Exercises

Create a new directory called `missing` under `/tmp`.

```
~$ mkdir /tmp/missing
~$ ls /tmp
config-err-zlkr8v
hsperfdata_root
missing
```

Look up the `touch` program. The `man` program is your friend.

```
~$ man touch
```

Use `touch` to create a new file called `semester` in `missing`.

```
~$ touch /tmp/missing/semester
~$ ls /tmp/missing
total 0
-rw-r--r-- 1 ramzel ramzel 0 Apr 27 16:15 semester
```

Write the following into that file, one line at a time:

```
#!/bin/sh
curl --head --silent https://missing.csail.mit.edu
```

```
~$ cd /tmp/missing
/tmp/missing$ cat > semester
#!/bin/sh
curl --head --silent https://missing.csail.mit.edu
^C
/tmp/missing$ cat semester
#!/bin/sh
curl --head --silent https://missing.csail.mit.edu
```

or

```
~$ cd /tmp/missing
/tmp/missing$ echo '#!/bin/sh' > semester
```

```
/tmp/missing$ echo "curl --head --silent https://missing.csail.mit.edu" >>
semester
/tmp/missing$ cat semester
#!/bin/sh
curl --head --silent https://missing.csail.mit.edu
```

Try to execute the file, i.e. type the path to the script (`./semester`) into your shell and press enter. Understand why it doesn't work by consulting the output of `ls` (hint: look at the permission bits of the file).

```
/tmp/missing$ ./semester
bash: ./semester: Permission denied
/tmp/missing$ ls -l semester
-rw-r--r-- 1 ramzel ramzel 61 Apr 27 16:40 semester
```

Permission denied. When executing a script using its path, the shell first checks for the file's executable permissions. Executing the file didn't work because the user doesn't have permission to execute (denoted by '`x`').

Run the command by explicitly starting the `sh` interpreter, and giving it the file `semester` as the first argument, i.e. `sh semester`. Why does this work, while `./semester` didn't?

```
/tmp/missing$ sh semester
HTTP/2 200
server: GitHub.com
content-type: text/html; charset=utf-8
last-modified: Tue, 20 Apr 2021 14:15:31 GMT
...
```

It works! Using the `sh` command interprets the script even without executable permissions (as long as the file is readable).

Use `chmod` to make it possible to run the command `./semester` rather than having to type `sh semester`. How does your shell know that the file is supposed to be interpreted using `sh`? See this page on the shebang line for more information.

```
/tmp/missing$ chmod +x semester
/tmp/missing$ ls -l semester
-rwxr-xr-x 1 ramzel ramzel 61 Apr 27 16:40 semester
```

```
/tmp/missing$ ./semester
HTTP/2 200
server: GitHub.com
content-type: text/html; charset=utf-8
last-modified: Tue, 20 Apr 2021 14:15:31 GMT
...
```

Now, the file executes even without the `sh` *command. Using* `chmod +x semester` *gives the file executable permissions. When executing* `./semester`*, the shell checks and recognizes the file's executable permissions and executes it.*

Use `|` *and* `>` *to write the "last modified" date output by* `semester` *into a file called* `last-modified.txt` *in your home directory.*

```
/tmp/missing$ ./semester | grep -i last-modified | cut --delimiter=',' -f2
 20 Apr 2021 14:15:31 GMT
/tmp/missing$ ./semester | grep -i last-modified | cut --delimiter=',' -f2
| xargs > ~/last-modified.txt
/tmp/missing$ cat ~/last-modified.txt
20 Apr 2021 14:15:31 GMT
```

Write a command that reads out your laptop battery's power level or your desktop machine's CPU temperature from `/sys`.

```
~$ ls /sys/
block  bus  class  dev  devices  firmware  fs  hypervisor  kernel  module
power
~$ cd /sys/class/power_supply/BAT0
/sys/class/power_supply/BAT0$ ls
alarm           charge_full_design  device        power          subsystem
voltage_min_design
capacity        charge_now          hwmon2        present        technology
voltage_now
capacity_level  current_now         manufacturer  serial_number  type
charge_full     cycle_count         model_name    status         uevent
/sys/class/power_supply/BAT0$ cat capacity
42
~$ cat /sys/class/power_supply/BAT0/capacity
41
```