

Nombre:

Fecha: /04/2012

Grupo: 1 □

PRÁCTICA 10
GESTIÓN DE USUARIOS Y DIRECTORIOS

1. Gestión de usuarios.

Una de las grandes ventajas que históricamente han presentado los sistemas operativos basados en Unix (como GNU/Linux) con respecto a Windows ha sido la gestión de usuarios en las máquinas. Desde sus inicios Unix fue pensado como un sistema multiusuario, de tal modo que en cada máquina se podían "logar" diferentes usuarios con diferentes privilegios. Las últimas versiones de Windows han conseguido hacer la gestión de usuarios más sencilla y parecida a la de Linux (por ejemplo, Windows ofrece en sus últimas versiones la posibilidad de ejecutar aplicaciones en modo "Administrador", de modo similar a como hace Linux con el usuario "root"). También Windows reúne ahora a los distintos usuarios de un sistema operativo en grupos que comparten privilegios (tal y como se hace en Linux).

A lo largo de la práctica veremos cómo se pueden usar las herramientas (en general a través del intérprete de mandatos) para crear y eliminar nuevos usuarios y grupos, cómo comprobar los usuarios que hay en una máquina, cómo cambiar de usuario....

También prestaremos atención al usuario "root" o "Administrador". Tanto en Linux como en Windows, existe un usuario particular que tiene derechos especiales para crear nuevos usuarios, eliminarlos, arrancar ciertos servicios, instalar o desinstalar aplicaciones.... Sus capacidades son similares a la que definimos como "modo privilegiado" de acceso a la computadora (por tanto, debemos ser especialmente cuidadosos cuando hagamos uso de los mismos; por ejemplo, muchas de las vulnerabilidades de Windows se ponen de manifiesto cuando usamos la cuenta "Administrador", mientras que usando otras cuentas no pueden afectar a nuestro ordenador). El resto de usuarios, por lo general, sólo contarán con el "modo protegido" de acceso al ordenador.

A lo largo de la práctica también repasaremos algunos de los mandatos que hemos visto en prácticas anteriores; también usaremos algunos de los visores y editores de ficheros de texto más comunes en Linux y Windows.

1. Vamos a entrar en Linux con nuestra cuenta habitual (user = "alumno" y password = "alumno"). Abre una terminal. El usuario con el que te has logado en la máquina lo puedes saber fácilmente (está escrito en la parte izquierda del mensaje del "prompt"). ¿Qué más usuarios existen en la máquina? Vamos a visualizar el fichero "/etc/passwd" con un visor (no un editor) de ficheros.

Teclea en el intérprete de mandatos:

```
$man less
```

¿Para qué sirve el mandato "man"? (Puedes ejecutar "man man" en el intérprete para comprobarlo). ¿Para qué sirve el mandato "less"? (Lee su "Descripción" y apunta su finalidad. Quizá tengas que ejecutar también "man more" para aclararte).

2. Ahora que sabemos lo que hace "less", vamos a ejecutarlo sobre el fichero "/etc/passwd". Ejecuta en el intérprete:

```
$less /etc/passwd
```

En pantalla debe aparecer la lista de todos los usuarios que están dados de alta en el ordenador. La mayor parte de ellos se utilizan para tareas de administración o de gestión o comunicación interna de la máquina. Fíjate en el primero (root) y el último (alumno) que aparecen en la lista. Apunta todas sus características. ¿Qué significa cada una de ellas? (<http://blogubuntu.com/chuleta-para-etc-passwd/>).

3. Otro programa que se utiliza con asiduidad en el intérprete de mandatos es "ls". Ejecuta en el intérprete el mandato:

```
$man ls
```

4. Ejecuta el mandato:

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

En la respuesta obtenida podrás observar información acerca del fichero "/etc/passwd". Ayudándote de lo explicado en <http://freeengineer.org/learnUNIXin10minutes.html#Listing>, comprueba el significado de cada uno de los atributos del fichero "/etc/passwd".

5. Vamos a intentar editar el fichero "/etc/passwd" por medio de un editor de texto de línea. Un conocido editor de textos en Unix y Linux es "nano". Ejecuta el mandato "man nano".

Ejecuta el mandato:

```
$nano /etc/passwd
```

El uso de "nano" es exclusivo a través de teclado (sin ratón). Las opciones que señala el editor como, por ejemplo, "^G", se refieren a la combinación de teclas "Ctrl + G". Moviéndote con las flechas del teclado trata de editar el contenido del fichero (por ejemplo, elimina un usuario por medio de la tecla "Supr"). Trata de guardar el fichero resultante ("^O") con el mismo nombre. Comprueba el mensaje obtenido. Sal sin guardar el fichero.

¿A través de qué usuario estabas intentando editar el fichero? ¿A quién pertenece el fichero? ¿Qué permisos tenía asignados el fichero?

6. Otro de los ficheros relevantes a la hora de gestionar grupos y usuarios en Linux es el fichero "/etc/group".

Visualiza el fichero. Puedes utilizar el mandato "less". También puedes probar el mandato "cat"; si no sabes qué hace "cat", prueba primero "man cat". No deberías usar "nano", ya que nuestro propósito no es editar el fichero, sino sólo verlo.

7. Veamos ahora cómo podemos agregar un nuevo usuario a nuestra máquina. Recuerda que hace falta ejecutar el mandato como administrador. Puedes encontrar más información sobre el mandato "sudo" en <http://es.wikipedia.org/wiki/Sudo>. Teclea el mandato:

```
$sudo adduser cuasi
```

Completa los parámetros necesarios (presta atención a las contraseñas).

8. Vamos a volcar ahora los dos ficheros `/etc/passwd` y `/etc/group` a un tercer fichero. El mandato `cat` que hemos mencionado antes se puede usar para concatenar ficheros de texto. Comprueba el resultado de la siguiente orden:

```
$cat /etc/passwd /etc/group
```

Por defecto, la salida del mandato es la propia `shell` (o `bash`). Podemos modificar la salida por defecto, y hacer que la misma sea un fichero. Para ello tienes dos opciones:

```
$cat /etc/passwd /etc/group > usuarios_y_grupos_linux
```

El anterior mandato crea el fichero `usuarios_y_grupos_linux` y vuelca el resultado de la operación `cat /etc/passwd /etc/group` en el mismo. Si el fichero ya existía, lo sobrescribe (perdiendo la anterior información).

```
$cat /etc/passwd /etc/group >> usuarios_y_grupos_linux
```

El anterior mandato crea el fichero `usuarios_y_grupos_linux` y vuelca el resultado de la operación `cat /etc/passwd /etc/group` en el mismo. Si el fichero ya existía, añade el resultado al final del mismo.

Utiliza el mandato que prefieras de los dos anteriores.

¿A qué grupos pertenece el usuario que acabas de crear (cuasi)? ¿Pertenece al grupo `admin`?

9. Veamos ahora cómo cambiar de usuario dentro de la misma `shell`. Ejecuta el mandato `man su`. ¿Para qué sirve el mandato `su`? Ejecuta el mandato:

```
$su cuasi
```

¿Qué ha sucedido en la parte izquierda del `prompt`? ¿En qué directorio te encuentras ahora?

10. Recuerda que el usuario `cuasi` no pertenece al grupo `admin`. Veamos algunas de las consecuencias de lo mismo. Para saber a qué grupos pertenece un usuario concreto, puedes hacerlo a través del mandato `groups`. Comprueba a qué grupos pertenecen `alumno` y `cuasi`:

```
$groups alumno
```

```
$groups cuasi
```

Ejecuta el mandato:

```
$sudo adduser cuasi2
```

¿Cuál es el resultado obtenido? ¿Quiénes son los `sudoers` (busca la respuesta en <http://es.wikipedia.org/wiki/Sudo>)?

No cualquier usuario de tu máquina puede hacer tareas de administración, sólo aquellos que pertenecen al grupo de los "sudoers" (por defecto, los miembros de los grupos "root" y "admin").

11. Veamos ahora cómo podemos modificar los grupos a los que pertenece un usuario. Para ello de nuevo debemos ser administradores de la máquina, así que vuelve a "logarte" como "alumno" (`$su alumno`). Observa que debería haber cambiado el prompt.

12. Vamos ahora a usar el mandato "usermod" para hacer que el usuario "cuasi" tenga privilegios de administrador. Ejecuta "man usermod". Ejecuta ahora el mandato:

```
$sudo usermod -a -G admin cuasi
```

Comprueba en el fichero "/etc/group" (por medio de `less`, `cat...`) que la modificación ha tenido lugar. ¿Podrías haber editado "a mano" el fichero "/etc/group" para haber hecho la misma operación como administrador? ¿Y el fichero "/etc/passwd" para añadir un nuevo usuario?

Una de las características de Linux es que la mayor parte de la información se encuentra repartida en ficheros de texto en el sistema. Esto facilita que muchas operaciones, si sabemos dónde son almacenadas, se puedan hacer a través de mandatos, o también con un simple editor de texto.

13. Cierra la sesión gráfica y comprueba que puedes abrir una nueva sesión con cualquiera de los usuarios:

```
alumno
cuasi
```

14. Por último, antes de cerrar la sesión Linux, vamos a comprobar cómo un administrador puede cambiar la contraseña de cualquier usuario (incluida la del root). Para ello solo debe hacer (donde `nombre_de_usuario` debe ser el usuario al cual queremos modificar la contraseña):

```
$sudo passwd nombre_de_usuario
```

Compruébalo cambiando la contraseña del usuario "cuasi" (que también pertenecía al grupo "admin").

Vamos a intentar ahora realizar algunas operaciones similares a las que hemos completado en Linux dentro de Windows.

15. Desde la pestaña de "Buscar programas y archivos" de la barra de Windows, ejecuta la aplicación "mmc". Vete a la pestaña de "Ayuda" de la aplicación que se ha iniciado y en "Acerca de Microsoft Management Console" comprueba la información de la aplicación que estamos ejecutando así como de su versión.

Dentro de la mmc, en el menú "Archivo", selecciona la opción "lusrmgr.msc".

16. Con la carpeta "Usuarios" abierta, selecciona en el menú "Acción" la opción "Exportar lista..." y guarda en el Escritorio un fichero de texto de nombre "usuarios.txt" que contenga los usuarios de la máquina. Comprueba el contenido del fichero.

17. Selecciona el usuario "alumno" y en su menú de "Propiedades" comprueba y anota en tu informe las opciones que puedes modificar sobre el mismo.

En la pestaña "Miembro de", en la opción "Agregar", comprueba que puedes realizar las modificaciones necesarias para que el mismo pase a formar parte de cualquier grupo de la máquina (puedes buscar dicho grupo en el menú "Opciones avanzadas" por medio de la opción "Buscar Ahora").

18. En la carpeta "Grupos" de nuevo en el menú "Acción", selecciona la opción "Exportar lista" y guarda en un fichero "grupos.txt" la lista de grupos que hay en la máquina. Comprueba el contenido del fichero.

19. Selecciona el grupo "Administradores" y apunta en tu informe los miembros actuales del mismo.

20. Existen algunos grupos o usuarios, y reglas sobre los mismos, que pueden resultar de utilidad a la hora de configurar un ordenador. Por ejemplo, el usuario "Invitado" por defecto suele estar deshabilitado. Sin embargo, el mismo puede ser de utilidad para permitir el acceso al ordenador a personas que no deberían acceder a tareas de administración de la máquina. Comprueba en la dirección <http://windows.microsoft.com/es-XL/windows7/Turn-the-guest-account-on-or-off> la utilidad de la cuenta de "Invitado", y desde la carpeta "Usuarios" trata de habilitarla.

2. Gestión de directorios.

Los sistemas de directorios de un sistema operativo tienen como misión fundamental organizar la información que tenemos en el mismo. El sistema de directorios forma parte del sistema de ficheros (FAT, FAT32, NTFS, ext3). Tal y como aprendimos en el Tema 1, los discos duros por lo general tienen la información organizada en sectores, que se repartían en cilindros que pertenecían a discos, que su vez podían tener una o múltiples cabezas. Esta estructura no guarda ninguna relación con la forma en que un usuario la puede observar a través de un sistema operativo (al menos, tanto en Linux como en Windows). El usuario final encuentra la información almacenada en ficheros que se encuentran a su vez en un sistema de directorios y subdirectorios propio del sistema operativo.

La estructura de un sistema de directorios suele ser la de un árbol. Dentro de ese árbol, los directorios se ordenan de forma jerárquica. Cada directorio puede tener uno o más subdirectorios. De este modo, en cada subdirectorio podemos encontrar ficheros con idéntico nombre (aunque parezca un hecho trivial, si no fuese por los directorios esto no sería posible). A partir de ahí, las diferencias entre los sistemas de directorios empiezan a aflorar. En algunos sistemas de directorios está permitido que un mismo fichero esté en varios directorios. En otros, como Windows, cada unidad de almacenamiento del sistema contiene su propio sistema de directorios; en Linux, todos los dispositivos conectados al ordenador se integran en una única jerarquía de directorios. Trataremos de ilustrar esas características y diferencias a lo largo de esta parte de la práctica.

1. Inicia Ubuntu. Cuando la máquina haya arrancado, teclea el siguiente mandato:

```
$pwd
```

Ejecuta también en la terminal "man pwd" y recuerda el significado del mandato. Quédate con la idea de que "pwd" son las siglas de "print working directory".

2. Aparte de saber en qué directorio estamos, es importante saber también a qué directorios nos podemos dirigir desde nuestro directorio actual. Ejecuta el mandato que ya conocemos:

```
$ls -l
```

De la lista que se muestra, todas las líneas que comienzan con "d" corresponden a directorios. Esto quiere decir que nos podemos "dirigir" a ellos por medio de "cd nombre_directorio".

3. Comprueba el significado del mandato "cd" ("change directory") por medio de:

```
$help cd
```

4. Dirígete ahora al directorio "Escritorio" por medio del mandato:

```
$cd Escritorio
```

5. Comprueba la ruta completa del directorio en el que te encuentras ahora por medio del mandato:

```
$pwd
```

6. Vamos a comprobar ahora los contenidos del directorio en que nos encontramos:

```
$ls
```

7. Al parecer la carpeta está vacía. Sin embargo, aún estando "vacía", todas las carpetas en los sistemas basados en Linux contienen dos directorios. Tecllea el mandato:

```
$ls -la
```

En primer lugar, debes recordar que la opción "-l" se usa para mostrar directorios y ficheros en "long format". Comprueba en <http://www.tuxfiles.org/linuxhelp/aboutfiles.html> la sección correspondiente a "hidden files and directories". Responde a las siguientes preguntas. ¿Para qué sirve la opción "-a" del mandato "ls"? ¿Qué caracteriza a los ficheros y directorios ocultos? ¿Cuál suele ser su utilidad? ¿Cómo puedes convertir un fichero o directorio "convencional" en un fichero oculto?

¿Qué directorios ocultos han aparecido en tu carpeta "Escritorio" al teclear "ls -la"? Vamos a comprobar "qué directorios" son. Ejecuta el mandato:

```
$cd .
```

Ejecuta ahora:

```
$pwd
```

¿Dónde estabas antes de hacer "cd ."? ¿Adónde te has dirigido? El directorio "." es una abreviatura que siempre equivale a "mi directorio actual" (y se encuentra en cualquier directorio de Linux). Aunque el directorio "." pueda parecer de poca utilidad por el momento, resultará práctico más adelante para hacer operaciones como "copiar ficheros de un directorio dado a mi directorio actual".

Veamos el significado del directorio "..". Ejecuta el mandato:

```
$cd ..
```

Ejecuta "pwd".

¿Dónde estabas? ¿Adónde te ha dirigido? El directorio ".." es una abreviatura que significa "sube un nivel en el árbol de directorios". También se encuentra en todos los directorios del sistema. Compruébalo por medio de:

```
$ls -la
```

8. Vamos a seguir subiendo por el árbol de directorios de nuestro sistema. Repite el mandato:

```
$cd ..
```

9. Ahora deberías encontrarte en el directorio "/home". Lee en <http://www.tuxfiles.org/linuxhelp/linuxdir.html> y apunta en tu informe la finalidad de este directorio. ¿Qué carpetas aparecen en este directorio en tu ordenador? ¿Qué usuarios "propios" (es decir, no para uso interno del sistema) hay en tu ordenador? (Recuerda que lo puedes comprobar por medio de "less /etc/passwd"). Observa que el usuario "root" no dispone de un directorio en la carpeta "home".

10. Seguimos ascendiendo por el árbol de directorios. Ejecuta de nuevo:

```
$cd ..
```

Ejecuta después:

```
$pwd
```

¿En qué directorio te encuentras? Comprueba en <http://www.tuxfiles.org/linuxhelp/linuxdir.html> el significado del directorio "/". Observa que el directorio "/", también conocido como "raíz", no tiene nada que ver con el usuario "root". El directorio "raíz" hace referencia a que el mismo es el origen del árbol de directorios, y por tanto todos los contenidos de tu ordenador (ficheros, discos duros que puedas manejar, unidades USB...) estarán "por debajo" de este directorio. ¿Podemos seguir subiendo por el árbol de directorios? Ejecuta los mandatos:

```
$cd .. ; pwd
```

El carácter ";" concatena los dos mandatos (primero se ejecuta "cd .." y a continuación "pwd").

¿Has ascendido un nivel en el árbol de directorios?

11. Vamos ahora a hacer un breve repaso de lo que nos podemos encontrar en el directorio "raíz" de un sistema Linux (puede que haya alguna pequeña diferencia entre distribuciones, pero en general todas las distribuciones mantienen una estructura muy similar). Ejecuta:

```
$ls -la
```

Observa el propietario de todos los directorios que se encuentran en el directorio raíz. ¿Tendremos permisos de escritura en el directorio raíz? Ejecuta el mandato:

```
$mkdir hola
```

(Puedes comprobar la función de "mkdir" por medio de "man mkdir"). ¿Qué ha sucedido?

Vamos a ver brevemente qué contiene cada uno de los directorios que "cuelgan" del directorio raíz. Ejecuta de nuevo:

```
$ls -la
```

En <http://www.linux-es.org/node/112> puedes encontrar una descripción detallada de la utilidad de cada uno de ellos. En <http://www.pinoxtux.com/linux/linux-file-directory-structure> tienes una representación gráfica que también debe serte útil. Es importante que retengas algunas ideas. ¿En qué directorio encontrarías los dispositivos de almacenamiento externos (CDROM, USB...)? ¿En qué directorio(s) encontrarías los mandatos que se ejecutan desde la consola? ¿En qué directorio encontrarías los directorios de inicio de los usuarios?

Hay una opción del mandato "ls" que permite mostrar el contenido de cada directorio de forma recursiva (explorando de este modo toda la estructura de directorios de la máquina Linux que se encuentre por debajo de nuestro directorio actual). Ejecuta el mandato (usamos "sudo" para que pueda listar el contenido de todos los directorios):

```
$sudo ls -R
```

Evidentemente la salida del mandato no ha sido de gran ayuda. Sin embargo, sí puede serlo si quieres explorar todos los directorios que cuelgan de uno dado (y no todos los de la máquina, como acabamos de hacer).

Antes de seguir trabajando con directorios, es importante distinguir la diferencia entre rutas absolutas y rutas relativas (problema que ya tratamos a la hora de hablar de los enlaces en html). Desde el directorio raíz dirígete al directorio "home" de alumno:

```
$cd /home/alumno
```

Ahora dirígete a su "Escritorio"; comprueba que para llevar a cabo tal tarea, los dos mandatos siguientes serían correctos:

```
$cd Escritorio  
$pwd
```

```
$cd /home/alumno/Escritorio  
$pwd
```

La primera ruta que hemos especificado (que no empezaba por "/") es una ruta relativa al directorio en que nos encontrábamos cuando hemos ejecutado el mandato "cd Escritorio" (es decir, "/home/alumno"). Por eso, el resultado final es "/home/alumno/Escritorio".

En el segundo caso (que empezaba por "/") la ruta es absoluta. Empezando desde el directorio raíz, "/" nos dirigimos al directorio "home", dentro de éste a "alumno" y después a "Escritorio".

12. Vamos a recuperar ahora las variables de entorno que ya vimos en una práctica anterior. Ejecuta el mandato:

```
$set | less
```

Apunta el valor de la variable \$HOME. Ejecuta ahora:

```
$cd $HOME; pwd
```

¿A qué directorio te has dirigido?

13. Los usuarios también pueden definir sus propias variables de entorno por medio del mandato "export". Ejecuta el mandato:

```
$export mi_escritorio="/home/alumno/Escritorio"  
$set | less
```

¿Encuentras la variable definida entre las variables del entorno? Ejecuta ahora el mandato:

```
$cd $mi_escritorio; pwd
```

Las variables definidas por medio de "export" (o de "set") sólo son válidas para la sesión de la shell que se encuentre abierta. Si quieres hacerlas permanentes (para futuras sesiones) debes introducirlas (con precaución), por ejemplo, en el fichero ".bashrc" (que se encuentra en tu directorio "\$HOME"). El uso de variables puede ser cómodo para crear "atajos" a carpetas. Otro atajo para la variable "\$HOME" de cada uno de los usuarios viene dado por "cd ~". Comprueba el resultado de ejecutar:

```
$cd ~; pwd
```

Pasamos ahora a ver la estructura de directorios propia de los sistemas Windows, con la que por lo general estamos más familiarizados.

14. Abre un intérprete de mandatos de Windows ("cmd"). Vamos a repetir algunos de los ejercicios que hemos hecho en Linux y ver lo que sucede. Para conocer el directorio en que te encuentras en Windows puedes usar el mandato:

```
>cd
```

Apunta el nombre del directorio en que te encuentras. Observa que en Windows, las rutas de carpetas y directorios se especifican por medio de la barra "\", en lugar de usar "/".

15. Vamos a ver ahora el mandato que nos permite listar el contenido de un directorio (ls en Linux):

```
>dir
```

Observa las dos primeras entradas del directorio. ¿Las reconoces de Linux?

16. El mandato "dir" nos permite usar algunas opciones adicionales. Puedes comprobarlas por medio de:

```
>help dir
```

Vamos a observar ahora quién es el propietario de cada uno de los ficheros de la carpeta en que nos encontramos. Para ello debes usar:

```
>dir /Q
```

17. Vamos a ascender ahora por el árbol de directorios de nuestra máquina Windows. Ejecuta el mandato:

```
>cd ..
```

18. Comprueba en qué directorio te encuentras por medio de "cd".

```
>cd
```

19. Vamos a comprobar también el valor del directorio ".":

```
>cd .  
>cd
```

20. Subimos un nuevo nivel:

```
>cd ..
```

Ahora deberías encontrarte en la unidad "C:". La unidad "C:" corresponde con una de las particiones de nuestro disco duro. Cada partición del mismo, o cada unidad externa (CDROM, disquetera, USB) tiene una letra propia, y dan lugar a un árbol de directorios propio. Observa y apunta en un informe la diferencia de esta organización con la propia de Linux.

21. Si tienes alguna otra unidad habilitada (por ejemplo, un USB etiquetado como en F:, o un cd etiquetado como D:), puedes acceder a ella por medio de:

```
>F:
```

22. Comprueba que estás en C: (por medio de "cd"). Si es así, ejecuta el mandato:

```
>dir
```

Observa los contenidos del directorio "C:". Trata de identificar algunas de las carpetas más relevantes en el mismo (posiblemente sea más difícil que en Linux). Observa en particular las carpetas "Windows", "Program Files" y "Users". ¿Cuál será su utilidad?

23. En Windows, al igual que en Linux, podemos trabajar con rutas relativas y rutas absolutas. Las rutas absolutas siempre serán de la forma "Unidad:\ruta\archivo", mientras que las relativas serán simplemente "ruta_relativa\archivo".

Desde el directorio en que te encuentras ahora, puedes explorar todo el árbol de directorios de la unidad C: (que no de todo el sistema, como en Linux). Ejecuta el mandato:

```
>tree
```

La información obtenida quizá no sea de gran utilidad, pero sí que debería servir para comprobar la estructura de árbol del sistema de directorios, y cómo unas carpetas se encuentran dentro de otras.

Ejecuta de nuevo el mandato sobre la carpeta "Users":

```
>tree Users
```

Observa cómo se distribuye la información en dicha carpeta.

24. En Windows también podemos hacer uso de las variables de entorno para dirigirnos de forma más sencilla a directorios. Ejecuta el mandato:

```
>set
```

Apunta el valor de la variable HOMEPATH. Ejecuta ahora:

```
>cd %HOMEPATH%
```

¿En qué directorio te encuentras? También podemos definir nuevas variables de entorno por medio del propio mandato "set":

```
>set escritorio=C:\Users\Alumno\Desktop
```

Comprueba ahora el resultado de ejecutar:

```
>cd %escritorio%
```

(Las variables de entorno que se definen por medio de "set" en el intérprete, al igual que en Linux, tampoco son permanentes; si quieres hacerlas permanentes tienes que hacerlo en Equipo -> Propiedades -> Configuración Avanzada del sistema -> Opciones avanzadas -> Variables de Entorno).

25. Ejecuta el mandato:

```
>DOSKEY /h
```

¿Qué ha sucedido? Redirige la salida del mismo a un fichero "mandatos_windows_practica_15".

26. Comprueba que el fichero contiene la información que necesitas por medio del mandato:

```
>type mandatos_windows_practica_15
```