Unidad 2 — Bash y Git

Contenidos

Esta unidad es una introducción a dos herramientas básicas para los desarrolladores, muy utilizadas en el ámbito del desarrollo web: una interfaz de línea de comandos (usando generalmente Bash) y Git.

CLI

La interfaz de línea de comandos o *command line interface* (CLI) es una forma de interactuar con el sistema operativo. Los usuarios de una computadora hoy en día usan exclusivamente una interfaz gráfica o GUI. La línea de comandos muchas veces se asocia con el pasado, con una forma de usar la computadora cuando no existía nada mejor. Esto no es cierto y como programadores tenemos que estar familiarizados con la línea de comandos para realizar muchas tareas. Además la línea de comandos es más expresiva que una interfaz gráfica, muchas tareas sólo pueden realizarse allí, o bien de manera más simple.

Shell

La shell es un programa del sistema operativo que interpreta los comandos que escribimos y los ejecuta. Toma su nombre de que es la capa más externa del sistema operativo, con la cuál interactuamos como usuarios. En los sistemas operativos derivados de UNIX (Linux y MacOS) usamos generalmente un programa llamado `bash`.

Emuladores de terminal

Si nos encontramos usando un entorno de escritorio, en Linux por ejemplo KDE o GNOME, no usamos directamente `bash` sino que nos comunicamos con el programa a través de un emulador de

terminal. Este programa que suele aparecer simplemente como terminal en el menú de aplicaciones puede cambiarse a gusto del usuario. En GNOME por ejemplo es `gnome-terminal` y en KDE `konsole`.

El prompt

Cuando abrimos una terminal usualmente nos encontramos con algo similar a esto.

```
[user@host ~]$
```

Esto se conoce como *prompt* y aparece cuando la terminal está dispuesta a recibir comandos. Además puede personalizarse para mostrar información útil, en el caso de arriba está indicando el nombre de usuario, de máquina y el directorio actual. Si en vez de ver un signo `$` vemos un `#` significa que la sesión de esa terminal tiene privilegios de superusuario.

Comandos

Los comandos en `bash` y en otras *shells* también tienen la siguiente estructura.

```
$ comando -opciones argumentos
```

El comando es simplemente el nombre del comando, como `cd`. Las opciones están precedidas por un guión o dos guiones si se usa la versión no abreviada. Las opciones también se conocen como *flags*. Por último los argumentos dependen del comando en cuestión y muchas veces se pueden aceptar varios. Por ejemplo el comando `ls` que lista los contenidos de uno o más directorios.

```
$ ls -a /usr/bin /dev
```

En el ejemplo anterior listamos los contenidos de dos directorios o carpetas, `/usr/bin` que contiene programas instalados en Linux y `/dev` que contiene archivos que representan dispositivos de hardware entre otras cosas. Además se usa la opción `-a` que incluye en el resultado los archivos y carpetas ocultas. Usando la versión larga del *flag* `-a` y listando los contenidos del escritorio del usuario se vería así.

```
$ ls --all /home/user/Desktop
```

Cabe destacar que los espacios en los ejemplos anteriores son importantes y se utilizan para delimitar el comando de las opciones y los argumentos.

Ayuda

En los sistemas derivados de UNIX cada comando debe tener una página de manual o *man page*. Esto es un estándar y una tradición en estos sistemas operativos. Para invocar la ayuda de

cualquier comando simplemente usamos `man comando` que nos imprime la *man page* del comando que nos interesa. Por ejemplo para ver la ayuda de `ls`

```
$ man ls
```

Navegación

Veamos unos comandos básicos de navegación para moverse por la estructura de archivos o *filesystem*.

`pwd`
: *Print working directory*. Imprime el directorio actual donde estamos parados.

`cd`
: *Change directory*. Cambia de directorio al especificado como argumento.

`ls`
: *List directory*. Lista los contenidos de uno o más directorios.

Manipulación de archivos y directorios

Los comandos básicos para borrar, crear, mover y copiar archivos y directorios.

`cp`
: *Copy*. Copia archivos y directorios.

`mv`
: *Move*. Mueve archivos y directorios, sirve también para renombrar.

`rm`
: *Remove*. Elimina archivos y directorios.

`mkdir`
: *Make directory*. Crea un directorio (carpeta).

`touch`
: Sirve para crear archivos.

Git

Git en la línea de comandos

Ejercicios