PRACTICA I

Sesión de Trabajo

Username

Cada uno de los usuarios dados de alta en un sistema UNIX están identificados por un ***nombre de usuario*** (***username***). Este identificador permite al usuario identificarse al establecer una sesión de trabajo en cualquier terminal del sistema. El identificador tiene un máximo de 8 caracteres alfanuméricos (incluyendo los signos de puntuación). Las letras mayúsculas se consideran caracteres diferentes a las minúsculas.

```
Red Hat Commercial Linux - Mother's Day Release
Kernel 1.2.11 on a i486

lac20 login: sotel99 <cr>
Password: ???????? <cr>

lac20_$ ▌
```

Password

Cada usuario posee una contraseña (***password***) que el sistema utiliza para verificar su identidad. Las contraseñas deben tener entre 6 y 8 caracteres de longitud (si son más largas se ignoran los caracteres adicionales). Aunque los passwords son relativamente seguros (ni siquiera el administrador puede obtener la lista de passwords del sistema), es posible encontrarlos haciendo pruebas si son muy triviales. Por eso es importante no usar palabras relacionadas con la identidad del usuario, de la máquina o de la organización, que serán las primeras en ser probadas por un usuario malicioso que pretenda ganar acceso a la máquina con la identidad de otro usuario. En general un password muy seguro será aquel formado por letras, dígitos y signos de puntuación combinados de forma arbitraria que no coincidan con ninguna palabra existente (en ningún idioma).

Es conveniente cambiar la contraseña periódicamente usando el comando ***passwd*** (o ***yppasswd*** en un sistema con servidor "páginas amarillas" (yellow pages) como ocurre en el laboratorio).

```
lac20_$ yppasswd
Changing password for sotel99
Old password:
New password:
```

El fichero /etc/passwd contiene la información relativa a cada usuario del sistema. Es un fichero de texto en el que cada línea corresponde a un usuario y contiene, entre otras cosas, su ***username***, la contraseña (encriptada), su directorio personal de trabajo y el programa que se ejecuta cuando inicia una sesión (normalmente un intérprete de comandos o ***shell***).

```
sotel99:WfUokcsaIs:5128:260:SO Tel- Grupo 99:/users/sotel/sotel99:/bin/csh
```

Esta información se ve haciendo ***ypcat passwd*** | ***grep sotel99*** si están instaladas las "yellow pages", en cuyo caso el fichero /etc/passwd no contiene las líneas asociadas a usuarios.

Comandos

Como se ve arriba, el primer programa que ejecutará el usuario ***sotel99*** será un intérprete de comandos (***/bin/csh***). Existen diversos intérpretes de comandos o ***shells***; los más conocidos son el Bourne Shell y el C Shell, aunque hay otros muy populares como el tcsh (versión extendida del C Shell) y el Z Shell (versión extendida del Bourne Shell). Más adelante trataremos en profundidad sobre las diferencias entre ***shells***.

El usuario indica al sistema lo que quiere hacer en cada momento mediante comandos. La estructura de un comando es la siguiente:

```
comando [opción(es)] [fichero(s)]
```

Las opciones de un comando modifican la forma en que trabaja éste. Una opción se especifica como un guión seguido por una o más letras. En el caso de opciones simples que no llevan argumentos, se pueden poner varias opciones siguiendo a un único guión. El fichero o ficheros indicados a continuación indicarán sobre que ficheros tiene lugar la acción del comando.

```
lac20_$ ls -l a*
-rw-r--r--   1 sotel99 alumnos  29 Mar 25 17:48 aaa
```

En este ejemplo, el comando ***ls*** produce una lista de todos los ficheros indicados. La opción -l indica al comando que se desea un listado que incluya toda la información disponible sobre el fichero y no solo su nombre.

Es posible introducir varios comandos seguidos en una sola línea, usando el punto y coma (;) como separador de comandos. También se puede escribir un comando muy largo dividido en varias líneas, usando la barra invertida (\) al final de cada línea para indicar que el comando no ha terminado y que continua en la siguiente.

Una vez se ha terminado el trabajo, el usuario debe cerrar su sesión para evitar que otra persona la use para acceder a sus datos. Para ello debe usar el comando ***logout***, o bien pulsar simultáneamente las teclas ***CTRL*** y ***D***.

Entorno y primeros comandos

Examinar el entorno y el sistema

date	muestra la fecha y hora actuales en el sistema.
cal [[mes] año]	muestra el calendario de la fecha especificada. El año se especifica con 4 dígitos.
whoami	muestra el identificador del usuario que ha abierto la sesión en este terminal.
who	lista los identificadores de usuario que tienen sesiones establecidas en este instante.
finger [usuario]	lista los identificadores de los usuarios que tienen establecidas sesiones en este instante e información adicional sobre sus sesiones como el nombre real del usuario, la hora de comienzo de la sesión, su localización física, etc. Si se especifica un identificador de usuario da información detallada sobre ese usuario.

Caracteres de control

En UNIX es posible programar el teclado del terminal para configurar las teclas que realizan ciertas funciones como detener la salida por pantalla o abortar el comando actual. A continuación se detallan las funciones más comunes del teclado junto con el carácter más habitual que la realiza. El comando ***stty*** permite cambiar el carácter o grupo de caracteres que realizan una función.

stop (^s)	detiene la salida escrita hacia el terminal.
start (^q)	reanuda la salida escrita hacia el terminal.
intr (^c)	aborta el comando que se esté ejecutando en ese instante.
erase (DEL | ^h)	borra el último carácter introducido.
eof (^d)	indica el fin de fichero.

En algunos terminales la tecla de borrado produce el carácter ***^h*** mientras en otros genera el carácter ***DEL***, por tanto puede que la función espere un carácter distinto del que genera el teclado. Ejecutando el comando ***stty -a*** veremos cuáles son las teclas programadas para cada función y podremos cambiar la tecla asignada a la función erase.

```
stty erase ^h
```

Examinar y modificar el directorio

La mayoría de los comandos especificados a continuación admiten una lista de ficheros y aplican la acción correspondiente sobre todos ellos. Sin embargo, hemos especificado la sintaxis con un solo fichero para simplificar.

ls	proporciona una lista con los ficheros de un directorio.
cat fichero	muestra en el terminal el contenido del fichero.
more fichero	muestra el contenido del fichero página a página.
wc fichero	cuenta el número de líneas, palabras y caracteres de un fichero.
mv fichero nuevo_nombre	cambia el nombre de un fichero. Si el nombre de destino incluye otro directorio, el fichero se mueve de directorio.
cp fichero destino	copia un fichero sobre otro. El destino puede ser un directorio. El directorio actual se especifica con un punto (.).
rm fichero	borra el fichero indicado.

Manuales

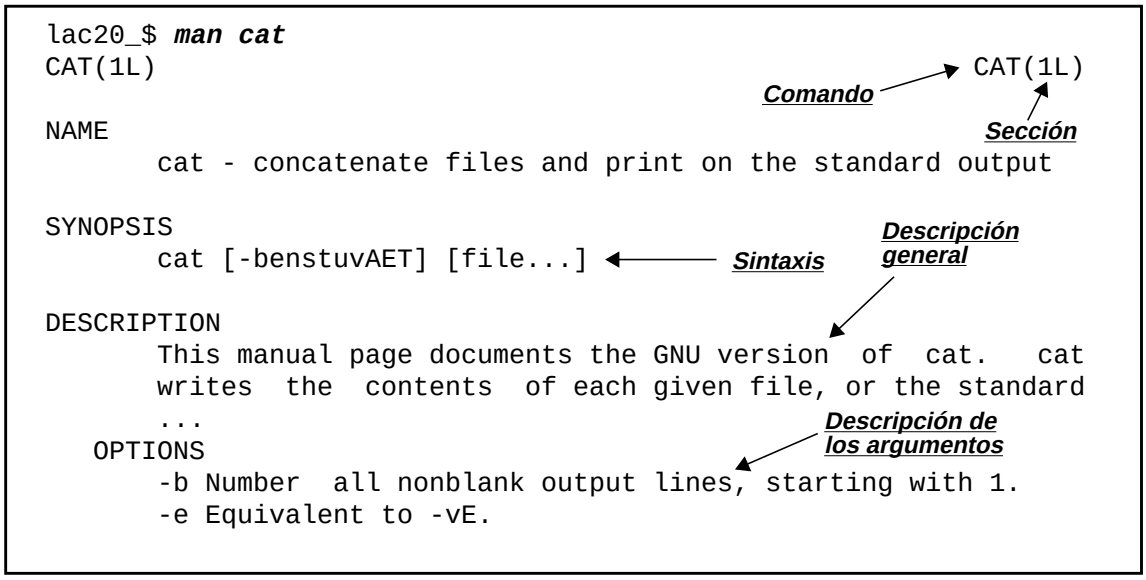
Los manuales sobre todos los comandos de UNIX están disponibles en el sistema operativo. El comando ***man*** permite consultar las páginas referentes a cualquier comando. La forma de usar este comando es:

man [seccion] nombre_comando

Es decir, podemos especificar opcionalmente la sección donde queremos buscar información del comando. Las distintas secciones se comentan en los siguientes párrafos. El comando ***man*** también está documentado, ¡por supuesto!

man man

Las páginas del manual tienen un formato uniforme que conviene conocer. Una forma de salir del man es pulsando “q”. El comando ***man*** resulta extremadamente útil cuando se nos plantea una duda en medio de una sesión, como por ejemplo, ¿hay alguna opción del comando ***ls*** que ordene los ficheros por fechas?



El manual almacenado en la máquina está dividido en varias secciones. Los comandos son tan sólo la sección 1 del sistema. El resto de secciones contienen lao siguiente:

(1) Comandos y programas de aplicación.

(2) Llamadas al sistema.

(3) Subrutinas y funciones del compilador C.

(4) Descripción de periféricos del sistema (***devices***).

(5) Formato de algunos ficheros de configuración.

(6) Juegos.

(7) Miscelánea.

(8) Comandos de administración del sistema.

(n) Comandos locales añadidos a este sistema.

Existen otras maneras de consulta del manual, ya que existen también índices que permiten localizar un comando por medio de palabras clave relacionadas con su función. Para ello usamos el comando ***apropos***, que lista todas las entradas del manual en las que aparezca la palabra especificada.

```
lac20_$ apropos mail
mail (1)             - send and receive mail
mailcap (4)          - metamail capabilities file
mailto (1)           - Simple mutlimedia mail sending program
mailto-hebrew (1)    - Run the mailto program to send Hebrew/
English mail
```

A la salida de apropos, vemos que cada línea especifica el comando relacionado, la sección del man al que pertenece y por último un resumen de su función.

Edición de ficheros

El editor ***vi*** forma parte de todas las distribuciones de UNIX. Para empezar conviene decir que es un editor pensado hace bastantes años, cuando los terminales solo permitían texto y eran mucho más primitivos. Por esta razón no ofrece un entorno muy amigable. Sin embargo, resulta conveniente saber usarlo ya que está presente en ***TODOS*** los sistemas UNIX, de forma que siempre podremos recurrir a el

en un sistema del que desconozcamos el software instalado.

vi divide la edición de un fichero en dos modos: modo de comando y modo de inserción. En la línea inferior de la pantalla se muestra el mensaje "***-- INSERT --***" mientras se encuentra en modo de inserción.

```
lac20_$ vi prueba1.txt
Mi fichero ▌
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Comandos básicos de vi

Al entrar en el editor, inicialmente estamos en el modo comando. Los comandos consisten en teclas que indican una operación determinada. Si pulsamos cualquiera de los comandos que inician una inserción se pasa a dicho modo. Al pulsar la tecla ***ESC*** se vuelve al modo comando.

i	insertar texto a la izquierda de la posición actual del cursor.
a	insertar texto a la derecha de la posición actual del cursor.
I	insertar texto al principio de la línea actual.
A	insertar texto al final de la línea actual.
o	abre una línea en blanco debajo de la línea actual e inicia la inserción en dicha línea.
O	abre una línea en blanco encima de la línea actual e inicia la inserción en dicha línea.

Cualquier tecla que se pulse en modo inserción (excepto ESC) será insertada en la línea actual. Sin embargo, para mover el cursor por el fichero es necesario estar en modo comando. Aparte de las teclas con flechas, se puede mover una página arriba (***^U***) o abajo (***^D***), ir al principio de la línea (caracter cero: "***0***") o al final de ésta (***$***).

Para borrar la línea actual, se usa el comando ***dd***. El comando ***x*** borrar el carácter que haya bajo el cursor. El comando ***D*** borra desde el carácter actual hasta el final de la línea ambos incluidos. Si uno se equivoca, se puede deshacer el último cambio con el comando "undo" (***u***).

Aparte de los comandos obtenidos pulsando una tecla, se pueden usar comandos más complejos usando el prefijo '***:***', que muestra un ***prompt*** en la línea inferior que nos invita a escribir un comando.

El comando ***:set number*** permite ver los números de línea a la izquierda de cada línea. Para desactivarlos, se debe introducir ***:set nonumber***. Un número se interpreta como ir a la línea indicada. De este modo ***:120*** salta a la línea 120 del fichero, ***:1*** va a la primera línea del fichero y ***:$*** va a la última línea.

Algunos comandos para buscar texto, copiar y pegar son:

/	Búsqueda hacia adelante

```
lac20_$ vi prueba1.txt
Mi fichero
~
~
~
~
~
~
~
~
~
:set number
```

?	Búsqueda hacia atrás
n	Repetir la búsqueda
N	Invertir el sentido de la búsqueda
yy	Copia una línea a un buffer
P	Coloca el texto del buffer antes del cursor
p	Coloca el texto del buffer después del cursor

Para salir del editor guardando los cambios realizados en el fichero, se usa el comando ***:x***. Para salir descartando todos los cambios se usa ***:q!***. El fichero se puede guardar en cualquier momento usando el comando ***:w***.

Finalmente hay que decir que se pueden poner comandos en el fichero ***.exrc*** situado en el directorio inicial del usuario, de forma que dichos comandos se ejecuten automáticamente cada vez que se ejecuta el comando ***vi***. Por ejemplo ***set number***.

Otros editores mas amigables y normalmente disponibles en instalaciones Linux son:

1. Emacs: El editor más popular en entornos Unix. Está escrito en Lisp y es extremandamente flexible y funcional.
2. xedit: Requiere X Windows. Suficiente para tareas sencillas.
3. xcoral: De particular interés para los programadores en C/C++.
4. joe: Teclas extendidas similares a las del Wordstar o al editor de TurboPascal.

Comunicación con otros usuarios

La comunicación con otros usuarios puede tener lugar de dos maneras: comunicarse de forma inmediata con otros usuarios que tengan una sesión establecida en el momento o dejar mensajes a usuarios que no están presentes en el sistema (no tienen sesiones abiertas) para que los lean más tarde (***mail***).

write usuario [terminal]	envía un mensaje directamente al terminal de otro usuario que tenga una sesión establecida en este momento.
mail [usuario]	envía mensajes a otros usuarios y permite examinar los mensajes recibidos hasta ahora.

Cuando se inicia una sesión, el shell comprueba si hemos recibido correo de otros usuarios. En este caso aparecerá el mensaje "***You have mail.***".

Para leer el correo podemos usar el comando ***Mail*** (cuidado con la 'm' mayúscula ya que existe otro comando distinto llamado ***mail***). Nos aparecerá un listado con todos los mensajes recibidos que

tengamos hasta ahora y un ***prompt*** que nos indica que se espera un comando. Simplemente pulsando ***RETURN*** se van leyendo los mensajes uno tras otro. Podemos poner también un número para leer un mensaje determinado de la lista. Si no tenemos ningún mensaje, el programa simplemente no hará nada y volverá a aparecer el indicador del ***shell***.

A continuación vemos algunos de los comandos que acepta el programa ***Mail***:

d [#]	borra el mensaje especificado o en su defecto el actual. Los mensajes no se eliminan hasta que no se sale del comando ***Mail***.
h	imprime un listado de cabeceras de los mensajes recibidos.
m usuario	crea un mensaje nuevo y lo envía al usuario especificado. Esta operación se puede llevar a cabo también especificando un nombre de usuario al invocar al comando ***Mail***.
n o RETURN	pasa al mensaje siguiente.
q	sale salvando los mensajes ya leidos en el fichero ***mbox*** y eliminando los mensajes borrados.
x	sale sin hacer ningún cambio (no elimina los mensajes borrados).
r	crea un mensaje de respuesta para el remitente del mensaje actual.
s [fichero]	salva (y borra) el mensaje actual en el fichero especificado (por defecto ***mbox***).
u [#]	recupera un mensaje borrado en esta ejecución del comando ***Mail***.
w [fichero]	salva el mensaje actual en el fichero especificado eliminando la cabecera. Por defecto salva en el fichero ***mbox***.

Para enviar correo a un usuario se debe teclear ***Mail nombre_usuario***. A partir de este momento todo lo que se teclea forma parte del mensaje y será enviado. Para terminar el mensaje y enviarlo basta pulsar ***^D*** en una línea vacía.

```
lac20_$ Mail sotel99
Quedamos esta tarde para estudiar Sistemas? <CR>
Respondeme por mail <CR>
^D
lac20_$ ▌
```

Otras aplicaciones de correo electrónico más amigables aunque no pertenecen al estándar de Unix son:

1. elm: Algo antiguo pero muy extendido.
2. pine: Con más opciones que elm (pine es el acrónimo de Pine Is No longer Elm).
3. mumail: para entornos X Windows.

Ejercicios

Responde a cada cuestión especificando que comando has usado para su realización

1. Teclee ***ls -lR /usr***. Detenga y reanude la salida por pantalla del comando[1].
2. Teclee ***ls -lR /usr*** y aborte el comando.

1. Nota: En los PC's que usan PCTCP para conectar con un servidor Linux se debe usar " | ***more***" al final de un comando cuando la salida de ese comando ocupa más de una pantalla.

3. Averigüe que hace la opción ***R*** del comando ls.
4. Cambie su password. Recuerde este password pues si no, no podrá iniciar otras sesiones.
5. Mire la hora del sistema.
6. ¿Cuál es el número mínimo de caracteres que puede tener un password?
7. Averigua si hay más usuarios presentes en el sistema ahora mismo.
8. Busque toda la información que pueda acerca de su sesión.
9. Vea por pantalla el contenido del fichero ***/etc/passwd***. Use los comandos ***cat*** y ***more***.
10. Obtenga una lista de sus ficheros por orden de antigüedad. Si no tiene ninguno, cree varios usando el editor ***vi***.
11. ¿Cuál es el nombre del mayor fichero que aparece en su directorio? ¿Hay alguna opción del comando ***ls*** que permita ver los ficheros ordenados por tamaño?
12. Envíese un mensaje por correo electrónico a si mismo. Lea el correo y extraiga cada uno de los mensajes recibidos en ficheros que se llamen ***mensaje1***, ***mensaje2***, ... A continuación compruebe el contenido de los ficheros.
13. Investigue en el manual si es posible hacer que el comando ***cp*** avise al copiar un fichero de que el destino es otro fichero que va a ser destruido.
14. ¿Es posible que al borrar una serie de ficheros se pida confirmar el borrado de cada uno de ellos?
15. Investiga los comandos del editor ***vi*** para aprender a borrar, cortar y pegar bloques de líneas.
16. Busque todos los comandos relacionados con "***time***". ¿Qué hace el comando ***time*** de la sección ***n*** del ***man***?