

# Gestión de ficheros

- Operación desde la CLI: el shell bash
  - Directorio de trabajo:
    - Directorio en el que el intérprete de comandos está actualmente posicionado.
    - El comando “pwd” envía a la salida estándar la ruta absoluta del directorio de trabajo.
  - Al emplear como argumento de comandos nombres de fichero o directorio que incluyan espacios será necesario encerrarlos entre comillas dobles.
  - Proporciona una utilidad de autocompletado de nombres de programas y de rutas de ficheros mediante la tecla TAB.
  - El comando interno “alias” permite crear nombres alternativos para comandos complejos.

# Gestión de ficheros

- Administración básica de directorios:
  - Comandos relacionados:
    - ls (comando externo):
      - `ls [<opciones>] [<fichero> / <directorio>] [<fichero> / <directorio>] ...`
      - Aplicado sobre directorios, lista el contenido de éstos, mostrando el nombre completo de cada fichero y subdirectorio que dependen de él, excepto aquellos cuyo nombre comience por "." (ocultos).
      - Aplicado sobre ficheros, muestra el nombre completo de cada fichero.
      - En caso de no recibir ningún fichero o directorio como argumento, actuará sobre el directorio actual.
      - Opciones importantes
        - -l      Formato largo (permisos, enlaces, propietario, ...).
        - -a      Muestra también ficheros ocultos.
        - -R      Lista recursivamente los directorios encontrados.

# Gestión de ficheros

- Administración básica de directorios:
  - Comandos relacionados:
    - `cd` (comando interno de *bash*):
      - `cd [<directorio>]`
      - Permite cambiar el directorio de trabajo.
        - `"."` representa al directorio actual.
        - `".."` representa al directorio padre.
        - `"/"` representa al directorio raíz.
      - Si no recibe argumentos establecerá como directorio de trabajo el directorio de casa del usuario.

# Gestión de ficheros

- Administración básica de directorios:
  - Comandos relacionados:
    - mkdir (comando externo):
      - *mkdir [<opciones>] <directorio1> [<directorio2>] ...*
      - Permite crear directorios.
      - Opciones importantes:
        - -p      Crea los directorios intermedios de la ruta.
      - Ejemplos:
        - *mkdir Música Música/Pop*
        - *mkdir -p Video/Películas/Terror*

# Gestión de ficheros

- Administración básica de directorios:
  - Comandos relacionados:
    - rmdir (comando externo):
      - *rmdir [<opciones>] <directorio1> [<directorio2>] ...*
      - Permite borrar directorios vacíos.
      - Opciones importantes:
        - -p Elimina los directorios intermedios de la ruta.
      - Ejemplos:
        - *rmdir Música/Pop Música*
        - *rmdir -p Video/Películas/Terror*

# Gestión de ficheros

- Edición y visualización de ficheros de texto:
  - Comandos y programas relacionados:
    - nano (comando externo):
      - *nano [<fichero1>] [<fichero2>] ...*
      - Es un editor de textos para la interfaz de línea de comandos.
    - cat (comando externo):
      - *cat [<opciones>] [<fichero1>] [<fichero2>] ...*
      - Concatena el contenido de los ficheros recibidos como argumento, o de la entrada estándar, enviando el resultado a la salida estándar.

# Gestión de ficheros

- Edición y visualización de ficheros de texto:
  - Comandos y programas relacionados:
    - head (programa de usuario):
      - *head* [*<opciones>*] [*<fichero>*]
      - Envía a la salida estándar las 10 primeras líneas del fichero recibido como argumento, o de la entrada estándar.
      - Opciones importantes:
        - -n <N> Especifica el número de líneas sobre el que actuar.
    - tail (programa de usuario):
      - *tail* [*<opciones>*] [*<fichero>*]
      - Envía a la salida estándar las 10 últimas líneas del fichero recibido como argumento, o de la entrada estándar.
      - Opciones importantes:
        - -n <N> Especifica el número de líneas sobre el que actuar.

# Gestión de ficheros

- Edición y visualización de ficheros de texto:
  - Comandos y programas relacionados:
    - less (programa de usuario):
      - *less [<fichero>]*
      - Visualiza y/o pagina el contenido del fichero recibido como argumento, o de la entrada estándar.
      - Comandos internos importantes:
        - /        Permite introducir una cadena a buscar.
        - n        Busca hacia abajo la siguiente coincidencia.
        - N        Busca hacia arriba la siguiente coincidencia.
        - <        Se posiciona en el principio del documento.
        - >        Se posiciona en el final del documento.
        - q        Salir.



# Gestión de ficheros

- Trayectorias, rutas o caminos:
  - Es la especificación de la ubicación de un fichero o directorio dentro del árbol de directorios del sistema.
  - Existen dos tipos de trayectorias:
    - Absolutas:
      - Al especificar la ubicación del fichero o directorio no se tiene en cuenta el directorio de trabajo. Por tanto, será necesario indicar la ruta de acceso completa al fichero o directorio.
      - Ejemplo:
        - `ls -l /etc/passwd`
    - Relativas:
      - Al especificar la ubicación del fichero o directorio se tiene en cuenta el directorio de trabajo.
      - Ejemplo:

```
$ pwd
```

```
/home/damian
```

```
$ ls -l ../etc/passwd
```

# Gestión de ficheros

- Administración de ficheros:
  - Comandos relacionados:
    - Crear ficheros:
      - Mediante editor de texto:
        - *nano fichero*
        - Nano, vi, gedit, ...
      - Mediante redirección:
        - *> fichero*
      - Mediante comando touch:
        - *touch fichero*
        - Crea un fichero vacío con el nombre indicado.

# Gestión de ficheros

- Administración de ficheros:
  - Comandos relacionados:
    - cp (comando externo):
      - Modo 1:
        - *cp [<opciones>] <fichero\_origen> <fichero\_o\_directorio\_destino>*
        - Copia un fichero regular.
        - Permite que la réplica tenga un nombre distinto al del fichero origen.
      - Modo 2:
        - *cp [<opciones>] <fichero1> <fichero2> ... <directorio\_destino>*
        - Copia varios ficheros regulares a un directorio destino.
      - Modo 3:
        - *cp -r [<opciones>] <directorio\_origen> <directorio\_destino>*
        - Copia un directorio recursivamente (en profundidad).
        - Si el directorio destino no existe previamente, éste representará el nombre a asignar al directorio raíz de la jerarquía copiada.
        - Si el directorio destino existe previamente, la copia se hará dentro de él, respetando en este caso el nombre del directorio original.

# Gestión de ficheros

- Administración de ficheros:
  - Comandos relacionados:
    - mv (comando externo):
      - Modo 1:
        - *mv [<opciones>] <fichero\_origen> <fichero\_o\_directorio\_destino>*
        - Mueve o renombra un fichero regular.
        - Permite que la réplica tenga un nombre distinto al del fichero origen.
      - Modo 2:
        - *mv [<opciones>] <fichero1> <fichero2> ... <directorio\_destino>*
        - Mueve varios ficheros regulares a un directorio destino.
      - Modo 3:
        - *mv [<opciones>] <directorio\_origen> <directorio\_destino>*
        - Mueve o renombra un directorio recursivamente (en profundidad).
        - Si el directorio destino no existe previamente, éste representará el nombre a asignar al directorio raíz de la jerarquía movida (puede implicar cambio de nombre).
        - Si el directorio destino existe previamente, el directorio origen se moverá a su interior, respetando en este caso el nombre del directorio original.

# Gestión de ficheros

- Administración de ficheros:
  - Comandos relacionados:
    - rm (comando externo):
      - *rm [<opciones>] <fichero1> [<fichero2>] ...*
      - Borra uno o mas ficheros regulares.
      - Mediante una de sus opciones también permite borrar directorios no vacíos.
      - Opciones importantes:
        - -r        Permite borrar directorios, aunque no estén vacíos.
        - -i        Solicita confirmación antes de cada borrado.
    - ln (comando externo):
      - ¿?

# Gestión de ficheros

- Selección múltiple de ficheros:
  - Comodines en bash:
    - Los comodines permiten crear **patrones** (expresiones que incluyen comodines y caracteres normales), útiles para referenciar varios ficheros a la vez y operar con ellos de forma conjunta.
    - Los comodines son interpretados (expandidos) por el intérprete de comandos, que convierte el patrón en la secuencia de nombres de ficheros con los que cuadra. No se deben entrecomillar.
    - Solo los patrones que comiencen explícitamente por el carácter "." pueden cuadrar con nombres de ficheros que comiencen por "."

# Gestión de ficheros

- Selección múltiple de ficheros:
  - Comodines en bash:
    - Algunos comodines en bash:

Comodín	Expansión
~	Ruta absoluta del directorio de casa del usuario.
*	Cero ó más caracteres.
?	Un único carácter.
[Conjunto de caracteres]	Cualquier carácter simple del conjunto de caracteres. Al poner ! delante de la secuencia se denota a los caracteres no incluidos en dicha secuencia.

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - GNU/Linux proporciona un fichero en /dev para representar a cada uno de los dispositivos presentes en el sistema. Por ejemplo:

/dev/console	Consola del sistema
/dev/tty1	Terminal 1
/dev/tty2	Terminal 2
/dev/pts/1	Pseudo-terminal 1
/dev/pts/2	Pseudo-terminal 2
/dev/null	Dispositivo nulo
/dev/ttyS0	Primer puerto serie
/dev/lp0	Primer puerto paralelo

- Los procesos de un sistema GNU/Linux pueden usar los ficheros de /dev para acceder a los dispositivos a los que éstos representan.
- Además, cada proceso puede acceder a tres dispositivos especialmente importantes:
  - **Entrada estándar (/dev/stdin)**, asociada inicialmente con el fichero que representa al terminal actual (teclado).
  - **Salida estándar (/dev/stdout)**, asociada inicialmente con el fichero que representa al terminal actual (monitor).
  - **Salida estándar de errores (/dev/stderr)**, asociada inicialmente con el fichero que representa al terminal actual (monitor).



# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - Un proceso que...
    - ...lea de la entrada estándar estará leyendo, por defecto, del teclado.
    - ...escriba en la salida estándar estará escribiendo, por defecto, en el monitor.
    - ...escriba en la salida estándar de errores estará escribiendo, por defecto, en el monitor.
  - Los comandos suelen leer de la entrada estándar, y suelen escribir en la salida estándar y en la salida estándar de errores.
  - Las **redirecciones** permiten modificar los dispositivos asociados por defecto a la entrada estándar, a la salida estándar, y a la salida estándar de errores:

Símbolo	Descripción	Sintáxis
> y >>	Redirección de salida estándar.	Programa > fichero
2> y 2>>	Redirección de salida estándar de errores.	Programa 2> fichero
<	Redirección de la entrada estándar.	Programa < fichero

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - Mediante las **tuberías** ó pipes ("|") se consigue comunicar la salida estándar de un comando o programa con la entrada estándar de otro comando o programa.
  - Es muy habitual usar una tubería para pasar la salida de un comando o programa a uno de los comandos de tipo filtro:
    - wc:
      - `wc [<opciones>] [<fichero1>] [<fichero2>] ...`
      - Recibe un conjunto de líneas a partir de uno o mas ficheros, o de la entrada estándar, y envía a la salida estándar el número de caracteres, palabras o líneas.
      - Opciones importantes: -m, -w, -l

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - Sort:
    - `sort [<opciones>] [<fichero1>] [<fichero2>] ...`
    - Recibe un conjunto de líneas a partir de uno o mas ficheros, o de la entrada estándar, y las envía ordenadas ascendentemente a la salida estándar.
    - Opciones importantes:
      - -r        Invierte el criterio de ordenación.
      - -t        Indica el delimitador de campos (por defecto, espacio).
      - -k        Indica el campo por el que se ordenará.
      - -n        Indica ordenación por valor numérico.
    - Ejemplos:
      - `sort alumnos.txt`
      - `ls -l | sort -k 5 -n` (ordenación por tamaño).
      - `sort -t ":" -k 3 -n /etc/passwd` (ordenación por UID).

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - Cut:
    - *cut <opciones> [<fichero1>] [<fichero2>] ...*
    - Recibe un conjunto de líneas a partir de uno o mas ficheros, o de la entrada estándar, y envía a la salida estándar solo los campos seleccionados. Las líneas de los ficheros se consideran estructuradas en campos delimitados, por defecto, por un tabulador.
    - Opciones importantes:
      - -c      Indica la posición de los carácter/es a mostrar.
      - -d      Indica el delimitador de campos.
      - -f      Indica la posición de los campo/s a mostrar.
    - Ejemplos:
      - *cut -d “:” -f 3 /etc/passwd (obtener UUIDs).*
      - *cut -d “:” -f 1,3 /etc/passwd (obtener nombres y UUIDs).*

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - grep (Global Regular Expression and Print):
    - `grep [<opciones>] <expresión_regular> [<fichero1>] [<fichero2>] ...`
    - Recibe un conjunto de líneas a partir de uno o más ficheros, o de la entrada estándar, y envía a la salida estándar las líneas que concuerden con un determinado patrón (proporcionado mediante una expresión regular).
    - Las expresiones regulares están formadas por letras y números, así como por **metacaracteres** (caracteres que tienen un significado especial). Los metacaracteres deben ir entre comillas dobles.
    - Metacaracteres útiles para la construcción de expresiones regulares:
      - `^`, `$`
      - `.`
      - `[<lista de caracteres>]`, `[^<lista de caracteres>]`
      - `?`, `*`, `+`
      - `|`
      - `()`

# Gestión de ficheros

- Redirecciones, tuberías y filtros:
  - grep (Global Regular Expression and Print):
    - Opciones importantes:
      - -E      Formato extendido para las expresiones regulares.
      - -c      Muestra el número total de líneas coincidentes.
      - -v      Muestra solo las líneas que no concuerdan.
    - Ejemplos:
      - *grep -E -c "^pablo" f1.txt*
      - *grep -E "go+gle" f1.txt*
      - *grep -E -c "[^0-9]\$" f1.txt*
      - *cat f1.txt | grep -E "un (usuario/administrador) del sistema"*
      - *grep -E -v "eliminar" f1.txt*
      - *grep -E "^.....\$" f1.txt*

# Gestión de ficheros

- Búsqueda de ficheros:
  - find:
    - *find* [*<directorio>*] [*<expresión>*]
    - Busca ficheros en un árbol de directorios de forma recursiva.
    - La verdadera potencia del comando reside en poder ejecutar acciones sobre los ficheros encontrados.
    - Expresiones importantes:
      - -name <patrón para find (entre comillas)>
      - -exec <comando o programa> \;
    - Ejemplos:
      - *find* ~ -name "\*.gif"
      - *find* . -name "f?.txt" -exec mv {} /tmp \;