

TRICKS AND SHORTCUTS OF THE CONSOLE

- a. **Tab** - autocomplete
- b. **Up/Down arrow** - navigate through the history commands
- c. **Ctrl-C** - stop the currently running program, useful when you have an infinite loop
- d. **Ctrl-Z** - suspend the execution of the current program, use command **fg** to bring the most recently suspended job back to the foreground
- e. **Ctrl-A** - jump to the beginning of the line
- f. **Ctrl-E** - jump to the end of the line
- g. **Ctrl-B** - move backward on character
- h. **Ctrl-F** - move forward on character
- i. **Ctrl-D** - send EOF (End of File) to stdin. It tells to the terminal "I have nothing more to type". For example we can type **cat** in terminal and it will wait us to type a file. It uses stdin to read the filename and we can use Ctrl-D to tell that we finished typing.
- j. **Ctrl-R** - search through the history of commands
- k. **Ctrl-K** - cut the text from the current position to the end of the line
- l. **Ctrl-Y** - paste what was cut with Ctrl-K

COMMANDS AND PATHS

To run a command, type its name followed by whatever arguments necessary, separated by space, for example :

- a. To list the content of the current directory run **ls**.
- b. To see the content of a file run **cat [FILENAME]**, here filename is an argument.
- c. To see the content of the current directory, with all the details and including the hidden files, run **ls -l -a** or **ls -l --all**. Here **-a** and **--all** have the same effect, one being the short form, and the other being the long form.
- d. To create a directory named abc, run **mkdir abc**.
- e. To display the content of the current directory, with all the details but without the annoying colors, run **ls -l --color=never**. Here, **--color=never** is an argument with value. Sometimes, the equal sign is not necessary, but always consult the manual (command **man**) or the **--help** option (ex. **ls --help**).
- f. To do the same thing above, for the directory **/etc**, run **ls -l --color=never /etc**.

Command structure :

- a. Space is separator
- b. First word is the command
- c. Next words are arguments
 - i. Values : **ls /etc**
 - ii. Options :
 1. Short form : **ls -l**
 2. Long form : **ls --all**
 3. Short form with value : **cut -d : -f 1,2,7 /etc/passwd**
 4. Long form with value : **cut --delimiter=: --fields=1,2,7 /etc/passwd**
 5. Combined short forms : **ls -l -a** is equivalent with **ls -la**

Paths :

- a. UNIX file system has a single root, unlike Windows which has a root for every drive mounter (C:, D:, etc)
- b. The UNIX file system root is **/**, and all drives are mounted as directories, somewhere in the file system
- c. The UNIX file separator is **/**, unlike Windows, where the separator is ****
- d. Every user has a home directory, which is the current directory when you connect over SSH. Run command **pwd** to find the path to your current directory.

LINUX MANUALS

1. If you need to learn about a command or C function or many other things, you can use the built-in manual pages.
2. To read about command **ls** run the command **man ls**, to read about C function **pthread_create** run **man pthread_create**
3. You can use the following commands in the man page
 - a. Search : /
 - b. Exit : q
4. Structure of man page
 - a. Synopsis
 - b. For C functions, list of headers to be included
 - c. Arguments
 - d. Return value
5. Finding the manual page you need
 - a. **apropos ls**
 - b. **whatis ls**
6. Manual sections
 - a. **man open**
 - b. **man 2 open**
7. Some built-in bash commands do not have their own manual pages, but rather they appear in the bash manual page, run **man bash** and scroll a lot to find details about **cd**, **read**, **shift**, **fg**, **bg**, **jobs**,...

Basic commands

- a. **cat** - Reads files sequentially and outputs their content.
- b. **chmod** (-R) - Changes the permissions of a file or directory.
- c. **cp** (-r) - Copies files or directories from one location to another.
- d. **cut** (-d, -f) - Extract specific fields from a line.
- e. **echo** - Prints text or variables to the terminal.
- f. **expr** - Evaluates expressions and outputs the result.
- g. **file** - Identifies the file type of a given file.
- h. **find** (-name, -type) - Searches for files and directories.
- i. **grep** (-E, -i, -q, -v) - Searches for patterns in text.
- j. **head** (-n) - Displays the first few lines of a file.
- k. **ls** (-l) - Lists files and directories.
- l. **mkdir** (-p) - Creates a new directory.
- m. **mv** - Moves or renames files and directories.
- n. **ps** (-e, -f) - Displays information about running processes.
- o. **pwd** - Displays the current directory.
- p. **read** (-p) - Reads input from terminal.
- q. **rm** (-f, -r) - Deletes files or directories.
- r. **sed** (-E and only the commands d, s, y) - Performs basic text transformations.
- s. **sleep** - Pauses for a specified time.
- t. **sort** (-n, -r) - Sorts the lines of a file.
- u. **tail** (-n) - Displays the last few lines of a file.
- v. **test** (numerical, string and file operators) - Evaluates conditional expressions.
- w. **true** - Always returns a successful (zero) exit status.
- x. **uniq** (-c) - Omits or counts duplicate lines from sorted input.
- y. **wc** (-c, -l, -w) - Counts words, lines, and bytes in a file.
- z. **who** - Displays information about users currently logged into the system.

BASICS OF C PROGRAMMING IN THE LINUX COMMAND LINE

1. Development style
 - a. Text editor
 - b. Command line compiling
 - c. Command line program execution
 - d. Debugging (Print to console)
 - e. Detecting memory problems : valgrind
2. C language
 - a. Strings vs byte array : strings end with 0, while buffers must be accompanied by their length
(`int length = sizeof(array) / sizeof(array[0])`)
 - b. There are no references in C, only pointers (memory addresses)
 - i. **&n** is the address of variable **n**
 - ii. ***p** is the content of pointer **p**
 - c. Command line arguments : **int main(int argc, char** argv)**
 - i. **argv[0]** - name of the command
 - ii. **argv[1]** - the first argument, **argv[2]** is the second argument, and so on
 - iii. **argc** - length of array **argv**
 - d. Memory
 - i. Allocation - **malloc**
 - ii. Deallocation - **free**
3. Text files
 - a. All files are binary, but some of them contain only bytes between 0 and 127, which can be displayed as text using the **ASCII** encoding.