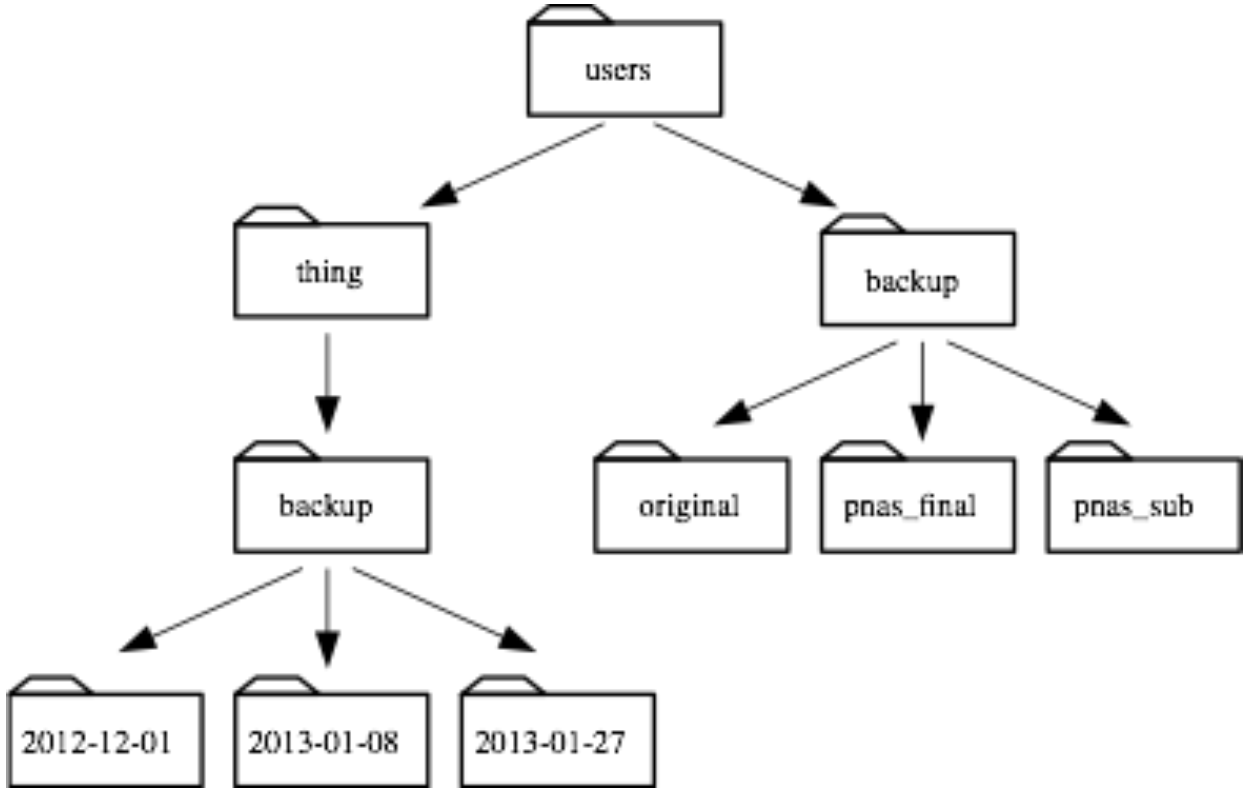
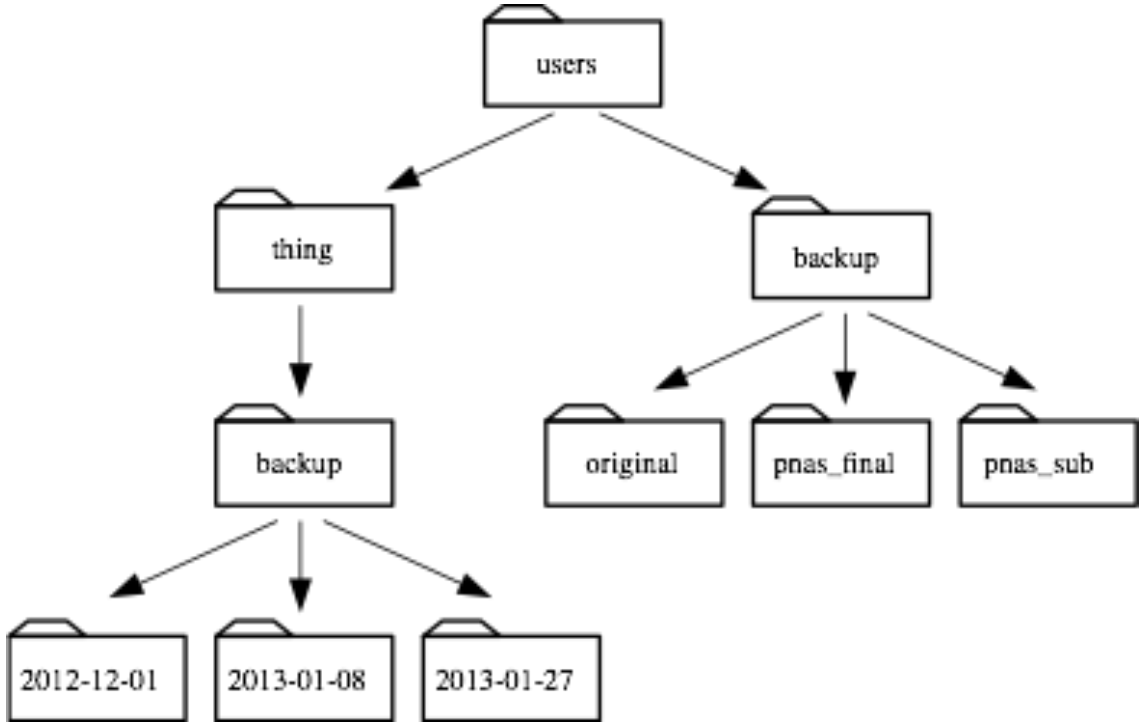


If pwd displays /users/thing, what will ls ../backup display?

A. ../backup: No such file or directory
B. 2012-12-01 2013-01-08 2013-01-27
C. 2012-12-01/ 2013-01-08/ 2013-01-27/
D. original pnas_final pnas_sub



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

A. ls pwd
B. ls –r -F
C. ls –r –F /users/backup

What does the command cd without a directory name do?

A. It has no effect.
B. It changes the working directory to /.
C. It changes the working directory to the user's home directory.
D. It produces an error message.

What does the command **ls** do when used with –s and –h arguments?

A. Lists only files that begin with s and h.
B. Lists file sizes of files that begin with h.
C. Lists ages of files in seconds .
D. Lists file sizes in blocks.

What is the output of the closing “ls”?

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

A. proteins.dat
B. proteins-saved.dat
C. proteins.dat recombine
D. recombine

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
```

A. mv fructose.dat analyzed
   mv sucrose.dat analyzed

B cp fructose.dat analyzed
   cp sucrose.dat analyzed

C. mv *.dat analyzed

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

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

A. Copies both .txt files into backup/
B. Copies thesis/citations.txt onto thesis/quotations.txt, ignores backup
C. Copies thesis/citations.txt onto thesis/quotations.txt and into 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
```

A. Copies intro.txt files over methods.txt and survey.txt
B. Gives an error message and doesn't do anything
C. Copies intro.txt files over methods.txt and copies methods.txt over survey.txt

How can we sort

```
10
2
19
22
6
```

and get

```
2
6
10
19
22
```

A. sort file.txt
B. sort –n file.txt
C. Sort –number file.txt
D. Either A or B

How many lines are in the output from running uniq on

```
coho
coho
steelhead
coho
steelhead
steelhead
```

A. 2

B. 4

A file called `animals.txt` contains the following data:

```
2012-11-05,deer
2012-11-05,rabbit
2012-11-05,raccoon
2012-11-06,rabbit
2012-11-06,deer
2012-11-06,fox
2012-11-07,rabbit
2012-11-07,bear
```

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

What's the first line in final.txt?

A. 2012-11-05,deer

B. 2012-11-07,bear

C. 2012-11-05-raccoon

D. 2012-11-06-rabbit

The command:

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

produces the following output:

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

What can we do with the output to get a list of animals with no duplicates?

A. | sort
B. | uniq | sort
C. | uniq
D. | sort | uniq

Suppose that `ls` initially displays:

```
fructose.dat    glucose.dat    sucrose.dat
```

What is the output of:

```
for datafile in *.dat
do
    ls *.dat
done
```

A. fructose.dat glucose.dat sucrose.dat

B. fructose.dat glucose.dat sucrose.dat
fructose.dat glucose.dat sucrose.dat
fructose.dat glucose.dat sucrose.dat

C. fructose.dat
glocuse.dat
sucrose.dat

In the same directory, what does this loop do?

```
for sugar in *.dat
do
    echo $sugar
    cat $sugar > xylose.dat
done
```

A. Prints fructose.dat, glucose.dat, and sucrose.dat, and copies sucrose.dat to create xylose.dat.
B. Prints fructose.dat, glucose.dat, and sucrose.dat, and concatenates all three files to create xylose.dat.
C. Prints fructose.dat, glucose.dat, sucrose.dat, and xylose.dat, and copies sucrose.dat to create xylose.dat.

What is the output of

```
for left in 2 3
do
    for right in $left
    do
        expr $left + $right
    done
done
```

A. 2 4 3 6

B. 2 3

C. 4 6

D. 5 5

```
$ expr 3 + 5
8
$ expr 30 / 5 - 2
4
```

Suppose that `ls` initially displays:

```
fructose.dat    glucose.dat    sucrose.dat
```

What would **bash example.sh *.dat** produce for example.sh

```
# Script 1
echo *.*
```

A. fructose.dat glucose.dat sucrose.dat

B. fructose.dat glucose.dat sucrose.dat
fructose.dat glucose.dat sucrose.dat
fructose.dat glucose.dat sucrose.dat

C. fructose.dat
glucuse.dat
sucrose.dat

Suppose that `ls` initially displays:

```
fructose.dat    glucose.dat    sucrose.dat
```

What would **bash example.sh *.dat** produce for example.sh

```
# Script 2
for filename in $1 $2 $3
do
    cat $filename
done
```

A. Prints the contents of fructose.dat, glucose.dat and sucrose.dat
B. Prints the contents of fructose.dat, glucose.dat and sucrose.dat, 3 times
C. Combines fructose.dat and glucose.dat and overwrites them into sucrose.dat

Grep –v junk file.txt finds lines which don't contain "junk".

Which of the following commands will find all files in /data whose names end in ose.dat (e.g., sucrose.dat or maltose.dat), but do *not* contain the word temp?

A. rm $(find ./ -name '*.tmp')

How could you find and remove all of the .tmp files in your thesis/ directory and its subdirectories?

A. rm $(find ./ -name '*.tmp')
B. rm *.tmp
C. find ./ -name '*.tmp’ | rm
D. rm -i $(find ./ -name '*.tmp')