

Introduction to Linux

iCB2 – Introduction to Computational Biology and Bioinformatics

November 9, 2015

Emidio Capriotti

<http://biofold.org/>



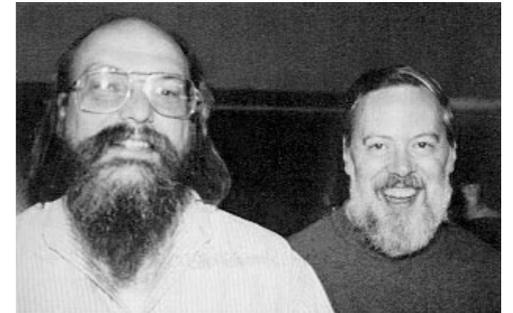
**Biomolecules
Folding and
Disease**

Institute for Mathematical Modeling
of Biological Systems
Department of Biology

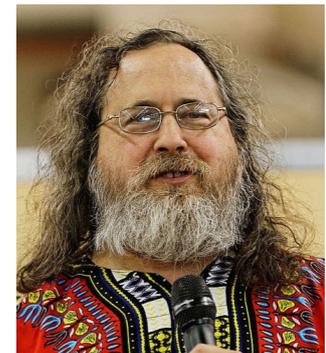

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Brief history

In 1969 a team of **Bell Lab researchers led by Thompson and Ritchie** developed a new operating system that was named UNIX in 1970.



In 1985, **Richard Stallman at MIT created Free Software Foundation (FSF)**. The dream was to create a “free” operating system. By 1990, he had almost everything except the kernel. This software stack is called GNU (GNU is Not Unix).



In 1991 **Linus Torvalds developed Linux**, a unix-like kernel, that was made free in 1992.

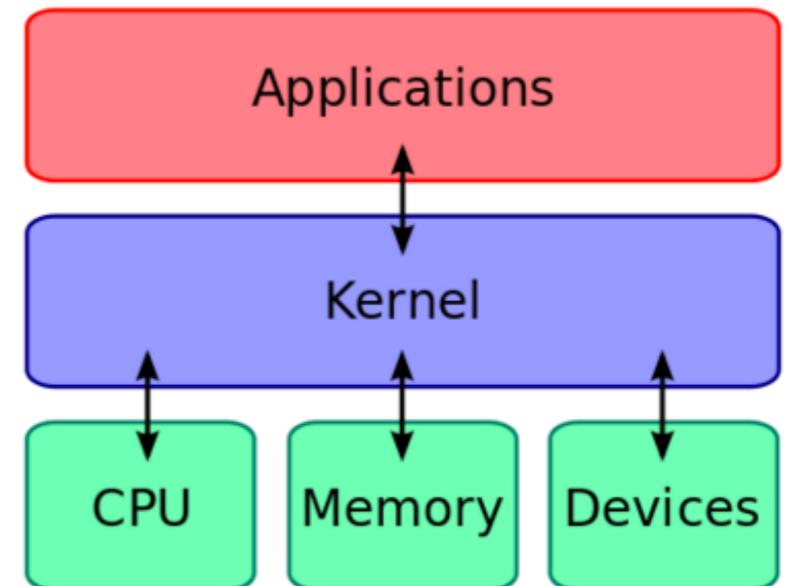


Another Unix like operating system is BSD which is included in OSX. Even on Windows has a Unix like system called CygWin.

Linux distribution

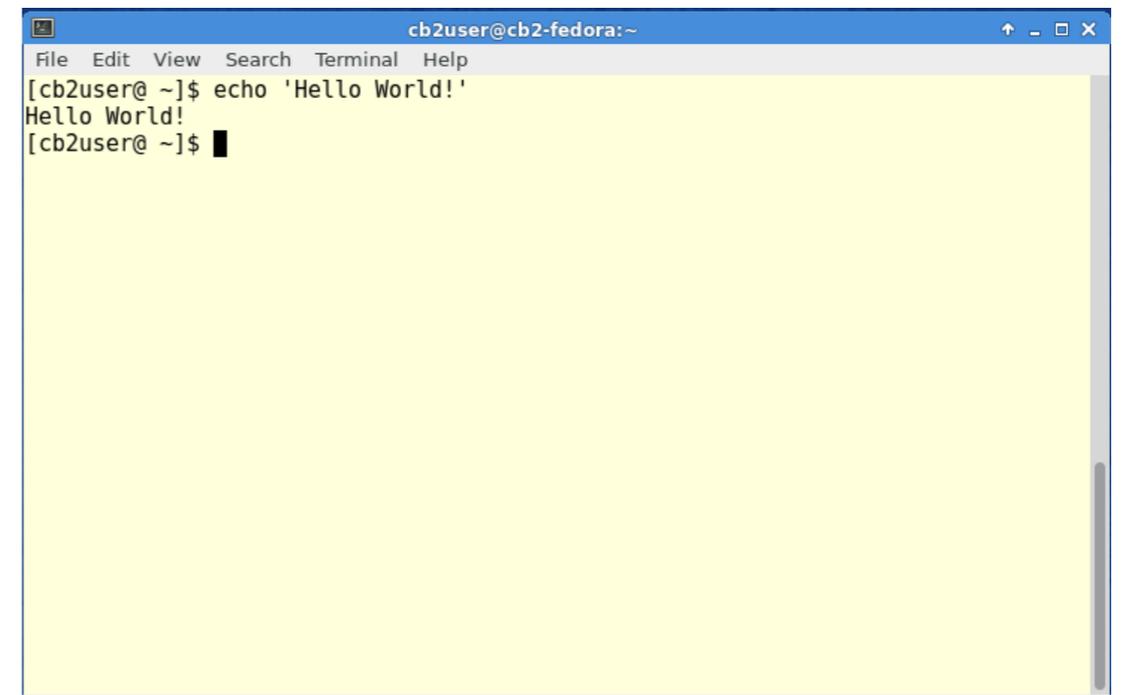
A Linux distribution includes

- The Kernel (Linux)
- An install system for the distribution
- Drivers
 - How the system can manage specific hardware
- A package manager
 - To install and update software
 - Usually different from one distribution to the other



Shell

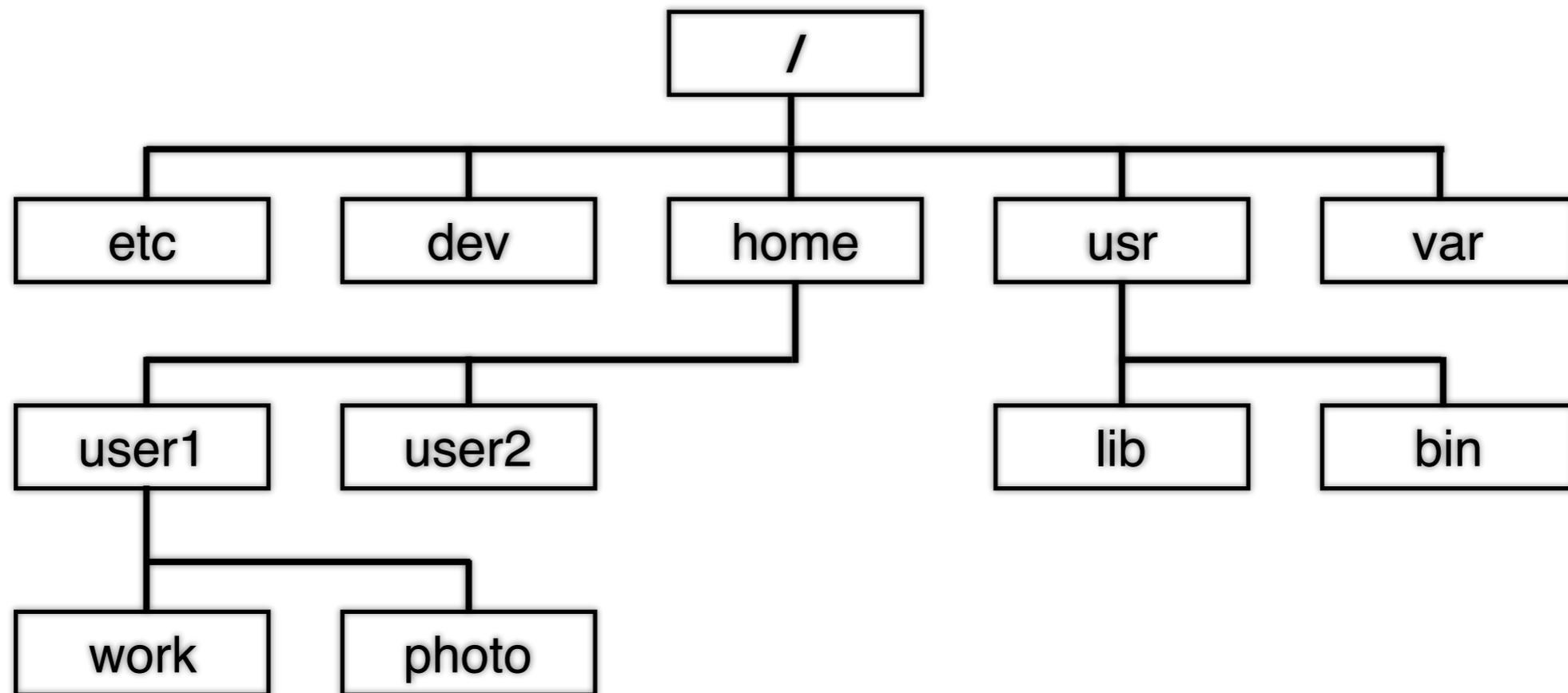
- It is the main interface with the system
- Can be used to:
 - Navigate the file system
 - Execute tools
 - Install software
 - Connect to other machines
 - Edit files
 - ... everything the system can do
- Also called Console, or Terminal



```
cb2user@cb2-fedora:~  
File Edit View Search Terminal Help  
[cb2user@ ~]$ echo 'Hello World!'  
Hello World!  
[cb2user@ ~]$ █
```

Filesystem

The **filesystem** has a **hierarchical structure** that allows to store files and directories



Basic Commands

- `man commandname` - will show you help about a particular command.
- `ls dirname` - shows a directory listing.
- `cd dirname` - changes directory
- `mkdir dirname` - creates dir
- `cp path1 path2` - copies dir or files
- `mv path1 path2` - renames or moves a file/dir.
- `rm namefile` - removes a file.

Note: `rm -rf` is a special command that will remove everything from the current directory without prompting.

If you accidentally execute this command in "/", it will try to wipe out everything from your computer.

Special Characters

The following characters have special meanings

- `*` - represent “one or more” characters
- `?` - represents one character
- `>` - redirect the output to a file
- `| (pipe)` - redirect the output to the standard input (STDIN)
- `2>` - redirect to standard error (STDERR)

To redirect the list of files and directories in your current location to the file names.txt

```
> ls > names.txt
```

Handling Files

- **cat** - dumps the content of the file to output.
- **less or more** - show an input page by page.
- **wc** - count the lines, words, and characters. head
- **head** - shows the first few lines of a file.
- **tail** - displays the last lines of the file.

The option “-n number_of_lines” for “head” and “tail” allows to specify the number of lines to shows. If you want to skip the first 2 lines of a file,

```
> tail -n+3 <filename>
```

File Path

The path indicates the **location of a resource in the filesystem**.

- **pwd** - current path
- **"."** - current directory.
- **".."** - parent directory

Exercise

What is the absolute path of your home? Using a relative path, go up all the way to root and come back again in the same directory

Environment variables

An environment variable is defined in bash as follows:

```
> export foo=bar
```

To run a command (or program), the location of the program has to be in a particular environment variable called PATH. You can add to the existing \$PATH by adding to it like this:

```
> export PATH=$PATH:/some_dir/of/my/choice
```

You may add line like this in your .bashrc or .bash_profile. You can also run a program by calling it by absolute path.

You can find the absolute path of a command by using “which command_name”.

File permissions

There are **three kind of permissions: read, write, and execute**. A file need to have executable permission in Linux to run.

You can change the permission of a file that you own by the command `chmod`.

You can **check the permission of a file by “ls -l”**. `chmod` is run like this:

```
> chmod a+wx filename
```

Exercise

Change the permission of a file that you own to executable by everyone.

Important Commands (I)

- **cut** - extracts the columns from a text file. The default field separator is tab. To extract 2nd column for a tab-delimited text file

```
> cat foo.txt | cut -f 2
```

- **sort** - sort the line of a input in alphabetical order.

```
> cat foo.txt | sort
```

- **uniq** - Removes duplicate lines if the are consecutive. You have to use sort to use uniq correctly.

```
> cat foo.txt | sort | uniq
```

- **wget** - is a swiss-army-knife of web downloader. wget URL

Important Commands (II)

- **grep** - searches for a pattern in file or input.

```
> grep 'string' filename
```

- **tr** - “translates” one or more character in a string to another. Import option “-d string” for deleting strings.

```
> echo 'emidio' | tr '[a-z]' '[A-Z]'
```

- **find** - searches files recursively going into a directory hierarchy

```
> find dirname -name filename
```

Exercise

Download the file of the human proteome from ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/reference_proteomes/Eukaryota/UP000005640_9606.fasta.gz using `wget` and find how many sequences are contained.

Compression and Archiving

The two most common **compression programs are gzip and bzip2**.

Traditionally they take .gz and .bz2 extensions, respectively. If you just type gzip file, the compressed file will be automatically named file.gz and original file will be replaced.

```
> echo "Hello there." | gzip -c >hello.gz  
and unzip with  
> gunzip hello.gz
```

To **compress many files** (including directories) you need to use **“tar”** first then use gzip/bzip2.

```
> tar -c Downloads/ | gzip -c >downloads.tar.gz  
or  
> tar -czvf downloads.tar.gz Downloads/
```

For uncompress use the option **“-xzvf”**. For bzip2 **“-czvf”** is replaced with **“-cvbf”**

Program Compilation

There is a **standard way to compile a C program in Linux**. Almost all the software are distributed as .tar.gz files. These are source codes. You should compile and install the software like this:

```
> tar -xvzf foo.tar.gz
> cd foo
> ./configure
> make
> make install
```

The last install step may require you switch to “root”.

Exercise

HMMER is a software for sequence analysis using profile hidden Markov model. Download HMMER source code from <http://selab.janelia.org/software/hmmer3/3.1b1/hmmer-3.1b1.tar.gz> then install the software in your machine.

Problem Set

- **Problem 1**

PFAM is a database of domains. It also provides pre-calculated domains for all proteomes. The current version can be found here

ftp://ftp.sanger.ac.uk/pub/databases/Pfam/current_release/proteomes/.

Each file is a proteome identified by its taxonomic ID. Human has the ID 9606.

Each of these files is tab-delimited and the 6th column is the domain ID.

Download the human proteome file using `wget`. After downloading write just a single line of `bash` to find how many domain types (unique domains) are there in human genome.

- **Problem 2**

E. coli MG1655 is the standard reference strain of *E. coli*. The protein FASTA file for this strain can be downloaded from ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr_MG1655_uid57779/NC_000913.faa.

Using just `bash` commands can you find out what is the average length of protein in this strain?