OSU Oregon State University

CENTER FOR GENOME RESEARCH & BIOCOMPUTING

"Introduction to Unix/Linux"
INX_U18, Day 3, 2018-07-27

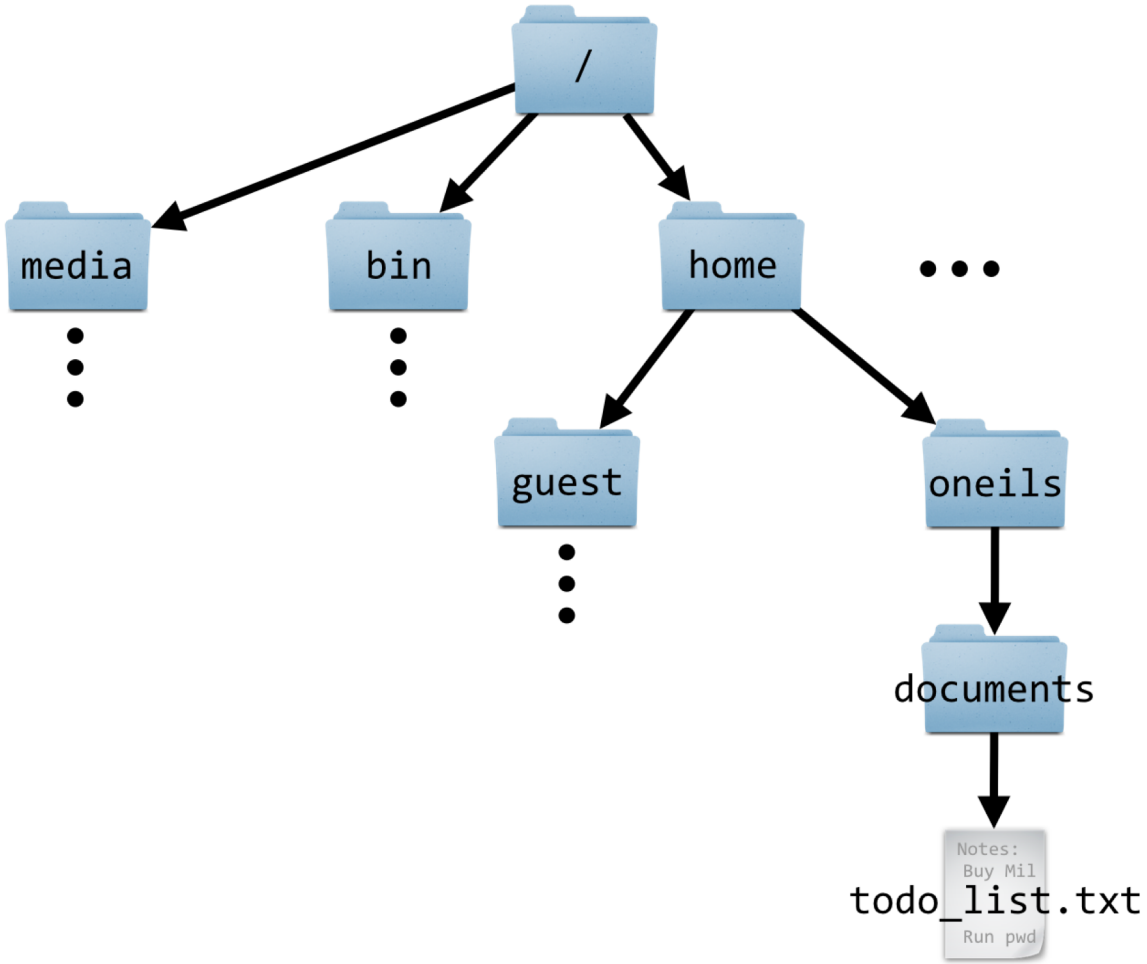
relative paths, file manipulation and other tools, shell history/tab completion

Learning Outcome(s):

Navigate and use the Unix/Linux file system, including understanding directory structure/permissions, and creating/editing/removing files and directories.

Matthew Peterson, OSU CGRB, matthew@cgrb.oregonstate.edu

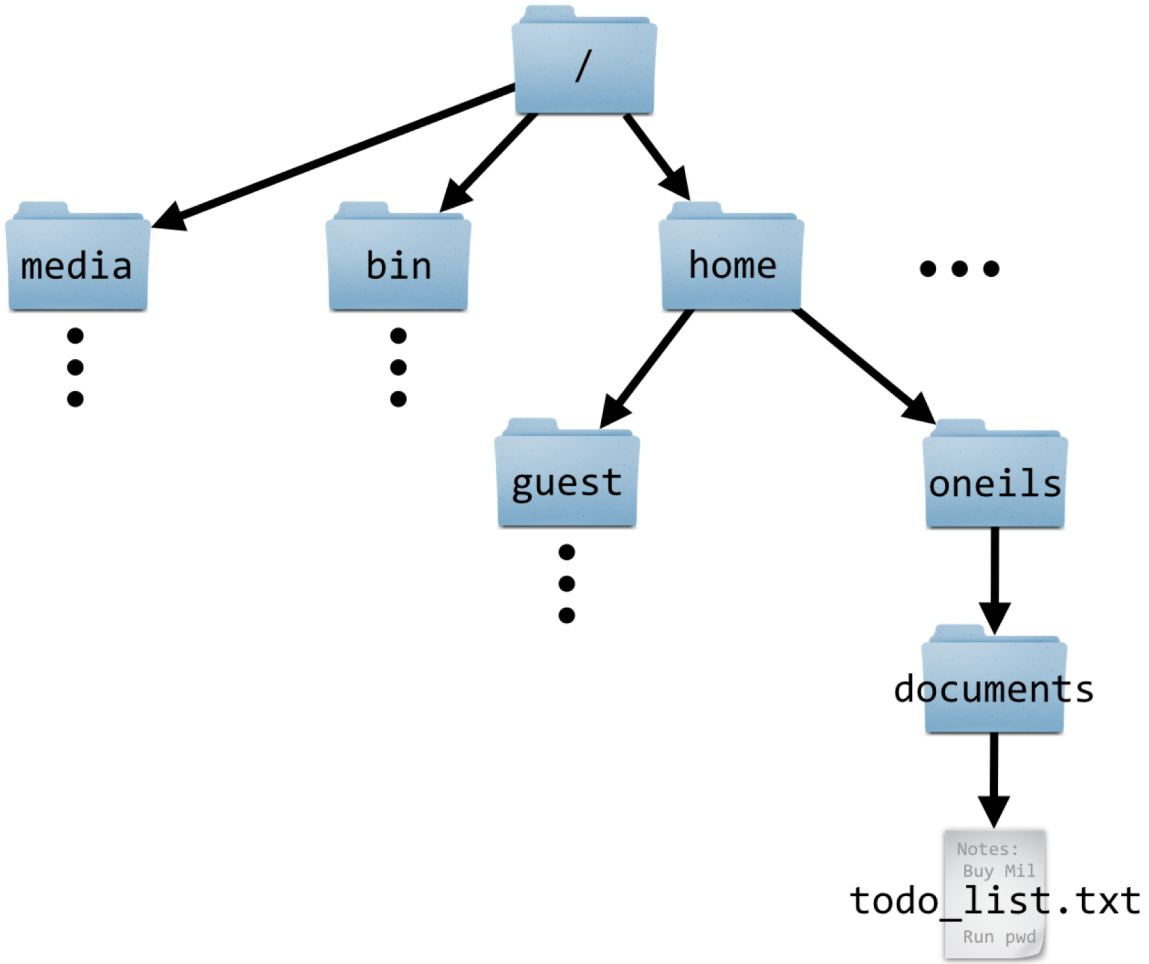
Linux Filesystem



Oregon State UNIVERSITY

http://library.open.oregonstate.edu/computationalbiology/chapter/the-command-line-and-filesystem/

Absolute Path (to a file)



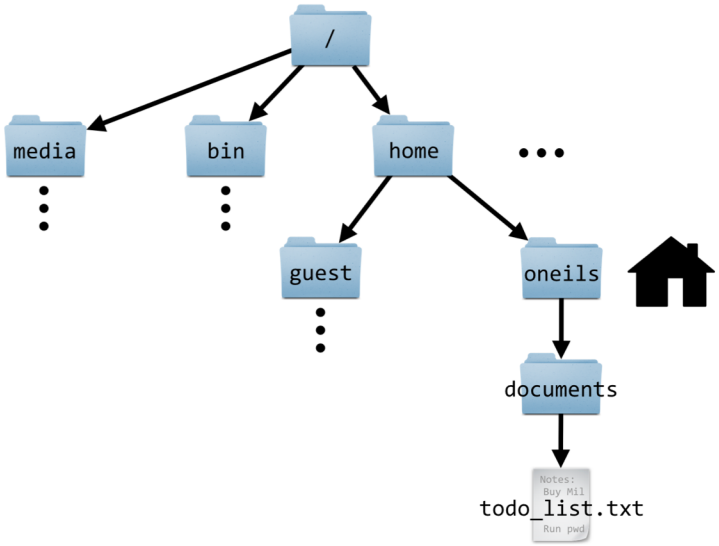
/home/oneils/documents/**todo_list.txt**

Oregon State UNIVERSITY

Relative Path

- Using an **absolute path** is not always necessary!
- A **relative path** locates a file or directory ***relative*** to **$PWD** (**P**resent **W**orking **D**irectory)
- **Relative path** example

```
cd $HOME
cd documents
```

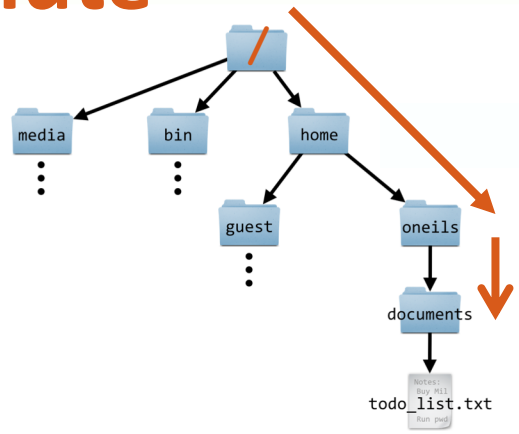


https://commons.wikimedia.org/wiki/File:Home_Icon.svg

Relative Path Example

- **Relative paths** are always ***relative*** to the **P**resent **W**orking **D**irectory ($PWD)
- They do **not** not start with a forward slash **/**:

```
cd /home/oneils # Absolute
cd documents # Relative
```



Does this work?

```
cd /home/oneils
cd /documents
```

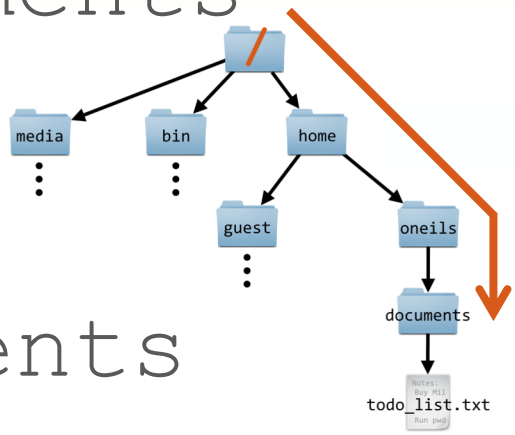
If it did work, is it what you intended?

Oregon State UNIVERSITY

Absolute Path Example

- **Absolute paths** must start with a: **/**
- "always start my path at the top of the filesystem tree and go down there…"

```
cd /home/oneils/documents
```



Does this work?

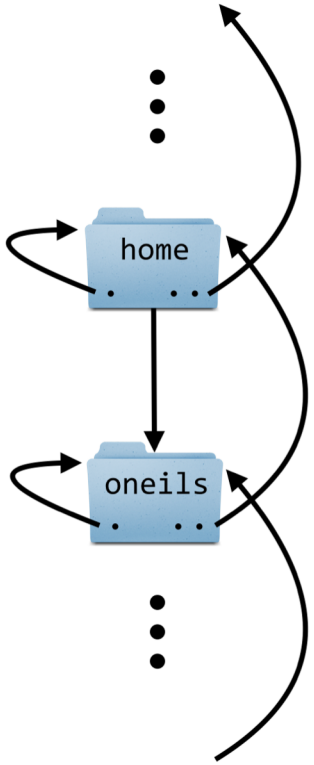
```
cd home/oneils/documents
```

If it did work, what was your $PWD ?
Could it not be what you intended?

Oregon State UNIVERSITY

`ls -a` and links: `.` and `..`

- Two special directories: `.` and `..`
- These are "virtual" folder links
- Present in **every** directory on the tree
- `.` refers to the folder it is contained in, "**here**" i.e., a directory link to itself
- `..` refers to the folder above the directory it is contained in, "**up**"



`.` = Go "here"
`..` = Go "up"

"Virtual" Directory Links

. = Go "here"

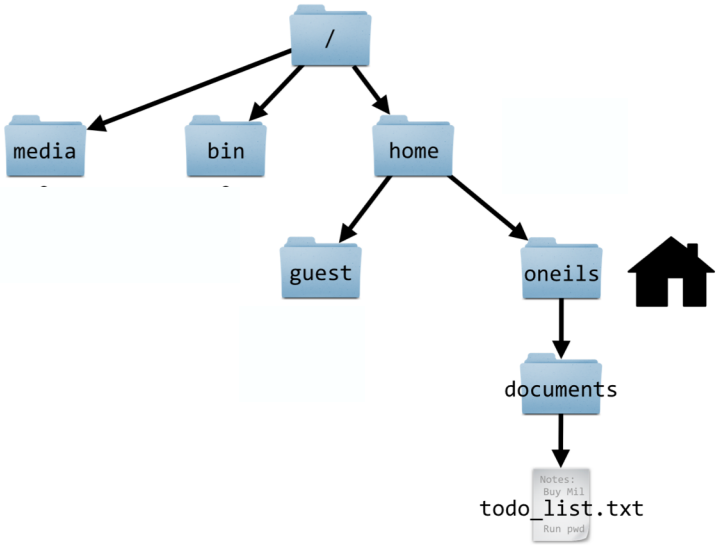
.. = Go "up"

Where does this take you?

```
cd $HOME
cd .
cd .
```

Where does this take you?

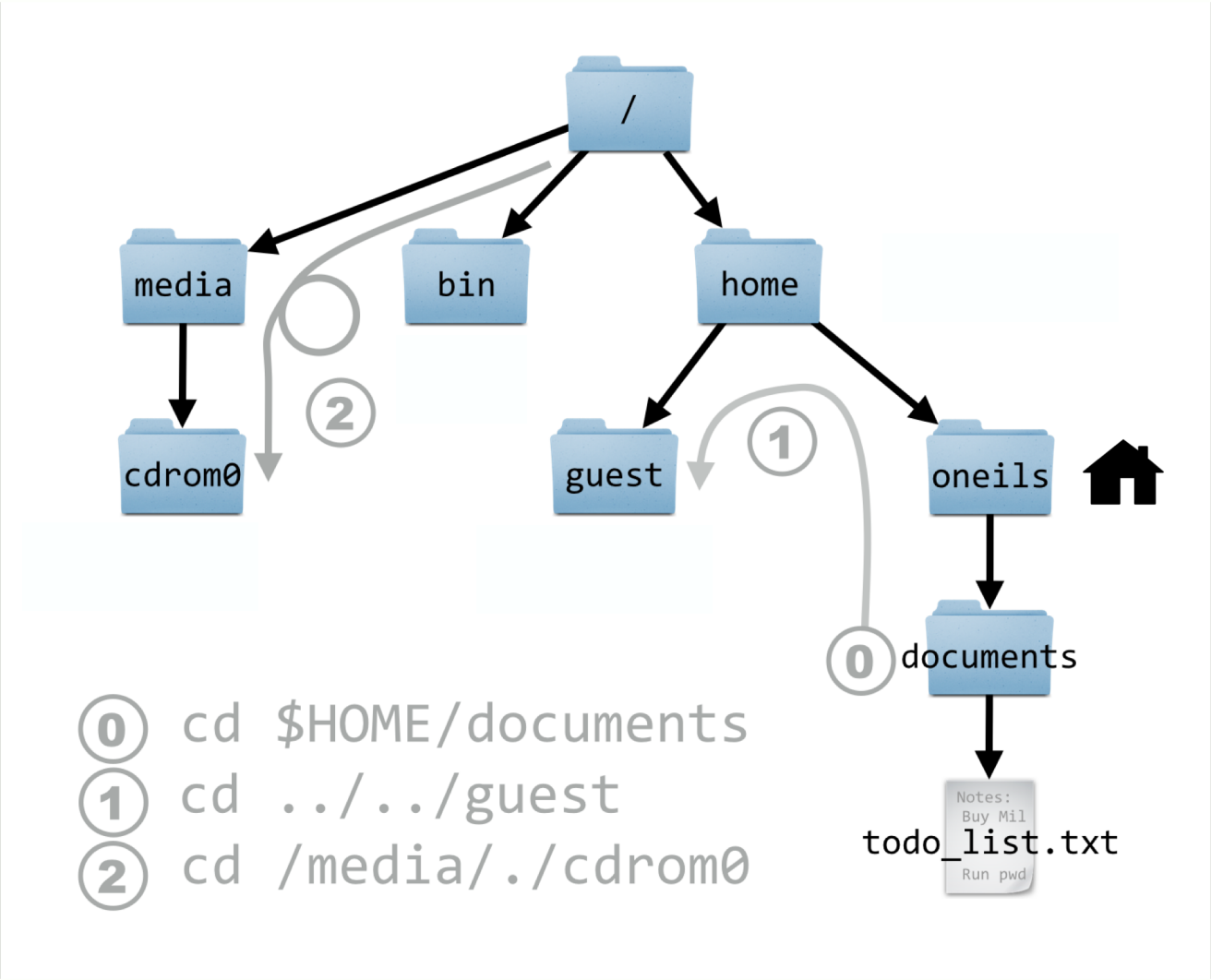
```
cd $HOME
cd ..
cd ..
```



Oregon State UNIVERSITY

https://commons.wikimedia.org/wiki/File:Home_Icon.svg

Use in absolute and relative paths



https://commons.wikimedia.org/wiki/File:Home_Icon.svg

`less` View the contents of a file

`less <pathtofile>`

- Scroll through a file with:
  - Arrow keys for up/down ↑ ↓
  - Return/Enter key for single line forward
  - Space bar for a page forward; `b` for back a page
  - A mouse is useless!
- `q` to quit out of `less`

Oregon State UNIVERSITY

`less` Search and line wrapping

- Search mode: **/** followed by the search pattern
  - **n** for next match
  - **p** for previous match
- By default less "wraps" long lines, which may be undesirable; disable with **-S**
  - Scroll left/right using the arrow keys ← →

Oregon State UNIVERSITY

mkdir Make directory

mkdir <new**dir**name>

- Creates a new directory in your $PWD

mkdir <validpath>/<new**dir**name>

- Creates a new directory given a valid, pre-existing absolute or relative path

```
cd
mkdir projects
```

Oregon State UNIVERSITY

mv Move or rename a file or dir

```
mv <sourcepath> <destpath>
```

- Can be relative **or** absolute paths
- Often just names in your $PWD to rename

```
cd
mv p450s.fasta renamed.fasta
mv renamed.fasta projects
mv projects projects_dir
```

Oregon State UNIVERSITY

mv Move multiple files and undo

```
mv one.txt two.txt projects
```

- We can move multiple files at a time

- To "undo" all of the previous moves and renames

```
cd
mv projects_dir/one.txt .
mv projects_dir/two.txt .
mv projects_dir/renamed.fasta p450s.fasta
mv projects_dir projects
```

Oregon State UNIVERSITY

`mv` Semantics ("a bit funny")

- If the destination does not exist, it is created.

`mv orig.txt`[x] *`new.txt`*[+]

- If the destination does already exist and is a:
  - **Directory**:
    - the file is moved into there with the same name

      `mv orig.txt`[x] `somedir`[+]

      Result: `somedir/orig.txt`[+]
  - **File**:
    - the file is overwritten with the source

      `mv orig.txt`[x] `exists.txt`[x+]

`cp` Copy files and directories

```
cp <sourcepath> <destpath>
```

- Similar to **`mv`** with two differences:
  - The original is not removed
  - If you want to copy an entire directory (and all its contents), you need to use the **`-r`** recursive flag

```
cp -r projects projects_clone
```

Oregon State UNIVERSITY

`rm` Remove files and directories

`rm <pathtofile>`

- Removes a file or files(s), e.g.,

`rm one.txt two.txt`

- If you want to remove a directory use the `-r` flag

`rm -r <pathtodir>`

- If you do not want to be prompted `y/n` to delete each file or sub directory, add `-f` (force)

`rm -rf <pathtodir>`

Oregon State UNIVERSITY

`rm` With great power...

...comes great responsiblity!

- If you **rm** a file, it is gone forever!
- There is no "recycle bin" to recover
- Be careful when removing files:

```
rm -rf projects-copy
```

- Removes "`projects-copy`"

```
rm -rf projects copy
```

- Removes "`projects`" **and** "`copy`"
  - Is this what you intended?!

Oregon State UNIVERSITY

du Disk usage of file or directory

du -s -h <path>

du -sh <path>

- Check the size of a file or directory
- **-s** summary output
- **-h** human readable (**K**ilo, **M**ega, **G**iga, **T**era byte)

du -sh projects

du -sh .

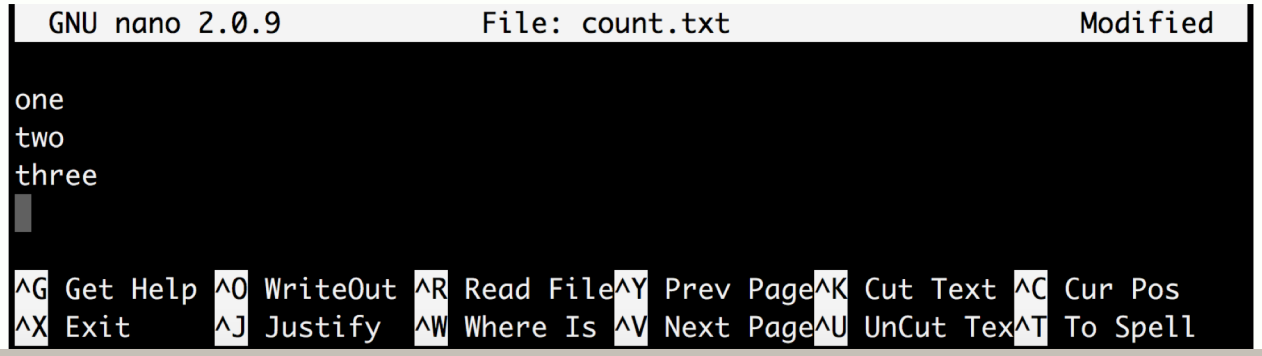
Oregon State UNIVERSITY

nano To edit a text file

nano <pathto**file**>

nano todo.txt

- Ctrl-**o** saves your changes
- Ctrl-**x** exits



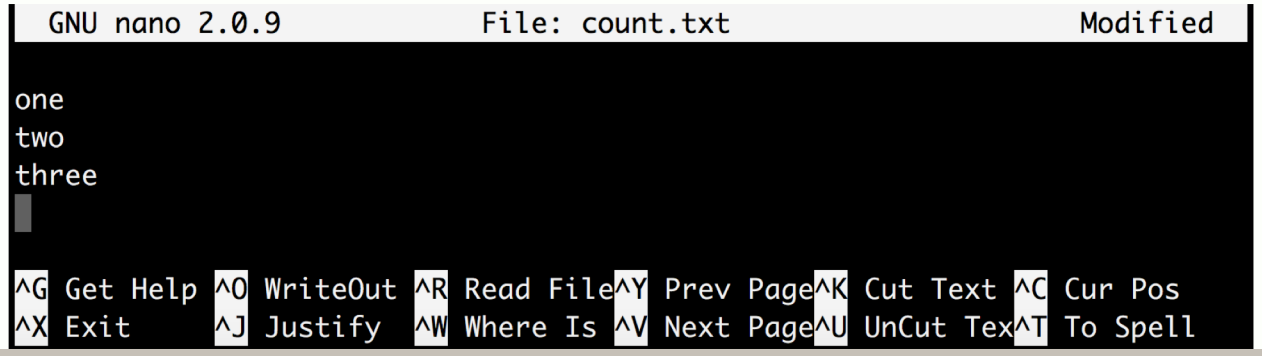
nano Word wrapping

nano -w <pathto**file**>

- Turns off word wrapping (for newly edited text), this is the default behavior, without specify **-w**

nano -I -r 40 <pathto**file**>

- **-I** ignores the default .rc configuration files and **-r 40** word wraps at 40 columns



`man` and `info` Getting help

```
man <command_name>
man ls
```

- Load the **man**ual page for a command

```
info <command_name>
info ls
```

- Load the info documents; not always available

Note: `less` is used to load these pages, so you can use the arrow keys and search via **/**

`top` Display processes

`top`

- Interactive program that shows running programs
- Processes sorted by amount of **%CPU** used
  - 0 to 100*CPU cores, e.g., 4800 = 48 cores@100% !
- **%MEM** percentage of total system memory used

```
top - 11:20:26 up 39 days, 21:40,  3 users,  load average: 0.24, 0.13, 0.10
Tasks: 1586 total,   1 running, 1585 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  529246968k total, 525912988k used,  3333980k free,   307784k buffers
Swap: 131071996k total,        0k used, 131071996k free, 520411468k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
39648 petersm3  20   0 22560 2588 1052 R  1.6  0.0   0:00.26 top
    1 root      20   0 25684 1708 1364 S  0.0  0.0   0:05.68 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.50 kthreadd
```

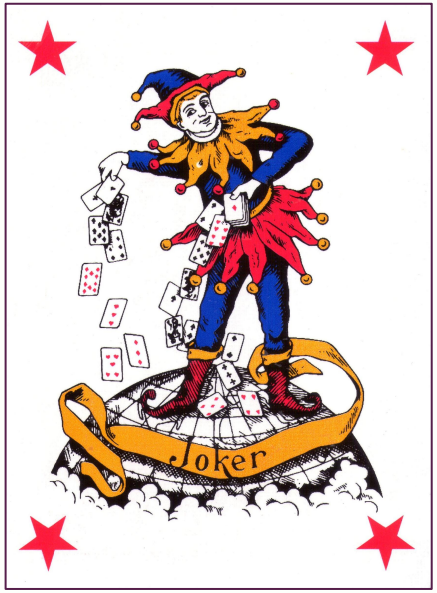
Oregon State UNIVERSITY

* and ? Wildcards

- Many commands can operate on more than one file at a time, e.g., **`cp`**, **`rm`**, and **`mv`**
- Two wildcard patterns that the shell will "expand" (fill in the matching results):

`*` Matches any 0 or more characters

`?` Matches any single character



Oregon State UNIVERSITY

Wildcard example

```
mkdir test
mv x1.txt x2.txt x91.txt test
```

Could be written as:

```
mv x*.txt test
```

To move only the first two files (**x1.txt** and **x2.txt**)

```
mv x?.txt test
```

Why was **x91.txt** not moved in this case?

Oregon State University

Wildcard continued

```
mv *.temp *.tmp
```

This will not work

```
ls -l *.txt
```

Lists in long format only those files ending in .txt

* Wildcard Warning

```
rm -rf *.tmp
```

Remove all files and directories ending in `.tmp`

```
rm -rf * .tmp
```

Space here!

Removes everything (in **$PWD**)!

i.e., * (everything) **and** a directory (or file) named `.tmp`

Oregon State UNIVERSITY

Shell Tab Completion

- Typing long path names can be tedious and error prone! Type the first few characters and then **<tab>** to attempt to "auto complete"

- If the path is ambiguous it will show options, keep typing and hit **<tab>** again. **Tab away!**

Oregon State UNIVERSITY

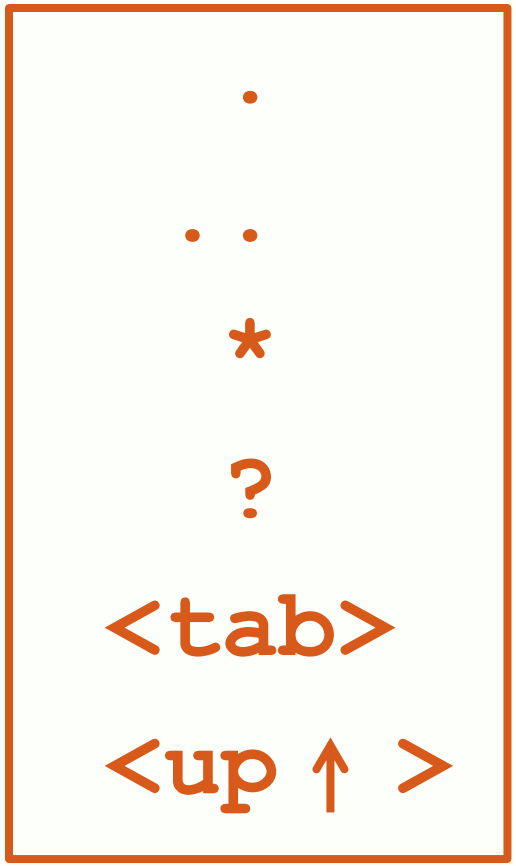
Shell history

- Previous command are remembered by your shell.
- Press the "up" arrow ↑ several times to find the command you want. Press "Enter" to re-run it.
- If you have a command you want to run and need to edit it, use the arrow keys ← → to find the spot to edit.
  - Press "Enter" to run that command; the cursor does **not** need to be at the end of the line!

Oregon State UNIVERSITY

Command / Concept Review

- less
- mkdir
- mv / cp / rm
- du
- nano
- man / info
- top



Oregon State UNIVERSITY