Working with files and directories

If you need help at any time, put your **red** sticky note on the back of your laptop. When you've finished the steps on the *front* of this page, put your **green** sticky note on the back of your laptop. Then, you can turn the page over and try out some of the more advanced tricks on the back while you wait for the rest of the group to be ready.

Commands in this lesson: `cp`, `rm`, `mv`, `cat`, `wget`, `scp`, `nano`

Moving files around the local filesystem

The easiest way to create a file is to just open it for editing. We will use the `nano` text editor to open file called `newfile.txt`:

```
nano newfile.txt
```

You can type some text into this file, then use Ctrl + O to write it **o**ut to file, and hit Enter to confirm the file name to which to save. Near the bottom of the screen, it should say e.g. "[Wrote 1 line]". Then use Ctrl + X to exit.

To see the contents of a file, we can print the contents of the file to the terminal output with `cat`:

```
cat newfile.txt
```

To copy a file, we use `cp`, and give the source and destination file names as arguments:

```
cp newfile.txt copy.txt
```

To move (or rename) a file, we use the `mv` command:

```
mv copy.txt mycopy.txt
```

and we use `rm` to delete a file:

```
rm mycopy.txt
```

With `rm`, there is no "Recycle Bin" and no getting back files you've deleted accidentally - so be very, very careful.

Moving files over the Internet

We will often have to move files over the Internet - for example, get a file from the Internet onto our local filesystem, or copy a file from a remote system that we access with SSH to our local filesystem.

Use `wget` to download from the Internet. For example, to download a file I've put at https://witestlab.poly.edu/bikes/README.txt we can run

```
wget https://witestlab.poly.edu/bikes/README.txt
```

Use

```
cat README.txt
```

to verify that you've retrieved the file and see its contents. Similarly, you can download anything from the web by URL.

Useful flags

Bash utilities typically have some flags you can use to modify the way they behave, or what their output looks like.

For example, take the `ls` command. We can:

- See one file per output line: `ls -1`
- See "long" output that includes file permissions, ownership, modification dates: `ls -l`
- See "long" output and also sort files in order of time of last modification: `ls -lt`
- See "long" output and sort files so that the most recently modified file is last: `ls -ltr`

With most utilities, you can use the `--help` flag to find out how to use the utility and what flags are available for it:

```
ls --help
```

Using `scp`

To move files back and forth between your laptop and a remote system that you access with `ssh`, we can use `scp`. The syntax is:

```
scp source destination
```

When using `scp`, you have to pay attention to *where* you are running a command. Assuming you have a file `README.txt` located in your home directory on our workshop server, you can run

```
scp NETID@server.camp.ch-geni-net.instageni.nysernet.org:~/README.txt .
```

(substituting your own NETID) from a shell on your *laptop* to retrieve that file from the server and copy it to whatever directory you are working in on your laptop. (The `.` shortcut indicates "put the file *here*".) Note that you'll have to make sure you have the necessary file permissions to write files to the directory you are working in!

You can then make changes to the file locally and copy it back to your home directory on the server, with

```
scp README.txt NETID@server.camp.ch-geni-net.instageni.nysernet.org:~/README2.txt
```

Online sharing from the command line

Sometimes we'll want to do the "reverse" of `wget` - upload a file to the Internet, using the Linux shell. There are several online services that provide an API for this. For example, to upload the "bikes" `README.txt` you downloaded earlier, you can run

```
curl --upload-file ./README.txt https://transfer.sh/README.txt
```

This will return a URL, at which you can see and download the file you've just uploaded.