Command Line Interfaces and Software Development  
Research  
What is a CLI?

Mason Adrales | mta49n | INFOTC 3530: UNIX Operating Systems

University of Missouri

Topic

The topic that I have chosen for this report is the Command Line Interface and how it can impact and influence Software Development. I chose this topic because I am planning to go into a career in Web Development/Software Development and I wanted to increase my knowledge of using and developing with CLI’s. Command Line Interfaces are now standard for most frameworks or technologies in order to streamline the process of installing and exhibiting the features of the application. In this report, I will cover the basic definition of a CLI along with the basic history of the use of CLI in software development. Then, I will use examples such as Angular, PHP, and Git to demonstrate how beneficial a CLI can be. Lastly, I will display applications of my research on my local machine to demonstrate how each CLI can be accessed and show some of the basic commands for the mentioned frameworks.

Command Line Interfaces and Software Development  
Research  
What is a CLI?

According to Margaret Rouse of SearchWindowsServer, a command line interface is a, “text-based user interface used to view and manage computer files”. In a more detailed explanation, a command line interface allows users to interact with the programs installed on their machines to access features, generate new items, or even to stop processes on the system among a plethora of other options. Although Windows, MacOS, and Unix-based systems all have access to a command line interface, it is most easily observed in the MacOS and Unix systems on the terminal. The shell in a system takes the command executed in the CLI and processes the actions that a command is meant to do. In UNIX systems, the CLI is offered as either graphical or non-graphical. This determines if there is a user interface that the program offers to show a visual representation of the commands that are being executed. A good metaphor for the applications of a CLI is to imagine the CLI as a waiter at a restaurant. The customer (active user) tells the waiter (the CLI) what they want (the actions they want to be done) and the waiter takes the requests to the kitchen (the shell) and returns back the food (response from shell) to the customer. Overall, the command line interface helps to navigate, edit, and execute actions and commands in Windows, MacOS, or UNIX-based systems.

Basic Uses of the Default CLI in UNIX Systems

In Software Development, there are a plethora of commands that are useful with the default command line interface in a UNIX system. Some basic commands, such as cd (change directory), vi/vim (text editor), and chmod (changing permissions of a file or directory) are common to ensure that users are editing the correct file, saving the changes, and having the permissions to do the actions they wish in the environment. In addition to these basic commands, software developers can also create shell scripts that run in the command line with no additional plugins or installations. Shell scripting can be a great way to automate tedious tasks in a UNIX environment and developers can use that to create basic programs that present a GUI to the user, all done via the CLI. Although the default CLI can provide some useful features, third-party CLIs really begin to expand the horizons in terms of what you can do from the terminal.

Other CLIs

When talking about CLIs that are not pre-installed on UNIX systems, Angular CLI is one that is very beneficial to software developers. Angular is a JavaScript framework that allows for single page applications to be created and developed. Angular has its own CLI title Angular CLI and in this command line interface, there are new commands that are added. First, the Angular CLI must be installed on the default CLI with the command *npm install -g @angular/cli*. Then, you can use *ng* commands that allow you to create a new application, serve/host the application, and even build it for production. The Angular CLI not only adds new commands that interact with Angular applications, but it also can help build the application for deployment in another environment. Angular is such a powerful software development tool. Its inception in 2010 as AngularJS proved to be a progressive movement towards other JS frameworks (with their own CLI’s) such as React and Vue. Whether you are looking to create a client facing website or just trying to fumble through a tutorial, JavaScript frameworks prepare you for the work ahead by supplying a beginner friendly CLI.

Another example of CLI that can be examined are the use of PHP commands. You can use *php text.php* to compile a PHP file named text.php on the CLI without the need of another development environment. This can speed up development as you can debug tricky php files right on the CLI instead of needing to find a third party application that will debug your code for you. Lastly, the Git CLI is immensely useful in Software Development as Git allows users to work collaboratively on the same project while tracking the version and edit history of each and every file. I personally have used Git in a multitude of classes and in a plethora of projects and I can say that it is my favorite CLI I have used so far. You can initialize a Git repository (git init), add files to the repository (git add), commit your changes (git commit), and push them up to the remote repository (git push) among other commands. Overall, command line interfaces allow users to modify, access, and create new files and projects in an expedient manner.

Applications

Example 1: Basic UNIX CLI Commands

A picture containing schematic

Description automatically generated

In this photo, I execute the pwd (Present Working Directory) command to show the directory I am currently working in. Then, I use the cd command (Change Directory) to change to the Desktop folder. Then, I execute pwd once more to show the change. Lastly, I use the ls -al command to show all of the files in the current directory.

Example 2a: Angular CLI Graphical user interface, text, application, chat or text message

Description automatically generated

In this screenshot, I have opened up a project that I have developed to be used with Angular. Using the command ng serve on the Angular CLI, the backend code transposes my project into a live browser version that can be accessed via <http://localhost:4200/> as shown in the screenshot.

Example 2b: Angular CLI cont. Graphical user interface

Description automatically generated

In this screenshot, I have demonstrated what the ng serve commands deployment actually looks like. This website was created in another one of my classes but using the Angular CLI on VSCode yields this return. This shows how powerful Angular CLI can be.

Example 3: Git CLI

Text

Description automatically generated

In this screenshot, I have already been using git on this project. I use the git status command to see the status of my branch. I then use git add to add the file to the staging branch. Next, I use a git commit to commit my changes and then a git push to push my changes to the remote repository. This can show how git CLI can really change development and make it simple and easy!

References

Command Line Interfaces and a Brief History of User Interfaces: <http://www.catb.org/esr/writings/taouu/html/ch02s02.html#:~:text=The%20concept%20of%20timesharing%20dates,of%20what%20came%20after%20it>.

Command Line Interfaces from SearchWindowsServer: <https://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI#:~:text=A%20command%20line%20interface%20(CLI,interfaces%20and%20character%20user%20interfaces>.

Documentation on Command Line Interfaces: <https://en.wikipedia.org/wiki/Command-line_interface>

Documentation on Git: <https://git-scm.com/>

Documentation on Angular CLI: <https://cli.angular.io/>