

Clase 08

Diego Armando Santillán Arriaga

16/01/19

1 Comandos útiles de Linux

1.1 Ejecución en primer y segundo plano

En Linux generalmente se colocan los 'procesos en **primer plano**, es decir, se escribe un comando y el proceso se ejecuta. También pueden haber procesos en **segundo plano**. Para ello estando en la terminal pauso la ejecución con **control+Z** y después para ejecutarlo en segundo plano se introduce el comando **bg**. El signo **ampersand** puesto a continuación del comando de activación de un proceso lo ejecuta en segundo plano. Para detener definitivamente (terminar) un proceso se presiona **control+c**. Si interrumpí un proceso con **control+z**, se continua con el comando **fg**.

1.2 Comando top

Comando **top + 1**: muestra los cores. En la parte de PID en el extremo izquierdo del menú de top está el código correspondiente a cada proceso. Otra forma de terminar un proceso es **kill -9 "código del proceso"**.

1.3 Bash

Bash es un intérprete de comandos: tiene un conjunto de instrucciones internas (comandos bash). Está a la espera de una instrucción para ejecutarse. Python también es un intérprete de comandos solo que con otras instrucciones.

find . -name "*.py"where Comando para buscar los archivos que terminan en .py Cuando se introduce una instrucción en el bash, intenta ejecutarla. **#!/usr/bin/python2.7** Si se coloca al principio de un programa como comentario(sin espacio entre el signo de gato y el signo de exclamación) se habilita a bash desplegar su contenido en la terminal. Este código nos indica la versión en la que fue hecho el archivo (la ejecución de un programa es exclusiva de la versión de python en la que se hizo).

2 Comandos útiles de python

En python el **;** se utiliza para indicar que finalizó una instrucción. Hay otra manera de escribir variables dentro de una cadena de texto. Esto se hace reemplazando las variables por **la"letra o nombre de la variable**

en mayúscula”:”formato del texto de acuerdo a los códigos del comando de variables visto anteriormente (.2f para decimales, e para exponencial, etc)”. De una biblioteca se pueden importar funciones utilizando: from ”biblioteca” import ”función”. Con el comando input asignado a una variable (entre paréntesis se puede colocar una cadena que indique lo que se debe de poner) el usuario le puede asignar un valor cualquiera a una variable.

3 Clase, objeto, atributo y método

Clase: un conjunto de objetos que comparte características generales. Cada miembro de la clase es un objeto. Cada objeto es distinto. Puede haber interacción entre objetos de distintas clases. Cada objteo tiene sus atributos, tiene asociado las tareas que puede realizar y la manera en que interactúa con otros objetos. Un método son las acciones que realiza un objeto o que se pueden aplicar a un objeto. Hay métodos en los que dos o más objetos interactúan (un objeto va a tener características y acciones). Hay métodos que modifican el estado del objeto.

1. Clases
2. Objetos (uno específico de la clase)
3. Atributos
4. Métodos

Notación

”objeto”.”método”(parámetro1, parámetro2, ...) (no siempre son necesarios los parámetros)

Un método siempre está asociado a un objeto ejemplo: 'x, y, z' objeto de la clase cadenas de texto. Como es un objeto, tiene métodos por lo que se puede escribir 'x, y, z'.”método”

Ejemplo: x es un objeto de la clase números por lo que se puede escribir x.”método”

Si no hay un formato de "objeto"."método" entonces se tiene una función.
Una función no depende un objeto, un método sí.