



Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES



Laboratorio 1:

Administración de Sistemas Linux

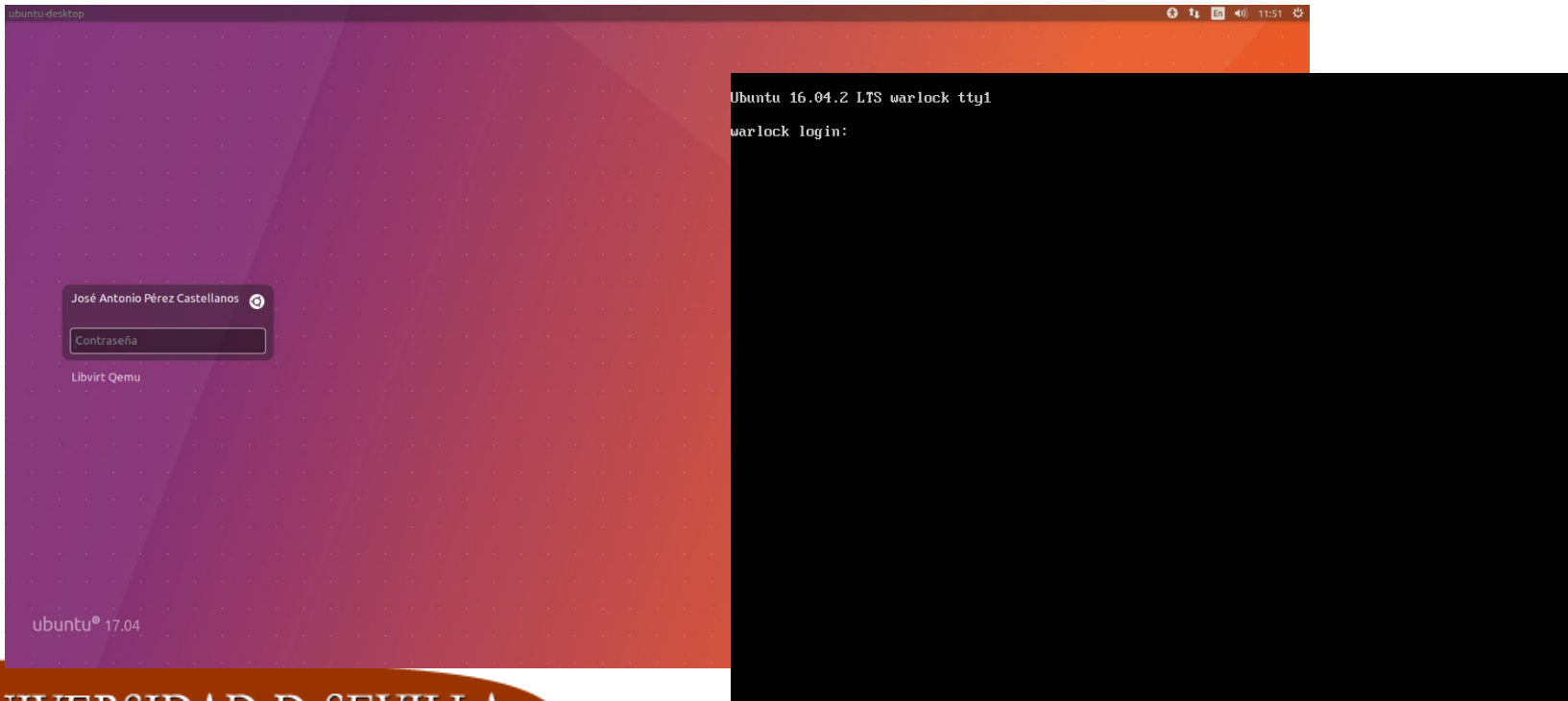
1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Primeros pasos

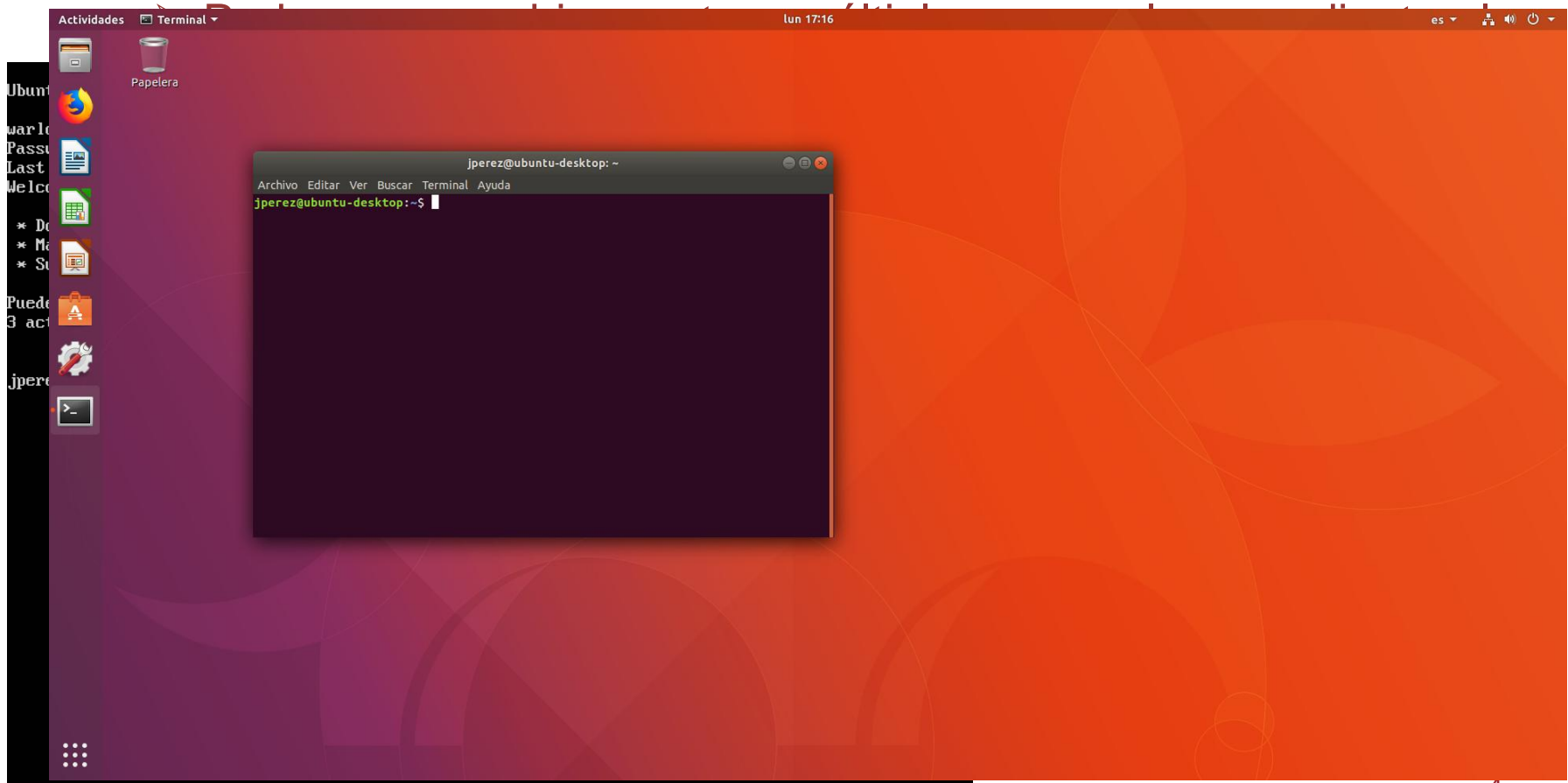
❖ Tipos de distribuciones Linux:

- Distribuciones de escritorio (Desktop)
 - Disponen de Interfaz Gráfica de Usuario
- Distribuciones para servidores (server)
 - Frecuentemente no disponen de Interfaz Gráfica de Usuario



Primeros pasos

- ❖ Distribuciones sin Interfaz Gráfica de Usuario:
 - Tras iniciar sesión entramos en consola de comandos



Primeros pasos

❖ Caracteres de control:

- CTRL-C: Aborta la ejecución del proceso en curso
- CTRL-D: Fin de entrada
- CTRL-Z: *Suspende* la ejecución del proceso en curso (se tratará en epígrafe “Procesos y Servicios”)

❖ Nombres de archivos y directorios (o carpetas):

- Se pueden usar nombres largos (no conviene abusar)
- Los caracteres '/', '-', '<', '>', '|', '&', '[', ']', '?', '*', '\$', '#', y '/' tienen significado especial. Se recomienda no usar.
- Los archivos y directorios ocultos, cuyo nombre comienza por '.', no aparecen por defecto al mostrar el contenido del directorio que los contienen
- Se distinguen mayúsculas de minúsculas

Primeros pasos

❖ Trayectos de directorios (o carpetas):

- Ubicación del directorio dentro de un sistema de archivos.
- Ejemplo de trayecto de un directorio llamado archivos: **/home/usuario/archivos**
- En todos los directorios de Linux:
 - '.': representa el directorio actual
 - '..': representa el directorio padre del actual
- **Trayecto absoluto** de un directorio o archivo: especificación de la ubicación del directorio o archivo en cuestión, indicando todos los directorios partiendo del directorio raíz (/) del sistema de archivos. Para el directorio anterior sería: **/home/usuario/archivos**.
- **Trayecto relativo** de un directorio o archivo: especificación de la ubicación del directorio o archivo en cuestión, relativa al directorio actual (.) o al directorio padre del actual (..).
 - Si por, por ejemplo, nuestro directorio actual es **/home**, podemos especificar el trayecto relativo a archivos mediante **./usuario/archivos**. En estos casos se puede omitir los caracteres **./** e indicar el trayecto de la siguiente forma: **usuario/archivos**).
 - La mayoría de comandos (no todos, puesto que existen excepciones) se pueden aplicar a los archivos del directorio actual indicando sólo su nombre. Ej.: **cat fichero1** sería lo mismo que **cat /home/usuario/archivos/fichero1** o **cat ./fichero1**.
 - Si por, por ejemplo, nuestro directorio actual es **/home/usuario/otro**, podemos especificar el trayecto relativo a archivos mediante **../archivos** (en estos casos no podemos omitir **../**, lo mismo pasaría para en el caso de ejecutar comandos con ficheros).

Primeros pasos

- ❖ Caracteres comodines para nombres de archivos:
 - '*' : representa cualquier secuencia de 0 o más caracteres
 - '?': representa a un único carácter
 - '[' *conjunto de caracteres* ']': representa a cualquier carácter del conjunto
 - El conjunto se puede expresar también por comprensión: '[' c₁ '-' c₂ ']'
 - '!': dentro de los corchetes representa negación. Es decir cualquier carácter que NO esté en el conjunto
- ❖ Cuando se usan comodines en un comando:
 - Si hay archivos que cumplen el patrón:
 - comando recibe sustitución ya realizada
 - Si no hay archivos que cumplen el patrón:
 - comando recibe el patrón literalmente, sin sustitución.
 - Para evitar sustitución: entrecomillar cadena con comodines

Primeros pasos

- ❖ Creación de variables: basta asignarles un valor:

variable="valor de la variable"

- Todas las variables son de tipo cadena
- 'Para usar el valor de una variable, se le antepone '\$' al nombre:

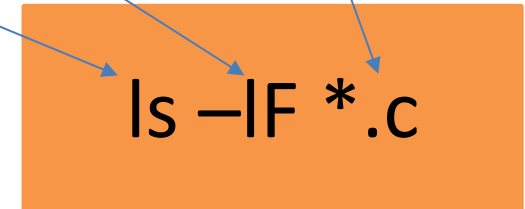
echo \$HOME

- ❖ Algunas variables predefinidas:

- HOME: directorio raíz del usuario
- LOGNAME, USER: nombre de usuario
- OLDPWD: anterior directorio de trabajo
- PATH: lista de directorios en que se buscan los ejecutables
- PWD: directorio de trabajo actual
- SHELL: intérprete de comandos en uso

Primeros pasos

- ❖ Formato de los comandos: **comando opciones argumentos**
- ❖ Varios comandos en una línea:
 - $c_1 \ \&\& \ c_2$: c_2 se ejecuta si c_1 no falla
 - $c_1 \ || \ c_2$: c_2 se ejecuta si falla c_1
 - $c_1 \ ; \ c_2$: c_2 se ejecuta falle o no falle c_1
- ❖ Cada comando se ejecuta como un nuevo **proceso**
- ❖ (Casi) todos los comandos que veremos son **externos**
 - Se corresponden con un programa ejecutable
 - Para localizar el ejecutable de un comando: **which comando**
- ❖ Variable **PATH**: lista los directorios en que se busca los ejecutables (separados por “:”)
 - No se buscan los ejecutables en directorio actual
 - ¡Y además directorio actual (.) no se debe añadir a PATH!
 - Para ejecutar comando en directorio actual: **./comando**



```
ls -lF *.c
```

Primeros pasos

- ❖ Todos los comandos que vamos a ver...
 - Muestran sus resultados en la *salida estándar* (descriptor 0)
 - Leen su entrada de la *entrada estándar* (descriptor 1)
 - Muestran errores en la *salida estándar de error* (descriptor 2)
- ❖ Por defecto:
 - Salida estándar: dirigida a la consola/ventana
 - Entrada estándar: dirigida de teclado asociado a la consola
 - Salida estándar de error: dirigida a la consola/ventana
- ❖ Redirección de la entrada estándar: operador “<”
 - comando < fuente
 - Ejecuta el comando, tomando su entrada desde el archivo “fuente”

Primeros pasos

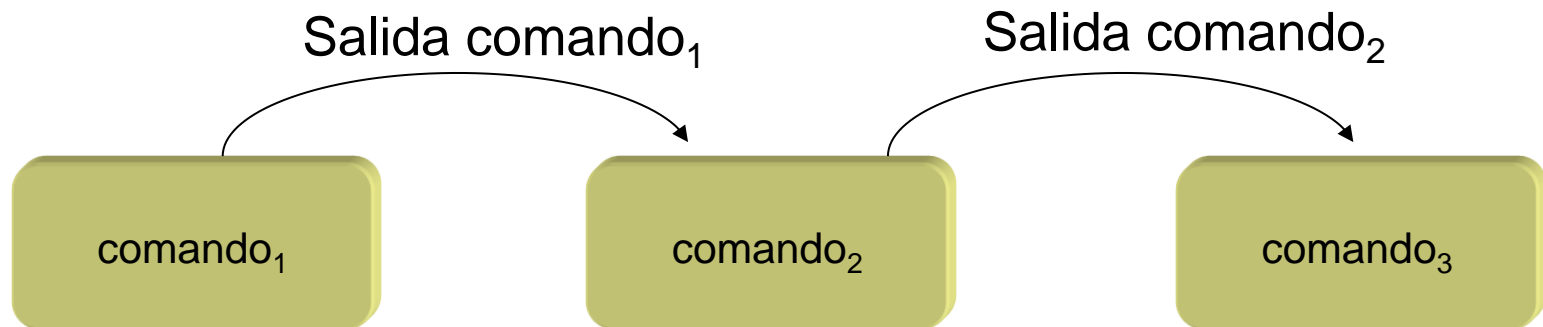
- ❖ Redirección de la salida estándar: operadores “>” y “>>”
 - comando > salida
 - Ejecuta el comando, generando su salida sobre el archivo “salida”
 - comando >> salida
 - Ejecuta el comando, añadiendo su salida al archivo “salida”
- ❖ Redirección de salida de error: operadores “2>” y “2>>”
 - comando 2> errores
 - Ejecuta el comando, escribiendo los errores (si los hay) en el archivo “errores”. Análogamente, el operador “2>>” añade los errores al final del archivo.
 - **/dev/null**: archivo *nulo*. Útil cuando se desea ignorar los errores.
- ❖ Sustitución de un comando por su salida:
\$(comando)
 - Principales usos:
 - Asignación del resultado de un comando a una variable
 - Uso del resultado de un comando como un argumento para otro comando

Primeros pasos

- ❖ Tubería: redirección de la salida de un proceso a la entrada de otro mediante operador “|”

$\text{comando}_1 | \text{comando}_2 | \dots | \text{comando}_n$

- Ejecuta simultáneamente todos los comandos (cada uno es un proceso distinto) redireccionando la salida de cada comando a la entrada del siguiente



Primeros pasos

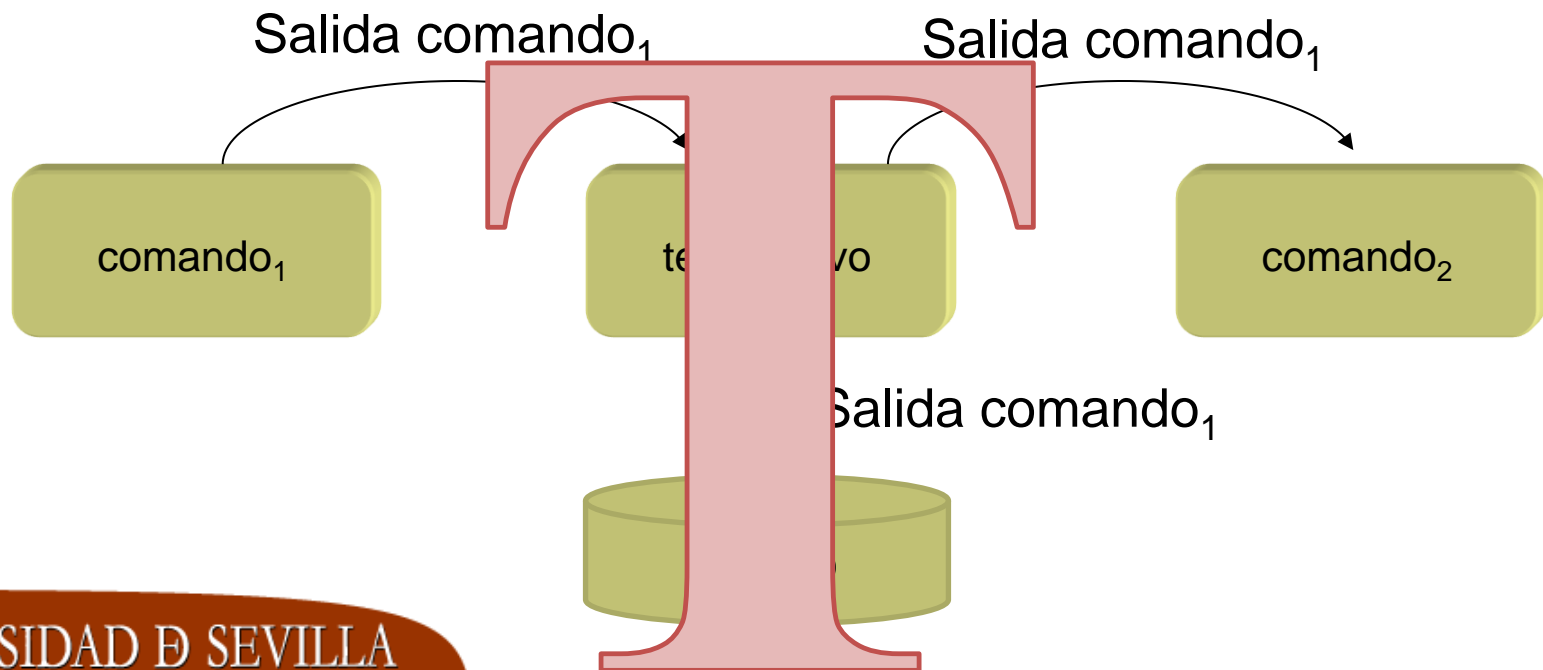
❖ Comando **tee** archivo:

➤ Envía su entrada estándar a su salida estándar, y al archivo que se pasa como argumento

- Opción `-a`: en lugar de crear un nuevo archivo, añade a uno existente

➤ Sirve para hacer una “T” en una tubería:

`comando1 | tee [-a] archivo | comando2`



Primeros pasos

Principales comandos para...

- ❖ Acceso a la documentación: **man**
- ❖ Manejo de directorios y archivos: **ls, mkdir, cd, cat, less, sort, file, rmdir, cp, rm, mv**
- ❖ Buscar información: **find, grep**
- ❖ Descargar archivos: **curl, wget**
- ❖ Empaquetar y comprimir: **tar**
- ❖ Edición de texto (**vi, joe, nano, gedit...**)

Primeros pasos

Comando: **man**

❖ **Propósito:** acceso a la documentación del sistema

❖ **Formato:**

`man [sección] materia`

`man -k clave`

- *materia* es el elemento sobre el que se solicita información
- *sección* (opcional) es la sección en la que se desea buscar
 - Secciones 1, 6 y 8: comandos del sistema
 - Secciones 2 y 3: llamadas al sistema (API)
 - Secciones 4, 5 y 7: temas relacionados con archivos
- *-k clave*: busca todos los temas que contienen la palabra clave

Primeros pasos

Comando: **ls**

❖ **Propósito:** lista un conjunto de archivos, el contenido de un árbol de directorios o cualquier combinación de lo anterior

❖ **Formato:**

`ls [opciones] [archivos]`

➤ *archivos* son los archivos o directorios a mostrar. Si no se especifica ninguno, muestra el contenido del directorio actual

➤ Opciones habituales:

- `-l`: muestra la información en formato detallado (largo)
- `-a`: lista todos los archivos, incluyendo los que empiezan por `.'`
- `-R`: lista los subdirectorios, de forma recursiva
- `-t`: ordena cronológicamente, primero los más recientemente modificados
- `-h`: Muestra tamaño de archivos en unidades *humanas* (KB, MB, GB, TB...)
- `-d`: Lista la información del propio directorio, en lugar de los archivos que contiene.
- `-F`: Agrega un carácter especial al final de cada nombre de archivo o directorio para indicar su tipo: `/` para directorios, `*` para ejecutables y `@` para enlaces simbólicos.

Primeros pasos

Comando: **mkdir**

❖ **Propósito:** crea directorios

❖ **Formato:**

`mkdir [opciones] directorio1 [directorio2... directorion]`

➤ *directorio_i*: directorio o directorios que se crean

➤ Opciones habituales:

- *-p*: No genera error si el directorio ya existe. Además, si alguno de los directorios en la ruta no existen, lo crea también. Útil para crear árboles de directorios.

Comando: **cd**

❖ **Propósito:** cambia el directorio por defecto

❖ **Formato:**

`cd directorio`

➤ *directorio*: trayecto absoluto o relativo a nuevo directorio

Primeros pasos

Comando: **rmdir**

❖ **Propósito:** elimina directorios

❖ **Formato:**

`rmdir directorio1 [directorio2... directorion]`

➤ *directorio_i*: directorios que se eliminan. ¡Deben estar vacíos!

Comando: **cat**

❖ **Propósito:** muestra el contenido de archivos de texto

❖ **Formato:**

`cat [opciones] archivo1 [... archivon]`

➤ *archivo_i*: archivos que se muestran

➤ Opciones habituales:

- *-b*: Enumera las líneas no en blanco
- *-n*: Enumera todas las líneas, incluidas las que están en blanco

Primeros pasos

Comando: **less**

❖ **Propósito:** comando interactivo que muestra un archivo página a página

❖ **Formato:**

`less [opciones] [archivo1 ... archivon]`

➤ *archivo_i*: archivo o archivos de texto que se muestran. Si se especifica más de uno, los muestra uno tras otro.

➤ Para salir del comando interactivo: tecla **q**.

Primeros pasos

Comando: **sort**

❖ **Propósito:** muestra el resultado de ordenar línea a línea un archivo, fusionar en orden varios archivos, o comprueba si un archivo está ordenado

❖ **Formato:**

`sort [opciones] [archivo1 ... archivon]`

➤ *archivo_i*: archivo o archivos de texto que se ordenan. Si se especifica más de uno, los fusiona ordenadamente línea a línea

➤ Opciones habituales:

- `-c`: comprueba si los archivos de entrada están todos ordenados
- `-m`: fusiona línea a línea los archivos de entrada, sin aplicar ordenación
- `-r`: ordena en orden inverso
- `-u`: elimina duplicados

Primeros pasos

Comando: **file**

- ❖ **Propósito:** Determina y muestra el tipo de los archivos
- ❖ **Formato:**

file [opciones] archivo₁ [... archivo_n]

➤ *archivo_i*: archivos que se analizan

➤ Opciones habituales:

- *-Z*: Analiza también el contenido de archivos comprimidos
- *--help*: Muestra todas las opciones soportadas

Primeros pasos

Comando: **cp**

❖ **Propósito:** copia archivos, directorios o un conjunto de los anteriores

❖ **Formato:**

`cp [opciones] origen1 [origen2 ... origenn] destino`

➤ *origen_i*: son los ficheros, conjuntos de ficheros especificados mediante comodines, o directorios que se copian o mueven.

Cuando se copian múltiples ficheros, el destino debe ser obligatoriamente un directorio

➤ *destino*: archivo o directorio destino al que se copia

➤ Opciones habituales:

- *-r*: copia archivos recursivamente
- *-u*: no sobrescribir archivos con el mismo nombre que sean más recientes
- *-f / -i*: No avisar o avisar (respectivamente) si se sobrescriben archivos

Primeros pasos

Comando: **rm**

❖ **Propósito:** Elimina archivos o directorios

❖ **Formato:**

`rm [opciones] archivo1 [archivo2 ... archivon]`

➤ *archivo_i*: archivo o archivos que se eliminan

➤ Opciones habituales:

- `-r`: borrar directorios recursivamente
- `-f`: no pedir confirmación de borrado en ningún caso
- `-v`: informar de los borrados que va realizando

Primeros pasos

Comando: **mv**

❖ **Propósito:** mueve archivos, directorios o un conjunto de los anteriores

❖ **Formato:**

`mv [opciones] origen1 [origen2 ... origenn] destino`

➤ *origen_i*: son los ficheros, conjuntos de ficheros especificados mediante comodines, o directorios que se copian o mueven.

Cuando se mueven múltiples ficheros, el destino debe ser obligatoriamente un directorio

➤ *destino*: archivo o directorio destino al que se mueve

➤ Opciones habituales:

- *-u*: no sobrescribir archivos con el mismo nombre que sean más recientes
- *-f / -i*: No avisar o avisar (respectivamente) si se sobrescriben archivos

Primeros pasos

Comando: **find**

❖ **Propósito:** busca archivos a partir de un directorio inicial. Para cada archivo, se evalúa una lista de expresiones llegar a una que no se cumpla o terminarla.

❖ **Formato:**

`find [opciones] inicio expresión1 [... expresiónn]`

➤ *inicio*: Directorio en el que comienza la búsqueda recursiva

➤ *expresión_i*: Expresiones que se evalúan para cada archivo.

Tipos de expresiones:

- **tests**: condiciones que evalúan a true o false dependiendo de alguna propiedad de cada archivo
- **acciones**: realizan algún tipo de acción con cada archivo que pasa el test. También evalúan a true o false.
- **operadores**: Utilizados para componer condiciones compuestas

Primeros pasos

Comando: **find (cont.)**

❖ Algunos tests:

- **-name patrón**: true si nombre de archivo encaja en patrón (¡no olvidar comillas!)
- **-amin n/+n/-n**: true si archivo se accedió hace n minutos (o más de n, o menos de n)
- **-mmin n/+n/-n**: true si archivo se modificó hace n minutos (o más de n, o menos de n)
- **-empty**: true si archivo (o directorio) está vacío
- **-size n/+n/-n [ckMG]**: true si el tamaño (redondeado al alza) es igual, mayor o menor que n bytes, kilobytes, megabytes o gibabytes
- **-user usuario**: true si el archivo es propiedad del usuario cuyo nombre o UID se especifica

Primeros pasos

Comando: **find (cont.)**

❖ Algunas acciones:

- **-print**: acción por defecto; muestra trayecto completo a archivo y evalúa a true.
- **-delete**: elimina el archivo y evalúa a true si borrado tiene éxito, o false en caso contrario.
- **-exec comando / -execdir comando**: Ejecuta un comando arbitrario.
 - Para hacer referencia en el comando al archivo encontrado usamos "{}".
 - Se debe terminar el comando con "\;"
- **quit**: aborta el comando find

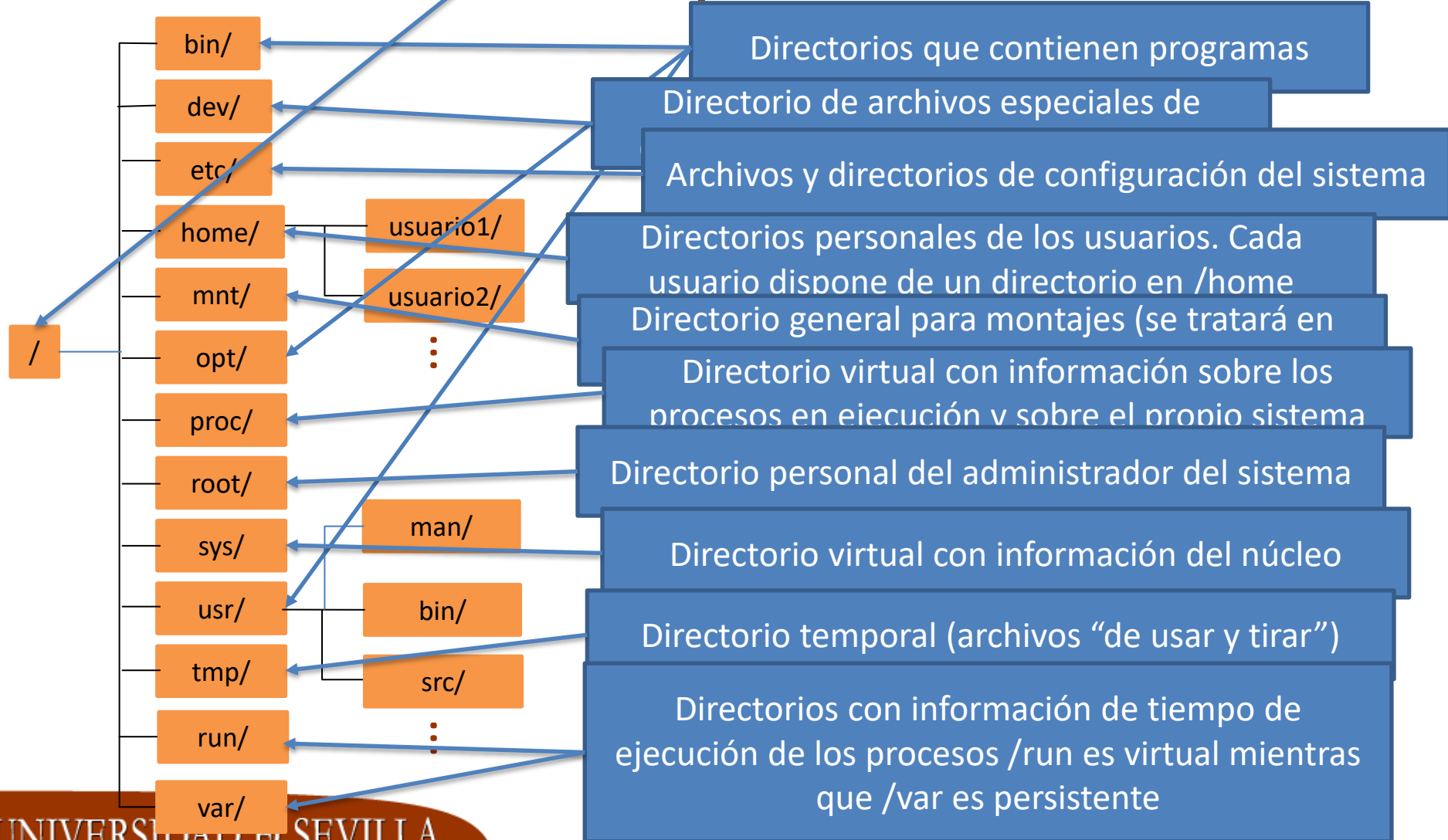
❖ Operadores:

- expresión₁ **-a** expresión₂: Operación lógica **Y** (por defecto, se puede omitir **-a**)
- expresión₁ **-o** expresión₂: Operación lógica **O**
- (expresión): Fuerza orden de precedencia
- **!expresión**: Operación lógica **NO**

Primeros pasos

Directorio RAÍZ

Estructura de directorios típica:



Primeros pasos

Directorio `/var/log`:

- ❖ Contiene archivos de log del sistema y de aplicaciones
- ❖ Para limitar tamaño de los archivos: se van creando archivos nuevos y renombrando los antiguos con n^o orden
- ❖ Principales archivos de log del sistema (puede variar entre distribuciones y versiones):
 - **syslog**: registro general de todos los eventos del sistema
 - **messages**, **boot.log**: registros de arranque del sistema y de arranque de los distintos servicios instalados
 - **dmesg**: información generada por el núcleo relacionada con el hardware
 - **auth.log**: Autenticaciones (exitosas y fallidas) efectuadas

Primeros pasos

Archivos `/etc/machine-id`, `/var/lib/dbus/machine-id`:

- ❖ Contienen identificador único del sistema
 - Se crea al realizar la instalación de sistema operativo
- ❖ Ambos archivos pueden ser enlaces a un mismo archivo
- ❖ Para generar un nuevo identificador de máquina:
 - Método 1:
 - Borrar **contenido** de los archivos (o del archivo, si es el mismo)
 - ¡No borrar los archivos! (en dicho caso, no se crea machine-id)
 - Se genera nuevo machine-id en el siguiente arranque
 - Método 2:
 - Eliminar archivo `/var/lib/dbus/machine-id`
 - Crear nuevo identificador: **dbus-uuidgen --ensure**
 - Crea nuevo identificador en `/var/lib/dbus/machine-id`
 - Copiar a (o preferiblemente, generar enlace) `/etc/machine-id`
 - ¡Se requiere derechos de administración! (se ven en epígrafe [2](#))

Primeros pasos

Comando: **grep**

- ❖ **Propósito:** muestra las líneas que cumplen un patrón determinado de un archivo o conjunto de archivos
- ❖ **Formato:**
grep [opciones] patrón [archivo₁ ... archivo_n]
 - *archivo_i*: archivo o archivos de texto cuyos finales se muestran.
 - Opciones habituales:
 - *-i*: no distingue mayúsculas de minúsculas
 - *-l*: muestra sólo los nombres de los archivos que contienen alguna ocurrencia del patrón
 - *-v*: muestra las líneas que no contienen el patrón
 - *-h*: suprime la cabecera con el nombre del archivo cuando se procesan múltiples archivos

Primeros pasos

Comando: **grep** (cont.)

grep [opciones] patrón [archivo₁ ... archivo_n]

➤ Patrón: expresión regular construida mediante caracteres literales y alguno de los metacaracteres siguientes:

- ‘.’: representa a cualquier carácter
- ‘[*lista de caracteres*]’ o ‘[*a-b*]’: representa a cualquier carácter de la lista, o a cualquier carácter comprendido entre *a* y *b*. Si el primer carácter tras el corchete es ^, el significado se invierte.
- ‘?’: pospuesto a una expresión, representa que la expresión es opcional
- ‘*’: pospuesto a una expresión, representa a **0 a n** ocurrencias de la misma.
- ‘+’: como anterior, pero representa **1 a n** ocurrencias
- {*n*}: pospuesto a una expresión representa **n** ocurrencias de la misma
- ‘^’: antepuesto a una expresión, significa que la expresión debe ocurrir al comienzo de la línea
- ‘\$’: pospuesto a una expresión, significa que la expresión debe ocurrir al final de la línea
- ‘\’: antepuesto a cualquier carácter, toma su valor literal (útil si la expresión contiene algún carácter que coincida con un metacarácter)

Primeros pasos

Comando: **curl**

❖ **Propósito:** Transfiere una URL

❖ **Formato:**

`curl [opciones] url`

➤ *url*: URL que se transfiere. La sintaxis depende de protocolo (http, https, ftp, ftps, gopher, etc.)

➤ Opciones habituales:

- *-L, --location*: Si el servidor indica que la URL se ha movido a otra ubicación, seguir dicha ubicación automáticamente
- *-o archivo, --output archivo*: Escribir resultado en archivo, en lugar de salida estándar
- *-O, --remote-name*: Escribir resultado en un archivo con el mismo nombre que el archivo que se obtiene en la petición
- *-s, --silent*: Modo silencioso. No mostrar nada en salida estándar.

❖ Alternativa: **wget**

Primeros pasos

Comando: **tar**

❖ **Propósito:** Utilidad para empaquetado de archivos.

❖ **Formato:**

`tar [opciones] [archivo1 ... archivon]`

➤ *archivo.tar*: Nombre del archivo en/del que se (des)empaquetan archivos.

➤ *archivo_i*: Nombre(s) de el/los archivo(s) o directorio(s) que se empaquetan

➤ Opciones habituales:

- *-f archivo.tar*: Nombre del archivo del/al que se (des)empaqueta
- *-c*: Crea el archivo dado en la opción *-f* (uso: empaquetar)
- *-a*: Añade contenido al archivo dado en la opción *-f* (uso: empaquetar)
- *-x*: Extrae contenido del archivo dado en la opción *-f* (uso: desempaquetar)
- *-z*: Usa compresión (uso: empaquetar y desempaquetar)
- *-v*: Muestra las acciones que realiza

Primeros pasos

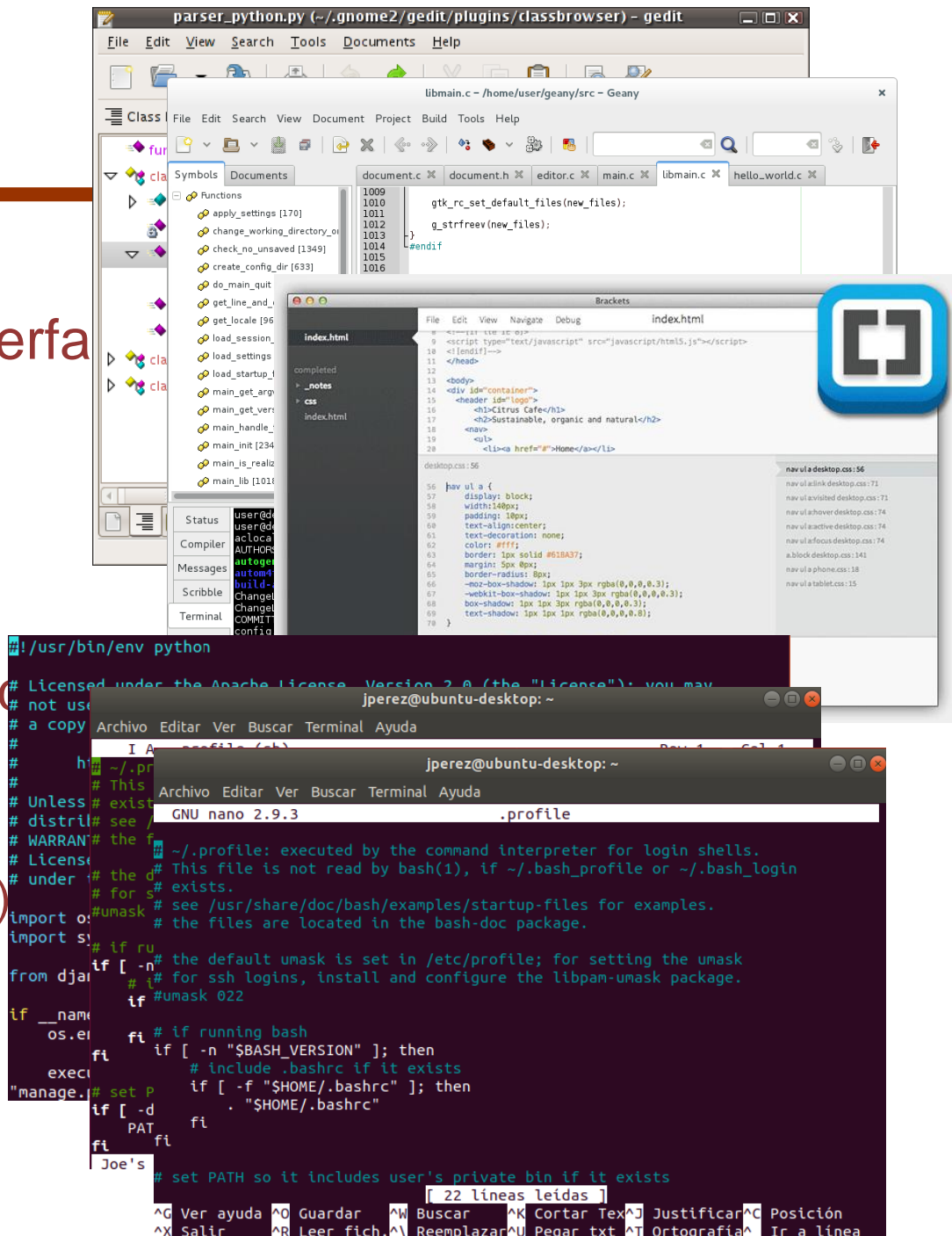
Comandos de edición:

❖ Distribuciones con Interfa

- gedit
- geany
- brackets
- ...

❖ En consola de comando

- vi (único estándar)
- joe
- nano (el más utilizado)
- ...





Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Usuarios, grupos y administradores

Usuario: toda entidad (persona o sistema) autorizada a ejecutar procesos en el sistema

- ❖ Todo usuario viene determinado por un UID numérico
- ❖ Usuario “de sistema”: no puede iniciar sesión
 - Se utiliza para aplicaciones que se ejecutan como propiedad de este usuario
- ❖ Relación de usuarios del sistema: archivo **/etc/passwd**
 - Nombre de usuario (login)
 - La contraseña (“x”) se almacena encriptada en **/etc/shadow**
 - UID y GID de grupo primario
 - Nombre y datos personales
 - Directorio personal (*home*)
 - Intérprete de comandos (*shell*) que utiliza si inicia sesión

Usuarios, grupos y administradores

Grupo: los usuarios pueden pertenecer a grupos, lo cual facilita la gestión de los recursos a los que tienen derecho a acceder

❖ Cada usuario pertenece a:

- Un grupo primario, que se establece en el momento de su creación
 - Si al crear el usuario no se indica un grupo primario, se crea un grupo con el mismo nombre que el usuario
- (opcional) otros grupos a los que se añade posteriormente

❖ Relación de grupos del sistema: archivo **/etc/group**

- Nombre del grupo
- Contraseña de grupo (no se utiliza, ¡inherentemente insegura!)
- GID
- Lista de usuarios del grupo

Usuarios, grupos y administradores

Comando: **who**

❖ **Propósito:** lista los usuarios actualmente conectados al sistema

❖ **Formato:**

who [opciones]

➤ Opciones habituales:

- *-m*: muestra información sobre el propio usuario (equivalente a *whoami*)
- *-q*: proporciona sólo el número de usuarios conectados
- *-u*: proporciona también el tiempo que lleva inactivo.
 - Un punto '.' significa que está activo en el último minuto
 - 'old' significa que lleva más de 24 horas inactivo

Usuarios, grupos y administradores

Comando: **adduser**

❖ **Propósito:** crea un nuevo usuario. Puede solicitar su contraseña e información adicional.

❖ **Formato:**

`adduser [opciones] usuario`

➤ *usuario*: nombre de *login* del usuario que se crea

➤ Opciones habituales:

- *--system*: Crea un *usuario sistema*. El grupo predeterminado de los usuarios de sistema es *nogroup*
- *--home directorio*: Usa *directorio* como directorio *home* para el usuario
- *--shell*: usa la aplicación especificada como intérprete de comandos para el usuario
- *--ingroup grupo*: añade al usuario al grupo *grupo* como su grupo primario en lugar de al predeterminado (uno con el mismo nombre que el nombre de login)

Usuarios, grupos y administradores

Comando: **adduser**

- ❖ **Propósito:** Añade un usuario ya existente a un grupo de usuarios (de forma adicional a su grupo primario)
- ❖ **Formato:**
 - adduser usuario grupo
 - *usuario*: nombre de *login* del usuario que se añade al grupo
 - *grupo*: grupo al que se añade el usuario

Usuarios, grupos y administradores

Comando: **addgroup**

❖ **Propósito:** Crea un grupo de usuarios

❖ **Formato:**

`addgroup [opciones] grupo`

`adduser --group [opciones] grupo`

➤ *grupo*: nombre del grupo que se crea

➤ Opciones habituales:

- `--system`: Crea un grupo sistema

Usuarios, grupos y administradores

Comando: **deluser**

❖ **Propósito:** Elimina un usuario

❖ **Formato:**

deluser [opciones] usuario

➤ *usuario*: nombre de login del usuario que se elimina

➤ Opciones habituales:

- *--remove-home*: Elimina el directorio de trabajo del usuario
- *--remove-all-files*: Elimina todos los archivos del usuario en el sistema (incluido el directorio de trabajo)
- *--backup-to*: indica la ubicación en la que se realiza una copia de seguridad de los archivos borrados

Usuarios, grupos y administradores

Comando: **deluser**

❖ **Propósito:** Elimina un usuario de un grupo

❖ **Formato:**

deluser [opciones] usuario grupo

- *usuario*: nombre de login del usuario que se elimina del grupo
- *grupo*: nombre del grupo del que es eliminado

Usuarios, grupos y administradores

Comando: **delgroup**

Propósito: Elimina un grupo de usuarios (¡el grupo, no los usuarios!). No se puede eliminar un grupo si es grupo primario de algún usuario existente.

❖ **Formato:**

delgroup [opciones] grupo

deluser -group [opciones] grupo

➤ *grupo*: nombre del grupo que se elimina

➤ Opción:

- *--only-if-empty*: el grupo sólo se elimina si no tiene ningún usuario

Usuarios, grupos y administradores

❖ Derechos de acceso a archivos y directorios:

- cada archivo y directorio tiene un propietario y un grupo (por defecto, es el grupo del propietario)
- Cada archivo y directorio tiene tres grupos de bits de derechos...

A terminal window titled 'jperez@ubuntu-desktop: ~' shows the command `ls -lF archivo-ejemplo` and its output: `-rwxrwxrwx 1 jperez jperez 8980 Jan 18 12:09 archivo-ejemplo*`. Three orange callout boxes point to the permission string: the first points to the first 'r' and is labeled 'Derechos del Propietario (user)'; the second points to the first 'w' and is labeled 'Derechos de los que pertenecen al grupo del archivo (group)'; the third points to the last 'x' and is labeled 'Derechos del resto de usuarios (other)'.

```
jperez@ubuntu-desktop: ~
Archivo Editar Ver Buscar Terminal Ayuda
jperez@ubuntu-desktop:~$ ls -lF archivo-ejemplo
-rwxrwxrwx 1 jperez jperez 8980 Jan 18 12:09 archivo-ejemplo*
jperez@ubuntu-desktop:~$
```

Derechos del Propietario (user)

Derechos de los que pertenecen al grupo del archivo (group)

Derechos del resto de usuarios (other)

Usuarios, grupos y administradores

❖ Cada bit significa que se concede derecho para...

Bit	En archivos	En directorios
r	Leer el archivo	Listar contenido de directorio
w	Escribir en el archivo	Crear, renombrar o desenlazar archivos en directorio
x	Ejecutar archivo	<i>Pasar a través de directorio</i>

Además: Sticky bit (t)

❖ Sólo se aplica a directorios:

❖ Significado: Si está activo, sólo se puede renombrar o desenlazar un archivo del directorio si:

- Se tiene derecho de escritura sobre el directorio, **y además**
- Se tiene derecho de escritura sobre el propio archivo

Usuarios, grupos y administradores

Y además: bit **SUID** / **SGID** (s)

- ❖ Sólo se aplica a archivos ejecutables (deben tener derecho “x” al menos para “user”)
- ❖ Significado SUID:
 - Si está activo, cuando el programa lo hará con el UID del propietario de archivo (en lugar de UID del proceso padre)
 - Este bit se aplica a “user”, y se muestra en el lugar del bit “x”
- ❖ Significado SGID:
 - Si está activo, cuando el programa lo hará con el GID del archivo (en lugar de GID del proceso padre)
 - Este bit se aplica a “group”, y se muestra en el lugar del bit “x”



Usuarios, grupos y administradores

Comando: **chmod**

❖ **Propósito:** Cambia los derechos de archivos o directorios

❖ **Formato:**

`chmod [-R] [u|g|o|a] [+|-|=] [s|t|r|w|x] archivo1 [archivo2 ... archivon]`

- *archivo_i*: archivos o directorios cuyos derechos se modifican
- *-R*: procesa recursivamente archivos en subdirectorios
- *u, g, o, a*: representan a propietario (**u**ser), otros usuarios del grupo del propietario (**g**roup), resto de usuarios (**o**ther), o todos los anteriores (**a**ll)
- *'+'* concede, *'-'* elimina y *'='* establece los derechos que siguen
- *s, t, r, w, x*: derechos que se establecen o eliminan
- Se pueden establecer o eliminar varios grupos de derechos, separados por comas

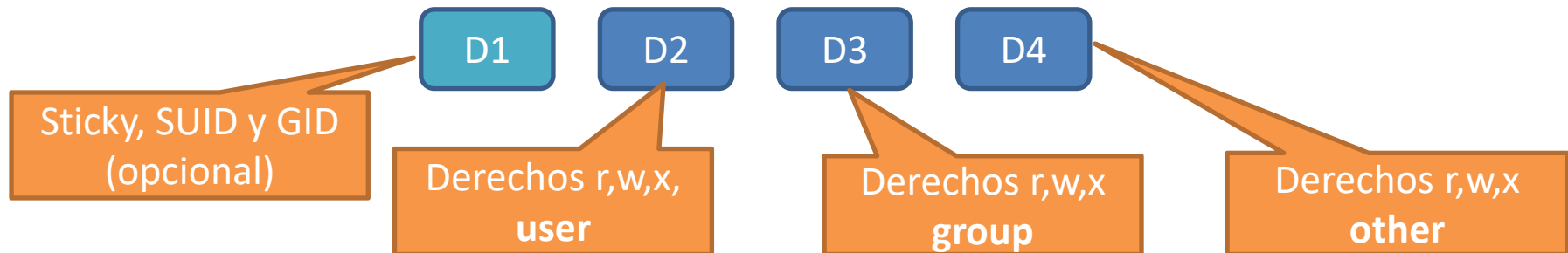
Usuarios, grupos y administradores

Comando: **chmod** (cont.)

❖ Formato alternativo:

`chmod [-R] código-octal archivo1 [archivo2 ... archivon]`

➤ *código-octal*: código de tres o cuatro dígitos, representando:



➤ Cálculo D1 (opcional): Suma de los valores de cada derecho:

- **Sticky:** 1, **SGID:** 2, **SUID:** 4

➤ Cálculo D2, D3 y D4: Suma de los valores de cada derecho:

- **x:** 1, **w:** 2, **r:** 4

➤ Ejemplos: **chmod 640 f1** equivale **chmod u=rw,g=r,o= f1**

chmod 6754 f1 equivale **chmod u=rwx,g=rxs,o=r f1**

Usuarios, grupos y administradores

Comando: **chown**

❖ **Propósito:** Cambia el propietario (y opcionalmente el grupo) de un archivo. Este comando requiere derechos de administración.

❖ **Formato:**

`chown [-R] usuario archivo1 [archivo2 ... archivon]`

- *archivo_i*: archivos o directorios cuyo propietario se cambian
- *-R*: procesa recursivamente archivos en subdirectorios
- *usuario*: nombre de login o UID del nuevo propietario del archivo.
- *Nota*: se puede usar la sintaxis *usuario:grupo* para cambiar simultáneamente usuario y grupo

Usuarios, grupos y administradores

Comando: **chgrp**

- ❖ **Propósito:** Cambia el grupo de un archivo. Este comando requiere derechos de administración.
- ❖ **Formato:**
`chgrp [-R] grupo archivo1 [archivo2 ... archivon]`
 - *archivo_i*: archivos o directorios cuyo grupo se cambian
 - *-R*: procesa recursivamente archivos en subdirectorios
 - *grupo*: nombre del nuevo grupo propietario del archivo

Usuarios, grupos y administradores

❖ Cada usuario solo puede modificar los derechos de **sus** archivos

❖ Usuario “**root**”: administrador del sistema

- Puede modificar los derechos de cualquier archivo
- Tiene atribuciones de administrador:
 - Instalar/desinstalar software
 - Crear o eliminar grupos y usuarios
 - ...



❖ Frecuentemente, por seguridad: **DESHABILITADO**

- La responsabilidad de administración recae en otro(s) usuario(s) autorizados para ello

❖ Forma segura de ejecutar comandos de administración: comando **sudo**

Usuarios, grupos y administradores

Comando: **sudo**

- ❖ **Propósito:** Ejecuta un comando con privilegios de administrador. La primera vez que se usa durante una sesión confirma la contraseña del usuario
- ❖ **Formato:**
 - sudo comando
 - *comando*: comando que se ejecuta con privilegio de administrador
 - **sudo** solo puede ser ejecutado por los usuarios autorizados por un administrador

Usuarios, grupos y administradores

Cómo



H



A

```
jperez@ubuntu-desktop: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.8.6 Archivo: /etc/sudoers.tmp

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include /etc/sudoers.d

```

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea

Usuarios, grupos y administradores

Comando: **su**

- ❖ **Propósito:** Ejecuta un nuevo intérprete de comandos bajo la identidad de otro usuario, previa autenticación como el mismo.
- ❖ **Formato:**
su [-]usuario
 - *usuario*: Nombre de login del usuario
- ❖ **NOTAS:**
 - Si se antepone el guión al nombre del usuario, el comando será equivalente a iniciar una sesión como dicho usuario.
 - Si no se antepone el guión al nombre del usuario, se mantendrá el directorio actual de trabajo y las variables de entorno actuales, excepto HOME, SHELL, LOGNAME y USER
 - Si está habilitado usuario root, **su -** es equivalente a **su -root**



Laboratorio 1:

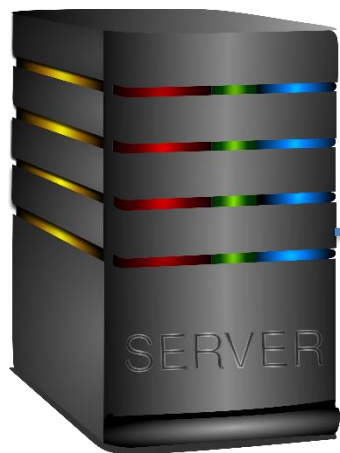
Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Conexión a sistemas remotos

- ❖ Podemos usar un sistema como cliente para conectarnos a un sistema remoto
 - Mediante consola: comando **ssh**
 - Ejecución de aplicaciones con interfaz gráfica de usuario



Conexión de red



Sistema remoto

Sistema cliente

Conexión a sistemas remotos

Comando: **ssh**

❖ **Propósito:** Ejecuta un comando o un intérprete de comandos en un sistema remoto, a través de la conexión de red usando un canal encriptado seguro

❖ **Formato:**

ssh [opciones] [usuario@]host [comando]

➤ *usuario*: nombre de login del usuario del sistema remoto con el que se va a autenticar. Si se omite, se usa el mismo nombre de login que en el de la sesión actual en el sistema cliente.

➤ *host*: nombre de red o dirección IP del sistema remoto

➤ *comando*: Comando que se ejecuta en el sistema remoto. Su salida estándar se muestra en el sistema cliente. Si se omite, inicia una sesión mediante un intérprete de comandos en el sistema remoto.

Conexión a sistemas remotos

- ❖ Si sistema **cliente** tiene interfaz gráfica de usuario: *X11 Forwarding*
 - Se activa mediante opción **-X** (mayúscula, ojo)
 - La aplicación se ejecuta **en servidor**, pero la ventana se muestra en GUI del cliente
- ❖ Protocolo de ssh: utiliza puerto TCP 22
- ❖ **ssh tunneling**: se puede utilizar una conexión ssh como soporte para otros protocolos a nivel superior
- ❖ Primera conexión **ssh** a un sistema remoto: muestra *fingerprint* de sistema remoto y solicita confirmación
- ❖ **ssh** solicita la contraseña por cada conexión

Conexión a sistemas remotos

❖ Alternativa: autenticación mediante pares de claves RSA

ssh-keygen

- Genera archivos **id_rsa** (clave privada) y **id_rsa.pub** (clave pública)
- Pide nombre de archivo y ubicación para **id_rsa** (dejar valor por defecto)
- Pide establecer contraseña para proteger clave privada en sistema cliente

ssh-add

- Añade al agente de autenticación la nueva identidad creada

ssh-copy-id usuario@sistema-remoto

- Envía a sistema remoto la clave pública de la identidad creada

❖ Protocolo de autenticación entre cliente y servidor:

- **Servidor** envía a cliente token encriptado con clave pública
- **Cliente**: si clave privada está protegida con contraseña...
 - Sistema **cliente** pide **la clave que protege la clave privada**
 - Si disponemos del agente de autenticación, sólo la pide una vez
- Cliente usa clave privada para desencriptar token y lo devuelve a servidor

Conexión a sistemas remotos

- ❖ Archivos implicados en cliente:
 - **\$HOME/.ssh/known_hosts**: *fingerprints* de sistemas conocidos
 - **\$HOME/.ssh/id_rsa**: Clave privada (deseable que encriptada)
 - **\$HOME/.ssh/id_rsa_pub**: Clave pública
- ❖ Archivos implicados en servidor:
 - **\$HOME/.ssh/authorized_keys**: lista de identidades recibidas (identidad + su clave pública)
- ❖ Para revocar la autenticación de una identidad:
 - En el **servidor**, eliminar su línea en el archivo **authorized_keys**
 - En cliente, eliminar clave privada de agente de autenticación:
ssh-add -d \$HOME/.ssh/id_rsa

Conexión a sistemas remotos

Comando: **sftp**

- ❖ **Propósito:** Ejecuta un intérprete de comandos especializado en la carga/descarga de archivos a/desde un sistema remoto, a través de la conexión de red usando un canal encriptado seguro
- ❖ **Formato:**
 - sftp usuario@host
 - *usuario*: nombre de login del usuario del sistema remoto con el que se va a autenticar. Si se omite, se usa el mismo nombre de login que en el de la sesión actual en el sistema cliente.
 - *host*: nombre de red o dirección IP del sistema remoto

Conexión a sistemas remotos

Comando: **sftp**

❖ **Comandos de navegación en una sesión sftp:**

- ***pwd/lpwd***: Muestra el directorio actual de la máquina remota/local.
- ***cd/lcd***: Cambia el directorio en la máquina remota/local.
- ***ls/l ls***: Lista los archivos de un directorio la máquina remota/local.
- ***mkdir***: Crea un directorio en la máquina remota (no dispone de versión para la máquina local).
- ***rmdir***: Elimina un directorio vacío en la máquina remota (no dispone de versión para la máquina local).
- ***rm***: Elimina un archivo en la máquina remota (no dispone de versión para la máquina local).
- ***bye* o *exit***: Cierra la sesión.

Conexión a sistemas remotos

Comando: **sftp**

❖ Comandos de carga/descarga en una sesión sftp:

- ***get /trayecto/remoto/archivo [/trayecto/local/archivo]***: Descarga un archivo remoto en la máquina local. Si no se especifica el trayecto de destino, el archivo remoto se descarga en el directorio actual de la máquina local.
- ***mget /trayecto/remoto/archivo1 /trayecto/remoto/archivo2 /trayecto/remoto/archivo3 ... [/trayecto/local]***: Descarga múltiples archivos remotos en la máquina local. Si no se especifica el trayecto de destino, los archivos remotos se descargan en el directorio actual de la máquina local.
- ***put /trayecto/local/archivo [/trayecto/remoto/archivo]***: Carga un archivo local en la máquina remota. Si no se especifica el trayecto de destino, el archivo local se carga en el directorio actual de la máquina remota.
- ***mput /trayecto/local/archivo1 /trayecto/local/archivo2 /trayecto/local/archivo3 ... [/trayecto/remoto]***: Carga múltiples archivos locales en la máquina remota. Si no se especifica el trayecto de destino, los archivos locales se cargan en el directorio actual de la máquina remota.



Laboratorio 1:

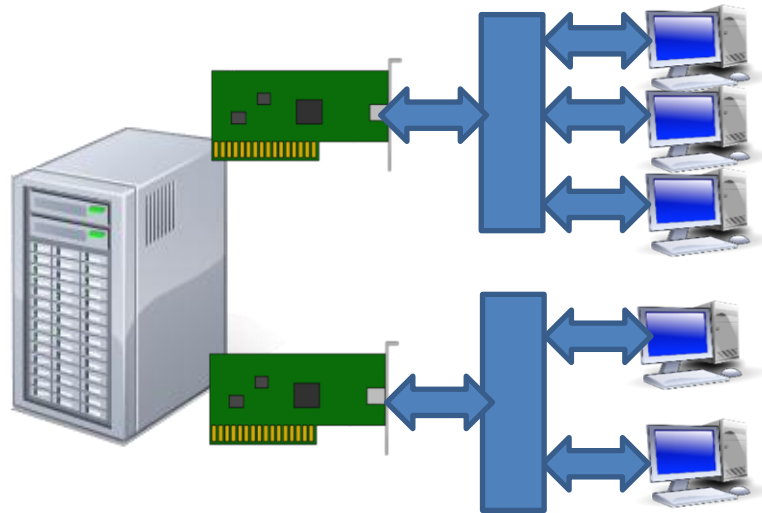
Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Configuración de red

- ❖ Sistema puede disponer de múltiples adaptadores de red
 - Cada adaptador conecta a una red
 - En cada red el sistema se identifica por (al menos) una dirección de red



- ❖ Por cada adaptador existe una *interfaz* de red
 - Existen también interfaces de red lógicas que no se corresponden con un adaptador de red físico
 - Ej: lo (loopback)

Configuración de red

Operaciones más habituales:

- ❖ Listar las interfaces de red existentes
 - **ip [-s] link show [dev *interfaz*] [up]**
 - -s: muestra estadísticas sobre información enviada y recibida
 - *dev interfaz*: muestra solo información sobre la interfaz especificada
 - *up*: muestra sólo interfaces activas
- ❖ Activar o desactivar una interfaz:
 - **ip link set *interfaz* up**
 - **ip link set *interfaz* down**
- ❖ Consultar, añadir o quitar IP a una interfaz
 - **ip addr show [dev *interfaz*]**
 - **ip addr add *dirección-ip/n* dev *interfaz***
 - **ip addr del *dirección-ip/n* dev *interfaz***
- ❖ ¡Cualquier cambio realizado así es temporal!

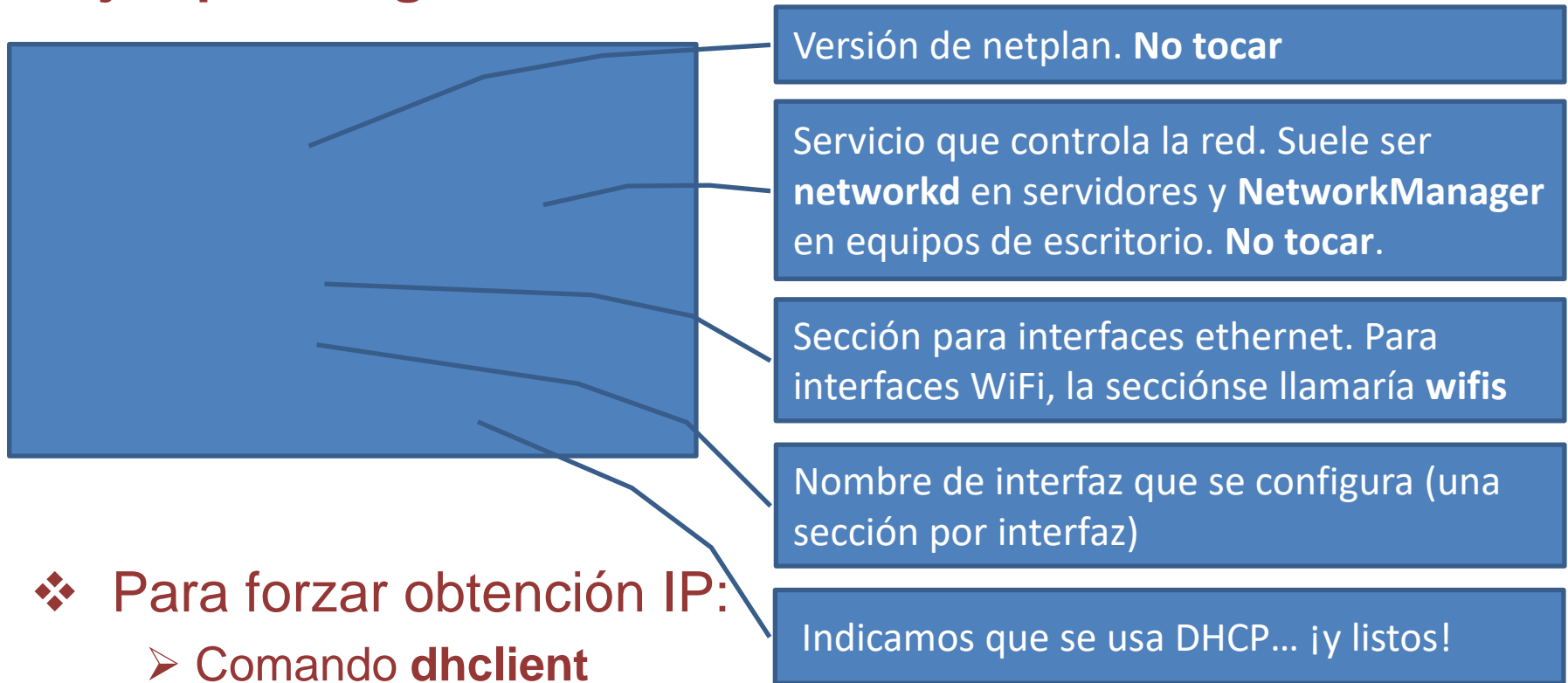
Configuración de red

Configuración permanente mediante `/etc/netplan`

- ❖ Algunos sistemas aún usan archivo `/etc/network/interfaces`
- ❖ Se procesan en orden alfabético los archivos en `/etc/netplan`
- ❖ Sintaxis de los archivos: YAML
 - ¡La indentación forma parte de la sintaxis de lenguaje!
- ❖ Recomendación: usar plantilla, en lugar de partir de cero
- ❖ Para aplicar cambios tras editar configuración:
 - **netplan try**
 - Comprueba la nueva configuración, pero no la hace efectiva
 - **netplan apply**
 - Comprueba y hace efectiva la nueva configuración

Configuración de red

Ejemplo: asignación de IP mediante DHCP



❖ Para forzar obtención IP:

➤ Comando **dhclient**

❖ Identificación ante el servidor DHCP: se usa **machine-id**

➤ Archivo **/etc/machine-id**

Configuración de red

Ejemplo: asignación de IP estática

network:

version: 2

renderer: networkd

ethernets:

eno4:

addresses:

- 150.214.188.46/24

gateway4: 150.214.188.1

nameservers:

addresses:

- 150.214.186.69

- 150.214.130.15

Igual que ejemplo anterior

Dirección (o direcciones) IP de esta interfaz con el tamaño de máscara de red

Dirección de la pasarela de red.
gateway4 está descontinuado, nueva sintaxis:

routes:

- to: default

via:150.214.188.1

Direcciones IP de los servidores DNS

Configuración de red

El archivo /etc/hosts

- ❖ Contiene asociaciones entre nombres y direcciones IP
- ❖ Utilidad: máquinas en red privada (no tiene sentido su inclusión en DNS)
- ❖ Sintaxis:
 - Una asociación por línea: **dirección-IP nombre**
 - Líneas que empiezan por # se tratan como comentarios:

```
127.0.0.1 localhost
192.168.101.1 warlock
192.168.101.2 halloween
192.168.101.3 manowar
192.168.101.4 server1
192.168.101.5 server2
192.168.101.6 server3
192.168.101.7 server4
```

- ❖ ¡Cada máquina en la red debe tener su archivo hosts, y estos deben ser coherentes!

Configuración de red

El archivo `/etc/hostname`

- ❖ Contiene el nombre de la máquina
- ❖ Si en `/etc/hosts` hay una entrada para `127.0.1.1`: debe coincidir con `hostname`
 - ¡No confundir con `127.0.0.1`!
- ❖ Para cambiar el nombre:
 - Comando **`hostname nuevo-nombre`**
 - El cambio solo es efectivo hasta nuevo inicio del sistema
 - Modificar `/etc/hostname` + reiniciar
 - Comando **`hostnamectl set-hostname nuevo-nombre`**
 - Cambio permanente sin necesidad de reiniciar
 - (derechos de administración necesarios en todos los casos)

Configuración de red

Comando: **ping**

❖ **Propósito:** Comprueba si hay conectividad con un host en la red mediante el envío de datagramas ECHO_REQUEST

❖ **Formato:**

ping host [opciones]

➤ *host*: Nombre de host o dirección IP (IPv4 o IPv6) del host cuya conectividad se comprueba

➤ Opciones habituales:

- *-c n*: Envía n paquetes ECHO_REQUEST. Si no se especifica, envía paquetes indefinidamente hasta que se interrumpe.
- *-4*: Usa sólo protocolo IPv4
- *-6*: Usa sólo protocolo IPv6



Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Procesos

Proceso en primer plano: proceso que se ejecuta bajo control de la consola. Esta no queda libre hasta que el proceso termina.

Procesos en segundo plano: procesos que se ejecutan sin control directo por parte de la consola

- ❖ Los procesos se identifican mediante un PID numérico
- ❖ Pueden haber múltiples procesos en segundo plano
- ❖ Para lanzar un comando como proceso en segundo plano: posponer carácter “&”
 - Cuando el proceso termina aparece una indicación en consola

Procesos

- ❖ Para pasar a segundo plano proceso en primer plano:
 - Pulsar **CTRL-Z** durante ejecución del proceso
 - Proceso queda bloqueado
 - Se le asigna un número de trabajo, que se muestra en consola
 - Para reanudar ejecución en segundo plano: comando **bg**
bg [trabajo]
 - *trabajo*: identificador del trabajo que se reanuda en segundo plano. Si no se especifica, reanuda el último detenido
 - Para traer de nuevo a primer plano: comando **fg**
fg [trabajo]
 - *trabajo*: identificador del trabajo que se trae a primer plano. Si no se especifica, trae el último detenido
 - Una vez en primer plano, el proceso vuelve a tomar el control de la consola
 - Para consultar la lista de trabajos en ejecución: comando **jobs**
- ❖ Para obtener información sobre los procesos en ejecución: comandos **ps** y **top**
- ❖ Para abortar un proceso: comando **kill**

Procesos

Comando: **ps**

❖ **Propósito:** lista los procesos en ejecución

❖ **Formato:**

ps [opciones]

➤ Opciones habituales:

- **-A:** muestra los procesos de todos los usuarios. Por defecto, solo muestra los del usuario actual.
- **-C *comandos*:** muestra los procesos cuyo ejecutable coincide con alguno de los nombres especificados en *comandos*
- **-p *pids*:** muestra los procesos cuyos PID's se indican en *pids*
- **--ppid *pids*:** muestra los procesos cuyos procesos padres son algunos de los indicados en *pids*.
- **-u *usuarios*:** muestra los procesos que pertenecen a alguno de los usuarios especificados en *usuarios*
- **-f, -F:** formato largo / extralargo. Proporciona información extendida
- **-L:** muestra los hilos

Procesos

Comando: **pidof**

❖ **Propósito:** Muestra PID's de procesos a partir de sus nombres

❖ **Formato:**

pidof [opciones] nombre₁ [... nombre_n]

➤ nombre_i: *Nombre* de los procesos cuyo PID se desea obtener

➤ Opciones habituales:

- -s: Muestra un único PID (pueden haber varios procesos con mismo *nombre*)
- -o *excep-pid*: No tiene en cuenta el proceso cuyo PID es *excep-pid*

➤ **IMPORTANTE:** Los procesos no tienen *nombre* como tal. Este comando usa como nombre el nombre del archivo ejecutable

- Puede ser práctico en ocasiones...
- ... pero es muy poco fiable

Procesos

Comando: **top**

- ❖ **Propósito:** muestra “en tiempo real” información detallada sobre los procesos en ejecución. Termina al pulsar la tecla “q”.
- ❖ **Formato:**
 - top [opciones]
 - Opciones habituales:
 - **-d ss.cc:** fija el periodo de actualización en *ss* segundos *cc* centésimas
 - **-n num:** actualiza la información solo *num* veces, y termina
 - **-p pid₁, ...pid_n:** muestra información solo sobre los procesos cuyos PID's son *pid₁, ...pid_n*
 - Este comando admite muchísimas más opciones. Se recomienda consultar su página de manual para más detalles.

Procesos

Comando: **kill**

- ❖ **Propósito:** Envía una señal a un proceso.

Concepto de señal: petición software dirigida a un proceso que altera su flujo de ejecución forzando un tratamiento de la señal.

- ❖ **Formato:**

kill -señal pid₁ [pid₂...pid_n]

➤ *pid_i*: PID (o PID's) de proceso al que se dirige la señal. Si se especifica -1, se envía a todos los procesos a los que se pueda enviar

➤ *señal*: Número o nombre de la señal que se envía. Las más frecuentes:

Nombre	Número	Acción por defecto
SIGTERM	15	Petición de terminación
SIGKILL	9	Abortar inmediatamente

Procesos

Comando: **pgrep, pkill**

- ❖ **Propósito:** Muestra (pgrep) o envía una señal (pkill) a procesos según criterios de selección
- ❖ **Formato:**
`pgrep [opciones] [patron] / pkill -señal [opciones] [patron]`
 - *señal:* Número o nombre de la señal que se envía (sólo pkill)
 - *patron:* Nombre (o parte del nombre) del ejecutable del proceso
 - Opciones habituales:
 - *-l, -L:* Muestra también el nombre (-l) o la línea de comandos completa (-L)
 - *-n:* Selecciona sólo el más reciente de los procesos
 - *-o:* Selecciona sólo el más antiguo de los procesos
 - *-u UID, -U UID:* Selecciona sólo los que tengan el UID efectivo (-u) o real (-U) especificado (se admite UID numérico o nombre de usuario)
 - *-g GID, -G GID:* Como -u y -U pero con el grupo. Si GID es cero selecciona procesos del mismo grupo que el propio comando.



Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Servicios

Servicio: proceso del sistema que realiza algún tipo de actividad en segundo plano

- ❖ Normalmente se inician tras el arranque
- ❖ Distintas distribuciones tienen distintos gestores de servicios

systemd: gestor de servicios más habitual

- ❖ Es el primer proceso que ejecuta el núcleo tras arrancar
- ❖ Se encarga de lanzar los demás servicios
- ❖ Gestiona un tipo genérico de recurso llamado *units*
- ❖ Comando para controlar systemd: **systemctl**
 - Requiere derechos de administración para realizar cambios

Servicios

Operaciones habituales con servicios:

1. Listar todos los servicios instalados en el sistema

- `systemctl list-unit-files --all --type=service [--state=estado]`
- Posibles estados:
 - **enabled**: El servicio se iniciará automáticamente tras el arranque
 - **disabled**: El servicio no se iniciará tras el arranque pero se puede arrancar manualmente
 - **masked**: El servicio no se puede iniciar mientras esté en este estado
 - **static**: solo pueden ser iniciado por otro servicio que lo requiera
 - **generated**: Servicio generado por el propio systemd

2. Listar todos los servicios junto información resumida:

- `systemctl --type=service --all [--state=running/exited/dead]`

3. Consultar estado detallado de un servicio:

- `systemctl status servicio`
 - `servicio`: nombre del servicio (nombre de la unidad, sobreentiende “.service”)

Servicios

unit: tipo genérico de recurso gestionado por **systemd**

- ❖ Formato del nombre de una *unit*: **nombre.tipo**
- ❖ Los servicios son un tipo de *unit* (tipo *.service*)
- ❖ Cada *unit* viene especificada por un archivo en **/usr/lib/systemd/system**
 - ¡No tocar nada en ese directorio!
- ❖ Para listar todos los archivos de *units*:
 - `systemctl list-unit-files [--type=tipo]`
 - `--type`: lista sólo los del tipo indicado, por ejemplo, `--type=service`
- ❖ Para listar todos las *units* que se han intentado cargar:
 - `systemctl list-units [--all] [--type=tipo]`
 - `--all`: muestra las que se han intentado cargar pero no están actualmente cargadas

Servicios

Operaciones habituales con servicios (cont.):

4. Habilitar o deshabilitar servicios:

- `systemctl enable/disable/is-enabled servicio`

5. Iniciar, detener o reiniciar servicios manualmente:

- `systemctl start/stop/restart/reload servicio`
 - **reload**: Ordena al servicio que cargue de nuevo su configuración sin necesidad de reiniciar (no todos los servicios lo permiten)

6. Comprobar si un servicio está activo (en ejecución):

- `systemctl is-active servicio`

7. Declarar servicio como no iniciable (o como iniciable):

- `systemctl mask/unmask servicio`

8. Mostrar información detallada sobre un servicio:

- `systemctl show servicio`

Servicios

Algunos usos adicionales de `systemctl`:

- ❖ Apagar el sistema:
 - `systemctl poweroff` (alternativa: `shutdown now`)
- ❖ Reiniciar el sistema:
 - `systemctl reboot` (alternativas: `reboot`, `shutdown -r now`)



Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Instalación y administración de soft.

- ❖ El software se obtiene y descarga de repositorios
 - Cada distribución Linux tiene sus repositorios oficiales
 - El sistema dispone localmente de una lista de los repositorios en los que busca las aplicaciones (oficiales o no)
 - Las aplicaciones se descargan en forma de paquetes
 - Cada paquete, puede depender de otros paquetes
- ❖ Distintas distribuciones Linux usan diferentes formatos de paquetes
- ❖ UBUNTU: hereda de Debian los comandos...
 - apt (o apt-get)
 - add-apt-repository
- ❖ Nuevo sistema UBUNTU: snap
 - Hay rechazo a este sistema por parte de otras distribuciones

Instalación y administración de soft.

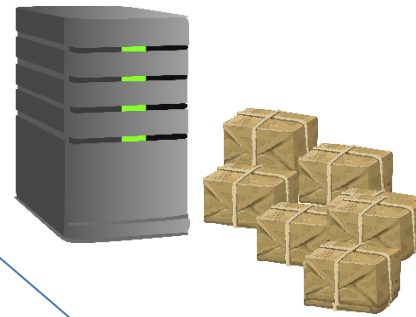
Listado de repositorios
(`/etc/apt/sources.list`)



Caché de paquetes
descargados
(`/var/cache/apt/archives`)

Copia local índices
(`/var/cache/apt/pkgcache.bin` y
`/var/cache/apt/srcpkgcache.bin`)

Canonical
(repositorios oficiales)



Repositorios
no oficiales



Instalación y administración de soft.

Comando: **apt**

- ❖ **Propósito:** Ejecuta los diferentes comandos de administración (instalación, desinstalación, actualización...) de paquetes de software
- ❖ **Formato:**
 - apt [opciones] comando
 - *comando*: comando de administración que se ejecuta. Los más importantes son update, dist-upgrade, install, remove, purge, autoremove, check, clean, autoclean
 - apt requiere derechos de administración. Por tanto, si el usuario que lo usa no es root, debe usarlo mediante el comando **sudo**.
 - Opción **-y**: Omite confirmaciones por parte del usuario.

Instalación y administración de soft.

Comando: **apt update**

❖ **Formato:**

apt update

❖ **Propósito:** Actualiza los índices locales del sistema con el contenido actual de los repositorios de aplicaciones.

- Incorpora a los índices locales las nuevas aplicaciones (o nuevas versiones) añadidas a los repositorios
- No descarga ningún paquete

Instalación y administración de soft.

Comando: **apt list**

❖ **Formato:**

apt list [nombre] [opciones]

❖ **Propósito:** Lista paquetes que cumplan con el criterio de búsqueda dado por nombre y opciones.

- *nombre*: Nombre (o patrón) de los paquetes que se listan
- Opciones habituales:
 - *--installed*: Muestra sólo paquetes instalados
 - *--upgradeable*: Muestra paquetes que pueden actualizarse
 - *--all-versions, -a*: Muestra todas las versiones de cada paquete

Instalación y administración de soft.

Comando: **apt dist-upgrade**

❖ **Formato:**

`apt dist-upgrade`

❖ **Propósito:** Actualiza los paquetes instalados a la versión más moderna.

- Requiere un comando previo **apt update** para determinar qué paquetes han sido actualizados
- Si algún paquete a actualizar depende de otro paquete para el que también hay una actualización...
 - Actualiza dicho paquete
 - Elimina la versión antigua

Instalación y administración de soft.

Comando: **apt install**

❖ **Formato:**

apt install paquete₁ [paquete₂ ... paquete_n]

❖ **Propósito:** Instala o actualiza los paquetes cuyos nombres se indican.

- *paquete_i*: Nombre del paquetes o paquetes que se instalan.
 - Si alguno de los paquetes ya está instalado, lo actualiza
 - Se puede añadir un carácter igual “=” seguido de un número de versión para indicar explícitamente la versión a instalar. En caso de no hacerlo, instala la versión más reciente.
 - Instalar una versión anterior a la actualmente instalada implica un *downgrade* del paquete: ¡usar con cuidado!
- Si un paquete a instalar depende de otros no instalados, dichos paquetes dependientes también se instalan

Instalación y administración de soft.

Comandos: **apt remove**, **apt purge**

❖ **Formato:**

`apt remove paquete1 [paquete2 ... paqueten]`

`apt purge paquete1 [paquete2 ... paqueten]`

❖ **Propósito:** Desinstala los paquetes cuyos nombres se indican.

- *paquete_i*: Nombre del paquetes o paquetes que se desinstalan
- *remove*: Los archivos de configuración del paquete o paquetes desinstalados se conservan
- *purge*: Se eliminan por completo los paquetes incluyendo archivos de configuración

Instalación y administración de soft.

Comando: **apt autoremove**

❖ **Formato:**

`apt autoremove`

- ❖ **Propósito:** Desinstala paquetes que se instalaron para satisfacer dependencias de otros paquetes, y que actualmente han dejado de ser necesarios

Comandos: **apt check**

❖ **Formato:**

`apt check`

- ❖ **Propósito:** Realiza una comprobación de consistencia del índice local de paquetes. Actualiza el índice y comprueba si hay dependencias rotas.

Instalación y administración de soft.

Comandos: **apt clean**

Formato:

`apt clean`

❖ **Propósito:** Borra el repositorio local de paquetes instalados

- `/var/cache/apt/archives/`
- `/var/cache/apt/archives/partial/`

Comandos: **apt autoclean**

❖ **Formato:**

`apt autoclean`

❖ **Propósito:** Borra del repositorio local de paquetes aquellos paquetes que no pueden ser descargados y que no son útiles

Instalación y administración de soft.

Comando: **add-apt-repository**

- ❖ **Propósito:** Añade o elimina repositorios de software (actualiza el archivo `/etc/apt/sources.list`)
- ❖ **Formato:**
 - `add-apt-repository [opciones] repositorio`
 - *repositorio*: repositorio que se añade o se elimina. Formatos habituales:
 - **URI** del repositorio:
Ejemplo: `add-apt-repository http://packages.mate-desktop.org/repo/ubuntu`
 - **Personal Package Archives (ppa)**: `ppa:usuario/nombre-ppa`
Ejemplo: `add-apt-repository ppa:gwendal-lebihan-dev/cinnamon-stable`
 - **-r**: Elimina el repositorio, en lugar de añadirlo
 - **-y**: Omite confirmaciones por parte del usuario
 - **-u**: Actualiza el índice local de repositorios tras añadir nuevo repositorio

The background of the slide features a large, faded, golden-brown image of the Angel of the University of Seville. The angel is depicted as a winged figure, standing and holding a long staff or scepter in its right hand. The figure is dressed in flowing robes and stands upon an ornate, classical-style pedestal. The entire image is set against a light yellow-to-white gradient background.

¡Gracias!

UNIVERSIDAD DE SEVILLA



Laboratorio 1:

Administración de Sistemas Linux

1. Primeros pasos
2. Usuarios, grupos y administradores
3. Conexión a sistemas remotos
4. Configuración de red
5. Procesos
6. Servicios
7. Instalación y administración de software

APÉNDICES

Edición de texto con vi

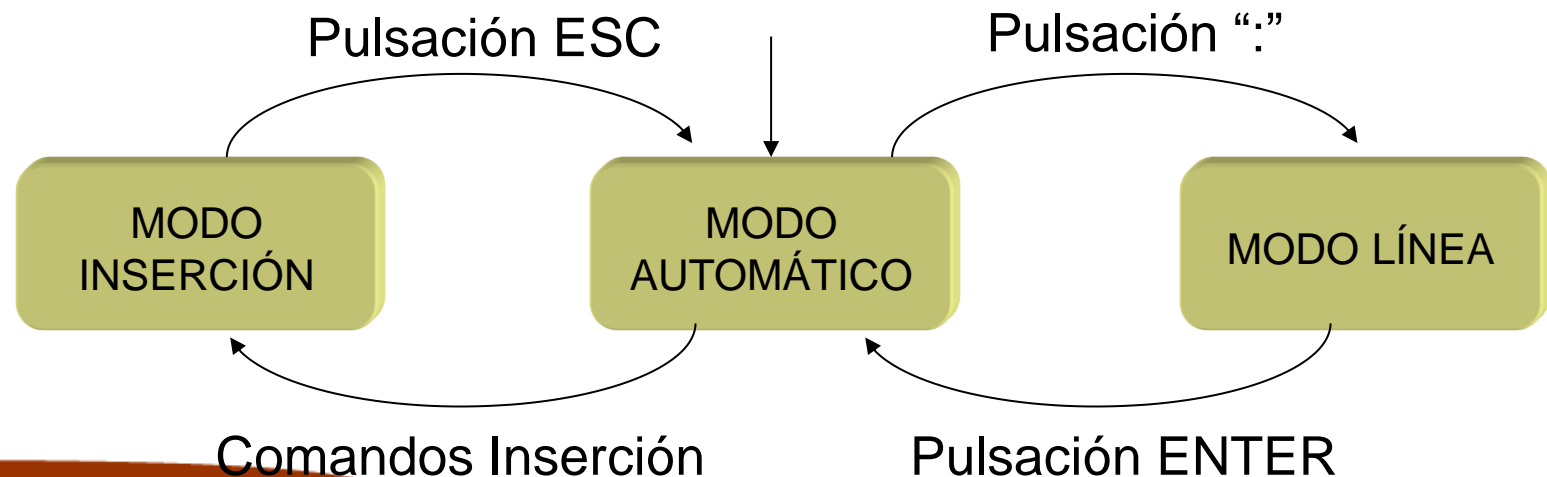
❖ Uso:

vi [opciones] [archivo]

➤ *archivo*: archivo que se edita. Si no se especifica uno, crea uno nuevo y hay que darle nombre al salir.

➤ Opciones habituales:

- *+n*: coloca el cursor inicialmente en la línea *n*
- *+*: coloca el cursor inicialmente en la última línea
- *+/patron*: coloca el cursor inicialmente en la primera ocurrencia del patrón (mismo tipo de patrones que comando **grep**)



Edición de texto con vi

Modo automático

- ❖ Permite mover cursor por el texto.
- ❖ Algunos comandos de movimiento:
 - *Flechas de cursor*: movimiento sobre el texto
 - *Ctrl-f*: avanza una página
 - *Ctrl-b*: retrocede una página
 - */patrón*: Mueve el cursor a la siguiente ocurrencia del patrón
 - *?patrón*: Mueve el cursor a la anterior ocurrencia del patrón
 - *x*: borra carácter bajo cursor
 - *dd*: borra la línea bajo el cursor

Edición de texto con vi

Modo inserción

- ❖ Permite la entrada de texto
- ❖ Algunos comandos de inserción:
 - *a*: añade texto detrás del cursor
 - *A*: añade texto al final de la línea
 - *i*: inserta antes del cursor
 - *I*: inserta texto antes de la línea
 - *o*: abre una línea debajo de la que está el cursor
 - *O*: abre una línea sobre la que está el cursor

Edición de texto con vi

Comandos en modo línea

- ❖ Algunos comandos en modo línea:
 - `:n`: mueve el cursor a la línea `n`
 - `:$`: mueve el cursor al a última línea
 - `:r archivo`: lee el archivo y lo coloca tras el cursor
 - `:w archivo`: escribe el texto en el archivo
 - `:q!`: abandona la edición sin guardar cambios
 - `:x`: abandona la edición, guardando el archivo si se han efectuado cambios