CIS241

System-Level Programming and Utilities

bash Basics

Erik Fredericks, frederer@gvsu.edu

Fall 2025

> **Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers**

What is a bash script?

Easy mode: a series of bash commands run in order

- Or, everything you've done so far!

However, bash is a full programming language!

- Control structures (loops, conditionals)
- Variables
- Functions
- Arguments (parameters)
- Arrays
- ...

But why?

- Automate all the things!
  - Way more efficient to script out repetitive tasks!
  - Even better if you can customize them

Examples!

- Sysadmins checking server status and running regular updates
  - Or auto-creating accounts and giving permissions
- Build and deploy personal website!
  - All the `git` actions for building/deploying, for example

OK LET'S GO

Basically, we're going to put all the bash commands we know *into a file*

- And then running it!

Convention is to use a `.sh` extension

- Run it with `bash file.sh`

Make it executable?

- `chmod u+x filename`
- `chmod +x filename`
- `./filename`

The pound-bang

Or hashbang, shebang, hashpling

(The top line telling your script where to find `bash`)

```
#!/bin/bash
```

- Q: Why bother adding it? It isn't *strictly* required...
  - A: Portability! Users don't need to necessarily know what to use to call the script
  - Also, easily run bash scripts from other shells!

Examples

- `#!/bin/bash`
- `#!/usr/bin/env bash`

Can use with others like Python:

- `#!/usr/bin/env python3`

`/usr/bin/env` vs. `/bin/bash`

- `env` uses whatever version of the executable comes first in `$PATH`
  - `env` - users can have different behavior!

Mac users (from Carrier)

- Use `brew install bash` to get updated version
- Different path from previous slide!
- Hers was: `#!/usr/local/bin/bash`

Comments!

`#` **begins a comment until the end of the line**

Exception:

- Hashbang on first line of script

Ex:

```
#!/bin/bash

echo  "Hello there"  # this is also a comment
# this is a comment
```