Sistemas Informáticos (Computer Systems)  
Unit 05. Activities 03

short line

Authors: Sergi García, Alfredo Oltra

Updated October 2022

Unit 05. Activities 03

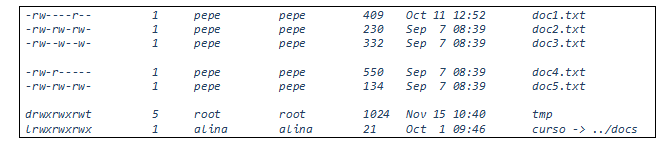
# Exercise 01

Do those exercises using “*touch*”, “*cat*”, “*cd*”, “*ls*”, “*mkdir*”, “*cp*”, “*mv*”, “*rmdir*”, “*rm*”, “*grep*”.

* Write a command to create a new file called “*names.txt*”.
* Write a command to view the content of “*names.txt*”.
* Write a command to view the content of your home directory in long format (permissions, size, date, etc.).
* Write a command to view the content of your current directory in long format, showing hidden files/directories (permissions, size, date, etc.).
* Write a command to list all files that end with “*.png*” and starts with “*ga*”.
* Write a command to store the result of a ls command in a file called “*myLS.txt*”, deleting existing content.
* Write a command to store the result of a ls command in a file called “*myLS.txt*”, adding the result to the end.
* Write a command to create a directory called “*Exercise1*” in your home.
* Write a command to move all files that starts with a from your home to directory, “*Exercise1*.
* Write a command to change name of directory “*Exercise1*” to “*Ex1*”.
* Write a command to show lines of “*/etc/passwd*” that contains word “*root*”.
* Delete all elements created.

# Exercise 02

We have obtained this result running “*ls -l*” command.



* **In symbolic mode:** add execution permission to owner of “*doc1.txt*”.
* **In symbolic mode:** delete write permission to group and others of “*doc2.txt*”.
* **In octal mode**: add execution permission to group of “*doc4.txt*”.
* **In octal mode**: delete write permission to group and read and write permissions for others of file “*doc5.txt*”.
* Write a command to change owner to “*Eulogio*” and group to “*Eulogio*” of all files of the directory.

# Exercise 03

1. Create user “*pepito*” in command line.

2. Create group “*tic*” in command line.

3. Change primary group of user “*pepito*” to “*tic*”.

# Exercise 04

Solves those exercises using “*grep*” command.

💬 **Interesting:** you can chain “grep” commands using “|” redirector.

* Show all lines of file “*list.txt*” that contain text “*lib*”.
* Show how many lines contain “*mp3*” in “*list.txt*”.
* Show files inside “/etc/ directory that contain “*host*” string inside.
* Show all lines of file “*list.txt*” that not contains letter “*a*” (uppercase or lowercase).
* Show all lines of file “*list.txt*” that not contains “*a*” (uppercase or lowercase) and contains “*m*” (lowercase).

💬 **Tip:**  “|” is a tool to create a redirection, that is, to use the output of a command as input of another command. For example: “*cat file.txt | sort*” . This command consists of two commands joined by “|”. The output of the “*cat*” command is passed as an entry of the sort command, so the final result you will see is the file “*file.txt*” sorted.

# Exercise 05

* Create a folder called “*shared*” in your home where everybody has all permissions.
* Create groups “*office1*” and “*office2*”.
* Create users “*pedro*” and “*pablo*”. Those users have to be members of group “*office1*”.
* Create users “*alba*” and “*nerea*”. Those users have to be members of group “*office2*”.
* As “*pedro*” create a file “*topsecret.txt*” that only “*pedro*” can read and write.
* As “*pedro*” create a file “*sales.txt*” that owner and group “*office1*” can read and write. Check as “*pablo*” if you can do those operations.
* As “*alba*” create a file “*employ.txt*” that every user can read and group “*office2*” can read and write. Check if it is right with “*pedro*” and “*nerea*”.

# Exercise 06

Questions about permissions. Try to answer and reason them:

* **Question 01**: if a user has read permission to a file, but that file is inside a directory that our user doesn’t have execution permission and our user have read permission. Could it read the file?
* **Question 02**: if a user has read permission to a file, but that file is inside a directory that our user doesn’t have read permission and our user have execution permission. Could it read the file?

# Exercise 06

Using bit SetUid and supposing that temporally (something like 1 hour) you have access to a machine as root and in that machine you have permanently access to a user called “*alumno*” without sudoer permissions:

* **Question 01:** How can we use bit SetUid bit to create a backdoor? (**Clue**: file “*/bin/sh*” could be useful).
* **Question 02:** How can we detect that kind of backdoors on our system? What kind of measures can we take to be safe against this kind of attack?