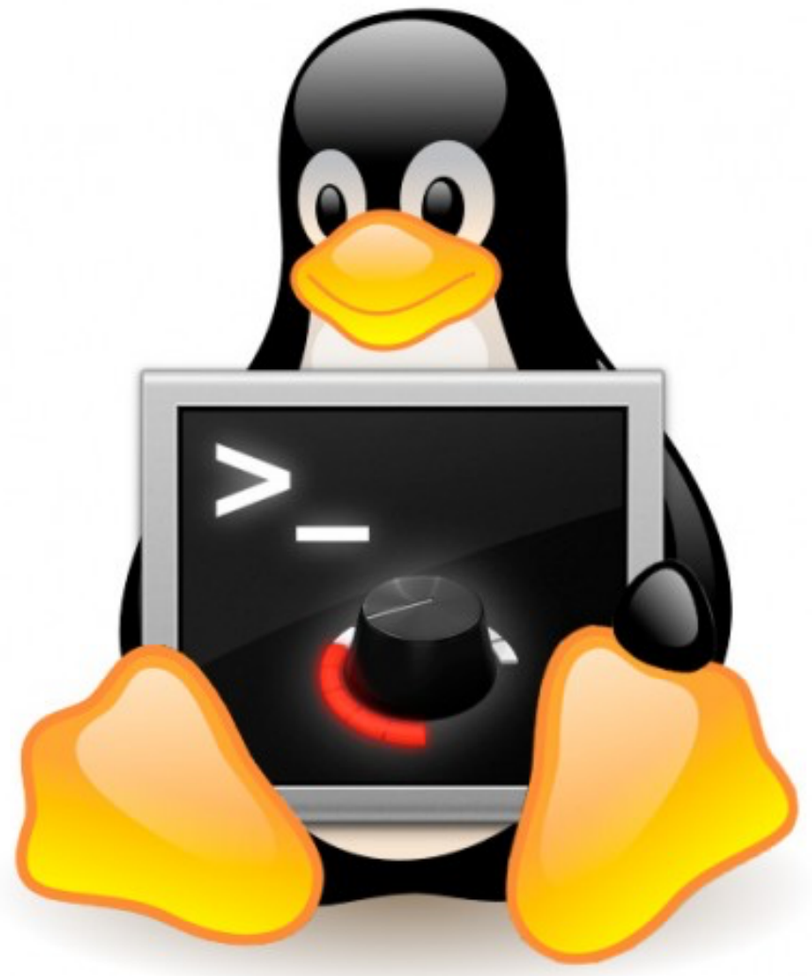
BIO 294: Bioinformatics for comparative and evolutionary genomics

Nov. 2019

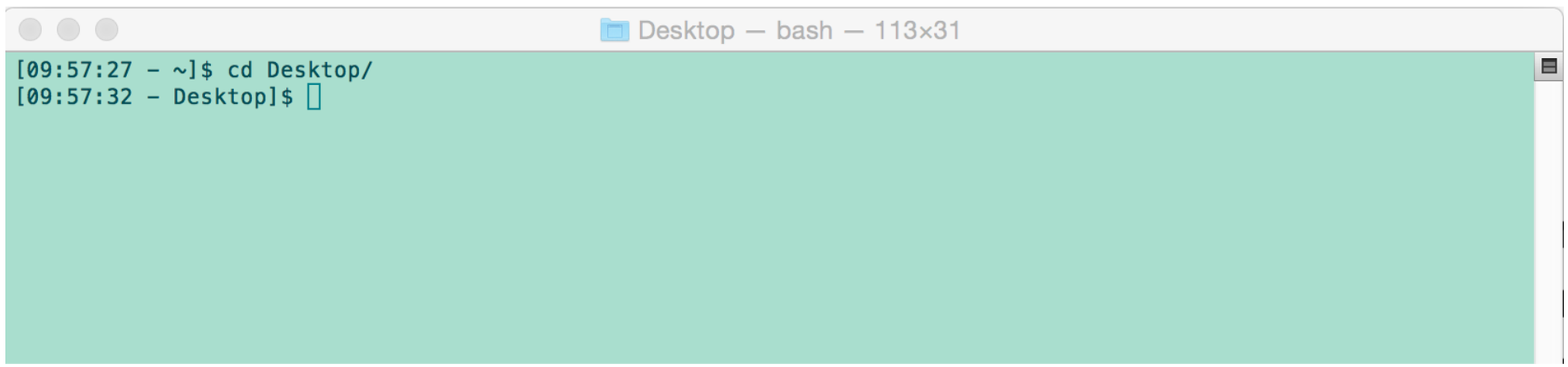
University of Zurich[UZH]

How to use a terminal ?



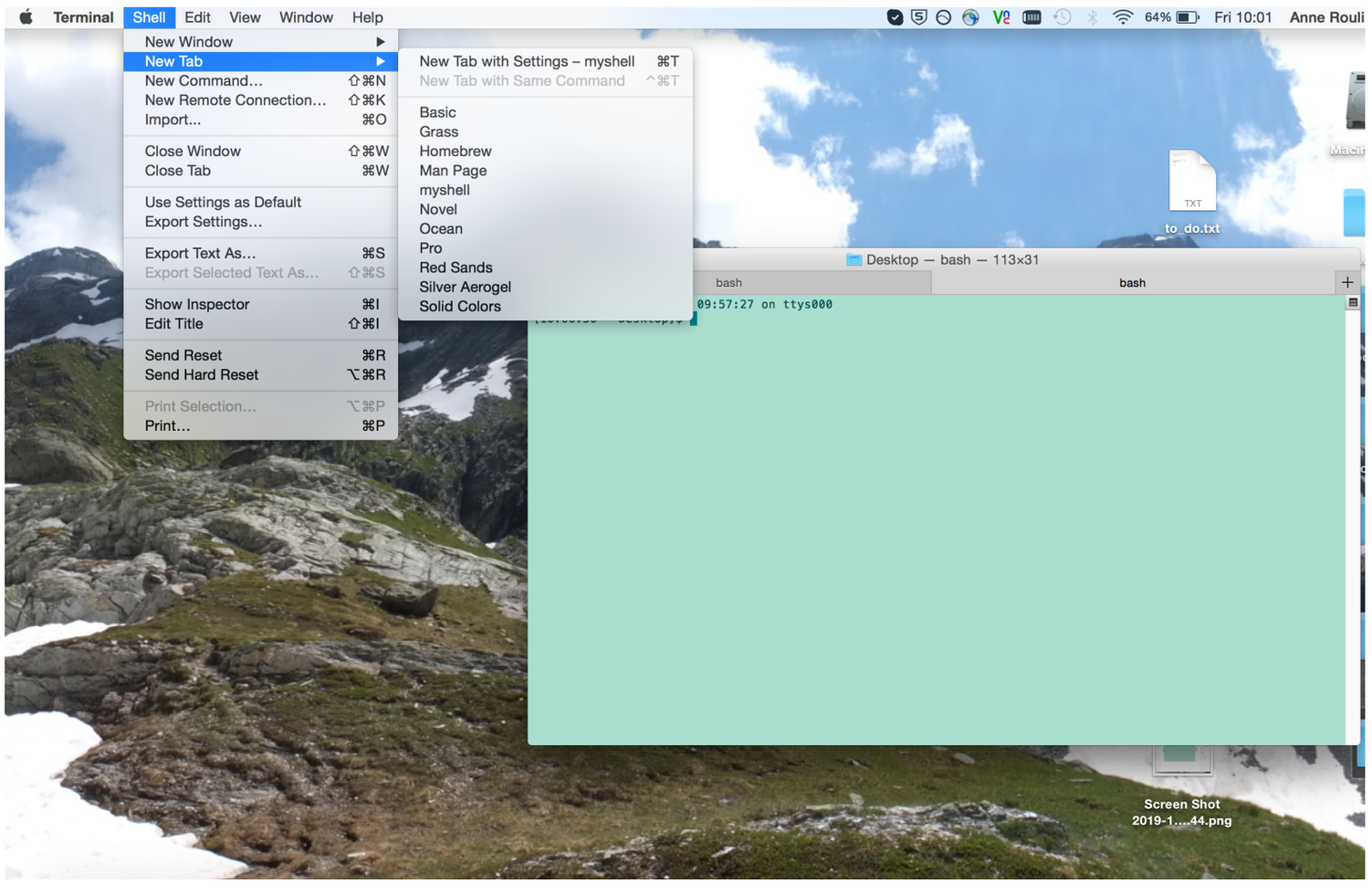
The **shell** is a program which processes commands and returns output. Most shells also manage foreground and background processes and command line editing. These features are standard in bash, the most common shell in modern linux systems.

A **terminal** refers to a wrapper program which runs a shell.

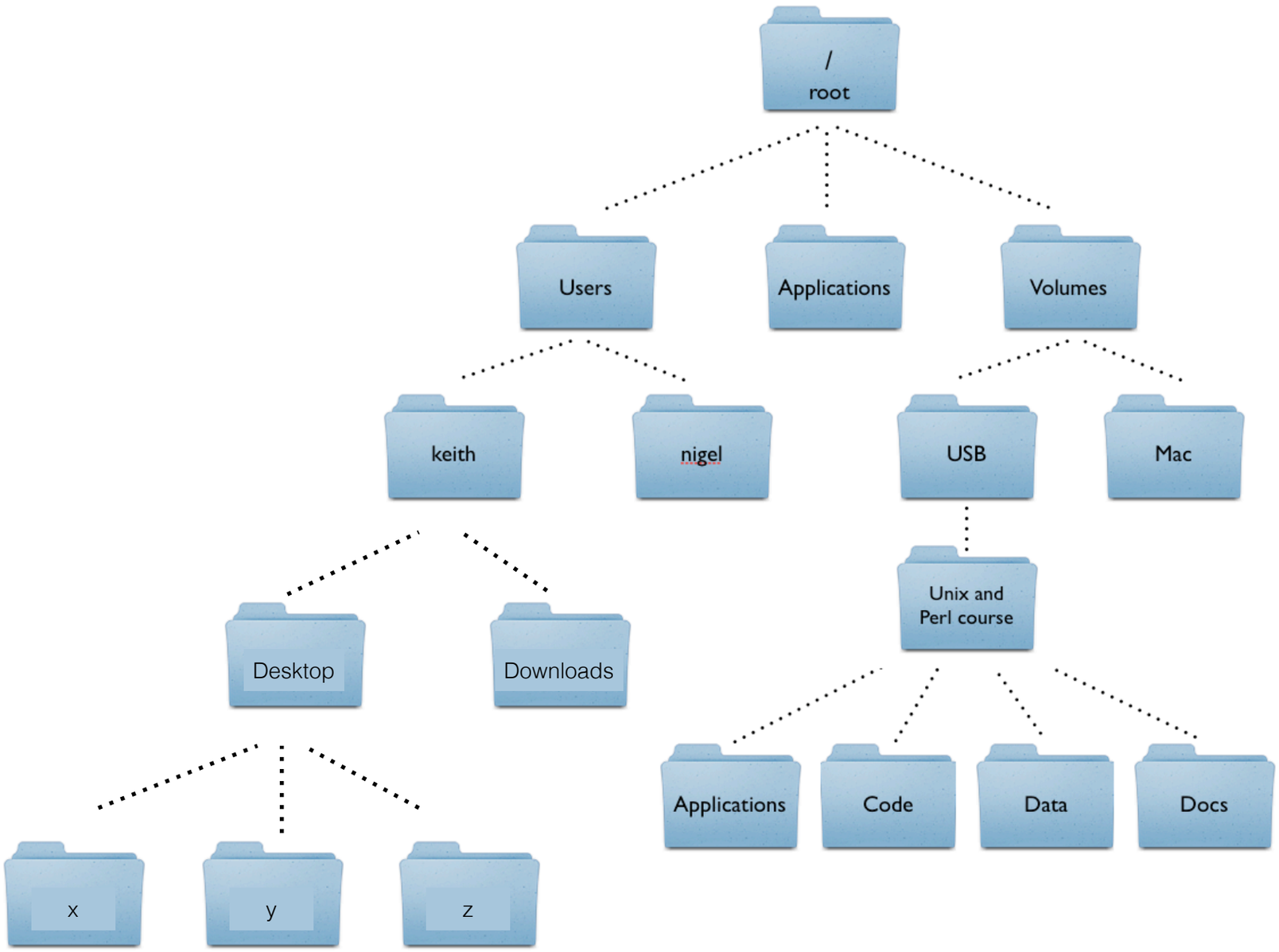


You can have:

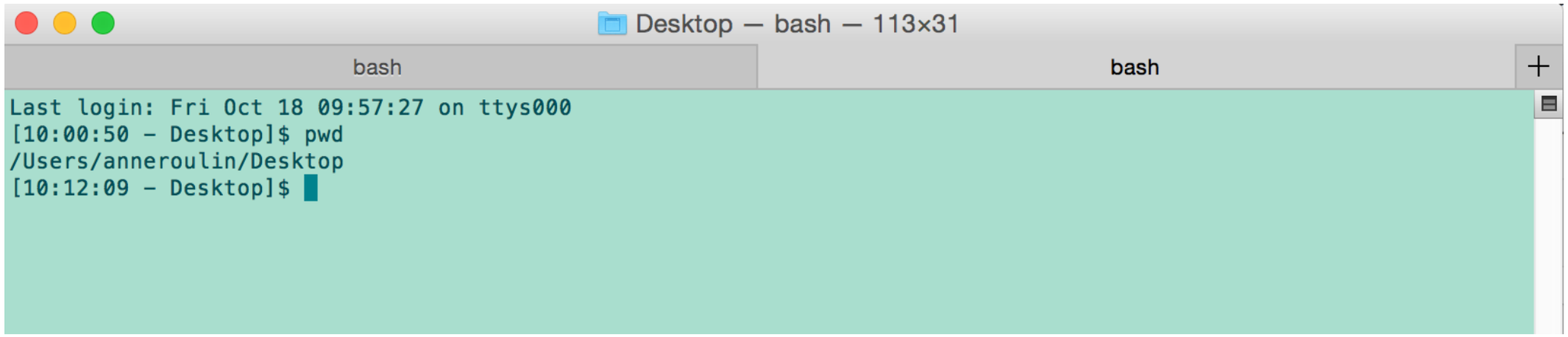
- have multiple terminal windows on screen (see ‘Shell’ menu)
- have multiple tabs open within each window. There will be many situations where it will be useful to have multiple terminals.



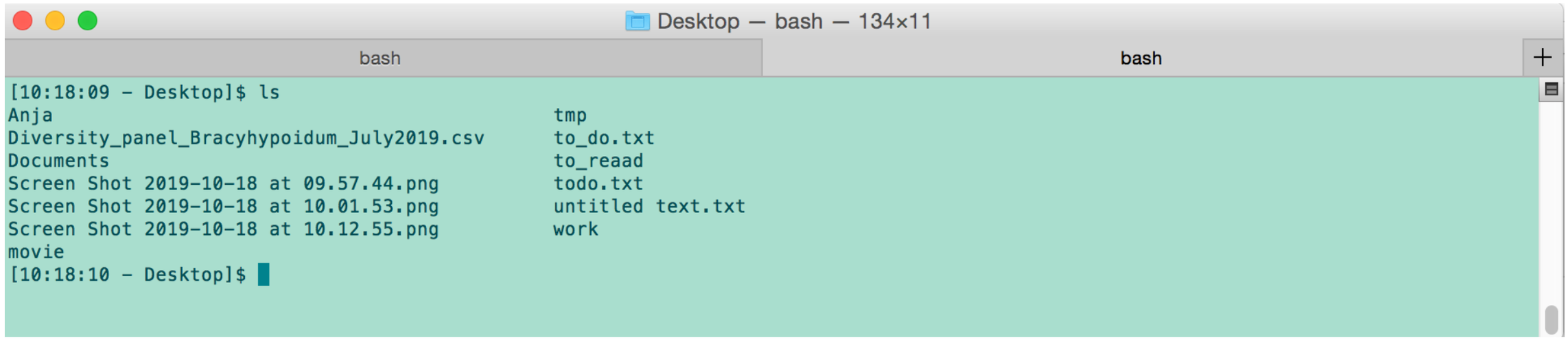
Unix keeps files arranged in a hierarchical structure



Finding out where you are: pwd (print working directory)



List the content of your directory: ls



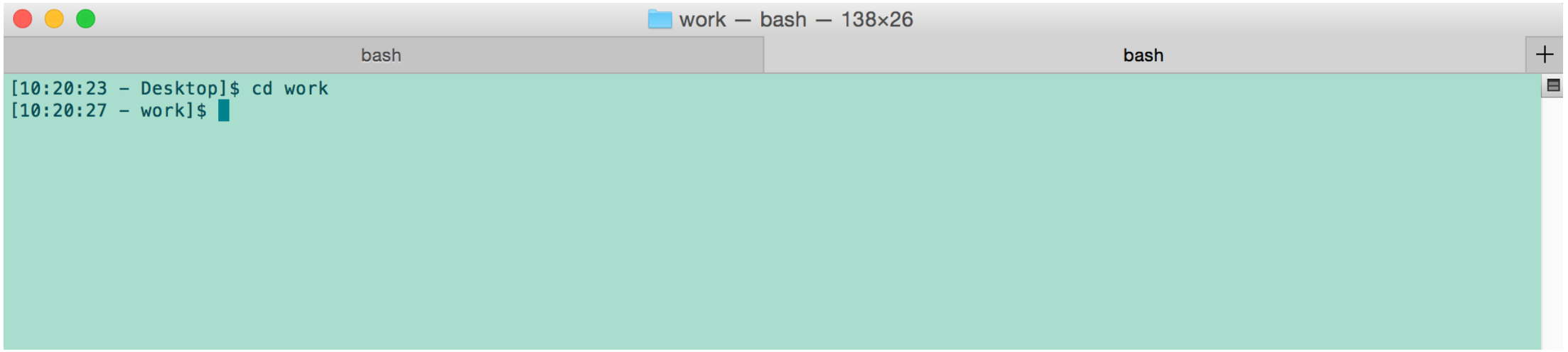
List the content of your directory: ll

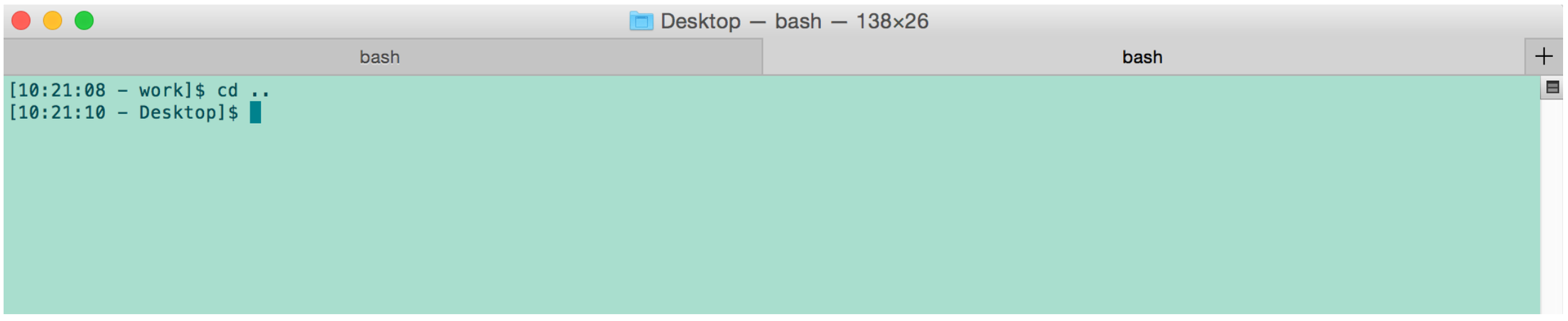
Desktop — bash — 138×26

bash | bash

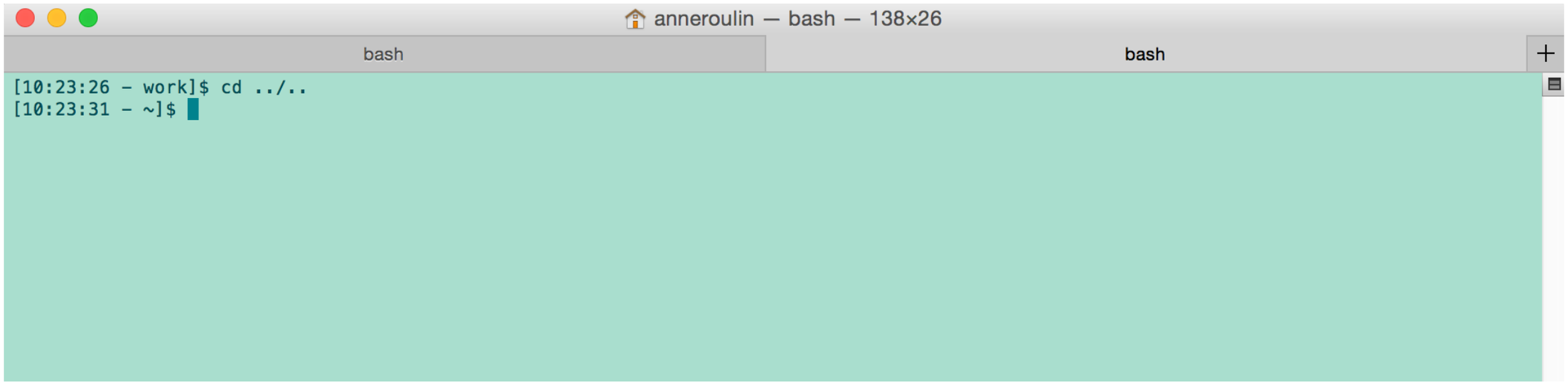
```
[10:18:15 - Desktop]$ ll
total 53128
drwx------+ 22 anneroulin  staff   748B Oct 18 10:18 .
drwxr-xr-x+ 86 anneroulin  staff   2.9K Oct  8 10:56 ..
-rw-r--r--@  1 anneroulin  staff    18K Oct 18 10:18 .DS_Store
-rw-r--r--   1 anneroulin  staff    20M Sep 22  2017 .RData
-rw-r--r--   1 anneroulin  staff     0B Mar  7  2018 .Rapp.history
-rw-r--r--   1 anneroulin  staff    24K Aug 10  2018 .Rhistory
-rw-r--r--   1 anneroulin  staff    49B Dec  1  2016 .bash_profile
-rw-r--r--   1 anneroulin  staff     0B Mar 11  2015 .localized
drwxr-xr-x  23 anneroulin  staff   782B Oct 18 09:39 Anja
-rw-r--r--@  1 anneroulin  staff    81K Jul 26 12:52 Diversity_panel_Bracyhypoidum_July2019.csv
drwx------  33 anneroulin  staff   1.1K Apr 10  2019 Documents
-rw-r--r--@  1 anneroulin  staff   531K Oct 18 09:57 Screen Shot 2019-10-18 at 09.57.44.png
-rw-r--r--@  1 anneroulin  staff   4.1M Oct 18 10:01 Screen Shot 2019-10-18 at 10.01.53.png
-rw-r--r--@  1 anneroulin  staff   842K Oct 18 10:12 Screen Shot 2019-10-18 at 10.12.55.png
-rw-r--r--@  1 anneroulin  staff   659K Oct 18 10:18 Screen Shot 2019-10-18 at 10.18.12.png
drwxr-xr-x   6 anneroulin  staff   204B Oct 17 20:17 movie
drwxr-xr-x   3 anneroulin  staff   102B Oct 15 06:09 tmp
-rw-r--r--@  1 anneroulin  staff   191B Apr 13  2019 to_do.txt
drwxr-xr-x  25 anneroulin  staff   850B Sep  4 19:18 to_reaad
-rw-r--r--@  1 anneroulin  staff   1.9K Aug 24 14:19 todo.txt
-rw-r--r--@  1 anneroulin  staff   1.3K Jul 14 14:23 untitled text.txt
drwxr-xr-x  29 anneroulin  staff   986B Jul 19 13:03 work
[10:18:16 - Desktop]$
```

Changing directory: cd





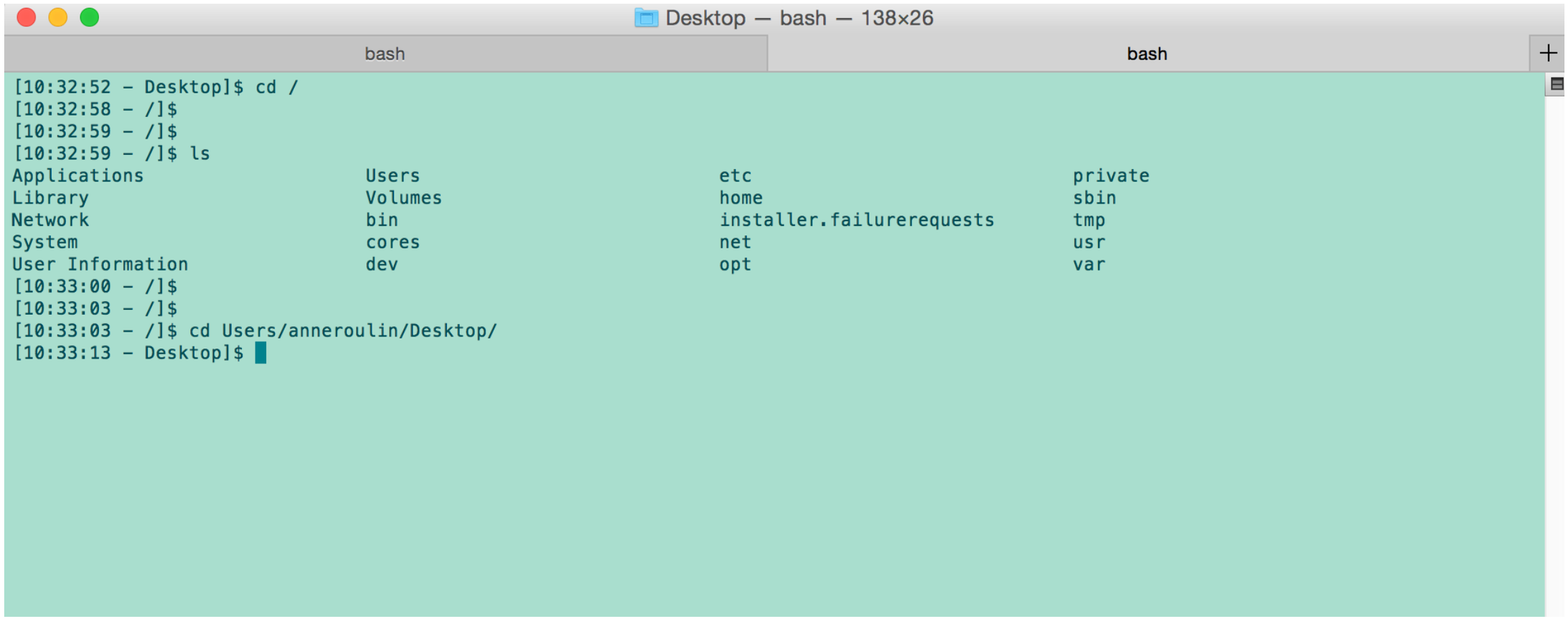
Changing directory: cd



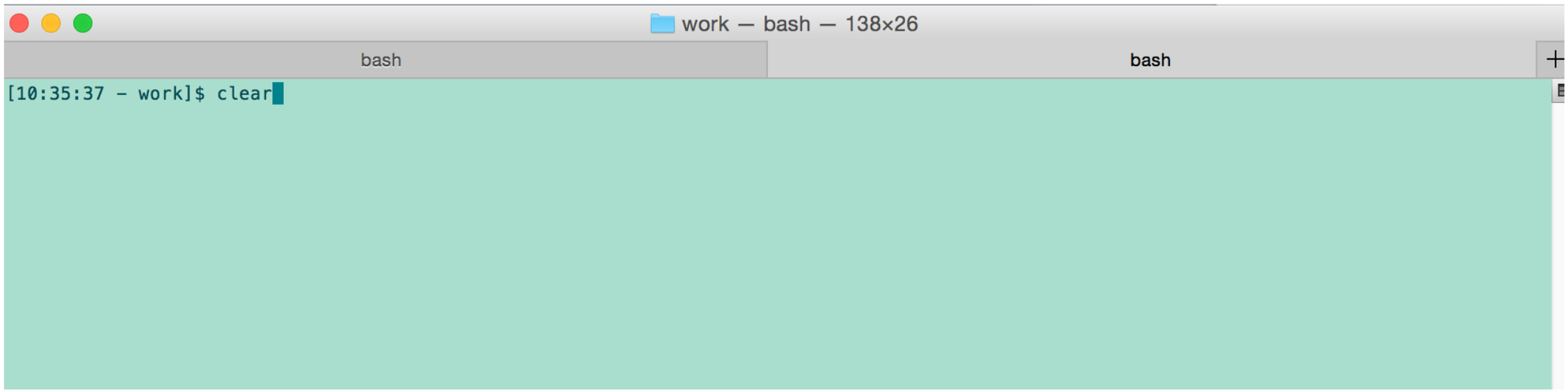
Nb: use 

to complete the path you are looking for

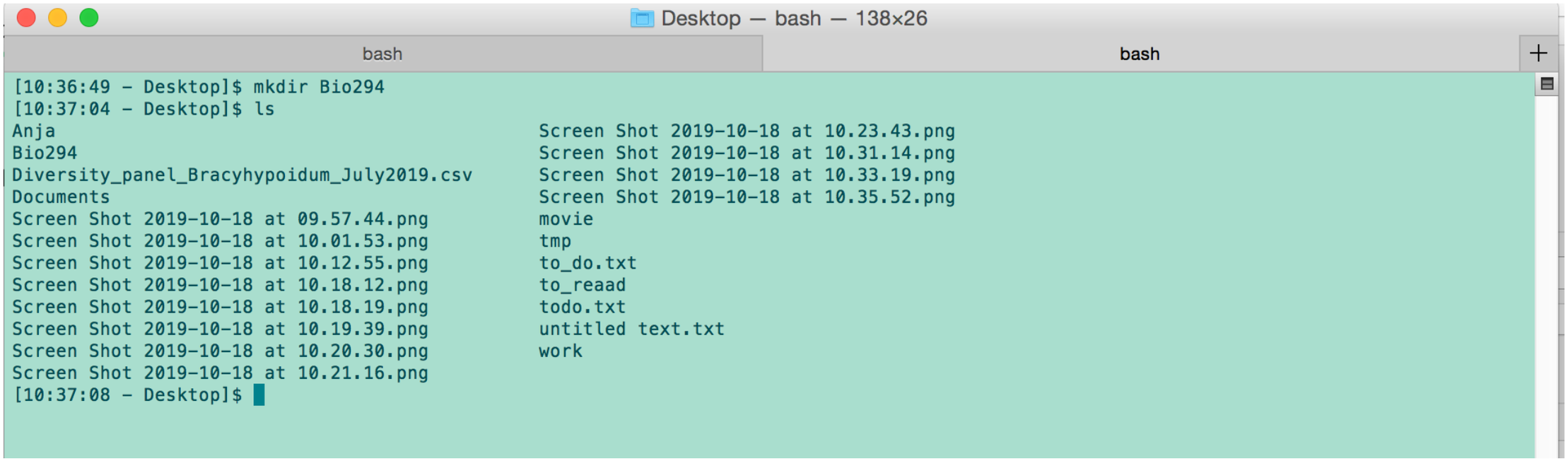
Back to the root: cd /



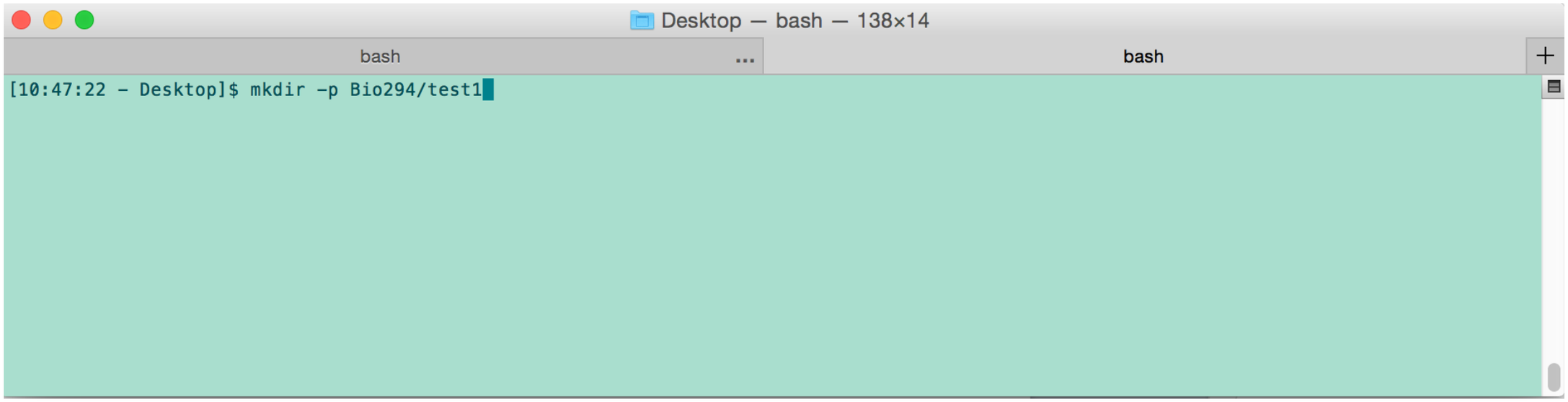
Clean your terminal: clear

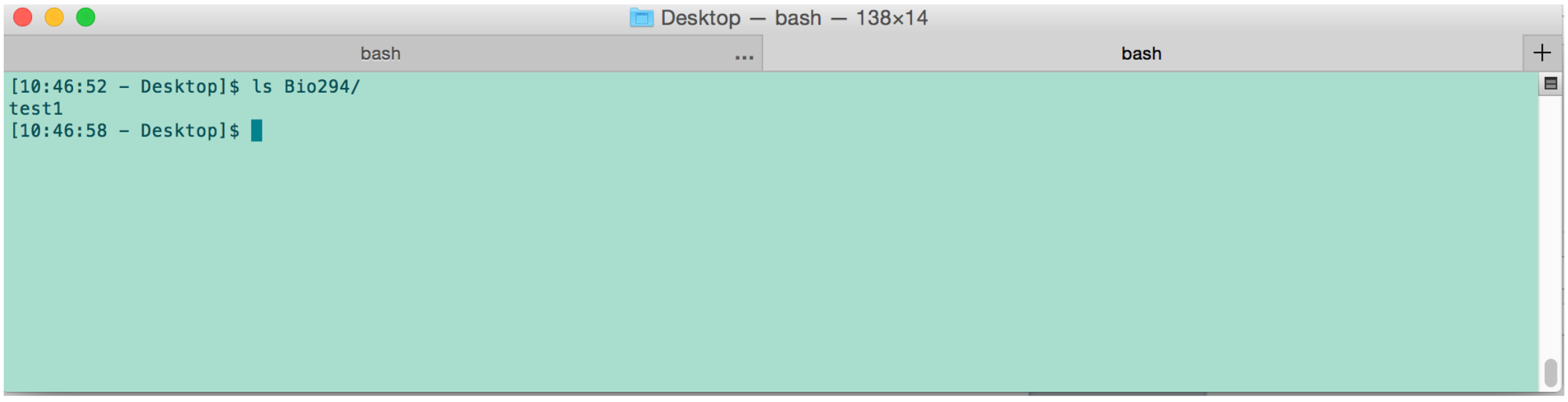


Make a new directory: mkdir

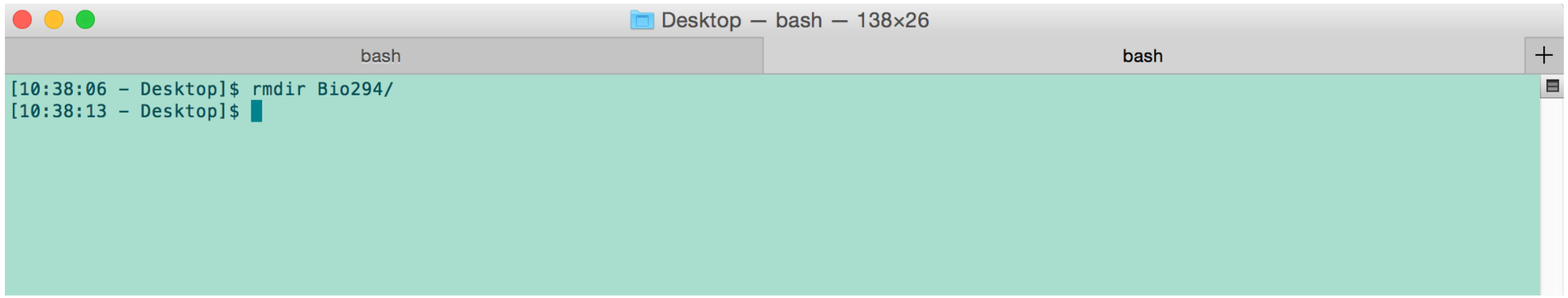


Make a new directory within a new directory: mkdir -p



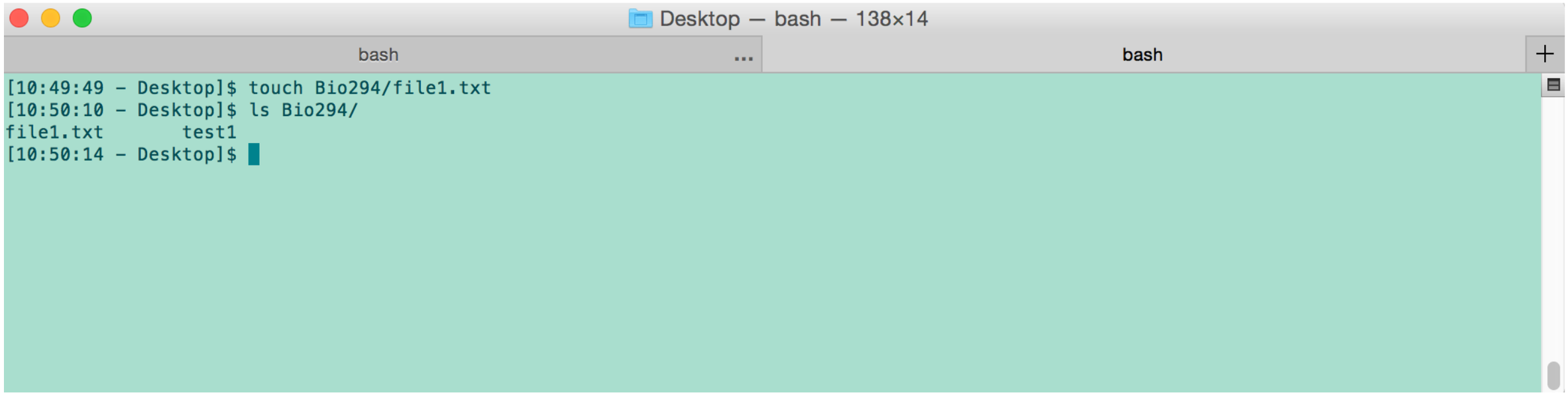


Delete a directory: mkdir

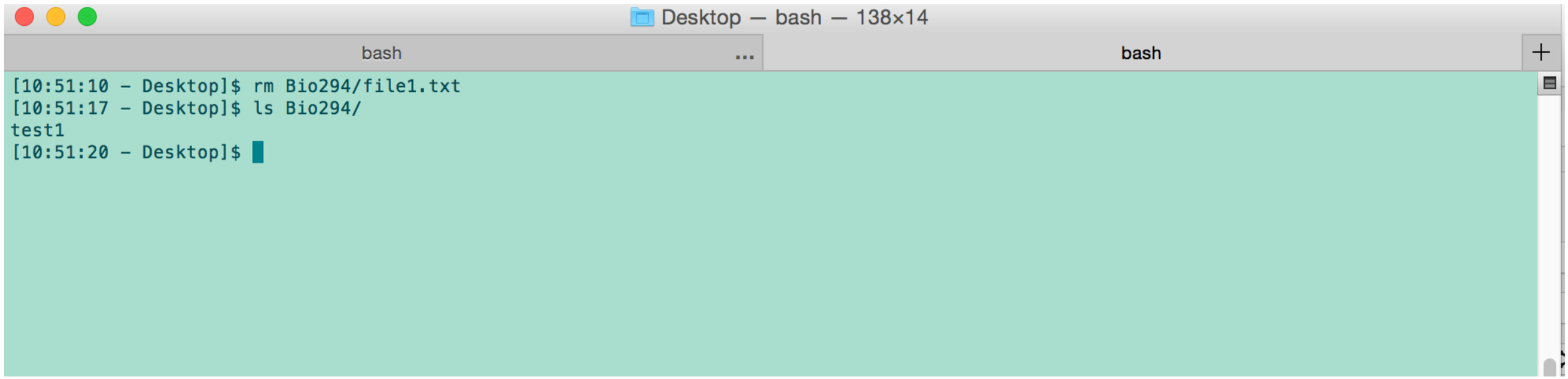


You will not be asked “are you sure you want to remove your directory”

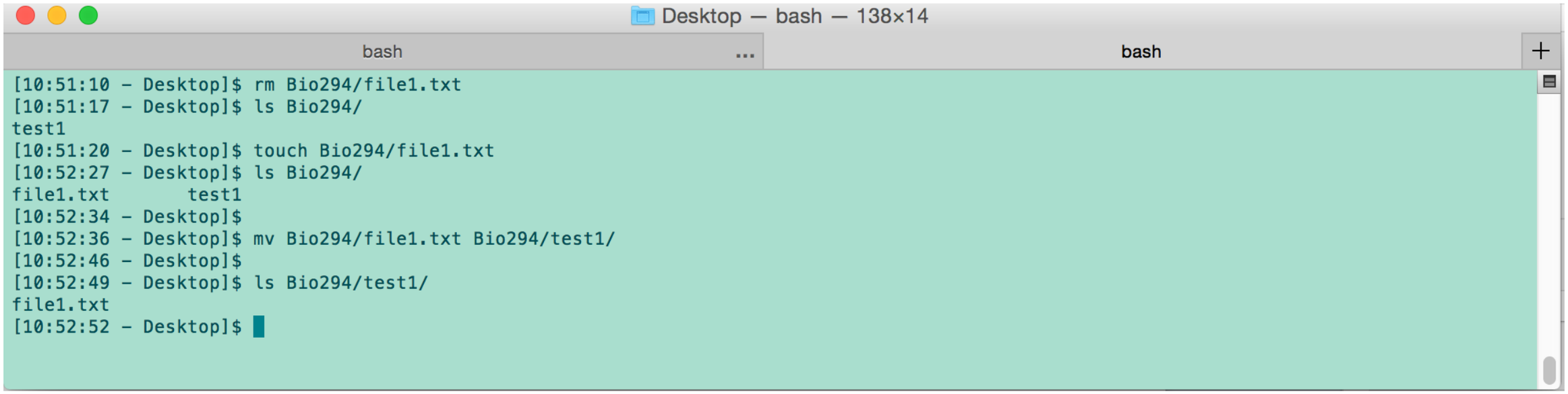
Create a new file: touch



Delete a file: rm



Move a file: mv



BIO 294 University of Zurich[UZH]

Create additional files…

Desktop — bash — 138×14

bash … bash

```
[10:49:49 - Desktop]$ touch Bio294/file1.txt
[10:50:10 - Desktop]$ ls Bio294/
file1.txt       test1
[10:50:14 - Desktop]$
```

… and move them all in one go

Desktop — bash — 138×14

bash … bash

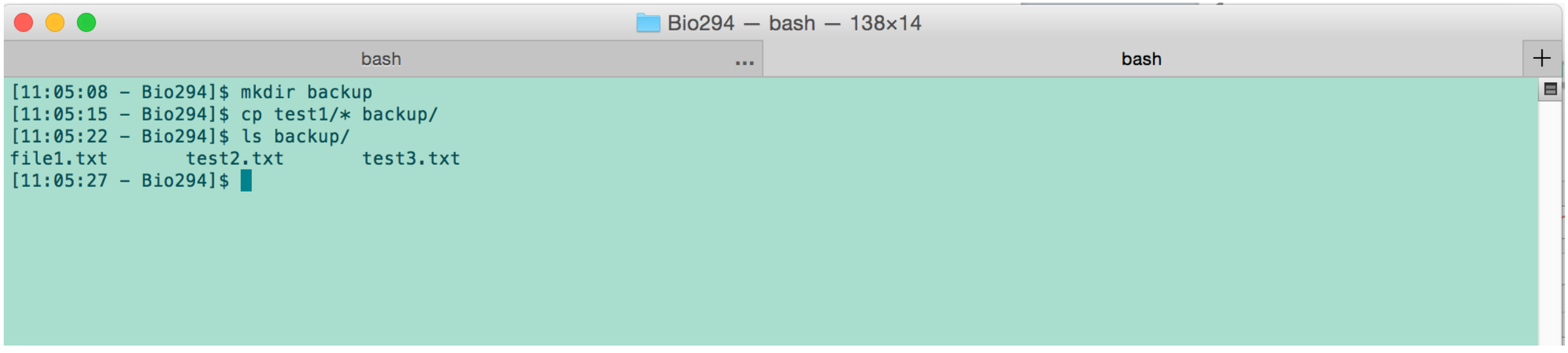
```
[10:57:14 - Desktop]$ touch Bio294/test2.txt
[10:57:16 - Desktop]$ touch Bio294/test3.txt
[10:57:19 - Desktop]$
[10:57:20 - Desktop]$ mv Bio294/*.txt Bio294/test1/
[10:57:30 - Desktop]$
[10:57:31 - Desktop]$
[10:57:31 - Desktop]$ ls Bio294/test1/
file1.txt       test2.txt       test3.txt
[10:57:36 - Desktop]$
```

What is the * doing?

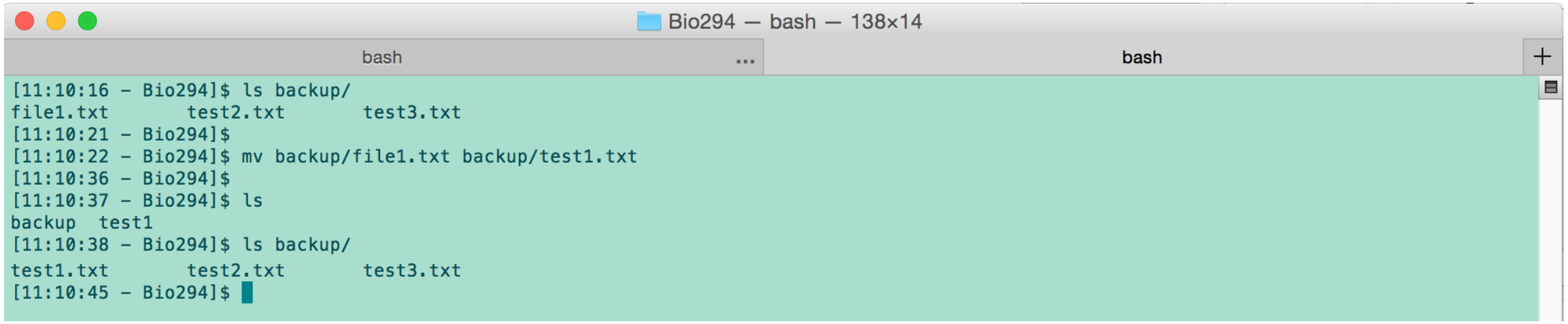
Work safely: rm and rmdir and “dangerous” command line.

You can always save your raw data in an additional folder that you will never touch.

To copy your file to a new folder: cp

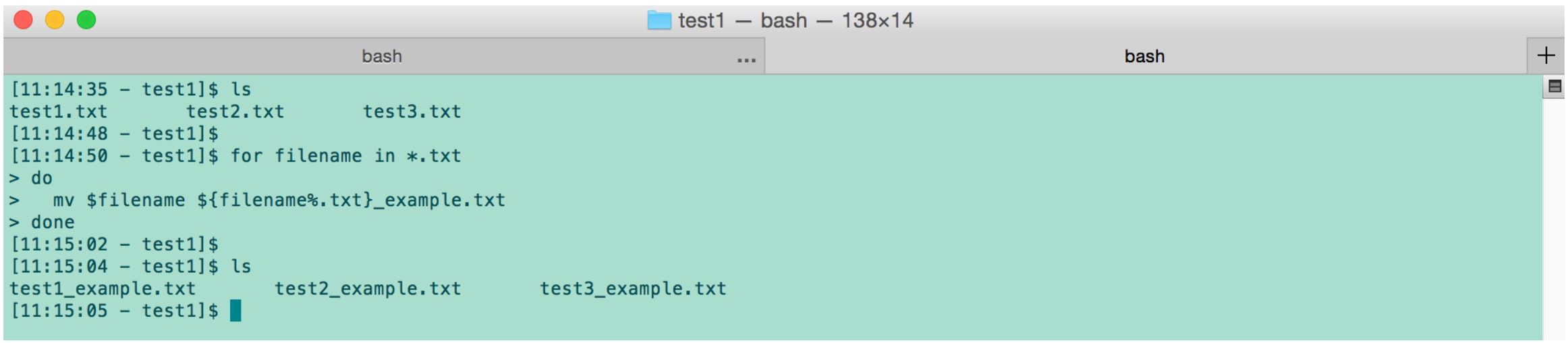


Rename files: mv



Rename all files in folder:

```
for filename in *.txt
do
    mv $filename ${filename%.txt}_example.txt
done
```



Can you explain how the loop is working?

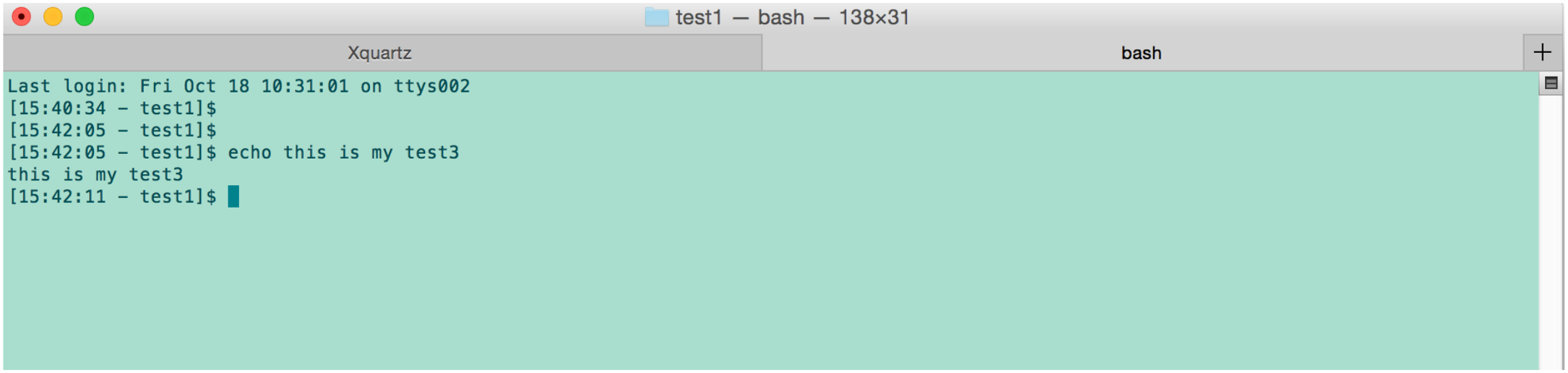
Output redirection: >

Desktop — bash — 138×31

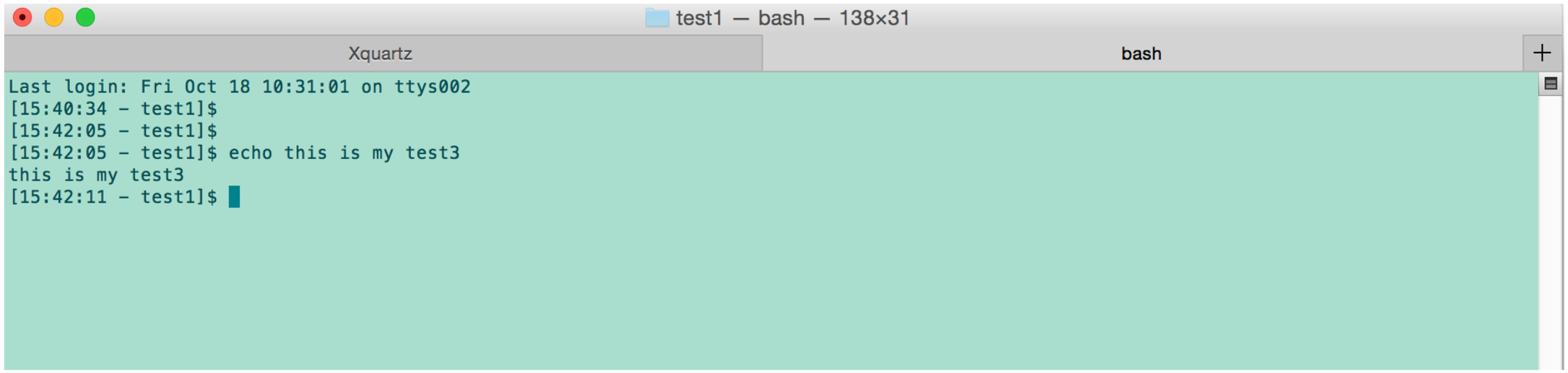
bash ... bash

```
[12:47:07 - Desktop]$ ls > myfiles_list.txt
[12:47:35 - Desktop]$ cat myfiles_list.txt
Anja
Bio294
Diversity_panel_Bracyhypoidum_July2019.csv
Documents
Screen Shot 2019-10-18 at 11.15.08.png
Screen Shot 2019-10-18 at 12.43.00.png
movie
myfiles_list.txt
tmp
to_do.txt
to_reaad
todo.txt
untitled text.txt
work
[12:47:46 - Desktop]$
```

Writes arguments to standard output: echo

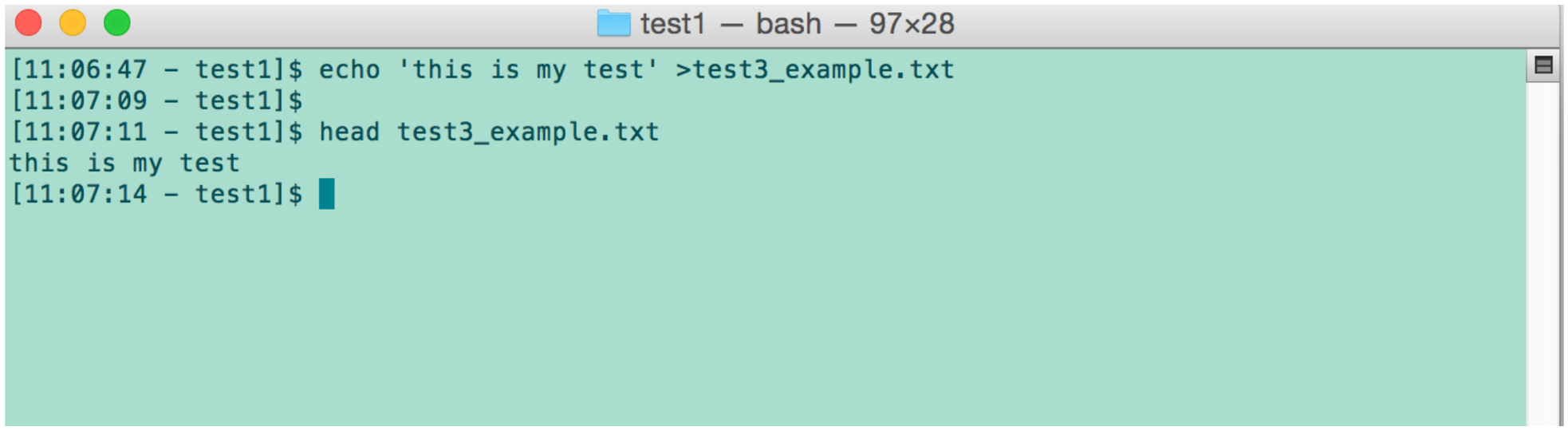


Writes arguments to standard output: echo



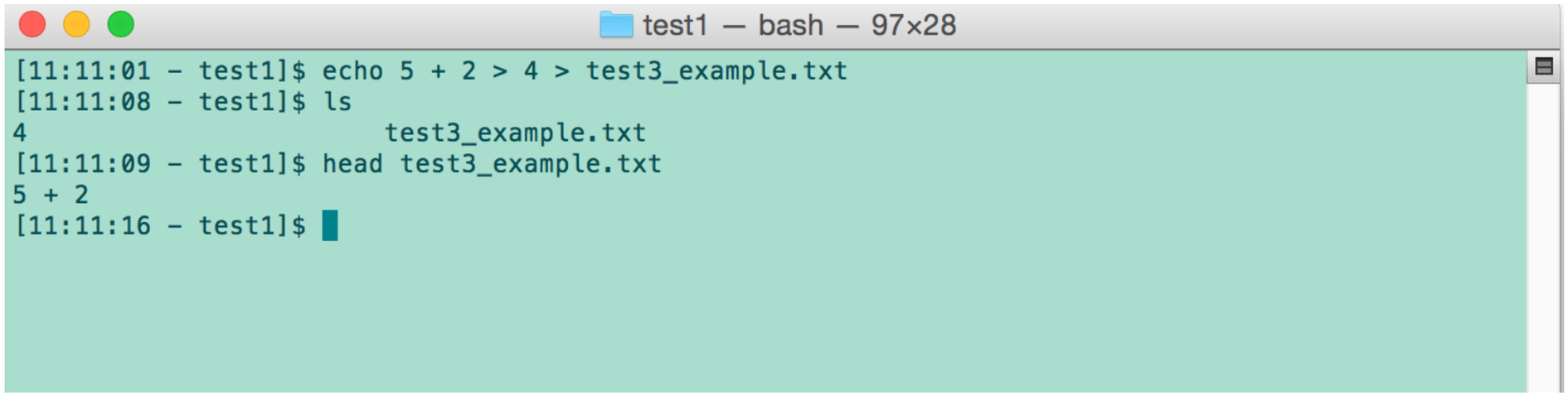
How would you write “this is my test 3” in the file test3_example.txt?

Writes arguments to standard output: echo

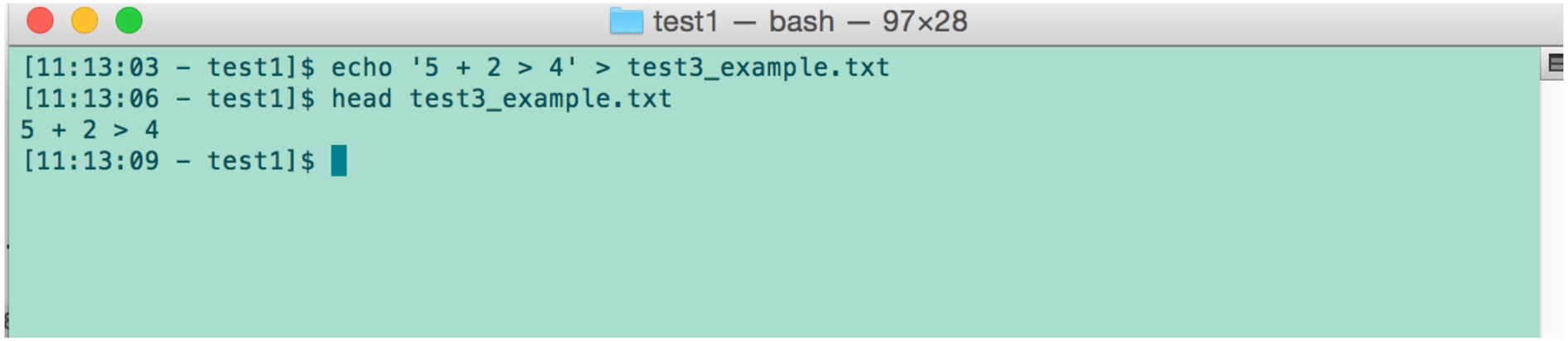


It is a good practice to use quotes to prevent the shell from interpreting a character.

for example, if you want to write 5 + 2 > 4 to the file test3_example.txt



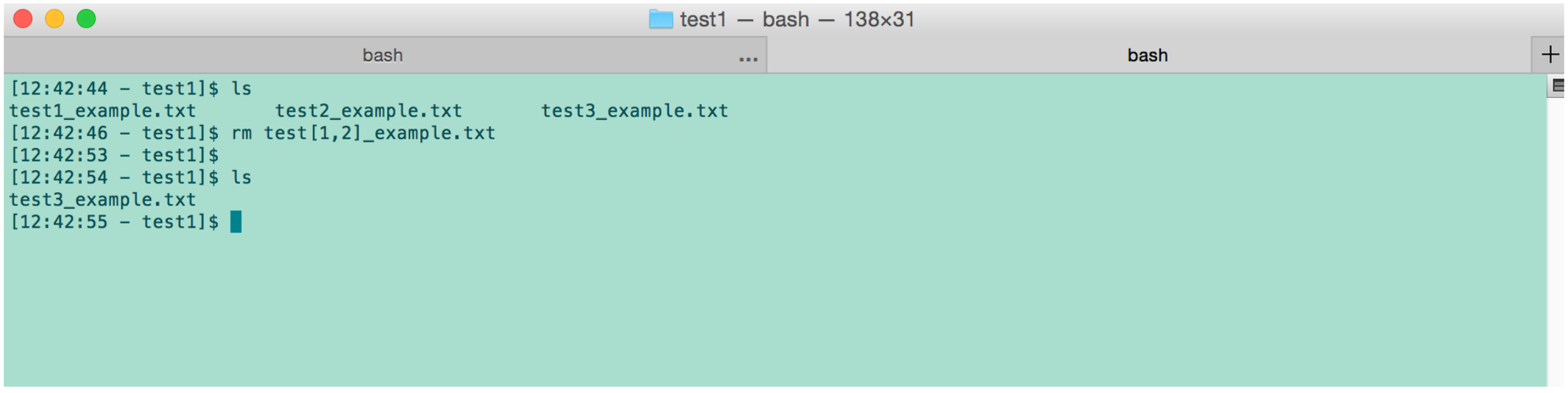
Now, protect your test with ‘’



For more on quotes, see https://www.gnu.org/software/bash/manual/html_node/Quoting.html or https://bash.cyberciti.biz/guide/Quoting

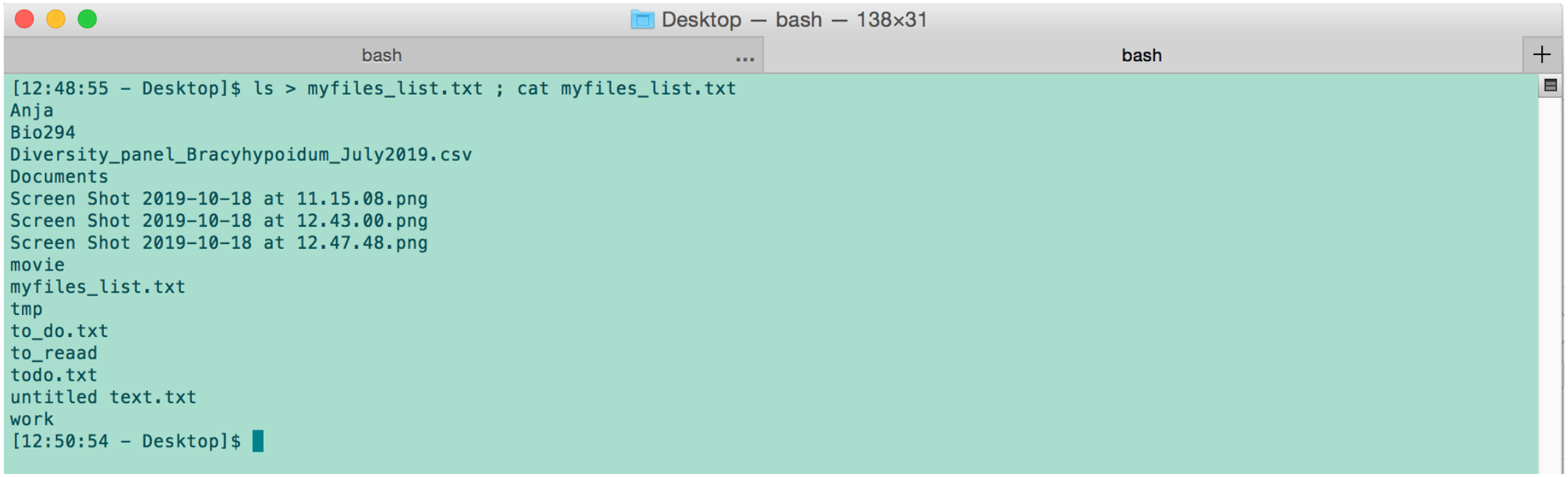
Few other tips:

Select a pattern that contains specific characters with “ [] “



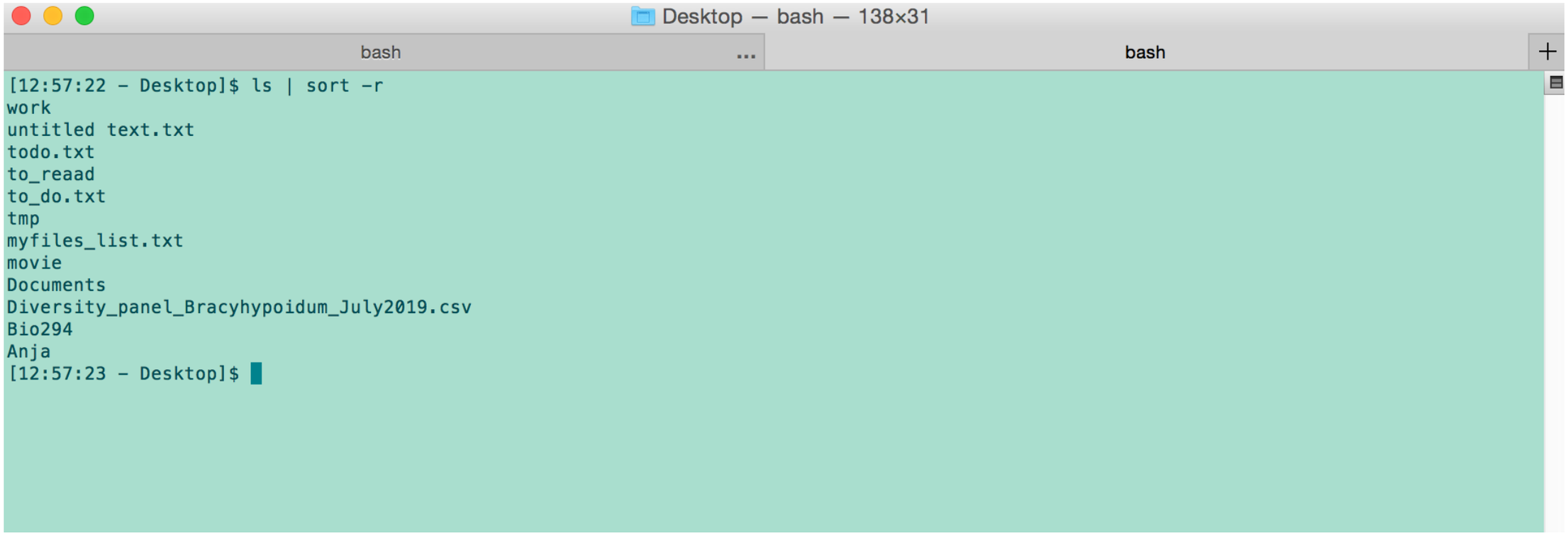
Few other tips:

Use “ ; “ as a command separator



Few other tips:

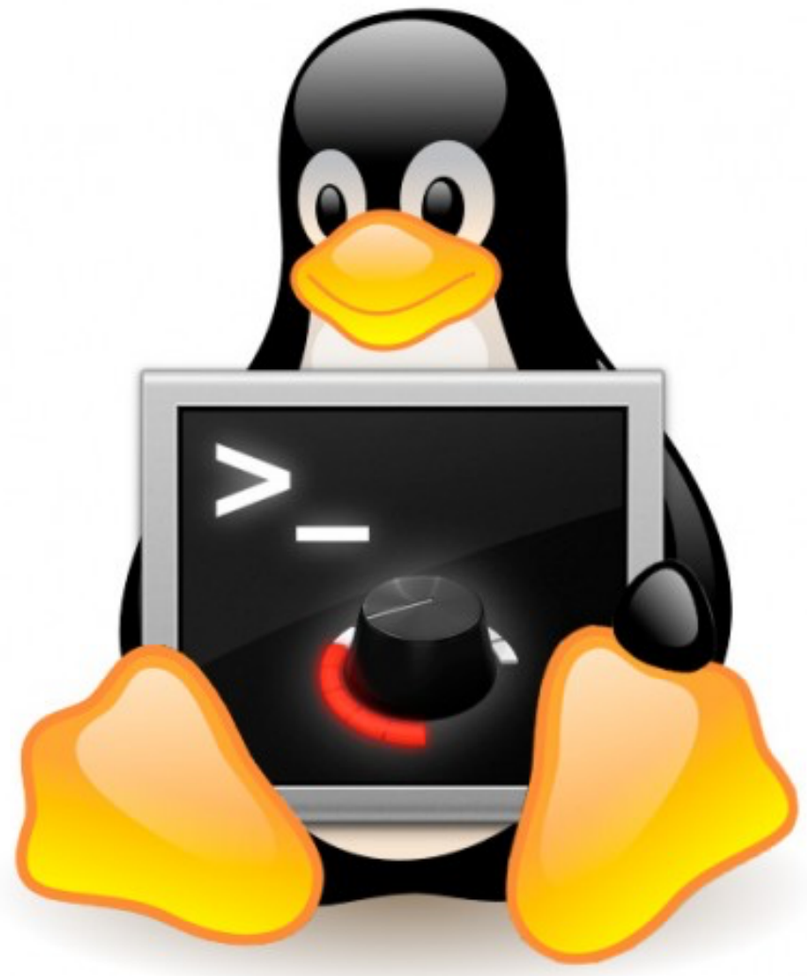
“ | “ : A “pipe” chains commands together. It takes the output from one command and feeds it to the next as input



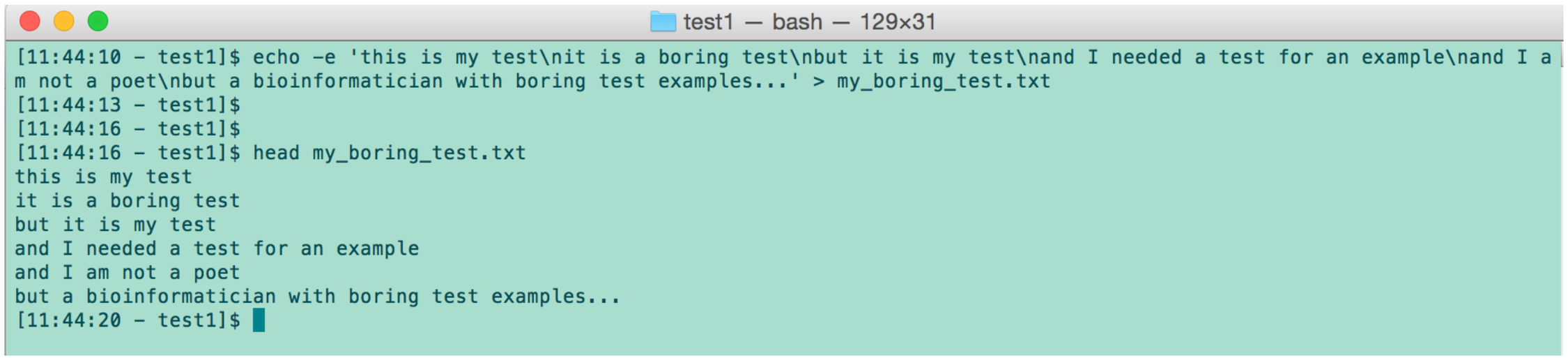
Few other tips:

ctrl + A : bring the cursor at the beginning of your command
ctrl + E : bring the cursor at the end of your command

How to work with files ?



Let's create a file containing the following text: echo -e 'this is my test\nit is a boring test\nbut it is my test\nand I needed a test for an example\nand I am not a poet\nbut a bioinformatician with boring test examples...' > my_boring_test.txt



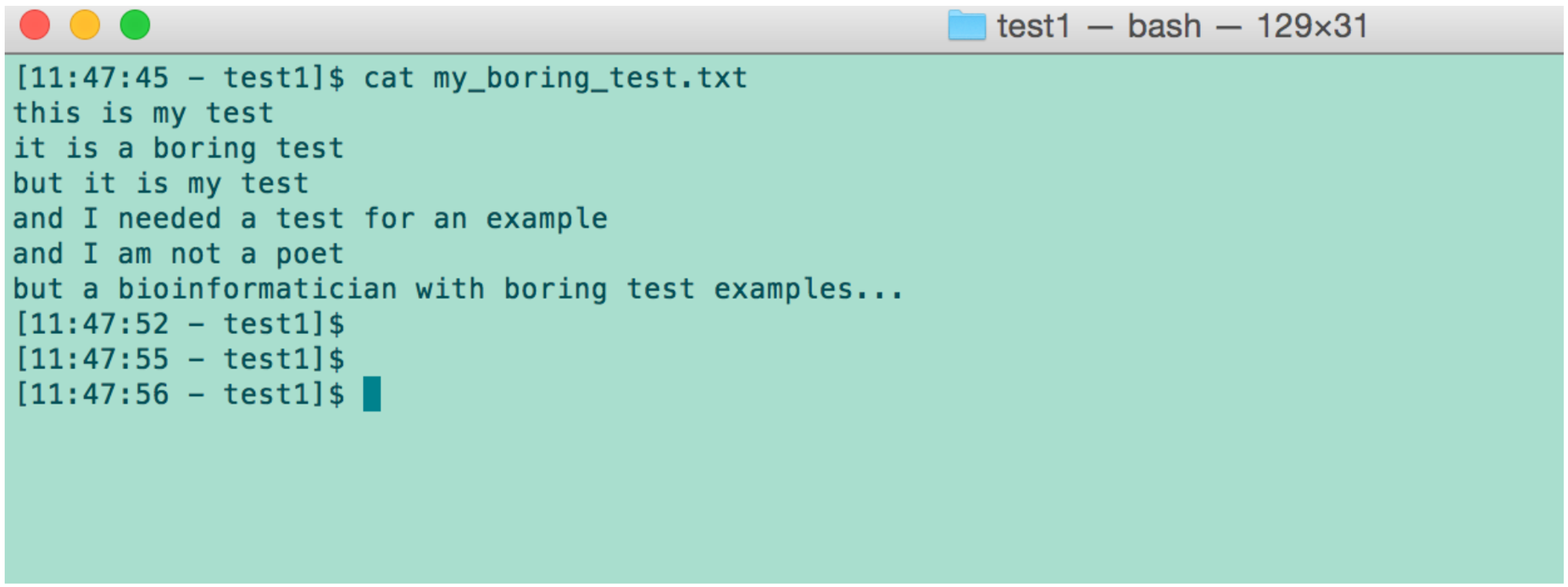
- \n = new line
- The '-e' option in Linux acts as interpretation of escaped characters that are backslashed.

University of Zurich[UZH]

Cat

Cat can be used for the following purposes:

- **Display text files on screen.**
- Copy text files.
- Combine text files.



NB: compared to “head”, cat display the entire content of the file

University of Zurich[UZH]

Cat

Cat can be used for the following purposes:

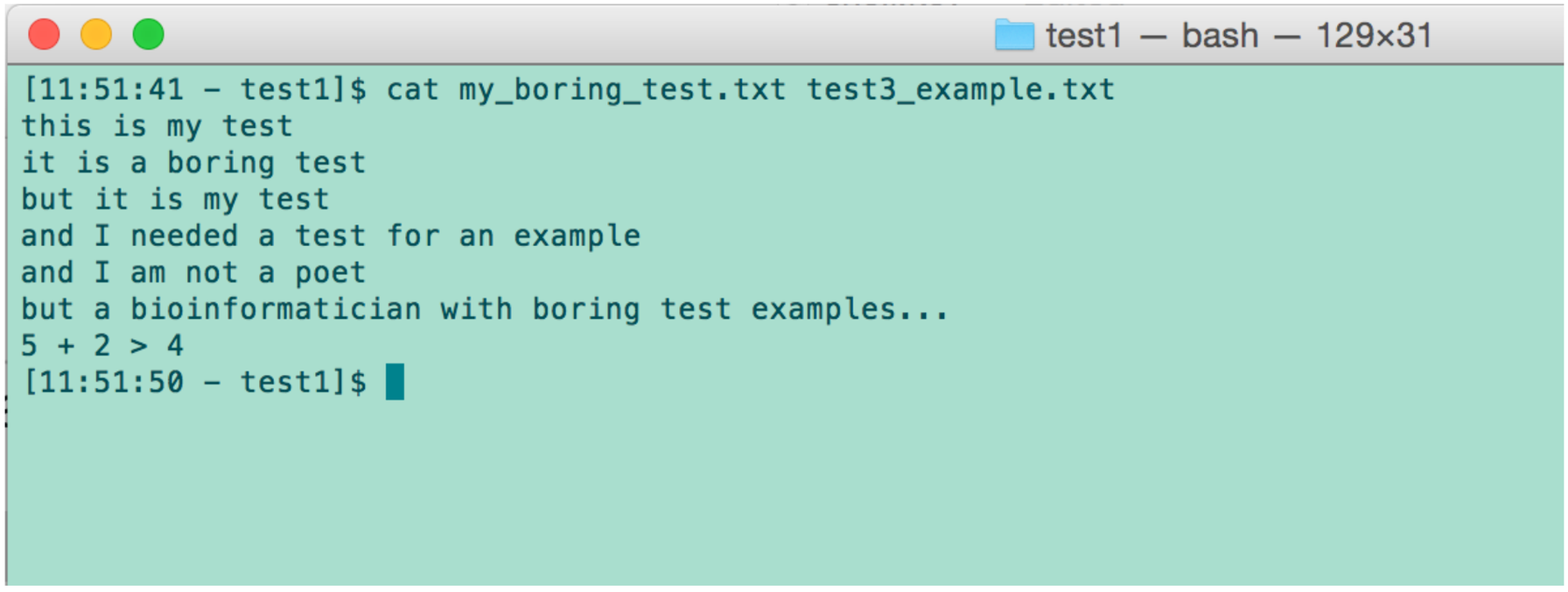
- Display text files on screen.
- **Copy text files.**
- Combine text files.

University of Zurich[UZH]

Cat

Cat can be used for the following purposes:

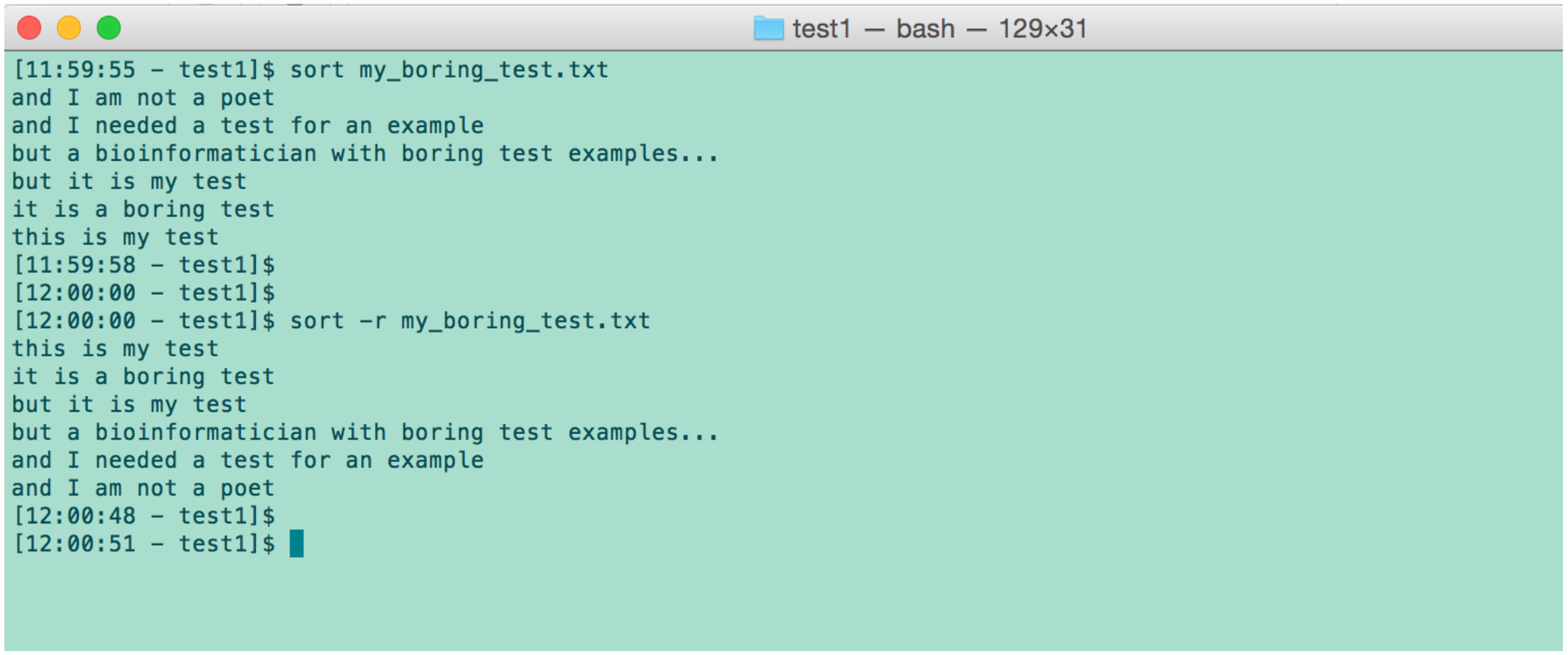
- Display text files on screen.
- Copy text files.
- **Combine text files.**



University of Zurich[UZH]

Sort

Can be used to sort files alphabetically or numerically



For more option on sort:
https://www.softwaretestinghelp.com/unix-sort-command/

University of Zurich[UZH]

SED

- SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
- SED command in unix supports regular expression which allows it perform complex pattern matching.

University of Zurich[UZH]

SED

Modify our previous text: echo -e 'this is my test\nit is a boring test\nbut it is my test and I needed a test for an example\nand I am not a poet\nbut a bioinformatician with boring test examples...' > my_boring_test.txt

University of Zurich[UZH]

SED

Replacing (substituting) string: sed s

```
test1 — bash — 129×31
[12:13:05 - test1]$ sed 's/test/Test/' my_boring_test.txt
this is my Test
it is a boring Test
but it is my Test and I needed a test for an example
and I am not a poet
but a bioinformatician with boring Test examples...
[12:13:10 - test1]$
```

sed -s: replace the first occurrence in each line

University of Zurich[UZH]

Little exercises

Can you explain these commands?

```
sed 's/test/Test/g' my_boring_test.txt
```

```
sed '2 s/test/Test/' my_boring_test.txt
```

```
sed '2,4 s/test/Test/' my_boring_test.txt
```

```
sed 's/test/Test/2' my_boring_test.txt
```

University of Zurich[UZH]

SED

Delete the nth line: d

```
test1 — bash — 129×31
[12:24:58 - test1]$ sed '4d' my_boring_test.txt
this is my test
it is a boring test
but it is my test and I needed a test for an example
but a bioinformatician with boring test examples...
[12:25:10 - test1]$
```

Or a line with a specific pattern

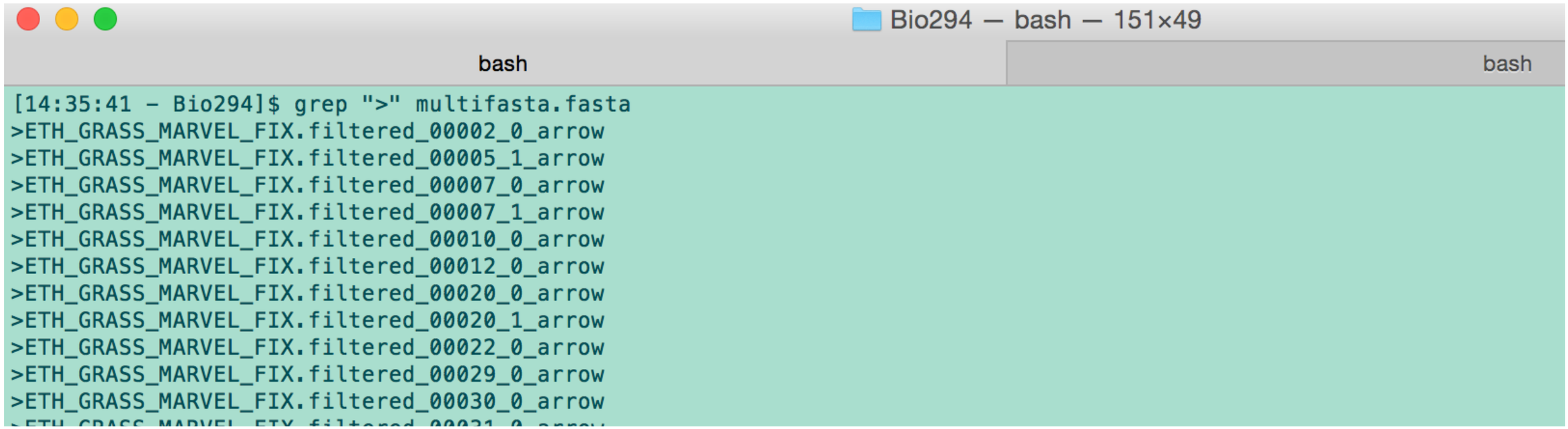
```
test1 — bash — 129×31
[12:27:11 - test1]$ sed '/poet/d' my_boring_test.txt
this is my test
it is a boring test
but it is my test and I needed a test for an example
but a bioinformatician with boring test examples...
[12:27:58 - test1]$
```

University of Zurich[UZH]

grep

grep is a command-line utility for searching plain-text data sets for **lines** that match a regular expression.

```
grep [OPTION...] PATTERNS [FILE...]
```



University of Zurich[UZH]

Little exercises

You will find more information about the different options available at: http://man7.org/linux/man-pages/man1/grep.1.html

How many lines contains the pattern “ATCG”

How many lines contains the pattern “atcg”

How many lines contains the pattern “ATCG” or “atcg”

Write all the headers of the fasta sequences in a new file

University of Zurich[UZH]

Answers

You will find more information about the different options available at: http://man7.org/linux/man-pages/man1/grep.1.html

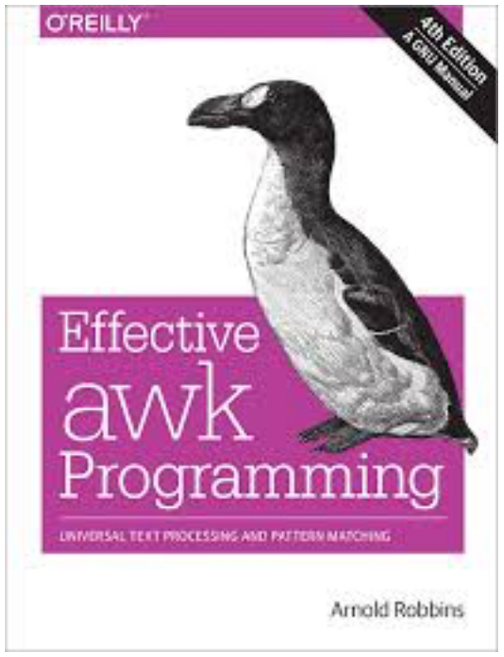
```
grep -c “ATCG” multifasta.fasta

grep -c “atcg” multifasta.fasta

grep -ci “atcg” multifasta.fasta

grep “>” multifasta.fasta > header.txt
```

University of Zurich[UZH]

awk

AWK is a domain-specific language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems.

pattern { action }



University of Zurich[UZH]

awk

```
awk '{print $1,$2,$3}' Supp_Table2.csv

awk '{print $1,$2,"data_set1"}' Supp_Table2.csv

awk '{if ($4 > 6)  {print $0}}' Supp_Table2.csv
```

Operator	Meaning
==	Is equal
!=	Is not equal to
>	Is greater than
>=	Is greater than or equal to
<	Is less than
<=	Is less than or equal to

University of Zurich[UZH]

awk

```
awk '$1 ~ "Bd1" {print $0}' Supp_Table2.csv
```

Operator	Meaning
~	Matches
!~	Doesn't match

Little exercises

- How many uniq “term” are present in Supp_Table2.csv?
- Keep only the lines for which the count is > 5 AND the Pvalue < 0.05, and sort the table according to descending Pvalues
- add a column specifying whether the Pvalue is significant (< 0.05) or not_significant

University of Zurich[UZH]

Answers

```
awk '{print $8}' Supp_Table2.csv|sort -u|wc -l

awk '{if ($5 > 5 && $9 <0.05) {print $0}}' Supp_Table2.csv| sort -rk9,9

awk '{if ($9 > 0.005) {print $0, "significant"} else {print $0,
"not_significant"}}' Supp_Table2.csv
```

University of Zurich[UZH]

http://manuals.bioinformatics.ucr.edu/home/emboss

University of Zurich[UZH]

First step

We will use the file multifasta.fasta

Rename each header so that the term “eth_grass_marvel_fix.filtered_” is replaced by “scaffold”, and write the sequences in a new file (eg. multifasta_renamed.fasta

```
sed 's/ETH_GRASS_MARVEL_FIX.filtered_/scaffold/g' multifasta.fasta| sed
's/_0_arrow//g'|cat > multifasta_renamed.fasta
```

University of Zurich[UZH]

Infoseq

```
infoseq multifasta_renamed.fasta
```

```
infoseq multifasta_renamed.fasta -only -name -length
```

```
infoseq multifasta_renamed.fasta:scaffold*
```

Create a table that contains only the name of each sequences as well as their length and GC content.

University of Zurich[UZH]

Seqret

Performs sequence retrieval from databases, feature parsing, sequence and alignment reformatting.

```
seqret multifasta.fasta -sbegin 15 -send 50 -out seqret_test.fasta
```

```
seqretsplit multifasta.fasta
```

Extract the sequence of scaffold00002_0_arrow

Extract the sequence of the scaffold between 200 and 300, and write them in a new fasta file. Use only 1 command line

University of Zurich[UZH]

Answers

Extract the sequence of scaffold00002_0_arrow

```
seqret multifasta_renamed.fasta:scaffold00002_0_arrow
```

Extract the sequence of the scaffold between 200 and 300, and write them in a new fasta file. Use only 1 command line

1) create a list: `grep "d002" multifasta_renamed.fasta|sed 's/>/multifasta_renamed.fasta:/g' >my_list|`

2) extract from list directly: `seqret @my_list`

University of Zurich[UZH]

Extractseq

Find out how to use extractseq…

And extract the first 1000 nt from the scaffold of your choice

Revseq

Find out how to use revseq…

And do a dot plot of a sequence against its reverse complement