Installation Variations

…or, stuff that installers of server/command-line software may or may not do differently

- These are difficult to pre-document in a rigorous step-by-step manner because:
  - Not all installers may do them
  - Ways to address them may vary based on platform
- They do fall under certain patterns, and that's what we'll seek to address here

- Some of these installation variations presume understanding of certain underlying concepts:
  - Environment variables
  - The file system
  - Process management
  - …and many more
- We'll try to cover then as much as possible, but there will always be more to know
- You can use the terminology and ideas given here for web searches, and these searches can lead to more in-depth information

The PATH Variable

- All software has access to environment variables—information provided externally which programs can look up internally
- The command line is a program like any other (if you're lucky you'll learn how to write one of your own one day) and as such it relies these as well
- One of the most important of these is PATH: this variable lists all of the places where a command line or terminal can potentially find a program

- If a program's location isn't listed in PATH, then invoking it from the command line will require specifying its location in full—so PATH isn't an absolute requirement but it's certainly convenient
- The exact format for PATH varies based on the platform, but it's generally a colon- or semicolon-separated list of absolute folder locations
- The main point is this: server software almost always comes with command-line executables and so will benefit from having their location added to the PATH variable—some installers do this for you and some installers don't. So for the latter, you'll want to learn how to edit the PATH variable yourself

Paths to PATH

As mentioned, platforms differ with respect to user interface and file layout, so no single set of instructions can cover "how to set the PATH variable"—however, there are some common patterns

First off, it is possible to set the PATH variable for a specific command line session only. This is useful when just testing to see if you have the right value

Exact syntax varies by platform, but generally looks like:

PATH=(new software location):(current value of PATH)

Variations on the above include the delimiter—it is `:` most of the time but in Windows it is `;`. The current value is written as `$PATH` in Unix-like systems while on Windows it is `%PATH%`

Making this permanent may vary as well:

- Unix-like environments expect a file edit (`bash` typically reads .bashrc; `zsh` typically reads .zshrc) to append the location of your executables to `PATH`—just add the `PATH`-setting statement above to that file, and it should now execute with every session
- In Windows, there is an environment variables setting/control panel where you can update `PATH`—find it and add the software location to the list

Server Autostart

Some server installers will both install the software and start it automatically

- This is convenient for pure-usage situations, but in our case we also want to get direct control of this so we will say a polite "No thanks" to this courtesy
- An autostarted server also means that a default setup is established—again something that we prefer not to avail of in our context, because this is precisely what we are seeking to learn

As with `PATH`, there's no single way to deactivate an autostarted server. Some things to try:

- Some systems have a "services" capability that manages these kinds of processes. Homebrew has a `brew services` command; Ubuntu Linux has something called `systemctl`
- Windows has a Services control panel that tells you whether something is running and lets you start/stop it as well as set whether it starts automatically

As a last resort, you can use your device's process manager (Activity Monitor on macOS, Task Manager on Windows, etc.) to quit/kill a process

Server Users and Data

In autostarting a server, installers may also create a specific user account under which to run that server as well as default database files against which that server runs

- Without a server process to use them, these are fairly innocuous and can sit dormant on your device
- If their unused presence bothers you (or perhaps you want to claim every ounce of disk space possible), you can delete these with whatever tools your platform provides for managing users and files

If you need these back at some future time (maybe you decide to run a database server as part of a project long after the class is over), recreating them with the same names, paths, and permissions should suffice

- Or you can reinstall and get them back that way…you might need to anyway as new versions are released
- The takeaway from this class is that these special users or default files are peripheral to the core functionality of server programs—in the end, they are just programs and you can always run them under your account as long as you know what they need

They may be best practices and conveniences, but not strictly essential for running these things