Beginners Guide to the Command Line

Petr Stepanov

petrs@jlab.org

Jun 1, 2021

What the heck is a Commánd Line

– A user interface based on lines of commands. User interacts directly with computer by typing commands.

Why master the Command Line

Sometimes you have to. No graphic user interface (GUI) is installed on the system. Setting up a web server or a super-computer.

Good on your resume. Command Line is #6 programming skill according to the 2020 statistics.

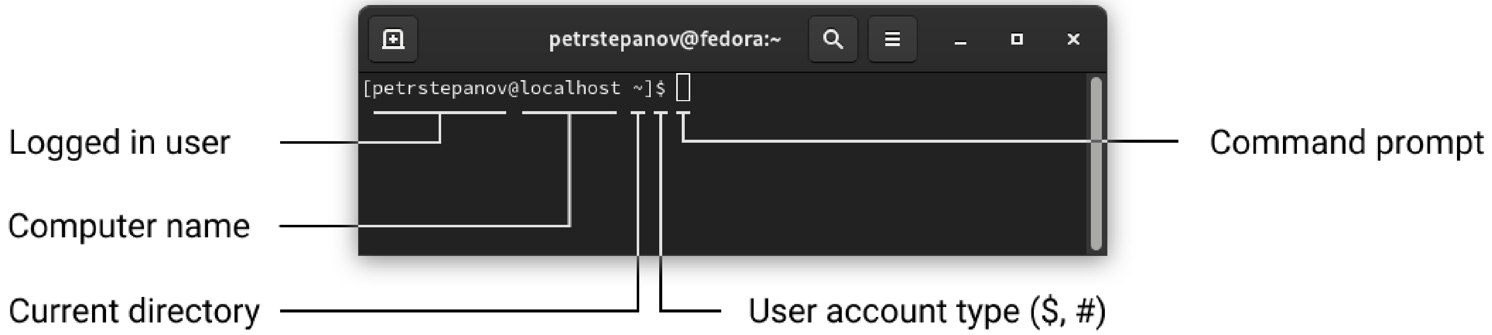
Saves your time. Automation of the repetitive tasks by writing scripts. E.g. download source, compile...
Example: run out of space!

How to access a Command Line on your computer?

- Windows computer. Install Windows Subsystem for Linux (WSL).
- Mac or Linux computer. Find and launch Terminal application.

Basic Structure and Concepts

Below you can see the Command Prompt. Following output signifies that computer is awaiting for the user input:



```
[petrstepanov@localhost ~]$
```

Command syntax

Every command has a name usually followed by arguments – options and parameters.

Options	pre-defined arguments that modify command output.
Parameters	arguments that provide information to the command or one of its. options

`command`	**`command --option`**	**`command <parameter>`**
Example: `cal`	Example: `cal --vertical`	Example: cal 2025

Tip: try combining options and parameters like ``cal -v june``

Navigating the Diréctory Tree

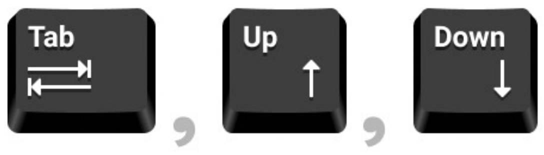
We start from learning basic commands that will allow us browsing the file system.

- **`pwd`** print working directory.
  Note: slash symbol / indicates the root (first) directory and also acts as a separator between directories.
- **`ls`** list directory contents.
  Note: home folder in Linux is similar to the Windows Explorer: Documents, Music...
- **`cd <directory>`** change directory.
  `cd ..` return to parent directory; **`` `cd -` ``** to previous directory.
  `cd` go to home folder.

Relative and Absolute Paths

Demonstrate relative (./) and absolute paths (/) specification for navigation (cd) and listing (ls) the file system. Show commands containing relative paths like ./../

HINT: Autocompletion and History



Command and directory names can be auto-completed in command prompt by pressing the TAB key on the keyboard. Access your history with UP and DOWN keys.

UNIX Directory Tree

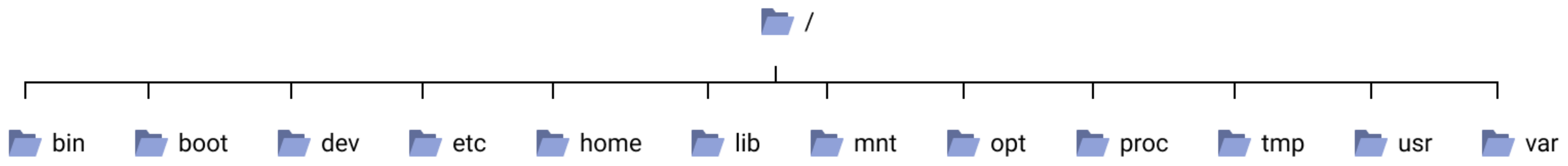
However a deeper look at Linux file system is somewhat confusing for the Windows user:

- Where is my drive C?
- Is operating system installed in "C:\Windows"?
- I cannot find the "C:\Program Files".
- Why myFile.txt, MyFile,txt, myfile.txt are different files (case sensitive file system)?

Filesystem Hierarchy Standard (FHS)

A convention that defines the directory structure in Linux distributions.

- Top file tree location is named root, denoted with `/` (slash or forward slash).
- Any document, directory or device – hard drive, printer, keyboard – is a file.



Question: how to print the above list of folders in Terminal?

UNIX Directory Tree (Continued)

Below we will discuss the most common Linux directories that are direct descendants of the root (/) tree node.

bin
binaries
- cat
- cd
- pwd
- ls

sbin
system binaries
- fdisk
- shutdown

binaries for maintance and administrative tasks

boot
don't mess around

kernel, bootloader (grub), splash screen, ...

dev
devices
- sda
- sda1

wifi, keyboard, touchpad...

etc
literally et cetera
- fstab

System configuration, not per-user settings.

Such as: repositories, system-wide mount points.

home
users' files and settings

Every user has its own "home folder" with Documents, Music, Videos, etc...
- bubbles
- julian
- ricky

lib
libraries

Files that applications use to perform various functions

mnt
mounted volumes

Originally for mounting filesystems of hard drives and removable devices.

Now linux use **media** to automatically mount removable devices

opt
optional

For software that is not part of default installation

proc
process information
- cpuinfo
- uptime

System memory, hardware configuration per process number
- 1
- 10
- ...

tmp
temporary folder

Files stored by currently running applications

usr
user system resources

Binaries, libraries used by the user. Non-essential for basic system operation.
- local/bin

Self compiled programs
- share

Shareable, architecture-independent files - icons, fonts etc...

var
variable size folder

Files and directories that grow in size: crashes, logs.
- crash
- log

“Man” is for Manual

`man` command in Linux is displays the user manual of any command that we can run in the Terminal:

```
man <any-command>
```

Example: man ls

“Echo” displays text

`echo` command outputs your text to standard output (screen). Give it a try!

```
echo <any-string>
```

HINT: Cancel Command Execution

Terminate the execution of any command by pressing CTRL+C key combination on your keyboard:

Example: sleep 100

Now, terminate the \`sleep\` command with CTRL+C.

Create, View and Modify Files

Creating, viewing and modifying files is a *second most important skill* you learn after mastering the file system navigation .

- **`touch <file1>`** create empty file(s) on the file system.
  Note: you can specify file path too: `touch /path/to/my/filename.txt`
- **`cat <file1> <file2>`** output file(s) to Terminal, concatenate files.
  Note: if provided one file, its contents will be displayed. multiple files,
- **`nano <file>`** command-line text editor with .
  `CTRL+x` exit nano.

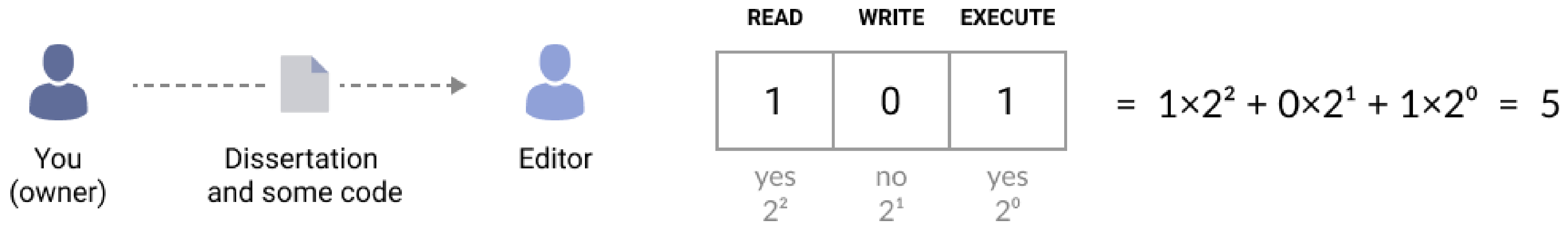
Example – create and join files

Create two files with sample content. Output files on the screen. Join files on the screen:

`touch file1.txt file2.txt`	
`nano file1.txt`	type contents of the first file; save and exit
`nano file2.txt`	type contents of the second file; save and exit
`cat file1.txt file2.txt`	outputs concatenated files on the screen

File Permissions in UNIX

You just wrote a paper and want to submit it to a journal. Editor will review your paper. We will let him read the paper, execute the code but not modify it (no write).


You (owner)
Dissertation and some code
Editor
READ
WRITE
EXECUTE
1
0
1
yes
no
yes
2^2
2^1
2^0
$= 1\times2^2 + 0\times2^1 + 1\times2^0 = 5$

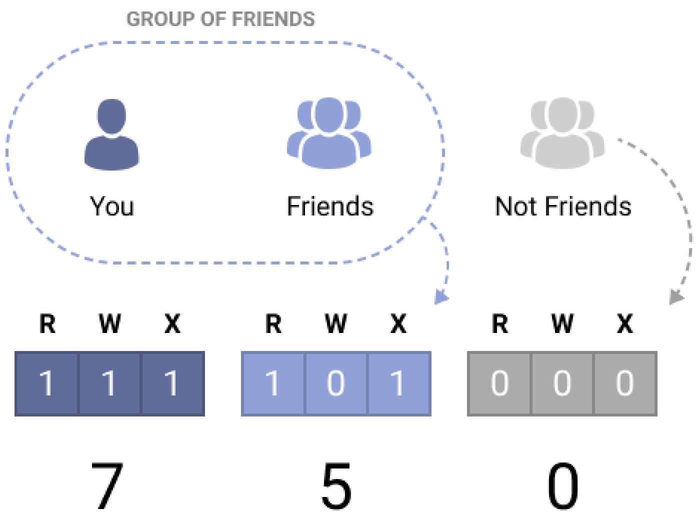
Basically, you convert a three-digit binary number (101) to a decimal number (5)

Questions:

- What does permission number "7" stand for?
- I want to let the editor only READ my dissertation. What number is that?
- I do not like the editor and I want restrict him from reading, writing and executing my stuff. What permission number corresponds to that?

File Permissions in UNIX (Continued)

In Linux users are agglomerated in groups. For instance, like on a Facebook. You have a group of your Friends and the rest of Facebook. We provide individual permission numbers for yourself, users in your group (friends) and others.



Viewing and editing file permissions

- To view permissions for files in current directory, list the files with option `-l`:
  `ls -l`
- To modify permissions of a certain file, use `chmod` command:
  `chmod 777 <file-name>`

  Use the pattern above to determine the file desired octal mode, 777, 751, etc...

Control Flow

Output of any command can be redirected from the screen (standard output, std::out in C++) to a file very easily:

```
command > filename
```

Example: cal `> my-test-file.txt`

If in above example `my-list-of-files.txt` already exists, it will be overwritten. Use two brackets `>>` In order to append the command output to the existing file.

```
command >> filename
```

Example: cal -v `>> my-test-file.txt`

HINT: Another way to create a file

I think we've just discovered a new way to create a file! Let's try it out:

```
echo Hello World! > new-file.txt
```

Cool Command Line Programs

Command Line is not necessary that boring. Check out neat Command Line programs:

- **Figlet**. Generate awesome ASCII art fonts. Fun to use in emails!

List available fonts:	`ls /usr/share/figlet/ | grep flf`
Use specific font:	`figlet -f <font-name> <your-text>`

- **Cowsay**. Draw a cow with text balloon.

List all cowfiles:	`cowsay -l`
Use cow file:	`cowsay -f <file>`
Support multiple lines	`cowsay -n`

- **Lolcat**. Spice up the Terminal output with colors!

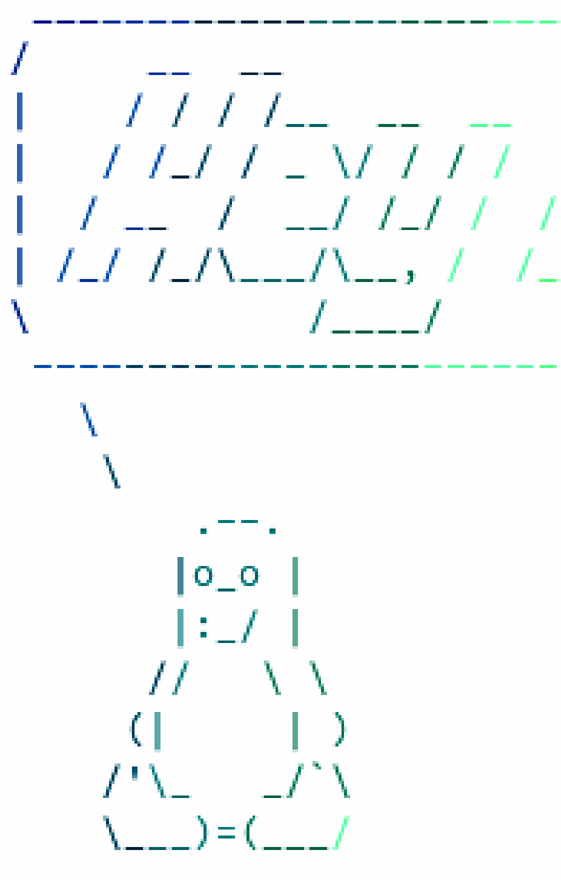
Piping

Pipe character `|` is used to use output from the command1 as input for command2. Let's see how piping works the output of the above commands!

Same output: `figlet hello` and `echo Hello | figlet`

Cool stuff you can do with piping:

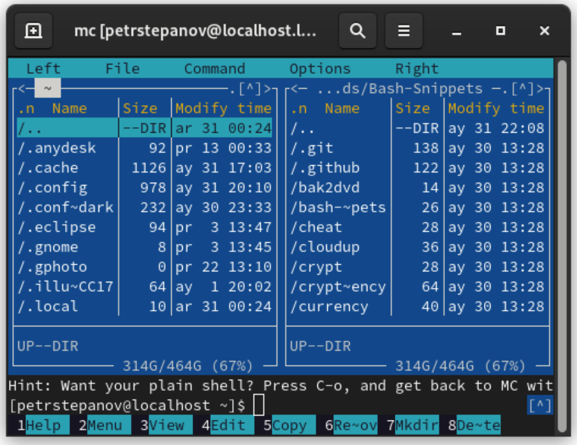
```
echo Hey Bro! | figlet -f slant | cowsay -n -f tux | lolcat
```



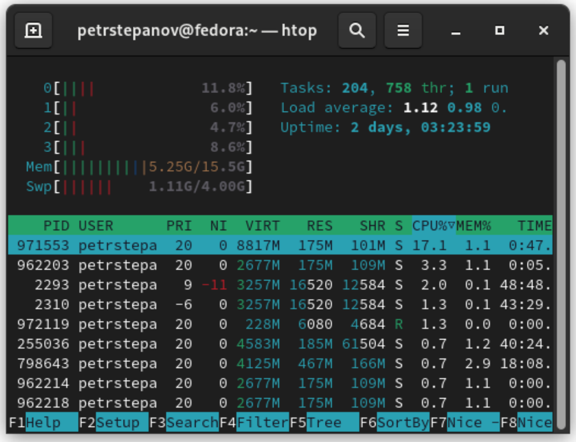
```
 ----------------------
/    __  __
|   / / / /__  __  __
|  / /_/ / _ \/ / / /  /
| / __  /  __/ /_/ /  /
| /_/ /_/\___/\__, /  /_
\           /____/
 ----------------------
   \
    \
        .--.
       |o_o |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
```

Useful Command Line Programs

Command Line is not just a single scary line. There are many programs that take advantage of the so-called “text-based User Interfaces”.


mc [petrstepanov@localhost.l...
Left File Command Options Right

```
┌<─ ~ ───────────────────.[^]>┐┌<─ ...ds/Bash-Snippets ─.[^]>┐
│.n  Name     │Size │Modify time││.n  Name     │Size │Modify time│
│/..          │--DIR│ar 31 00:24││/..          │--DIR│ay 31 22:08│
│/.anydesk    │   92│pr 13 00:33││/.git        │  138│ay 30 13:28│
│/.cache      │ 1126│ay 31 17:03││/.github     │  122│ay 30 13:28│
│/.config     │  978│ay 31 20:10││/bak2dvd     │   14│ay 30 13:28│
│/.conf~dark  │  232│ay 30 23:33││/bash-~pets  │   26│ay 30 13:28│
│/.eclipse    │   94│pr  3 13:47││/cheat       │   28│ay 30 13:28│
│/.gnome      │    8│pr  3 13:45││/cloudup     │   36│ay 30 13:28│
│/.gphoto     │    0│pr 22 13:10││/crypt       │   28│ay 30 13:28│
│/.illu~CC17  │   64│ay  1 20:02││/crypt~ency  │   64│ay 30 13:28│
│/.local      │   10│ar 31 00:24││/currency    │   40│ay 30 13:28│
│UP--DIR                        ││UP--DIR                        │
└──────── 314G/464G (67%) ─┘└──────── 314G/464G (67%) ─┘
Hint: Want your plain shell? Press C-o, and get back to MC wit
[petrstepanov@localhost ~]$                                [^]
 1Help  2Menu  3View  4Edit  5Copy  6Re~ov 7Mkdir 8De~te
```



```
    0[||||              11.8%]   Tasks: 204, 758 thr; 1 run
    1[||                 6.0%]   Load average: 1.12 0.98 0.
    2[||                 4.7%]   Uptime: 2 days, 03:23:59
    3[|||                8.6%]
  Mem[|||||||||||5.25G/15.5G]
  Swp[||||||      1.11G/4.00G]

    PID USER       PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME
 971553 petrstepa   20   0 8817M  175M  101M S 17.1  1.1  0:47.
 962203 petrstepa   20   0 2677M  175M  109M S  3.3  1.1  0:05.
   2293 petrstepa    9 -11 3257M 16520 12584 S  2.0  0.1 48:48.
   2310 petrstepa   -6   0 3257M 16520 12584 S  1.3  0.1 43:29.
 972119 petrstepa   20   0  228M  6080  4684 R  1.3  0.0  0:00.
 255036 petrstepa   20   0 4583M  185M 61504 S  0.7  1.2 40:24.
 798643 petrstepa   20   0 4125M  467M  166M S  0.7  2.9 18:08.
 962214 petrstepa   20   0 2677M  175M  109M S  0.7  1.1  0:00.
 962218 petrstepa   20   0 2677M  175M  109M S  0.7  1.1  0:00.
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice
```

Left to right: Midnight Commander file manager and htop process viewer.

Where to learn more?

- Find introductory interactive videos on YouTube: https://www.youtube.com/results?search_query=linux+terminal+tutorial
- Use `man <command>` to get help on variety of arguments for a particular command.
- Complete Command Line tutorial from Canonical: https://ubuntu.com/tutorials/command-line-for-beginners

Homework

Due date June 8.

1) Windows users: install Windows Subsystem for Linux (WSL).
   macOS users: use macOS Terminal or install install VirtualBox + Ubuntu Linux.
2) Run Command prompt. Discover root filesystem folders that were not mentioned in current lecture.
3) Compose your TODO list. Create three files each containing one line of text (TODO item). For instance, `item1.txt`, `item2.txt`, `item3.txt`.
   Expert: try using a unique approaches to create every `line?.txt` on the file system. Look up how to create a file with `touch`, `nano` and `echo`.
   Extra creative: instead of the TODO list try composing a three-line Haiku.
4) Use Command Line Control Flow to join above three files into a single file on your computer, e.g. `todo.txt`.
   Expert: install `cowsay` and pipe the concatenated file to the `cowsay -n`.
5) Privacy is the key! Modify your `todo.txt` permissions. Restrict read, write and execute for other users on the system.
6) Do some research online and find some linux terminal script or program that you think is cool (try looking on Reddit, Google, YouTube, etc). Maybe on the next lesson we will try installing some of the terminal programs you found and will give them a try.

Questions and Answers

- Slack channel: https://cua-reu-2021.slack.com
- CUA NP GitHub page: https://github.com/CUA-NP/NP_SummerStudents

Future Plan

- Creating folders, removing files and folders.
- Environment variables.
- `grep` command and regular expressions.
- Connecting to remote computers via Secure Shell (SSH)
- Running a computer simulation?