

**Práctica**

**3**

GESTIÓN Y UTILIZACIÓN DE REDES LOCALES

---

Curso 2001/2002

# Conexión a otras máquinas

## Introducción

En la actualidad las redes de ordenadores permiten conectar entre sí casi cualquier tipo de sistemas, siendo posible que coexistan en un mismo entorno diferentes plataformas hardware. Así, por ejemplo, es habitual encontrar en una misma red ordenadores **Macintosh**, ordenadores **PC compatibles** y estaciones de trabajo de diversos fabricantes, como pueden ser **Hewlett Packard, Sun, Compaq**, etc., todas ellas con arquitecturas diferentes.

A su vez, la heterogeneidad de los sistemas conectados a la red se ve incrementada por la variedad de sistemas operativos que puede correr cada una de las plataformas hardware. Por ejemplo, la plataforma **Macintosh** puede correr diferentes versiones de su sistema operativo, e incluso, en ocasiones, ejecutar **Linux**. Por su parte, es habitual que los **ordenadores PC** estén configurados para usar tanto **Windows** (en cualquiera de sus múltiples versiones) como **Linux**, siendo asimismo frecuente que en muchos de ellos sólo esté disponible uno de dichos sistemas operativos. Finalmente, las estaciones de trabajo suelen utilizar el sistema operativo **Unix**, teniendo cada fabricante su propia versión del mismo.

La diversidad de hardware y software disponible en una red presenta al mismo tiempo ventajas e inconvenientes. La principal ventaja es que cada combinación hardware-software ofrece unas características diferentes, destacando especialmente en algunas de ellas. De esta forma se puede aprovechar la diversidad disponible para utilizar en cada momento el equipo que más se adecue a las necesidades particulares de ese instante. Por ejemplo, un sistema Windows está especialmente indicado para tareas de **propósito general** como puedan ser edición de textos, retoque fotográfico, multimedia, etc., mientras que si se necesita un **servidor de ficheros**, quizá lo más adecuado sea usar Windows NT o mejor aun Linux o Unix. Por otra parte, si se necesita un **servidor de cálculo** para dar servicio de forma simultánea a multitud de usuarios, será necesario utilizar Linux o Unix.

La variedad de hardware y software en una red también presenta varios inconvenientes, siendo el principal de ellos la dificultad para comunicar unos equipos con otros de forma eficiente. Dado que cada sistema posiblemente utilizará diferentes protocolos, diferentes formatos de ficheros y diferentes interfaces de usuario, el hecho de trabajar de forma remota en un equipo cuyo sistema es diferente al nuestro puede llegar a ser una tarea en ocasiones difícil.

Sin embargo, a pesar de la posible dificultad, desde el punto de vista del usuario el entorno de trabajo ideal es aquel que permite trabajar en diferentes máquinas sin tener que enfrentarse con los detalles de la comunicación entre ellas, sino que estos detalles son ocultados por el software de forma que el usuario tiene una impresión de transparencia donde todo funciona a pesar de ser consciente que está utilizando recursos que son ajenos a su máquina.

Esta práctica tiene como objetivo familiarizar al alumno con la comunicación entre diferentes máquinas, no siempre ejecutando los mismos sistemas operativos. En particular nos vamos a centrar en la conexión de un usuario de Windows a máquinas Linux. También vamos a ver cómo un usuario de máquinas Linux se puede conectar a otras máquinas Linux.

La práctica se divide en dos partes. En la primera parte utilizaremos Windows en nuestra máquina local. En la segunda parte reiniciaremos nuestra máquina para que pase a ejecutar Linux.

## Parte I: Windows → Linux

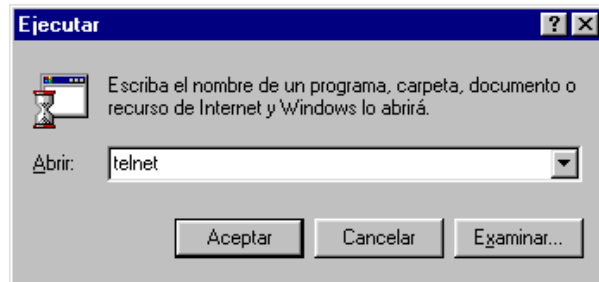
### Telnet

Telnet nos permite el acceso a sesiones de “terminal remota”. Podemos trabajar desde nuestro ordenador en una máquina remota igual que si estuviéramos sentados delante del sistema remoto, pudiendo encontrarse ese sistema en cualquier lugar de Internet (obviamente, debe tener activado un servidor de telnet). El funcionamiento de telnet es el

mismo, tanto si nos conectamos a una máquina de nuestra red local como si el destino es un sistema en cualquier otra parte del mundo, con la excepción del tiempo de respuesta, que puede verse penalizado cuando el tráfico en Internet sea elevado.

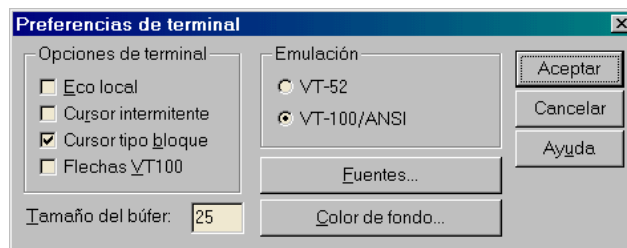
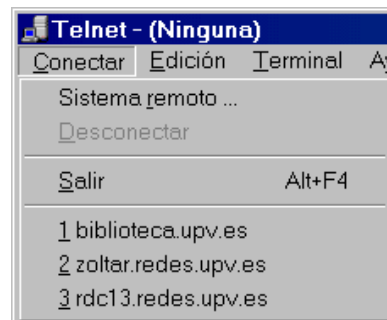
Podemos usar un cliente de telnet para conectarnos a un ordenador en el que tengamos cuenta, que puede ser Linux o Unix. Esto generalmente implica que para superar el control de acceso tendremos que proporcionar un nombre de usuario y una contraseña asociada, como hacíamos en la práctica anterior para entrar en Linux. Una vez cumplido este requisito accederemos al sistema de la forma usual.

Existen multitud de clientes telnet disponibles de forma gratuita para Windows, que suelen utilizar menús y ser bastante amigables. Entre ellos hemos elegido el que incluye el propio Windows 98. Para ejecutarlo debemos ir al menú de **Ejecutar** (Inicio → Ejecutar).



El cliente de telnet abierto nos presenta un menú con cuatro ítems disponibles: **“Conectar”**, **“Edición”**, **“Terminal”** y **“Ayuda”**. Los más utilizados son:

- **Conectar:** Es, probablemente, el único que utilizaremos de forma regular (una vez que hayamos configurado el programa a nuestro gusto al usarlo por primera vez). Las opciones que nos permite se aprecian en la figura. Podemos especificar el sistema remoto con el que queremos establecer conexión (con la opción **“Sistema remoto...”**), finalizar una conexión previa, terminar la ejecución del programa, etc.
- **Terminal:** Nos permite especificar preferencias sobre el tipo de terminal. Lo usual es seleccionar el cursor tipo bloque y poner la emulación de terminal como **VT-100**. Podemos elegir también el color del fondo de la ventana, así como el tipo de fuente empleado en las letras del texto.



**EJERCICIO: Conectarnos al servidor Linux zoltar.redes.upv.es. El nombre de usuario es rdcxx, donde xx son dos dígitos que dependen del puesto en el que estéis. Por ejemplo, si estáis en el puesto iidos02, entonces debéis usar el login rdc02.**

**¿Cuántos usuarios hay conectados? ¿Qué están haciendo?**

Para finalizar la sesión telnet que hemos abierto debemos teclear **exit**, igual que hacíamos en una sesión de Linux (al fin y al cabo, telnet es lo mismo pero en remoto).

## Ftp

Uno de los servicios más utilizados en Internet es el de **ftp**, que nos permite transmitir y recibir ficheros. Mediante ftp podemos transferir información a nuestro ordenador desde cualquier otro ordenador de Internet que disponga de un servidor ftp. No obstante, al igual que ocurría con la aplicación telnet, con ftp es necesario superar un control de acceso. Por tanto, en principio sólo podemos acceder a aquéllas máquinas en las que tengamos una cuenta de usuario.

Para que el control de acceso no limite las posibilidades de este servicio, en la mayoría de los sistemas el administrador registra un usuario bajo el nombre de **anonymous**, que tiene características especiales. Como contraseña para este usuario suele utilizarse la dirección de correo electrónico de la persona que accede al servicio (la mayoría de los sistemas realmente no lo comprueban, aunque algunos sí lo hacen). De esta forma el administrador del sistema puede llevar un registro de las personas que acceden a él. Este usuario **anonymous** tiene permiso restringido a la hora de moverse por los directorios remotos y, como medida de seguridad, no tiene acceso a todos ellos. Generalmente tampoco goza de permiso de escritura en la máquina remota, aunque en algunas ocasiones sí existe un directorio donde el visitante puede aportar información.

En esta práctica vamos a utilizar el cliente que incorpora Windows 98<sup>1</sup>. Aunque no es el más amigable de los disponibles para Windows, es el que siempre está disponible, por venir de serie. Además, se trata de un cliente muy similar a los disponibles para Linux (o Unix). El funcionamiento de este cliente se basa en una serie de comandos. Los más frecuentes aparecen en la tabla siguiente:

Orden	Función
<b>? o help</b>	muestra la lista de órdenes disponibles
<b>help orden</b>	muestra la ayuda de la orden indicada
<b>ascii</b>	pone la conexión en modo ASCII
<b>binary</b>	pone la conexión en modo binario
<b>bye</b>	termina la sesión ftp y sale del programa
<b>cd &lt;dir&gt;</b>	cambia el directorio en la máquina remota
<b>close</b>	termina la sesión ftp sin salir del programa
<b>dir</b>	lista el directorio de la máquina remota
<b>get &lt;fichero&gt;</b>	trae un directorio desde la máquina remota
<b>hash yes/no</b>	imprime un “#” por cada bloque transferido
<b>lcd &lt;directorio&gt;</b>	cambia el directorio en la máquina local
<b>ls</b>	lista el directorio de la máquina remota
<b>mget &lt;fichero&gt;</b>	permite traer múltiples ficheros a la máquina local
<b>mput &lt;fichero&gt;</b>	permite llevar múltiples ficheros a la máquina remota
<b>open &lt;nombre&gt;</b>	establece una conexión con la máquina indicada
<b>prompt</b>	activa o desactiva el modo interactivo
<b>put &lt;fichero&gt;</b>	envía un fichero a la máquina remota
<b>quit</b>	termina la sesión ftp y sale del programa

Para iniciar una sesión ftp es necesario conocer el nombre de la máquina con la que deseamos conectar (donde reside el servidor ftp). Podemos especificar el nombre en la misma línea de órdenes desde la que invocamos al programa **ftp**.

<sup>1</sup> Se ejecuta igual que el cliente de telnet: Botón de Inicio → Ejecutar

**EJERCICIO: Para continuar con la práctica es necesario que os conectéis a un servidor de ftp. Un servidor interesante es ftp.uniovi.es, al cual debéis entrar de forma anónima.**

Una vez establecida la conexión y pasado el control de acceso comenzaremos nuestra sesión, en la que podemos utilizar las órdenes indicadas en la tabla. Normalmente, la primera orden que ejecutaremos será **dir**, que nos permite observar el contenido del directorio remoto. Podemos desplazarnos por el árbol de directorios remoto con el uso de la orden **cd** y transferir ficheros con **get** y **put**.

Ftp contempla dos modos de transferencia: **ASCII** (alfanumérica) y **binaria**. La primera permite adaptar la información transferida al sistema de representación del destino convirtiendo, cuando sea necesario, la secuencia de control que indica fin de línea en los ficheros intercambiados (algunos sistemas utilizan el código ASCII 13 como indicador de final de línea, otros la combinación ASCII 13 y ASCII 10, etc.). Por otra parte, las transferencias binarias permiten intercambiar ficheros sin que estos sufran ningún tipo de transformación (de interés cuando transferimos ficheros ejecutables, imágenes o sonidos digitalizados, bases de datos, ficheros comprimidos, etc). Para escoger el tipo de transferencia se dispone de las órdenes **binary** y **ascii**. En la mayoría de sistemas el tipo por defecto cuando se establece la conexión es el **ascii**.

Para realizar transferencias múltiples se emplean las órdenes **mput** y **mget** que permiten especificar “comodines” en los nombres de fichero, como el símbolo “\*”. Si no deseamos autorizar la transferencia de cada fichero individualmente podemos emplear la orden **prompt** para desactivar el modo interactivo (y permitir la transferencia múltiple sin que se nos pida confirmación).

Las transferencias (simples o múltiples) se realizan entre los directorios actuales de las máquinas local y remota (salvo otra especificación en la orden de transferencia). No es posible la transferencia de múltiples directorios con una sola orden.

Para finalizar una sesión emplearemos la orden **quit** (o **close** si no deseamos abandonar el programa ftp).

**EJERCICIO: Después de haberos movido por el servidor de ftp de la Universidad de Oviedo (ftp.uniovi.es), ¿Qué tipo de información hay disponible en él?**

**EJERCICIO: Conectaros de forma no anónima a zoltar.redes.upv.es. ¿En qué directorio entráis? ¿Puedes escribir ficheros? ¿Puedes ir a cualquier directorio?**

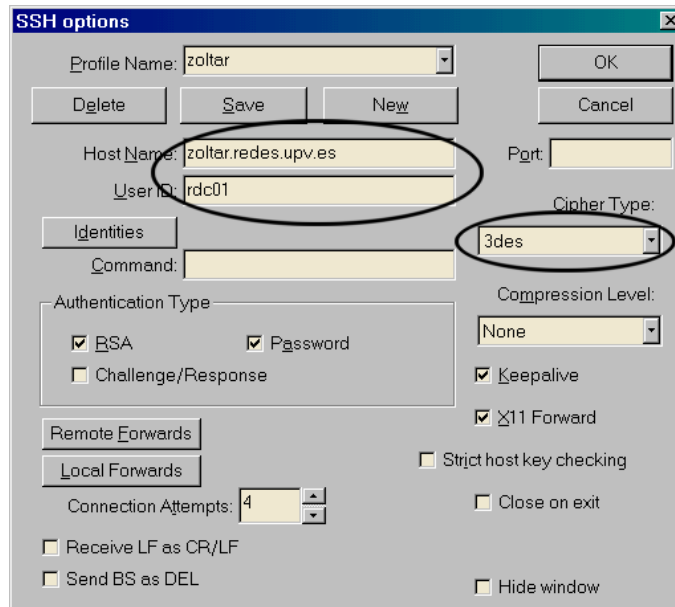
## Ssh

Cuando nos conectamos a una máquina remota, sea utilizando la aplicación telnet, sea con la aplicación ftp, la información que intercambiamos con dicha máquina viaja por una o más redes. Como veremos en una práctica posterior, cualquiera puede “pinchar” la red utilizando un programa monitor de red, y de esta forma ver la información que se está transfiriendo. Estos programas monitores de red son sencillos de conseguir, y de hecho hay sistemas operativos, como Linux, que vienen de serie con varios de estos monitores.

Para evitar que cualquier desaprensivo pueda hacerse con la información que estamos intercambiando existe una aplicación llamada **Secure Shell (ssh)**. Esta aplicación es similar a telnet, pero a diferencia de ésta, encripta la información antes de enviarla. En la máquina remota, obviamente, la información que llega debe ser desencriptada para poder utilizarla. De esto se encarga el propio sistema operativo remoto. En el sentido contrario ocurre lo mismo. Cuando utilizamos esta aplicación el sistema operativo remoto nos envía la información encriptada y nuestro Secure Shell la desencripta antes de pasárnosla a nosotros.

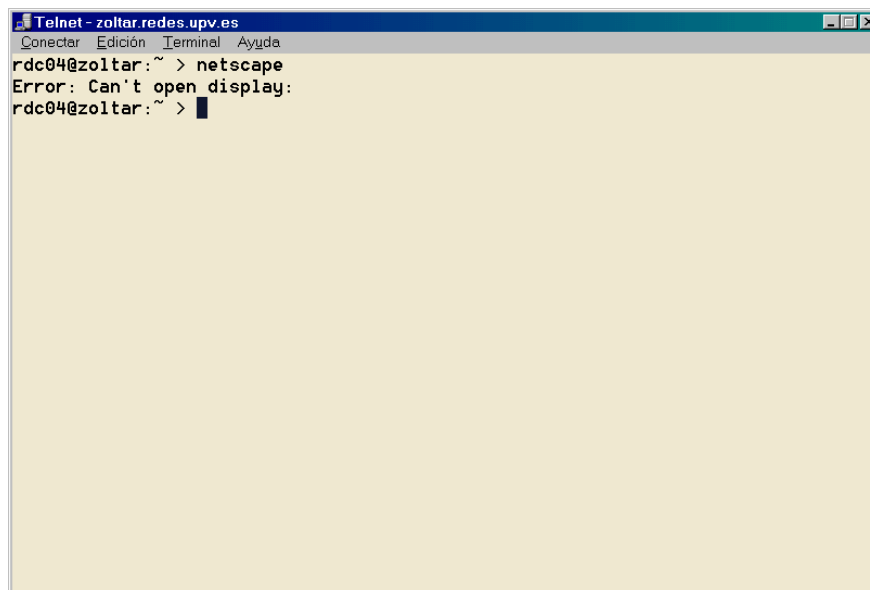
### PRACTICA 3: CONEXIÓN A OTRAS MÁQUINAS

Windows 98 lamentablemente no incorpora ningún cliente de Secure Shell, a pesar de que sí incorpora otros clientes como el de telnet o el de ftp. Para conseguir un cliente de **ssh** debéis hacer un ftp anónimo a zoltar y bajaros el fichero llamado **ssh.exe**, que es un fichero autodescomprimible. Una vez descomprimido, ejecutar el programa **ssh32.exe**. Después de una primera ventana con información variada, aparece la ventana de más abajo. Para conectarnos a zoltar debemos configurar el tipo de encriptación (3des), y por supuesto el nombre de la máquina remota y el usuario. ¡A disfrutarlo!



## Conexión remota gráfica

Una forma sencilla de utilizar recursos de otra máquina desde nuestro propio equipo consiste en utilizar la aplicación “telnet”, como hemos visto en una sección anterior. La aplicación telnet está pensada para realizar trabajos orientados a texto, esto es, la única forma que tenemos de interactuar con la máquina remota es mediante el teclado y la pantalla, en modo ASCII. Por tanto, si nos conectamos a una máquina remota, como zoltar.redes.upv.es, y tratamos de ejecutar una aplicación gráfica, como puede ser **Netscape**, podemos encontrarnos con una respuesta similar a la de la figura siguiente:



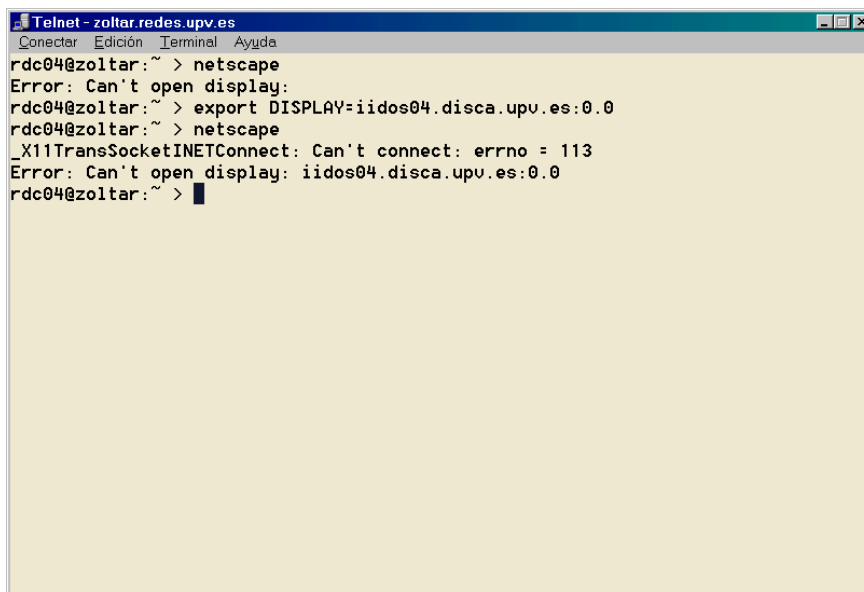
### PRACTICA 3: CONEXIÓN A OTRAS MÁQUINAS

Como se puede ver en la figura anterior, no es posible ejecutar el programa Netscape. La razón que nos da el sistema es que el “display” no se puede abrir. Esto es debido a que para ejecutar una aplicación gráfica, los servicios que proporciona la terminal remota (telnet) no son suficientes, ya que telnet sólo puede trabajar con caracteres. En consecuencia el sistema operativo remoto trata de abrir una terminal gráfica, que es otro tipo de terminal que sí que proporciona los servicios necesarios. Esta nueva terminal, obviamente, la debe abrir en nuestra pantalla. La forma que tiene el sistema remoto de saber dónde debe abrir la nueva terminal gráfica es mediante la variable del sistema “display”. Por tanto, lo que en realidad significa el mensaje de error de la figura anterior es que la variable display está vacía, por lo que el sistema remoto no sabe dónde debe abrir la nueva terminal gráfica.

Si queremos que la aplicación gráfica que intentamos ejecutar utilice nuestro monitor, entonces tenemos que cambiar el valor de la variable “display” para que “apunte” a nuestro ordenador. Para ello tenemos que ejecutar la siguiente orden en el sistema remoto:

```
export DISPLAY=iidos04.disca.upv.es:0.0
```

De esta forma le estamos indicando al sistema remoto que toda la salida gráfica la debe volcar a nuestro ordenador (que en el ejemplo hemos supuesto que es iidos04). Si tecleamos esta orden en nuestra conexión de telnet, y volvemos a ejecutar el Netscape, lo que obtendremos será algo similar a la figura siguiente:



```
Telnet - zoltar.redes.upv.es
Conectar Edición Terminal Ayuda
rdc04@zoltar:~ > netscape
Error: Can't open display:
rdc04@zoltar:~ > export DISPLAY=iidos04.disca.upv.es:0.0
rdc04@zoltar:~ > netscape
_X11TransSocketINETConnect: Can't connect: errno = 113
Error: Can't open display: iidos04.disca.upv.es:0.0
rdc04@zoltar:~ > █
```

Nuevamente no es posible ejecutar la aplicación gráfica, a pesar de que ahora el sistema remoto ya conoce dónde debe abrir la nueva terminal gráfica. El problema es que Windows y la aplicación gráfica que tratamos de ejecutar no están utilizando el mismo protocolo de comunicación gráfica, por lo que la aplicación gráfica no encuentra disponibles los servicios que solicita. Necesitamos, pues, dotar a Windows del protocolo de comunicación gráfica que solicita la aplicación. Este protocolo se llama “X”. Su origen se remonta a 1980, cuando en el MIT (Massachusetts Institute of Technology) se propusieron el desarrollo de un protocolo de comunicación gráfica distribuido e independiente del hardware. Hasta entonces todas las aplicaciones gráficas tenían que ejecutarse directamente sobre el hardware encargado de visualizar los resultados, es decir, sobre la propia máquina. No era posible trabajar desde tu ordenador en una máquina remota y ver los resultados de aplicaciones gráficas remotas, como puede ser Netscape, en tu propio ordenador.

**Ejercicio: Si no lo habeis hecho ya, desde vuestra máquina Windows abrid una sesión telnet y comprobad que no es posible ejecutar aplicaciones gráficas remotas. Cambiar el valor de la variable DISPLAY para que apunte a vuestra máquina y comprobar qué ocurre.**

### •Estructura del sistema X

A diferencia del entorno gráfico de Windows 9x, el protocolo X (también conocido como sistema de ventanas X) no forma parte del sistema operativo, sino que se ejecuta como un proceso de usuario más (recordemos que los sistemas Unix (Linux) admiten, a diferencia de Windows 9x, la multitarea real).

La estructura del sistema de ventanas X se divide en tres capas:

- **Servidor X:** programa encargado de aceptar peticiones de dibujo y realizar los dibujos pedidos sobre una terminal gráfica, la cual controla directamente. También tiene que gestionar los eventos tanto de ratón como de teclado y enviárselos a la aplicación correspondiente. El tipo de trabajo que es capaz de realizar este programa es de muy bajo nivel, por ejemplo, operaciones de trazado de rectas, relleno de áreas, dibujo de círculos, etc. Generalmente, el servidor X se ejecuta en la máquina que dispone de la terminal gráfica o monitor (es decir, se ejecuta en el ordenador local desde el que estamos conectados).
- **Gestor de ventanas:** programa que se encarga de ofrecer un entorno de trabajo más amigable a los usuarios y de administrar el uso del área de trabajo entre las distintas aplicaciones que en ese momento lo estén utilizando. El gestor de ventanas puede ejecutarse bien en la máquina local desde la que hemos establecido la conexión bien en la máquina remota. El que se ejecute en una máquina o en otra depende de las necesidades del usuario y de las posibilidades que ofrezcan cada una de las máquinas. Trabajaremos sobre esto más adelante.
- **Aplicaciones gráficas:** Son los programas que necesitan una interfaz gráfica para poder ejecutarse. A estos programas también se les conoce como aplicaciones **Xwindows**. Obviamente se ejecutan en la máquina remota. Un ejemplo de aplicación gráfica sería **Netscape**.

### •Administración de las aplicaciones X

Cuando nos conectamos desde nuestro PC (Windows) a una máquina remota Linux y tratamos de ejecutar una aplicación gráfica, esta aplicación debe volcar sus resultados en nuestra pantalla (utilizando la nomenclatura de la sección anterior, diríamos que la aplicación gráfica debe volcar sus resultados sobre nuestro servidor X). Si otro usuario del sistema Linux se conecta desde su PC y trata de ejecutar la misma aplicación gráfica, el resultado de la nueva ejecución debe volcarse sobre el servidor X del segundo usuario, no sobre el nuestro. De esta forma, si una aplicación puede tener una salida gráfica sobre varios servidores X ¿cómo se puede asociar una aplicación gráfica con el servidor X adecuado, que ha de representar los resultados de ésta?

La respuesta viene de la mano de la variable **DISPLAY**. Todas las aplicaciones X utilizan la variable de entorno **DISPLAY** para saber cuál es el servidor al que tienen que enviar los resultados o del que tienen que recoger los eventos de entrada (teclado y ratón). Si queremos que las aplicaciones que estamos ejecutando en el sistema remoto usen nuestro servidor X, tenemos que cambiar el valor de **DISPLAY** en la máquina remota para que “apunte” al servidor X destino (nuestro servidor X). Para ello tendríamos que ejecutar

la siguiente orden: `export DISPLAY=iidos04.disca.upv.es:0.0`, donde indicamos el nombre de la máquina desde la que estamos trabajando (iidos04.disca.upv.es) seguido de “:0.0”, que representa el primer servidor X que se ejecuta en nuestra máquina (es posible disponer de varios servidores X en una misma máquina, pero esto es ya otro tema).

Otra cuestión a explicar es que la variable de entorno **DISPLAY** no es común a todos los usuarios, es decir, cada usuario tiene la suya propia. Así pues, si un usuario trata de ejecutar aplicaciones gráficas remotas, y para ello modifica la variable **DISPLAY** como hemos descrito anteriormente, en realidad está modificando su propia variable **DISPLAY**. Por este motivo este cambio no se refleja en las variables **DISPLAY** del resto de usuarios. Esto permite que varios usuarios ejecuten la misma aplicación gráfica a la vez, y la salida de la aplicación se muestre en el monitor adecuado sin ningún tipo de interferencias entre los diferentes usuarios.

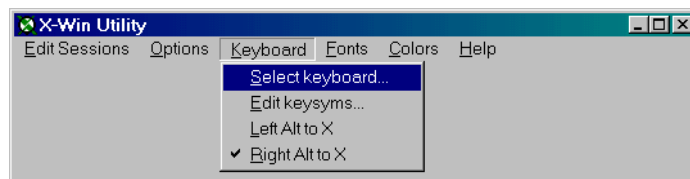
## **-X-Win**

En esta práctica vamos a utilizar el programa **X-Win** para dotar a Windows del protocolo de comunicación gráfica “X”, o dicho en otras palabras, vamos a utilizar el programa X-Win como servidor X. El programa se encuentra ya instalado en cada uno de los ordenadores del laboratorio. Podéis encontrarlo en la entrada **X-Win32** del menú de **Programas**. En lo que sigue vamos a ver cómo efectuar la comunicación gráfica entre Linux y Windows de dos maneras diferentes:

1. Usando como gestor de ventanas nuestro Windows 98 local.
2. Usando el gestor de ventanas de la máquina remota.

### **1) Usando el gestor de ventanas de nuestra máquina local**

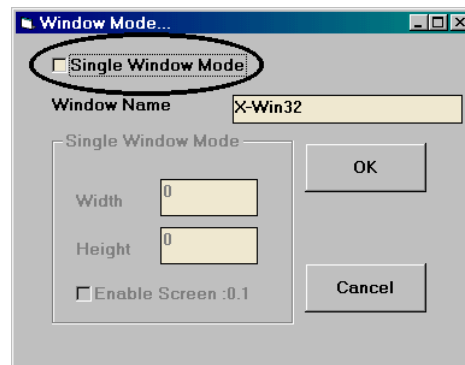
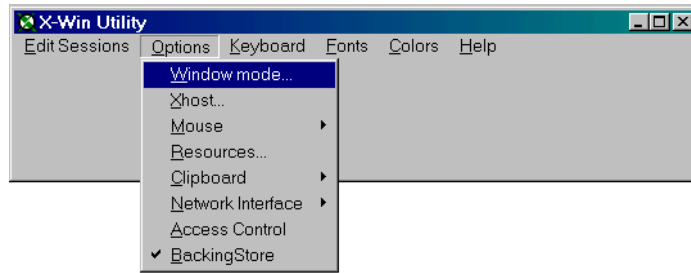
La idea detrás del enunciado de esta sección es conseguir ejecutar nuestro “Netscape” desde la ventana de telnet que hemos abierto anteriormente (si la has cerrado es el momento de abrirla de nuevo). Para conseguir ejecutar el “Netscape” (o cualquier otra aplicación gráfica), lo primero que debemos hacer es asignar el valor adecuado a la variable **DISPLAY**, de acuerdo a lo comentado en secciones anteriores. Por otra parte, también será necesario configurar adecuadamente la aplicación **X-Win32** (nuestro servidor X). Para ello, ejecutaremos el programa **X-Util**. Este programa presenta multitud de opciones, pero las que nos interesan en este momento son dos: el teclado y el modo de ventana.



En cuanto al teclado, debemos elegir el teclado castellano, en la correspondiente entrada del menú “**Keyboard**”. Respecto al modo de ventana, hay dos modos diferentes en lo que respecta a las ventanas que se van abriendo como resultado de ejecutar las aplicaciones gráficas. En el primer modo, llamado de **ventana múltiple**, las diferentes ventanas se abren directamente sobre nuestro gestor de ventanas (el Windows 9x). Esta forma de trabajar es a la que estamos acostumbrados. En el segundo modo, de **ventana única**, todas las ventanas que vayamos abriendo aparecerán dentro de la ventana del servidor X, de forma que, por ejemplo, al minimizar la ventana del servidor, todas las ventanas se ocultan a una. Este modo no es de gran utilidad en estos momentos, pero volveremos a él más tarde.

### PRACTICA 3: CONEXIÓN A OTRAS MÁQUINAS

Para seleccionar el modo de ventana debemos ir al menú “Options”, y en la ventana que aparece tras seleccionar la entrada “Window mode...”, deseleccionar la casilla “Single Window Mode”.



Una vez hayamos configurado el servidor X, debemos ponerlo en marcha, ejecutando el programa “X-Win 32”. Tras ejecutar el programa, tenemos el servidor X esperando a recibir peticiones por parte de aplicaciones gráficas remotas. Ahora es el momento de ejecutarlas. En la ventana de telnet que ya hemos abierto previamente, podemos probar a ejecutar el Netscape. Tras unos segundos aparecerá en vuestro monitor la ventana principal de Netscape. Hay que tener en cuenta que con aplicaciones gráficas remotas el retardo es mayor, pues la cantidad de información a transferir entre la máquina remota y vuestra máquina local es mucho mayor que en el caso de aplicaciones orientadas a carácter.

Otras aplicaciones que podéis probar a ejecutar son editores de texto mucho más potentes que el editor vi y también aplicaciones de edición de imágenes. Entre los editores que podéis ejecutar se encuentran “emacs” y “nedit” (este último es mucho más cómodo e intuitivo de usar que el emacs). Para editar imágenes podéis usar el programa “gimp”.

Otra posibilidad que ofrece el disponer de un servidor X en nuestra máquina Windows es la de usar el programa “xterm”. Este programa es un emulador de terminal remota, pero trabaja bajo la arquitectura X, en vez de trabajar bajo la arquitectura de Windows, como hace el cliente de telnet. Por este motivo resulta mucho más potente y cómodo de usar. Para usarlo basta con ejecutar en nuestra ventana de telnet el comando **xterm**. Aparecerá una nueva ventana en nuestra pantalla en la cual podemos introducir comandos como hacíamos en la ventana de telnet que ya teníamos abierta. Sin embargo, ahora las posibilidades son mayores, como vimos en la práctica 2. Podemos cambiar el tamaño de letra si pinchamos con el botón derecho en la ventana “xterm” al mismo tiempo que pulsamos la tecla “Control” de nuestro teclado. También podemos hacer más grande la ventana, para tener más área de trabajo. De esta forma, al listar el contenido de un directorio podemos retener más líneas en la ventana. En caso de que algunas líneas hayan desaparecido por la parte superior, con el scroll que hay en la parte izquierda podemos acceder a ellas. También podemos hacer scroll con la combinación de teclas “SHIFT+Re Pag” y “SHIFT+Av Pag”, y copiar y pegar texto, como vimos en la práctica de introducción a Linux.

La aplicación xterm es más potente que el cliente de telnet que incorpora Windows y no tiene tantos problemas de compatibilidad como tiene este último. Por ejemplo, si tratamos de ejecutar el comando “top” en el cliente de telnet, para que nos liste los

procesos que más CPU están consumiendo en el sistema, la salida de este comando quizá no se visualice correctamente. Con una xterminal esto no ocurre. Si ejecutamos el comando top en nuestra xterminal, comprobaremos que ahora sí que obtenemos la salida completa del comando.

**Ejercicio: Antes de pasar a la siguiente sección, configurad el programa X-Win de acuerdo a lo visto en esta sección, ejecutando diferentes aplicaciones gráficas en modo remoto. Asimismo, probad a configurar el servidor X en modo de ventana única, y observad cuál es la diferencia en el uso de las ventanas. Probad a abrir varias aplicaciones gráficas al mismo tiempo.**

## 2) Usando el gestor de ventanas de la máquina remota

Hemos comentado que en la arquitectura del sistema X hay tres capas:

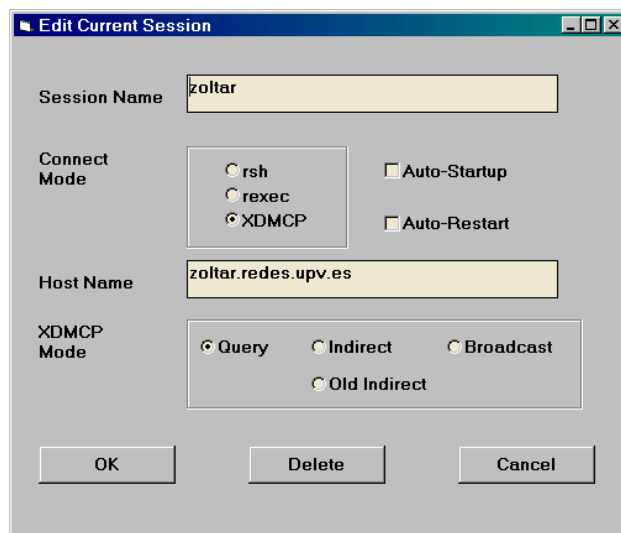
- El servidor X, que se ejecuta en la máquina local desde la que estamos haciendo la conexión
- El gestor de ventanas, que se puede ejecutar en la máquina local o en la máquina remota
- Las aplicaciones X, que se ejecutan en la máquina remota

En la sección anterior hemos ejecutado aplicaciones X remotas utilizando el gestor de ventanas de nuestra máquina local, esto es, hemos utilizado Windows como gestor de ventanas. De esta forma, todas las ventanas que las aplicaciones remotas creaban eran gestionadas directamente por Windows. Otra posibilidad es ejecutar el gestor de ventanas de la máquina remota. En este caso las ventanas son gestionadas por la máquina remota. Además, todo el interfaz gráfico también es gestionado por la máquina remota.

Para usar el gestor de ventanas remoto tenemos que configurar el servidor X de nuestra máquina con la aplicación “X-Util”. Hay que configurar que el modo de ventana sea “Single Window Mode”. Por otra parte, en el menú “Edit Sessions” debemos ir a la opción “New Session ...” y configurar la nueva sesión de acuerdo a la siguiente figura:

En esta ventana debemos configurar que el modo de conexión va a ser “XDMCP Query” y que el host al que nos vamos a conectar es zoltar. De esta forma estamos indicando que cuando arranque el servidor X en nuestra máquina local, haga una petición a la máquina remota (zoltar) para iniciar allí el gestor de ventanas.

Ahora podemos iniciar la sesión con zoltar ejecutando el servidor X. Pinchando con el botón derecho en su entrada en la barra de tareas aparece el menú correspondiente. En la entrada “Sessions” debemos elegir la opción “zoltar”. Tras unos segundos aparecerá en nuestro monitor, a pantalla completa, la pantalla de bienvenida a zoltar. Tras introducir el login y el password adecuados, entraremos en el gestor de ventanas de la máquina remota. Obviamente, la velocidad de actualización del escritorio no es la misma que si estuviéramos trabajando en forma local, pues en nuestro caso los datos han de atravesar toda una red.



**Ejercicio: En el menú “Edit Sessions” del programa X-Util cread una nueva sesión con los datos de la figura anterior. Conectaos a Zoltar utilizando la nueva sesión y comprobad la diferencia de funcionamiento respecto al modo de trabajo presentado en la sección anterior.**

## Parte II: Linux → Linux

En esta segunda parte de la práctica vamos a estudiar los mismos aspectos que hemos revisado en la primera parte, con la diferencia de que ahora nuestro equipo local usará Linux. Por tanto, hay que reiniciar el ordenador para que pase a Linux. Una vez dentro de nuestra sesión de Linux, debemos iniciar el entorno gráfico.

### Telnet

La aplicación telnet en linux es similar a la que hemos visto en Windows. Para ejecutarla no hay más que teclear **telnet** en una ventana de *shell*.

**Ejercicio: Inicia una sesión de telnet en zoltar, con el login rdcxx, xx dependiendo del puesto en el que estés trabajando.**

En un entorno en el que hay varios usuarios, es habitual que éstos quieran comunicarse entre ellos. En Linux podemos hacerlo de dos formas diferentes: mandando un mensaje, o bien, manteniendo una conversación.

Para enviar un mensaje a otro usuario debemos usar la orden "**write**", seguida del nombre del usuario al que queremos enviar el mensaje. Una vez hayamos tecleado la orden, el sistema se queda esperando a que escribamos el mensaje. Cada vez que pulsamos la tecla de retorno de carro, la línea correspondiente aparece en el terminal del otro usuario. Para finalizar el mensaje hay que pulsar la combinación de teclas **control+D**.

Si no queremos ser molestados mientras trabajamos podemos deshabilitar la recepción de mensajes con la orden "**mesg n**". Para volver a habilitar la recepción de mensajes debemos usar la orden "**mesg y**".

La segunda forma de comunicarnos con otros usuarios es mediante la orden "**talk**". Esta orden nos permite mantener una conversación con otro usuario, que no necesariamente tiene por qué estar conectado a la misma máquina a la que estamos conectados nosotros. La sintaxis de la orden es "**talk nombre\_usuario@nombre\_maquina**". A diferencia del comando "**write**", con el que enviábamos un mensaje sin establecer una conexión (algo similar a enviar una carta), con el comando "**talk**" hay que establecer una conexión, es decir, ambos extremos deben ponerse de acuerdo para conversar (algo parecido a llamar por teléfono). Cuando un usuario ejecuta la orden **talk**, el usuario destino debe ejecutarla también (con el símil del teléfono sería descolgar el teléfono). Por tanto, el extremo que recibe la llamada puede decidir no contestarla, y en este caso no se llevaría a cabo la comunicación. Para finalizar la conversación hay que pulsar la combinación de teclas **control+C**.

### Ftp

La aplicación ftp de Linux funciona exactamente igual que la hemos visto en Windows. La forma de ejecutarla es mediante la orden **ftp**. La principal diferencia estriba en el entorno en que ejecutamos la orden. Puesto que la ventana en la que la ejecutamos ftp es una ventana de *shell*, podemos utilizar todas las ventajas vistas (scroll, tamaño de letra, copiar y pegar texto, etc.). Estas posibilidades en Windows no existían.

## PRACTICA 3: CONEXIÓN A OTRAS MÁQUINAS

Como hemos comentado anteriormente, hay disponibles muchos clientes de ftp. Linux incorpora un cliente de ftp con interfaz gráfica, llamado **xftp**, cuya ventana principal se muestra en la figura de más abajo. La principal diferencia con respecto al cliente que hemos visto antes (además del entorno gráfico, que facilita mucho las cosas) es que es capaz de transferir directorios completos, manteniendo su estructura de árbol de directorios.



**Ejercicio: Ejecuta xftp y conéctate a zoltar. Prueba a traerte alguno de los directorios del sistema.**

## Ssh

En Linux también está disponible la aplicación ssh. De hecho suele venir de serie con el sistema operativo. Para ejecutarla tenemos que teclear **ssh**. El interfaz es un tanto diferente al visto en Windows, pero el funcionamiento es el mismo.

## Conexión remota gráfica

Cuando nos conectamos de una máquina Linux a otra y queremos ejecutar alguna aplicación gráfica, como Netscape, el sistema remoto nos devuelve el mismo resultado que cuando lo hacíamos desde Windows: no conoce dónde debe representar los resultados. Sin embargo, en este caso la solución es mucho más sencilla, pues no necesitamos dotar a nuestro sistema Linux de un servidor X, como debíamos hacer en Windows, pues Linux ya incorpora uno. Por tanto, lo único que debemos hacer es ajustar el valor de la variable **DISPLAY** remota (**export DISPLAY=iidos04.disca.upv.es:0.0**). Adicionalmente, debemos autorizar al sistema remoto a que utilice nuestra pantalla, o más exactamente, nuestro servidor X (si no fuera necesaria esta autorización, cualquiera podría usar nuestra pantalla). La forma de autorizar al sistema remoto a usar nuestra pantalla es mediante la orden **xhost**, ejecutada en nuestro sistema local. Como parámetro de la orden xhost debemos poner el nombre de la máquina a la que estamos autorizando a usar nuestro servidor X. La orden sería: **xhost zoltar.redes.upv.es**.