CAPÍTULO 1.
LINUX -
COMANDOS BASICOS -
PROGRAMACION SHELL

MSc. Ing. Marcelo I Palma Salas
PhD Postulate
Docente UMSA - Investigador UNICAMP

LINUX
COMANDOS BÁSICOS

Introducción al S.O.

Material auxiliar

- Guía Linux - www.guiafoca.org
- The Linux Documentation Project – tldp.org
- Curso online gratuito “The Linux Foundation”
- https://training.linuxfoundation.org/training/introduction-to-linux/

La mejor forma de aprender Linux es usando

- Instalación en la PC
  - Dual Boot junto con Windows u otro SO
- Uso a través de un virtualizador
  - VirtualBox o VMWare dentro de otro SO
  - Imágenes de varias distribuciones ya listas en https://www.osboxes.org/virtualbox-images
- Live CD o PenDrive USB
  - https://osl.ugr.es/2020/05/20/como-instalar-ubuntu-1/
  - Ejecute Linux a partir de un pendrive sin alterar su sistema
  - https://www.youtube.com/watch?v=BJfAv9yOej8

GNU/LINUX

- GNU/Linux es el nombre del sistema operacional que usualmente conocemos como Linux.
- GNU: conjunto de herramientas como shell, compiladores, editores desarrolladas como alternativa libre a las herramientas de UNIX.
- Linux: Kernel (núcleo) del sistema operativo.
- LINUX + GNU = GNU/Linux o simplemente Linux.

¿Donde está Linux?

Sistemas embarcados

Servidores

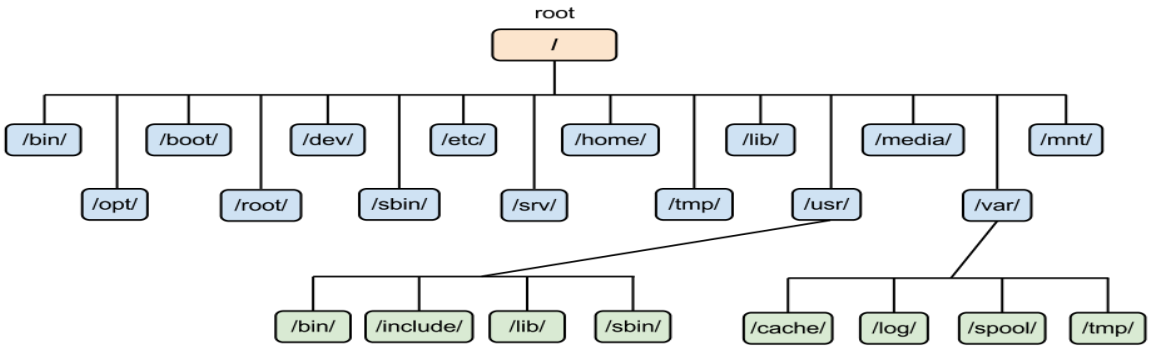
Supercomputadores (HPC)

PCs

Linux en Android

- Kernel Linux modificado por **Google**.
- Sin herramientas GNU
- Convergencia: Linux on Galaxy
- https://www.youtube.com/watch?v=lC0yVtu7NYw

Sistema de Ficheros: Definición

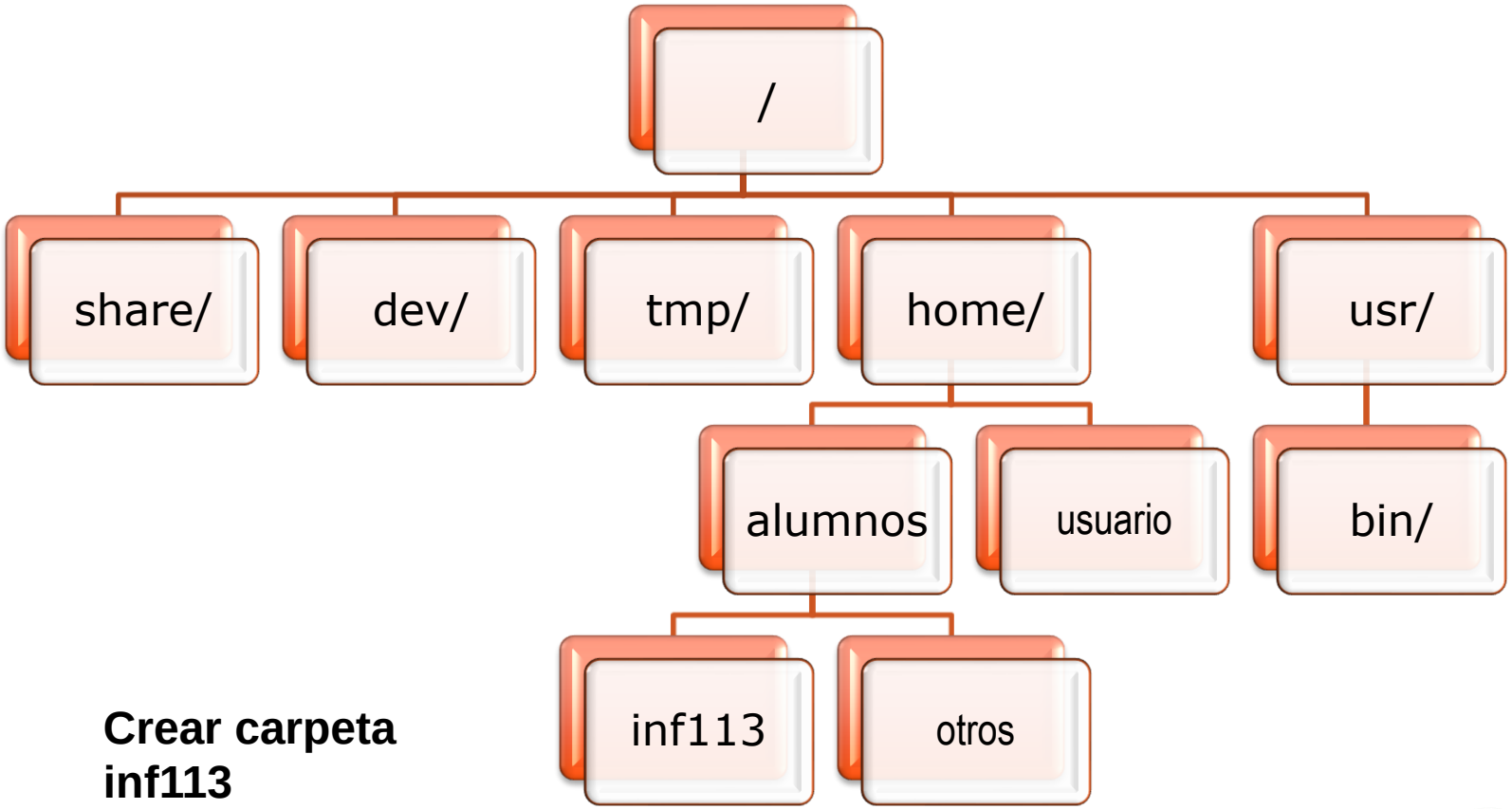


- El sistema de ficheros de un S.O. está formado por aquellas estructuras lógicas y sus correspondientes métodos que utiliza el propio sistema para organizar los ficheros en disco (memoria secundaria).

Sistema de archivos de Linux

- Estructura jerárquica de archivos
- Archivos
  - Directorios
  - Ficheros
  - Especiales
- Jerarquía: Se organiza en niveles
  - Nivel más alto Directorio Raíz /

Sistema de archivos de Linux



Crear carpeta inf113

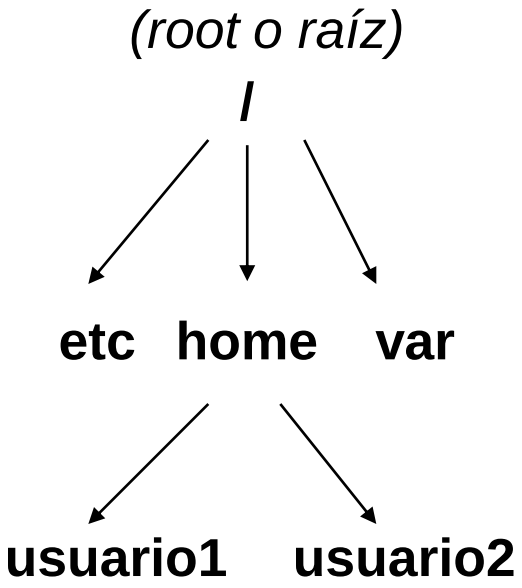
Directorios especiales

- . (punto): directorio actual.
- .. (dos puntos): directorio padre.
- ~ (caracter tilde): directorio home del usuario.
- ~alguien: directorio home de otro usuario.
- / (barra): directorio raiz del sistema.

- Nombres de archivos
  - Ficheros: nombre.extension
  - Directorios: nombre
- Diferencia may/min
- Nombres significativos

Sistema de Ficheros: Estructura

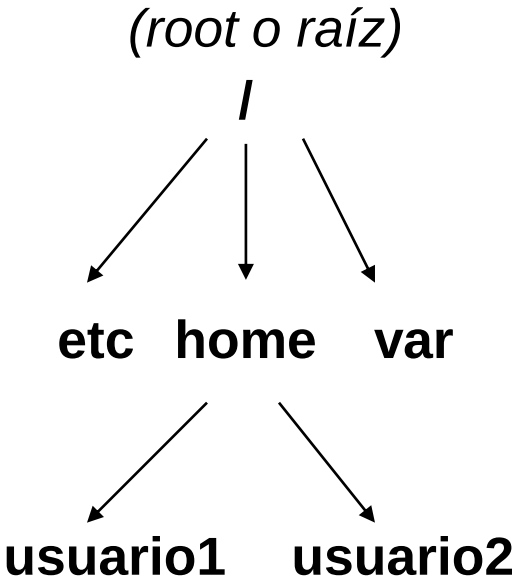
- GNU/Linux posee una estructuración jerárquica o “en árbol”.
- El sistema contiene unos directorios (que a su vez podrían contener más subdirectorios).
- Asocian características a los ficheros guardados en la partición.



Sistema de Ficheros: Estructura (II)

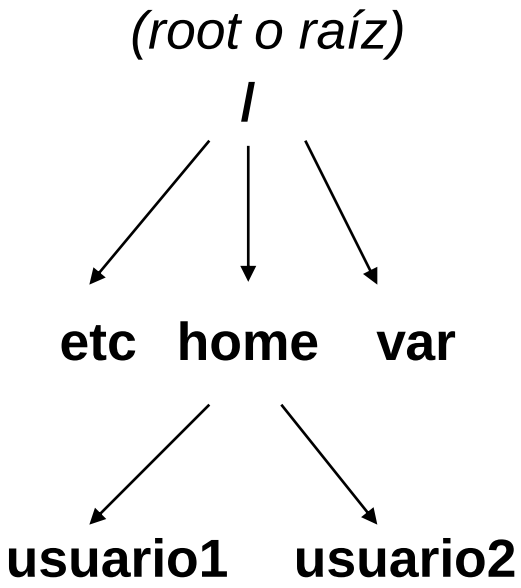
- Existen varios tipos de sistemas de ficheros en Linux.

- Se suelen clasificar en:
  - Sistemas de ficheros en DISCO.
  - Sistemas de ficheros en RED.
  - Sistemas de ficheros virtuales.
  - Sistemas de ficheros especiales.



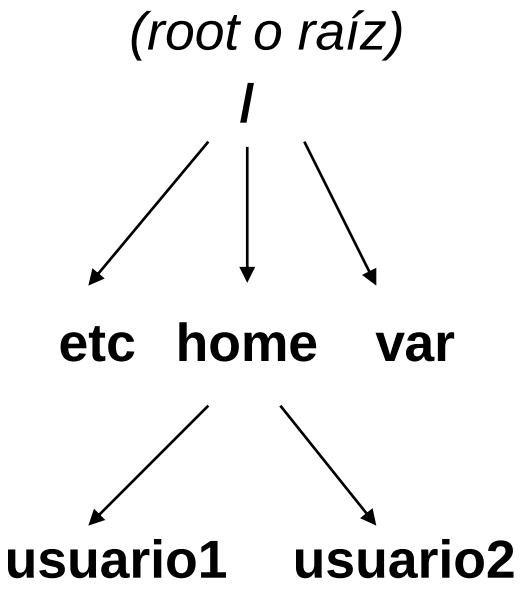
Sistema de Ficheros: Estructura (III)

- Sistemas de ficheros en DISCO.
  - Ext2
  - Ext3
  - ReiserFS
  - XFS
  - ISO9660
  - ...



Sistema de Ficheros: Estructura (IV)

- Sistemas de ficheros en RED.
  - NFS (Network Filesystem)
  - CIFS (Common Internet Filesystem)



Directorios importantes

- Directorio actual o de trabajo (**.**)
  - **pwd**
- Directorio Raíz (**/**)
- Directorio Padre (**..**)
- Directorio $HOME

Directorios importantes
Rutas y Rutas Absolutas

- Desde el directorio raíz
- Comienza con /
- Es un camino único, exacto
  - entre cada directorio poned una **/**

/home/alumnos/inf113

/home/usuario/Ejercicios

- Desde el directorio actual
- Se puede utilizar **./ ../**
- Podemos entrar a otras direcciones desde el directorio actual.

Sistema de Ficheros: Organización

- Organización de directorios por defecto:

/	Directorio raíz
/boot	Arranque del sistema
/bin	Binarios
/dev	Ficheros de dispositivos, periféricos,…
/etc	Ficheros de configuración
/home	Directorios de usuarios
/lib	Librerías compartidas
/mnt	Usado para montar particiones temporales
/proc	Información sobre el kernel y procesos

Sistema de Ficheros: Organización (II)

- Organización de directorios por defecto:

/tmp	Ficheros temporales de aplicaciones
/usr	Software, documentación, ...
/usr/src	Aquí están los fuentes del kernel
/var	Ficheros de log… (el contenido de este directorio puede variar mucho)
/var/spool/mail/	Se guardan los emails de los usuarios
/var/www/	Páginas que sirve apache

Sistema de Ficheros: Organización (III)

- Ficheros:
  - Hay varios tipos de ficheros
  - Podemos ver el tipo en el primer carácter del bloque de permisos:
    - **-** Fichero ordinario
    - **b** Dispositivo de bloques (disco duro, usb, ...)
    - **c** Dispositivo de caracteres (impresora, ...)
    - **d** Directorio
    - **l** Enlace

Consejos Básicos

- Mayúsculas distintas de minúsculas
- Cuidado con las flechas
- Al iniciar sesión:
  - No corregir errores al introducir la password
  - Presionar un par de veces ENTER hasta que aparezca el login
- Acordarse de cerrar la sesión
- Al iniciar linux hacer actualización.

  > sudo apt update & apt –y upgrade

Comandos Básicos

- **man (linux manual):** ayudad del sistema.
  - Sintaxis: **man <comando>**
  - Muestra la página del manual con la información correspondiente a este comando.
  - En ocasiones la ayuda se encuentra en inglés.
  - Puede ocurrir que no exista página para determinados comandos…

Estructura de un comando

comando [-opciones] [argumentos]

- Comando: nombre de la orden : ACCION
- -opciones : modifica el comportamiento del comando
- argumentos: nombres de ficheros o directorios sobre los que ejecutar el comando

Estructura de un comando
Comandos simples

- date
- cal
- who
  –q –H –b
- man nombre_de_comando
- clear
- ps

Comandos: Gestión de ficheros

- **cd (Change Directory):** cambiar de directorio
  - Al hacer login cada usuario está situado en su directorio de usuario
  - Sintaxis: **cd <PATH>**
  - PATH puede ser:
    - absoluto: creado desde el raíz
    - relativo: creado a partir del directorio actual
  - Si no ponemos el directorio volvemos al directorio HOME
- **pwd (Print Working Directory):** muestra el directorio actual
- Sintaxis: **pwd**

Comandos: Gestión de ficheros (II)

- **ls (List):** Muestra el contenido de un directorio
  - Sintaxis: **ls <nombredirectorio>**
  - Si no ponemos nombre de directorio saca el listado del directorio actual
  - Algunas opciones:
    - a Muestra todos los ficheros, ocultos incluidos
    - d Muestra el nombre de un directorio, no su contenido
    - l Formato largo, mostrando detalles
    - r Orden inverso
    - R Listado recursivo
    - color={never,always,auto} Colores basados en tipo de fichero

Comandos: Gestión de ficheros (III)

- **file :** Identifica el tipo de los ficheros
  - Sintaxis: **file <nombrefichero>**
  - Los tipos pueden ser:
    - Fichero de texto normal
    - Directorio
    - Impresora
    - Disco duro
    - Floppy
    - ...

Comandos: Gestión de ficheros (IV)

- **cat (Concatenate):** Concatena ficheros y escribe el resultado en la salida
- estándar
  - Sintaxis: **cat <fichero>**
  - Si no indicamos nombre de fichero, la entrada sería la entrada estándar
  - Algunas opciones:
    - b Numera las líneas no vacías
    - n Numera todas las líneas
    - v Muestra los caracteres no imprimibles

Comandos: Gestión de ficheros (V)

- **more :** Paginador de texto, muestra el texto en la pantalla poco a poco
  - *Sintaxis:* ***more <nombrefichero>***
  - Opciones:
    - D Muestra mensajes de información
    - num Especifica el número (*num*) de lineas por pantalla
    - +num Empieza a mostrar desde la línea *num*
  - Combinación de teclas útiles:
    - ESPACIO Muestra la siguiente pantalla de texto
    - RETURN Muestra la siguiente línea
    - q or Q or INTERRUPT Terminar
    - = Muestra el número de línea actual

Comandos: Gestión de ficheros (VI)

- **less** : Paginador de texto, muestra el texto en la pantalla poco a poco
  - Sintaxis: **less <nombrefichero>**
  - less añade la siguiente funcionalidad:
    - Permite usar los cursores para ir hacia delante y atrás
    - Navegar mediante número de líneas, porcentaje de fichero
    - No se termina al terminar el fichero
    - Combinaciones de teclas útiles:
      - num + G Va a la línea número num
      - g Va al inicio del fichero
      - G Va al final del fichero

Comandos: Gestión de ficheros (VII)

- **wc (Word Count)** : Muestra el número de catacteres, líneas o palabras de un fichero.
  - Sintaxis: **wc <fichero>**
  - Si ponemos el nombre de varios ficheros, mostrará los resultados de cada fichero y el total de todos ellos.
  - Opciones:
    - c Cuenta el número de caracteres
    - w Cuenta el número de palabras
    - l Cuenta el número de líneas

Comandos: Gestión de ficheros (VIII)

- **head** : Muestra las primeras líneas de un fichero
  - Sintaxis: **head <fichero>**
  - Por defecto muestra 10 líneas
  - Si queremos variar el número de líneas le pasamos el parámetro:
    - n donde n es el número de líneas que queremos que se muestren.

Comandos: Gestión de ficheros (IX)

- **tail :** Muestra las últimas líneas de un fichero
  - Sintaxis: **tail <fichero>**
  - Por defecto muestra 10 líneas
  - Opciones:
    - n Muestra las *n* últimas líneas del fichero
    - +n Muestra el contenido desde la línea *n* hasta el final
    - f Muestra como se va actualizando el contenido de un fichero

Comandos: Gestión de ficheros (X)

- **touch :** Cambia la fecha y hora del último acceso o modificación
  - *Sintaxis:* ***touch [opciones] [fecha] <fichero>***
  - Si el archivo no existe, lo crea con tamaño 0
  - Algunas opciones:
    - a Modificar la fecha de acceso
    - m Cambiar la fecha de modificación
    - t time Use la fecha especificada en lugar de la del sistema (MMDDhhmm

Comandos: Gestión de ficheros (XI)

- **cp (Copy) :** Copia archivos y directorios
  - *Sintaxis:* ***cp <origen> <destino>***
  - Por defecto sobreescribe ficheros del mismo nombre
  - Algunas opciones:
    - b No sobreescribe ficheros (hace backup)
    - i Modo interactivo (pregunta cuando puede haber problemas)
    - v Muestra los ficheros que han sido copiados
    - f Forzar la copia
    - r o R Copia recursiva

Comando CP

> $ cp archivo1.txt archivo2.txt

Copia archivo1.txt para archivo2.txt

> $ cp teste.txt /home/curso/fulano/backup

Copia el archivo teste.txt para el directorio /home/curso/fulano/backup

> $ cp * /home/curso/fulano/backup

Copia todos los archivos del directorio actual para /home/curso/fulano/backup

> $ cp /home/curso/fulano/aula1/* .

Copia todos los archivos del directorio /home/curso/fulano/aula1 para el directorio actual

> $ cp -R /home/curso/fulano /home/curso/backup

Copia el directorio /home/curso/fulano y todos los archivos/sub-directorios existentes para el directorio /tmp.

Comandos: Gestión de ficheros (XII)

- **mv (Move) :** Renombra o mueve ficheros
  - *Sintaxis:* ***mv [opcion] <origen> <destino>***
  - Opciones:
    - b Crea una copia de seguridad de los ficheros que sobreescribe
    - i Pregunta antes de sobreescribir ficheros
    - v Muestra el nombre de los ficheros que ha movido
    - f Fuerza a que se realiza la operación
  - También se usa para mover directorios:
    - Si el directorio destino no existe, el directorio origen es renombrado
    - Si el directorio destino existe, se copia el directorio como un subdirectorio de éste

Comandos: Gestión de ficheros (XIII)

- **rm (Remove):** Borra ficheros y/o directorios
  - *Sintaxis:* ***rm [opcion] <fichero>***
  - Opciones:
    - i Pide confirmación antes de borrar
    - f Fuerza el borrado de archivos protegidos contra escritura sin preguntar
    - r Borra ficheros y directorios recursivamente
  - Una opción útil, siempre usada con precaución, es ***-rf***, que borra recursivamente y sin preguntar.

Comando rm

> $ rm teste1.txt

Elimina el archivo teste1.txt sin consultar.

> $ rm alumno*

Elimina todos los archivos del directorio actual cuyo nombre comienta con el texto “alumno*”

> $ rm -rf /home/curso/fulano/backup

Elimina el directorio /home/curso/fulano/backup y todos sus archivos y subdirectorios. (COMANDO PELIGROSO)

Comandos: Gestión de ficheros (XIV)

- **mkdir (Make Directory) :** Crea un directorio
  - *Sintaxis:* ***mkdir [opcion] <directorio>***
  - Por defecto el directorio padre debe existir
  - Opciones:
    - p Crea todos los directorios padres si no existen
    - m Especifica los permisos que queremos dar al directorio

Comandos: Gestión de ficheros (XV)

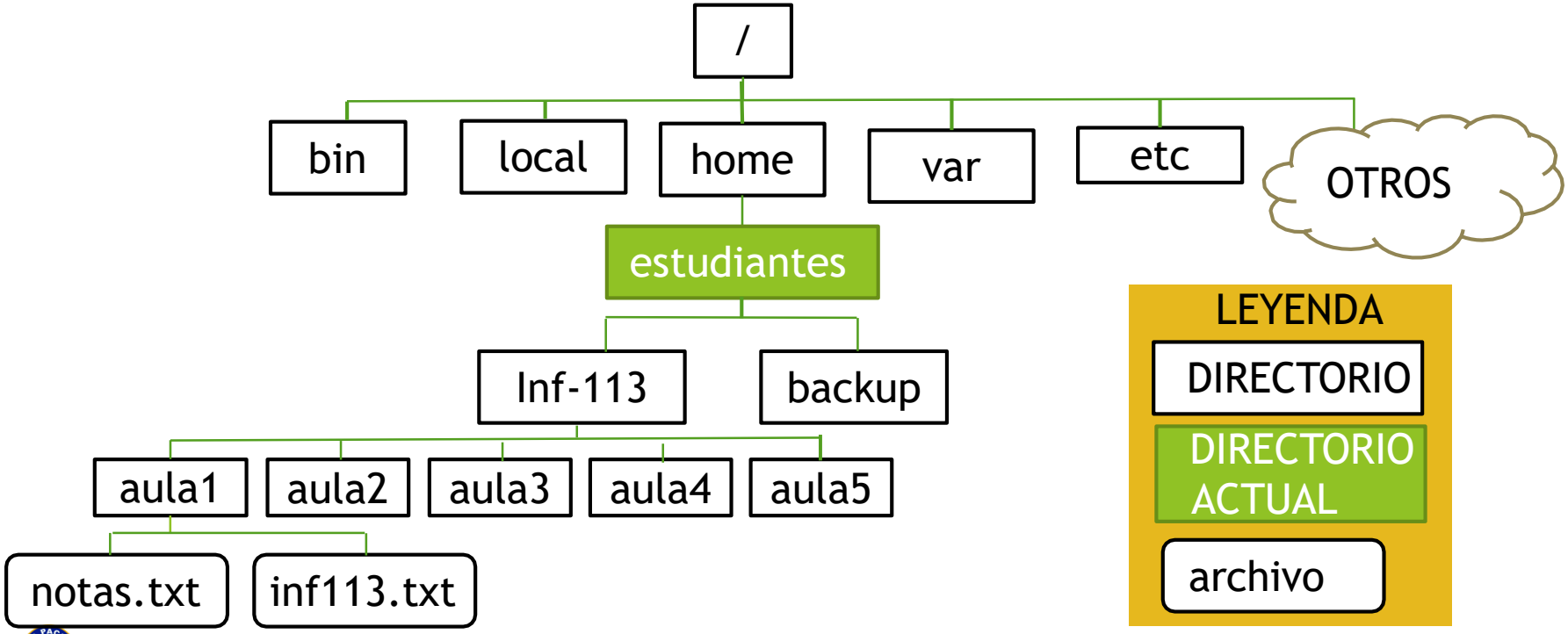
- **rmdir (Remove Directory):** Borra directorios
  - *Sintaxis:* ***rmdir [opcion] <directorio>***
  - Solo borra directorios que estén vacíos
  - Opciones:
    - p Elimina los directorios padres también

Comandos: Gestión de ficheros (XVI)

- **nano:** Pequeño y flexible editor de textos.
  - *Sintaxis:* ***nano <fichero>***
  - *Opciones de edición:*
    - *CTRL+O Guarda los cambios hechos en el fichero.*
    - *CTRL+X Sale del programa.*

Ejercicio 1

- Cree la siguiente estructura de directorios dentro de /home/estudiantes/inf113. Cree los archivo notas.txt y inf113.txt dentro de aula1.



Ejercicio 2

- En el archivo inf113.txt coloque sus anotaciones de aula.
  - Resuma como funciona los siguientes comandos:
    - cp
    - rm
    - mv
    - mkdir
    - rmdir
    - touch
    - ls
- En el archivo notas.txt escriba:
  - Alumno: nombre completo
  - E-mail: su email
  - Especialidad: Ing. de Sistemas, Ciencias de la Computación, etc. (https://shorturl.at/orLX5)

Ejercicio 3

- Haga una copia de toda la estructura del subdirectorio y archivos de /home/estudiantes/inf-113 para /home/estudiantes/backup con el comando cp.

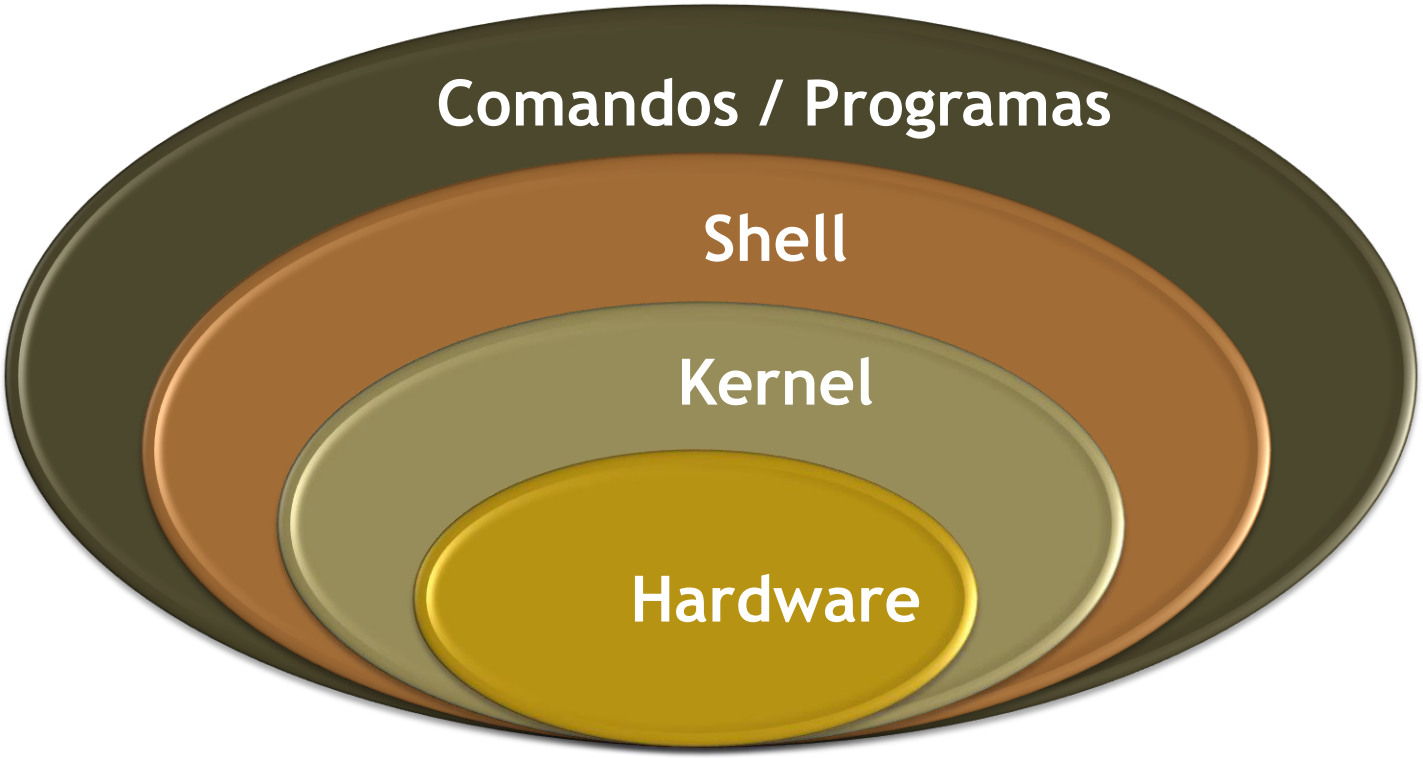
Desafio para la casa (opcional)

- Instale Linux o otra distro en una máquina virtual ejecutando dentro de Windows (usando VirtualBox o VMWare).
- Cree un live Linux en un pendrive USB.
- Sugiero usar Ubuntu, CentOS, Fedora, Debian, Kali (hacking) o otra distribución que usted este familiarizado.

Comando Avanzados Linux

Ingresando al Shell de Linux

Sistema Linux



Distribución Linux

- Kernel Linux + Shell + Programas que funcionan bien en conjunto.
- Centenas de distribuciones disponibles.
- Existen 3 familias conocidas (aunque hay más).

SLACKWARE

FEDORA

Distribución Linux

- Escoja una distribución con un largo historico de actualizaciones.
- Soporte de la comunidad.
- LTS (Long Term Support)

SLACKWARE

FEDORA

Bash – GNU

- Basado en el shell de UNIX (Stephen Bourne).
- Interpretador de comandos y lenguaje de programación.
- Historico de comandos y alias (apellidos):

  > $ history

  > $ alias
- Símbolo de menos (**-**) mas una letra para opciones.
  - ls -a
  - ls -al
- Dos señales de menos (**--**) y una palabra
  - ls --all (Equivalente a ls –a)
  - ls --almost-all (equivalente a ls –A)

Bash – GNU

- **Teclas especiales en el Shell**
  - Telcas de control de programas:
    - Ctrl+c ← Concluye el programa
  - Teclas útiles:
    - Ctrl+L ← Limpia la pantalla
    - Ctrl+U ← Elimina toda la línea de comandos
    - Tab → Completa comandos
    - Tecla ↑ o ↓ Intercambia comandos

Bash – GNU Bourne-Again SHell

- **Teclas especiales en el Shell**
  - *: Significa "Cualquier cosa" una o más de una vez.
    - Ex.: ls *.txt
  - ?: Significa "Cualquier cosa" exactamente una vez.
    - Ex.: ls inf???.txt
  - []: especifica un conjunto de caracteres de una lista o intervalo.
    - [a-z]
    - [0-9]
    - [0,2,4]
  - {}: todos los elementos de la lista o intervalo
    - > $ echo X{0,1,2}
    - $ echo X{1..5} X1 X2 X3 X4 X5
- E.g. en el aula pasada creamos directorios aula1, …, aula 5.
  - ¿Como crear los 5 directorios en un único comando?

  > $ mkdir aula{1..5}

Comandos: Redirecciones (I)

- **Entrada/Salida**
  - Los programas no suelen estar programados para coger la entrada o salida de un sitio concreto.
  - En lugar de eso se define el concepto de entrada estándar y salida estándar.

- La entrada estándar es el dispositivo en el que los comandos reciben los datos de entrada. Por defecto, el teclado.

- La salida estándar suele ser el dispositivo de salida de los programas. Por defecto, el monitor.
  - El intérprete de comandos (Shell) es el que se encarga de establecer la entrada/salida estándar para los comandos que se ejecutan

Comandos: Redirecciones (II)

- Podemos definir una **redirección** como el cambio de la entrada/salida de un comando.

- Tipos de redirecciones:

< fichero	Hace que la entrada estándar sea el fichero.
> fichero	Hace que la salida estándar sea el fichero, si el fichero existía perdemos su contenido.
>> fichero	Hace que la salida estándar sea el fichero, pero concatena la salida al final de este fichero.
|	Hace que la salida estándar de un comando sea la entrada estándar del siguiente.

Comandos: Redirecciones (III)

- **Vamos a ver algunos ejemplos:**
  - > $ cat > fichero
  - > $ cat < fichero
  - > $ cat fichero | less
  - > $ cat fichero | wc l
  - > $ ls > fichero
  - > $ ls | cat > fichero
  - > $ cat >> fichero
  - > $ cat < fichero >> fichero2

Comandos: Filtros

- **grep :** Busca línea a línea en un fichero un patrón o una cadena
- *Sintaxis:* ***grep [opciones] patron <lista ficheros>***
  - Opciones:
    - n Muestra el número de cada línea que coinciden
    - c Muestra la cantidad de líneas que coinciden
    - v Muestra las líneas que no coinciden
    - f Le pasamos un fichero que contiene los nombres de los ficheros a buscar

Comandos: Filtros (II)

- **grep**
  - Opciones:
    - i Ignora las diferencias mayúsculas/minúsculas
    - w Busca las coincidencias de la palabra entera
  - Hay que encerrar las expresiones con comillas simples. Las dobles pueden funcionar,pero es recomendable usar simples

Comandos: Filtros (III)

- **sort :** Ordena líneas en base a la ordenación de los caracteres ASCII
  - *Sintaxis:* ***sort [opcion] <fichero>***
  - Opciones:
    - +num Ordena a partir del campo número *num*
    - num Ordena hasta el campo número *num*
    - t delim Especifica el delimitador de campos, por defecto es tabulador o espacios
    - r Ordenación inversa
    - n Ordenación numérica
  - Ejemplos:
    - sort +1 fichero
    - sort t:+2 fichero

Comandos: Hardware

- En ocasiones es importante conocer ciertos detalles del hardware de un equipo.
  - **lsusb**: muestra los dispositivos usb conectados.
  - **lspci**: muestra los dispositivos conectados al bus pci.
  - **lshw**: muestra todos los dispositivos de hardware del equipo.

Comandos: Monitorización del sistema

- **du :** Devuelve el espacio ocupado por un fichero o directorio recursivamente
  - *Sintaxis:* ***du [opciones] [fichero]***
  - Opciones:
    - a Muestra el tamaño de los ficheros
    - b Muestra el resultado en bytes
    - h Muestra las medidas en forma más intuitiva

Comandos: Monitorización del sistema

- **df:** Devuelve la información sobre el tamaño de dispositivos como particiones.
  - Sintaxis: ***df [opciones] [fichero]***
  - Devuelve:
    - Tamaño del dispositivo
    - Número de bloques libres
    - Número de bloques ocupados
    - Porcentaje de espacio libre
    - Punto de montaje
  - Opciones:
    - h Muestra información sobre las unidades de medida utilizadas
    - m Muestra en megabytes

Comandos: Monitorización del sistema

- **top:** Es una forma de ver los procesos en ejecución.
  - Además muestra información sobre memoria y memoria swap
  - Tiene muchos comandos interactivos:
    - h Ayuda
    - M Ordenar procesos por memoria
    - P Ordenar procesos por uso de CPU
    - l Muestra las medias de uso

Comandos: Monitorización del sistema

- **ps**: Permite ver los procesos en ejecución
  - Opciones útiles:

    - f Procesos vistos en forma de árbol
    - a Muestra todos los procesos
    - u Muestra el usuario y la fecha de inicio
    - x Muestra los procesos asociados a un terminal

Comandos: Monitorización del sistema

- "Matar" procesos o tareas.
  - **kill** : Envía la señal de terminación a un proceso, identificado por su PID.
    - Sintaxis: **kill -9 <pid>**
  - **killall** : Envía la señal de terminación a todos los procesos que tengan un determinado nombre.
    - Sintaxis: **killall <nombre_proceso>**

Comandos: Monitorización del sistema

- **dmesg** :Cuando se está cargando el kernel va mostrando una serie de mensajes informativos…
  - Estos mensajes los guarda en **/var/log/messages**
  - Es útil, por ejemplo, para ver errores en la carga de software o de drivers, así como para identificar los nombres asignados a dispositivos hardware, como tarjetas de red, etc…

Comandos: Documentación

- Existen infinidad de comandos más…
- Se puede encontrar una referencia en:
- http://man.he.net/

Ejercicio 4 – Preparar el ambiente

1. En el directorio /home/estudiantes/inf-113/aula2 crear subdiretorios “ejemplos” e “ejercicios”.
2. Dentro de /home/estudiantes/inf-113/aula2/ejemplos ejecute el comando

    touch {a..z}{i,ii,iii}{1..10}.{a,c,so,o}

Este comando demorará um poco para crear todos los archivos.

Ejercicio 5

- Ejecute man ls y vea 3 opciones que puedan ser utiles. Describalos con sus palabras em:

  > ~/inf-113/aula2/ejercicios/ex02.txt

- Lea los manuales de los comandos: cp, mkdir, e rm
- Destaque una opción que pueda ser útil de cada manual y describa en sus propias palabras en ~/inf-113/aula2/ejercicios/ex03.txt

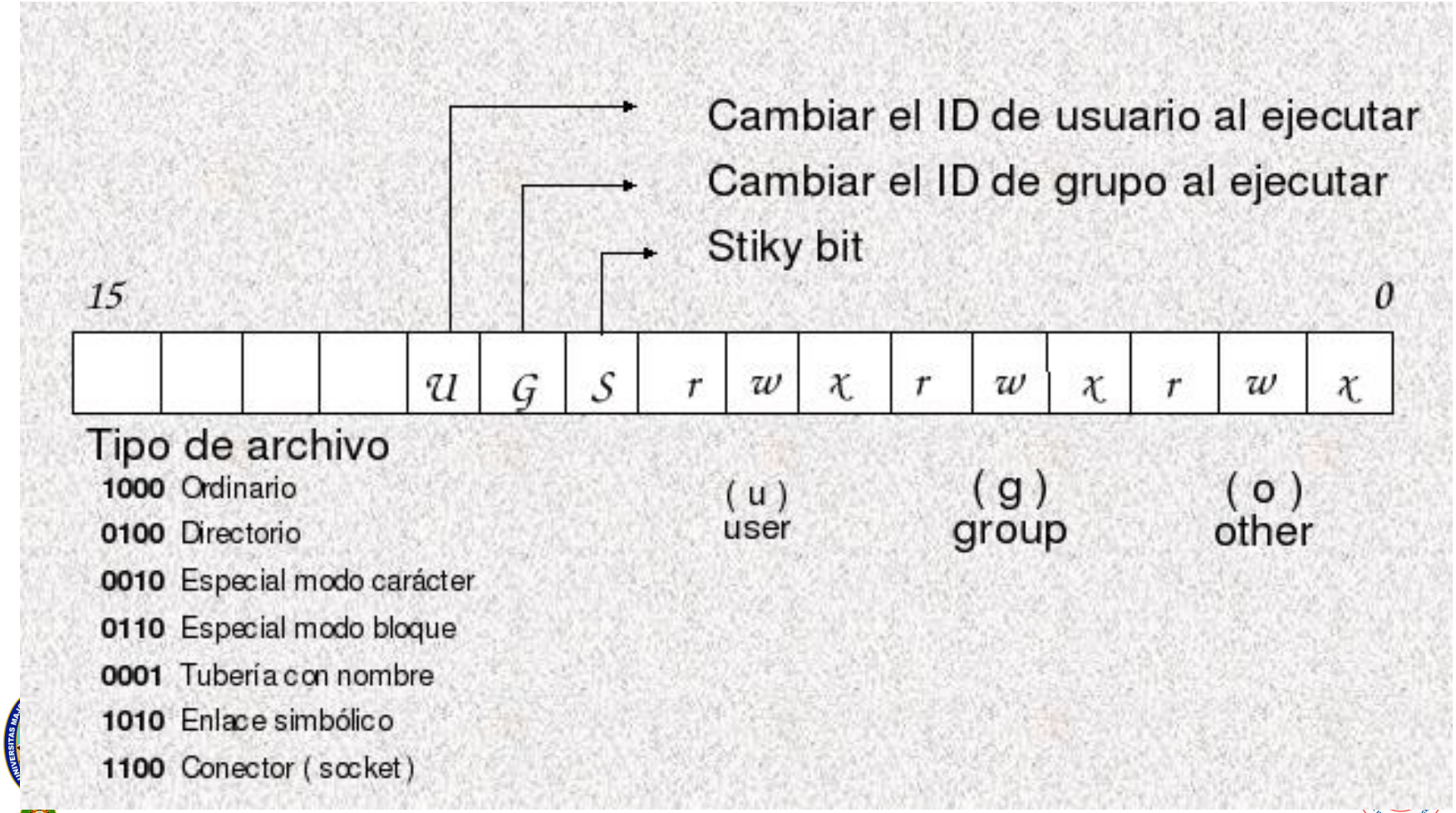
Sistema de Ficheros: Propiedades

- Los “objetos” de un sistema de ficheros, tienen una serie de propiedades.

- Las más importantes son:
  - El tipo de fichero: para indicar al SO cómo tratar con él.
  - Los permisos asociados: para implementar medidas de seguridad con ese “objeto”.

Sistema de Ficheros: Propiedades (I)

- En Linux, esas propiedades son una palabra de 16 bits:



Sistema de Ficheros: Propiedades (II)

- Según el diagrama anterior, los tipos de ficheros disponibles en Linux son:
  - Normales (ordinarios o regulares)
  - Directorios (carpetas)
  - Especiales:
    - Modo carácter
    - Modo bloque
  - Enlaces
  - Tuberias
  - Sockets

Sistema de Ficheros: Permisos

- Dueño y grupo
  - Todos los ficheros del sistema poseen un dueño y un grupo
  - El dueño suele ser el que ha creado el fichero, y el grupo suele ser el grupo por defecto de ese usuario.

```
-rwxr--r-- 1 root root 512 Nov 24 17:59 arranque.mbr
```

Sistema de Ficheros: Permisos (II)

- El bloque de permisos consta de 10 caracteres:
  - Tipo de fichero (explicado anteriormente…)
  - Permisos del dueño (3 caracteres)
  - Permisos del grupo (3 caracteres)
  - Permisos para los demás (3 caracteres)
- Los permisos básicos de un fichero se representan mediante:
  - r (read)
  - w (write)
  - x (execute)

```
-rwxr--r-- 1  root  root  512  arranque.mbr
```

Sistema de Ficheros: Permisos (III)

- **Acceso a ficheros**
  - r (read):
    - Permite ver el contenido del fichero, incluyendo la edición con un editor
  - w (write):
    - Permite modificar el contenido del fichero
    - No podemos editarlo si no tenemos permiso de lectura
    - No permite borrar el fichero
  - x (execute):
    - Permite ejecutar el fichero

Sistema de Ficheros: Permisos (IV)

- **Acceso a directorios**
  - **r:**
    - Se puede listar el contenido del directorio
    - No significa que podamos entrar en ese directorio
  - **w:**
    - Permite crear ficheros y directorios dentro
    - También permite borrar ficheros que haya dentro, incluso aunque no tengamos
    - permisos sobre ellos
  - **x:**
    - Permite cambiarse a ese directorio
    - Si no tenemos r no podremos ver el contenido

Detallando la salida del comando ls -al

```
curso@ubuntu$ ls -al
total 868
drwxr-xr-x  10 curso  cenapad   4096 Mar 20  08:27 .
drwx------  42 curso  cenapad   4096 Ago 27  10:53 ..
-rw-r--r--   1 curso  cenapad 373822 Mar 20  08:26 apostila_C.pdf
-rw-r--r--   1 curso  cenapad 450004 Mar 27   2017 curso_c.zip
drwxr-xr-x   2 curso  cenapad   4096 Mar 28   2017 lab01
drwxr-xr-x   2 curso  cenapad   4096 Mar 20  08:24 lab02
-rw-r--r--   1 curso  cenapad     32 Mar 27   2017 otro.txt
-rwxr-xr-x   1 curso  cenapad   8503 Ago 28   2017 renan
-rw-r--r--   1 curso  cenapad     66 Ago 28   2017 renan.c
-rw-r--r--   1 curso  cenapad    824 Mar 27   2017 texto.txt
```

DUEÑO

GRUPO

Tamaño en Bytes

PERMISOS

Permisos

Archivos

- r → leer archivo
- w → alterar el archivo
- x → ejecuta el archivo

Directorios

- r → listar el directorio, comando ls
- w → escribir en el directorio, modificar el contenido de este.
- x → ejecutar en el directorio, comando cd

Detallando la salida del comando ls -al

```
curso@ubuntu$ ls -al
total 868
drwxr-xr-x  10 curso  cenapad   4096 Mar 20  08:27 .
drwx------  42 curso  cenapad   4096 Ago 27  10:53 ..
-rw-r--r--   1 curso  cenapad 373822 Mar 20  08:26 apostila_C.pdf
-rw-r--r--   1 curso  cenapad 450004 Mar 27   2017 curso_c.zip
drwxr-xr-x   2 curso  cenapad   4096 Mar 28   2017 lab01
drwxr-xr-x   2 curso  cenapad   4096 Mar 20  08:24 lab02
-rw-r--r--   1 curso  cenapad     32 Mar 27   2017 otro.txt
-rwxr-xr-x   1 curso  cenapad   8503 Ago 28   2017 renan
-rw-r--r--   1 curso  cenapad     66 Ago 28   2017 renan.c
-rw-r--r--   1 curso  cenapad    824 Mar 27   2017 texto.txt
```

DUEÑO

GRUPO

Tamaño en Bytes

PERMISOS

Sistema de Ficheros: Permisos (V)

- **Cambiar propietario de un fichero…**
  - **chown :** Cambia el dueño de un fichero.
    - Sintaxis: **chown [opcion] <usuario> <archivo>**
    - Opciones:
      - c Muestra información de todos los cambios
      - f No muestra mensajes de error
      - R Recursivamente
  - **chgrp** : cambia el grupo.
    - Sintaxis: **chgrp <grupo> <ficheros>**
    - Se puede hacer también con chown:
      - chown usuario[:|.]grupo fichero(s)
      - chown [:|.]grupo fichero(s)

Sistema de Ficheros: Permisos (VI)

- **Cambiar los permisos de un fichero…**
  - **chmod :** Cambia los permisos de acceso a un fichero.
    - Sintaxis: ***chmod <quien> <cambio> <permisos> [ficheros]***
    - quien:
      - **u** propietario **g** grupo **o** otros
      - Si no se especifica, lo cambia a todos
    - cambio:
      - **-** quita permisos **+** dá permiso
    - permisos:
      - **r w x**

Sistema de Ficheros: Permisos (VII)

- **Cambiar los permisos de un fichero…**
  - **chmod :**
    - Opciones:
      - R recursivo
      - c Muestra los nombres de fichero de los que se han cambiado los permisos
      - f No muestra mensajes de error
    - Solo el propietario o el root puede cambiar los permisos de un fichero o directorio

Sistema de Ficheros: Permisos (VIII)

- **Cambiar los permisos de un fichero…**
  - **chmod :** Ejemplos
    - *chmod g+w fichero*
      - *Añade permiso de escritura para el grupo*
    - *chmod g=w fichero*
      - *El grupo solo tiene permiso de escritura*
    - *chmod ug+x fichero*
      - *Añade permiso de ejecución para propietario y grupo*
    - *chmod ug=x fichero*
      - *Configura permiso de ejecución para dueño y grupo*
    - *chmod +rwx fich**
      - *Dá permiso de lectura, escritura y ejecución a dueño, grupo y otros de los ficheros que cumplen ese patrón*
    - chmod o=g fichero
      - *Pone los permisos de otros iguales a los del grupo*

Sistema de Ficheros: Permisos (IX)

- **chmod : con números…**
  - En realidad, los permisos se identifican mediante 3 octetos
  - Cada octeto representa los permisos de lectura, escritura y ejecución en binario:
    - 111 > 7 (rwx)
    - 110 > 6 (rw-)
    - 100 > 4 (r--)
    - 010 > 2 (-w-)
    - 001 > 1 (--x)
  - Al chmod se le pueden pasar estos número para cada uno de los bloques de permisos:
    - chmod 764 fichero (Permisos rwxrwr)

Sistema de Ficheros: Usuarios

- Usuario **root:**
  - Cuando instalamos un sistema el único usuario que se crea el es el root
  - root es el administrador del sistema, puede hacer cualquier cosa:
    - Puede acceder a cualquier dispositivo
    - Puede borrar cualquier parte del sistema
  - Por eso el usuario root solo debe ser usado para realizar tareas de administración

Sistema de Ficheros: Usuarios (II)

- Comando **su:**
  - Permite realizar cambios de usuario con el que se está “logeado”
  - Se puede usar para:
    - Un usuario normal tiene que realizar algo como root
    - El usuario root necesita hacer algo como un usuario normal
  - Si queremos que todas las variables de entorno se cambien al usuario que cambiamos hay que usar la opción. Ejemplo:
    - su – <usuario>

Sistema de Ficheros: Usuarios (II) 2

- Comando **sudo:**
  - **sudo <comando a ejecutar>**
  - (Para ejecutar algo en concreto con el super usuario)

What do the fields in ls –al output mean?

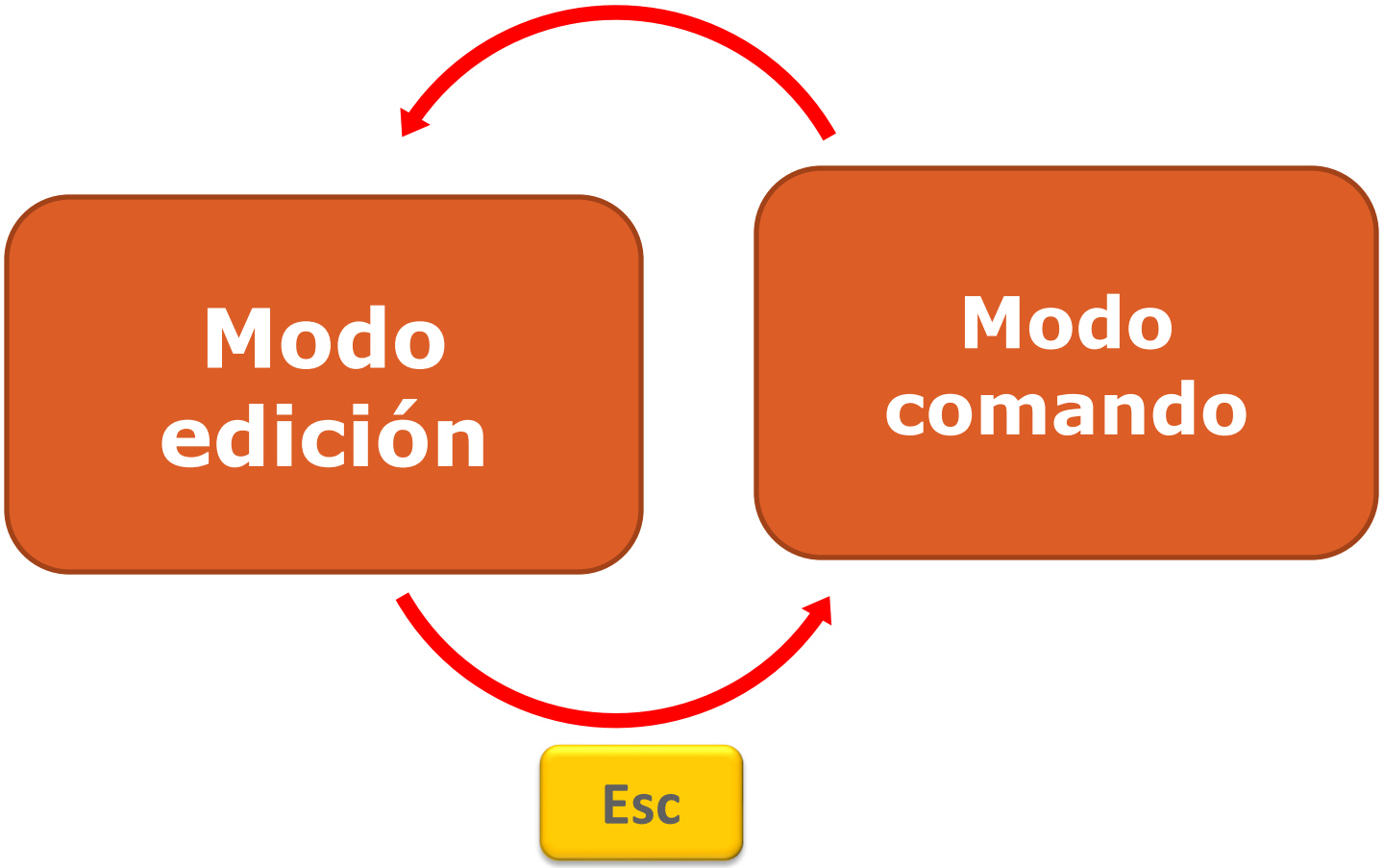
- StackExchange es um excelente sitio web para encontrar respuestas de calidad para dudas de informática (Linux/programación/web etc.)
  - https://unix.stackexchange.com/questions/103114/what-do-the-fields-in-ls-al-output-mean
- Por ejemplo, en el terminal:
  - Digite ls –al
    - -a (todos los archivos,inclusive los ocultos)
    - -l (formato largo)
  - Ahora digite ls –l
  - ¿Cual es la diferencia?

Editor VIM para la programación en Shell

Editor vim

- Sirve para crear nuevos ficheros, editar/escribir sobre ficheros, crear programas etc…
- No tiene menú, ni interfaz gráfico
- Tiene dos modos de trabajo:
  - **Modo comando:** para llevar a cabo operaciones sobre el fichero editado
  - **Modo edición:** para escribir
- vim nombre_del fichero.ext

Editor Vim



VIM (VI Improved)

- ¿Porque usar VIM?
  - Se puede usarlo en un terminal (conectado a un computador remoto).
  - Es un padrón. Está presente en cualquier sistema Linux.
- Editor de texto muy poderoso y muy complicado, pero puede ayudarlo en su día a día.
- **Sugerencia:** conozco lo básico o suficiente para salvarlo en situaciones de emergencia.

Editor Vim

- Modo comando:
  - Moverse:
    - con las teclas del cursor
  - Borrar
    - un carácter : esc + x
    - una línea : esc + dd
  - Deshacer
    - el último cambio: esc + u
  - Refrescar la pantalla
    - CTRL L

Editor Vim

- Modo comando:
  - Buscar una palabra
    - Hacia delante : esc + /palabra_a_buscar
    - Siguiente : esc + n
    - Hacia atrás: esc + ?palabra_a_buscar
  - Copiar un bloque de texto
    - esc + numeroyy
  - Pegar el bloque copiado
    - esc + p

Editor Vim

- Modo comando:
  - Ir a una línea
    - esc + :numerolinea
  - Colorear
    - esc + :syntax on
  - Indentar
    - esc + :set autoindent

Editor Vim

- Modo comando:
  - Guardar : esc + :w
  - Guardar y salir : esc + :wq
  - Salir sin guardar: esc + :q!
  - Salir (habiendo guardado): esc + :q

Ejercicio 6

- En aula3 cree un archivo llamado nombres.txt con VIM
- Escriba el nombre de 10 personas ficticias, una por línea, con algunos nombres repetidos. Use VIM
- Ejecute los siguientes comandos.
  - cat arquivo.txt | sort
  - cat arquivo.txt | sort | uniq
  - cat arquivo.txt | sort | uniq –c
  - cat arquivo.txt | sort | head –n 3
  - cat arquivo.txt | sort | tail –n 3
  - cat arquivo.txt | sort | head –n 3 > tres_primeros.txt
- Cree un archivo llamado notas.txt en aula3 y describa que hace cada comando de arriba.

Introducción a Programación en SHELL

Comando ps

- La salida contiene PID TTY TIME COMMAND
  - $ ps –e (lista todos los procesos)
  - $ ps –ef (lista todos los procesos, con más detalle)
  - $ ps –fu “usuario” (lista todos los procesos del usuario)

- PID: process id
- TTY: terminal device
- TIME: tiempo de cpu usada hasta ese instante.

Comando para Matar Procesos

- Comando **kill:**
  > $ kill –SIGKILL pid (mata proceso)
  > $ kill –SIGSTOP pid (para proceso)
  > $ kill –SIGCONT pid (continua proceso)
  > $ kill –SIGSEGV pid (genera un error de segmentación en el proceso para finalizarlo)

Comandos diversos y Ejecución en paralelo

- file – detecta el tipo de archivo
- Gerenciamiento:
  - free → memoria ram
  - df → disco
  - passwd → cambiar contraseña

- Ejecutar procesos en background:
  - firefox &
  - gedit &

Ejercicio 7

- Use el comando ps para saber
  - Que procesos está ejecutando el actual usuario.
  - Que procesos están ejecutando todos los usuarios.
  - Abra Firefox u otro navegador y encuentre su PID.
  - Mate el proceso del navegador con el comando kill.

- Abra gedit via línea de comando.

- Use el top para determinar que procesos están consumiendo mas recursos.

- Use “man top” para entender todo respecto al comando top. Escriba un resumen en /aula4/ex04.txt.
- Mate algún proceso no esencial con top y tecla “k”.

Ejercicio 8

- Cree un script llamado forever.sh en aula4, usando vim o nano.
- Copie el contenido abajo y guardelo. Cambie los permisos del archivo para conceder permisos de ejecución para forever.sh.
- Ejecute el script con $ ./forever.sh &
- Encuentre el script con ps.
- Mate el script con kill.

```
#!/bin/bash
echo "Iniciando el proceso infinito"
while [1]
do
        sleep 60
        echo "Un bucle infinito por siempre"
done
```

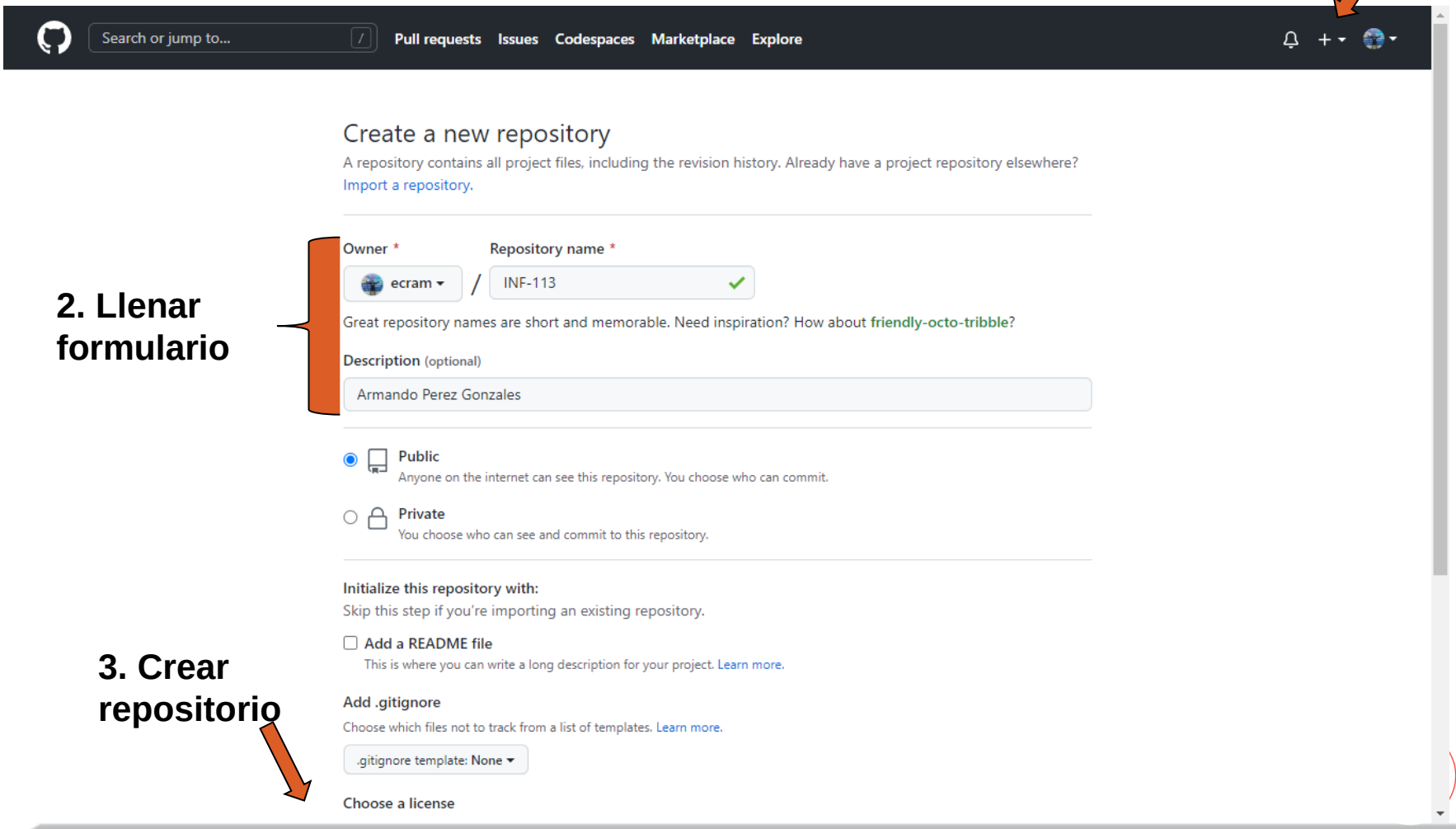
Pasos Básicos para Crear un Repositorio en GitHub (I)

- GitHub nos va a permitir gestionar nuestros proyectos en un repositorio en línea.
- El curso de INF-113 utilizará esta plataforma para subir las prácticas de Linux, HTML, CSS, JavaScript, etc.
- Siga los pasos:
  1. Crea una cuenta en github.com
  2. Crea un repositorio haciendo click en "+" en la esquina superior derecha.
     1. Nombre del Repositorio: INF-113
     2. Descripción: Nombre del estudiante
     3. Tipo: *Public* o público
     4. *Create repository* o crear repositorio.

104

Pasos Básicos para Crear un Repositorio en GitHub (II)

1. Nuevo repositorio



2. Llenar formulario

3. Crear repositorio

Pasos Básicos para Crear un Repositorio en GitHub (III)

- En el computador ejecutar los siguientes comandos:
  1. En /home/estudiantes/Documentos crear una carpeta llamada inf113 e ingresar:

     > $ cd ~/Documentos
     > $ mkdir inf-113
     > $ echo "# inf-113" >> README.md
     > $ git init
     > $ git add README.md
     > $ git commit -m "primera confirmacion"
     > $ git branch -M main
     > $ git remote add origin git@github.com:**<nombre_usuario>**/INF-113.git
     > $ git push -u origin main
  2. **Importante:**
     1. GitHub puede solicitar su email y usuario de github:

        git config --global user.email "email@gmail.com"
        git config --global user.name "usuario"

Pasos Básicos para Crear un Repositorio en GitHub (IV)

- Por último, ingresar al siguiente formulario y adjuntar la dirección de su repositorio GitHub INF-113.
  - https://forms.gle/x3jahMgFoXjDAr4p7
- En caso de dudas puede revisar el extenso material en Internet y Youtube (https://www.youtube.com/watch?v=eQMcIGVc8N0)
- En caso de errores:
  - StackOverFlow

Pasos Básicos para Actualizar su Repositorio en GitHub (V)

- Una vez hecho algún cambio en su repositorio, deberá realizar los siguientes pasos:
  1. Verificar los cambios en el repositorio:
     > $ git status
  2. El sistema exibirá los archivos modificados y se deberá agregar los mismos con git add:
     > $ git add .
  3. A continuación se realiza un commit adicionando un mensaje:
     > $ git commit –m “Se agrega un mensaje en el main”
  4. Git push para que se guarde en el directorio:
     > $ git push –u origin main
  5. Se subio los archivos y se actualizó nuestro repositorio en GitHug, podemos verificar con commits realizados:
     > $ git log