Introduction to Transcriptomics Analysis

Class 01 - First Steps using a CLI in Linux

INSTRUCTOR:
Aureliano Bombarely
Department of Bioscience
Universita degli Studi di Milano
aureliano.bombarely@unimi.it

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

1. Navigating in the terminal
2. Working with files and directories

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

1. Navigating in the terminal
2. Working with files and directories

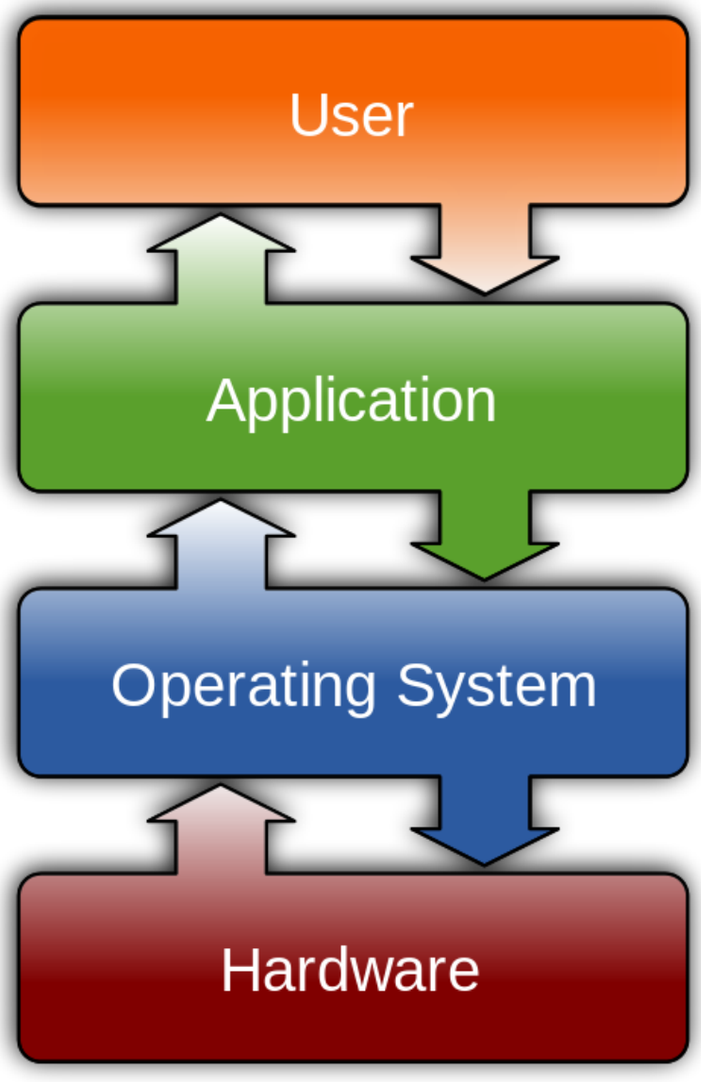
A brief history of Unix

- UNICS—UNiplexed Information and Computing Service
- Developed in the 1970s
- Multiuser, multitasking computer OS

"Ken Thompson (sitting) and Dennis Ritchie at PDP-11 (2876612463)" by Peter Hamer - Licensed under CC BY-SA 2.0 via Commons - https://commons.wikimedia.org/wiki/

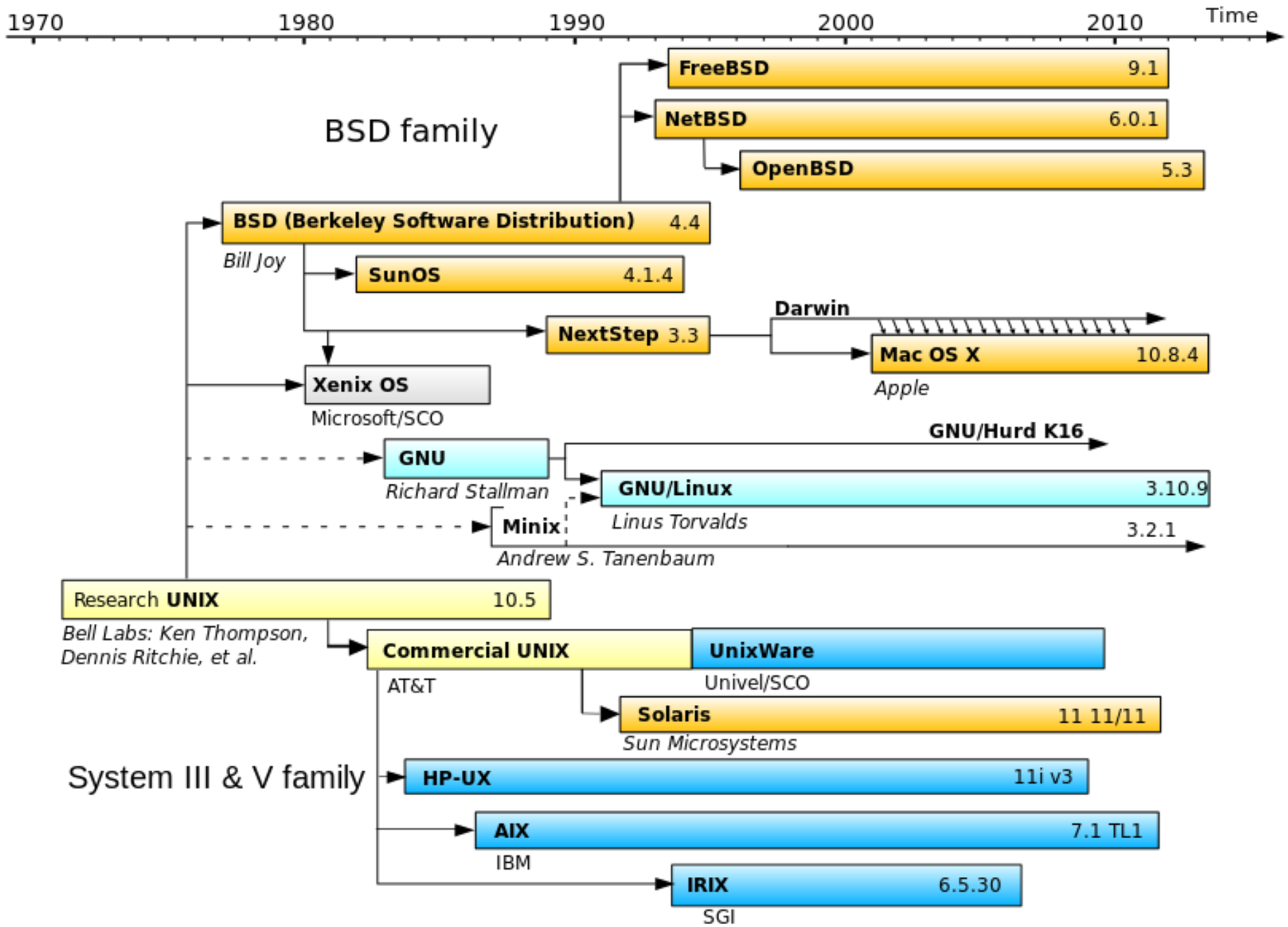
A brief history of Unix

- Universal interaction with mainframe
- Simultaneous passing of instructions
- MULTIUSER!



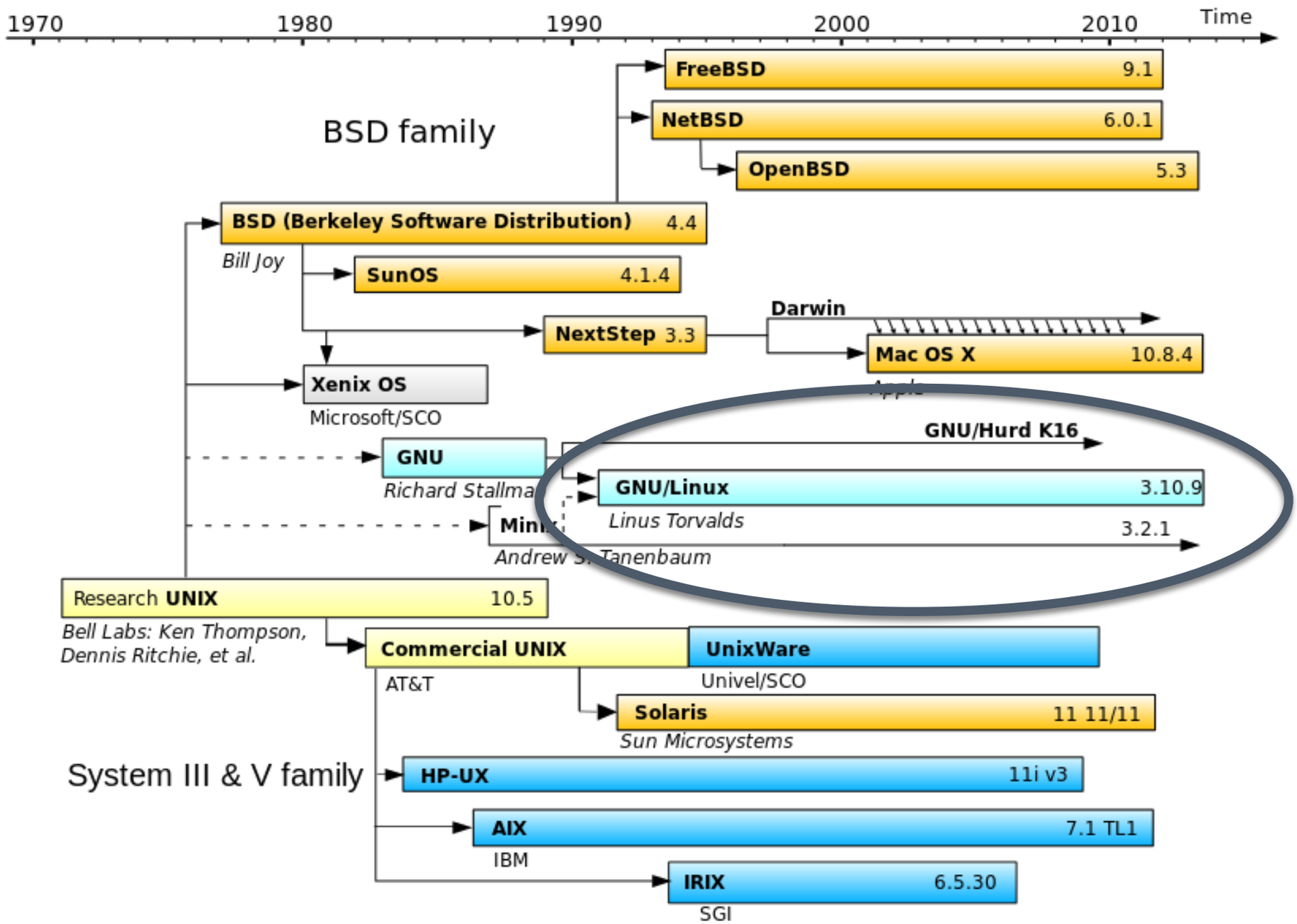
https://upload.wikimedia.org/wikipedia/commons/e/e1/Operating_system_placement.svg

1. What is Linux?



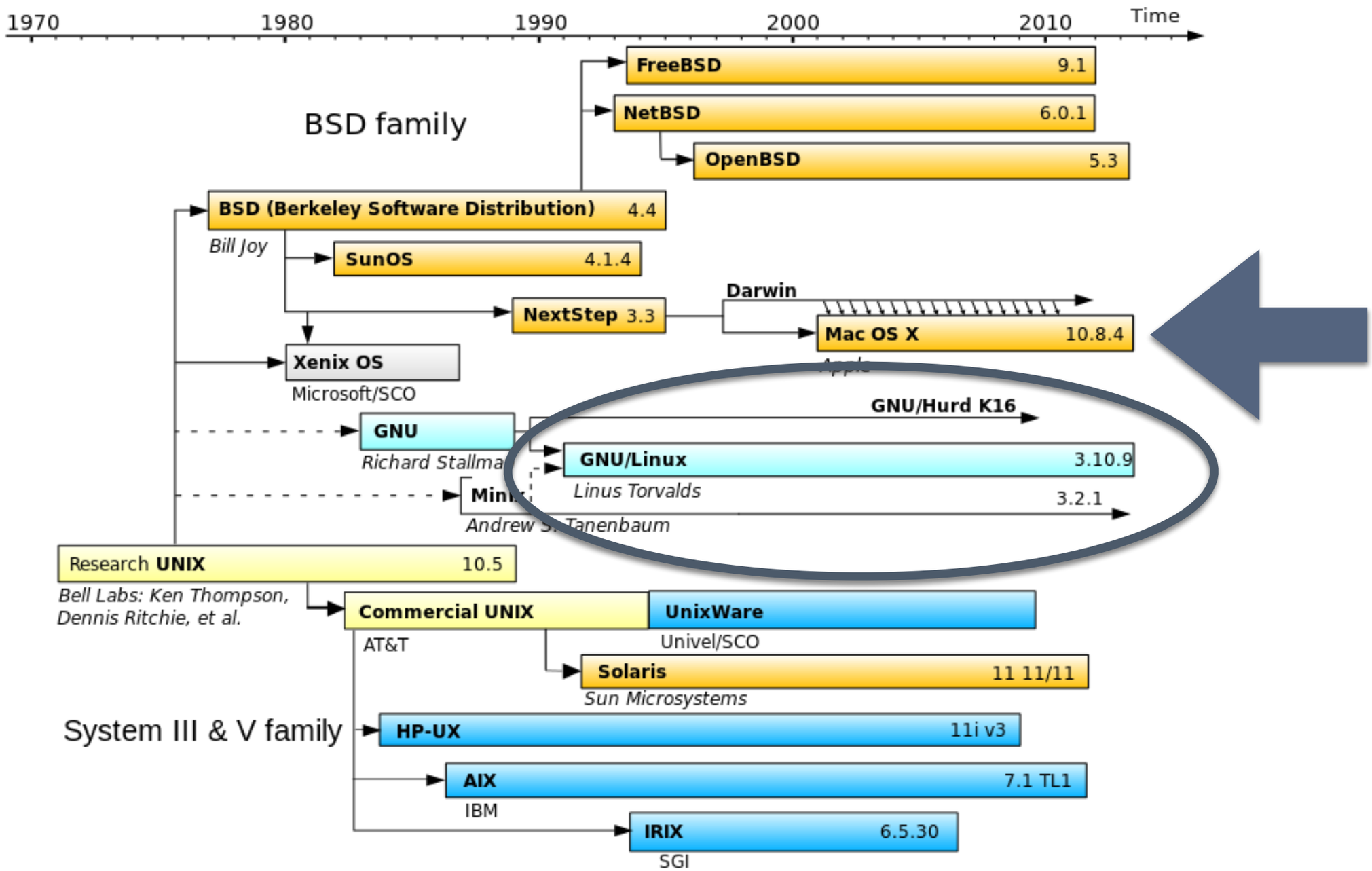
- "Unix timeline.en" by Guillem, Wereon, Hotmocha (copied from old version's history)Christoph S. (redrew the image with Inkscape) - Original image: Image:Unix.png. Licensed under Public Domain via Commons - https://commons.wikimedia.org/wiki/File:Unix_timeline.en.svg#/media/File:Unix_timeline.en.svg

1. What is Linux?



- "Unix timeline.en" by Guillem, Wereon, Hotmocha (copied from old version's history)Christoph S. (redrew the image with Inkscape) - Original image: Image:Unix.png. Licensed under Public Domain via Commons - https://commons.wikimedia.org/wiki/File:Unix_timeline.en.svg#/media/File:Unix_timeline.en.svg

1. What is Linux?



- "Unix timeline.en" by Guillem, Wereon, Hotmocha (copied from old version's history)Christoph S. (redrew the image with Inkscape) - Original image: Image:Unix.png. Licensed under Public Domain via Commons - https://commons.wikimedia.org/wiki/File:Unix_timeline.en.svg#/media/File:Unix_timeline.en.svg

1. What is Linux?

- Unix-like and POSIX compliant
- Free and open-source
- Designed for personal computers

"LinuxCon Europe Linus Torvalds 03" by Krd - Own work. Licensed under CC BY-SA 4.0 via Commons - https://commons.wikimedia.org/wiki/File:LinuxCon_Europe_Linus_Torvalds_03.jpg#/media/File:LinuxCon_Europe_Linus_Torvalds_03.jpg

Linux Distribution:

Distributions (often called distros for short) are **Operating Systems** including a large **collection of software applications** such as word processors, spreadsheets, media players, and database applications.

The operating system will consist of the **Linux kernel** and, usually, a **set of libraries** and **utilities** from the GNU Project, with graphics support from the X Window System.

https://en.wikipedia.org/wiki/Linux_distribution

1. What is Linux?

Linux Distribution:

Different libraries and utilities.

Web Browser
installed by default

https://en.wikipedia.org/wiki/Linux_distribution

1. What is Linux?

Linux Distribution: ... more than 600 distributions

https://en.wikipedia.org/wiki/Linux_distribution

1. What is Linux?

Linux Distribution:

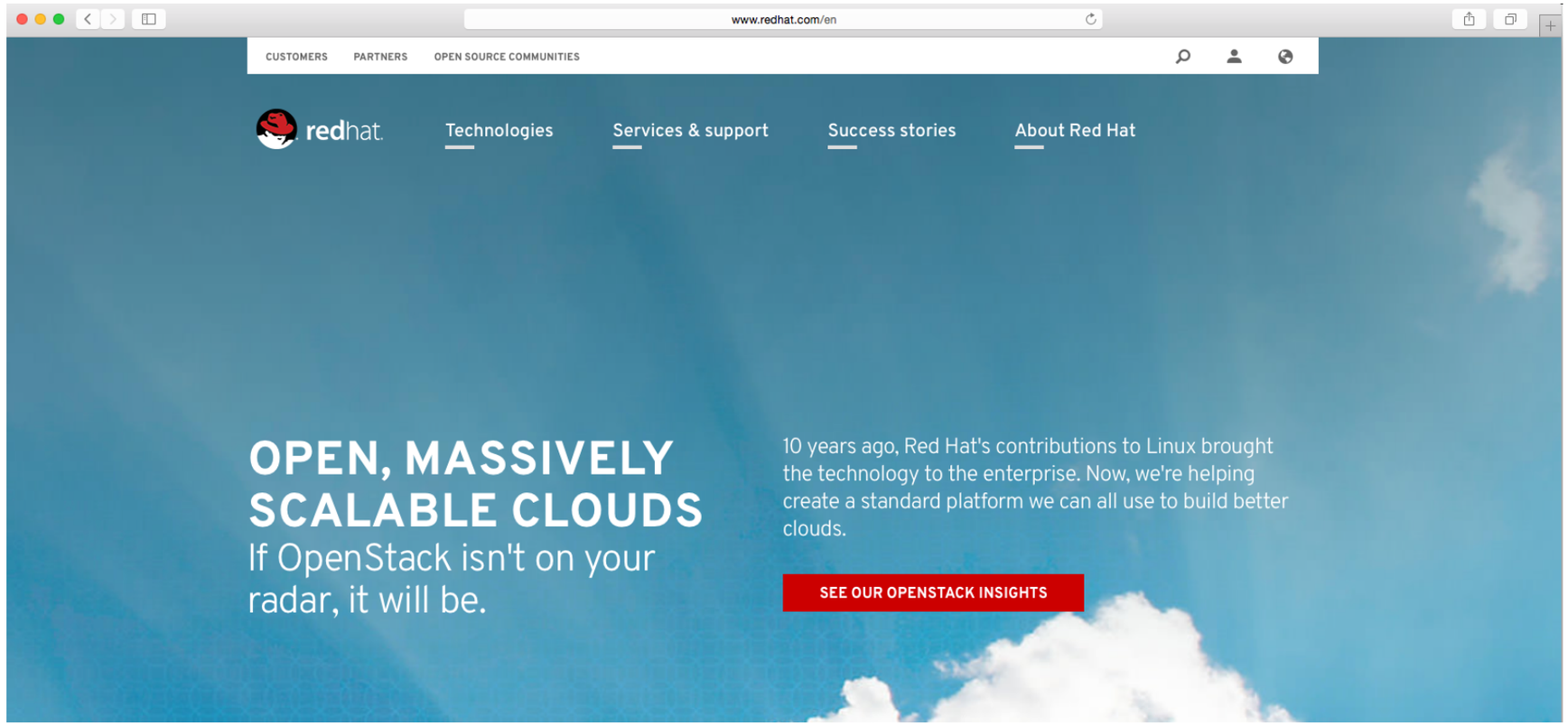
One can distinguish between:

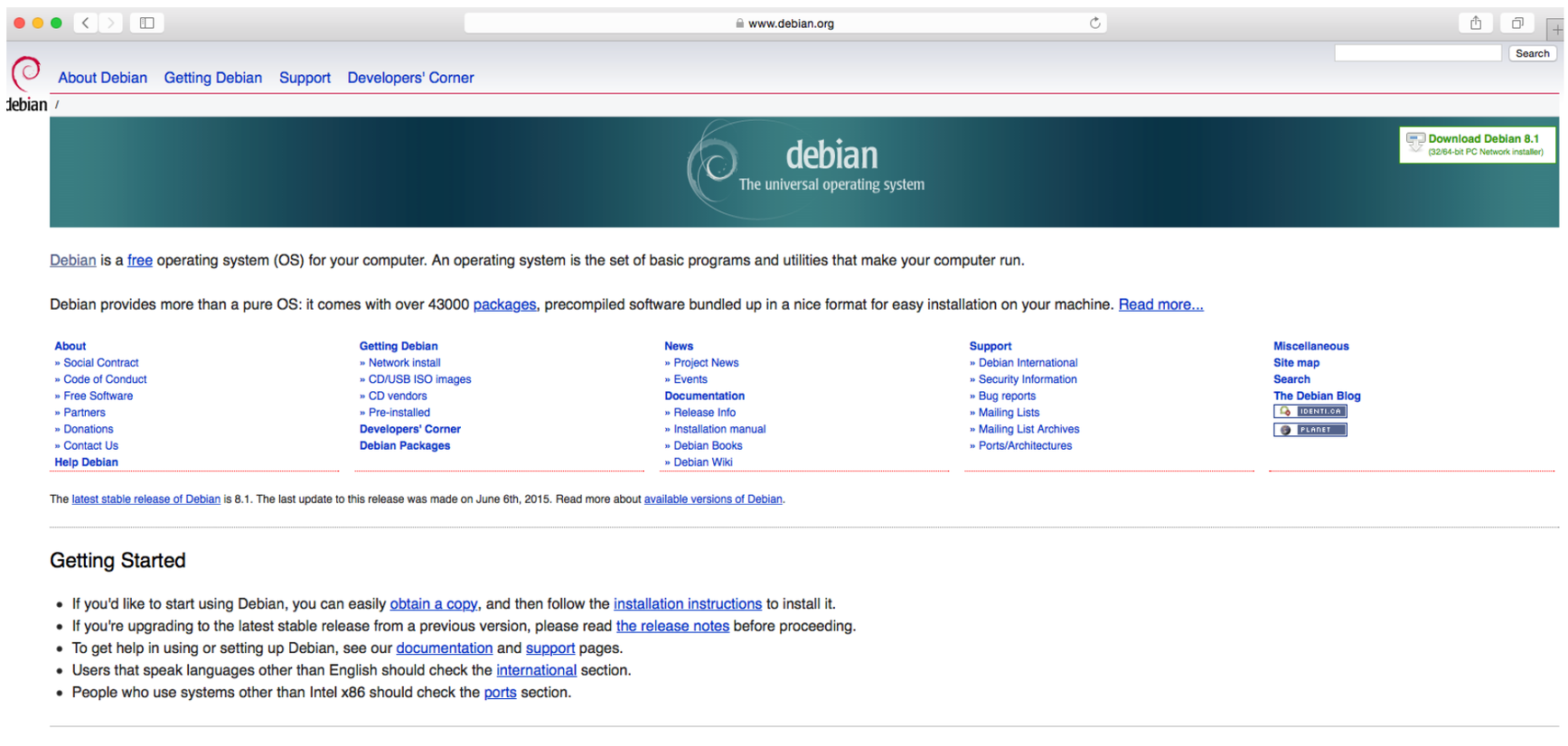
1) Commercially-backed distributions, such as:
   Fedora (Red Hat),
   OpenSUSE (Novell),
   Ubuntu (Canonical Ltd.)
   Mandriva Linux (Mandriva)

2) Entirely community-driven distributions, such as:
   Debian.
   Gentoo.

https://en.wikipedia.org/wiki/Linux_distribution

1. What is Linux?





Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

1. Navigating in the terminal
2. Working with files and directories

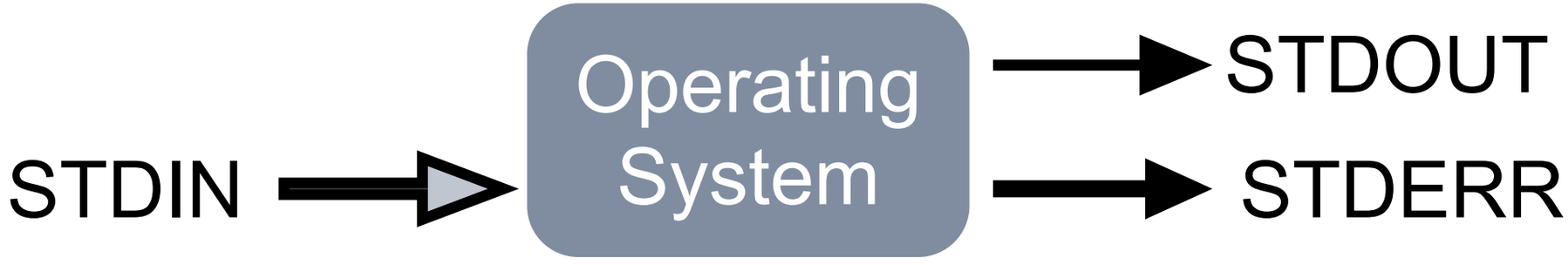
2. The Terminal

What is a console ?

Computer terminal or system consoles are the **text entry and display device** for system administration messages, particularly those from the BIOS or boot loader, the kernel, from the init system and from the system logger. It is a **physical device consisting of a keyboard and a screen**.

2. The Terminal

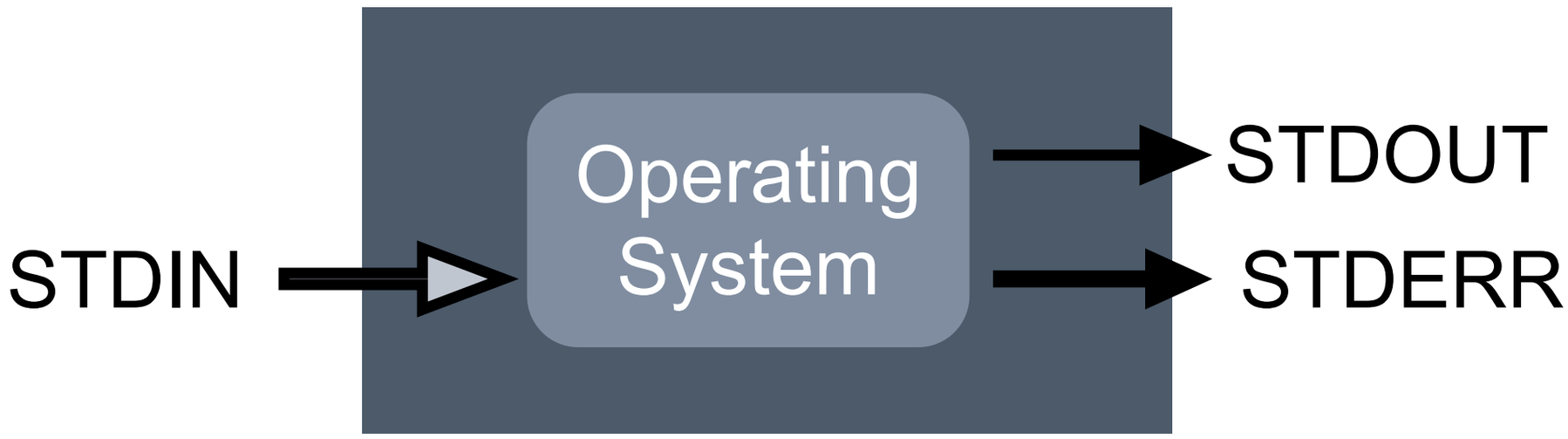
How do we interface with the operating system?



Shell

2. The Terminal

How do we interface with the operating system?



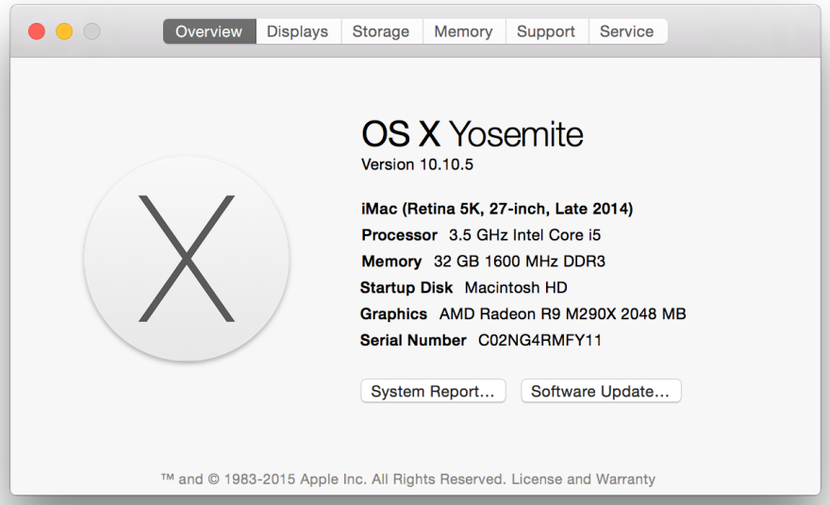
Shell

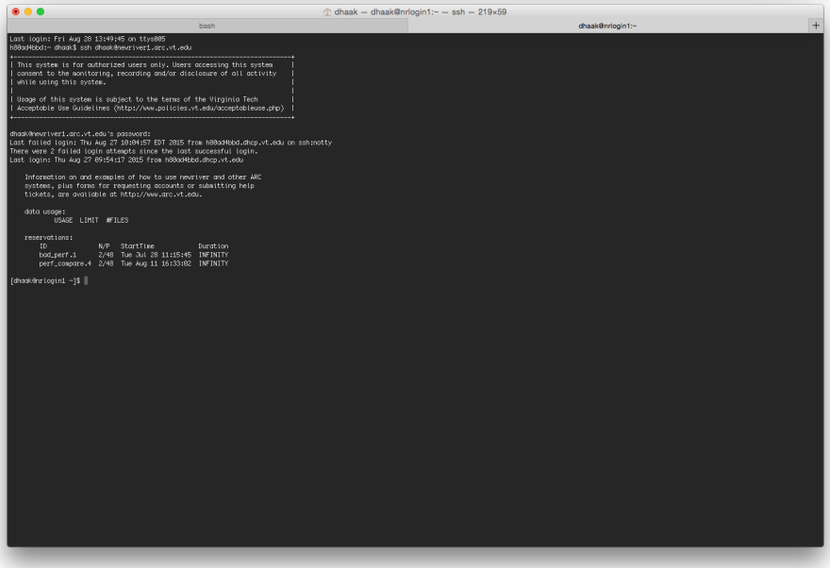
2. The Terminal

Shell: Software that provides an interface with the OS. There are two categories:

1. **Graphical User Interface (GUI):**

2. **Command-line interface (CLI):**



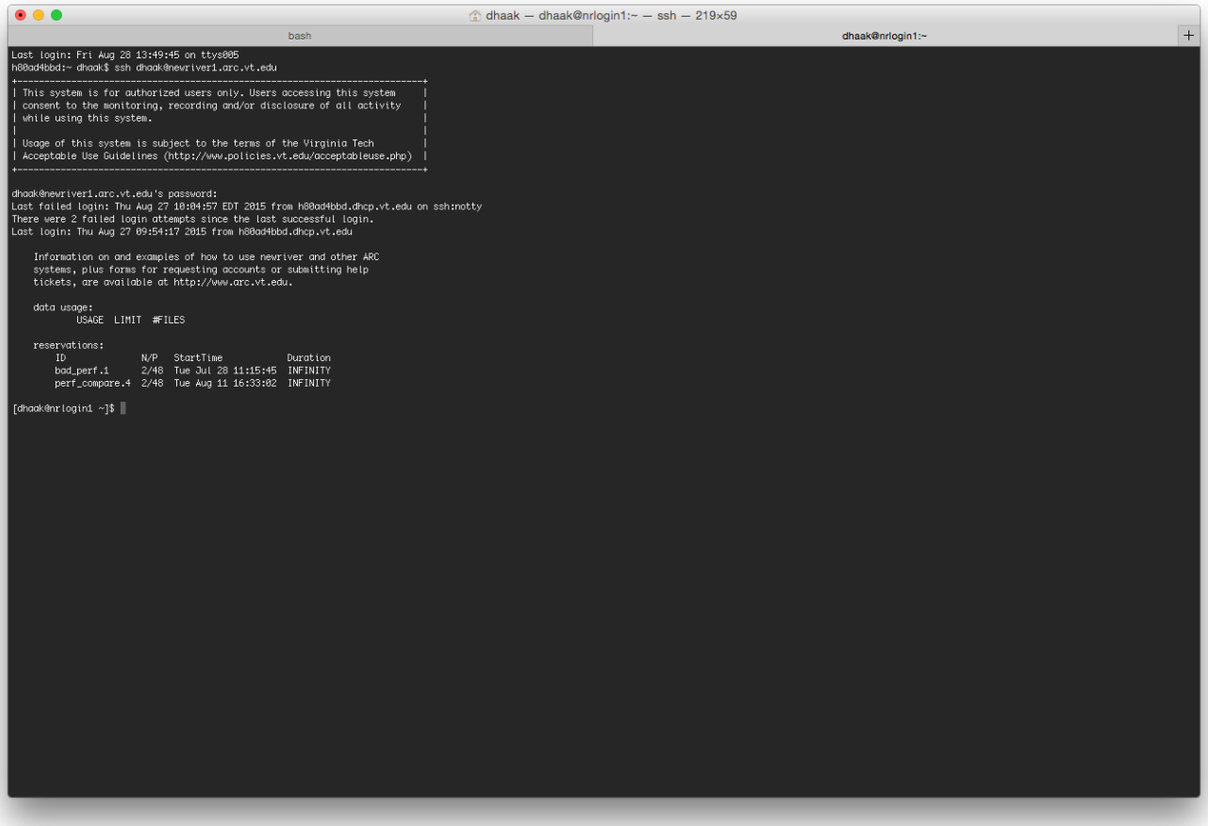


2. The Terminal

Command-line interface (CLI):

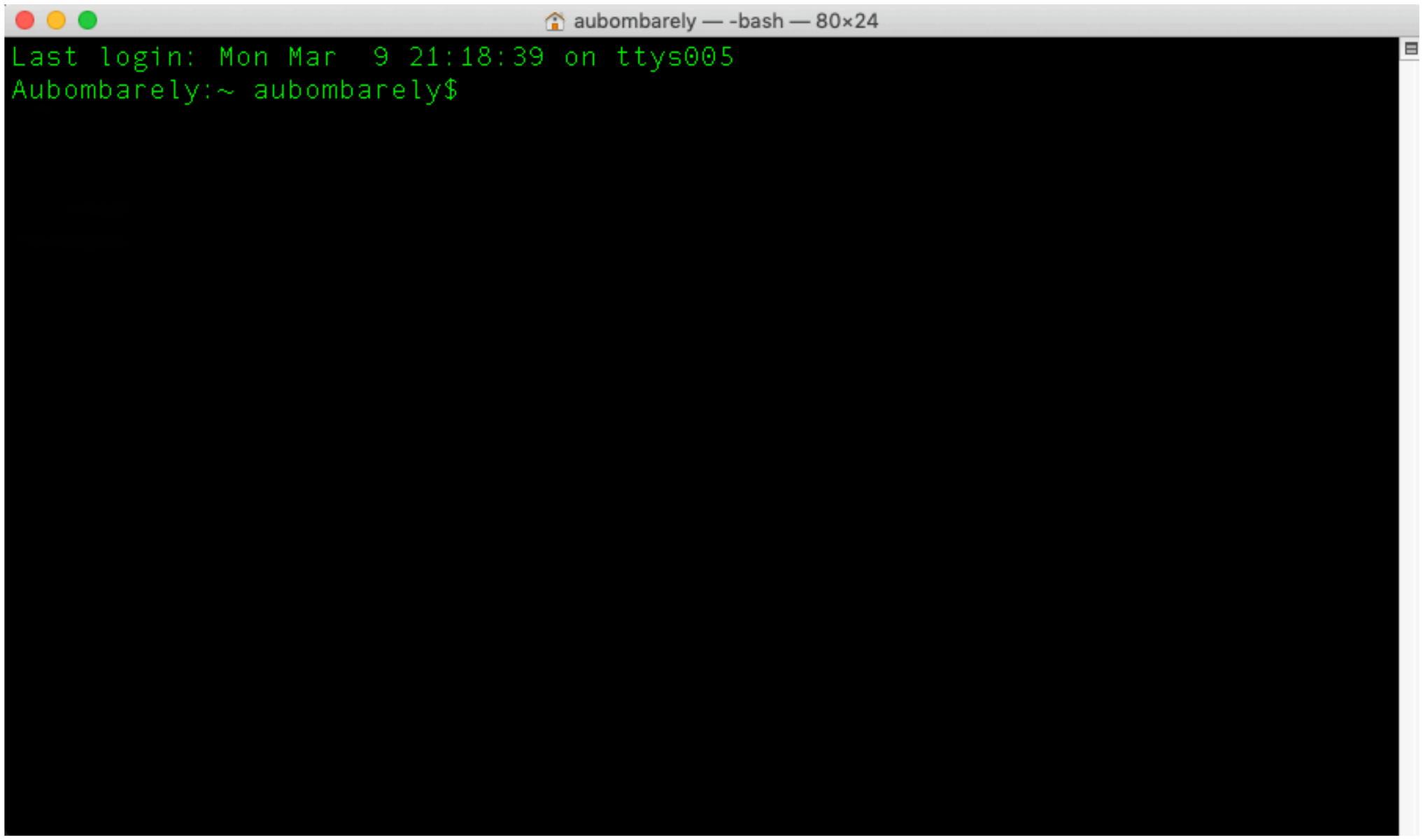
Mechanism for **interacting** with a computer operating system or software by typing **commands** to perform specific tasks.

The command-line interpreter may be run in a **text terminal** or in a **terminal emulator** window as a remote shell client such as PuTTY.



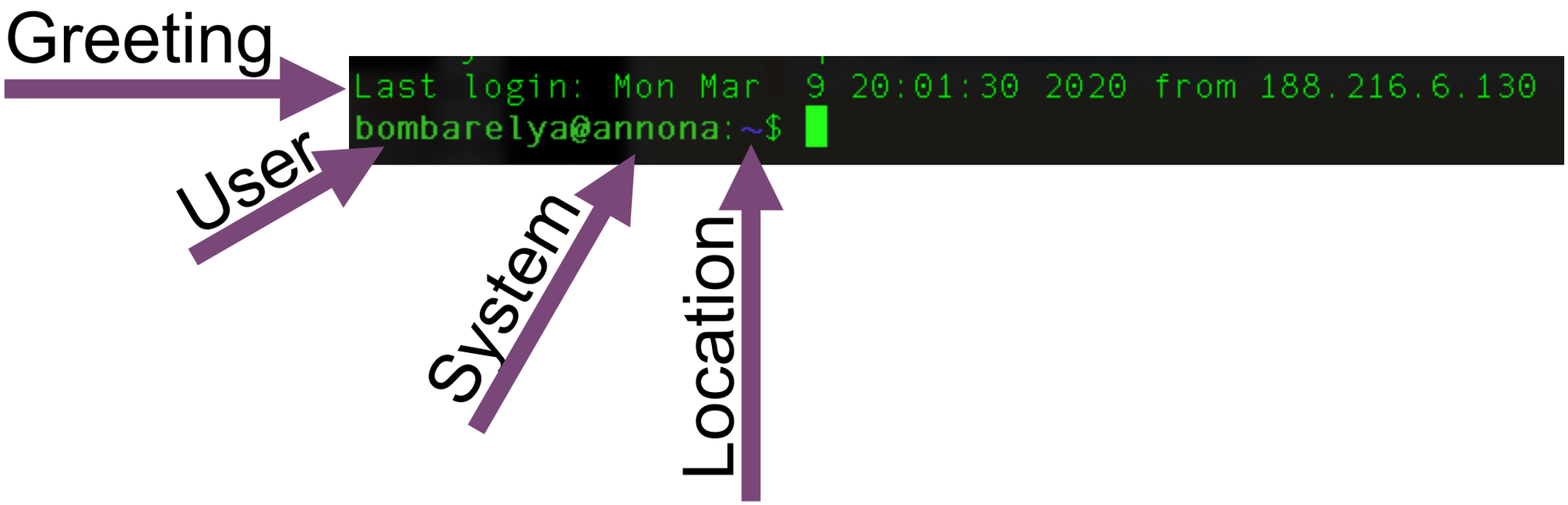
2. The Terminal

Bash-Bourne again shell



2. The Terminal

Bash-Bourne again shell



2. The Terminal

Bash-Bourne again shell

Check your shell type:

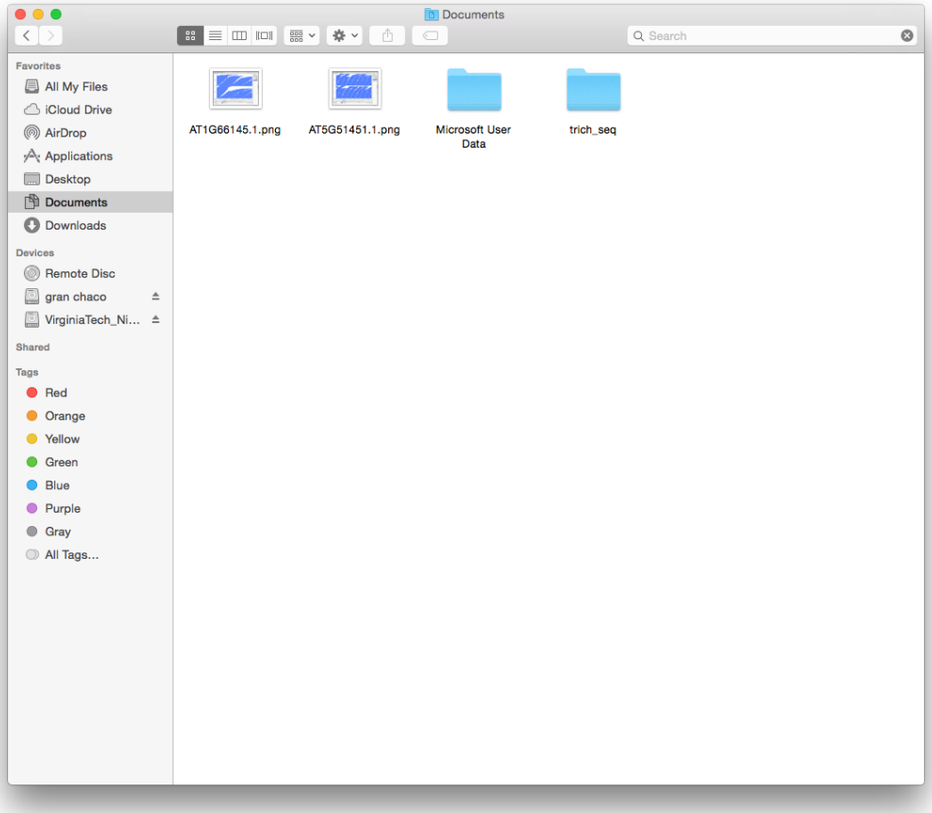
```
userid$ echo $SHELL
```

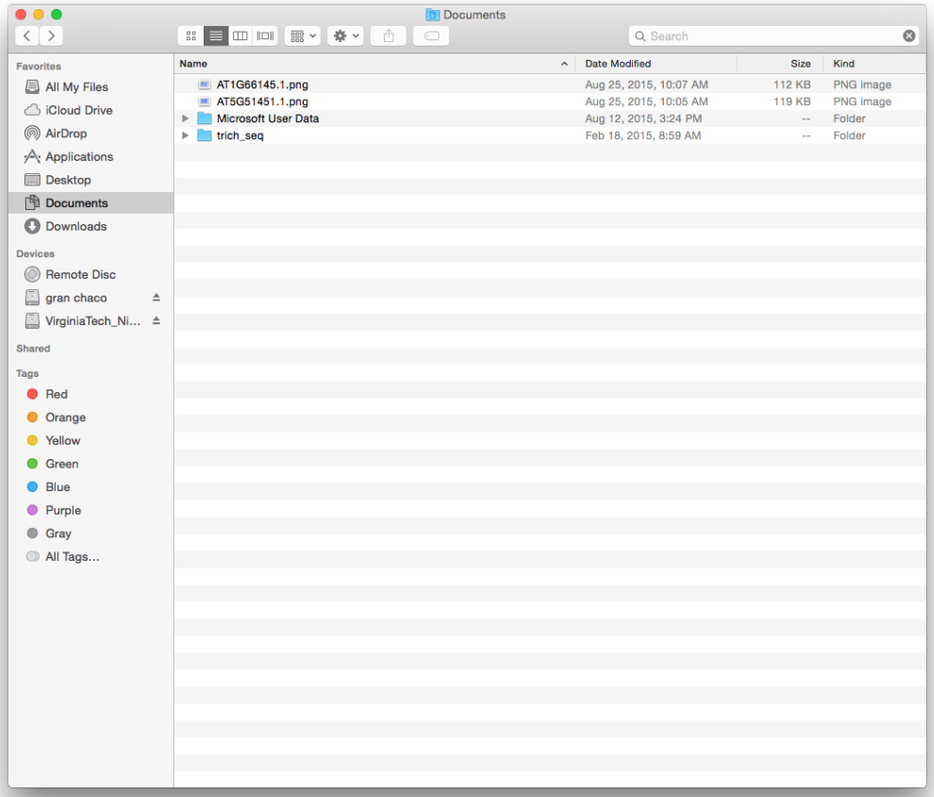
Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

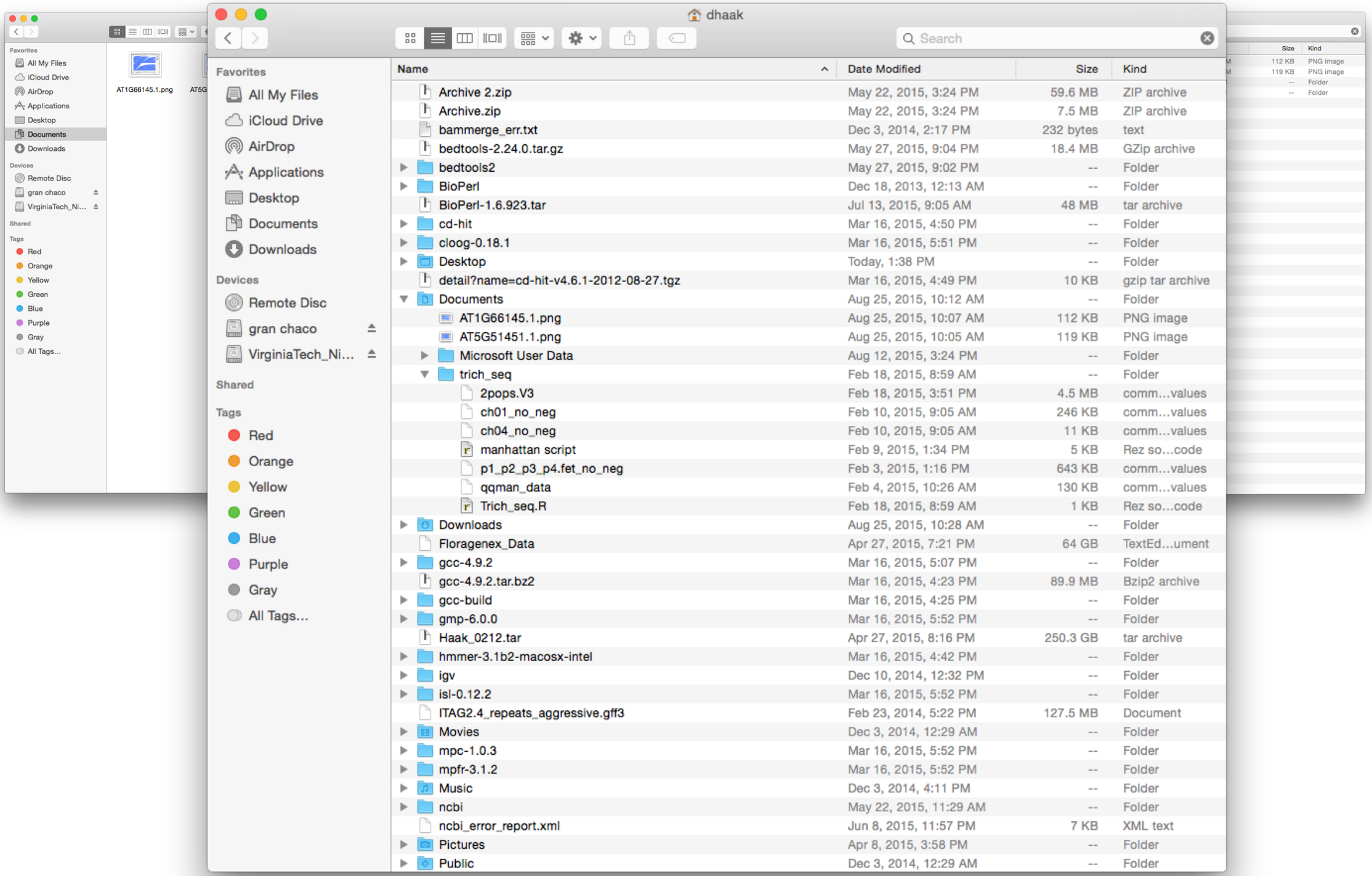
1. Navigating in the terminal
2. Working with files and directories

3. Files and Directories

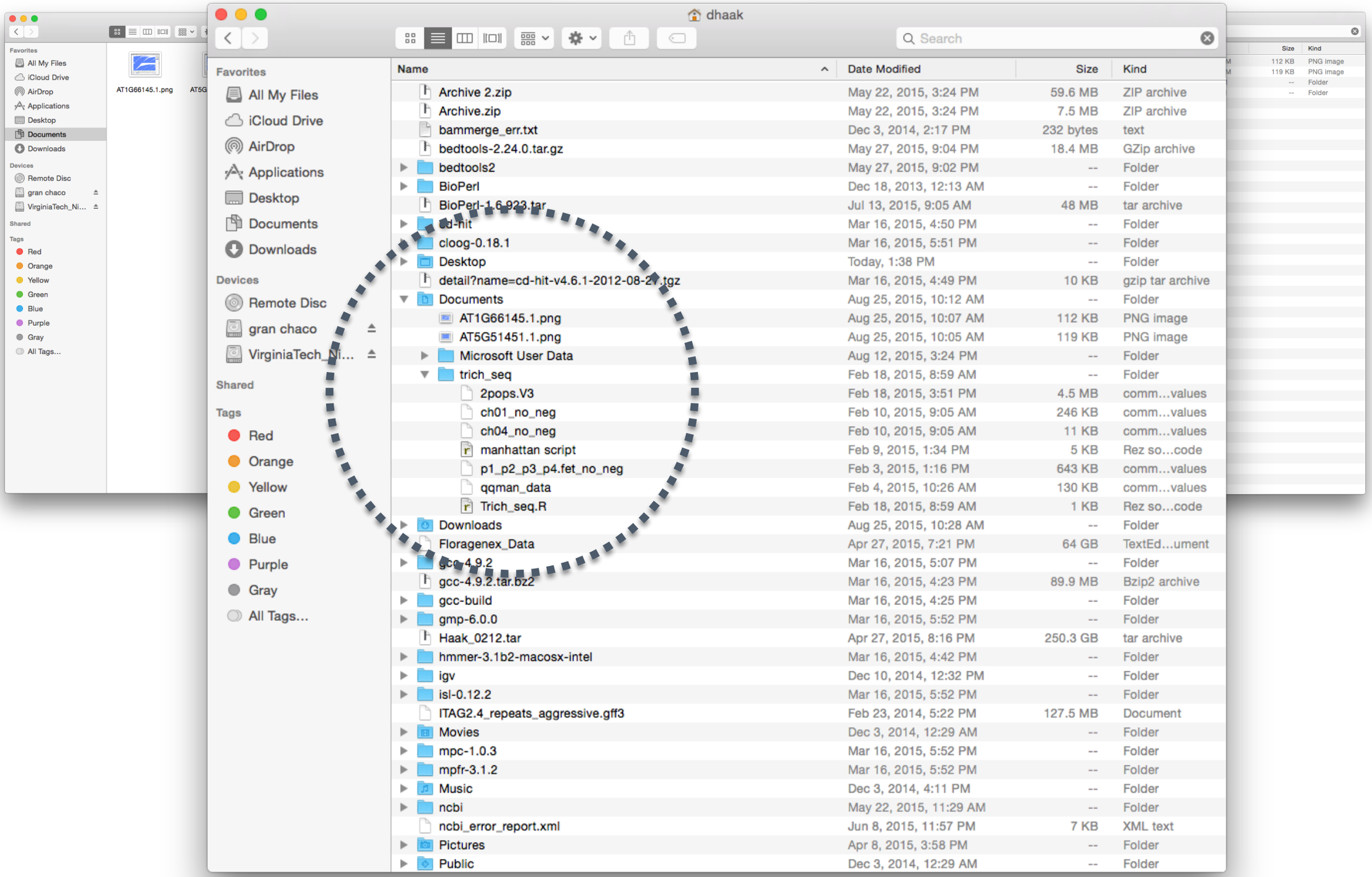




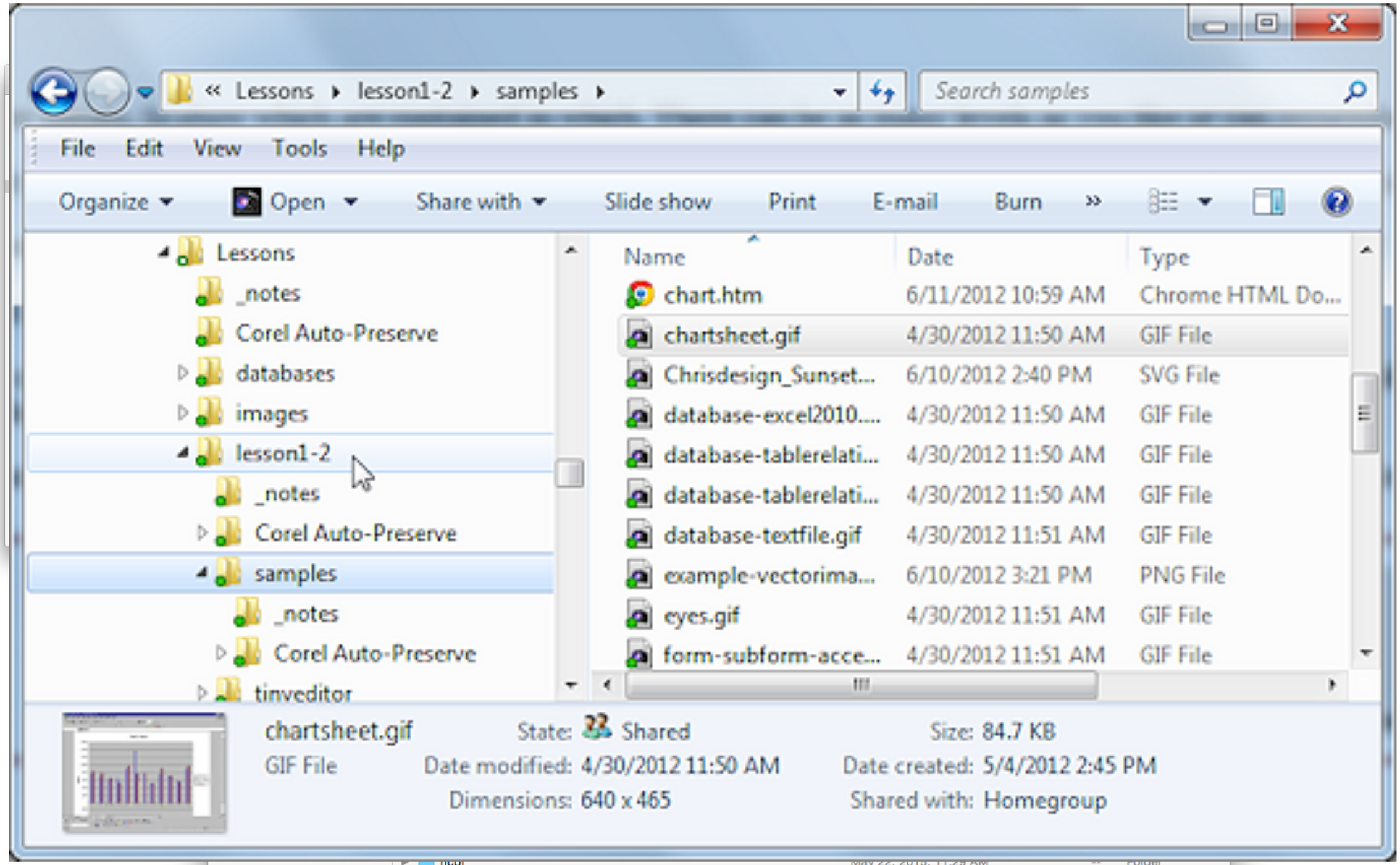
3. Files and Directories



3. Files and Directories



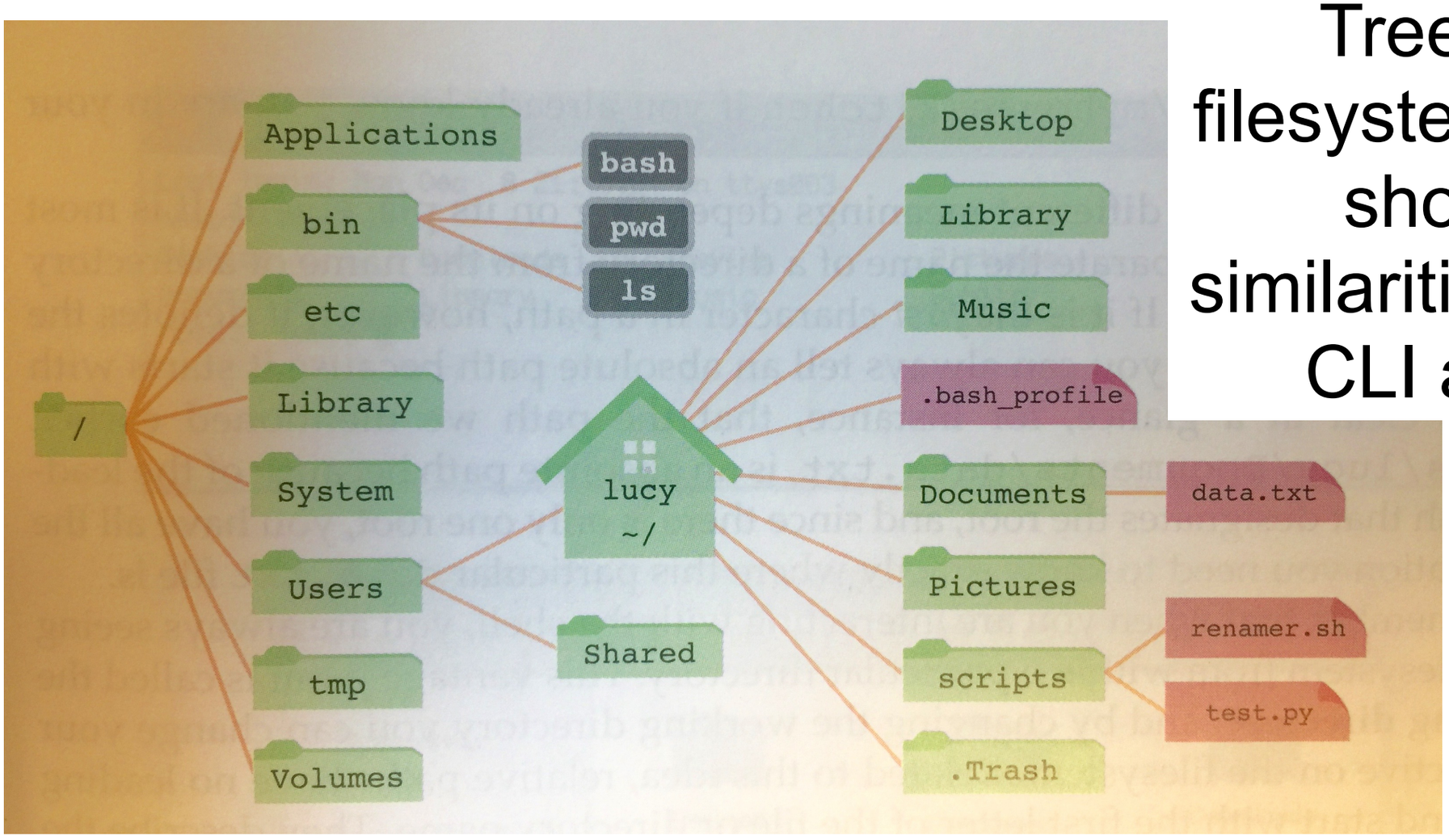
3. Files and Directories



3. Files and Directories

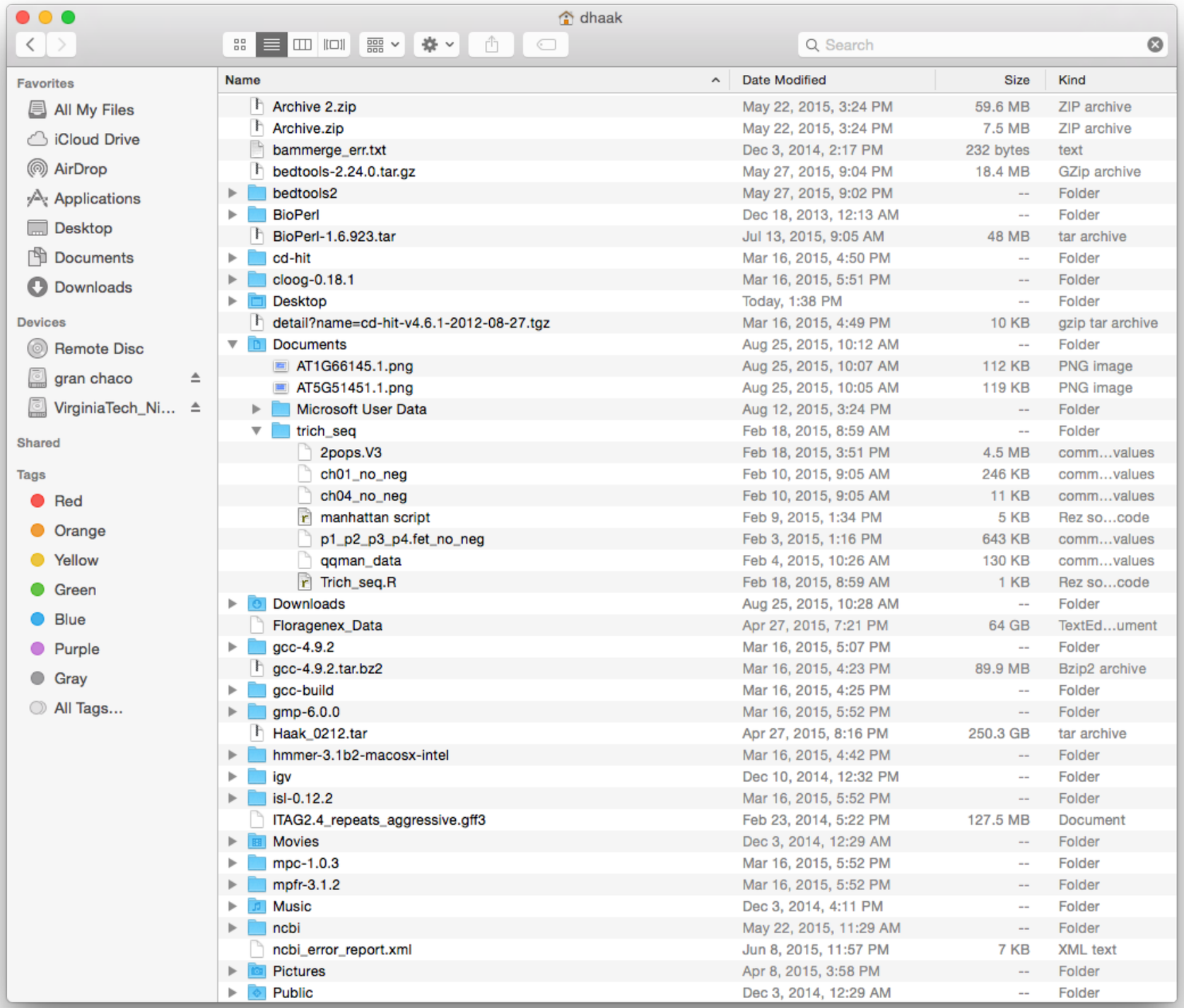
- Is file access convenient?
- What about repeated access?
- What about speed?
- Cross platform use?

3. Files and Directories

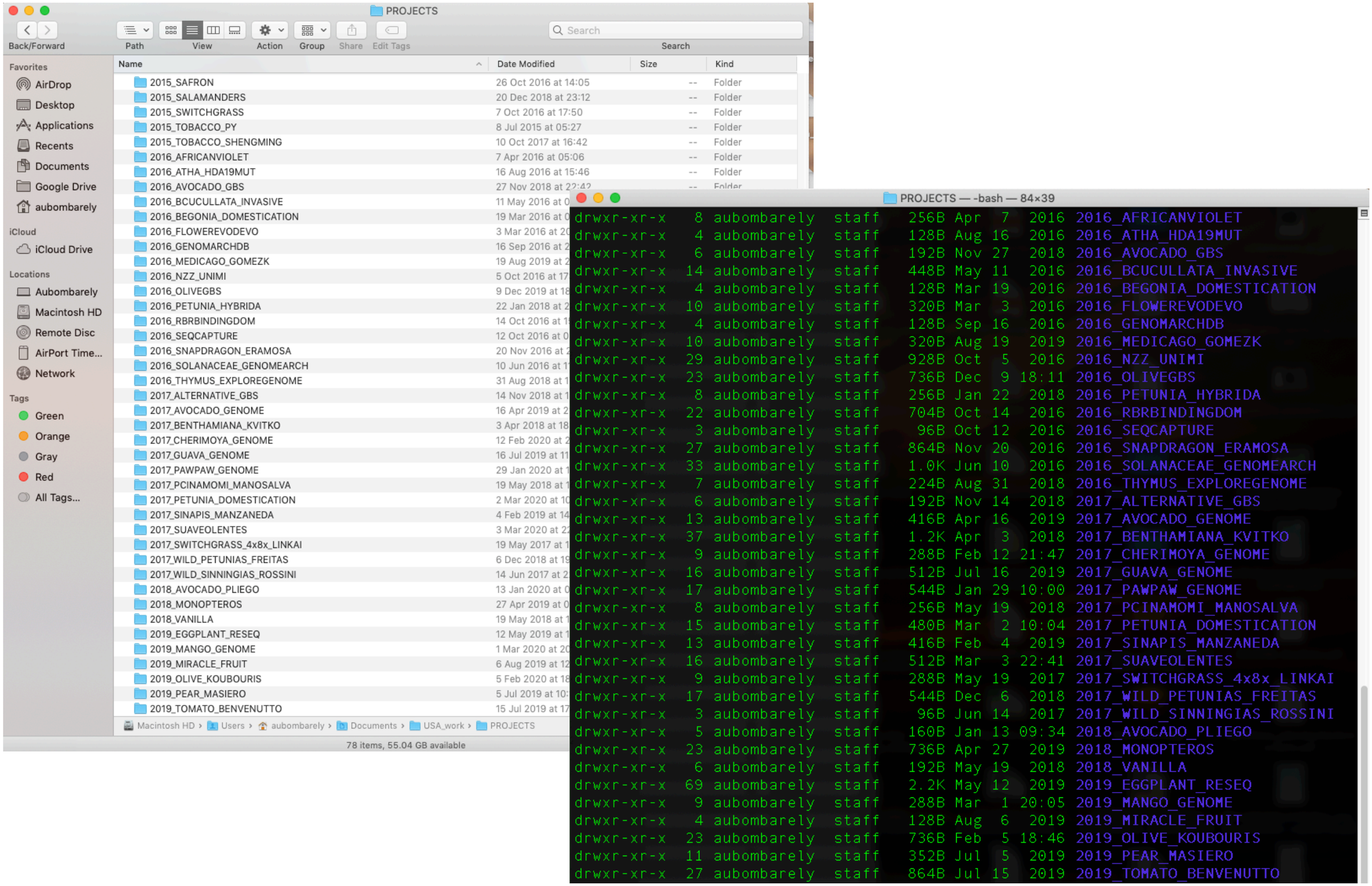


Tree format filesystem structure shows the similarities between CLI and GUI

3. Files and Directories



3. Files and Directories



3. Files and Directories

Home versus root

```
bombarelya@annona:~$ ls -lh
total 24K
drwxrwxr-x 3 bombarelya bombarelya 4.0K Jan  9 17:09 Avocado
drwxrwxr-x 3 bombarelya bombarelya 4.0K Feb 11 21:37 Cherimola
drwxrwxr-x 3 bombarelya bombarelya 4.0K Nov  6 12:51 Mango
drwxrwxr-x 5 bombarelya bombarelya 4.0K Jan 17 16:12 perl5
drwxrwxr-x 3 bombarelya bombarelya 4.0K Feb 28 15:09 R
drwxrwxr-x 6 bombarelya bombarelya 4.0K Feb 19 10:58 Suaveolentes
```

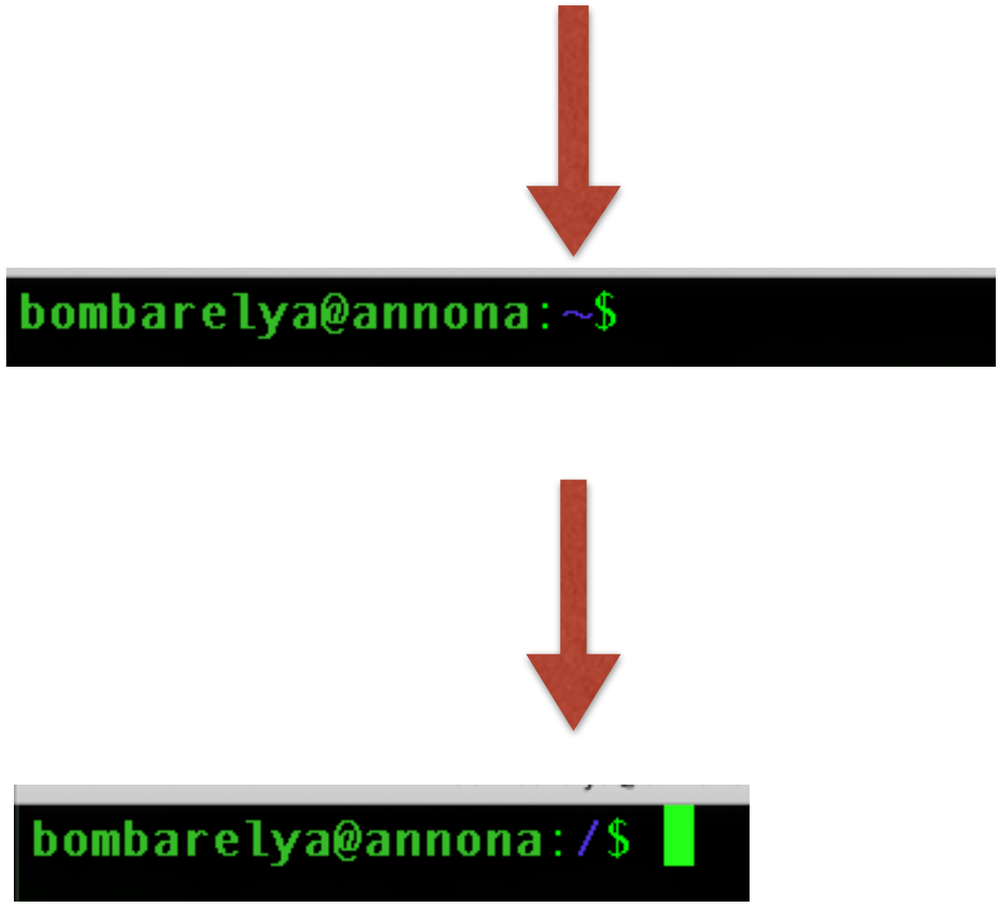
Home directory
~/bombarelya
/data/bombarelya

```
bombarelya@annona:/$ ls -lh
total 8.1G
drwxr-xr-x   2 root root 4.0K Sep 20 14:55 altos
drwxr-xr-x   2 root root 4.0K Feb 11 09:23 bin
drwxr-xr-x   4 root root 4.0K Feb 20 06:40 boot
drwxr-xr-x   2 root root 4.0K Sep 20 14:23 cdrom
drwxr-xr-x  19 root root 4.0K Mar  9 09:30 data
drwxr-xr-x  20 root root 4.1K Feb 16 09:16 dev
drwxr-xr-x 116 root root 4.0K Mar  9 20:23 etc
drwxr-xr-x   3 root root 4.0K Sep 20 14:32 home
lrwxrwxrwx   1 root root   33 Feb 19 06:49 initrd.img -> boot/initrd.img-4.15.0-88-generic
lrwxrwxrwx   1 root root   33 Feb 19 06:49 initrd.img.old -> boot/initrd.img-4.15.0-76-generic
drwxr-xr-x  22 root root 4.0K Nov  6 12:57 lib
drwxr-xr-x   2 root root 4.0K Aug  5  2019 lib64
drwx------   2 root root  16K Sep 20 14:23 lost+found
drwxr-xr-x   2 root root 4.0K Aug  5  2019 media
drwxr-xr-x   2 root root 4.0K Aug  5  2019 mnt
drwxr-xr-x   2 root root 4.0K Aug  5  2019 opt
dr-xr-xr-x 702 root root    0 Feb 16 09:14 proc
drwx------   4 root root 4.0K Feb 11 08:18 root
drwxr-xr-x  26 root root  980 Mar  9 20:27 run
drwxr-xr-x   2 root root  12K Feb 11 09:23 sbin
drwxr-xr-x   4 root root 4.0K Sep 20 14:32 snap
drwxr-xr-x   2 root root 4.0K Aug  5  2019 srv
-rw-------   1 root root 8.0G Sep 20 14:24 swap.img
dr-xr-xr-x  13 root root    0 Feb 16 09:14 sys
drwxrwxrwt  16 root root 4.0K Mar  9 20:22 tmp
drwxr-xr-x  10 root root 4.0K Aug  5  2019 usr
drwxr-xr-x  13 root root 4.0K Aug  5  2019 var
lrwxrwxrwx   1 root root   30 Feb 19 06:49 vmlinuz -> boot/vmlinuz-4.15.0-88-generic
lrwxrwxrwx   1 root root   30 Feb 19 06:49 vmlinuz.old -> boot/vmlinuz-4.15.0-76-generic
```

Root directory
/bin, /sbin, /dev
/data, /home

3. Files and Directories

Home versus root



Home directory
~/bombarelya
/data/bombarelya

Root directory
/bin, /sbin, /dev
/data, /home

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

1. Navigating in the terminal
2. Working with files and directories

4. Absolute and Relative Paths

ABSOLUTE PATH: Location of a file **from root (/)**

Latham Hall, 311
220 Ag Quad Ln.
Blacksburg, VA 24060
USA

RELATIVE PATH: Location of a file **from working dir**

Room 311

4. Absolute and Relative Paths

Relative filepath depends of the working directory

```
bombarelya@annona:~/Avocado$ ls RNASeq_PliegoC/
```

Absolute filepath starts at root

```
bombarelya@annona:~/Avocado$ ls /data/bombarelya/Avocado/RNASeq_PliegoC/
```

4. Absolute and Relative Paths

Find the present working directory

```
pwd
```

Change the working directory

```
cd <directory>
```

Goes to the root

```
cd /
```

/home/user

```
cd ~
```

up one dir

```
cd ..
```

What does this do?

```
cd
```

4. Absolute and Relative Paths

Create a directory

```
mkdir dirname
```

Create an empty file

```
touch filename
```

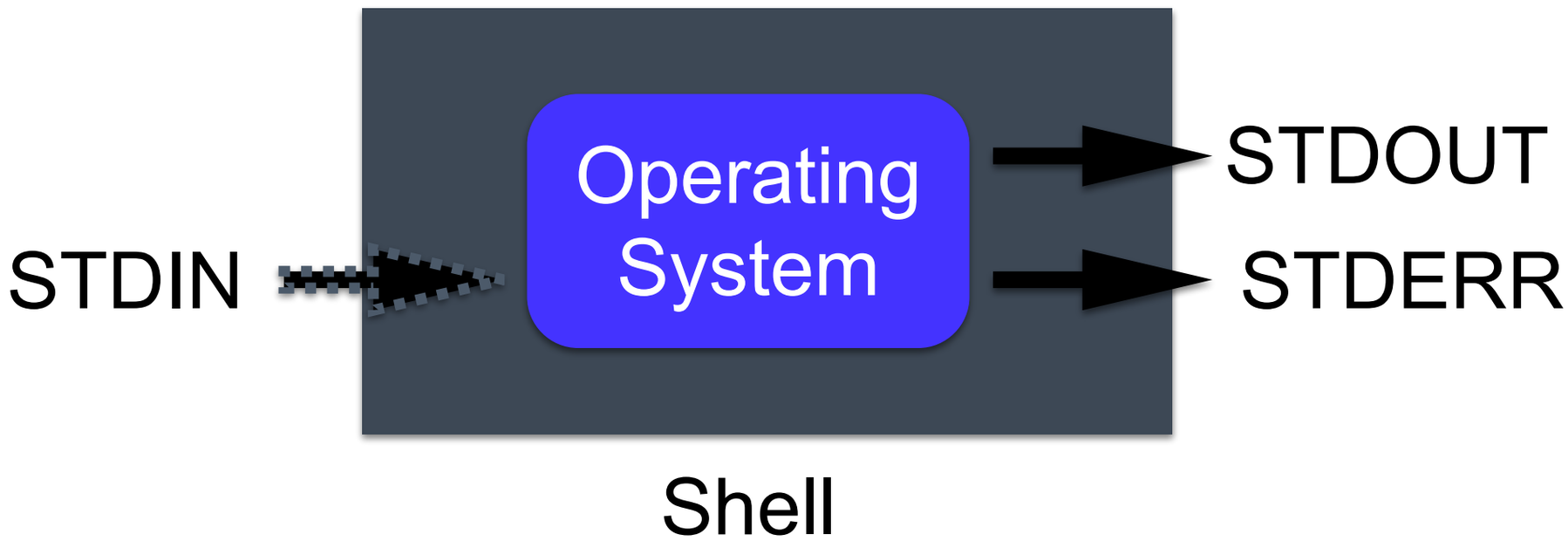
Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Shortcuts
8. Environmental variables
9. Monitoring resources
10. Networking
11. Installing programs

1. Navigating in the terminal
2. Working with files and directories

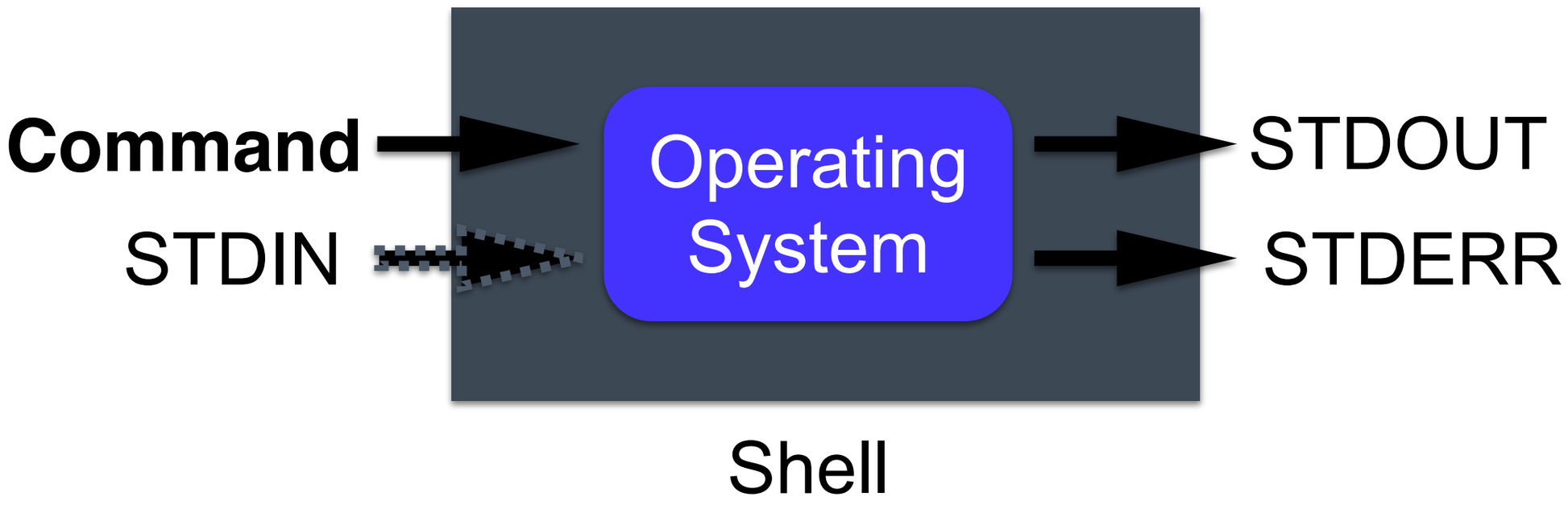
5. Basic Command Structure

How do we interface with the operating system?



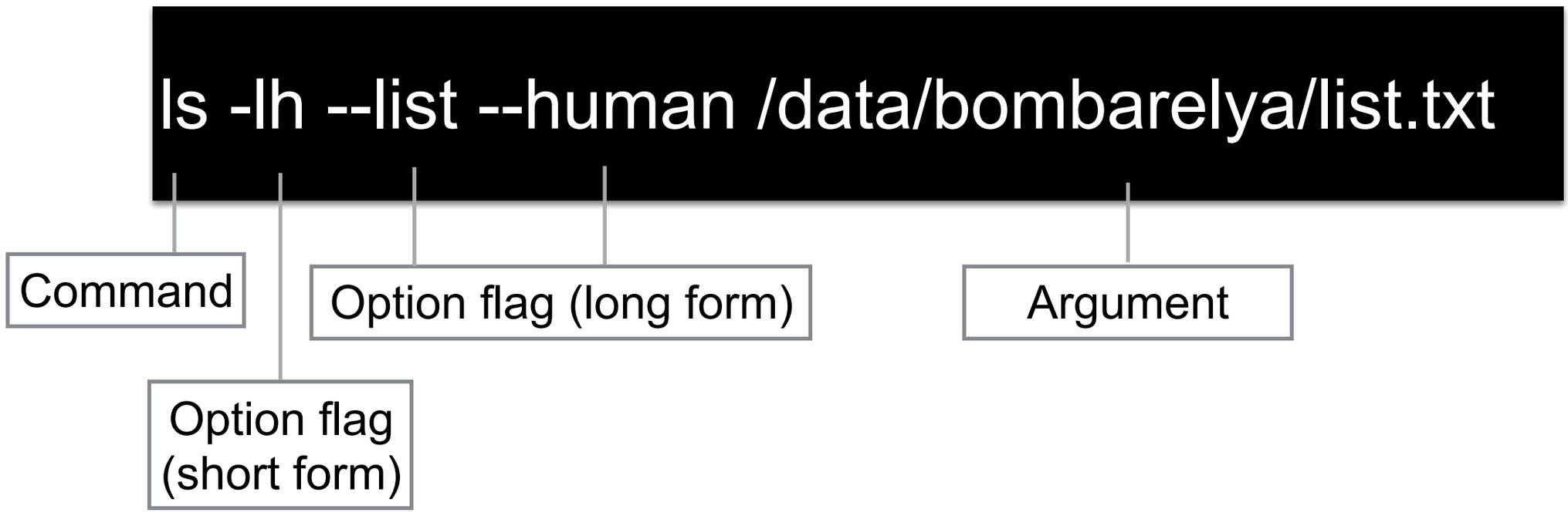
5. Basic Command Structure

How do we interface with the operating system?



5. Basic Command Structure

Anatomy of a UNIX command



Exercise 1

The behavior of a command can be modified using **options** such as -l or -a.

1. Create a directory with the name “exercise01”
2. Change the working directory to “exercise01”
3. Create an empty file with the name “test01.txt”.
4. Create another empty file with the name “”test02.txt

Exercise 1

5. Run **ls**, without options (1), with **-lh** (2), with **-l -h** (3) and with **-lha** (4).

What are the differences in the output?

5. Basic Command Structure

Special characters in bash:

Character	Meaning
SPACE	Separate commands and arguments
# HASH	Comment
; SEMICOLON	Command separator to run multiple commands
. DOT	Source command OR filename component OR current directory
.. DOUBLE DOTS	Parent directory
‘ SINGLE QUOTES	Use expression between quotes
, COMMA	Concatenate strings
\ BACKSLASH	Escape for single character
/ SLASH	Filename path separator
* ASTERISK	Wildcard for filename expansion
>,<,>> CHARACTERS	Redirection input/outputs
| PIPE	Pipe outputs between commands
! BANG	Run a command

Special characters in bash:

```
ls Solanum lycopersicum
```

Bash interprets spaces as separators

```
ls ‘Solanum lycopersicum’
ls Solanum\ lycopersicum
```

Use single quotes or escape ‘\’ for special characters

Special characters in bash:

The wildcard ‘*’ is your friend

ls ~/*txt	Lists all text files in your directory
ls ~/D*p	Lists files/directories in your home folder that start with D and end with p

Note: With great power comes great responsibility

Getting more information for a command:

```
man <command>
```

```
man ls
```

note: use “q” to exit the interface.

5. Basic Command Structure

Tracing your steps:

```
$ history
```

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

Permissions

6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

Permissions

Links

6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

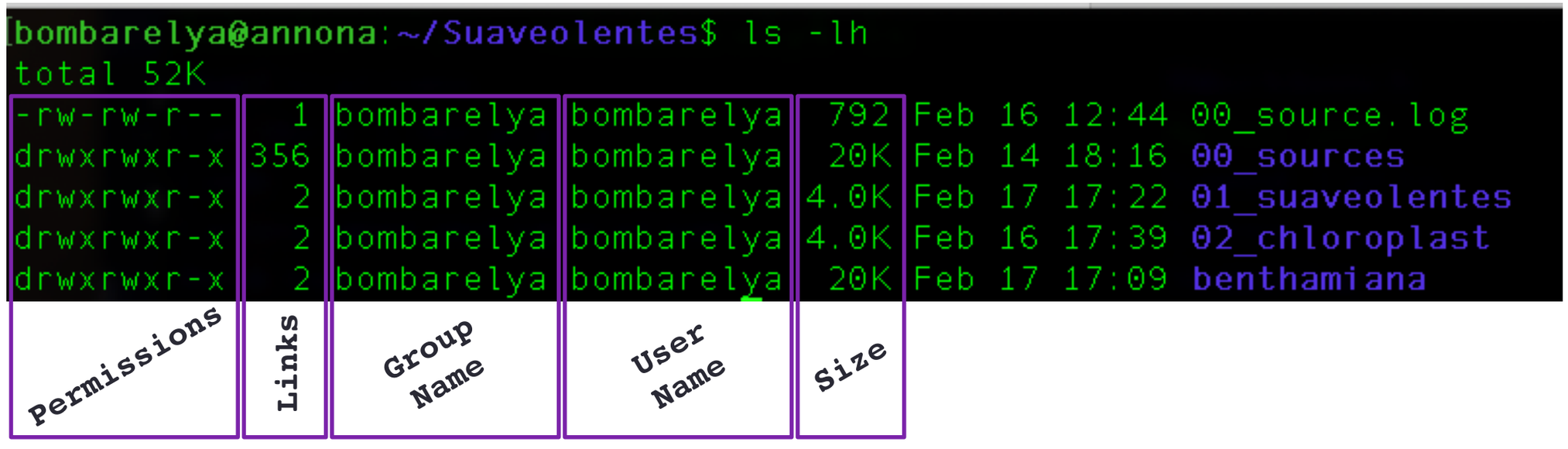
Permissions | Links | Group Name

6. User, groups and permissions

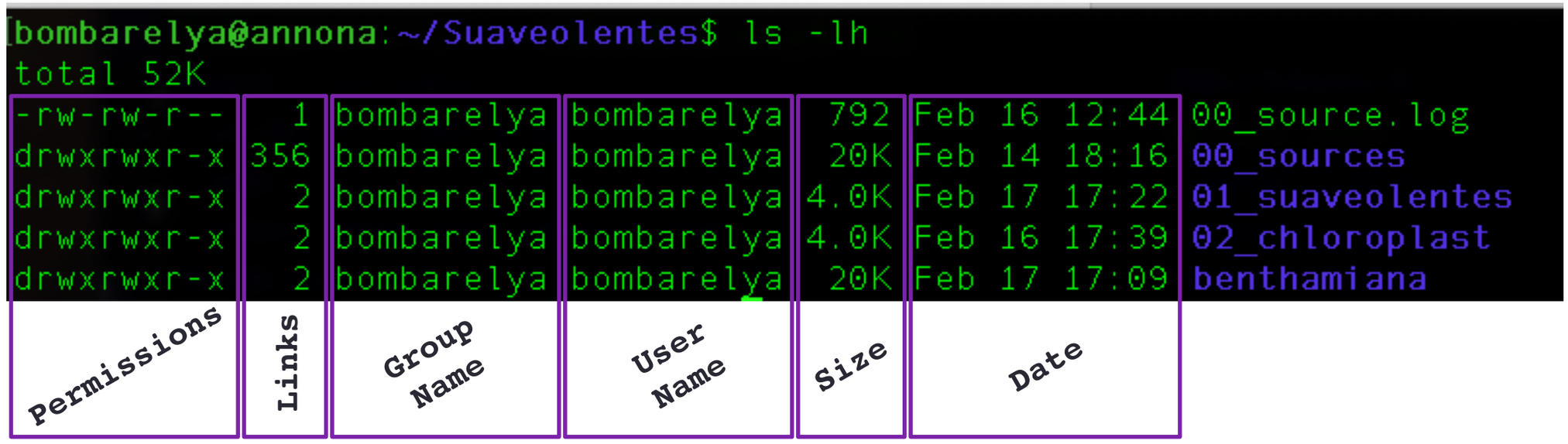
```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

Permissions | Links | Group Name | User Name

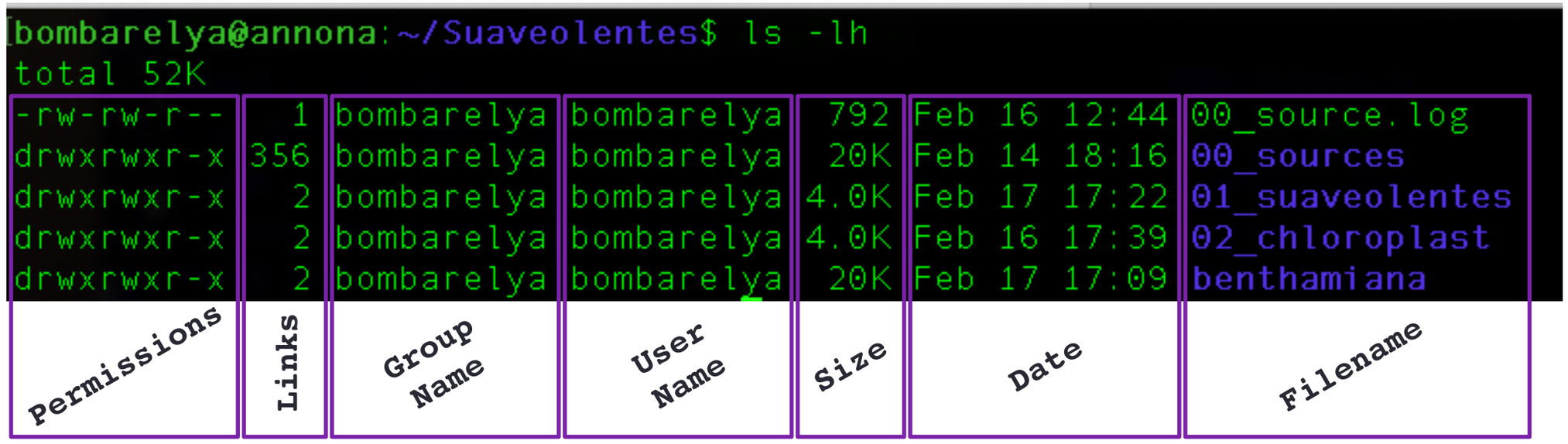
6. User, groups and permissions



6. User, groups and permissions



6. User, groups and permissions



6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

Permissions | Links | Group Name | User Name | Size | Date | Filename

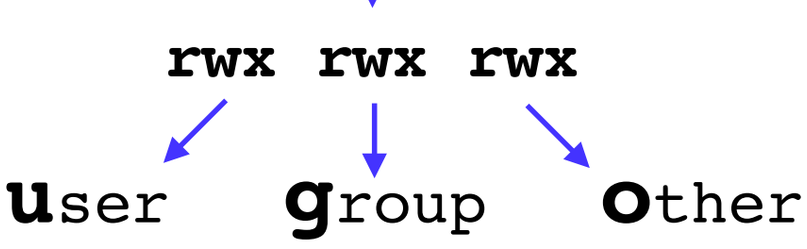
Symbol	Meaning
-	regular file
d	directory
l	link
c	special file
s	socket
p	named pipe
b	blocked device

6. User, groups and permissions

```
bombarelya@annona:~/Suaveolentes$ ls -lh
total 52K
-rw-rw-r--   1 bombarelya bombarelya  792 Feb 16 12:44 00_source.log
drwxrwxr-x 356 bombarelya bombarelya  20K Feb 14 18:16 00_sources
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 17 17:22 01_suaveolentes
drwxrwxr-x   2 bombarelya bombarelya 4.0K Feb 16 17:39 02_chloroplast
drwxrwxr-x   2 bombarelya bombarelya  20K Feb 17 17:09 benthamiana
```

Permissions | Links | Group Name | User Name | Size | Date | Filename

Symbol	Meaning
-	regular file
d	directory
l	link
c	special file
s	socket
p	named pipe
b	blocked device



6. User, groups and permissions

Permissions:

Information about the file:

`ls -l;` `ls -l <target>`

Examples:

```
-r--r--r--
-rwxr--r--

drw-rw-r--
```

-r--r--r-- → Readable for everyone

-rwxr--r-- → Readable for everyone, writable or executable only for the user-owner

drw-rw-r-- → Dir readable and writable for user and group, readable for everyone.

To change the onwer/group:

chowner owner:group file

To change permissions:

chmod [ugo] [+-=] [rwx] file
chmod [0-7] [0-7] [0-7] file

```
|rwx|rwx|rwx|
|421|421|421|
```

6. User, groups and permissions

Permissions:

sudo, is a program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user (normally the superuser, or root). Its name is a concatenation of the su command (which grants the user a shell for the superuser) and "do", or take action.

```
sudo cp ./myscript.pl /usr/local/bin
```

Exercise 2

1. List the permissions for the directory “exercise01”
2. Change the permissions to “r- - r - - r - -“
3. Change the working dir to “exercise01”

What happened? Why?

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

Recap movement commands:

Find the present working directory `pwd`

Change the working directory `cd`

List files in the working directory `ls`

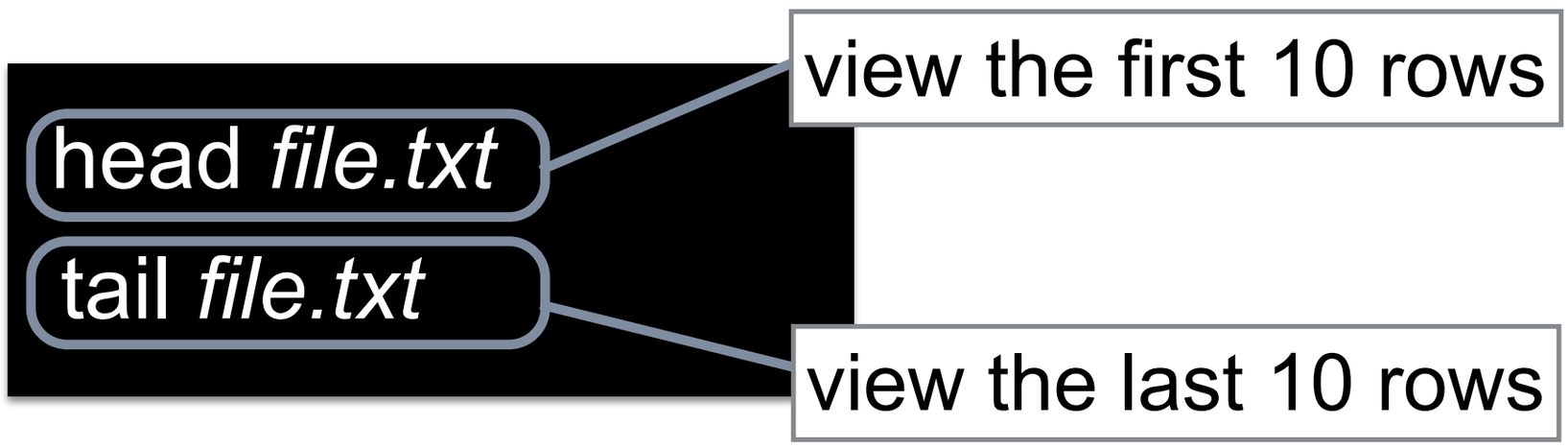
Reading text files:

less — view sections of a file via scrolling

```
less file.txt
```

note 1: “q” to quit!
note 2: SPACE to scroll forward N lines!
note 3: less -N <file> open the file with the line number
note 4: less +n -N <file> opens to a particular line number

Reading text files:



note 1: use -n X to view more than the default 10 lines
(e.g., head -n 20 *file.txt* to view 20 lines)

Manipulating text files:

cat — combine multiple files

```
cat file1.txt file2.txt
```

Combines and prints the file.

```
cat file1.txt file2.txt > combined_file.txt
```

Combines the files and redirects the STDOUT to a new file.

Note: You can use cat *file.txt* to view the contents of a file, but should you?

Manipulating text files:

mv — move and/or rename a file

```
mv file1.txt file1_new.txt
mv file1.txt  ~/Desktop/scripts/file1_new.txt
```

Renames the file in the present location

Renames the file and moves to a different directory.

Manipulating text files:

cp — copy a file

```
cp file1.txt file1_copy.txt
cp file1.txt ~/Desktop/file1_copy.txt
```

Copies a file in the pwd.

Copies a file to a new location.

Manipulating text files:

touch — marks a new timestamp on a file or
creates an empty file if none exists

```
touch existing_file.txt
touch new_file.txt
```

Mark a new timestamp

Creates a new *empty* file in the pwd

NB: use filenames in lowercase with underscores rather than spaces

Manipulating text files:

mkdir — creates a new *empty* directory

rmdir — removes an *empty* directory

```
mkdir trimmed
mkdir -p reads/trimmed
rmdir trimmed
```

Make a new directory

Makes a new directory ‘trimmed’ and parent directory ‘reads’

Removes a directory adding -Rf removes the directory and all files

Manipulating text files:

rm — removes files

```
rm file1.txt
rm -rf trimmed/
rm *.sam
```

removes a file in the pwd

removes a directory and all of the contents (recursive ‘r’ and forced ‘f’)

removes all files ending in .sam in the pwd

NB: With great power comes great responsibility.

Manipulating text files:

locate — find a file quickly not resident on all systems

find — find a file with more options but slower

```
locate file.txt
find file.txt
```

find file by name

find file by name, age, owner, permissions, timestamp, file type, location…etc

NB: locate is a global search whereas find is local.

7. Manipulating files

COMMAND	USE	EXAMPLE
ls	List infomrmation	ls -lha /home
pwd	Print working directory	pwd
cd	Change directory	cd ..
less	Open a text file	less file.txt
head / tail	Print the first / last 10 lines	head file.txt
cat	Concatenate and print two files	cat file1.txt file2.txt
mv	Move (rename) a file	mv from/file.txt to/file.txt
cp	copy files	cp file1.txt file1_copy.txt
touch	Create an empty file or new timestamp	touch test.txt
mkdir	Create a new directory	mkdir test_dir
rmdir	Remove an empty directory	rmdir test_dir
rm -rf	Remove a directory and files	rm -rf test_dir
locate	Find files by name	locate file.txt
find	Find files by multiple categories	find ./test file.txt

Exercise 3

1. Download the file “Araport11_genes.201606.pep.fasta.gz” using the following command:

   ```
   wget https://www.arabidopsis.org/download_files/Genes/Araport11_genome_release/Araport11_blastsets/Araport11_genes.201606.pep.fasta.gz
   ```

2. Unzip the file using the command

   ```
   gunzip Araport11_genes.201606.pep.fasta.gz
   ```

3. Print the first ten lines

Exercise 3

4. Create a new directory called “Sequences”

5. Move the “Araport11_genes.
201606.pep.fasta” file into the new directory

6. Print the last twenty lines

Controlling the STDOUT:

1. Redirecting into a file

Use of “>” symbol - Create and add (it overwrites!!!)

```
doej@annona:~$ ls > test_capture_ls.txt
```

Use of “>>” symbol - Append

```
doej@annona:~$ ls >> test_capture_ls.txt
```

Controlling the STDOUT:

2. Redirecting into a new command (pipe)

Use of “l” symbol -Pipe the STDOUT of a command into a new one

```
doej@annona:~$ ls exercise01/ | grep 01
test01.txt
```

Other commands to extract information from files:

COMMAND	USE	EXAMPLE
grep	Print matching lines as STDOUT	`grep 'ATG' myfile`
cut	Cut columns and print as STDOUT	`cut -f1 myfile`
sort	Sort lines and print as STDOUT	`sort myfile`
uniq	Select uniq words (-c to count uniq).	`uniq -c myfile`
sed	Replace ocurrences, print lines STDOUT	`sed 's/ATG/CTG/' myfile`
wc	Word count	`wc myfile`

Exercise 4

NOTE: To execute the exercise 4, it is necessary to know what a FASTA file is. For more information check:

https://en.wikipedia.org/wiki/FASTA_format

```
>AT1G01010.1 | NAC domain containing protein 1 | Chr1:3760-5630 FORWARD LENGTH=429 | 201606
MEDQVGFGFRPNDEELVGHYLRNKIEGNTSRDVEVAISEVNICSYDPWNLRFQSKYKSRD
AMWYFFSRRENNKGNRQSRTTVSGKWKLTGESVEVKDQWGFCSEGFRGKIGHKRVLVFLD
GRYPDKTKSDWVIHEFHYDLLPEHQRTYVICRLEYKGDDADILSAYAIDPTPAFVPNMTS
SAGSVVNQSRQRNSGSYNTYSEYDSANHGQQFNENSNIMQQQPLQGSFNPLLEYDFANHG
GQWLSDYIDLQQQVPYLAPYENESEMIWKHVIEENFEFLVDERTSMQQHYSDHRPKKPVS
GVLPDDSSDTETGSMIFEDTSSSTDSVGSSDEPGHTRIDDIPSLNIIEPLHNYKAQEQPK
QQSKEKVISSQKSECEWKMAEDSIKIPPSTNTVKQSWIVLENAQWNYLKNMIIGVLLFIS
VISWIILVG
>AT1G01020.1 | ARV1 family protein | Chr1:6915-8666 REVERSE LENGTH=245 | 201606
MAASEHRCVGCGFRVKSLFIQYSPGNIRLMKCGNCKEVADEYIECERMIIFIDLILHRPK
VYRHVLYNAINPATVNIQHLLWKLVFAYLLLDCYRSLLLRKSDEESSFSDSPVLLSIKVL
IGVLSANAAFIISFAIATKGLLNEVSRRREIMLGIFISSYFKIFLLAMLVWEFPMSVIFF
VDILLLTSNSMALKVMTESTMTRCIAVCLIAHLIRFLVGQIFEPTIFLIQIGSLLQYMSY
FFRIV
>AT1G01020.2 | ARV1 family protein | Chr1:7315-8666 REVERSE LENGTH=191 | 201606
MAASEHRCVGCGFRVKSLFIQYSPGNIRLMKCGNCKEVADEYIECERMIIFIDLILHRPK
VYRHVLYNAINPATVNIQHLLWKLVFAYLLLDCYRSLLLRKSDEESSFSDSPVLLSIKVL
IGVLSANAAFIISFAIATKGLLNEVSRRREIMLGIFISSYFKIFLLAMLVCCSFTSHLIP
NIEVPNFLSIP
```

Exercise 4

1. Using the file from the Exercise 3 “Araport11_genes.201606.pep.fasta” print all the lines with the “>” symbol

2. Print all the lines with the sequence “LCLCL”

3. Count how many sequences are in the file with grep

Exercise 4

4. Using the redirection symbol, capture all the sequence IDs from the file in a new file names “seqids.txt”

5. Using the pipe command print only the first ten sequence IDs

6. Count how many “kinases” are in the *Arabidopsis thaliana* proteome

Outline of Topics

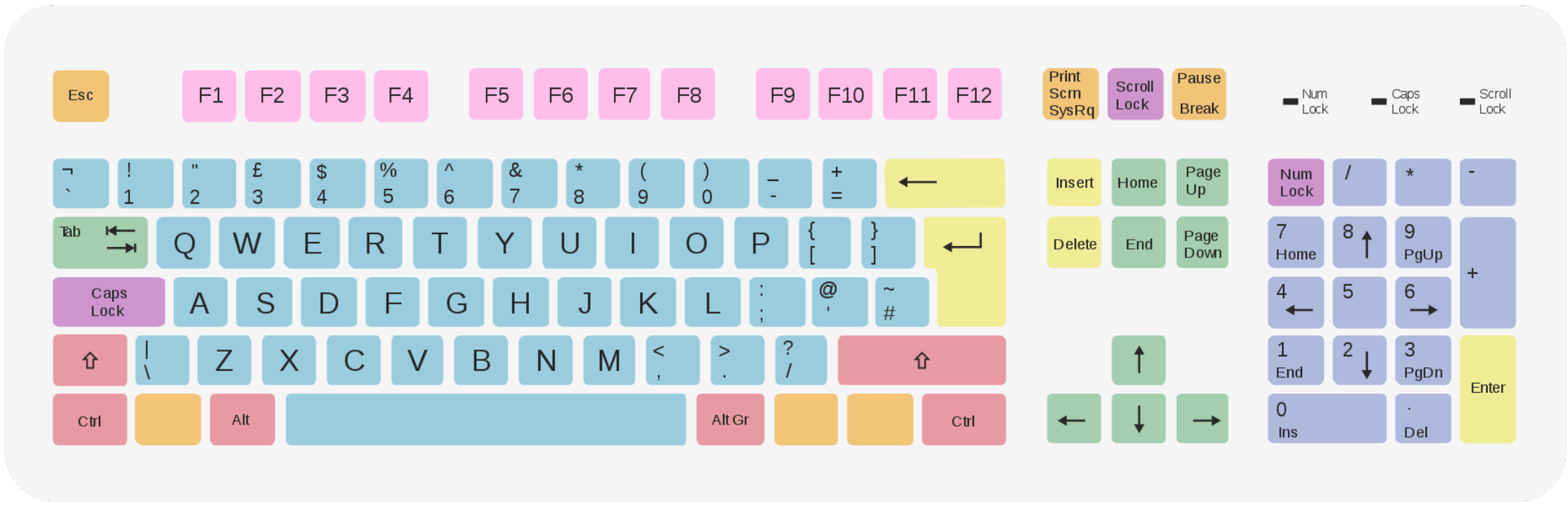
1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

8. Shortcuts

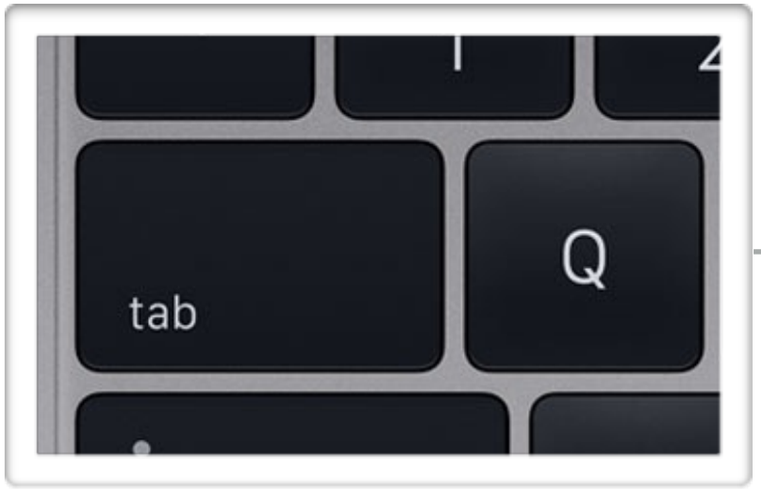
Shortcuts are combinations of keys with specific functions

MODIFIER KEY + CHARACTER KEY



https://en.wikipedia.org/wiki/Keyboard_layout

8. Shortcuts



Tab to autocomplete names or show all matches

```
doej@annona:~$ ls
exercise01
doej@annona:~$ cd e
```

Type “cd e”

Push Tab

```
doej@annona:~$ ls
exercise01
doej@annona:~$ cd exercise01/
```

It will autocomplete

8. Shortcuts

Tab to autocomplete names or show all matches

```
doej@annona:~/exercise01$ less test0
```

Type “less te”

↓ Push Tab

```
test01.txt  test02.txt
```

It gives you two options

↓ Type 1 + push Tab

```
doej@annona:~/exercise01$ less test01.txt
```

It will autocomplete

8. Shortcuts

Up arrow to move up in history
Down arrow to move back in history

Command history is in:

~/.bash_history

8. Shortcuts

COMMAND	ACTION
Tab	Autocomplete files or folder names
↑	Scroll up in the command history
↓	Scroll down in the command history
Ctrl + a	Go to the beginning of the active line
Ctrl + e	Go to the end of the line active line
Ctrl + u	Clear the line up to the cursor
Ctrl + c	Kill the active process
Ctrl + d	Exit the current shell
Ctrl + z	Put the active process in the background. Use command fg to recover it.

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

9. Environmental Variables

There are two types of variables in a Linux shell:

1) System variables

   Variables defined by the system such as home dir or the executable path.

2) User defined variables

   Variables defined by the user during a bash session.

9. Environmental Variables

System variables

SYSTEM VARIABLE	MEANING
SHELL	Shell name
BASH	Shell name
BASH_VERSION	Shell version
COLUMNS	Number of columns printed on the screen
LINES	Number of lines printed on the screen
HOME	Home directory
LOGNAME	Login name
OSTYPE	Operating system type
PATH	Path directories
PS1	Prompt settings
PWD	Current working directory
USERNAME	Username currently logged in to the system

9. Environmental Variables

Commands to interact with system variables

set — prints the commands for setting environmental variables

env — prints the environmental variables

set

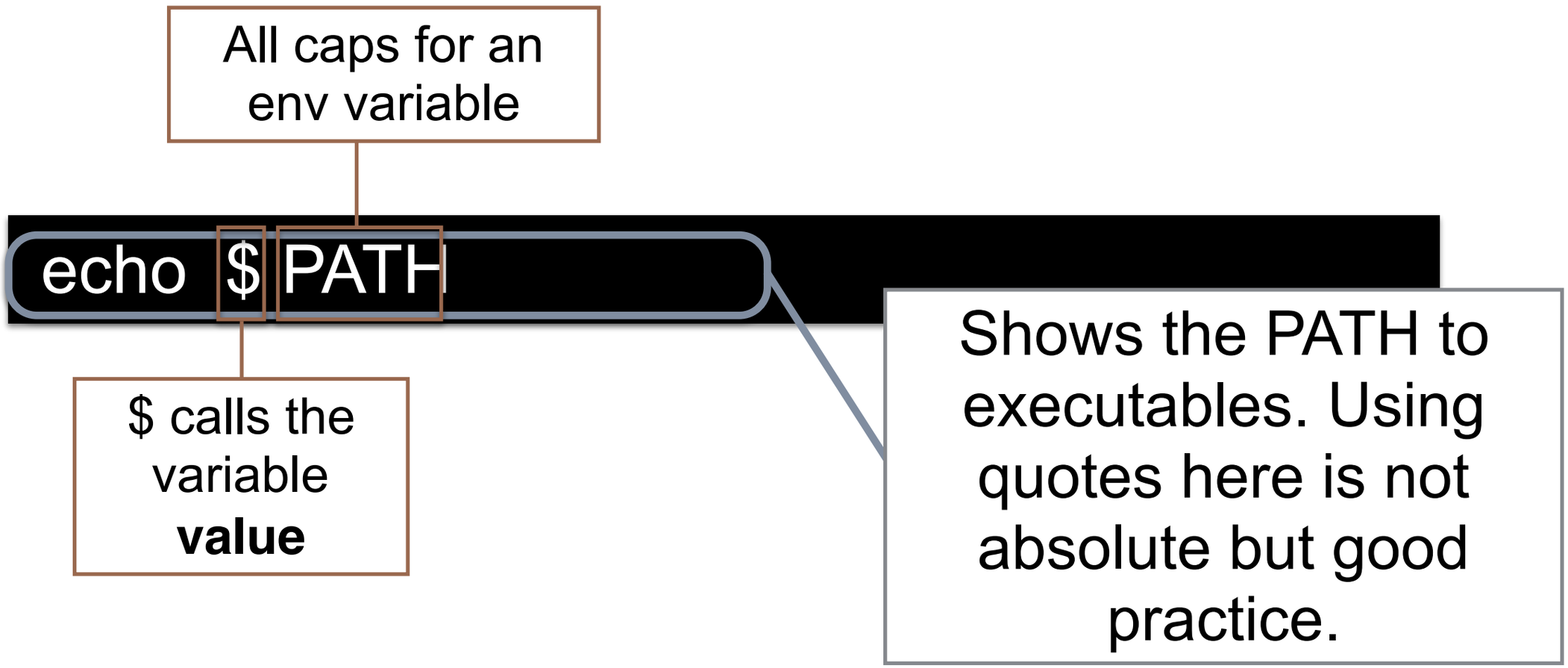
env

~ — bombarelya@annona: ~ — ssh bombarelya@159.149.160.131 | ...SA_work/PROJECTS/2019_MANGO_GENOME/01_Genome — zandror@annona: ~ — -bash | ~ — -bash

```
bombarelya@annona:~$ env
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;4
;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.
01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.z
01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01
*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.
=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.t
01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.web
;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;
.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aa
36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;
*.spx=00;36:*.xspf=00;36:
SSH_CONNECTION=188.216.6.130 57267 159.149.160.131 22
LESSCLOSE=/usr/bin/lesspipe %s %s
LANG=en_US.UTF-8
XDG_SESSION_ID=922
USER=bombarelya
PWD=/data/bombarelya
HOME=/data/bombarelya
SSH_CLIENT=188.216.6.130 57267 22
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
SSH_TTY=/dev/pts/0
MAIL=/var/mail/bombarelya
TERM=xterm-256color
SHELL=/bin/bash
SHLVL=1
LOGNAME=bombarelya
XDG_RUNTIME_DIR=/run/user/1001
PATH=/data/bombarelya/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
LESSOPEN=| /usr/bin/lesspipe %s
_=/usr/bin/env
OLDPWD=/data/bombarelya/Suaveolentes
```

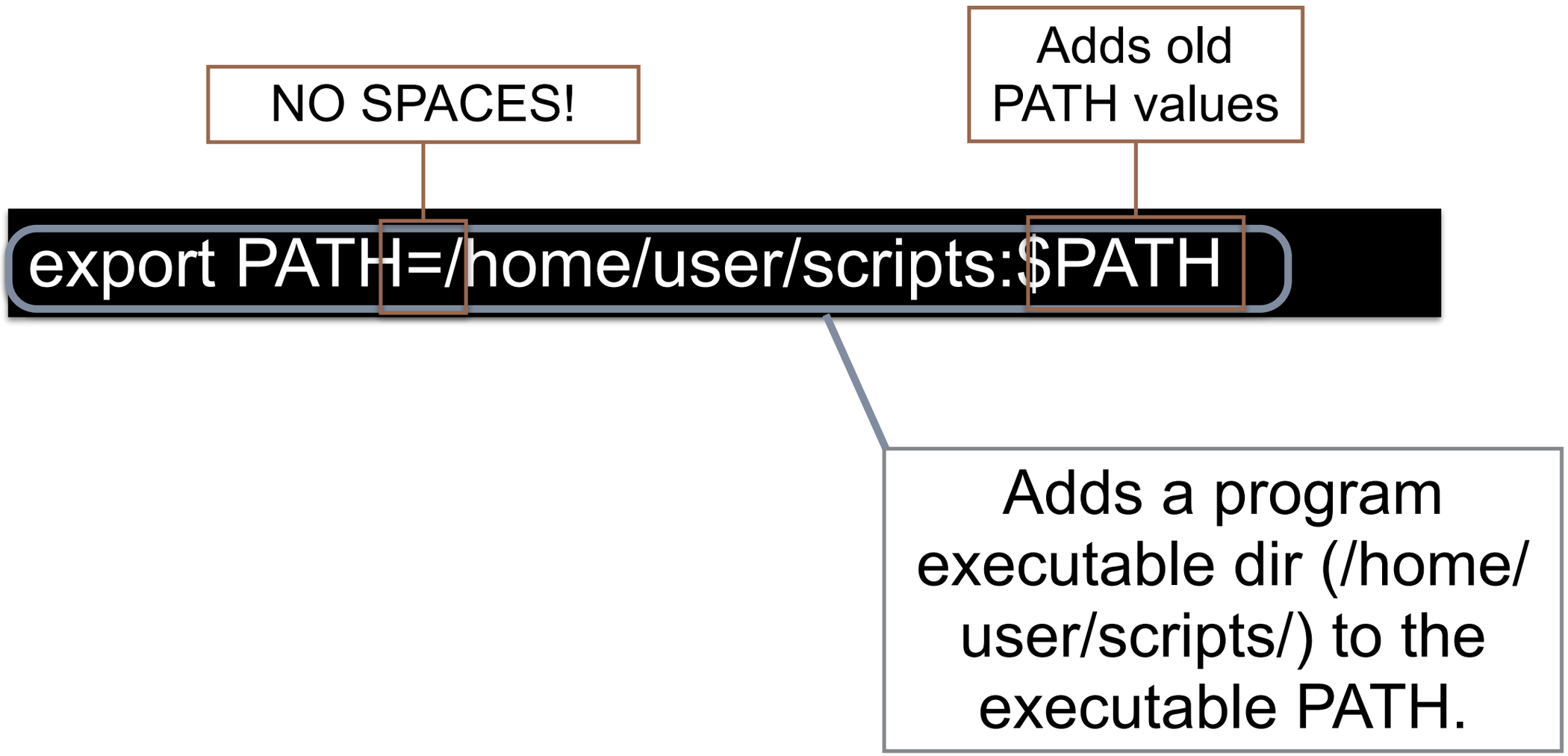
Commands to interact with system variables

echo — prints the environmental variable **value**



Commands to interact with system variables

export — Create an env. variable for all child processes



Commands to interact with system variables

What about adding multiple new paths

```
export PATH=/home/user/scripts:/home/user/software:$PATH
```

Paths are separated by a ':' including $PATH

```
export PATH=/home/user/scripts:$PATH
export PATH=/home/user/software:$PATH
```

9. Environmental Variables

User defined variables

Syntax is important

```
TEST_VAR='~/.bashrc'
echo $TEST_VAR
cat $TEST_VAR
```

This creates a defined variable that is a shortcut to login to BlueRidge

Returns the value of the variable.

Executes the command.

Note: Remove unwanted variables with: **unset** 'variable name'.

Commands to interact with system variables

source —execute commands from a file name

```
source home/user/scripts/myscript.sh
```

Will execute the shell script ‘myscript.sh’

9. Environmental Variables

Commands to interact with system variables

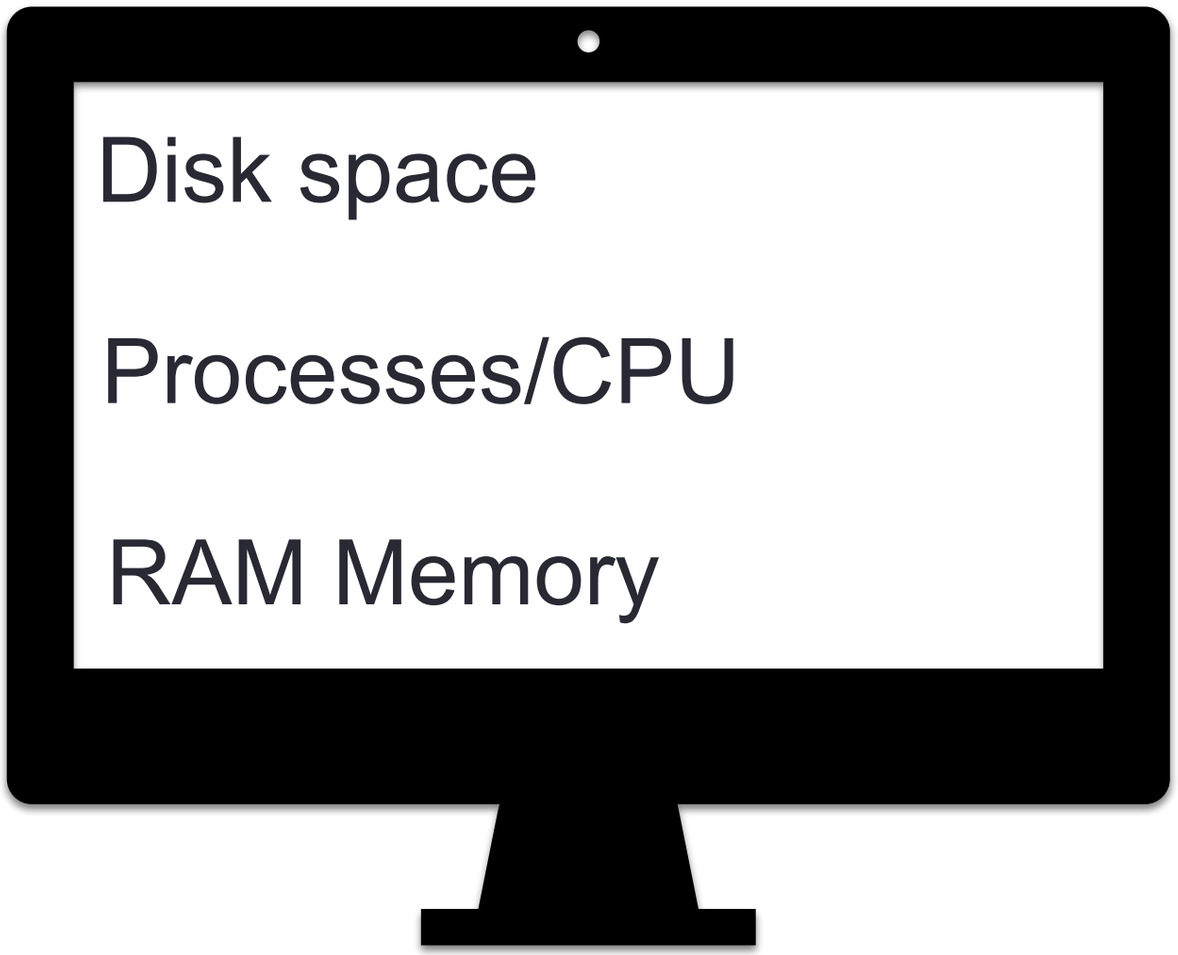
COMMAND	MEANING
env	Print environment variables
set	Print shell variables
echo	Print environment variable **value**
export	Create an env. variable available to all child processes
alias	Provide a short name for a long string
source	Execute commands from a file name

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

10. Monitoring resources



How much disk space is available?

df — disk free prints available disk space for all the partitions

```
[bombarelya@annona:~$ df -lh
Filesystem      Size  Used Avail Use% Mounted on
udev            126G     0  126G   0% /dev
tmpfs            26G  2.5M   26G   1% /run
/dev/sdb2       879G   17G  818G   2% /
tmpfs           126G     0  126G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           126G     0  126G   0% /sys/fs/cgroup
/dev/loop1       92M   92M     0 100% /snap/core/8592
/dev/sdb1       511M  6.1M  505M   2% /boot/efi
/dev/sdd1        11T  6.1T  4.3T  59% /data
tmpfs            26G     0   26G   0% /run/user/1001
/dev/loop2       92M   92M     0 100% /snap/core/8689
tmpfs            26G     0   26G   0% /run/user/1006
tmpfs            26G     0   26G   0% /run/user/1013
```

Note: Use df -lh to print in human readable form

How much disk space I am using?

du — disk usage prints the disk space used by a directory

```
[doej@annona:~$ du -lh
4.0K    ./exercise01
4.0K    ./.cache
4.0K    ./.gnupg/private-keys-v1.d
8.0K    ./.gnupg
36K     .
```

Note: Use du -lh to print in human readable form

10. Monitoring resources

Which process are being run?

top — prints real time processes

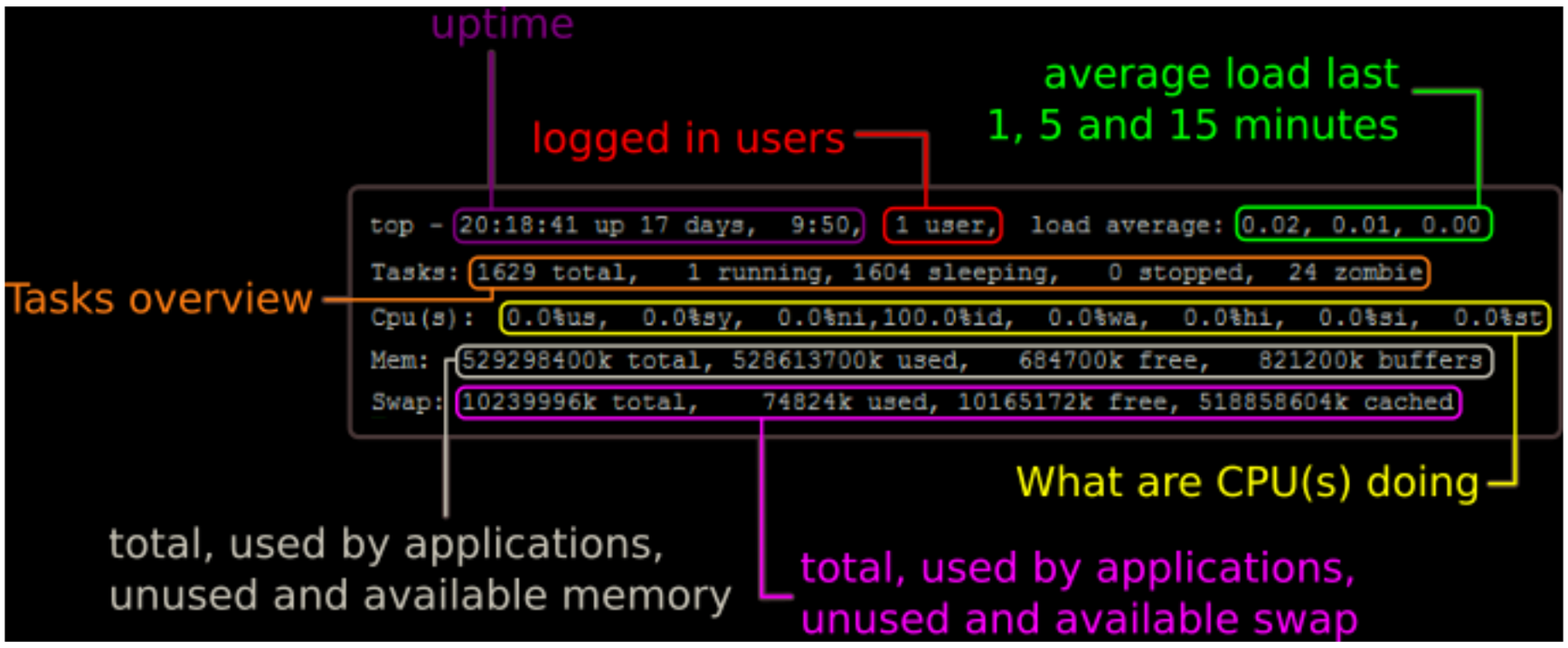
```
top - 00:25:27 up 22 days, 15:10,  1 user,  load average: 2.42, 2.01, 2.20
Tasks: 691 total,   2 running, 389 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.1 us,  0.0 sy,  0.0 ni, 96.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 26378531+total,  1369348 free,  1157384 used, 26125857+buff/cache
KiB Swap:  8388604 total,  8370684 free,    17920 used. 26078164+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 4661 bombare+  20   0 4433200  37976   5164 R 199.7  0.0   2:58.13 iqtree
 4740 doej      20   0   43452   4568   3232 R   1.0  0.0   0:00.19 top
    1 root      20   0   78324   9376   6736 S   0.0  0.0   1:24.34 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.54 kthreadd
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H
    7 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
    8 root      20   0       0      0      0 S   0.0  0.0   0:17.52 ksoftirqd/0
```

Note: Use “q” to exit/quit

Which process are being run?

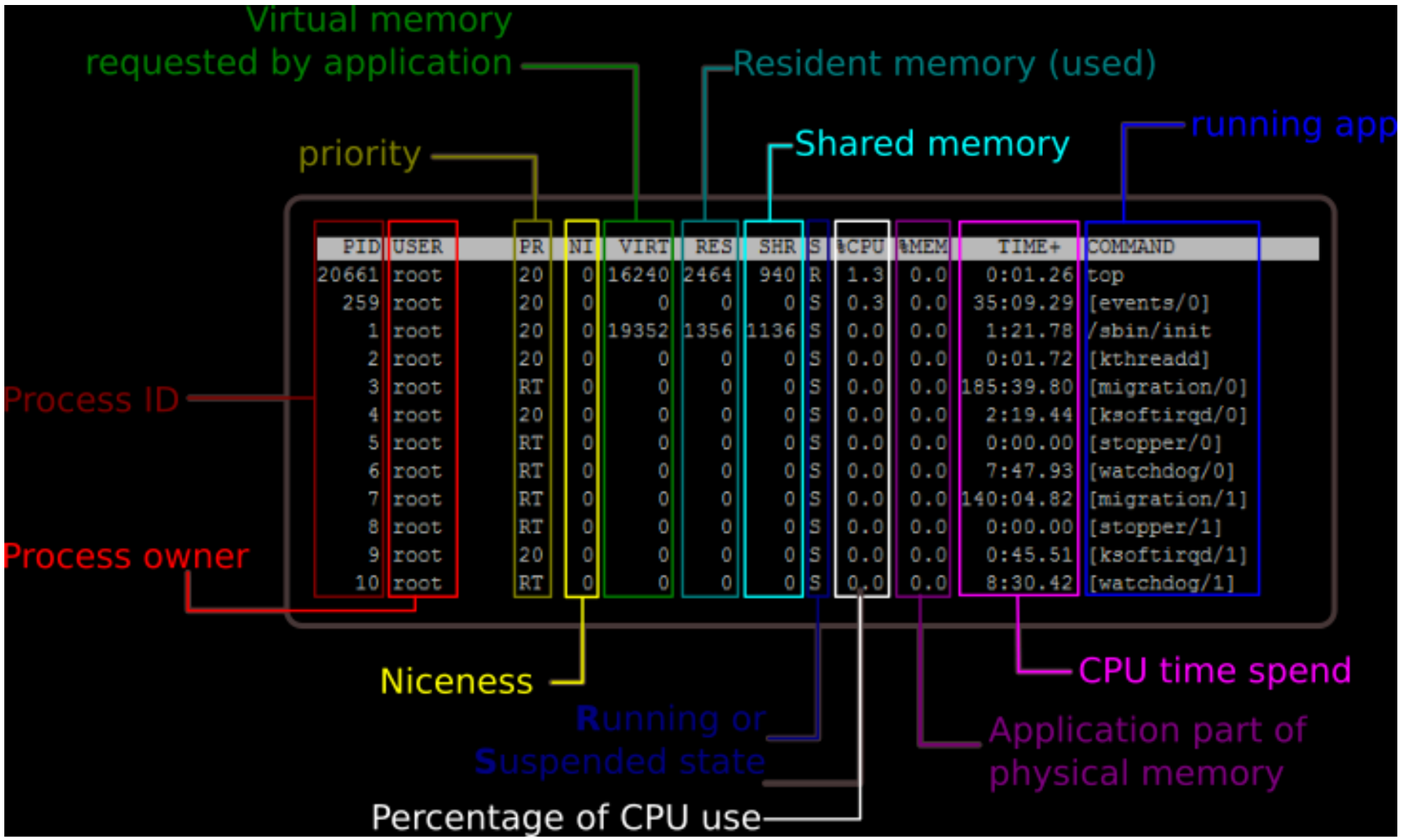
top — prints real time processes



https://www.svennd.be/top-explained-visually/

Which process are being run?

top — prints real time processes



https://www.svennd.be/top-explained-visually/

10. Monitoring resources

Which process are being run?

ps aux — report a snapshot of the current processes.

```
[doej@annona:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0  78324  9376 ?        Ss   Feb16   1:24 /sbin/init
root           2  0.0  0.0      0     0 ?        S    Feb16   0:00 [kthreadd]
root           4  0.0  0.0      0     0 ?        I<   Feb16   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        I<   Feb16   0:00 [mm_percpu_wq]
root           8  0.0  0.0      0     0 ?        S    Feb16   0:17 [ksoftirqd/0]
root           9  0.0  0.0      0     0 ?        I    Feb16  30:06 [rcu_sched]
root          10  0.0  0.0      0     0 ?        I    Feb16   0:00 [rcu_bh]
root          11  0.0  0.0      0     0 ?        S    Feb16   0:02 [migration/0]
root          12  0.0  0.0      0     0 ?        S    Feb16   0:04 [watchdog/0]
root          13  0.0  0.0      0     0 ?        S    Feb16   0:00 [cpuhp/0]
root          14  0.0  0.0      0     0 ?        S    Feb16   0:00 [cpuhp/1]
root          15  0.0  0.0      0     0 ?        S    Feb16   0:02 [watchdog/1]
```

Note: Use “q” to exit/quit

How much memory is available?

free men — get a detailed report on the system's memory usage

```
doej@annona:~$ free mem -g
              total        used        free      shared  buff/cache   available
Mem:            251           1           1           0         249         248
Swap:             7           0           7
```

Note: Use -g to see the results in Gb

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

Checking internet connection:

ping — test the ability of the source computer to reach a specified destination computer

```
ping www.google.com
ping www.unimit.it
ping 159.149.160.131
```

Note: Stop the process with CTR + C

Connecting to a remote host:

ssh — create a secure shell connection from the terminal.

```
ssh userid@servername
ssh userid@serverip
ssh userid@servername -p portname
```

username: does
remote host: 159.149.160.131
port: 22

Connecting to a remote host:

ssh — create a secure shell connection from the terminal.

```
ssh doej@159.149.160.131 -p 22
```

username: does
remote host: 159.149.160.131
port: 22

Copy a file FROM a remote host:

scp — copy a file from a host to your computer

```
scp username@address:<file_path> <local_location>
```

username: does
remote host: 159.149.160.131
port: 22

Copy a file TO a remote host:

scp — copy a file from your computer to a host

```
scp <local_location> username@address:<remote>
```

username: does
remote host: 159.149.160.131
port: 22

11. Networking

Download a file from a public FTP site:

wget — Download a file from a public site

```
wget ftp://ftp.solgenomics.net/genomes/Solanum_lycopersicum/annotation/ITAG3.2_release/ITAG3.2_proteins.fasta
```

Download a file from a public FTP site:

wget — transfer information to or from a web based location ***not available by default*** *on OS X*

```
wget ftp://ftp.solgenomics.net/tomato_genome/annotation/
ITAG2.4_release/ITAG2.4_assembly.gff3
```

curl — transfer information to or from a web based location

```
curl -O “ftp://ftp.solgenomics.net/tomato_genome/annotation/
ITAG2.4_release/ITAG2.4_assembly.gff3”
```

Note: curl option -O keeps the original file name. Use option -o to rename the file on download (eg, curl “http” -o xyz.txt)

Outline of Topics

1. What is Linux?
2. The terminal (or emulator)
3. Files and directories
4. Absolute and relative paths
5. Basic command structure
6. Users, groups and permissions
7. Manipulating files
8. Shortcuts
9. Environmental variables
10. Monitoring resources
11. Networking
12. Installing programs

1. Navigating in the terminal
2. Working with files and directories

12. Installing programs

Ways to Install Programs:

1) Using **Packages Managers**

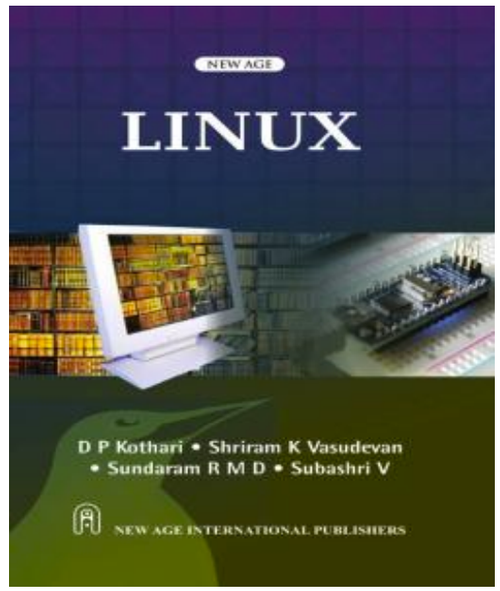
1.1) Graphical package manager
(Example: Synaptic for Ubuntu).
1.2) High level command-line package manager
(Example: apt for Debian, Yum for Red Hat)
1.3) Low level command-line package manager
(Example: dpkg for Debian, rpm for Red Hat)

2) Moving **Executable** file program to the PATH* and the libraries need to their corresponding locations.

2.1) Precompiled.
2.2) From-source.

Recommended Reading

- Author: Shriram, K V
- Book: Linux
- Publisher: New Age International
- ISBN: 81-224-3438-X, 978-81-224-3438-5
- Date:12/01/2014
- Free ebook from lib.vt.edu
- Chapter 1, 2, 3



Recommended Reading

- Author: Shotts, William E
- Book: The Linux command line : a complete
- Publisher: No Starch Press
- ISBN: 1-59327-389-4, 978-1-59327-389-7
- Date: 2011
- Free ebook from lib.vt.edu
- Section I.1, I.2, I.3, I.4, I.9