

Exercises

Read `man ls` and write an `ls` command that lists files in the following manner:

- Includes all files, including hidden files
- Sizes are listed in human readable format (e.g. 454M instead of 454279954)
- Files are ordered by recency
- Output is colored

Used flags: -l for list listing format, -a to include hidden files, -h for human-readable format

Used option: --color (omitted since most Linux distros these days has ls colors enabled)

```
$ man ls
$ ls -lath
total 15M
-rw-rw-r-- 1 ramzel ramzel 15 Apr 28 13:59 .hidden-file.txt
drwxr-xr-x 4 ramzel ramzel 4.0K Apr 28 13:58 .
drwxr-xr-x 72 ramzel ramzel 12K Apr 28 09:47 ..
-rw-rw-r-- 1 ramzel ramzel 1.7M Oct 23 2020
PHSiklab+Day+--+Lab+--+S3+.pdf
-rw-rw-r-- 1 ramzel ramzel 811K Oct 23 2020
PHSiklab+Day+--+Lab+--+EC2.pdf
-rw-rw-r-- 1 ramzel ramzel 7.4M Oct 23 2020
'Silkab+Day+1+--+Afternoon+(Databases).pdf'
-rw-rw-r-- 1 ramzel ramzel 4.9M Oct 23 2020
Siklab+Day+1+--+Morning.pdf
drwxr-xr-x 2 ramzel ramzel 4.0K Oct 23 2020 'Day 2 - AWS Jam Day'
drwxr-xr-x 3 ramzel ramzel 4.0K Oct 23 2020 'Day 1 - Lecture and
Lab'
```

Write bash functions `marco` and `polo` that do the following. Whenever you execute `marco` the current working directory should be saved in some manner, then when you execute `polo`, no matter what directory you are in,

`polo` should `cd` you back to the directory where you executed `marco`. For ease of debugging you can write the code in a file `marco.sh` and (re)load the definitions to your shell by executing `source marco.sh`.

Creating file `marco.sh` with `cat` command and writing to it

```
/tmp/missing$ cat > marco.sh
#!/bin/bash

marco () {
    marco_wd=$(pwd)
}
^C
```

Loading the `marco` function to the shell then executing it

```
/tmp/missing$ source marco.sh
/tmp/missing$ marco
```

Creating file `polo.sh` with `cat` command and adding content

```
~/Documents$ cat > polo.sh
#!/bin/bash

polo () {
    cd "$marco_wd"
}
^C
```

Loading `polo` to the shell then executing it from another directory

```
~/Documents$ source polo.sh
~/Documents/Virtual-Swag-Kit$ polo
/tmp/missing$
```

Executing `polo` changed the directory to where `marco` was executed

Say you have a command that fails rarely. In order to debug it you need to capture its output but it can be time consuming to get a failure run. Write a bash script that runs the following script until it fails and captures its standard output and error streams to files and prints everything at the end. Bonus points if you can also report how many runs it took for the script to fail.

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

n=$(( RANDOM % 100 ))

if [[ n -eq 42 ]]; then
    echo "Something went wrong"
    >&2 echo "The error was using magic numbers"
    exit 1
fi

echo "Everything went according to plan"
```

Creating a file for the rare_fail script

```
$ cat > rare_fail.sh
#!/usr/bin/env bash

n=$(( RANDOM % 100 ))

if [[ n -eq 42 ]]; then
    echo "Something went wrong"
    >&2 echo "The error was using magic numbers"
    exit 1
fi

echo "Everything went according to plan"
^C
```

Adding execute permissions to rare_fail.sh

```
$ chmod +x rare_fail.sh
```

Writing a script for debugging `rare_fail.sh`

```
$ cat > catch_fail.sh
#!/bin/bash

n_runs=1

./rare_fail.sh > rare_fail_stdout.log 2> rare_fail_stderr.log

while [[ "$?" -ne 1 ]]; do
    (( n_runs++ ))
    ./rare_fail.sh >> rare_fail_stdout.log 2>> rare_fail_stderr.log
done

echo "Script ran for $n_runs times before failing."
^C
```

Adding executable permissions to `catch_fail.sh` then executing it

```
$ chmod +x catch_fail.sh
$ ./catch_fail.sh
Script ran for 51 times before failing.
$ ls
catch_fail.sh  rare_fail.sh  rare_fail_stderr.log
rare_fail_stdout.log
```

Your task is to write a command that recursively finds all HTML files in the folder and makes a zip with them. Note that your command should work even if the files have spaces (hint: check `-d` flag for `xargs`).

```
$ ls -R *.htm? | xargs -d '\n' tar -cvf html_files.tar
aacxs.html
AdDisplayTrackerServlet.html
adfeth.html
$ ls -t | head -n3
html_files.tar
show ad.html
aacxs.html
```

(Advanced) Write a command or script to recursively find the most recently modified file in a directory. More generally, can you list all files by recency?

To find the most recently modified file in a directory:

```
$ find . -type f | xargs -d '\n' ls -th | head -n1
./files received/PUP-Internship-MOA_Loto,RamzelRenz.pdf
```

Sorting files by recently modified:

```
$ find . -type f | xargs -d '\n' ls -lth
-rw-rw-r-- 1 ramzel ramzel 74K Apr 28 17:10 ./files
received/PUP-Internship-MOA_Loto,RamzelRenz.pdf
-rw-rw-r-- 1 ramzel ramzel 202K Apr 28 12:08 ./files received/MSI-
008-2021 (1).pdf
-rw-rw-r-- 1 ramzel ramzel 8.3K Apr 28 12:07 ./2104.txt
-rwxrwxrwx 1 ramzel ramzel 2.8K Mar 22 06:44 ./dailyroutine.txt
...
-rw-rw-r-- 1 ramzel ramzel 4.6K May 31 2020 ./2004.txt
-rw-rw-r-- 1 ramzel ramzel 5.2K Dec 22 2019 './topik review.txt'
```