Exercises: Navigating in Bash

- Look at the man page for `ls` or using `ls --help`, what does the `-h` (`--human-readable`) option do? **Hint:** Try using it with `ls -l`

  `ls -l -h` prints file sizes in human readable formats (6.2M instead of 6492707)

Exercises: Navigating in Bash

- Starting from **`/home/bioresearch/amy/data`**, which of the following commands could Amy use to navigate to her home directory which is **`/home/bioresearch/amy`**?

    1. **`cd .`** - **This would not change the directory at all**
    2. **`cd /`** - **This would change the directory to the root directory (/)**
    3. **`cd /home/bioresearch/amy`**
    4. **`cd ../..`** - **This would change the directory to /home/bioresearch/**
    5. **`cd ~`**
    6. **`cd home`** - **This would display the error "`home: No such file or directory`"**
    7. **`cd ~/data/..`**
    8. **`cd`** - **This would not change the directory at all**
    9. **`cd ..`**

Exercises: Navigating in Bash (cont.)

- The command `ls -R` lists the contents of directories recursively, i.e. lists their sub-directories, sub-sub-directories, and so on in alphabetical order at each level.

  The command `ls -t` lists things by time of last change, with most recently changed files or directories first.
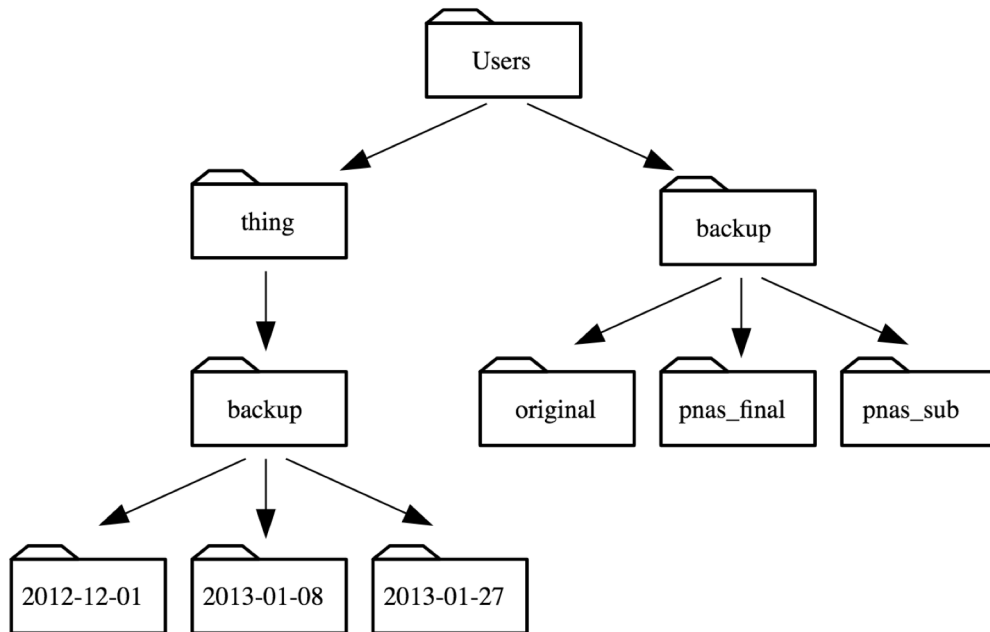
  In what order does `ls -R -t` display things?

  Hint: `ls -l` uses a long listing format to view timestamps.

  ls -R -t would display things recursively (depth-first), ordered by the time of the last change

Exercises: Navigating in Bash (cont.)

Using this filesystem diagram:

- if `pwd` displays **/Users/thing**, what will

  `ls -F ../backup` display?
  1. `../backup: No such file or directory`
  2. `2012-12-01      2013-01-08      2013-01-27`
  3. `2012-12-01/     2013-01-08/     2013-01-27/`
  4. `original/ pnas_final/    pnas_sub/`

- If `pwd` displays **/Users/backup**, which `ls` command

  will display:

  `pnas_sub/      pnas_final/   original/`
  1. `ls pwd`
  2. `ls -r -F`
  3. `ls -r -F /Users/backup`
  4. Either #2 or #3 above, but not #1

Schedule

9:00 – 10:30	Introduction to HPC Connecting to the Clusters Navigating in Bash
10:30 – 10:45	Break
10:45 – 12:00	**File Manipulation** **Wildcards and Pipes**
12:00 – 1:00	Lunch
1:00 – 2:15	Writing Reusable Scripts Running Applications on the Clusters
2:15 – 2:30	Break
2:30 – 4:00	Submitting Jobs Transferring Data with Globus

Home vs Work

/home/group/user ($HOME)

- Not intended for I/O to active jobs

- Quota-limited to 20GB per user

- **Backed up for best-effort disaster recovery**

- Meant for items that take up relatively small amounts of space.
    - Source code
    - Program binaries
    - Configuration files

- **Read-Only to worker nodes**

/work/group/user ($WORK)

- Designed for I/O for running jobs

- 50TB quota per group
    - To check group usage
        - `hcc-du`

- **Not backed up**

- **Purge policy**
    - Files will be deleted after 6 months inactivity
    - To check files eligible for purge
        - `hcc-purge`

$WORK Usage Best Practices

- **Avoid large numbers of files**
  - Having hundreds of files in your $WORK directories can degrade the performance of the file system

- **Storage resources are finite**
  - Be judicious about the data you store.
  - Periodically review your files; remove what is no longer needed.

- **Backup valuable data**
  - Users are responsible to backup their valuable research data on $WORK periodically

- **Use Globus for file transfers**
  - We will talk more about this later

Exercises: Writing and Reading Files

- Change directory to your **$WORK** directory

- Create a text file called **bio.txt** – edit this file to include your name and department

- Make a new directory named **contacts** and copy **bio.txt** into this directory

- Enter the **contacts** directory. move the **bio.txt** file, changing the file name to **right.txt**

- Edit **right.txt** to include the name and department of your neighbor to your right (or left if there is no one to your right)

- Now delete the contacts directory

Exercises: Writing and Reading Files

- Copy the tutorial files to your **$WORK** directory with the command:

  `git clone https://github.com/unlhcc/qiime2_workshop_2018.git`

  Explore the directory that was created. How many sub-directores does it contain? What file types are there?

Exercises: Writing and Reading Files

- Change directory to your **$WORK** directory

  cd $WORK

- Create a text file called **bio.txt** – edit this file to include your name and department

  nano bio.txt

- Make a new directory named **contacts** and copy **bio.txt** into this directory

  mkdir contacts

  cp bio.txt ./contacts

Exercises: Writing and Reading Files

- Enter the **contacts** directory. move the **bio.txt** file, changing the file name to **right.txt**

    ```
    cd contacts
    mv bio.txt right.txt
    ```

- Edit **right.txt** to include the name and department of your neighbor to your right (or left if there is no one to your right)

    ```
    nano right.txt
    ```

- Now delete the files and directories you've created

    ```
    rm right.txt bio.txt
    cd ..
    rmdir contacts
    ```

Exercises: Writing and Reading Files

- Copy the tutorial files to your $WORK directory with the command:

    `git clone https://github.com/unlhcc/qiime2_workshop_2018.git`

    Explore the directory that was created. How many sub-directores does it contain? What file types are there?

Exercises: Wildcards and Pipes

- You can use more than one * at a time. How would you run `wc -l` on every file with fb in it?

- Create a folder called `fastq` and move all of our fastq files there in a single mv command.

- How many files are there in the `fastq` directory? Hint: use a shell command to count them!

- Save the output of your previous command to a file named counts.txt