

BASH y la Terminal Linux

BASH

- Lenguaje con el que se interpretan los comandos en la terminal de Ubuntu.
 - Hay otros lenguajes, pero éste es probablemente el más utilizado.
- Se pueden automatizar ciertos comandos de la línea de comando con este lenguaje.

Características

- No es compilado.
 - Es interpretado directamente por la terminal.
- Sólo requiere un archivo de texto que comience con la línea:

```
#!/bin/bash
```

- Y se corre como:
bash archivo_de_texto

Ejemplo Básico

```
#!/bin/bash
```

```
ls
```

Ejemplo Básico

- Guardar ese texto en un archivo llamado “ejemplo”, y luego hacer:
 bash ejemplo
- Se corre el comando “ls” de Linux.

¿Para qué?

- Se pueden hacer varios trucos para facilitarles la vida más adelante.
- Por ejemplo, ¿cómo le harían para correr dos programas al mismo tiempo?
 - Como un localizador de fuentes sonoras y un separador de fuentes sonoras.

Ejemplo: Ping

```
#!/bin/bash
```

```
ping -c 5 google.com
```

```
ls
```

Ejemplo: Ping

- El comando “ping” prueba la conexión a un servidor (en este caso google.com).
- El argument -c le dice cuántos intentos hacer (en este caso 5).
 - Si no se le da este argumento, se queda haciéndolo hasta que se le CTRL+C.

Ejemplo: Ping

- Al correr el ejemplo, se puede observar que primero se hace “ping” y hasta el final se hace “ls”.
- Si quisieramos correr “ping” y “ls” al mismo tiempo, podemos forzar que “ping” se corre en el “fondo” haciendo:

```
ping -c 5 google.com &
```

Ejemplo: Ping

```
#!/bin/bash
```

```
ping -c 5 google.com &
```

```
ls
```

Ejemplo: Ping

- Al correrlo ahora parecería que se corrió primero “ls”, pero realmente se están corriendo los dos en paralelo.
 - Nada más que “ls” termina muchísimo antes.
- Podemos agregar algunos retrazos para que se vea más apropiado.

Ejemplo: Sleep

```
#!/bin/bash
```

```
ping -c 5 google.com &
```

```
ls
```

```
sleep 2
```

```
ls
```

```
sleep 2
```

```
ls
```

Ejemplo: Sleep

- El comando “sleep” hace que la terminal se “duerma” por una cantidad entera de segundos.
 - Le hace un *floor* al argumento entrante.

Ejemplo: Sleep

- Ahora se puede observar como el comando “ping” y los diversos “ls” con sus “sleep” intermedios corren en paralelo.

Ejemplo: Ping Infinito

```
#!/bin/bash
```

```
ping google.com &
```

```
ls
```

```
sleep 2
```

```
ls
```

```
sleep 2
```

```
ls
```

Ejemplo: Ping Infinito

- Si un comando está en el fondo, no se puede matar con CTRL+C.
 - Esto sólo mata al comando que está “enfrente”.
- Nuevos amigos:
 - El comando “ps”
 - El comando “kill”

PS

- “ps” imprime en pantalla los procesos que están corriendo actualmente. Posibles argumentos:
 - “u” con formato orientado a un humano
 - “x” procesos que estén corriendo en alguna terminal
 - “a” de todos los usuarios (no es necesario por ahora)
- En otra terminal correr:
ps ux

PS

- Cada renglón representa un proceso activo.
- Las columnas muestran, en orden:
 - Usuario
 - Identificador de proceso (PID)
 - Porcentaje de CPU que ocupa
 - Porcentaje de memoria que ocupa
 - Memoria virtual ocupada
 - Memoria real ocupada
 - Terminal en la que está corriendo
 - Estado del proceso
 - Comando que se corrió.

¿Cómo encuentro mi proceso?

- Otro amigo: grep
 - Encuentra expresiones regulares en el texto que se le pasa.
- Ejemplo:
 - grep sleep ejemplo
 - Imprime las líneas del archivo “ejemplo” que contienen la palabra “sleep”.

¿Cómo encuentro mi proceso?

- Correr:

```
ps ux | grep ping
```

- El “|” es el caracter *pipe* y se encuentra al lado del tecla del número 1 en teclados latinoamericanos.
 - Le dice a Bash que la salida del primer comando “ps ux” se pase como entrada al segundo comando “grep”.
- Este comando entregará las líneas impresas por “ps ux” que contenga la palabra “ping”.

¿Cómo encuentro mi proceso?

- Se puede ignorar la línea que presenta al proceso “grep”, porque ese es el que acabamos de correr.
- La otra línea será el proceso que queremos matar, con todo y su PID.

¿Cómo mato a mi proceso?

- Correr:
kill PID
 - reemplazando PID por el número que otorgó el comando “grep”
- Para asegurar que murió, correr de nuevo “ps ux | grep ping” y ya no debería estar nuestro proceso en la lista impresa.

¿Cómo mato mi proceso?

Versión Master

- Otro amigo: killall
 - Parecido a “kill” pero recibe el nombre del comando.
 - Todo comando que tenga ese nombre será matado.
- Volver a correr “bash ejemplo” y luego correr:
killall ping
- Para asegurar que murió, correr “ps ux | grep ping” y ya no debería estar nuestro proceso en la lista impresa.

Mayor Documentación

- Hay muchísimo más que se puede ver con Bash y la terminal de Linux, pero esto es suficiente para lo que pueden utilizar.
- Para mayor información:

<https://www.gnu.org/software/bash/manual/bash.html>