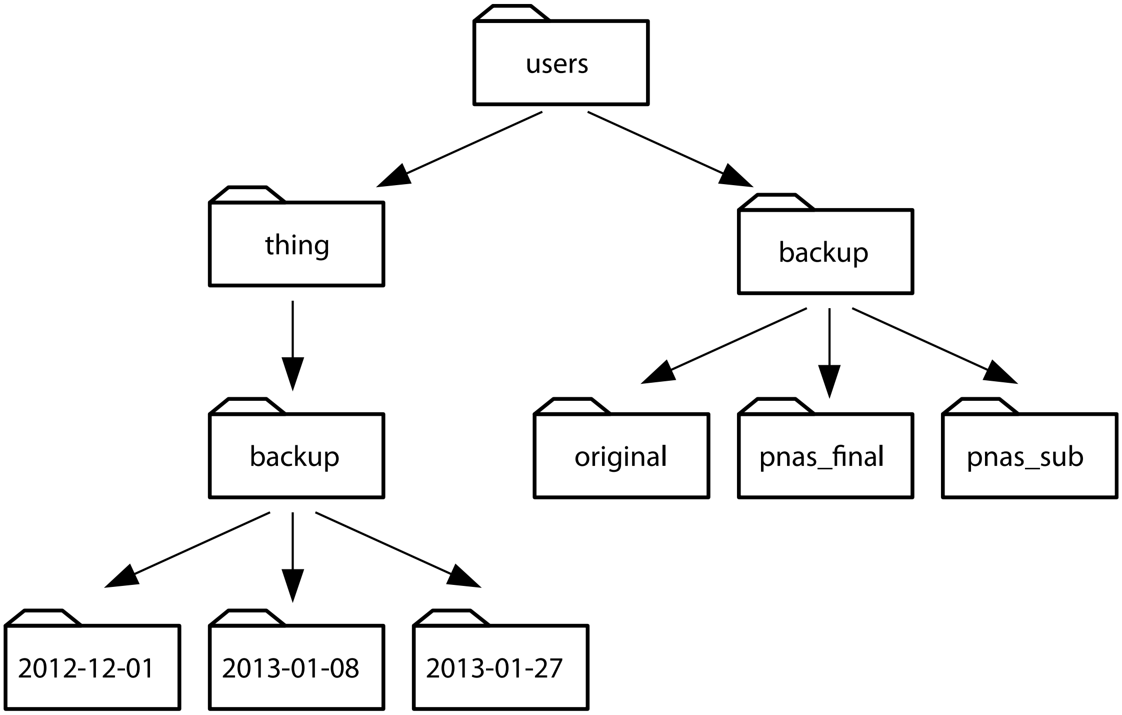
The File System 1



1.If `pwd` displays `/users/thing`, what will `ls ../backup` display?

2.If `pwd` displays `/users/backup`, and `-r` tells `ls` to display things in reverse order, what command will display "`pnas-sub/ pnas-final/ original/`"?

3.Similarly, if `pwd` displays `/users/2012-12-01`, what command will display the same output as before?

4.What does the command cd without a directory name do?

The File System 2

1.What is the output of the closing ls command in the sequence shown below?

```
$ pwd
/home/thing/data
$ ls
proteins.dat
$ mkdir recombine
$ mv proteins.dat recombine
$ cp recombine/proteins.dat ../proteins-saved.dat
$ ls
```

2.Suppose that:

```
$ ls -F
analyzed/  fructose.dat    raw/   sucrose.dat
```

What command(s) could you run so that the commands below will produce the output shown?

```
$ ls
analyzed   raw
$ ls analyzed
fructose.dat    sucrose.dat
```

3.What does cp do when given several filenames and a directory name, as in:

```
$ mkdir backup
$ cp thesis/citations.txt thesis/quotations.txt backup
```

What does cp do when given three or more filenames, as in:

```
$ ls -F
intro.txt    methods.txt    survey.txt
$ cp intro.txt methods.txt survey.txt
```

Why do you think cp's behavior is different from `mv`'s?

4.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 doesls -R -t display things?

Pipes & Filters

(Navigate to the `data/pipes/` directory for these examples.)

1.If we run `sort` on the file `numbers.txt`, what is the output? If we run `sort -n` on the same input, what do we get instead? Explain why `-n` has this effect.

2.What is the difference between

```
$ wc -l < mydata.dat
```

and

```
$ wc -l mydata.dat
```

3.The command `uniq` removes adjacent duplicated lines from its input. For example, run the following commands:

```
$ cat salmon.txt
$ uniq salmon.txt
```

Why do you think `uniq` only removes *adjacent* duplicated lines? (Hint: think about very large data sets.) Which other command could you combine with it in a pipe to remove *all* duplicated lines?

4.Examine the file called `animals.txt` using `less`. What text passes through each of the pipes and the final redirect in the pipeline below?

```
cat animals.txt | head -5 | tail -3 | sort -r > final.txt
```

5.The command:

```
$ cut -d , -f 2 animals.txt
```

produces the following output:

```
deer
rabbit
raccoon
rabbit
deer
fox
rabbit
bear
```

What other command(s) could be added to this in a pipeline to find out what animals the file contains (without any duplicates in their names)?

Searching & Sorting

1.Write a short explanatory comment for the following shell script:

```
$ find . -name '*.dat' -print | wc -l | sort -n
```

2.The `-v` flag to `grep` inverts pattern matching, so that only lines which do *not* match the pattern are printed. Given that, which of the following commands will find all files in `data/chem/pdb` whose names end in `ose.dat` (*e.g.*, `sucrose.dat` or `maltose.dat`), but do *not* contain the string `temp`?

1.`$ find data/chem/pdb -name '*.pdb' -print | grep ose | grep -v temp`

2.`$ find data/chem/pdb -name ose.pdb -print | grep -v temp`

3.`$ grep -v temp $(find data/chem/pdb -name '*ose.pdb' -print)`

4.None of the above.