

# SISTEMAS OPERATIVOS

## TEMA 2 – INTÉRPRETE DE MANDATOS

### ACTIVIDADES

#### Actividad 1 (básica): entrada en el sistema

1. Iniciar el sistema.
2. Entrar en la cuenta de alumno (palabra clave: **urjc**).
3. Abrir un terminal. Opciones:
  - a. Hacerlo desde la barra de favoritos (a la izquierda)
  - b. Hacerlo desde la barra superior de menús
  - c. Pinchando con el botón derecho del ratón sobre el escritorio y seleccionando **Abrir terminal**.
  - d. Una vez abierto un terminal, desde el menú **Archivo** de su propia ventana es posible abrir más terminales en nuevas ventanas o pestañas adicionales.
4. Usar el mandato **whoami** para conocer el nombre del usuario con el que hemos entrado en el sistema.
5. Usar el mandato **who** para saber qué usuarios están conectados en el sistema. ¿Qué información adicional proporciona este mandato?

#### Notas

- Dentro de un terminal, el tamaño de letra se puede aumentar pulsando **Ctrl +** y reducir pulsando **Ctrl -**.
- El tamaño de la ventana del terminal se puede redimensionar con el ratón.
- Mediante la opción **Preferencias** del menú **Editar** se puede cambiar el tamaño inicial del terminal, el tipo de letra, los colores del texto, el aspecto del cursor, etc.

#### Actividad 2 (básica): mandato echo

1. Ejecutar el mandato **echo** con diferentes parámetros. Por ejemplo:

```
$ echo "Hol a Mundo"
$ echo Hol a Mundo
$ echo No sé multiplicar 4*2
$ echo PATH
$ echo $PATH
```
2. Ejecutar el mandato **echo** varias veces en una única línea separando cada ejecución mediante el carácter **;** con diferentes parámetros. Por ejemplo:

```
$ echo "Hol a"; echo ; echo "Mundo"
$ echo Directorios para ejecutables: ; echo $PATH
```
3. Probar lo siguiente:

```
$ echo Qui ero imprimir el carácter ;
$ echo Qui ero imprimir el carácter ";"
```

#### Nota

- Si surgen dificultades en algún momento, pulsar **Ctrl C (^C)**. Puede que el terminal se cierre. En tal caso, abrir uno nuevo para seguir practicando.

### Actividad 3 (básica): man (texto)

1. Ver la página de manual del mandato **echo**.
2. Ver la página de manual del mandato **man**.

#### Notas

- En el material de clase se indica qué teclas hay que utilizar para navegar dentro de una página de manual en la versión de texto.
- Para salir de la página de manual pulsar la tecla **Q**.

### Actividad 4 (básica): xman (gráfico)

1. Ejecutar **xman** desde la línea de mandatos.
2. En la ventana pequeña inicial de **xman**, pinchar en **Manual Page**.
3. En la nueva ventana que aparece, pinchar en **Sections** y, manteniendo el botón pinchado, arrastrar el ratón hasta **(1) User Commands** y soltar el botón. Así se pueden ver los ítems de manual que están en la sección 1.
4. Dentro de la lista de mandatos, que están ordenados en orden alfabético, pinchar sobre el mandato **echo**. Ver notas más abajo para saber cómo navegar en la lista.
5. Buscar la página de **man**. Para ello, repetir el paso 3 y buscar **man** en la lista.
6. Repetir el paso 3 pero entrando en otras secciones del manual.
7. Para salir, pinchar en **Options** y arrastrar hasta la opción **Quit**, o bien pinchar en **Quit** en la ventana pequeña inicial de **xman**.

#### Notas

- Para movernos hacia delante pulsaremos con el botón izquierdo del ratón sobre la barra de desplazamiento de la ventana, que está situada a nuestra izquierda.
- Para movernos hacia atrás pulsaremos sobre la misma barra de desplazamiento con el botón derecho.
- Si pinchamos muy abajo en la barra avanzaremos mucho, y si pinchamos muy arriba avanzaremos poco.
- Estos consejos sirven para avanzar o retroceder dentro de la lista de mandatos, y también dentro de la página de manual.

### Actividad 5 (básica): man, whatis, apropos

1. Consultar la página de **write** como mandato.
2. Consultar la página de **write** como llamada al sistema.
3. Usar el mandato **whatis** para ver si hay páginas de manual llamadas **ls**.
4. Usar el mandato **whatis** para ver si hay páginas de manual llamadas **write**.
5. Utilizando **apropos** y/o **man -k**, ver qué páginas de manual hablan sobre directorios, buscando mediante el término *directory*.
6. Utilizando **apropos** y/o **man -k**, ver qué páginas de manual hablan sobre ficheros, buscando mediante el término *file*.
7. Buscar páginas sobre otros mandatos y términos.

## Actividad 6 (básica): info

1. Entrar en la página **info** del mandato **echo**. Usar las teclas indicadas arriba para moverse por la descripción del mandato.
2. Situar el cursor encima de **\*note Common options** y pulsar **Intro** para saltar al nodo relacionado. Explorar las opciones comunes indicadas en el nuevo nodo que aparece.
3. Pulsar **L** para volver a la descripción del mandato **echo**.
4. Pulsar **T** para ir a la página de introducción de las utilidades de GNU.
5. Para salir, pulsar **Q**.
6. Explorar la página **info** de otros mandatos.
7. Ejecutar **info** sin parámetros para ir al nodo superior de la jerarquía. Sin salir de **info**, navegar por este nodo y entrar al nodo superior de la descripción del editor **nano**.

### Notas

- En el material de clase se indica qué teclas hay que utilizar para navegar dentro de una página de **info** en la versión de texto.
- Para salir de la página de manual pulsar la tecla **Q**.

## Actividad 7 (básica): bash

1. Escribir mandatos incompletos y completarlos con el tabulador (**info**, **echo**, etc).
2. Recorrer la lista de mandatos escritos previamente con las flechas de cursor.
3. Modificar líneas introducidas previamente, editándolas con las flechas izquierda y derecha, y las teclas de inicio y fin.
4. Pulsar **Ctrl R** y escribir parcialmente algún mandato anterior. Pulsar **Intro** cuando haya aparecido para ejecutarlo.

## Actividad 8 (básica): control de la ejecución (jobs, bg, fg, ps, top, kill)

1. Ejecutar **xman** en primer plano. ¿Qué sucede con el *prompt*?
2. Terminar **xman**.
3. Ejecutar **xman** en segundo plano. ¿Qué sucede con el cursor?
4. Ejecutar de nuevo **xman** en primer plano.
5. Manteniendo el foco en la ventana del terminal, pulsar **Ctrl Z**. ¿Qué mensaje aparece? ¿Se puede ahora buscar la página de manual de **echo** desde **xman**?
6. Poner en segundo plano el **xman** cuya ejecución hemos lanzado en el paso 4 (el que está detenido). ¿Se puede ahora buscar la página de manual de **echo**?
7. Ejecutar de nuevo **xman** en segundo plano.
8. Poner en primer plano el **xman** cuya ejecución hemos lanzado en el paso anterior.
9. Pulsando **Ctrl Z**, suspender la ejecución del **xman** que acabamos de poner en primer plano.
10. Ejecutar de nuevo **xman** en segundo plano.
11. Ver la lista de procesos con **ps**.
12. Ver el listado de procesos con información extendida mediante **ps -l**.
13. Usar **ps** para ver los procesos lanzados por el usuario **alumno**.
14. Usar **ps** para ver la lista de todos los procesos activos.
15. Ver la lista de procesos con **jobs**.

16. Ver la lista de procesos con **jobs -l**. ¿Qué información nueva aparece que no se mostraba al ejecutar **jobs** sin parámetros?
17. Terminar la ejecución del **xman** lanzado en el paso 7 ejecutando un único mandato.
18. Terminar la ejecución del **xman** lanzado en el paso 4.
19. Ejecutar **top** y buscar el *pid* del último **xman** que queda.
20. Terminar la ejecución del último **xman** que queda.

### Actividad 9 (básica): sistemas de ficheros (stat, mount, df, du, tree)

1. Ejecutar **stat fichero** para ver un resumen de los datos del i-nodo correspondiente a un fichero.
2. Usar **mount** para mostrar los sistemas de ficheros montados.
3. Escribir **df** para mostrar el espacio libre y utilizado en los sistemas de ficheros existentes.
4. Escribir **df -h** para mostrar el espacio libre y utilizado en los sistemas de ficheros existentes, pero mostrando la información en un modo más amigable (**-h**: *human readable*).
5. Mostrar el espacio libre del sistema de ficheros **/**.
6. Escribir **du** para conocer el espacio ocupado por los directorios del usuario. Repetir el mandato, pero mostrando el resultado en formato *human readable*.
7. Mediante el mandato **du**, mostrar el espacio ocupado por los directorios del usuario con dos niveles de profundidad.
8. Mediante el mandato **du**, mostrar el espacio ocupado por los directorios y los ficheros del usuario con dos niveles de profundidad, en formato *human readable*.
9. Usar **tree** para mostrar el árbol completo de directorios y ficheros a partir del directorio raíz del usuario (que es **/home/alumno**).
10. Usar **tree** para mostrar el árbol de directorios y ficheros a partir del directorio **/usr**, con dos niveles de profundidad.

#### Nota

- El mandato **tree** no suele venir instalado por defecto. En distribuciones Linux derivadas de Debian (como lo es Ubuntu), se puede instalar mediante la orden **sudo apt-get install tree**. La *shell* pedirá la palabra clave del usuario antes de proceder a la instalación.

### Actividad 10 (básica): manejo de directorios (cd, mkdir, rmdir, ls, tree, pwd)

1. Se pretende crear la siguiente estructura de directorios a partir del directorio home del usuario:



- a. Posicionarse en el directorio **home** del usuario.

- b. Crear el directorio **Datos**.
  - c. Entrar dentro del directorio **Datos**.
  - d. Comprobar que estamos en el directorio **Datos**.
  - e. Listar todos los ficheros del directorio **Datos**, incluyendo los ocultos.
  - f. Dentro del directorio **Datos**, crear los directorios **Inversiones**, **Nominas** y **Stocks**.
  - g. Listar de nuevo todos los ficheros del directorio **Datos**, incluyendo los ocultos y mostrando la información en formato largo.
  - h. Volver al directorio **home** del usuario. Comprobar que estamos en él.
  - i. Crear el directorio **Textos**.
  - j. Entrar en el directorio **Textos**.
  - k. Dentro del directorio **Textos**, crear los directorios **Cartas** e **Informes**.
  - l. Entrar en el directorio **Cartas**.
  - m. Dentro del mismo, crear los directorios **Avisos** y **Circulares**.
  - n. Volver al directorio **home** del usuario.
  - o. Usar el mandato **tree** para comprobar que la estructura de directorios creada es correcta.
2. Cambiar el directorio de trabajo actual a los siguientes directorios, comprobando con **pwd** que todos los cambios se hacen correctamente:
    - a. Cambiar a **/usr/bin**.
    - b. Cambiar a **/home**.
    - c. Cambiar al directorio **home** del usuario utilizando **\$HOME**.
    - d. Cambiar a **Nominas** utilizando una ruta relativa.
    - e. Cambiar a **Inversiones** utilizando una ruta relativa.
    - f. Cambiar a **Avisos** utilizando una ruta relativa.
    - g. Cambiar al directorio **home** del usuario utilizando el comando **cd** sin argumentos.
    - h. Cambiar a **Circulares** utilizando una ruta absoluta.
    - i. Cambiar a la raíz o **home** del usuario.
  3. Siendo el directorio de trabajo actual el directorio raíz del usuario, y **sin moverse de él**:
    - a. Mostrar la lista de ficheros del directorio raíz del usuario recursivamente, con todos los ficheros de sus subdirectorios.
    - b. Mostrar la lista de ficheros del directorio **/usr/bin** (colgando de la raíz del sistema).
    - c. Mostrar la lista de ficheros (incluidos los ocultos) del directorio **Cartas**.
    - d. Mostrar la lista de ficheros (con todos sus detalles) del directorio **tmp** que cuelga de la raíz del sistema.
  4. Cambiar el directorio de trabajo actual al directorio **Informes**, y **sin moverse de él**:
    - a. Mostrar la lista de ficheros del directorio **Cartas**.
    - b. Mostrar la lista de ficheros del directorio padre de **Informes**.
    - c. Mostrar la lista de ficheros del directorio **Nominas**, con todos sus detalles e incluyendo los ocultos.
  5. Borrar los directorios **Nominas** y **Cartas** (recordar que para borrar un directorio antes hay que vaciarlo).

### Actividad 11 (intermedia): manipulación de ficheros (**touch**, **nano**, **cp**, **ln**, **mv**, **rm**)

Al ejecutar cada acción indicada en esta actividad, se comprobará con los mandatos **ls**, **cat** o **less** que la acción se ha realizado correctamente antes de pasar a la siguiente.

1. Mediante el mandato **touch**, crear en **Stocks** un fichero denominado **Trimestre.17.1.txt** vacío.
2. Editar el fichero recién creado para que contenga lo siguiente:

## 2017 - Primer trimestre

Enero	1500
Febrero	3350
Marzo	4520

3. Copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.17.todos.txt**.
4. Copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.17.antiguos.txt**.
5. Copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.18.1.txt**.
6. Editar este fichero y cambiar el año por 2018.
7. Copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.18.2.txt**.
8. Editar este fichero y cambiar la palabra **Primer** por **Segundo**, y los meses por **Abril, Mayo y Junio**.
9. Volver a copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.18.3.txt**.
10. Editar este fichero y cambiar la palabra **Segundo** por **Tercero**, y los meses por **Julio, Agosto y Septiembre**.
11. Volver a copiar el fichero anterior, en la misma carpeta con el nombre **Trimestre.18.4.txt**.
12. Editar este fichero y cambiar la palabra **Tercero** por **Cuarto**, y los meses por **Octubre, Noviembre y Diciembre**.
13. Volver a copiar el fichero anterior, en la misma carpeta con el nombre **borrador.txt**.
14. Cambiar el nombre de este último fichero para quitar la extensión, de manera que se llame **borrador**.
15. Mover el fichero **borrador** a la carpeta padre de **Stocks**.
16. Listar los ficheros del directorio padre de **Stocks** con información extendida. Anotar el número de enlaces a **borrador**.
17. Dentro de la carpeta **Stocks**, crear un enlace físico (*hard link*) llamado **enlacefisico** que apunte a **borrador**.
18. Listar de nuevo los ficheros del directorio padre de **Stocks** con información extendida. Comparar el número de enlaces que tiene ahora **borrador**.
19. Listar los ficheros del directorio **Stocks** con información extendida. Anotar el número de enlaces que tiene **enlacefisico**.
20. Crear un enlace simbólico (*symbolic link*) al fichero **borrador** llamado **enlacesimbolico** dentro de la carpeta **Stocks**.
21. Listar de nuevo los ficheros del directorio padre de **Stocks** con información extendida. Comparar el número de enlaces que tiene ahora el fichero **borrador**.
22. Listar los ficheros del directorio **Stocks** con información extendida. Anotar el número de enlaces que tienen **enlacefisico** y **enlacesimbolico**.
23. Mostrar el contenido de **borrador**, **enlacefisico** y **enlacesimbolico**.
24. Borrar el fichero **borrador**.
25. Listar los ficheros del directorio **Stocks** con información extendida. ¿Qué se muestra ahora? (fijarse en el número de enlaces y en el enlace simbólico).
26. Mostrar el contenido de **enlacefisico** y **enlacesimbolico**.
27. Borrar **enlacefisico** y **enlacesimbolico**.
28. Ir al directorio **/home/alumno**, y utilizar el mandato **tar** para crear un archivo comprimido llamado **archivo.tgz** que contenga todo lo que se encuentra bajo los directorios **Datos** y **Textos**.
29. Crear un directorio llamado **pruebatar**.
30. Copiar **archivo.tgz** en **pruebatar** con su mismo nombre.
31. Entrar en **pruebatar**, y descomprimir **archivo.tgz**.
32. Usar **tree**, **ls** y **cat** para comprobar que lo que se ha descomprimido coincide con los contenidos originales bajo **Datos** y **Textos**.

33. Volver al directorio **/home/alumno** y, ejecutando un solo mandato, borrar recursivamente el directorio **pruebatar**.
34. Copiar todos los ficheros acabados en **.txt** en **Inversiones**.
35. Borrar los ficheros que tienen por nombre **Trimestre.17** seguido de cualquier cosa.
36. Borrar los ficheros que tienen por nombre **Trimestre.18**, seguido de un uno o un dos y acabados en **.txt**.

#### Nota

- Para realizar los pasos 34, 35 y 36 es preciso estudiar cómo se realizan las sustituciones de nombre de fichero.

### Actividad 12 (intermedia): filtros (**wc**, **cut**, **head**, **tail**, **sort**, **tr**, **sed**, **cat**, **more**, **tee**)

1. Mediante el mandato **wc**, contar las líneas, las palabras y los bytes del fichero **/etc/passwd**. Contar sólo las líneas del fichero. Contar sólo las palabras del fichero. Contar sólo sus bytes. Contar sus líneas y sus palabras.
2. Mediante el mandato **cut**, mostrar el *uid* de todos los usuarios del sistema, junto con su directorio raíz y la *shell* que se arranca por defecto cuando abren un terminal.
3. Mediante el mandato **head**, mostrar los datos de los cinco primeros usuarios del sistema que aparecen en **/etc/passwd**. Mostrar todos los usuarios menos los cinco últimos. Mostrar los 100 primeros bytes del fichero.
4. Mediante el mandato **tail**, mostrar los datos de los cinco últimos usuarios del sistema que aparecen en **/etc/passwd**. Mostrar todos los usuarios menos los cinco primeros. Mostrar los 100 últimos bytes del fichero.
5. Usar el mandato **sort** para mostrar todas las líneas del fichero **/etc/passwd** ordenadas por orden alfabético. Mostrarlas por orden alfabético inverso.
6. Mediante el mandato **tr**, reemplazar todas las letras '**x**' introducidas a través de la entrada estándar por '**X**', las '**y**' por '**Y**' y las '**z**' por '**Z**' (finalizar la introducción de líneas por la entrada estándar pulsando **Ctrl D**).
7. Mediante el mandato **tr**, reemplazar todas las letras minúsculas introducidas a través de la entrada estándar por mayúsculas (finalizar la introducción de líneas de la entrada estándar pulsando **Ctrl D**).
8. Mediante el mandato **sed**, mostrar el contenido del fichero **/etc/passwd**, pero reemplazando la letra '**a**' por '**AAA**'. Repetir lo anterior, pero ahora reemplazando '**sys**' por '**SYSTEM**'.
9. Mostrar el contenido del fichero **/etc/passwd**. Volver a mostrar el contenido del mismo fichero, pero página a página.
10. Copiar la entrada estándar en la salida estándar, y al mismo tiempo guardarla en disco en el fichero **/home/alumno/copia**.

### Actividad 13 (intermedia): búsquedas en ficheros (**grep**)

1. Mostrar las líneas del fichero **/etc/passwd** que contengan la tira de caracteres **root**.
2. Mostrar las líneas del fichero **/etc/passwd** que comiencen por **sys**.
3. Mostrar las líneas del fichero **/etc/passwd** que terminen por **bash**.
4. Mostrar las líneas del fichero **/usr/share/dict/words** que contengan palabras de cuatro letras que comiencen por '**c**' y terminen por '**h**'.
5. Mostrar las líneas del fichero **/usr/share/dict/words** que contengan la subcadena '**as**'.

6. Mostrar las líneas del fichero **/usr/share/dict/words** que contengan subcadenas de tres letras que comiencen por **'b'**, terminen por **'g'** y en medio tengan una **'a'** o una **'e'**.
7. Mostrar las líneas del fichero **/usr/share/dict/words** que contengan subcadenas de tres letras que comiencen por **'b'**, terminen por **'g'** y en medio tengan cualquier carácter excepto una **'a'** o una **'e'**.
8. Mostrar las líneas del fichero **/usr/share/dict/words** que contengan las subcadenas **'car'** ó **'truck'**.

#### Actividad 14 (intermedia): búsqueda de ficheros (find)

1. Buscar los **ficheros** contenidos bajo **/usr/lib** cuyo nombre comience por **'ls'** y termine por **'.so'**.
2. Buscar los **ficheros** contenidos bajo **/usr/lib** cuyo tamaño sea mayor que 2 MB.
3. Buscar los **ficheros** contenidos bajo el directorio raíz del usuario sobre los que el propio usuario tenga permiso de escritura.
4. Buscar los **directorios** contenidos bajo el directorio raíz del usuario cuyo nombre contenga la subcadena **'os'** y sobre los que el propio usuario tenga permiso de escritura.

#### Actividad 15 (intermedia): manipulación de permisos de los ficheros (chmod, umask)

1. Escribir **umask** para saber cuál es la máscara por defecto para los permisos de creación de nuevos ficheros y directorios. Anotar dicho valor.
2. Usar **touch** para crear un fichero llamado **permisos**. Comprobar los permisos de acceso del fichero recién creado y compararlos con los especificados con la máscara obtenida en el paso anterior.
3. Cambiar los permisos del fichero **permisos** para que el propio usuario no pueda leerlo (asegurarse de que no se puede ver su contenido con **cat**).
4. Volver a darle permiso de lectura para el propio usuario.
5. Darle permiso de modificación a los usuarios del grupo.
6. Quitarle permiso de lectura a otros usuarios.
7. Comprobar con **ls -l** que los permisos son **rwrw**.
8. Cambiar la máscara por defecto de modo que los permisos para los nuevos ficheros y directorios queden del siguiente modo:
  - Propietario: únicamente tendrá permiso de lectura.
  - Grupo: sólo permiso de ejecución.
  - Resto: permiso de lectura, escritura y ejecución.
9. Crear un directorio nuevo llamado **nuevodir**.
10. Hacer **ls -l** para comprobar los permisos del directorio recién creado.
11. Intentar entrar en **nuevodir**. ¿Qué sucede? Añadir un único permiso en **nuevodir** para que el propietario pueda entrar en él.
12. Una vez dentro de **nuevodir**, crear con **touch** un fichero vacío llamado **nuevofichero**. ¿Qué sucede? Cambiar un único permiso en **nuevodir** para que la creación del fichero sea posible, y crear el fichero.
13. Hacer **ls -l** para comprobar los permisos del fichero recién creado.
14. Intentar modificar el contenido de **nuevofichero**. ¿Qué sucede? Cambiar un único permiso en **nuevofichero** para que el propietario pueda modificarlo, y modificar el contenido del fichero.
15. Restaurar los permisos del paso 1 con **umask**.



## Actividad 16 (intermedia): redirección de entrada y salida

Las actividades 12 y 13 comprenden mandatos de tipo filtro que pueden tomar sus datos de entrada desde la entrada estándar. La mayoría de esos mandatos también aceptan entrada procedente de un fichero, como así se solicita en muchos de ellos. En esta actividad se pide repetir los ejercicios de las actividades 12 y 13, pero:

- En todos los casos, la entrada procederá de la entrada estándar redirigida desde un fichero (<).
- En todos los casos, la salida estándar se redirigirá a un fichero de texto (>). Como propuesta, en algunos casos la salida se redirigirá hacia un fichero existente mediante concatenación (>>).

## Actividad 17 (intermedia-avanzada): tuberías

1. Utilizar una tubería para ver la lista de ficheros de **/usr/bin** página a página (**ls, more**).
2. Utilizar una tubería para filtrar la lista de nombres de ficheros del directorio **/usr/bin** página a página, para ver sólo los que contengan el carácter "m" sin tener que usar un fichero (**ls, grep, more / ls, more**).
3. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla cuántos usuarios se encuentran conectados al sistema (**who, wc**).
4. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla en orden alfabético los nombres de los usuarios que permanecen conectados (**who, sort**).
5. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla cuántos usuarios hay en el sistema (sin tener en cuenta si están conectados o no) (**cat, wc**).
6. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla los *login names* de los usuarios del sistema. Realizar dos versiones:
  - a. Mostrar todos los *login names* por orden alfabético (**cut, sort**).
  - b. Mostrar sólo aquéllos cuyo *login name* comienza por la letra "r" (**cut, grep**).
7. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla por orden alfabético los *login names* de los usuarios del sistema junto con su directorio *home*. El *login name* del usuario estará separado de su directorio *home* por un tabulador (**cut, sort, tr**).
8. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla cuántos subdirectorios hay en el directorio actual (**ls, grep, wc**).
9. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla cuántos ficheros hay en el directorio actual (**ls, grep, wc**).
10. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla sobre cuántos ficheros y directorios tenemos permiso de escritura en el directorio actual (**ls, cut, grep, wc**).
11. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla sobre cuántos ficheros tenemos permiso de escritura en el directorio actual (**ls, cut, grep, wc / find, wc**).
12. Escribir una línea de mandatos conectados entre sí para mostrar por pantalla sobre cuántos directorios tenemos permiso de escritura en el directorio actual (**ls, cut, grep, wc / find, wc**).

## Actividad 18 (avanzada): xargs / sustitución de mandato

1. Borrar todos los ficheros que se encuentren por debajo del directorio actual y cuyo nombre coincida con el patrón "**\*~**" (**find, xargs, rm**).
2. Repetir el ejercicio anterior, pero ahora utilizando sustitución de mandato (**find, rm**).

3. Copiar todos los ficheros que se encuentren por debajo de la carpeta *home* del usuario y que encajen con el patrón `"*.sh"` en una carpeta llamada **temporal** creada previamente mediante un mandato anterior (**find**, **xargs**, **cp**).
4. Repetir el ejercicio anterior, pero ahora utilizando sustitución de mandato (**find**, **cp**).

### Actividad 19 (intermedia): variables y expresiones aritméticas

1. Crear una variable llamada **DATOA** con valor 25, y otra variable llamada **DATOB** con valor 4.
2. Calcular el valor de **DATOA\*DATOB+5** mediante el mandato **expr**.
3. Calcular el valor de **DATOA\*DATOB+5** mediante los mandatos **echo** y **bc**.
4. Calcular el valor de **DATOA\*DATOB+5** mediante el mandato **echo** y la evaluación con **\$((operación))**.
5. Mostrar las variables de entorno cuyo nombre termine por la cadena **PATH** (**env**, **grep**).

### Actividad 20 (avanzada): scripts

0. (Paso previo) Posicionarse en el directorio *home* del usuario, y crear en él un directorio llamado **scripts**. Entrar en el directorio recién creado. Descargar en el directorio **scripts** el fichero de apoyo **scripts.tgz**, y descomprimirlo mediante la orden **tar xvfz scripts.tgz**. Al final de este paso, todos los ficheros contenidos en **scripts.tgz** deben encontrarse en el directorio **scripts**, que además debe ser nuestro directorio de trabajo actual.
1. En la línea de mandatos, escribir directamente **cambiar\_dir.sh** para invocar el *script*. ¿Qué sucede?
2. Invocar el *script* **cambiar\_dir.sh** especificando su ruta parcial. ¿Qué sucede?
3. Cambiar los permisos de **cambiar\_dir.sh** únicamente, para que el usuario pueda ejecutarlo. A continuación, ejecutarlo precedido de su ruta parcial. ¿Qué sucede? ¿En qué directorio nos encontramos? Comprobarlo mediante el mandato **pwd**.
4. Escribir **source cambiar\_dir.sh** para ejecutar el mandato. ¿Qué sucede? ¿En qué directorio nos encontramos? Comprobarlo mediante el mandato **pwd**.
5. El *script* **cambiar\_dir.sh** acepta un único argumento, que es un directorio. Probar a ejecutar **cambiar\_dir.sh** de las dos formas, con y sin anteponer el mandato **source**, para tratar de cambiar a otros directorios existentes, y observar los efectos que se producen.
6. Volver al directorio **scripts**.
7. Escribir **bash imprimir\_pid.sh** para ejecutar **imprimir\_pid.sh** sin cambiar los permisos de acceso. Anotar el *pid* de la *shell* que ejecuta el *script*. Ejecutar varias veces esta orden y anotar los *pids* que van saliendo.
8. Escribir **source imprimir\_pid.sh** para ejecutar **imprimir\_pid.sh** sin cambiar los permisos de acceso. Anotar el *pid* de la *shell* que ejecuta el *script*. Ejecutar varias veces esta orden y anotar los *pids* que van saliendo.
9. Comparando los *pids* obtenidos en los dos pasos anteriores, ¿qué se puede concluir al respecto? Para ayudar en las conclusiones, ejecutar el mandato **ps** y ver el *pid* de la *shell* actual.
10. Cambiar los permisos de todos los *scripts* de este directorio para que puedan ser ejecutados únicamente por el usuario.
11. Ejecutar **script\_init.sh** y analizar su código.
12. Ejecutar **script\_test.sh** y analizar su código. ¿A qué equivale **\$?** ?
13. Ejecutar **script\_test.sh** de nuevo, de modo que se escriba (o sobreescriba) el fichero destino. Inmediatamente después, ejecutar **echo \$?** . ¿Qué se imprime por pantalla?

14. Ejecutar **script\_test.sh** de nuevo, poniendo como destino un fichero existente e indicando que no se sobrescriba. Inmediatamente después, ejecutar **echo \$?** . ¿Qué se imprime ahora por pantalla?
15. Ejecutar **script\_for.sh** y analizar su código. ¿De dónde sale la lista de valores que se utilizan para iterar en el bucle?
16. Ejecutar **script\_for\_txt.sh** y analizar su código. ¿De dónde sale la lista de valores que se utilizan para iterar en el bucle?
17. Analizar el código de **script\_param2.sh**. Ejecutar **script\_param2.sh** con cuatro argumentos cualesquiera, indicando la ruta de acceso al mismo para que no se produzcan errores. Observar el resultado. Ejecutar **script\_param2.sh** varias veces con diferentes números de parámetros. Comparar los resultados obtenidos con el código fuente del *script*.
  - a. ¿A qué equivale **\$#**?
  - b. ¿A qué equivale **\$\***?
  - c. ¿A qué equivale **\$@**?
  - d. ¿A qué equivale **\$0**?
  - e. ¿Podría haberse escrito el bucle **for** de forma más breve?
18. Ejecutar **script\_factorial.sh** y analizar su código.
19. Ejecutar **script\_funciones.sh** y analizar su código.
20. Escribir **source script\_funciones.sh** para ejecutar de nuevo el *script* anterior. ¿Qué sucede y por qué?