



UNIVERSIDAD DEL BÍO-BÍO

SISTEMAS
OPERATIVOS

Aspectos Básicos del Shell de Linux

Luis Gajardo
lgajardo@ubiobio.cl



¿QUÉ ES UN SHELL?

- Es un programa especial utilizado como una interfaz entre el usuario y el núcleo del Sistema Operativo (kernel).
- Interpreta y ejecuta comandos de dos maneras:
 - Manual: cuando el usuario lo escribe directamente en una terminal,
 - Shell-script: agrupando comandos en un archivo de texto que luego serán ejecutados.
- Estos shell-scripts, permiten también algunas instrucciones básicas de programación, como selección e iteración entre otras.
- Existen varios tipos de programas Shell (sh, bash, kshell etc.)

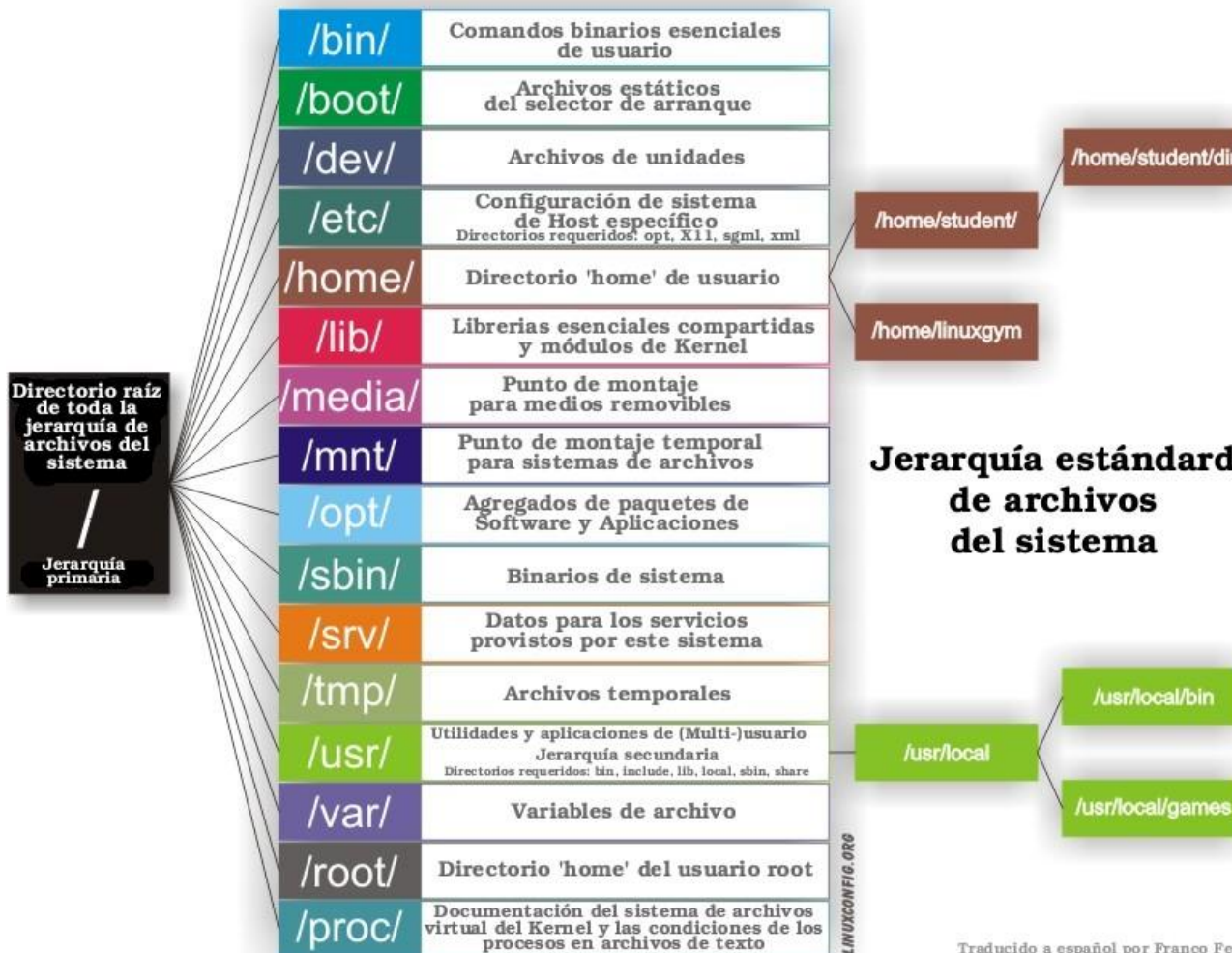


FILOSOFÍA DE UNIX/LINUX

- Sus principios derivan del proyecto MULTICS, estos son:
 - Todo se maneja como cadena de bytes (*streams* o *flujos*) o como entes que las producen.
 - Manejo de 3 descriptores estándar por defecto:
 - ***stdin***: usado para obtener datos (leer), por defecto el teclado.
 - ***stdout***: usado enviar datos (escribir), por defecto la pantalla.
 - ***stderr***: usado para enviar los errores, por defecto la pantalla.
 - Capacidades de entubar (pipe) y redireccionar: los descriptores pueden usarse para:
 - Cambiar el lugar en donde se leen los datos
 - Cambiar el lugar donde se envían los resultados o los errores.



EL SISTEMA DE ARCHIVOS



En una distribución GNU/Linux se basan en el estándar **F.H.S** (*File system Hierarchy Standard*), que deriva de la estructura tradicional de los sistemas UNIX.



VER EL DIRECTORIO

- Los comando más importantes para navegar por el sistema de archivos son:
- *ls* - Lista el contenido de un directorio.
 - `$ ls [opción]` Dónde opción puede ser:
 - `-a` : muestra todos los archivos del directorio, incluso los ocultos.
 - `-k` : muestra el tamaño de los archivos en KiloBytes.
 - `-l` : muestra toda la información sobre archivos y/o directorios.
 - `-x` : lista los archivos en columnas, ordenados horizontalmente.
 - `-R` : lista los contenidos de todos los subdirectorios recursivamente.
 - `-X` : ordena el contenido del directorio alfabéticamente.
 - `-t` : ordena el contenido del directorio por fecha de modificación.
- Ejemplo: `$ls -l`



VER EL DIRECTORIO

- Supongamos que ejecutamos el comando *ls*, de la sgte. manera:

```
$ls -l 
```

-rwxr-x--- 1 pepito depart1 4348 Nov 24 16:19 test

① ② ③ ④ ⑤ ⑥

- El resultado de este comando se puede interpretar como:
 - La primera columna, nos dice:
 - El primer guión indica que es un archivo (una d indica directorio).
 - El resto (rwxr-x---) indica los permisos de acceso que tiene el archivo.
 - La segunda columna, nos dice quien es el dueño del archivo, (pepito en este caso).
 - La tercera columna depart1, nos dice cual es el grupo del archivo (depart1 en este caso).
 - La cuarta columna 4348, nos dice el tamaño del archivo en bytes.
 - La quinta columna, nos dice cual es la fecha y hora de la última modificación.
 - La sexta columna test, nos dice cual es el nombre del archivo o directorio



NAVEGAR POR EL SISTEMA DE ARCHIVOS

- ***cd* - Change Directory** - Permite el cambio entre directorios.
 - `$ cd ..` Sube un directorio.
 - `$ cd` Se posiciona en el directorio /home del usuario.
 - `$ cd -` Se devuelve al directorio anterior.
 - `$ cd /home/alumno/tarea` Absoluto - Se cambia al directorio tarea
- ***pwd* - Print Working Directory** - Permite visualizar la path hacia el directorio actual.
 - `$ pwd`



CREAR Y BORRAR DIRECTORIOS/ARCHIVOS

- ***mkdir* - Make Directory** - Permite crear un directorio.
 - `$ mkdir <nombre-directorio>` Sube un directorio.
- Ejemplo:
 - `$ mkdir tareas` Crea el directorio tareas
- ***rm*** - Permite borrar un archivo o directorio.
 - `$ rm <nombre-archivo>`
 - `$ rm -r <nombre-directorio>`
- Ejemplos:
 - `$ rm /tareas/tarea1.c`
 - `$ rm /tareas/*.c`
 - `$ rm /tareas`



COPIAR Y MOVER ARCHIVOS/DIRECTORIOS

- *cp* - Permite copiar archivos o directorios.
 - `$ cp <archivo_origen> <destino>` Sube un directorio.
 - `$ cp -r <directorio_origen> <directorio_destino>`

- *mv* - Permite mover archivos o directorios.
 - `$ mv <archivo_o_directorio_origen> <destino>`

- Ejemplo:

```
$ mv *.txt /home/tareas
```

```
$ mv file_1.txt file_2.txt
```

Este comando es útil a la hora de renombrar archivos o directorios (moviéndolo con el nombre con el que se lo quiere renombrar) dado que la mayoría de las shell de Linux no poseen un comando específico para esto.



CARACTERES RESERVADOS DEL SHELL

Caracter	Descripción
\	Es el carácter de escape. Como se explicaba anteriormente, sirve para anular las funciones especiales de algunos caracteres.
/	Es el separador de directorios. Es usado para separar las distintas palabras que definen la ruta de un directorio
.	Directorio Actual
..	Directorio Padre
[]	Se puede usar para representar un rango de valores. Por ejemplo: tarea[1-9].txt representa tarea1.txt, tarea2.txt, etc.
	Pipe o tubería. Usado para redireccionar la salida de un comando a la entrada de otro.
>	Redirecciona la salida de un comando a un archivo de texto.
>>	Redirecciona la salida de un comando a la última línea de un archivo.



CARACTERES RESERVADOS DEL SHELL

Caracter	Descripción
<	Redirecciona un archivo como la entrada de un programa (lo mismo que pipe pero en diferente orden)
;	Se parador de comandos. Varios comandos en una sola línea.
&&	Separador de comandos, al igual que el anterior, pero con la condición que el primer comando debe ejecutarse sin ningún error antes de ejecutar el sgte.
&	Ejecuta un comando en background, es decir, en segundo plano. Permite recuperar el Shell para seguir escribiendo otros comando de inmediato.



PERMISOS DE USUARIO

- Son 3 tipos de permisos: lectura (r), escritura (w) y ejecución (x).
- Sobre 3 categorías de usuarios: propietario, grupo y otros.

Propietario			Grupo			Otros (público)		
r	w	x	r	w	x	r	w	X
4	2	1	4	2	1	4	2	1

- El comando *chmod* permite modificar los permisos de acceso de archivos y directorios:

`$chmod 750 test`

- da permisos **rw**x al usuario (**7**=111)
- da permisos **r**-**x** al grupo (**5**=101)
- da permisos --- a los demás (**0**=000)

`$chmod -R 755 directorio/` • -R indica recursivamente



EJEMPLO PERMISOS DE USUARIO

- Supongamos que ejecutamos el comando *ls*, de la sgte. manera:

```
$ls -l
```

```
-rwx r-x --- 1 pepito depart1 4348 Nov 24 16:19 test
```

① ② ③

- El resultado de este comando se puede interpretar como:
 1. pepito puede leer, escribir/modificar, y ejecutar el fichero test.
 2. Los usuarios pertenecientes al grupo depart1 puede leer, y ejecutar pero no escribir/modificar.
 3. Los demás usuarios no pueden hacer nada, ni leerlo, ni escribir/modificar, ni ejecutarlo.



EJEMPLOS DE PERMISOS DE USUARIO

Valor	Sentido
777	(rwxrwxrwx) No hay restricciones en los permisos. Cualquier persona puede hacer todo. En general, no es un escenario deseable.
755	(-rwxr-xr-x) El propietario del archivo puede leer, escribir y ejecutar el archivo. Todos los demás pueden leer y ejecutar el archivo. Este ajuste es común para los programas que son utilizados por todos los usuarios.
700	(rwx-----) El propietario del archivo puede leer, escribir y ejecutar el archivo. Nadie más tiene derecho. Esta configuración es útil para los programas que sólo el propietario puede utilizar y debe mantenerse privada de los demás.
666	(rw-rw-rw-) Todos los usuarios pueden leer y escribir en el archivo.
644	(rw-r - r--) El propietario puede leer y escribir en un archivo, mientras que todos los demás sólo pueden leer el archivo. Un valor común para los archivos de datos que todo el mundo puede leer, pero sólo el propietario puede cambiar.
600	(rw-----) El propietario puede leer y escribir en un archivo. Todos los demás no tienen derechos. Un valor común para los archivos de datos que el propietario quiere mantener en privado.



EJEMPLOS DE PERMISOS DE USUARIO

Valor	Sentido
777	(rwxrwxrwx) No hay restricciones en los permisos. Cualquier persona puede listar los archivos, crear nuevos archivos en el directorio y eliminar archivos en el directorio. En general no es una buena opción.
755	(-rwxr-xr-x) El propietario del directorio tiene acceso completo. Todos los otros pueden enumerar el directorio, pero no pueden crear archivos ni eliminarlos. Este ajuste es común para los directorios que desee compartir con otros usuarios.
700	(rwx-----) El propietario del directorio tiene acceso completo. Nadie tiene ningún derecho. Esta configuración es útil para los directorios que sólo el propietario puede utilizar y debe mantenerse privada de los demás.



- 16/16