ICA0002: IT Infrastructure Services

Linux Terminal Basics

Roman Kuchin
Juri Hudolejev
2022

Basic terminal commands

10 terminal command you MUST know how to use

This is the minimal set you will need to continue with this course

```
cat   cd   cp   ls   man   mkdir   mv   nano*   pwd   rm
```

** Nano is probably the simplest terminal based text editor available, however, it's fine to use any other text editor you are happy with*

Basic terminal commands

Terminal command is usually followed by parameters separated by spaces
Example:

```
cd mydir
```

Some commands do not need parameters:

```
pwd
```

Others may have optional flags:

```
ls -a mydir
```

4 terminal commands for directories

`pwd` Print working directory

`ls [PATH]` List files and directories (show directory content)
`[PATH]` here means that this parameter is optional

`cd [PATH]` Change directory (go to)
Without parameters changes to user's home directory

`mkdir PATH` Make (create) directory

5 terminal commands for files

`cat [FILE]...`	Concatenate (print file content) `...` here means multiple parameters may be provided
`cp SRC DEST`	Copy file
`mv SRC DEST`	Move (rename) file
`rm FILE`	Remove (delete) file Use with care! There is no undo, and no confirmation prompt!
`nano FILE`	Edit file (open editor) There are other editors available as well: vi, emacs, ed, mcedit -- you can use whichever you like

Most important command :)

`man COMMAND` Print manual for this command

Answers 99% of your questions about this command usage
Example:

```
    man ls
```

Common alternative is **`--help`** parameter
Usually provides less info but is mostly enough for simple cases
Example:

```
    pwd --help
```

'Special' files and directories

`.`	This (current) directory
`..`	Parent directory (one level above)
`~`	This user home directory
`.foo`	'Hidden' file or directory (name starts with dot)
`*`	All files and directories in this directory (excluding hidden)

What will these commands do?

```
1.  cd  .. ;   pwd
2.  cp foo.txt bar.txt ;  cat bar.txt
3.  mv foo.txt bar.txt ;  cat foo.txt
4.  cp foo.txt  ~/bar.txt ;  rm foo.txt
5.  mkdir foo ;  cd foo ;  ls
6.  mkdir foo ;  mv foo bar ;  cd foo
7.  mkdir ~/foo ;  mv bar.txt ~/foo/ ;  ls ~/foo
8.  mkdir ../foo ;  mv bar.txt ../foo/ ;  cat ../foo/*
```

Useful tricks (1)

(Probably the most important terminal trick)

Use <TAB> key to autocomplete paths, arguments etc.

Try it out:

`ls ~/an<TAB>` (type `ls ~/an` and press <TAB>)

`cd r<TAB>` (in the directory where you have your Git repo cloned)

Useful tricks (2)

No need to change directory to view or edit the file. Don't do

```
    cd roles

    cd test_connection

    cd tasks

    cat main.yaml
```

Instead, try this command:

```
    cat roles/test_connection/tasks/main.yaml
```

Did you use `<TAB>`
for path autocompletion
in the previous example?

Try again:

```
    cat r<TAB>/t<TAB>/t<TAB>/m<TAB>
```

Useful tricks (3)

Change to the previous directory:

```
cd -
```

Change to the home directory (~):

```
cd
```

(no parameters)

How to exit the text editor?

Nano:

1. Press `<Ctrl>`+`<x>` (together) to start the exit dialog
1. Type `y` to save the changes, or type `n` to discard them
2. Press `<Enter>` to confirm the file name to write if asked so

Vi and alike:

1. Press `<Esc>` to exit the edit mode
2. To save changes (if any) -- type `:wq`
3. To exit without saving changes -- type `:q!`

Questions?