

CURSO LINUX ADMINISTRATOR



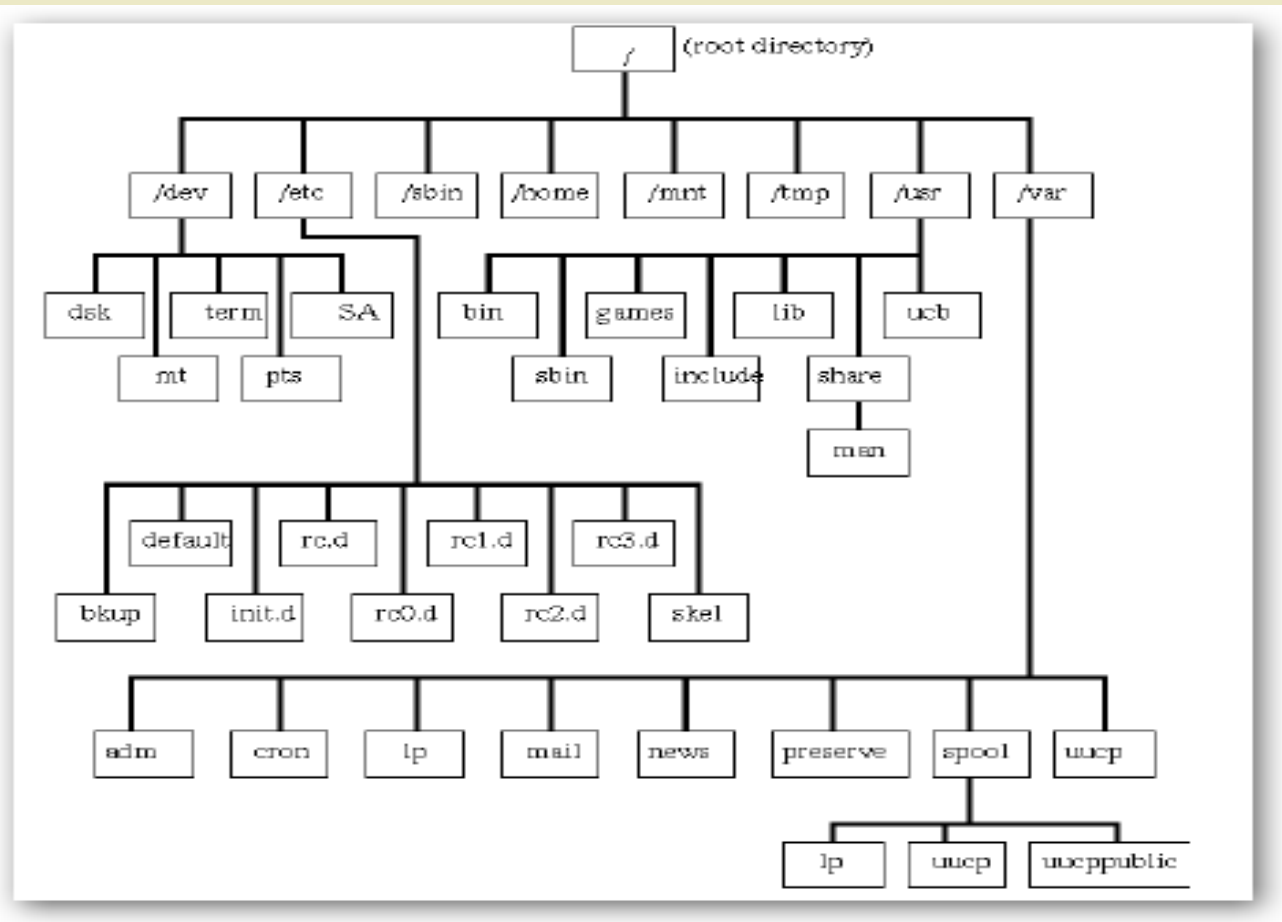
Linux
many flavours, many choices. Freedom

- INSTRUCTOR: RUDY SALVATIERRA
RODRIGUEZ

ESTRUCTURA DE DIRECTORIOS



ARBOL DE DIRECTORIOS EN LINUX



SISTEMA DE FICHEROS



La forma en que se almacenan los archivos:

- **Ext2:**
 - Desarrollado para y junto con Linux.
 - Permite todas las características que el sistema de archivos virtual de Linux soporta:
 - ✦ Directorios y archivos como inodos (con permisos, dueños, tamaño y ubicación),
 - ✦ Enlaces duros y suaves, fue durante mucho tiempo el sistema de archivos por defecto de linux.

SISTEMA DE FICHEROS



- **Ext3:**
 - Tiene las mismas características que el Ext2
 - Es transaccional, es el sistema de archivos por defecto de Linux actualmente aunque en la actualidad ha sido remplazado por su sucesor, ext4.
- **Ext4:** ("fourth extended filesystem" o "cuarto sistema de archivos extendido") es un sistema de archivos transaccional (Journaling), anunciado el 10 de octubre de 2006 por Andrew Morton, como una mejora compatible de ext3.

SISTEMA DE FICHEROS



- **Ext3:**
 - Tiene las mismas características que el Ext2
 - Es transaccional, es el sistema de archivos por defecto de Linux actualmente.
- **Ext4:** ("fourth extended filesystem" o "cuarto sistema de archivos extendido") es un sistema de archivos transaccional (Journaling), es una mejora compatible de ext3. Las principales mejoras son:
 - Soporte de volúmenes de hasta 1024 PiB.
 - Soporte añadido de extent.
 - Menor uso del CPU.
 - Mejoras en la velocidad de lectura y escritura.

SISTEMA DE FICHEROS



- **Xfs:**

Es un sistema de archivos de 64 bits con registro de bitácora o journaling de alto rendimiento creado por SGI (antiguamente Silicon Graphics Inc.) para su implementación de UNIX llamada IRIX. En mayo de 2000, SGI liberó XFS bajo una licencia de código abierto.

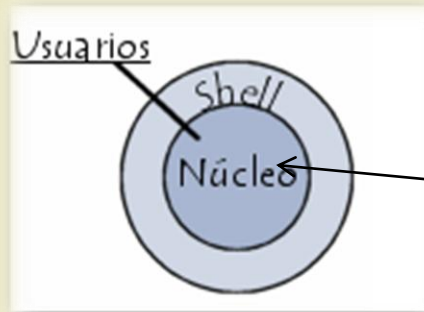
- Se incorporó a Linux a partir de la versión 2.4.25
- Los programas de instalación de las distribuciones de SuSE, Gentoo, Mandriva, Slackware, Fedora Core, Ubuntu y Debian ofrecen XFS como un sistema de archivos más.

CONCEPTOS BASICOS

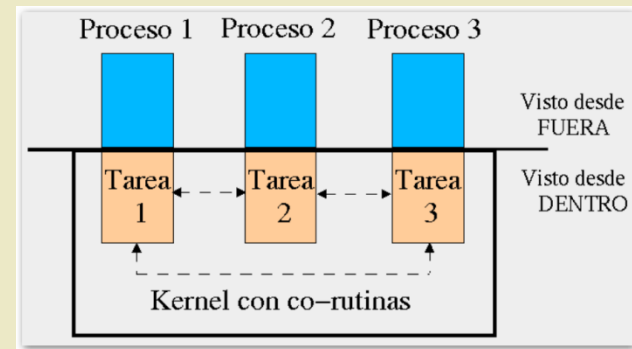


- **kernel:**

- Es el núcleo del sistema operativo.
- Es la parte encargada de gestionar los diferentes procesos y las posibles comunicaciones entre el hardware de un ordenador con los programas que están en funcionamiento.
- Es el que facilita el acceso a datos en los distintos soportes posibles (CD-ROM, unidad de disco duro, unidad ZIP, etc.), o el que arranca el ordenador, o el que resetea todos los dispositivos que sean necesarios.



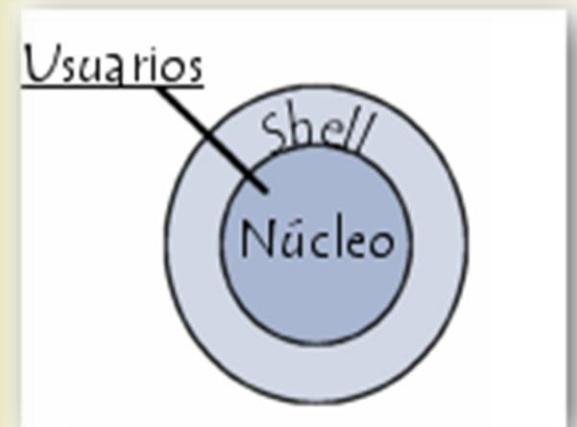
kernel



CONCEPTOS BASICOS



- **Shell:** Es el intérprete de comandos es la interfaz entre el usuario y el sistema operativo. Por esta razón, se le da el nombre inglés "**shell**", que significa **caparazón**.
- La Shell es un archivo ejecutable que debe interpretar los comandos, transmitirlos al sistema y arrojar el resultado.
- Existen varios shells. La más común es **sh** (llamada "Bourne shell"), **bash** ("Bourne again shell"), **csh** ("C Shell"), **Tcsh** ("Tenex C shell"), **ksh** ("Korn shell") y **zsh** ("Zero shell"). Generalmente, sus nombres coinciden con el nombre del ejecutable.



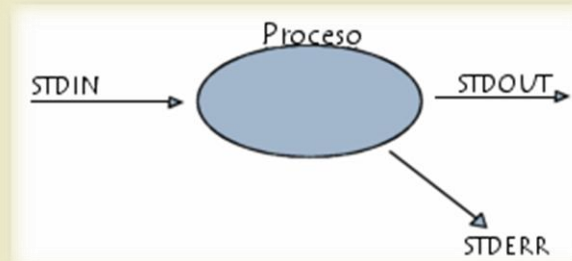
CONCEPTOS BASICOS



- **Entorno de trabajo:**

- Case sensitive: sensible a mayúsculas (ECHO != echo).
- Sintaxis: comando arg1 arg2... ArgN.
- Si un programa no está en el PATH: ./programa.
- Prompt:
 - ✦ \$: usuario normal.
 - ✦ #: usuario administrador (root).

- **Estrada y salida estandar:** Una vez que se ejecuta un comando, se crea un proceso. Este proceso abre tres flujos:



CONCEPTOS BASICOS



- **Estrada y salida estandar:**

- **stdin**, denominado **entrada estándar**, en cuyo caso el proceso lee los datos de entrada. De manera predeterminada, stdin se refiere al teclado. STDIN se identifica con el número 0.
- **stdout**, denominado **salida estándar**, en cuyo caso el proceso escribe los datos de salida. De manera predeterminada, stdout se refiere a la pantalla. STDOUT se identifica con el número 1.
- **stderr**, denominado **error estándar**, en cuyo caso el proceso escribe los mensajes del error. De manera predeterminada, stderr se refiere a la pantalla. STDERR se identifica con el número 2.

CONCEPTOS BASICOS



- Comandos para el manejo del sistema de ficheros
 - ls (lista contenido de directorios)
 - mkdir / rmdir (crea / elimina directorios vacíos)
 - cd / pwd (cambia /muestra la ruta de directorio)
 - touch (crea fichero vacío o actualiza existente)
 - cp / mv / rm (copia / mueve/ elimina fichero)
 - man / info (ayuda sobre comandos)
 - echo (muestra una línea de texto)
 - date / cal (muestra la hora del sistema)
 - file (muestra el tipo de fichero)
 - halt / reboot (apaga el sistema)

CONCEPTOS BASICOS



- Comandos para el manejo del sistema de ficheros
 - find (permite realizar busquedas de ficheros).
 - du (permite ver el tamaño de un directorio en kbp, megas, gigas).
 - df (permite ver las particiones montadas).
 - locate (permite ver la localizacion de un archivo).
 - tail (permite ver las ultimas lineas).
 - head (permite ver las primeras lineas).
 - mount (permite montar dispositivos)
 - umount (permite desmmontar dispositivos)
 - ln (permite crear enlaces simbolicos)
 - sort (permite ordenar lineas)

COMANDOS BASICOS



- Comandos para el manejo del sistema
 - who (Muestra el nombre de los usuarios que han ingresado al sistema.)
 - whoami (Muestra el nombre del usuario con el que estamos trabajando.)
 - grep (Muestra todas las líneas de un fichero dado que coinciden con un cierto patrón.)
 - top (Muestra la carga del sistema, CPU, etc.)
 - free (Muestra la cantidad de espacio libre y utilizado en la memoria RAM y swap.)
 - ps (Muestra los procesos que se están ejecutando en nuestro sistema actualmente).

COMANDOS BASICOS



ls : El comando para mostrar los ficheros y/o directorios.

Opciones

- **\$ ls -l** Permite ver los archivos y directorios en lista, informándonos sobre sus permisos, dueños de los archivos, tamaño, fecha y hora de creación y su nombre respectivo.
- **\$ ls -a** Permite ver los archivos y carpetas ocultas que contiene el directorio actual.
- **\$ ls -lh** Igual que ls -l, solo que el tamaño de los archivos están en kb o mb.
- **\$ ls -la** Igual q ls -l, solo que ahora también lista los archivos ocultos.
- **\$ lspci** Ver dispositivos conectados a la placa madre mediante un bus PCI.
- **\$ lsusb** Ver los buses USB y los dispositivos conectados a los mismos.
- **\$ lsmod** Ver los módulos del kernel

COMANDOS BASICOS



mkdir : Comando que nos permite crear directorios

- **Estructura basica:**

\$ mkdir primer_nombre_alumno

\$ mkdir -p semestre/curso_alumno/tema

- **Ejemplos:**

\$ mkdir carlos

\$ mkdir “carlos alberto”

\$ mkdir -p “2008-II/administración de servidores/comandos básicos”.

rmdir : Comando que nos permite borrar directorios

- **Estructura basica:**

\$ rmdir nombre_archivo

COMANDOS BASICOS



cd : Comando que nos permite movernos a través del árbol de directorios

- **Estructura basica:**

\$ cd / Esto nos moverá al directorio raíz.

\$ cd .. Subir un nivel en el árbol de directorios.

\$ cd ../ directorio2 Moverse en el mismo nivel de directorios.

\$ cd directorio Bajar un nivel, a la carpeta directorio.

\$ cd - Retornar al directorio que se ubicaba anteriormente.

- **Ejemplos:**

\$ cd nombre_completo_alumno

\$ cd /home/uladech/nombre_completo_alumno

COMANDOS BASICOS



rm : Borrar archivos y/o directorios. Este es un comando que debemos utilizar con mucho cuidado, ya que si borramos algunos archivos por equivocación, Sera imposible recuperarlos.

\$ rm archivo Borra un archivo.

\$ rm archivo1 archivo2 archivon Borrar varios archivos a la vez.

\$ rm * Borrar todos los archivos que se encuentran en la carpeta desde donde es ejecutado el comandos. (¡CUIDADO...!)

\$ rm carpeta/* Borra todos los archivos que se encuentran en carpeta, solo si esta vacia.

\$ rm -rf carpeta Borra todos los archivos y carpetas que contenga.

- **Ejemplos:**

- **\$ rm primer_nombre_alumno**

- **\$ rm -rf semestre**

COMANDOS BASICOS



cp : Copiar archivos y/o directorios.

- **Ejemplos**

\$ cp archivo /ruta/directorio Copiar archivo en /ruta/directorio/.

\$ cp arch1 arch2 arch3 /ruta Copiar arch1, arch2 y arch3 en /ruta

\$ cp archivo1 archivocopia Hacer una copia de archivo1 con otro nombre(archivocopia).

\$ cp * /ruta/directorio Copiar todo el contenido de la carpeta donde nos encontramos al momento de ejecutar el comando /ruta/directorio.

\$ cp -R directorio /ruta Copia la carpeta con todo su contenido en forma recursiva, hacia la carpeta /ruta.

\$ cp -rf directorio /ruta Copia carpeta y contenido a la ruta /ruta

COMANDOS BASICOS



mv: Comando con el cual puedo borrar archivos y/o directorios, también permite renombrarlos.

- **Ejemplos**

mv archivo /ruta mover archivo a la carpeta /ruta.

mv * /ruta mover todos los archivos y carpetas que se encuentran en la carpeta actual en el directorio /ruta.

mv archivo arch _renombrado cambiar de nombre a archivo por arch_renombrado.

mv directorio nuevo_directorio cambiar de nombre a directorio por nuevo_directorio.

COMANDOS BASICOS



pwd: El comando pwd indica el camino absoluto del directorio en el cual nos encontramos actualmente.

- **\$ pwd**

file: El comando file determina con cierto grado de precisión el tipo de un fichero que se le pasa como argumento.

- **\$ file nombre_fichero**

du: Comando que permite conocer el tamaño ocupado por un archivo y su respectiva jerarquía de directorios .

- **\$ du -h** Visualiza los tamaños de los directorios en forma representativa (M para Megabytes y K para kilobytes).

- **\$ du -sh directorio** Visualiza los tamaños de los directorios que contiene directorio.

COMANDOS BASICOS



less: Muestra un archivo en la pantalla pagina por pagina.

- **\$less archivo**

cat: Muestra un archivo en la pantalla.

- **\$ cat archivo**

tail: visualiza las ultimas líneas de un archivo.

- **\$ tail archivo**
- **\$ tail -n 10 archivo** muestra las 10 ultimas líneas del archivo.
- **tail -f archivo** muestra las líneas del archivo dinámicamente.

head: visualiza las primeras líneas de un archivo.

- **\$ head archivo**
- **\$ head -n 10 archivo** muestra las 10 primeras líneas de un archivo.

COMANDOS BASICOS



df: Se emplea para conocer información acerca de las particiones y dispositivos montados actualmente en el sistema, el espacio libre y ocupado por los dispositivos.

- \$ **df -h** Ver particiones montadas actualmente en Mb y Gb.
- \$ **df -Th** Ver particiones montadas con su formato de archivos.

fdisk: En Linux el particionador estándar es el fdisk. Este posee una interfaz texto que permite crear, modificar y borrar particiones de diversos tipos (Linux, FAT12/16 / 3 2, NTFS, minix, Linux Swap, HPFS, Novell, etc.).

- \$ **fdisk -l** Listar las particiones.

cfdisk: Editor de particiones fdisk (¡cuidado solo expertos!)

- \$ **cfdisk /dev/sda** sda es el primer disco duro.

COMANDOS BASICOS



touch: Permite crear ficheros vacíos

\$ **touch** Nombre_archivo.

Opciones

-**d** define el formato de la fecha(2009-08-15).

-**a** modifica la fecha de acceso.

-**m** modifica la fecha de modificación.

Otra forma de crear archivos es de la siguiente manera

\$ **>** Nombre_archivo.

more: Muestra un archivo en la pantalla pagina por pagina.

- \$**more** archivo

COMANDOS BASICOS



echo : Muestra en la terminal los argumentos pasados.

- **\$echo “cadena”**

sort : Permite ordenar lineas de un archivo.

- **\$sort archivo**
- **\$sort -r archivo** muestra las lineas en orden inverso
- **\$sort -u archivo** Ordenar un fichero eliminando las líneas repetidas:
- **\$sort -o archivo -u archivo** cuando se quiere eliminar las líneas repetidas de un archivo y dejar el contenido en el mismo archivo

COMANDOS BASICOS



whoami: Permite mostrar el usuario con el que estamos conectados actualmente.

whoami

who: Permite mostrar a todos los usuarios conectados en el sistema.

who

grep: Muestra todas las líneas de un fichero dado que coinciden con un cierto patrón.

grep <patrón> archivo

top: Muestra la carga del sistema, CPU, etc. dinámicamente.

top

COMANDOS BASICOS



free: Muestra la cantidad de espacio libre y utilizado en la memoria RAM y swap.

free

- **Opciones:**

-b muestra el espacio en bytes , -k muestra el espacio en KB,

-m muestra el espacio en MB

ps: Muestra los procesos que se están ejecutando en nuestro sistema actualmente.

ps

- **Opciones:**

a muestra todos los procesos, x muestra las consolas en las que han sido inicializados , u muestra el proceso en formato para el usuario.

COMANDOS BASICOS



date: Permite mostrar la fecha y hora del sistema.

\$date

- **Estableciendo la fecha hora y del sistema**

Para establecer la fecha y hora del sistema se usa 'date' seguido del siguiente patrón de entrada de datos:

`[MMDDhhmm[[CC]YY][.ss]]`

MM = mes, DD = día, hh = hora, mm = minuto, CC = siglo(Century),
YY = año, ss = segundos.

- Para establecer entonces la fecha al 20 de Julio del 2011 a las 8:05 de la noche:

- **\$date** 0720200511

vie jul 20 20:05:00 CDT 2011

COMANDOS BASICOS



- Pero regresemos a date, y ahora veamos como podemos obtener resultados de fechas en el pasado o presente, los más simples primero y todo con la opción "-d":

date -d "tomorrow" el dia de mañana

date -d "yesterday" el dia de ayer

date -d "today" el dia de hoy

wc: Permite contar las líneas de un fichero.

Estrucutura Basica

wc archivo

opciones

-l cuenta sólo las líneas de un fichero, -w cuenta las palabras de un fichero, -c cuenta los caracteres de un fichero

COMANDOS BASICOS



history: Permite mostrar el historial de todos los comandos ejecutados.

Estructura Basica

history

opciones

history 5 muestra los 5 últimos comando ejecutados.

!numero_comando ejecutar el comando que esta en la fila numero_comando.