UNIVERSIDAD Đ CÓRDOBA

# Standardizing CLIs

**Elena Álvarez Sánchez    Jesús Izquierdo Estévez    Rafael Egea Jurado**

UCO    Informatics Engineering    Legislation and Standardization

## Introduction

Our main concern is to unify commands line interface's instructions so people should not to worry about learning any specific cli's language.

As we all know, commands in UNIX-like operating systems are different than Windows. Even in different Linux distribution some instructions changes too. For instance, yum or apt. It may be a little bit confusing for users. Due to that, we aim at changing it in order to make it easier for users.

## Instructions comparison in differents OS

- Screen cleaning
  - cls (Windows) vs clear (Linux)
- Directory content display
  - dir (Windows) vs ls or dir (Linux)
- Package management
  - yum (CentOS) vs apt (Debian) vs homebrew (MacOS)
  - dpkg (Debian) vs rpm (Red Hat)
- Elements removal
  - del (Windows) vs rm (Linux)
- Elements copy
  - copy (Windows) vs cp (Linux)
- Elements movement
  - move (Windows) vs mv (Linux)

## Advantages

- Easy to use
- Easy to learn
- Intuitive
- Possibility to work on any operating system
- Suitable for any user

## Definitions

- Cli´s language: command-line interface is a method that allows users to give instructions to a computer program through a simple line of text.
- Cli: According to (What is CLI, 2010) command-line interface is a command line program that accepts text input to execute operating system functions.

## Object and scope

This standard's target is to establish an unique command line interface's language thereby every single OS has the exactly same instructions.

## Method

In order to face every functionality required, we are going to collaborate with the currents providers. We will work with them so it would be obtained the perfect command line interface via these phases.

- In a first approach, we would include instructions for managing the operation system at the lowest level of abstraction. Such as directory listing, memory management, creation, movement, copy and deletion of documents and directories.
- In a second phase, we need to enlarge the previous phase so that our user is able to work easily offline. Package management need to be perfectly provided.
- In a third phase, interfaces for download, connect and interact with remote servers via ssh, ftp, telnet... must be perfectly functional and highly user-friendly.
- Finally, it should be realized certain user surveys for realising missing commands and provide them.

Through this phases we want to obtain a fully functional command line interface that through the years will evolve.

## Conclusion

Our motivation is bringing harmony to this computerized world. Facilitating students, it workers and every kind of user the translation between OSs. Allowing them to choose the best of each world.

It could be thought that the implementation of an interface as functional as the current ones would take a large amount of time. However, it must be take into account that it would be done in collaboration with the current ones so it would take the more complete and meaningful functionalities.

## References

- Unix First Edition Manuals. (1990). Unix Manual Page archive. http://man.cat-v.org/unix-1st/
- What is CLI. (2010b). M3schools. https://www.w3schools.com/whatis/whatiscli.asp