Bash

Networking, Debugging, Misc

wget

- `wget` is a non-interactive downloader
  - Only supports HTTP, HTTPS, and FTP
  - Depending on the website you are downloading from, may support continuing a paused download
  - Also has options to create an entire local copy of a website

```
wget [OPTIONS] URL
```

In [ ]:
```
wget www.umbc.edu
```

In [ ]:
```
head index.html.1
```

Common Courtesy with `wget`

- `wget`, and many other command line tools can theoretically launch 100s of request a second
  - This is mean, potentially illegal, and a good way to get your IP blocked
- `wget` has many options to prevent this if you are downloading multiple files at once
  - --limit-rate sets a maximum bandwidth to use
  - --wait sets the number of seconds to wait between each request
  - --random-wait will jitter the amount of time the wait actually is

In [ ]:

```
wget --mirror --page-requisites \
--convert-links --adjust-extension  \
-P./local_443-2 --wait 1 --random-wait \
https://www.csee.umbc.edu/~bwilk1/433/
```

More Useful `wget` Features

- `wget` allows you specify a list of urls to download by using the -i flag
- The type of files downloaded can be controlled by the following flags
  - --accept takes a comma separated list of file endings to accept
  - --reject takes a comma separated list of file endings to reject

Real-World Example

- As a computational linguist, one of the most important steps in research is to gather data
- In this example, pretend we want to build a dataset of text found on academic websites
- The steps we will take are:
    1. Get a list of URLs from a website
    2. Extract the URLS
    3. Use `wget` to download the websites
    4. Use `sed` and other tools to strip the text out from the website

In [ ]:
```
# Get lsit of addresses from "https://univ.cc/search.php?dom=edu&key=&start=1"
```

In [ ]:
```
# Extract the URLS
```

In [ ]:
```
mapfile sites_to_get < targets
```

In [ ]:
```
# Process files using wget and sed

# Get URL

# Get School Name
```

In [ ]:
```
cat Abilene_Christian_University.txt
```

`curl`

- `curl` is a more powerful tool that allows uploading and download over
  - (S)FTP
  - HTTP(S)
  - SCP
  - LDAP
- `curl` prints to STDOUT

In [ ]:
```
curl http://www.umbc.edu
```

In [ ]:
```
curl -I http://www.umbc.edu
```

POST requests

- We will look at HTTP requests more in detail in a few weeks
- IF you submit something in a form and don't see a crazy web address, it was probably submitted using POST
- `curl` allows POST by using the `-X` flag

```
curl -X POST -d "DATA" URL
```

Debugging in `bash`

- The bash command itself has several flags are are useful in debugging
- The flags are included as part of the shebang line

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

- The main flags for debugging are
  - -n Step the through the script but do not running, good for finding syntax errors
  - -x Prints traces of commands and their arguments

In [ ]:
```
cat src/shell/syntax_example.sh
```

In [ ]:
```
./src/shell/syntax_example.sh
```

In [ ]:
```
cat src/shell/syntax_error_example.sh
```

In [ ]:
```
./src/shell/syntax_error_example.sh
```

In [ ]:
```
cat ./src/shell/cla_debug.sh
```

In [ ]:
```
./src/shell/cla_debug.sh Arg1 SOmething goes here
```