

# BASICS IN COMMAND LINE



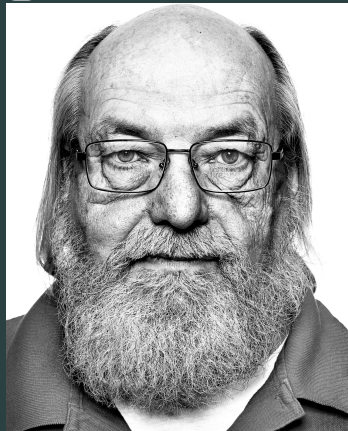
Juan Martínez Palazón  
@huan\_mp

**Let's check out the basics! :)**

## TABLE OF CONTENTS

0. The Shell
1. Move faster in CLI
2. Find files
3. Read files
4. Identify the differences
5. History and command persistence
6. Practical example

**sh**



**1971**

**Thompson  
Shell**

(Ken Thompson)

**sh**



**1979**

**Bourne  
Shell**

(Stephen R. Bourne)

**bash**



**1989**

**Bourne-again  
shell**

(Brian Fox)

## The following shortcuts will help you moving faster

CTRL

<-

Jump over words

CTRL

->

HOME

Move to the beginning / end of the line

END

CTRL

K

Delete the end of the line

CTRL

L

Clear the screen

CTRL

D

Close the session

\$ move faster\_

## Well, where are my demo files?

Search all files starting with "num\*"

```
$ find . -name 'num*'
```

List them with sizes

```
$ find . -name 'num*' -exec du -h {} \;
```

Discard big files

```
$ find . -name 'num*' -size -500k -exec du -h {} \;
```

\$ find\_

## That's fine. I want to see the content

Show the whole file

```
$ cat nums1
```

Navigate through the file

```
$ more nums2
```

```
$ less nums2
```

**\$ cat, more & less\_**

Show number of lines

```
$ wc -l nums2
```

Show the beginning of the file

```
$ head nums2
```

```
$ head nums{2|4}
```

All lines except last X

```
$ head -n-X
```

All lines except first X

```
$ tail -n+X
```

Display data being appended to a file

```
$ tail -f /var/log/syslog
```

Combining both

```
$ cat file | head -n 25 | tail -n 5
```



**\$ wc, head & tail\_**



## So... Do I have two identical files?

Show differences in command line

```
$ diff nums2 nums4 --side-by-side --color
```

Show them in Vim

```
$ gvimdiff nums2 nums4
```

Calculate the hash

```
$ md5sum nums{2,4}
```

**\$ diff, gvimdiff & md5sum\_**

## Uf... I'm not going to remember!

Check the command history

```
$ history
```

Store the command as an alias

```
$ alias l='tree -L 1'
```

Add stuff to your profile

```
$ vim ~/.bashrc
```

**\$ history, alias & profile\_**

**Enough. Let's put this in practice**

(demo time)

# Ultra-simple one-liner events monitor

mydb.csv

```
28,375146,email-received,2019-08-22 20:29:48.346071
27,375462,email-sent,2019-08-22 19:56:56.854571
26,375580,email-sent,2019-08-22 19:56:12.463198
25,375566,email-sent,2019-08-22 19:55:24.155422
23,375501,email-sent,2019-08-22 19:52:42.558356
22,375555,email-sent,2019-08-22 19:50:41.026608
21,324899,email-received,2019-08-22 17:20:31.838284
20,320395,email-received,2019-08-22 17:20:21.442271
19,374966,email-received,2019-08-22 16:50:35.842953
...
```

Take columns 3 and 4, using comma as delimiter

Filter lines with "email-received"

Sort by second column

```
$ cut -f 3,4 -d ',' | mydb.csv | grep "email-received" | cut -d ' ' -f 1 | sort -k 2 | uniq -c
```

From this file

Remove the last column, using space as delimiter

Remove coincidences and count repetitions

# Ultra-simple one-liner events monitor

Using **watch** we can see how the file is being modified

```
Every 1,0s: ./monitor.sh  
  
3 email-received,2019-08-21  
9 email-received,2019-08-22  
1 email-received,2019-08-24  
3 email-received,2019-08-25  
2 email-received,2019-08-26  
1 email-received,2019-08-27  
4 email-received,2019-08-29
```

Run every one second

Script containing previous command

```
$ watch -n 1 -d monitor-script.sh
```

Highlight changes between updates

# Thanks!

Does anyone have any question?

— — — —  
**Juan Martínez Palazón**  
**@huan\_mp**