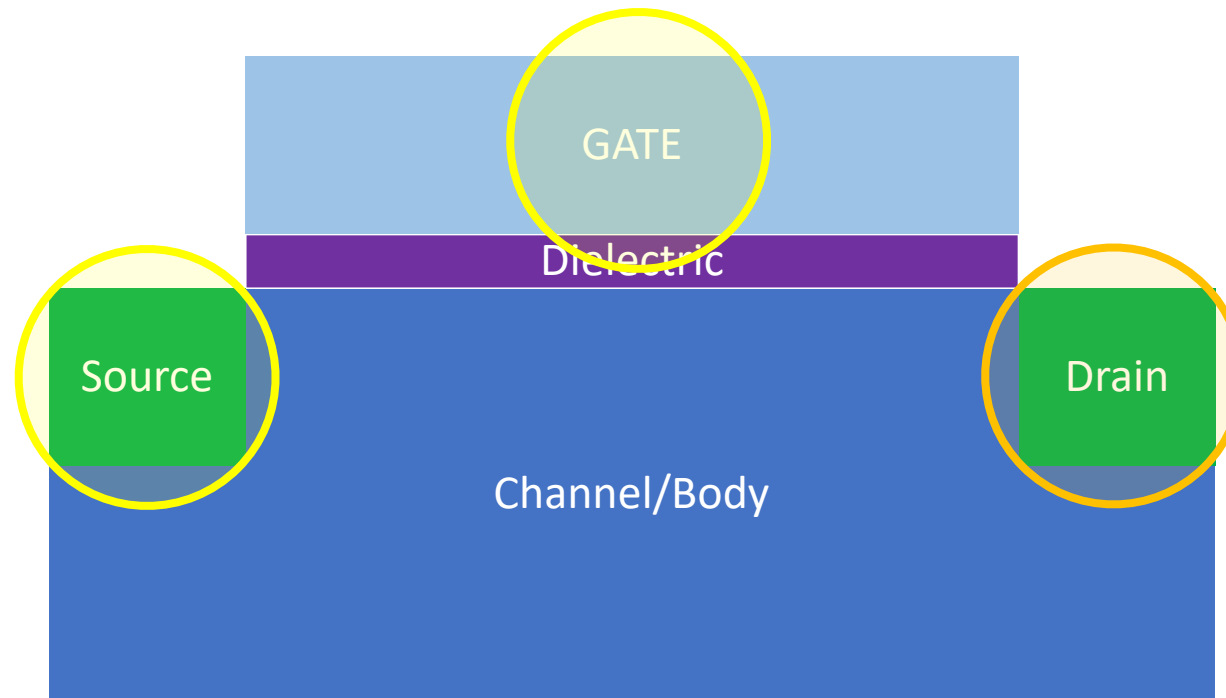
How about 10?

$$\begin{aligned} (10)_{10} &= 1 * 10^1 + 0 * 10^0 \\ (10)_2 &= 1 * 2^1 + 0 * 2^0 = (2)_{10} \end{aligned}$$

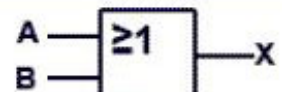
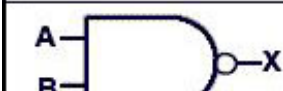
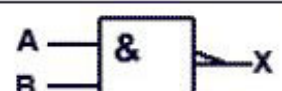
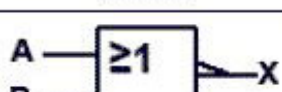
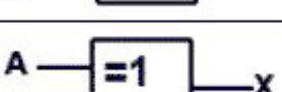
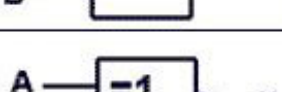
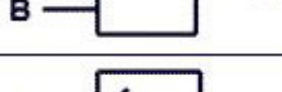
Here, we changed to base-2 notation, where 10 is equivalent to 2.

$$(10101010001)_2 = 1 * 2^{10} + 0 * 2^9 + 1 * 2^8 + \dots + 1 * 2^0 = (1361)_{10}$$

- We need 1's and 0's...
- We need a device...



- We want computers to accomplish basic tasks.
- They must “Think”:
 - **AND** – If two things are true, then do some task
 - **NAND** – If not AND – If two things are both not true at the same time, then do some task
 - **OR** – If at least one of two things are true, then do some task
 - **NOR** – If not OR – If neither of the two things are true, then do some task
 - **XOR** – If exactly only one of two things is true, then do some task

ANSI Symbol	IEC Symbol	NAME
		AND
		OR
		NAND
		NOR
		XOR
		XNOR
		NOT

	1	0
1	1	0
0	0	0

AND – If two things are true, then True

	1	0
1	0	1
0	1	1

NAND – If not AND – If two things are both not true at the same time, then True

	1	0
1	1	1
0	1	0

OR – If at least one of two things are true, then True

	1	0
1	0	0
0	0	1

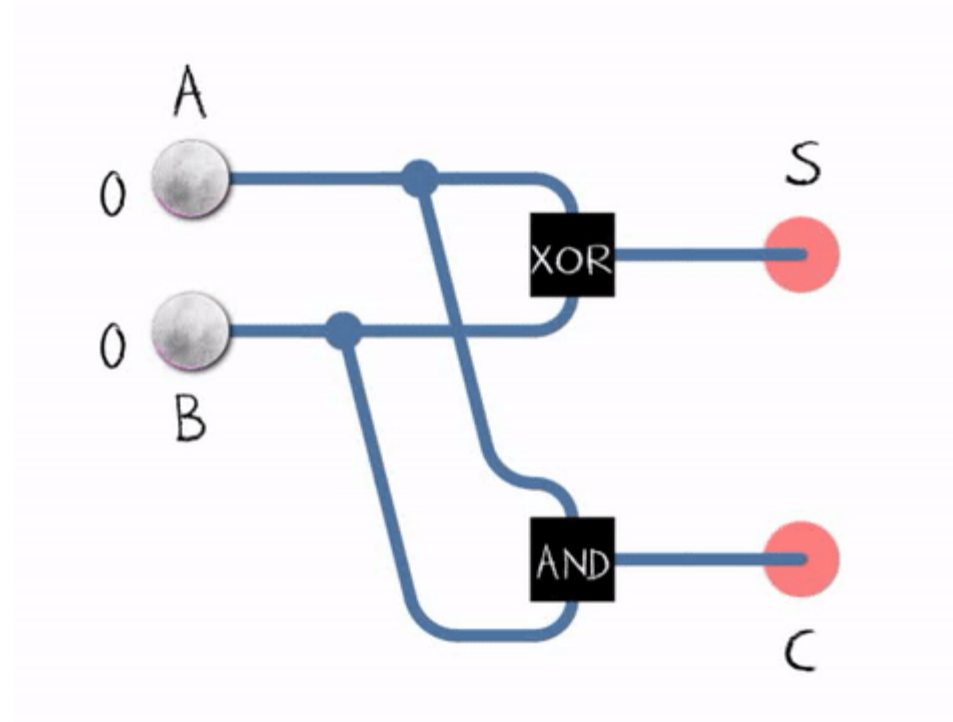
NOR – If not OR – If neither of the two things are true, then True

	1	0
1	0	1
0	1	0

XOR – If exactly only one of two things is true, then True

	1	0
1	AND OR	NAND OR XOR
0	NAND OR XOR	NAND NOR

The Adder



This is a simple example of a subcomponent that can exist inside of a processor.
In reality, processors can be extremely complex.

That Magical Black Box



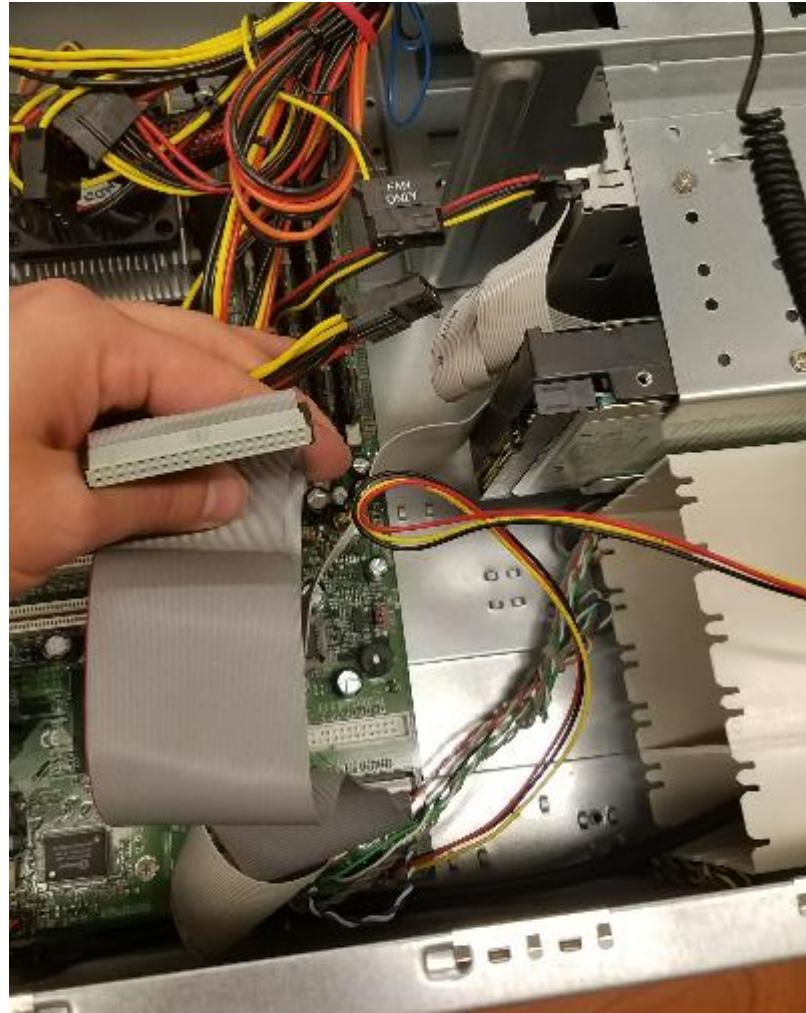
The Case

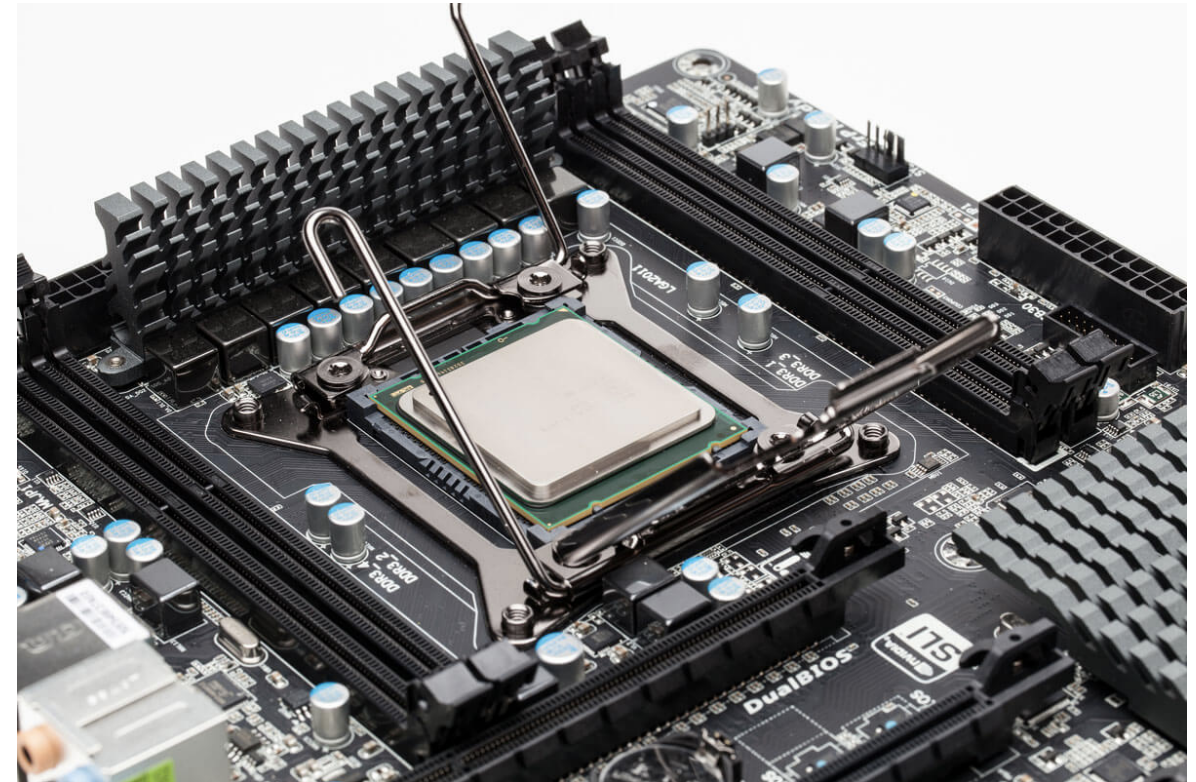


The First Look



Lots of Cables





CPU Cooler



Motherboard



GPU and RAM



The Front Panel



Power Supply



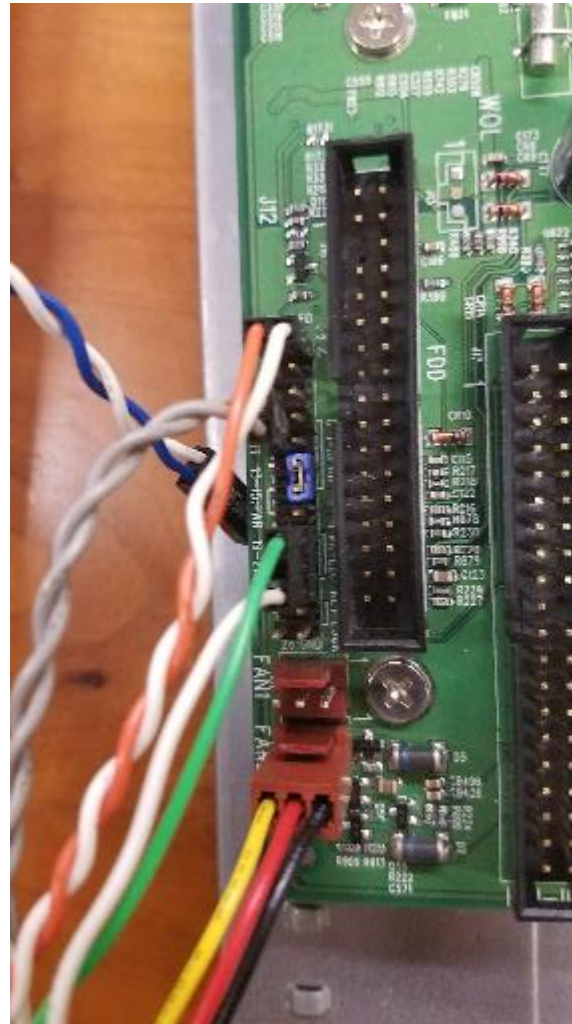
Hard Drive and Optical Drive



Putting it Back Together



Putting it Back Together



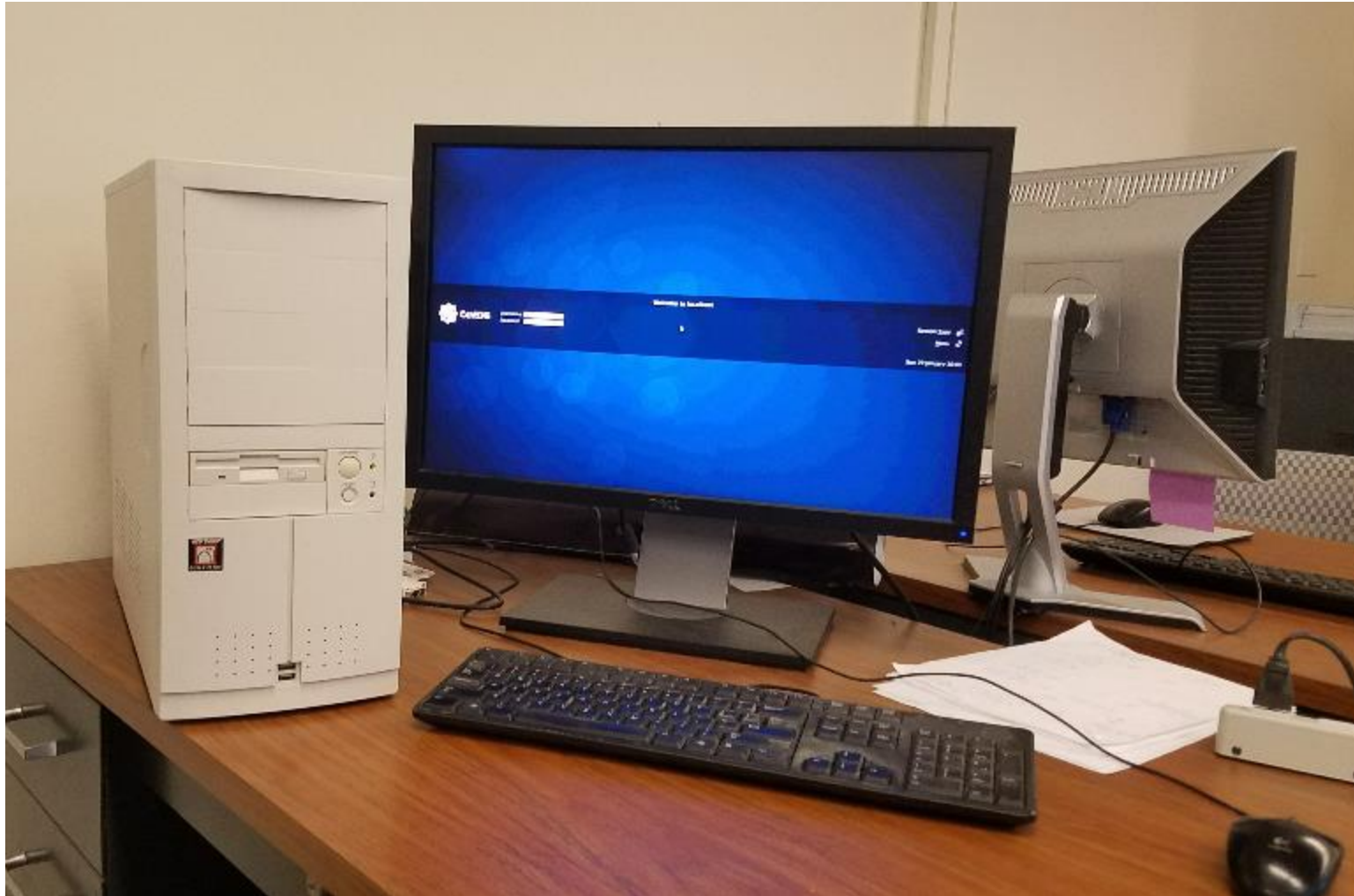
Putting it Back Together



Putting it Back Together



Putting it Back Together



There are two things you need:

1. Text editor

○ Simple text editors:

- Notepad/TextEdit
- Vim/Nano
- **Sublime Text** (text editor & IDE)
- Jupyter Notebook (web-based; do not turn in Jupyter notebooks!)

○ Integrated Development Environments (IDEs):

- **PyCharm**
- Spyder (comes with Anaconda; “input” function is buggy)
- VS Code
- Eclipse + PyDev

There are three things you need:

2. Command Line Interface

○ Windows:

- **Command Prompt**
- Windows Terminal (currently in beta) or Windows PowerShell
- Windows Subsystem for Linux (WSL)
- **Git Bash**
- Anaconda PowerShell Prompt

○ macOS/Linux:

- **Terminal**
- **iTerm**

3. Python (note: *conda* is a command line tool / python package, like *pip*)

Anaconda (Python + conda + meta package):

- Open-source distribution of Python geared towards scientific computing/data science.
- Over 1500 scientific packages auto installed at once
- Needs more time to install & disk space
- Make sure you download for the correct OS that you are using
 - Go to <https://www.anaconda.com/download>
 - Macs make sure to check for CPU vs M1/M2.

Miniconda (Python + conda):

- Open-source distribution of Python geared towards scientific computing/data science.
- Must install each packaged you want individually
- Smaller & more lightweight
- Make sure you download the Python 3.10 version for the correct OS that you are using.
 - Go to <https://docs.conda.io/en/latest/miniconda.html>
 - [Miniconda3 macOS Apple M1 64-bit pkg](#) for M1 Macs for example