

# Comandos Linux

## Comandos de Linux

### Procesamiento de Datos

Los datos son ingresados en un proceso y esto emite una información a la salida.

Canales por donde ingresan los datos a un proceso y por donde sale la información son conocidos como FLUJOS.

|                        |  |
|------------------------|--|
| Entrada Estándar       | Por defecto: teclado. Puede suceder que nosotros queramos ingresar información al proceso a través de un archivo. Para esto, realizamos una REDIRECCION. |
| <u>Salida Estándar</u> | Por defecto: Pantalla  |
| <u>Error Estándar</u>  |  |

## Comandos

|   |  |
|---|--|
| \$ mysql -h ipservidor -u usuario -pcontraseña < nombearchivo.sql | Ejecuta todos los comandos dentro del archivo sql en el servidor especificado.                                     |
| \$ <u>ls &gt; archivos.txt</u>                                    | Envía los datos mostrados hacia una salida que puede ser un archivo especificado                                   |
| \$ <u>ls -l   more</u>  | Muestra un resultado largo en varias páginas   |
| \$ <u>cat nombearchivo   wc -l</u>                                | El comando wc cuenta la cantidad de caracteres en un archivo, junto a -l permite identificar la cantidad de líneas |

### Administración de procesos en background y foreground

La ejecución de los procesos normalmente es rápida. Sin embargo, cuando el proceso se está ejecutando la terminal no permite hacer nada más que ver el cursor titilando.

Este modelo de ejecución es **EJECUCIÓN EN PRIMER PLANO**.

## Comandos

|  |  |
|--|--|
| \$ nombre_comando &  | Permite realizar la ejecución en segundo plano   |
| <u>Control + Z</u>   | Manda un comando al background   |
| <u>\$ fg</u>   | Trae los comandos a foreground   |
| <u>\$ ps</u>   | Muestra los procesos en ejecución  |
| <u>\$ ps ax</u>  | Muestra todos los procesos del sistema   |
| <u>\$ ps ax   grep palabra</u>   | Muestra los comandos que tengan la palabra dentro del comando en ejecución   |
| <u>\$ top</u>  | Muestra como los procesos van cambiando y como está la computadora.  |
| <u>\$ kill -</u><br><u>numero_prioridad</u><br><u>numero_proceso</u>       | Sirve para terminar un proceso en ejecución. Tiene un nivel de prioridad yendo del 9 al 0 siendo 9 el de mayor rango |
| <u>\$ killall -</u><br><u>numero_prioridad</u><br><u>nombre_ejecutable</u> | Tiene un nivel de prioridad yendo del 9 al 0 siendo 9 el de mayor rango  |

Los *demons* son procesos ejecutándose en background constantemente. Se ejecutan siempre para que el sistema tenga sentido.

Permisos sobre archivos: El sistema de permisos octal

Todos los permisos deben ser dados por otro.

Los archivos de Unix tienen:

|  |   |
|--|---|
| Dueño Asociado                             | Persona que ha creado el archivo                                |
| <u>Grupo de Usuarios</u>                   | Personas que pueden acceder al archivo                          |
| <u>Permisos a otros</u><br><u>usuarios</u> | Es para personas que no son ni el dueño ni asociados al archivo |

Permisos

|                 |
|-----------------|
| Leer            |
| <u>Escribir</u> |
| <u>Ejecutar</u> |

## Matriz de operaciones

| Property     | Lectura | Escritura | Ejecución |
|--------------|---------|-----------|-----------|
| <u>Dueño</u> | x       | x         | x         |
| <u>Grupo</u> | x       |           |           |
| <u>Otros</u> |         |           |           |

## Comandos

|  |  |
|--|--|
| <u>\$ ls -l</u>  | Nos muestra una visión completa sobre los permisos-rw-rw-rwx r: Read   w: Write   x: EjecuciónAdministradorGrupoOtro |
| <u>Alterar permisos de archivos asociados</u>          |  |
| <u>\$ chmod o-w nombre_archivo.txt</u>                 | Cambia individualmente los permisos del archivo.O: Para otros usuariosW: Quitar permiso de escritura                 |
| <u>\$ chmod +x nombre_archivo.txt</u>                  | Agrega permiso de ejecución en los permisos del archivo  |
| <u>\$ chmod 760 nombre_archivo.txt</u>                 |  |
| <u>\$ sudo chown nombre_usuario nombre_archivo.txt</u> | Cambia el propietario del archivo  |
| <u>\$sudo chgrp nombre_grupo nombre_archivo.txt</u>    | Cambia el grupo de un archivo hacia otro.  |

## Sistemas de manejo de paquetes

Paquetes de Software: Permiten realizar la instalación de diversas aplicaciones mediante línea de comandos.

Administradores de Paquetes: Conocen de donde realizar las descargas, que otros paquetes tenemos instalados y como realizar la instalación para que no existan conflictos

## Comandos

| Column 1                          | Column 2                                       |
|-----------------------------------|--|
| <u>\$ apt install nombre_rama</u> | Permite realizar la instalación desde una rama |

## Herramientas de compresión y combinación de archivos

Al momento de realizar el envío de archivos, vamos a querer realizar la disminución de su tamaño.

### Comandos

|   |   |
|---|---|
| <u>\$ ls nombre_archivo.txt -lh</u>     | Revisa el peso de un archivo  |
| <u>\$ gzip nombre_archivo.sql</u>       | Zipa el archivo para que pueda pesar menos. Pasará a ser un archivo .gizp |
| <u>\$ gzip -d nombre_archivo.sql.gz</u> | Descomprime el archivo  |
| <u>\$ tar cf nombre.tar ruta/*</u>      | Comprime en nombre.tar todos los archivos que contengan la ruta           |
| <u>\$ tar tf nombre.tar</u>             | Verifica el contenido del archivo nombre.tar                              |

## Herramientas de búsqueda de archivos

### Comandos

|                   |   |
|-------------------|---|
| <u>\$ locate</u>  | Herramienta que permite hacer una búsqueda en todo el sistema de archivos   |
| <u>\$ whereis</u> | Permite realizar la búsqueda dentro de archivos binarios o comandos   |
| <u>\$ find</u>    | Realiza la búsqueda dentro de un árbol de directorios que pertenecen al usuario y tengan los permisos binarios. Además permite hacer búsquedas más personalizadasEjemplo: \$ find . -user mauro -perm 644 |

## Herramientas para interactuar a través de HTTP

### Comandos

|                             |  |
|-----------------------------|--|
| <u>\$ curl paginaweb</u>    | Usado para hacer pedidos "Crudos". Se pide algo, se recibe la respuesta HTTP y eso sale en pantalla. En este caso nos devuelve lo que el navegador recibe cuando solicitamos una página web. |
| <u>\$ curl -v paginaweb</u> | No solo recibimos lo que recibe el navegador si no también toda la comunicación que se realiza via HTTP  |

|                                    |  |
|------------------------------------|--|
| \$ curl<br>paginaweb               | Usado para hacer pedidos "Crudos". Se pide algo, se recibe la respuesta HTTP y eso sale en pantalla. En este caso nos devuelve lo que el navegador recibe cuando solicitamos una página web. |
| \$ <u>wget</u><br><u>paginaweb</u> | Revisa descargas desde servidores HTTP y las realiza. Este comando descargará lo que se encuentre en el URL especificado.  |

## Acceso seguro a otras computadoras

Hay veces en las que queremos acceder a otra máquina por conexión remota.

### Comandos

|   |   |
|---|---|
| \$ ssh ip@usuario   | Permite la conexión hacia el ip especificado con el usuario             |
| \$ <u>echo "mensaje"   mail -s "Subject"</u><br><u>correo@electronico.com</u> | Permite realizar el envío de un correo a través de los parámetros dados |

## Qué son y cómo se utilizan las variables de entorno

Comando: Nombre de un programa con sus parámetros y modificadores.

La variable de entorno es una definición global a la que todos los procesos tienen acceso. Principalmente el sistema operativo o la terminal.

### Comandos

|                                 |   |
|---------------------------------|---|
| Column 1                        | Column 2  |
| \$ <u>echo</u><br><u>\$PATH</u> | Busca todos los archivos donde se pueden encontrar los comandos ejecutables. En estas rutas podremos tanto leerlas como modificarlas. |

## Asignación de las variables de entorno

Para un comando en particular (muy raro de realizar) o para toda la sesión

### Comandos

|                                       |  |
|---------------------------------------|--|
| \$ export VAR =<br>valor              | Sirve para declarar una variable en toda la sesiónEjemplo: export \$MI_VAR="Eyner" |
| \$ <u>VAR=valor</u><br><u>comando</u> | Sirve para modificar durante una ejecución en particular y para todo el entorno.   |

## Cómo y para qué escribir scripts en Bash

ScriptingBash: Se pueden realizar comandos para automatizar tareas

## Comandos

| Column 1                                  | Column 2  |
|---|---|
| <u>\$ vim</u><br><u>nombre_archivo.sh</u> | Ingresamos a la consola de vim para poder crear un archivo que se pueda ejecutar solo |

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

NEW\_DIR=platzi\_bash

```
if [ ! -d""/Users/carlosramirez/$NEW_DIR"" ]; then
```

```
mkdir""/Users/carlosramirez/$NEW_DIR""
```

fi

```
cp file.txt ""/Users/carlosramirez/$NEW_DIR/""
```

```
echo ""`date`: Todo listo CHARLY!!""`````
```

## Dejar tareas programadas

## Comandos Linux Clase

|  |  |
|--|--|
| \$ service -status-all                                   | Muestra el estado de todos los servicios actualmente.  |
| \$ sudo service <b>atd</b> start                         | Inicia el servicio de <b>atd</b>   |
| \$ sudo service <b>cron</b> start                        | Inicia el servicio de <b>cron</b>  |
| \$ at now + <b>numero_tiempo</b><br><b>tiempo_ingles</b> | Programa que se ejecuta una tarea después de un tiempo <b>numero_tiempo</b> determinado en unidades <b>tiempo_ingles</b> .<br>Ejemplo<br>at now +2 minutes<br>at> echo "Mensaje" > ~/tarea.txt<br>Se sale de la tarea con Ctrl + d   |
| \$ contab -e   | Muestra las tareas que están programadas de forma periódica y se pueden editar, dentro de este se ve:<br><div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <i>Minuto</i><br/>0<br/><i>Hora</i> </div> <div style="text-align: center;"> <i>Dia Mes</i><br/>18<br/> </div> <div style="text-align: center;"> <i>Dia Semana</i><br/>*<br/><i>Mes</i> </div> <div style="text-align: center;"> <i>Comando</i><br/> </div> </div> <u>echo "Hola" &gt; hola2.txt</u> |

```
*$ crontab -e
```