CS 261 IT Basics

Connecting to a remote machine using SSH and the basics of terminal-based text editors

SSH in a Nutshell

SSH is a program for connecting to a remote machine and running commands there.

More about SSH

- Each command you type during an SSH session is run on the remote machine you’re connected to.
- All of the files available within an SSH session live on the remote machine you’re connected to.

More about SSH

For example, if you use SSH to connect to `flip.engr.oregonstate.edu`, you will be running commands on that machine and looking at files on that machine.

- You'll need to be comfortable doing exactly this for CS 261.

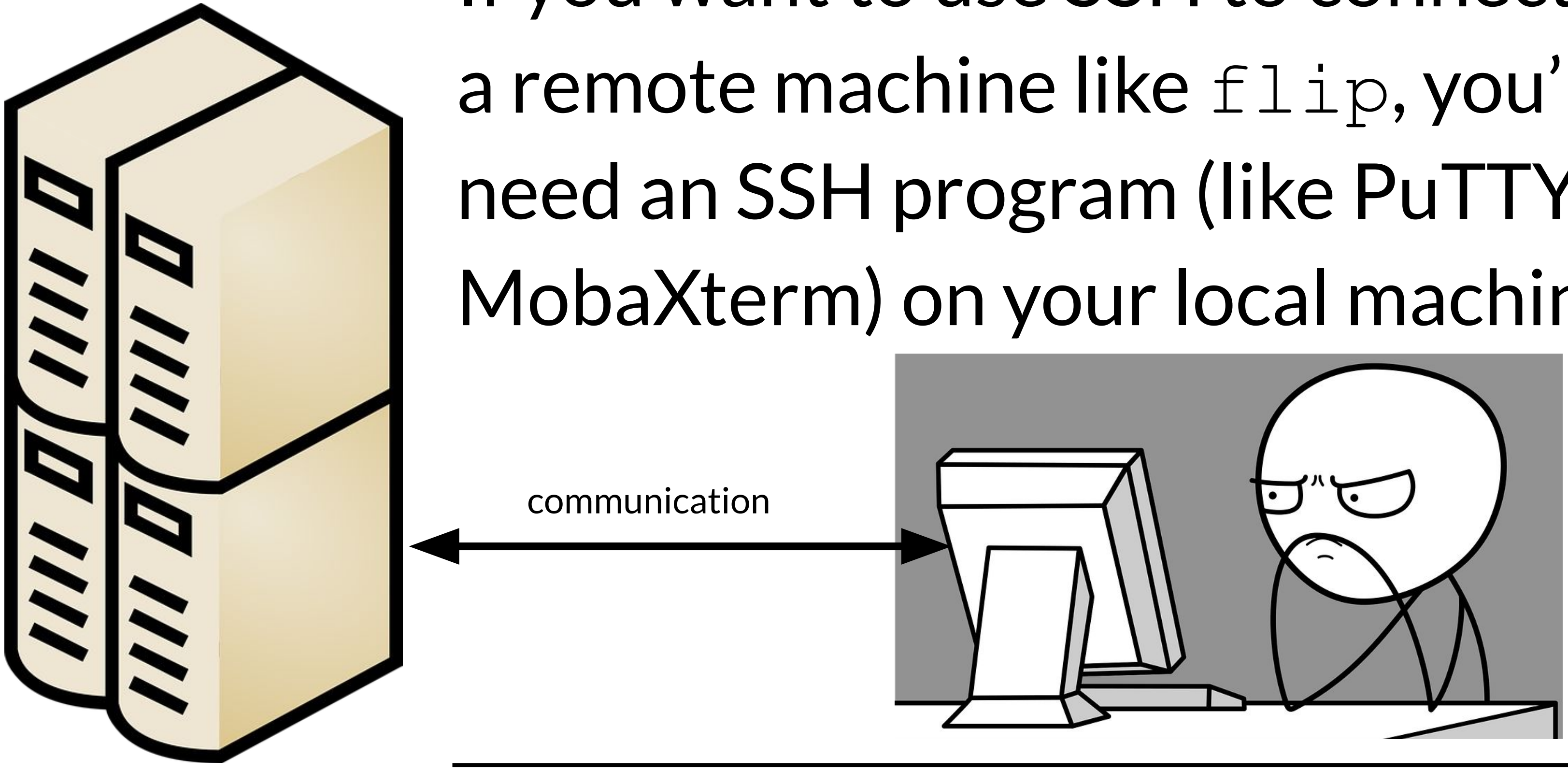
Remote Machines to Connect to

There are two main machines run by ENGR you'll use for CS 261:

- `flip.engr.oregonstate.edu`
  - for on/off campus
- `flop.engr.oregonstate.edu`
  - for on/off campus

Getting SSH

If you want to use SSH to connect to a remote machine like `flip`, you’ll need an SSH program (like PuTTY or MobaXterm) on your local machine.



communication

`flip.engr.oregonstate.edu`

You (with SSH program)

SSH on Windows

The most common SSH program for Windows is PuTTY. You can get it here:

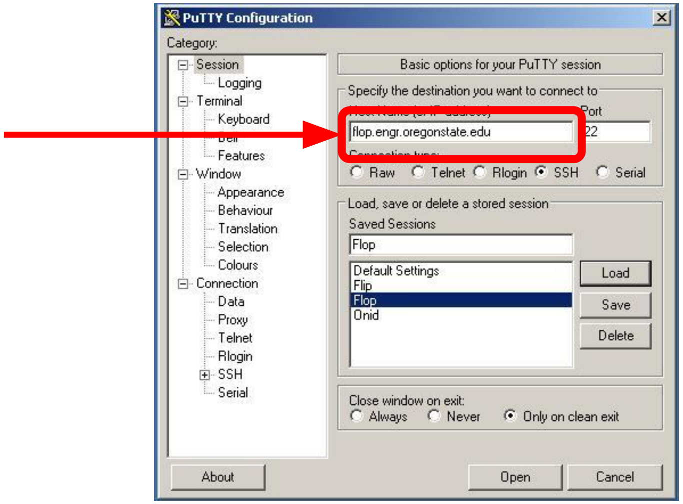
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

A lot of you will also be familiar with MobaXterm and can use that, too.

Using PuTTY on Windows: Step 1

Once you have PuTTY, just type the name of the machine you want to connect to and hit the “Open” button.

This says
`flop.engr.oregonstate.edu`



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

flop.engr.oregonstate.edu

Port 22

○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

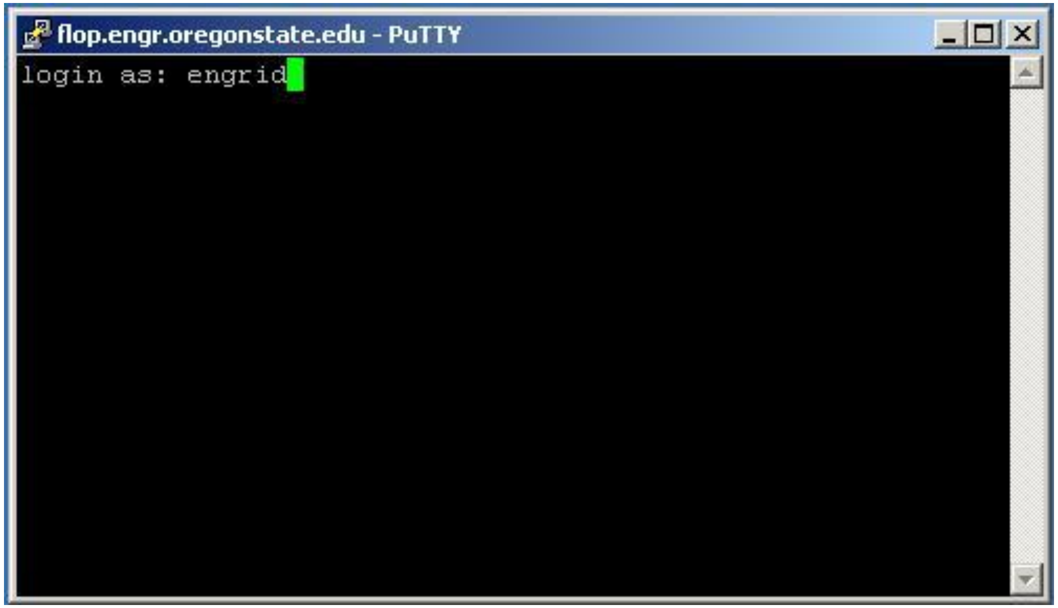
Saved Sessions: Flop

Default Settings
Flip
Flop
Onid

Close window on exit:
○ Always ○ Never ● Only on clean exit

Using PuTTY on Windows: Step 2

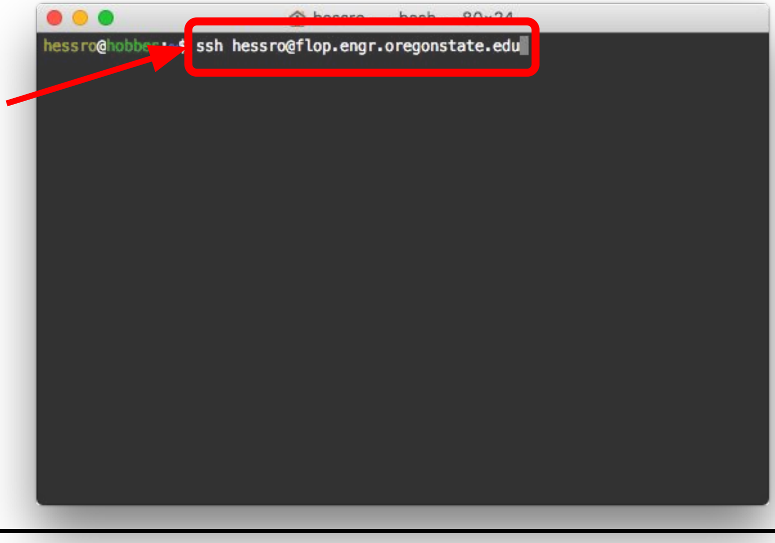
Then, just enter your OSU ENGR username and password.



```
login as: engrid
```

SSH on Mac and Linux

On Mac or Linux, just open your "Terminal" app and use the `ssh` command there.



```
hessro@hobbes:~$ ssh hessro@flop.engr.oregonstate.edu
```

This says `ssh hessro@flop.engr.oregonstate.edu` where `hessro` is my ENGR username and `flop`... is the machine I want to connect to. I'll be asked for my password.

Once you’re connected

Once you’ve followed those steps and are connected to a remote machine, everything (commands, files, etc.) will be the same regardless of whether you’re connected from Windows, Mac, or Linux.

- Specifically, it’s a Unix session on the remote machine.

Basic Unix Commands

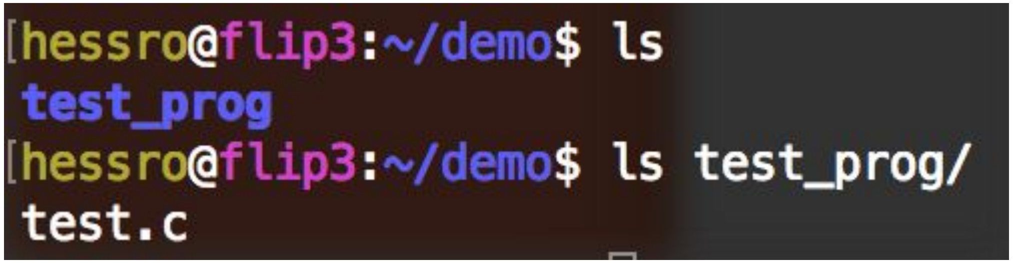
Let’s review some basic Unix commands.

- Some things to remember:
  - In Unix, directories in a path name are separated with a `/` character.
  - One dot (`.`) represents the current directory.
  - Two dots (`..`) represents the parent directory.

Basic Unix Commands: `ls`

The `ls` command lists the files in a directory.

- Without arguments, it lists the current directory.

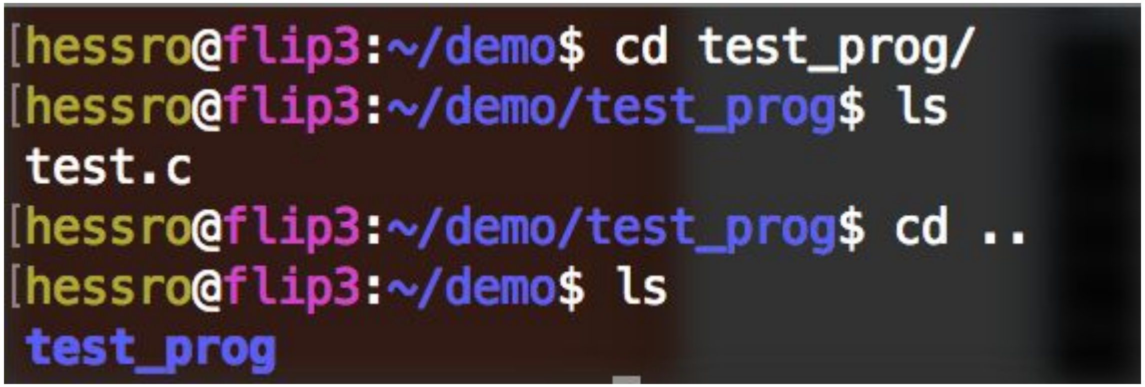


```
[hessro@flip3:~/demo$ ls
test_prog
[hessro@flip3:~/demo$ ls test_prog/
test.c
```

Basic Unix Commands: `cd`

The `cd` command navigates to a new directory.

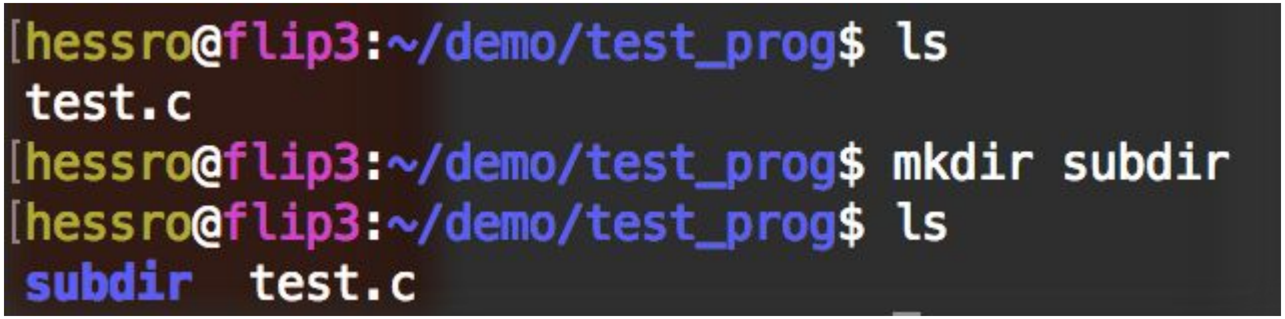
- Without arguments, it navigates to your home directory.



```
[hessro@flip3:~/demo$ cd test_prog/
[hessro@flip3:~/demo/test_prog$ ls
test.c
[hessro@flip3:~/demo/test_prog$ cd ..
[hessro@flip3:~/demo$ ls
test_prog
```

Basic Unix Commands: `mkdir`

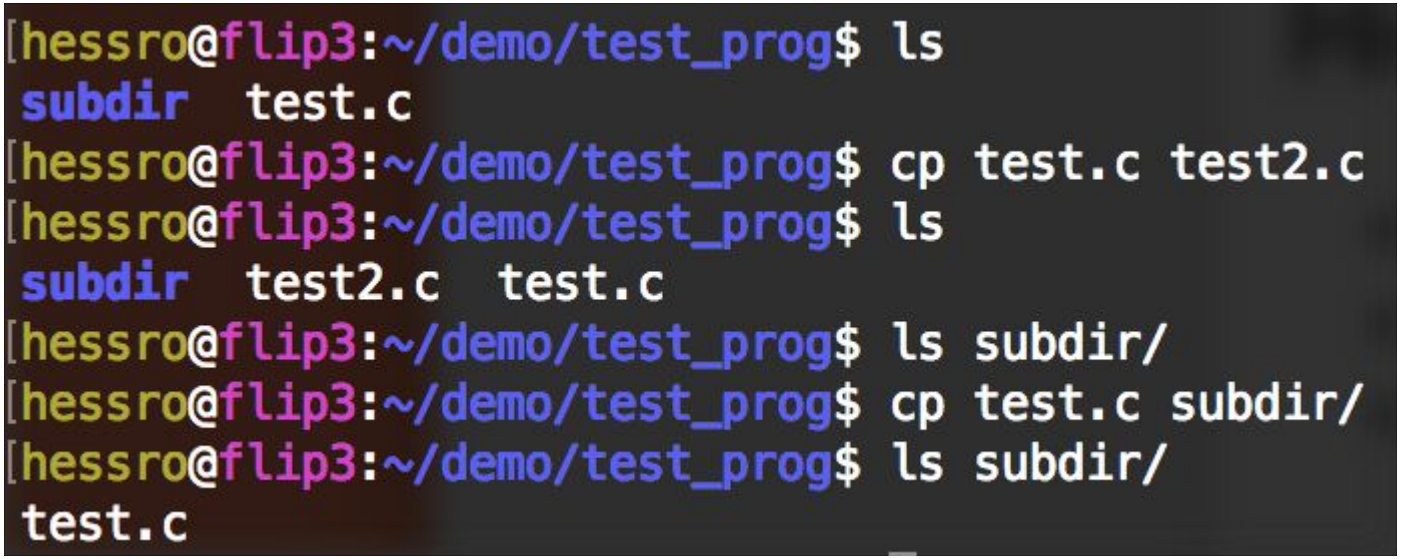
The `mkdir` command creates a new directory.



```
[hessro@flip3:~/demo/test_prog$ ls
test.c
[hessro@flip3:~/demo/test_prog$ mkdir subdir
[hessro@flip3:~/demo/test_prog$ ls
subdir  test.c
```

Basic Unix Commands: `cp`

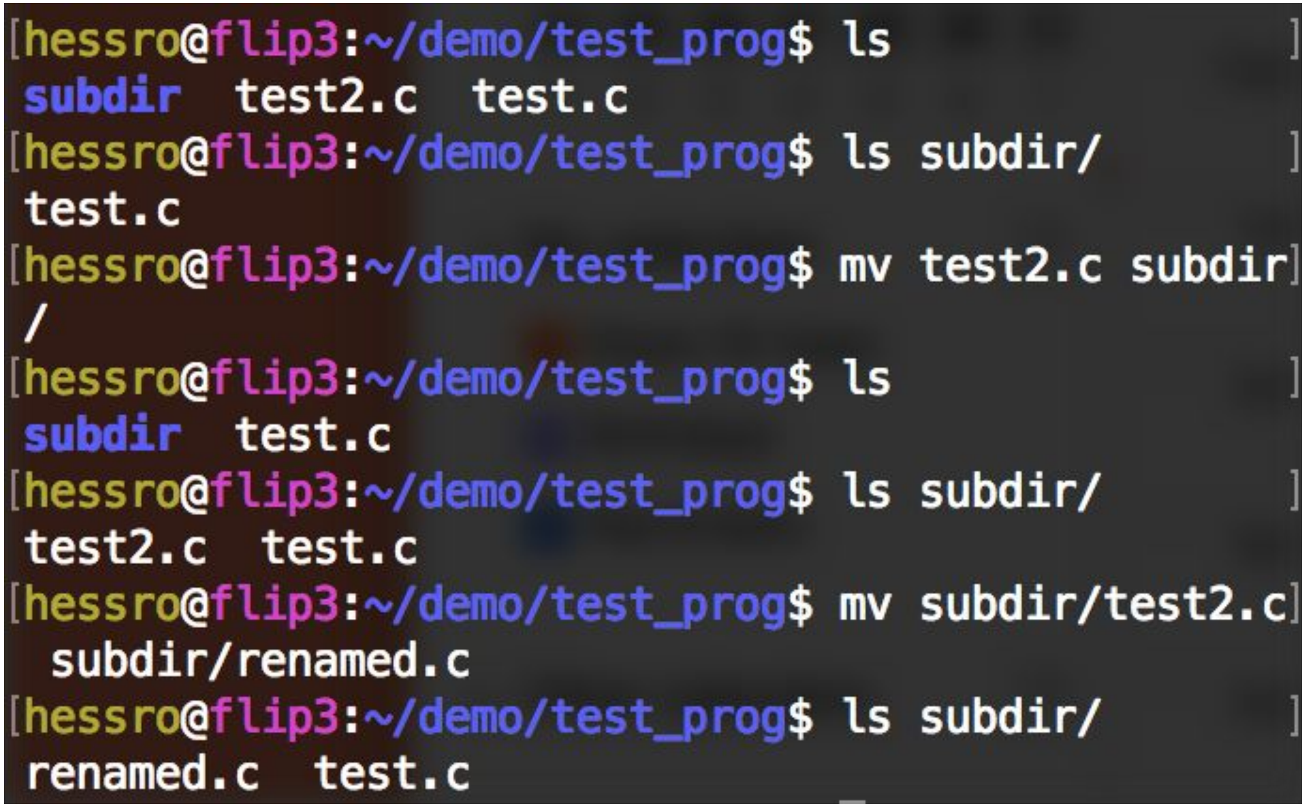
The `cp` command copies a file or directory.



```
[hessro@flip3:~/demo/test_prog$ ls
subdir  test.c
[hessro@flip3:~/demo/test_prog$ cp test.c test2.c
[hessro@flip3:~/demo/test_prog$ ls
subdir  test2.c  test.c
[hessro@flip3:~/demo/test_prog$ ls subdir/
[hessro@flip3:~/demo/test_prog$ cp test.c subdir/
[hessro@flip3:~/demo/test_prog$ ls subdir/
test.c
```

Basic Unix Commands: `mv`

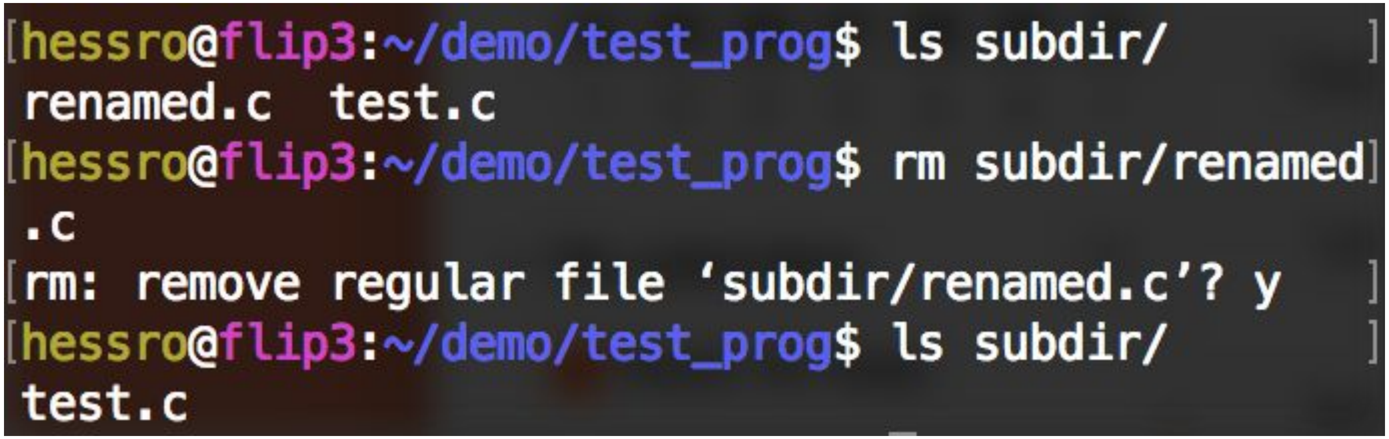
The `mv` command moves or renames a file or directory.



```
[hessro@flip3:~/demo/test_prog$ ls                              ]
subdir  test2.c  test.c
[hessro@flip3:~/demo/test_prog$ ls subdir/                      ]
test.c
[hessro@flip3:~/demo/test_prog$ mv test2.c subdir]
/
[hessro@flip3:~/demo/test_prog$ ls                              ]
subdir  test.c
[hessro@flip3:~/demo/test_prog$ ls subdir/                      ]
test2.c  test.c
[hessro@flip3:~/demo/test_prog$ mv subdir/test2.c]
 subdir/renamed.c
[hessro@flip3:~/demo/test_prog$ ls subdir/                      ]
renamed.c  test.c
```

Basic Unix Commands: `rm`

The `rm` command removes (deletes) a file.



```
[hessro@flip3:~/demo/test_prog$ ls subdir/                ]
renamed.c  test.c
[hessro@flip3:~/demo/test_prog$ rm subdir/renamed]
.c
[rm: remove regular file ‘subdir/renamed.c’? y           ]
[hessro@flip3:~/demo/test_prog$ ls subdir/                ]
test.c
```

Basic Unix Commands: `exit`

The `exit` command exits (terminates) an SSH session.

Luke Skywalker executing `exit` command on SSH session to Death Star.

In-Terminal Text Editors

When you’re connected on an SSH session, there are a few text editors you can use to edit files directly on the remote machine from the terminal, e.g.:

- vim
- emacs

vim in a Nutshell

- To open a file: `vim <filename>`
- Two main modes for working with a file:
  - Normal (read-only) mode: `esc`
  - Insert (editing) mode: `i` (from normal mode)
- Use arrow keys to move cursor
- Important commands (from normal mode):
  - Save: `:w`
  - Quit: `:q`
- Interactive tutorial: http://www.openvim.com/

emacs in a Nutshell

- To open a file: `emacs <filename>`
- Only one mode (editing mode) for working with a file
- Use arrow keys to move cursor
- Important commands (from normal mode):
    - Save: `ctrl-x ctrl-s`
    - Quit: `ctrl-x ctrl-c`
- Beginner's guide: http://www.jesshamrick.com/2012/09/10/absolute-beginners-guide-to-emacs/