



CENTER FOR
GENOME RESEARCH &
BIOCOMPUTING

“Introduction to Unix/Linux” INX_U18, Day 2, 2018-07-25

shell(s), prompt, setenv, echo, quoting, filesystem, paths, pwd, cd, ls

Learning Outcome(s):

Navigate and use the Unix/Linux file system,
including understanding directory structure

Matthew Peterson, OSU CGRB, matthew@cgrb.oregonstate.edu
Please do not redistribute outside of OSU; contains copyrighted materials.

Command prompt

The diagram illustrates the structure of a command prompt. It shows the following components:

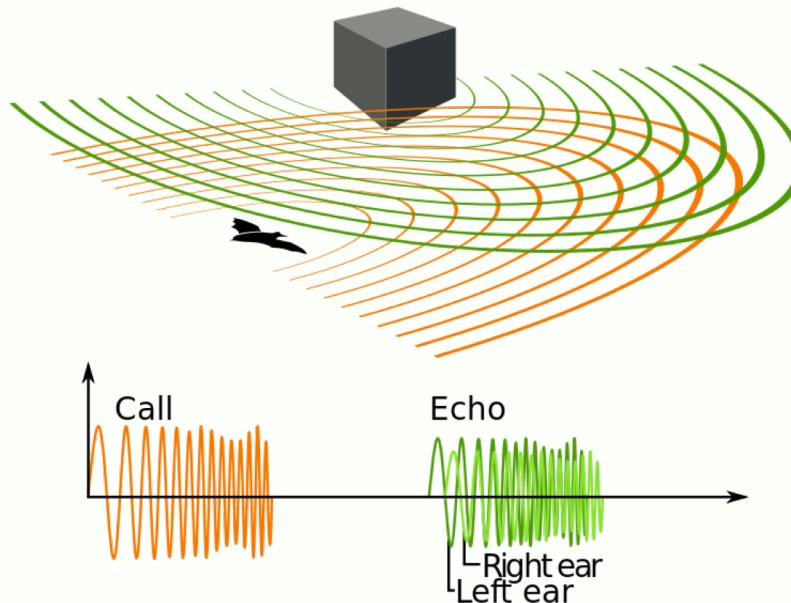
- Username:** `oneils`
- Machine Name:** `@atmosphere`
- Working Directory:** `~$`
- Command:** `echo`
- Parameter 1:** `hello`
- Parameter 2:** `there`
- Output:** `hello there`

A bracket labeled "Output" groups the final two lines of text: `hello there`.

echo

- “display a line of text”
- Example:

echo hello!



Environment Variables

- Bits of information (preferences) stored your shell
 - Can be used by you or by other programs
 - Variables start with a **\$** sign

```
echo $USER
```

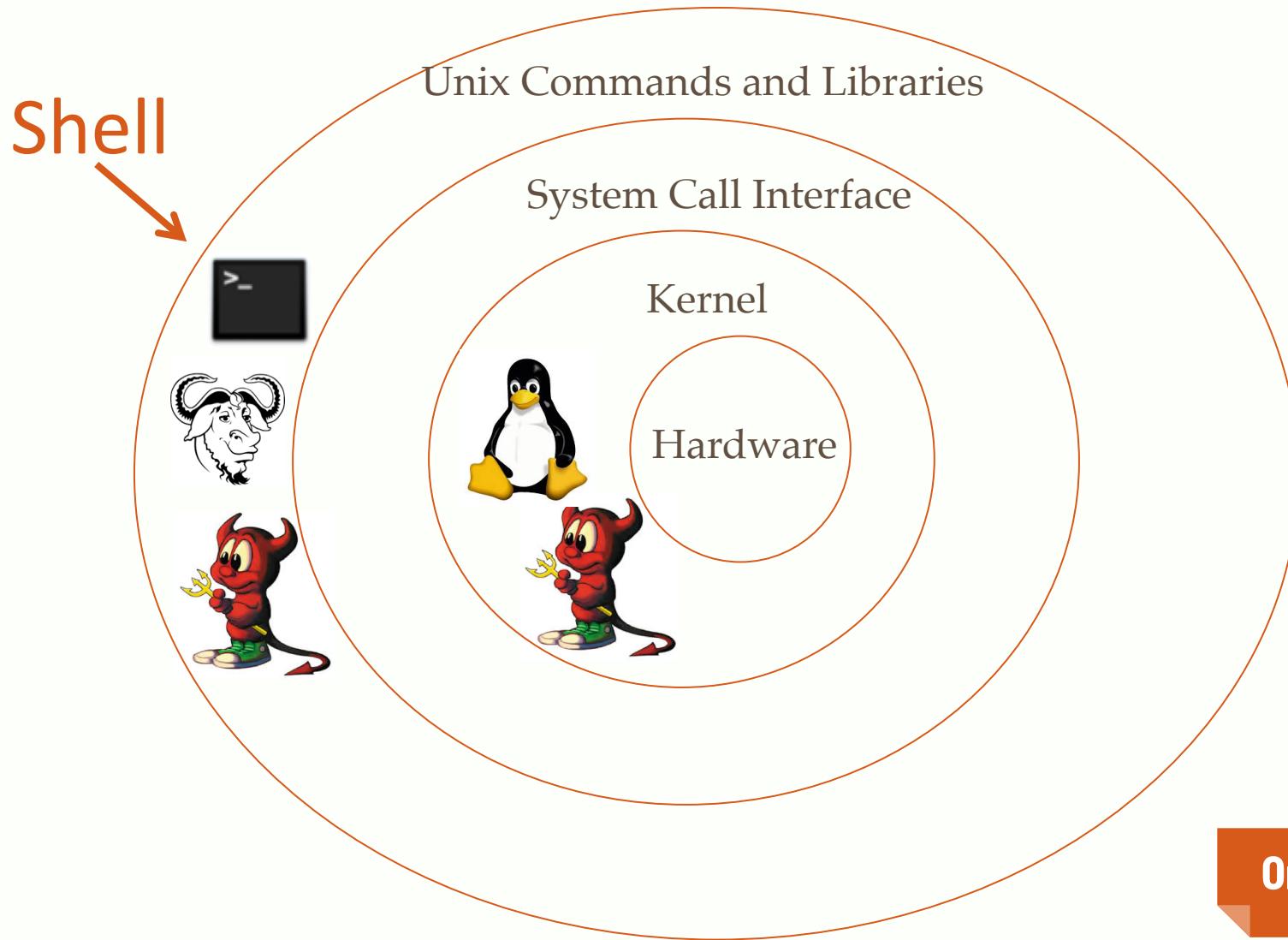
```
echo $0
```

- List of all the current environment variables set for your shell?

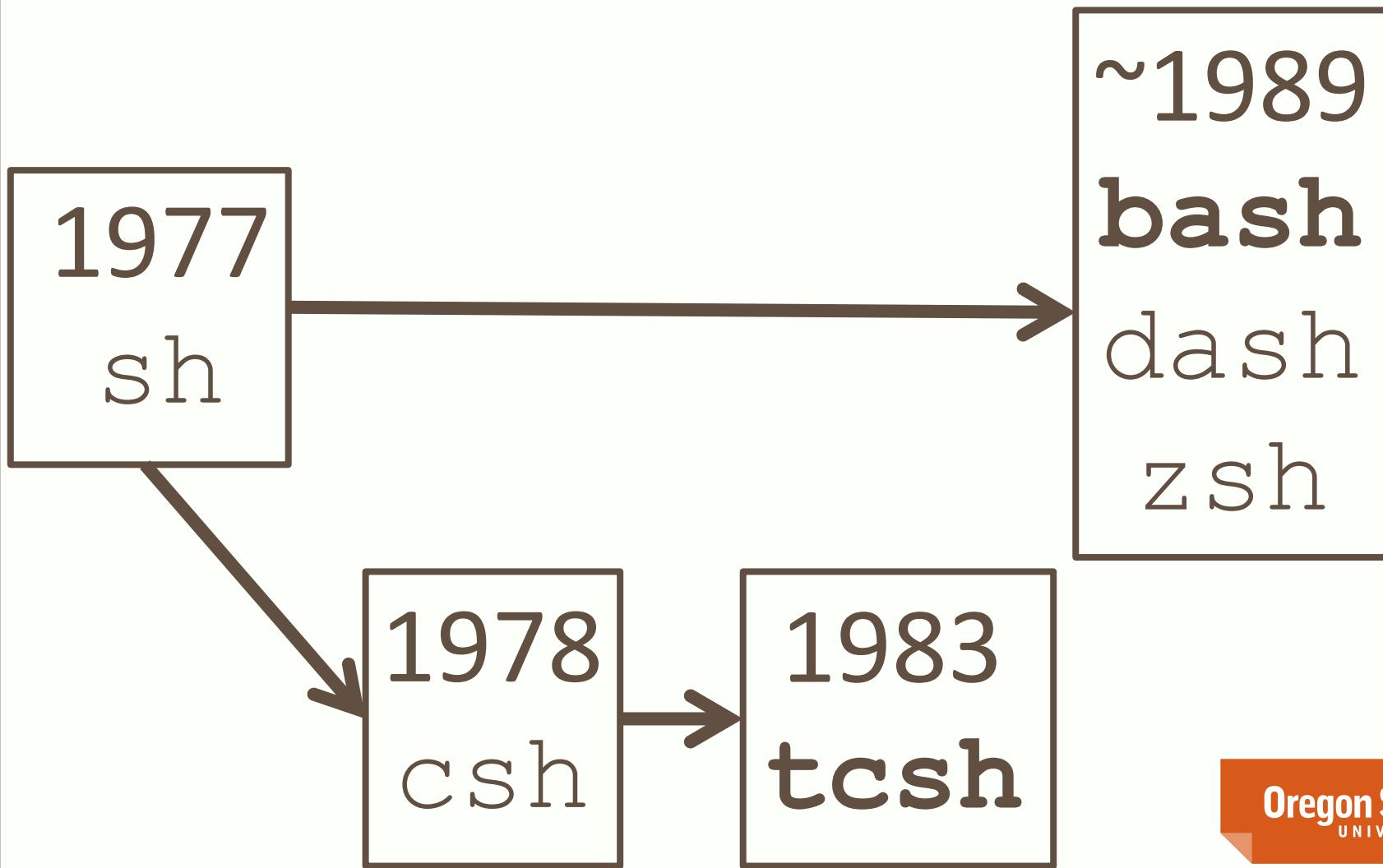
```
env
```



“Kernel,” “Shell,” “Applications”



Shell:



Shell: **bash**

“it is a default shell on the major Linux distributions and OS X.”

[https://en.wikipedia.org/wiki/Bash_\(Unix_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))

MacOSX switched from **tcsh** being its default shell to **bash** in approximately 2003.

Setting Environment Variables

setenv <var name> <value>

setenv greeting hello

setenv greeting hello class

setenv greeting 'hello class'

setenv greeting "hello \$USER"

What happens for each?

echo \$greeting

Quotes

- Used to “group up” words, such that they are a single ‘parameter’ (value)
- Can be used to “escape” special characters, e.g., spaces ''
- Example not using quotes for a space:

```
setenv greeting hello\ class  
echo $greeting
```

Quotes: single ' and double "

- Single quotes '' do not expand the contents of variables inside them
- Double quotes "" do expand the contents of variables inside them

```
setenv greeting 'hello $USER'
```

```
echo $greeting
```

```
setenv greeting "hello $USER"
```

```
echo $greeting
```

setenv in **bash** is: **export**

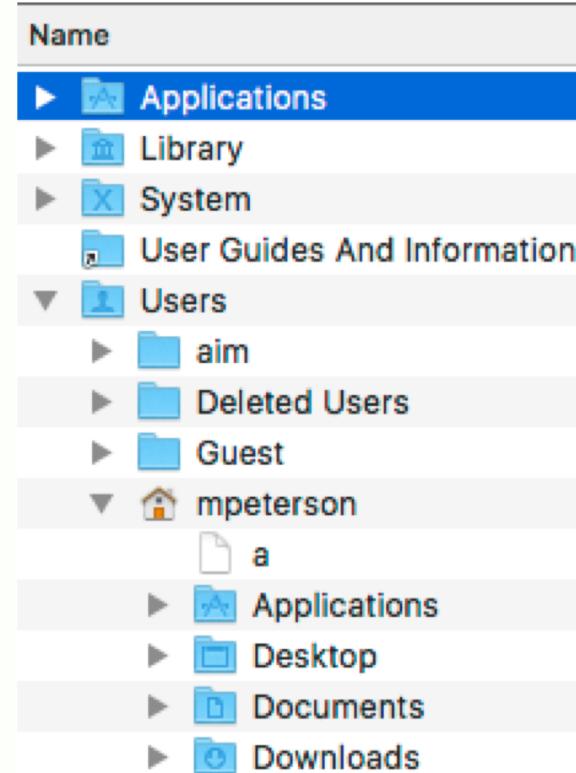
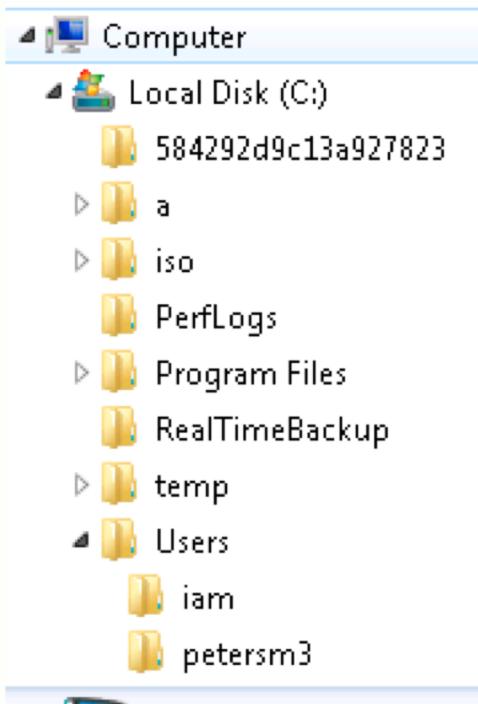
bash

```
export greeting="hello $USER"  
echo $greeting
```

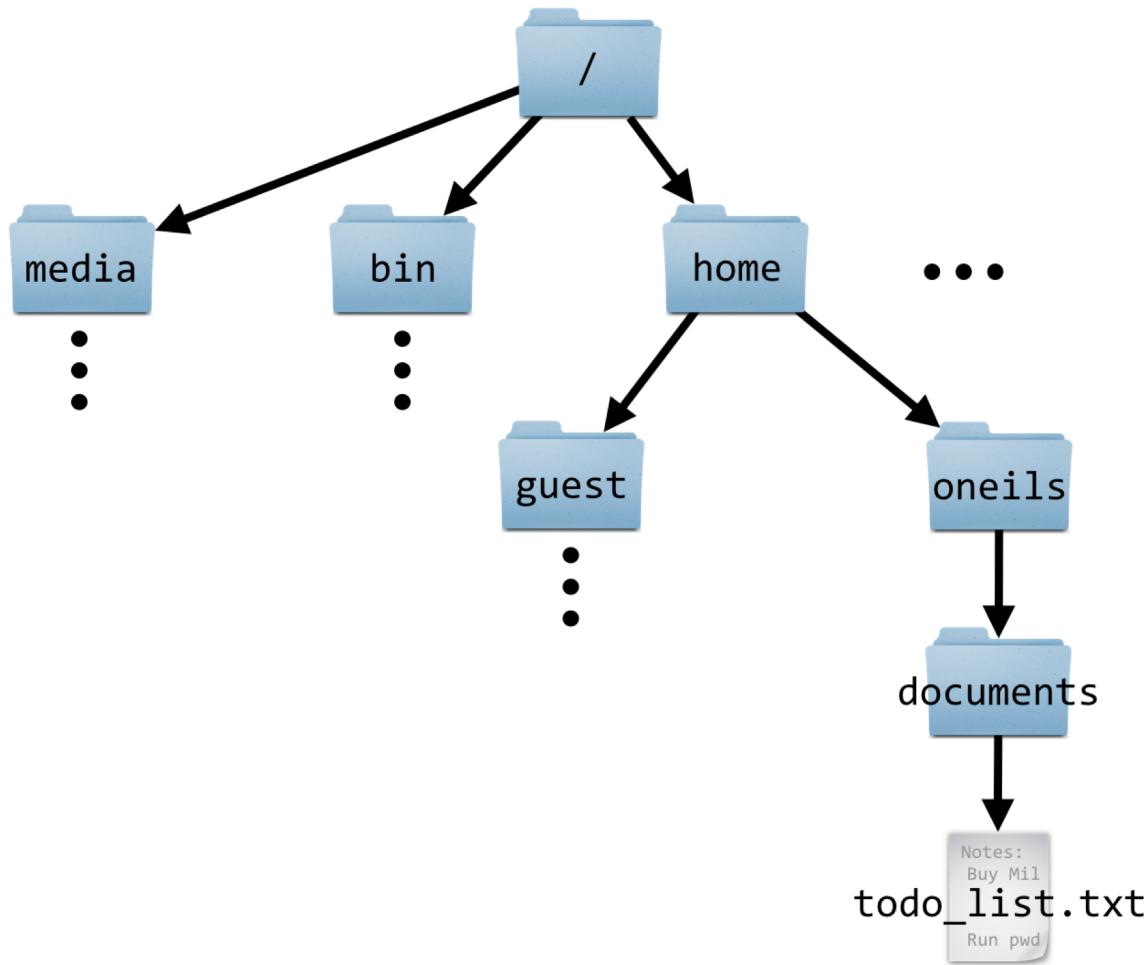
Lifetime of a variable?

- Variables set in a shell “last” until you log out (**exit**)
- When you log back in next time, they will not exist
- There are ways to permanently set variables (that persist between logins/logouts); we will address in a later class.

Windows and Mac Filesystems

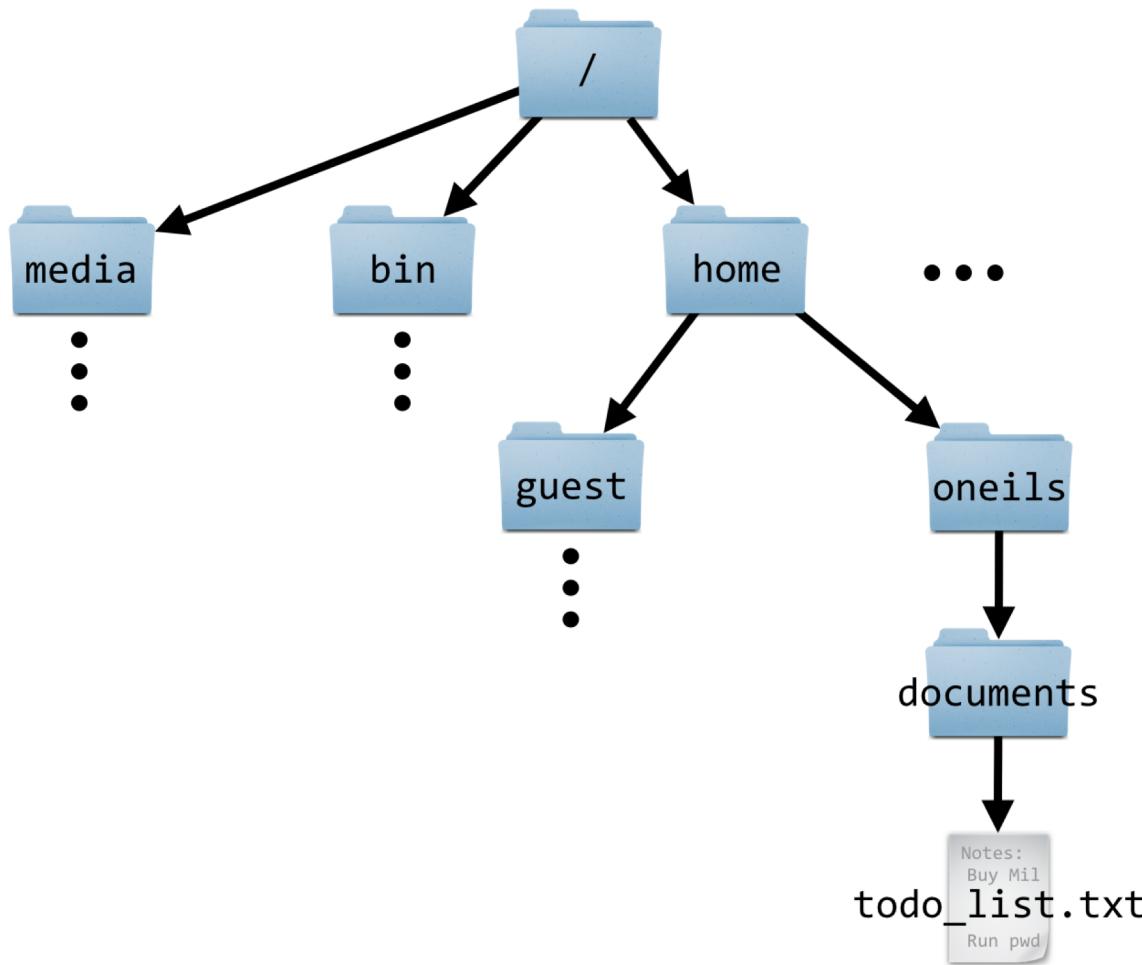


Linux Filesystem



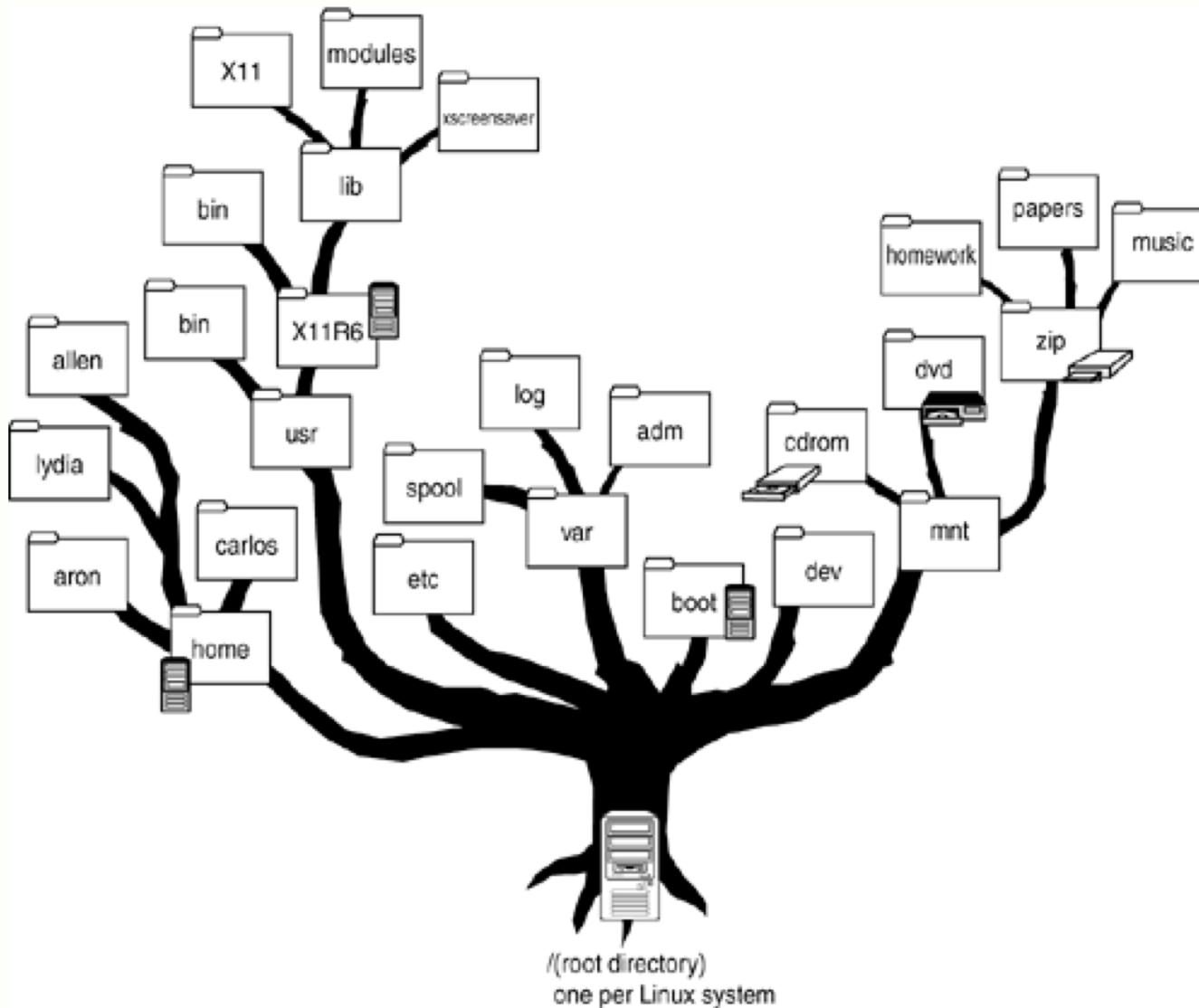
http://teaching.cgrb.oregonstate.edu/CGRB/oneil/primer/partI_3_the_command_line_and_filesystem.html

Absolute Path (to a file)

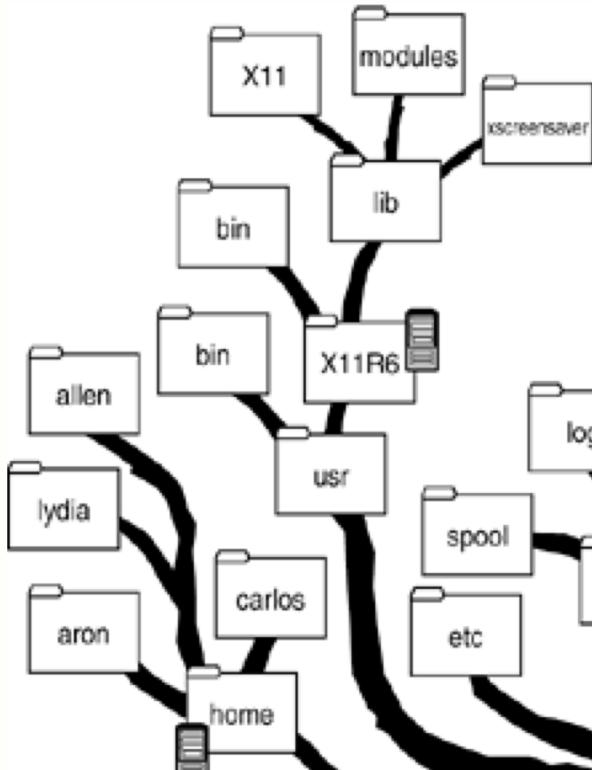


/home/oneils/documents/**todo_list.txt**

Linux Filesystem (upside down?)



Absolute Path to “music”?



/mnt/zip/**music**



mnt

//mnt/zip/**music** vs. /mnt/zip/**music**

Print Working Directory

echo **\$PWD**

pwd

Acronyms:

- PWD = “Present Working Directory”
- CWD = “Current Working Directory”

cd Change Directory

- To move around the file system (“tree”) we can use the command: **cd**
- This allows us to move “up”, “down”, and jump all around the filesystem

Going back to your Home Directory

There are several ways to get back to your home directory:



If you know the absolute path, type it in:

```
cd $HOME
```

```
cd
```

```
cd ~
```

Feeling lost?



There's no place like **\$HOME**

ls Listing files and directories

- List files and directories in your **PWD/CWD** (“where I am located right now in the filesystem tree”)
- Running **ls** after **cd** is a good way to familiarize yourself with the filesystem
- In the shell directories are denoted by:
 - **BLUE** highlighting of the name
 - **/** at the end of the name

`ls` Listing “hidden” (“dot” files)

`ls -a`

- Configuration files often start with a: `.`
- There's no reason you need to see them on a daily basis when working (but they are there).

ls -l Long format

ls -l

- We will go into detail what all of these individual fields mean (and how to manipulate them) in a later class.
- By default it shows file sizes in bytes
- One byte is *approximately* 1 “character”,
e.g., x or 5 or @

`ls -h` “human readable”

`ls -l -h`

- The **-h** option is used with **-l**
- Shows file sizes noted by:
 - **K** (kilobytes)
 - **M** (megabytes)
 - **G** (gigabytes)
 - **T** (terabytes)

ls “consolidation” of arguments

ls -l -h -a

Can be written also as:

ls -lha

Note: Not all commands/programs support the “merging” of multiple options.

Command / Variable Review

- echo
- env / setenv
- tcsh / bash
- pwd
- cd
- ls

\$0
\$USER
\$HOME
\$PWD

Exercises / Practice

- Moving around (**cd**)
- Listing files/directories (**ls**)
- Checking where you are (**pwd**)
- Setting some (temporary) environment variables (**setenv**)
 - Displaying them (**echo**)