

Resumen de clase

03

Emilio López Sotelo

9/01/2019

La tercera clase comenzó con la introducción a nuevos comandos y a un argumento nuevo para el comando "ls" así como también el uso de ">" en bash. Los nuevos comandos que se vieron fueron:

- *cat*, el cual tiene como función mostrar en la pantalla del interprete el contenido del archivo.
- *mkdir*, el cual crea un directorio nuevo en la ruta actual. Si se quiere crear un directorio y la ruta completa a ese nuevo directorio no existe se debe usar el argumento "-p" para crear los directorios faltantes, por ejemplo: si la ruta actual es home/Documents/ y la carpeta Documents/ esta vacía pero se quiere crear el directorio home/Documents/a/b/c/d/ se debe escribir el comando "mkdir -p /a/b/c/d/" para poder crear la carpeta d/.
- *rmdir*, el cual elimina un directorio vacío.

El nuevo argumento para el comando "ls" fue "-la" el cual sirve para enlistar los archivos y directorios ocultos dentro de la ruta en donde se esta. El caracter ">" tiene como propósito guardar el resultado de un comando dentro de un archivo de texto plano.

Después de esto aprendimos una forma de evitar el tener que salir de bash para escribir un comentario al momento de ejecutar el comando "git commit", simplemente hay que ejecutar el comando con el argumento "-m" seguido del comentario, por ejemplo:

- `git commit -m "cambios al archivo abc.txt"`

También tuvimos nuestro primer encuentro con los editores de texto integrados a las distribuciones de Linux que son "vi" y "gedit". Para comenzar a usarlos basta con escribirlos en bash como si fueran un comando. Hacer esto crea un archivo de texto plano .txt nuevo en la ruta actual, si se quiere editar un archivo ya existente se debe escribir el nombre del editor de texto seguido de la ruta del archivo desde la ruta actual. Por ejemplo, si uno se encuentra en home/ y quisiera editar el archivo abc.txt que se encuentra en la ruta home/Documents/abc.txt usando gedit uno tendría que escribir el comando "gedit /Documents/abc.txt". En cuanto a los editores "gedit" cuenta con una interfaz grafica similar a la de Wordpad en Windows, es decir que contiene botones para abrir opciones del archivo como podrían ser guardar, abrir, etc. Por otra parte "vi" es un interprete de comandos, esto quiere decir que para empezar a escribir o guardar un archivo y salir hay que presionar una combinación de caracteres en el teclado. Algunos de estos comandos de vi son:

- *i*, para empezar a escribir.
- *Esc*, para salir del modo de edición.
- *w + Intro*, para guardar el archivo, si es la primera vez que se guarda forzosamente hay que escribir el nombre del archivo seguido de haber presionado "w" y entonces presionar "Intro".
- *q + Intro*, para salir de vi, si uno no ha guardado su archivo "vi" no lo dejara salir y en este caso se debe presionar "q! + Intro" para salir sin guardar.

Finalmente se termino la clase con una discusión sobre como abordar y resolver un problema ademas de una pequeña primera introducción a Python. La discusión tuvo como objetivo el saber identificar un problema y los pasos a seguir para poder resolverlo de manera satisfactoria. La siguiente enumeración funciona entonces como un esquema general para resolver un problema con dentro de la computación:

1. Definir el problema, es decir que se quiere resolver.

2. Delimitar el problema, es decir analizar como se puede resolver y que herramientas se encuentran disponibles para resolverlo.
3. Buscar soluciones ya existentes para el mismo problema o en su caso problemas similares.
4. Realizar soluciones para casos específicos y describir detalladamente como se llego a esas soluciones.
5. Tratar de encontrar una solución general que sea aplicable en todos los distintos casos del mismo problema.

Después de este esquema se planteo un problema especifico a resolver, inspirado en los temas vistos en las clases de física de la preparatoria. El problema que se escogió fue el de encontrar la posición de un objeto como función del tiempo dentro del escenario del tiro parabólico, este fenómeno esta dado por la ecuación:

$$y(t) = V_0 - 1/2gt^2$$

donde " $y(t)$ " es la posición como función del tiempo, " V_0 " es la velocidad inicial del proyectil, " g " es la aceleración a causa de la gravedad y " t " es el tiempo de vuelo del proyectil

Es importante anotar que aunque la función esta definida para todo el conjunto \mathbb{R} en un contexto de física solo nos interesan los valores donde $t \geq 0$ ya que no nos interesan tiempo negativos, es decir lo que ocurre antes de que el proyectil entre en movimiento. Podemos entonces acotar nuestro nuevo dominio como $t \in [0, 2V_0/g]$. Tampoco nos interesan los valores donde $y < 0$ ya que consideramos que en esas posiciones nosotros como observadores ya no tenemos visibilidad sobre el proyectil. Estas dos anotaciones son ejemplos de como se puede delimitar el problema.

Pasando ya a la parte de Python, la gran mayoría de las distribuciones de Linux cuentan con alguna versión de Python ya instalada. En mi caso Ubuntu venia con la versión mas reciente de Python ,que es la 3.6, incluida, sin embargo se llego al consenso dentro de la clase de utilizar la versión 2.7 de Python ya que esta versión lleva mas tiempo en operación y por lo tanto contiene menos errores ya que ha sido probada muchas mas veces.

También es importante destacar que para nuestro trabajo durante este curso no hay ninguna diferencia entre ocupar la versión 3.6 o 2.7 por lo que se prefiere consistencia antes de nuevas características. Además nos pidieron que instaláramos "Idle" el cual es un IDE o un entorno integrado de desarrollo (Interactive Development Environment). Un IDE es a grandes rasgos un programa hecho para escribir y ejecutar código dentro de el mismo, esto quiere decir que por lo general contiene un compilador, formatos de texto específicos para el lenguaje en que se esta escribiendo como por ejemplo auto indentación o el uso de colores para palabras reservadas, comprobación de errores y demás herramientas. En nuestro caso específico "Idle" es un IDE para Python y esto hace que sea mucho mas fácil trabajar con Python comparado con ocupar un editor de texto como "gedit" o trabajar desde el shell de Python que se puede abrir en bash. De hecho una de las herramientas mas poderosas de "Idle" es que incluye su propio shell de Python lo cual hace mucho mas facil y rapido el correr código, modificarlo o hacer pruebas dentro del mismo shell. A continuación van los comandos utilizados para hacer ambas instalaciones:

- *sudo apt install python2.7*
- *sudo apt install idle-python2.7*