

PRÁCTICA SISTEMAS OPERATIVOS: UNIX

En esta práctica nos centraremos en el manejo básico en entorno Unix, especialmente en la interfaz de comandos (*shell*), así como los procedimientos para entrar y salir del sistema, información sobre los procesos y estado del sistema, de la conexión a red, y la conexión a otros equipos.

Unix es un sistema operativo **Multitarea** y **Multiusuario**. Aunque accedamos al sistema desde la consola principal, y como usuario único, el sistema podría manejar varias sesiones similares a la nuestra presentándola como si el sistema estuviera disponible sólo para nosotros, con una serie de recursos propios, y compartiendo recursos comunes (red, impresoras, discos de sistema, y demás periféricos). Asimismo, podemos iniciar **varios procesos** desde una **única sesión**, y el sistema los hará funcionar simultáneamente en modalidad de tiempo compartido.

Lo que está indicado como **MEMORIA** son los contenidos mínimos que se deben incluir en la memoria. La práctica se realizará durante los días 13 y 20 de enero. La memoria se entregará antes del 30 de enero.

1. Acceso al sistema

El acceso al sistema lo vamos a hacer desde un equipo con Windows mediante el comando telnet. Nos conectaremos al servidor del departamento (tutatis) mediante el comando "telnet tutatis"

La interfaz de entrada al sistema consiste en un "banner" o trozo de texto (configurable) que describe el sistema al que nos conectamos, y un "prompt" de login que espera a que introduzcamos el nombre de nuestro usuario:

```
Welcome to SuSE Linux 8.2 (i586) - Kernel 2.4.20-4GB-athlon (6).
```

```
tutatis login:
```

Ahí debemos introducir el usuario asignado para trabajar con el sistema, y la "password" correspondiente. Para las prácticas, el usuario será el mismo que en windows. Si se produce un error al teclear la password, o el usuario introducido no existe en el sistema, el sistema avisa y vuelve a mostrar el login: para volver a intentar la entrada.

Cambio de las contraseñas

Una vez en el sistema, se cambiará la contraseña usando el comando passwd

=> utilizar passwd para cambiar la contraseña del sistema.

En el servidor Linux está instalado un programa llamado Samba, este programa utiliza el protocolo SMB para comunicarse con equipos Windows, de forma que nos permite ver el servidor Windows como si fuera un servidor Windows más, acceder a él desde equipos windows, utilizar un directorio que como una unidad compartida desde el explorador de Windows, etc.

Nuestra contraseña de usuario para samba puede ser distinta de la contraseña del sistema. Esta contraseña se cambia con el comando smbpasswd. Se recomienda que la contraseña se cambie por la misma que se utiliza en windows, de esa forma windows se autentifica directamente al conectarnos al servidor Unix desde el explorador de Windows

=> utilizar smbpasswd para cambiar la contraseña de samba

MEMORIA:

- ¿Cómo es el proceso de login en un sistema linux?
- ¿Cuántas veces se permite intentar el login con error en el servidor tutatis?
- ¿Cómo se cambia la contraseña de usuario en unix?

2. Mapear el directorio personal a una unidad de Windows.

Una vez cambiada la contraseña de samba (smbpasswd), desde windows, a través del entorno de red se entrará al servidor tutatis. Este se encuentra en un dominio (o grupo de trabajo) samba. Al entrar en el equipo aparece el directorio personal del usuario. Este directorio se mapeará, de forma permanente, a una unidad de red Y:, de forma que este "disco duro virtual" esté disponible siempre que iniciemos una sesión en Windows, de la misma forma que tenemos disponible la unidad Z:.

MEMORIA:

- ¿Cómo se monta el directorio personal del equipo Linux en una unidad de red desde Windows?

3. Mandatos básicos

Una vez dentro, aparece el prompt del sistema, que para un usuario común será el símbolo>. A partir de ese momento, podemos comenzar a utilizar el sistema. Para ello, **Practicar** con las instrucciones siguientes (los nombres de los directorios y ficheros son ficticios, el alumno debe trabajar con lo que encuentre en el sistema al conectarse). Los términos en cursiva indican valores que debe suministrar el usuario. Las opciones adicionales que se pide probar NO incluyen la coma (:):

Mandato	Qué hace	Probar también con estas opciones
pwd	Indica en qué punto del árbol de directorios estamos	--help
touch file	Crea un fichero vacío con el nombre especificado. Crear varios ficheros con nombres distintos para hacer pruebas (a.txt, b.txt, asdasdasdasdas.erwerwer, asd.asd.asd.123.123.123, etc)	
ls file	Lista el contenido del directorio actual. Se pueden especificar subconjuntos de ficheros como *.txt, a*.tx*, a*bg.*xt, y también otros directorios como /usr, /	-a, -la, --help
cd dir	Cambia la posición al directorio especificado, sea relativo (<i>./ftp</i>) o absoluto (<i>/var/tmp</i>)	--help
cat file	Saca por pantalla el contenido de un fichero, de una vez. Comparar su uso con los comandos more y less	cat file more, --help
cp file1 file2	Copia un fichero de una ubicación a otra. <i>file1</i> debe especificar un fichero, y <i>file2</i> puede ser simplemente un directorio de destino.	--help, -i, -r
rm file	Borra un fichero del directorio. Se puede especificar el "path" completo, relativo o ninguno, así como un subconjunto de ficheros (aaa.*)	-r, --help
mv file1 file2	Traslada un fichero de una ubicación a otra. <i>file1</i> debe especificar un fichero, y <i>file2</i> puede ser simplemente un directorio de destino.	--help
mkdir dir	Crea un directorio. <i>dir</i> puede especificar un "path" relativo o completo	--help
rmdir dir	Borra un directorio. <i>dir</i> puede especificar un "path" relativo o completo, pero debe estar vacío.	-f, --help

Con respecto a la nomenclatura de ficheros y directorios, es importante resenar que **los caracteres en mayúscula y minúscula NO son equivalentes**. La longitud de los nombres es ilimitada (a efectos prácticos), puede incluir varios puntos (no existen extensiones al nombre del fichero, como en DOS).

MEMORIA:

- Comprobar cual es el límite de número de caracteres de un nombre de fichero en Linux.
- ¿Se pueden utilizar cualquier caracter en un nombre de fichero Unix? ¿Cuales no están permitidos?
- Explica la utilización y las opciones más importantes de los comandos: mkdir, rmdir, rm cp, mv, cd y ls.

4. Estado del Sistema

Los mandatos Unix para listar el estado del sistema que deberán estudiarse son **who**, **ps**, **top**, **df** y **du**.

- who**: Proporciona información sobre los usuarios que hay conectados en el sistema.
PRACTICA: Observar las diferentes salidas de los mandatos, conectándose al servidor tutatis.
who
who -q
who -m
who -H
who -H -q
¿Que usuarios están conectados al sistema?
- ps**: Permite ver los procesos abiertos en el sistema en ese momento. Las opciones importantes, de las muchas que tiene, son **-x**, **-aux**, para proporcionar información sobre todos los procesos del sistema (la opción **-x**), y **-e** para obtener toda la información del entorno del proceso ejecutándose. Para obtener todas las posibles opciones de ordenación, se puede utilizar la opción **--help**.
PRACTICA: Ejecutar el comando **ps** con combinaciones de las opciones anteriores, en tutatis. ¿que procesos estás ejecutando como tu usuario? ¿y tus compañeros?
- top**: Es una instrucción que funciona de modo continuo, proporcionando una lista de procesos ordenados, por defecto, de mayor a menor consumo de recursos de la máquina. En la cabecera proporciona información sobre el consumo del sistema, en especial, del tiempo de CPU de procesos de usuario, de sistema, e inactivo (idle).
PRACTICA: Observar durante un rato qué procesos consumen más recursos del sistema. ¿Cada cuánto tiempo se actualiza la información?
- df**: Indica el uso de los sistemas de ficheros de Unix que estén utilizándose. Proporciona el número de sistemas de ficheros, libres y total para cada sistema de ficheros, así como el punto, en el árbol de directorios, en que está montado el sistema de ficheros.
PRACTICA: Utilizando la opción **--help** para ver qué otras opciones tiene el mandato **df**, obtener una lista de los sistemas de ficheros de cada equipo, proporcionando la información de ocupación en Mbytes, en lugar de en bloques de 1024 bytes. Obtener también el tipo de sistema de ficheros. ¿Cuanto espacio libre hay en el sistema de ficheros? ¿Y que tipo de sistema de ficheros es?
- du**: Proporciona información sobre el espacio que ocupa, en bloques, el directorio especificado, así como los que euguen de él.
PRACTICA: Obtener información sobre el espacio que ocupa el directorio **/usr/bin** (utilizar **--help** para ver las opciones posibles). Obtener el tamaño de cada uno de los ficheros de ese directorio. Obtener el tamaño total de **/home** y los directorios que cuelgan de él. ¿al es el tamaño total de /home? ¿y de /usr?

Todos los mandatos Unix pueden consultarse en detalle desde el propio sistema utilizando la utilidad **man** (manual), que se invoca simplemente tecleando **man instrucción**

Una vez abierta la página del manual, es posible avanzar/retroceder por ella utilizando los cursors, las teclas de **AvPag** y **RePag** y la barra espaciadora.

También es posible consultar con **man** información sobre rutinas de programación y ficheros de configuración como **passwd**, **fstab**, **terminfo**, **printrcap**, etc.

MEMORIA:

- Describir para que sirven los comandos **who**, **ps**, **top**, **df** y **du**.
- Responder a las preguntas realizadas en los apartados marcados como **PRACTICA**

5. Mandatos de uso en red

Una característica común a todos los sistemas Unix es que la interfaz para conexión a otras redes va incluida con el sistema operativo (y lo fué desde el principio, a diferencia de los sistemas DOS o Windows). Por ello, indicamos a continuación algunas instrucciones comunes que indican el estado de las conexiones de red, que permiten observar a otros equipos y conectarse a ellos.

Más explícitamente, los equipos Unix incluyen de forma natural controladores para redes IP. En este tipo de redes, cada "host-IP" tiene asociada una dirección de red de la forma xxx.yyy.zzz.nnn, donde xxx, yyy, zzz y nnn son números entre 0 y 255. Sin entrar en más detalles, caracterizamos cada sistema por una dirección IP única, visible desde el exterior puesto que utiliza un dispositivo externo para conectarse (tarjeta de red ethernet, o puerto serie para conexiones por modem), y por una dirección interna para conexiones dentro del propio sistema, no visible desde el exterior (interfaz de **loopback**). Las direcciones IP de cada máquina han de ser únicas dentro de la red en que estén conectadas (la de la red del laboratorio es de la forma **192.168.2.n** donde n varía de un PC a otro: 10-22 para el aula 15, 50-62 para el aula 14, 1,2 y 3 para los servidores), mientras que la dirección interna o *loopback* siempre tiene el mismo valor: **127.0.0.1**.

La forma de averiguar la dirección IP de nuestra máquina es utilizar el comando **/sbin/ifconfig**, que proporciona una salida de la forma

```
$ /sbin/ifconfig
lo Link encap:Local Loopback
inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING MTU:3584 Metric:1
RX packets:18 errors:0 dropped:0 overruns:0
TX packets:18 errors:0 dropped:0 overruns:0

eth0 Link encap:Ethernet HWaddr 00:20:AF:C3:6A:E2
inet addr:150.244.27.31 Bcast:150.244.27.240 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:125 errors:0 dropped:0 overruns:0
TX packets:48 errors:0 dropped:0 overruns:0
Interrupt:11 Base address:0x300

$
```

Por otro lado, cada máquina tiene un nombre asociado, o hostname. Para averiguar cual es el nombre de la maquina, simplemente hay que teclear **hostname**. Se puede utilizar ese nombre para conectar a la propia maquina, en lugar de utilizar loopback. Es importante tener en cuenta que ese nombre (el hostname) es una especie de alias de la dirección IP, que es la que se necesita realmente para poder conectar con otras máquinas.

Las conexiones a otras máquinas se realizan a través del comando telnet. Las conexiones entre máquinas son de muy diversos tipos, cada una asociada al tipo de intercambio de datos o servicio entre ambos equipos (FTP, HTTP, rlogin, NTP, NNTP, etc.) En esta práctica haremos pruebas de conexión de login remoto de un equipo (el propio) a otro (cualquier otro PC del laboratorio). Para ello, es necesario conocer la dirección IP del otro PC, o su nombre de equipo, y tecleando **telnet <IP>** (o telnet <nombre>). Es obvio que es necesario conocer un usuario/password en el equipo remoto, que en nuestro caso es el mismo en todos los PCs, y que hemos indicado al comienzo de la práctica.

Para salir de la sesión telnet, una vez dentro del otro equipo, se teclaea **exit** y regresamos al equipo de partida. También es posible **abortar** la sesión tecleando **Ctrl-]** y cuando aparece el prompt **telnet>** tecleando **quit**.

PRACTICA:

- Conectarse a la propia maquina haciendo telnet al *hostname*, y despues a *loopback*. ¿Que diferencia aprecias?
- Conectarse a un equipo de un compañero en el aula14. ¿cual es su IP?
- ¿Que ocurre cuando haces telnet a una dirección IP que no existe?

MEMORIA:

- ¿Cual es la IP de tutatis?
- Responder a las preguntas realizadas en los apartados marcados como **PRACTICA**

5. EJECUCION y REDIRECCIÓN

Mediante una serie de caracteres especiales es posible controlar el flujo de caracteres que van desde/hacia los mandatos Unix, asignandoles valores de entrada desde un fichero, desde la salida de otro mandato; también podemos redirigir la salida de un mandato hacia un fichero, creándolo nuevo o añadiendo el texto a un fichero existente.

Pero antes hemos de mencionar algo sobre la ejecución de mandatos en Unix: puesto que Unix es **multitarea**, es posible poner en ejecución un mandato en *'background'* desde la línea de mandatos, mediante el comando **&** (*shell*). Este mandato arranca una nueva sesión de mandatos (*shell*) sobre la que se ejecuta el mandato especificado, y la tierra cuando éste termina. También es posible encadenar varios mandatos secuencialmente, utilizando el carácter ;

PRACTICA: Encadenar dos instrucciones secuencialmente de la forma:

```
$ who ; ps -aux
```

y observar la diferencia con ejecutarlos en shells diferentes:

```
$ who &
```

```
$ ps -aux &
```

MEMORIA:

- Explicar las diferencias observadas en la ejecución de las instrucciones anteriores

La entrada de datos estándar (*stdin*) es por teclado, y la salida (*stdout*) es hacia la pantalla. Esos valores pueden modificarse con caracteres especiales:

>	Permite redireccionar la salida de un mandato hacia un fichero	\$ ls -la > lista.ficheros
>>	Lo mismo, pero añadiendo el texto a lo que ya contenga el fichero	\$ ls -la ../*.txt >> lista.ficheros
<	Especifica de dónde leera los datos de entrada el mandato a ejecutar	\$ mail alumno4 < texto.txt
<<	Termina de añadir texto de entrada cuando se teclaea la secuencia de caracteres indicada. Mientras se puede introducir texto aparece el prompt >	\$ cat << parate
	Redirecciona la salida de un mandato hacia otro, tomándola este último como parámetros/calificadores	\$ ls -la * more

PRACTICA: Crear dos ficheros, mostrarlos por pantalla y almacenarlos concatenados en otro fichero sin utilizar **cp**. Se puede crear un fichero directamente de la consola mediante

```
$ cat > fichero.txt
```

```
PRUEBA DE FICHERO DE TEXTO
ESTA ES LA LINEA 1
ESTA ES LA LINEA 2
[ctr]+D
$
```

(La secuencia final (Ctrl+D) indica el carácter de fin de fichero)
También se puede crear un fichero mediante

```
$ cat << parate > fichero.texto
> prueba
> linea 1
> linea 2
> parate
$
```

MEMORIA:

- Describir las distintas formas de redirigir la entrada y la salida

6. FILTROS

Los filtros en Unix son mandatos que leen de entrada una corriente (*stream*) de caracteres, los procesan, y los presentan en la salida standard (que a su vez puede ser redirigida hacia otro mandato Unix). Estudiaremos algunos de los más habituales:

grep	Permite buscar cadenas de texto en ficheros o <i>stream</i> de datos	ps -aux grep alumno
sort	Ordena las palabras de un <i>stream</i> , o los registros de un fichero	who sort
wc	Cuenta el numero de bytes, palabras y lineas en ficheros o <i>streams</i>	wc -l fichero
uniq	Quita las líneas duplicadas de un fichero ordenado, o de un <i>stream</i>	uniq -c fichero
tr	Cambia los caracteres especificados en un fichero o <i>stream</i> por otros	cat fichero tr a-z A-Z

PRACTICA: Estudiar qué realizan los siguientes grupos de mandatos:

```
$ touch a.dat ; touch b.dat ; touch c.dat ; ls | grep dat | wc -l

$ echo 'abcde abcde 12345 12345' | tr a-z A-Z > fichero1.txt

$ echo '12343' >> fich2.txt
$ echo '234234' >> fich2.txt
$ echo '50923' >> fich2.txt
$ echo '231234' >> fich2.txt
$ cat fich2.txt | sort -d | uniq -c | sort -nr

$ cat << parate > fich3.txt
> asdasdasd
> xcvcvcvcvcv
> qweqwewqewqew
> asd asd qwd asd
> 123*09 098239
> zcvz zcvz zcvz zcvz
> 123OIU XIUPOIU POIU
> parate
$ cat fich3.txt | tr -cs '[a-zA-Z0-9]' '\n'
```

(En el último grupo de instrucciones, el prompt '>' aparece directamente como consecuencia de la redirección '<<').

MEMORIA:

- Describir para que sirven los comandos touch, echo y cat, de forma general.
- Describir qué realizan los grupos de mandatos especificados en el apartado **PRACTICA**

7. PERMISOS DE ARCHIVOS

Al hacer un listado largo, con "ls -l" se ven los permisos de cada archivo en formato **rwX rwX rwX**, donde primer campo: tipo de archivo:
- = fichero normal
d = directorio
l = enlace simbólico
segundo campo: permisos para el propietario:
r = lectura
w = modificación (incluye borrado)
x = ejecución (entrar si es un directorio)
tercer campo: permisos para el grupo
cuarto campo: permisos para el resto de usuarios

Para cambiar los atributos de un fichero se utilizan los comandos **chmod** (cambio de permisos), **chown** (cambio de usuario), **chgrp** (cambio de grupo).

PRACTICA:

- ¿Qué permisos tiene tu directorio personal /home/<usuario>?
- Modifica los permisos de tu directorio personal de forma que queden **rwrx-r--**
- Copia el archivo /home/prueba/leeme a tu directorio personal, léelo y haz lo que se indica en el

MEMORIA:

- ¿Qué permisos tenía tu directorio personal?
- ¿Cómo los has cambiado?
- ¿Qué has hecho con el archivo leeme, y cuales han sido los resultados?

8. EDITORES

El editor joe es un editor de pantalla completa (similar al edit de microsoft), pero los distintos archivos es similar el antiguo WordStar. La edición es interactiva, y las combinaciones de teclas de control se pueden consultar pulsando "KH, con lo que nos aparece en la parte superior de la pantalla una ayuda con las distintas combinaciones de teclas de control.

Otro programa muy útil es el midtnight commander (mc), clon del norton comander disponible para ms-dos. Nos permite copiar archivos, moverlos, etc y también posee un editor de texto sencillo.

PRACTICA:

El archivo /etc/DIR_COLORS especifica el nombre con el que se muestran los distintos archivos al hacer un "ls". El archivo se puede copiar en el directorio personal, cambiándole el nombre a **dir_colors** (en minúscula y precedido por un punto), para personalizar los colores para cada usuario.

- Copiar el archivo /etc/DIR_COLORS en tu directorio personal y cambiale el nombre a **dir_colors**
- Consulta el comando **man dir_colors** para ver como "funciona" este archivo de configuración.
- Modifica lo necesario para que los directorios se listen en verde y los enlaces simbólicos en rojo.
- Abre una nueva sesión y comprueba que funciona correctamente.

MEMORIA:

- Describir como se ha realizado la copia y el cambio de nombre del archivo.
- Describir qué modificaciones se han hecho en el archivo **dir_colors** y por qué.

9. El entorno gráfico Xwindow

XWindow es el encargado de visualizar la información de manera gráfica y es totalmente independiente del sistema operativo (los sistemas Unix/Linux no necesitan de XWindow para funcionar, pudiendo trabajar en modo texto). La gran diferencia entre XWindow y la interfaz gráfica de otros sistemas operativos es que XWindow distribuye el procesamiento de aplicaciones, especificando un enlace cliente-servidor. El cliente X especificará "Que hacer" al servidor X, que se encargará de "Como hacerlo". La gran ventaja de XWindow es que el servidor X de una aplicación y el cliente X donde trabajamos no tienen porqué estar en la misma máquina. Podemos estar utilizando XWindow en nuestra máquina, conectarnos a otra remota, ejecutar un programa en esta máquina remota y utilizar/ver este programa en nuestra máquina local. Todo esto independientemente de la plataforma/sistema operativo que el ordenador remoto utilice.

- Utilización de Xwindow en el equipo local

En los equipos de aula 14 podemos iniciar una sesión local (usuario alumno) en un equipo Linux. Los equipos están configurados para iniciarse directamente con un entorno gráfico Xwindow con el gestor de escritorio KDE.

PRACTICA:

Inicia una sesión Xwindow en un equipo del aula 14, con el usuario alumno. Utiliza el navegador Konkeror (menu de Inicio-personal) para moverte por el sistema de archivos local y para conectar a internet. Abre el fichero /var/lib/named/192.168.2.zone, donde están definidas las direcciones IP de los equipos y busca como se llama el equipo con la dirección 192.168.2.1.

- Utilización de Xwindow de forma remota mediante un servidor X

Desde cualquier equipo, con cualquier sistema operativo se puede iniciar una sesión X remota en un equipo UNIX y ejecutar aplicaciones de forma remota, pero viendo sus resultados de forma local. Para ello necesitamos arrancar un servidor Xwindow en el equipo local, iniciar una sesión en el equipo remoto y ejecutar las aplicaciones gráficas indicándonos que redirigir su salida hacia nuestro equipo local.

Hay dos formas de hacerlo, una de ellas es iniciar una sesión por telnet en el equipo remoto, y cada aplicación de la que queramos redirigir su salida ejecutarla de la siguiente forma:

```
aplicacion -display 129.168.2.5:0 (donde la IP 192.168.2.5 se sustituirá por la IP del equipo local o su nombre)
```

La otra forma de hacerlo es iniciar una sesión ssh en el equipo remoto, con la opción **-X**, que hace que la salida de todos los comandos gráficos se redirija de forma automática al equipo local.

PRACTICA:

- Desde un equipo con Windows, iniciar el servidor XWin.
- Iniciar una sesión remota por ssh en tutatis, indicando las opciones

```
ssh -X usuario@tutatis
```

- ejecutar el comando

```
fperal@tutatis> /opt/kde3/bin/kwrite
```

que es un editor de texto, y crear un archivo de texto en tutatis.

- Utilización de Xwindow de forma remota mediante un servidor VNC

El servidor VNC nos permite "ver" la pantalla completa del equipo remoto en una ventana, y trabajar sobre esa ventana "como si estuviéramos en el otro equipo". Para ello es necesario un servidor VNC (en el equipo remoto) y un cliente (o visor) VNC en el equipo local.

PRACTICA:

- Desde un equipo con Windows, iniciar una sesión remota por telnet en tutatis.
- Ejecutar:

```
fperal@tutatis> vncserver -geometry 800x600
```

```
New 'tutatis:2 (fperal)' desktop is tutatis:2
```

```
Starting applications specified in /fperal/.vnc/xstartup
```

```
Log file is /fperal/.vnc/tutatis:2.log
```

eso iniciará una sesión remota a tamaño de pantalla 800x600. La respuesta del comando indica que se ha abierto la sesión en el terminal :2

- Ejecutar el visor VNC (vncviewer4) en el equipo local, en el nombre de equipo se le dará: **tutatis:2** (o el número devuelto al ejecutar vncserver, que indica el número de terminal gráfico que hemos abierto en el servidor remoto. Si todo ha sido correcto, nos aparecerá nuestra sesión gráfica en una ventana (se puede hacer que cambie a pantalla completa)

MEMORIA:

- Describe las tres formas de ejecutar aplicaciones Unix de forma remota.
- Describe las diferencias que ves entre ellas, y cual de ellas te parece mejor.