



**Facultad de Ingeniería en
Electricidad y Computación**

Programación de Sistemas

CCPG1051

Federico Domínguez, PhD.

Unidad 1 – Sesión 2: Introducción a UNIX / LINUX

Agenda

1. Arquitectura UNIX / LINUX
2. Sistema de archivos LINUX
3. ¿Qué es el Shell de Linux?
4. Comandos básicos: *ls, cd, pwd, date, rm, mkdir, mv...*

Introducción a Linux/Unix

Breve historia de UNIX

- 1970: Creado en Bell Labs por Ken Thompson, Dennis Ritchie, Doug McIlroy y Joe Ossanna.
- 1973: Dennis Ritchie inventa el lenguaje **C** y lo usa para reescribir UNIX.
- 1980s: UNIX se hace popular en universidades y se crean numerosas versiones.
 - Unix 4.xBSD (Berkeley Software Distribution)
 - System V Unix (Bell Labs)
 - Solaris (Sun Microsystems)
 - AIX (IBM)
 - ...
- 1988: IEEE crea IEEE Std 1003.1-1988 -- mejor conocido como *Portable Operating System Interface (POSIX)* -- para estandarizar todas las distribuciones de UNIX.
- Actualidad: Usado ampliamente como sistema operativo de servidores comerciales por IBM, Oracle, HP y otros. FreeBSD es una versión código abierto de UNIX.



Introducción a Linux/Unix

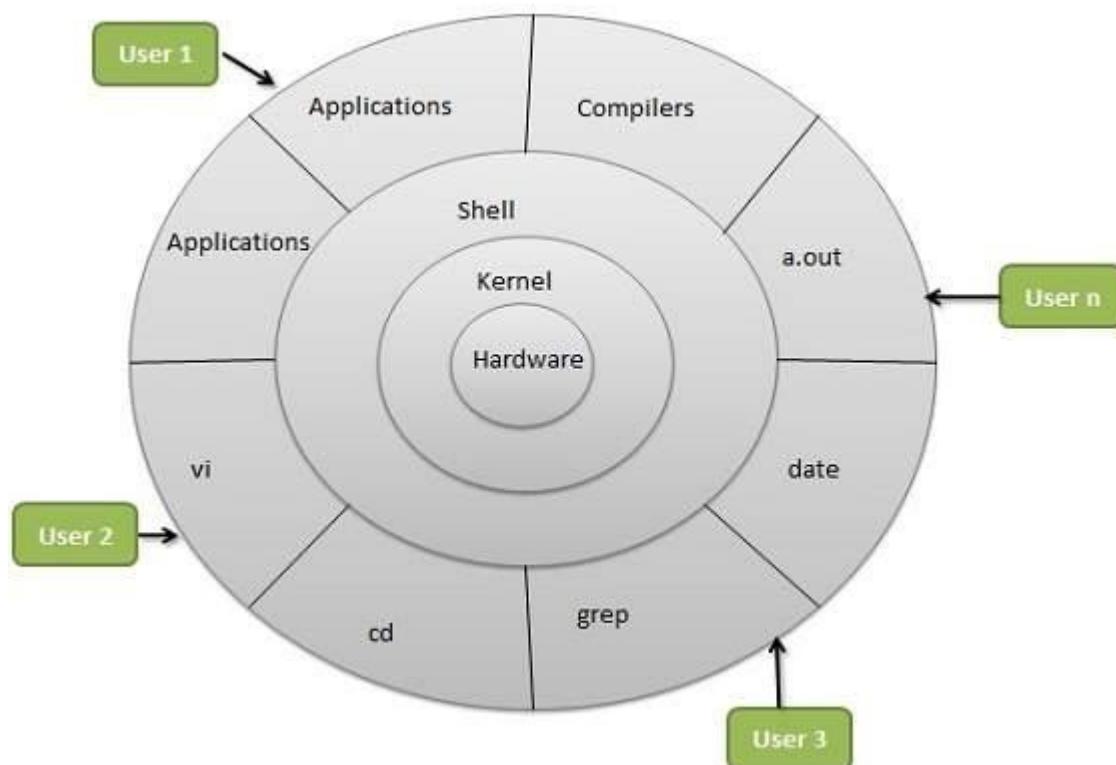
Breve historia de LINUX

- 1991: Linus Torvalds crea, como hobby, un sistema operativo código abierto basado en MINIX (una versión educativa de UNIX).
- Un juego de palabras entre UNIX y Linus, LINUX técnicamente se refiere tan solo al *kernel* del sistema operativo. LINUX se fusiona con el proyecto GNU de Richard Stallman para crear un sistema operativo completo.
- En lo posible, el *kernel* de LINUX y el software que lo acompaña es POSIX, manteniendo compatibilidad con UNIX.
- Software libre: El *kernel* de LINUX y varios otros componentes del sistema operativo usan la licencia GNU General Public License (GPL).
- Actualidad: Extremadamente popular, es el sistema operativo más usado. Desde supercomputadoras hasta relojes, el *kernel* de Linux esta instalado en miles de millones de dispositivos en todo el mundo.



Arquitectura de Linux/Unix

Arquitectura de LINUX



Hardware: CPU y periféricos

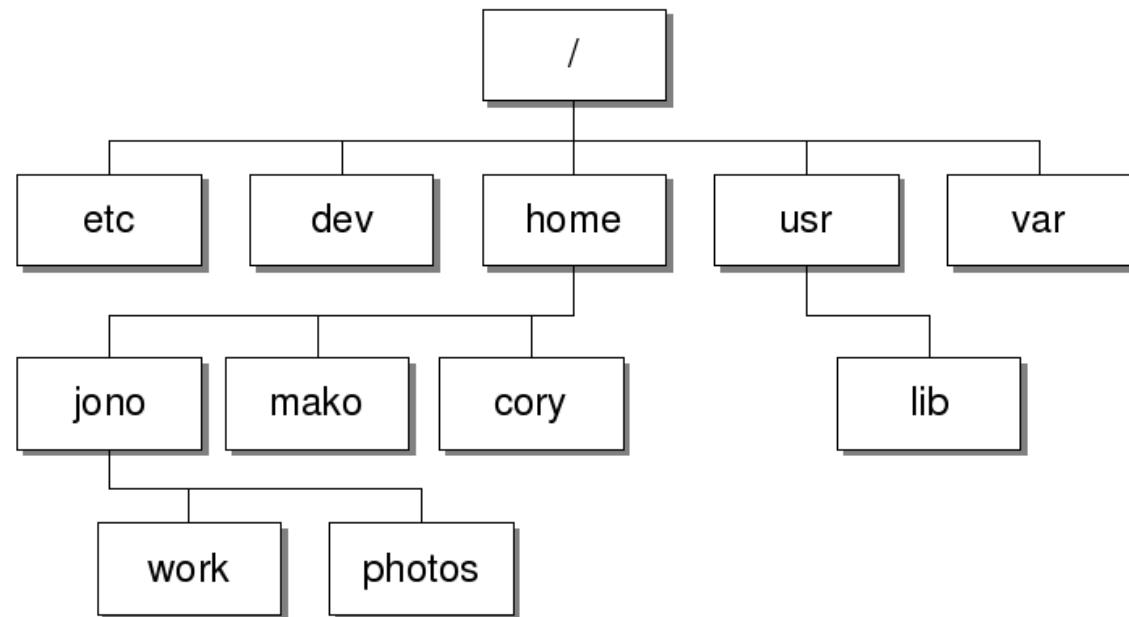
Kernel: Núcleo del sistema operativo, controla todo el funcionamiento del sistema.

Shell: Interface entre el Kernel y las aplicaciones de usuario. Esconde la complejidad del Kernel, ejecuta comandos de usuarios y aplicaciones.

Sistema de archivos en Linux/Unix

Sistema de archivos

- Es monolítico, empieza desde la raíz o “root” representada por “/”



Sistema de archivos en UNIX/LINUX

La distribución y uso del sistema de archivos de Linux se mantiene relativamente similar entre distribuciones.

- /etc: Archivos de configuración globales
 - /etc/passwd – Usuarios del sistema
 - /etc/fstab – dispositivos de almacenamiento y sus puntos de montaje
 - /etc/network – Configuración de red del equipo
- /home: Carpetas de usuarios
- /lib: Librerías del sistema
- /media: Puntos de montaje de dispositivos de almacenamiento
- /mnt: Igual que /media pero para versiones viejas/otras de Linux
- /usr: Contiene todos los programas y archivos de soporte
 - /usr/bin: Contiene los ejecutables del sistema.
 - /usr/lib: Contiene librerías compartidas
- /var: Datos que pueden cambiar son grabados aquí (bases de datos, logs, correos del sistema, etc..)
 - /var/log: Aquí se guardan los logs
- /dev: Representación de periféricos y dispositivos de hardware del sistema.

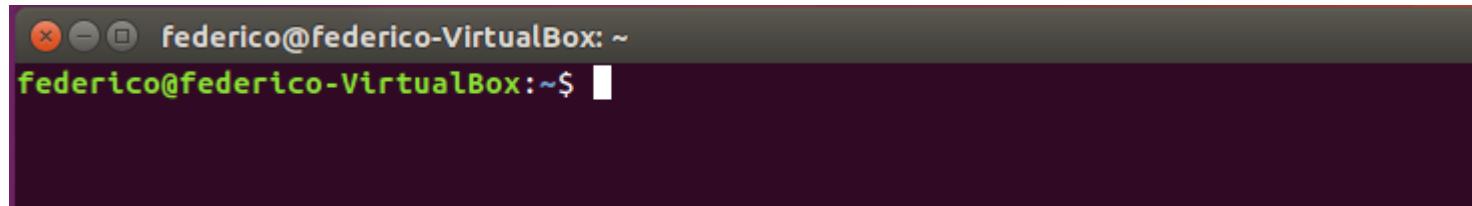
¿Qué es el Shell de Linux?

Conocido también como la línea de comandos, es el programa que interpreta los comandos del usuario y los pasa al sistema operativo (kernel) para su respectiva ejecución.

En Linux, el Shell más popular es **Bash** (*Bourne-again Shell*), creado para el proyecto GNU por Brian Fox. **Bash** está basado en **sh**, el Shell usado en UNIX y creado por Steve Bourne.

bash y **sh** no son los únicos shells, por ejemplo:

- ash: Shell de bajos recursos computacionales usado en FreeBSD y similares.
- dash: Shell usado en Debian.
- csh: Shell que usa C como lenguaje de scripting.

A screenshot of a terminal window titled "federico@federico-VirtualBox: ~". The window has a dark background and light-colored text. It shows a command prompt at the bottom: "federico@federico-VirtualBox:~\$".

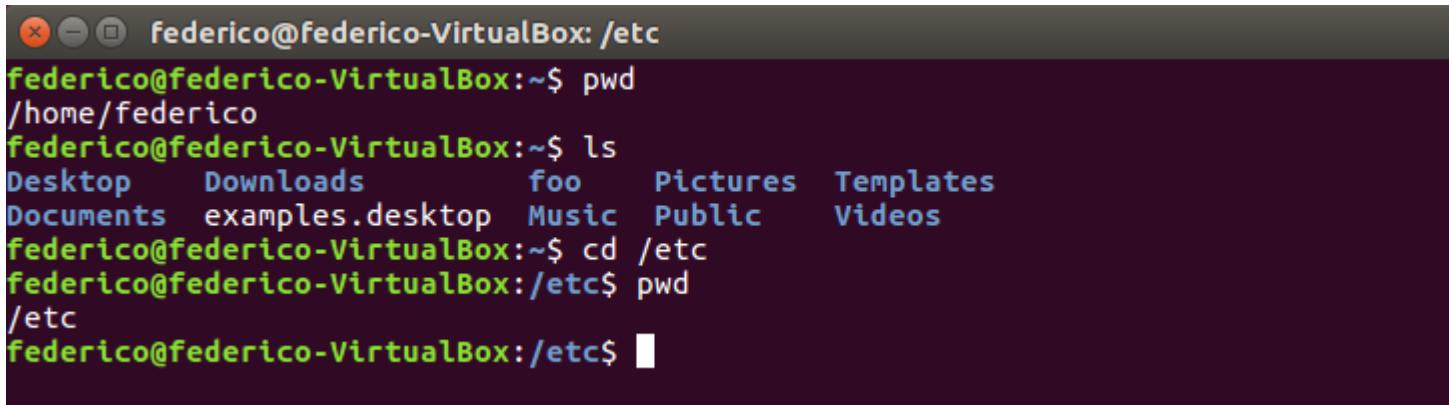
```
federico@federico-VirtualBox: ~
federico@federico-VirtualBox:~$
```

Comandos básicos del shell

¿Dónde estoy? → **pwd**: Print Working Directory

¿Qué hay aquí? → **ls**: List directory

Me voy a ... → **cd**: Change directory



A screenshot of a terminal window titled "federico@federico-VirtualBox: /etc". The window contains the following text:

```
federico@federico-VirtualBox:~$ pwd
/home/federico
federico@federico-VirtualBox:~$ ls
Desktop    Downloads      foo    Pictures  Templates
Documents  examples.desktop  Music  Public    Videos
federico@federico-VirtualBox:~$ cd /etc
federico@federico-VirtualBox:/etc$ pwd
/etc
federico@federico-VirtualBox:/etc$ █
```

Pathnames

Pathnames (caminos) es una dirección en el sistema de archivos. Tiene dos formas:

- Absoluta: Empieza desde la raíz
 - /etc/passwd
 - /home/pedro
 - /usr/lib/ssl/libcrypt.so.5
- Relativa: Empieza desde el directorio actual usando la notación especial “./”
 - ./bin o bin (si no empieza con “/” se asume que es relativo)
 - Casos especiales:
 - ../bin, .. significa el directorio padre
 - ~/Pictures, ~ significa el home del usuario

Opciones y argumentos

Comandos de Shell pueden recibir opciones y argumentos:

comando -opciones argumentos

Opciones cortas, son una letra, se especifican con –

- `ls -l` : Listar con formato largo
- `ls -a` : Listar mostrando archivos ocultos

Las opciones cortas se acumulan:

- `ls -la` : Listar con formato largo mostrando archivos ocultos

Opciones largas, son una palabra, se especifica con --

- `ls --all` : igual a `ls -a`

Formato largo de ls -l

% ls -al							
total 94							
drwxr-xr-x	2	john	doc	512	Jul 10 22:25	.	
drwxr-xr-x	4	bin	bin	1024	Jul 8 11:48	..	
-rw-r--r--	1	john	doc	136	Jul 8 14:46	.exrc	
-rw-r--r--	1	john	doc	833	Jul 8 14:51	.profile	
-rw-rw-rw-	1	john	doc	31273	Jul 10 22:25	ch1	
-rw-rw-rw-	1	john	doc	0	Jul 10 21:57	ch2	

Diagram illustrating the fields of the ls -l output:

- type
- access modes
- # of links
- owner
- group
- size (in bytes)
- modification date and time
- name

Explorar el contenido de un archivo de texto

Muchos archivos en Linux son de texto (en contraste a los archivos binarios). Un uso común de los archivos de texto son los archivos de configuración.

Para explorar el contenido de archivos de texto:

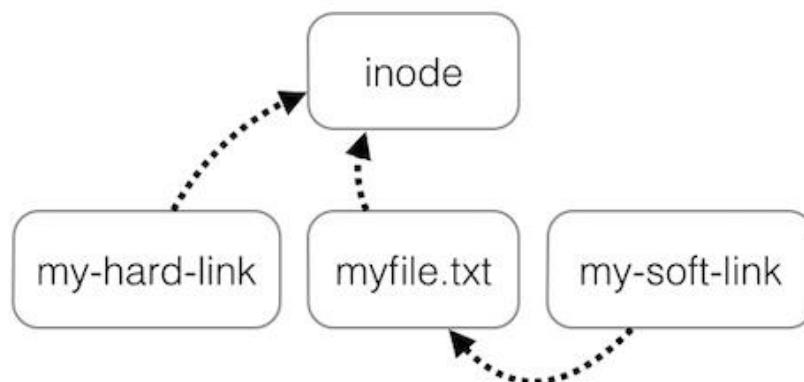
- `less`
- `more`
- `cat`
- `tail`
- Editores de texto como `vi`, `vim`, `nano`, `emacs`, etc.
- Ejemplo, ver las últimas 5 líneas del archivo de log del sistema:
 - `tail -n 5 /var/log/syslog`
- Recomendado: `less`, bastante completo y `cat`, bastante simple

Enlaces simbólicos

Comando `ln` crea enlaces o “alias” de archivos.

Dos tipos de enlaces simbólicos:

- `ln` : Enlace hard, es igual al archivo original, es un enlace a bajo nivel
- `ln -s` : Enlace soft, similar a un acceso directo.
- Ejemplo:
 - `ln -s /etc/passwd usuarios` (crea un enlace simbólico suave llamado `usuarios` al archivo `/etc/passwd`)



Wildcards

Una característica del Shell que nos permite agrupar nombres de archivos y directorios:

Wildcard	Meaning
*	Matches any characters
?	Matches any single character
[<i>characters</i>]	Matches any character that is a member of the set <i>characters</i>
[! <i>characters</i>]	Matches any character that is not a member of the set <i>characters</i>
[[: <i>class</i> :]]	Matches any character that is a member of the specified <i>class</i>

Wildcards

Pueden ser usadas con cualquier comando del shell, ejemplo:

- Listar todos los archivos que empiezan con b: ls b*
- Mostrar el contenido de todos los archivos que terminan con un número: cat *[0-9]

Data???	Any file beginning with “Data” followed by exactly three characters
[abc]*	Any file beginning with either an “a”, a “b”, or a “c”
BACKUP . [0-9][0-9][0-9]	Any file beginning with “BACKUP.” followed by exactly three numerals
[:upper:]*	Any file beginning with an uppercase letter
[![:digit:]]*	Any file not beginning with a numeral
*[:lower:]123	Any file ending with a lowercase letter or the numerals “1”, “2”, or “3”

Comandos de manipulación de archivos y directorios

`cp` : copiar archivos

`mv`: mover archivos, cambiar de nombre a archivos

`rm`: borrar archivos

`ln`: crear enlaces simbólicos

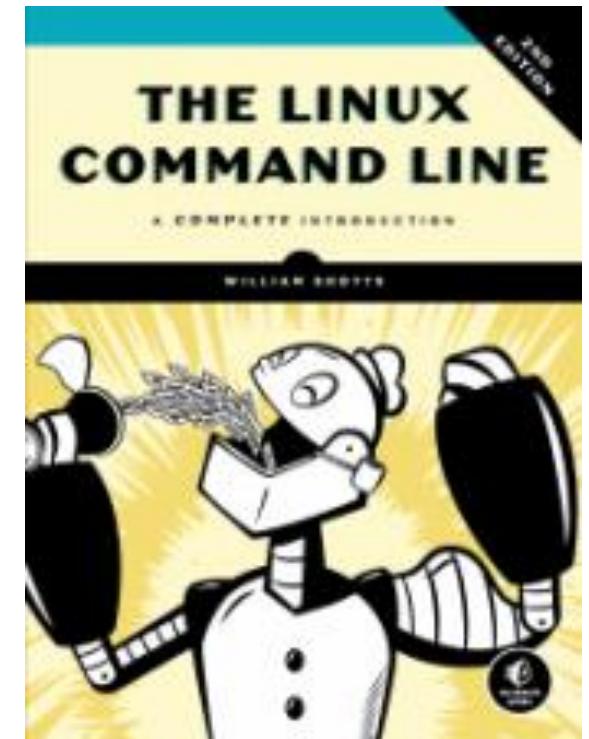
`mkdir`: Crear un directorio

Ejemplos:

- Cambiar de nombre al archivo `hola.txt` a `hello.txt`: `mv hola.txt hello.txt`
- Mover todos los archivos que terminan en `.txt` al directorio `texto`: `mv *.txt ./texto`

Referencias

Capítulos 1 al 6 y 9 del libro The Linux Command Line (TLCL)



Lectura para la siguiente sesión

Capítulos 1 y 2 de **The C programming Language** (Brian W. Kernighan and Dennis M. Ritchie) 2da edición

Control de lectura...

