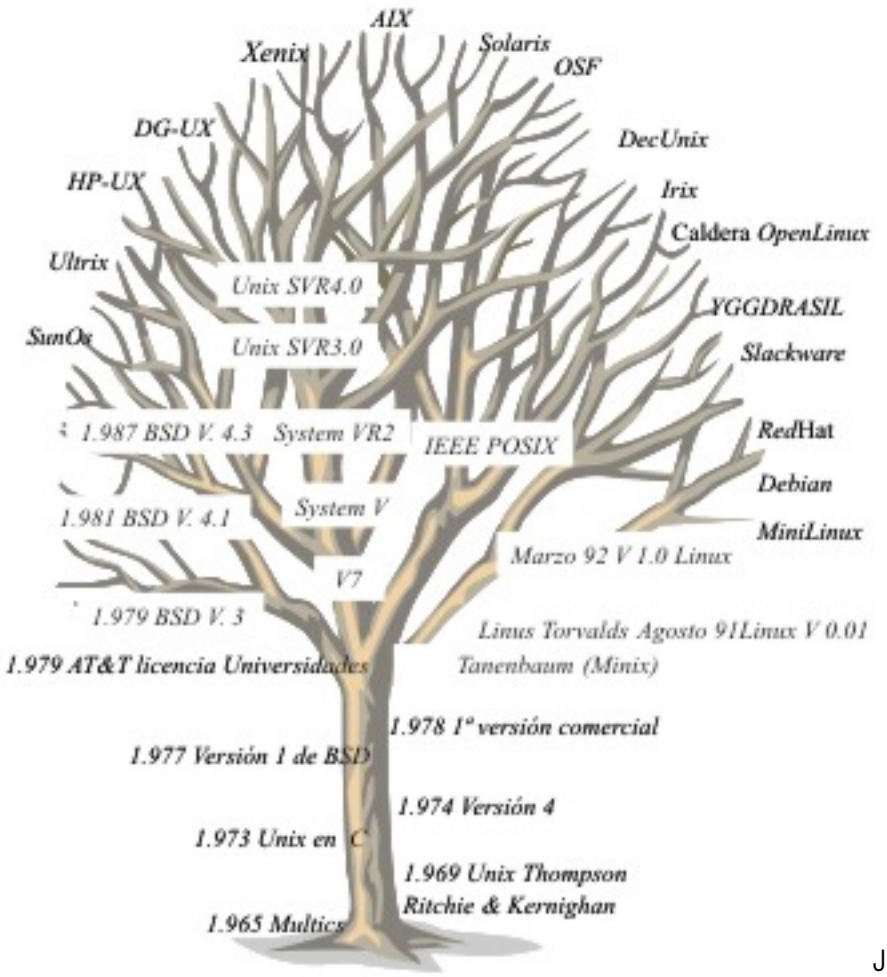
Administración Avanzada de Redes de Sistemas UNIX/LINUX

Un enfoque práctico



Javier Carmona Murillo
David Cortés Polo
Alfonso Gazo Cervero
José Luis González Sánchez
Manuel Domínguez Dorado
Francisco Javier Rodríguez Pérez

ISBN: 978-84-695-2036-9

Administración avanzada de redes de sistemas UNIX/LINUX

Un enfoque práctico.

Javier Carmona Murillo
David Cortés Polo
Manuel Domínguez Dorado
Alfonso Gazo Cervero
José Luis González Sánchez
Francisco Javier Rodríguez Sánchez

1ª edición: Enero de 2012
ISBN13: 978-84-695-2036-9

Objetivos del libro

• *UNIX/Linux* se ha convertido, con el paso de los años, en uno de los sistemas operativos más extendido. Sus características abiertas (no propietario y portable) y su íntima relación con Internet, han permitido que, después de varias décadas, siga siendo, en sus múltiples distribuciones, el sistema multiusuario más empleado en todo el mundo.

• *UNIX/Linux* es un sistema multiusuario, multitarea, multiplataforma y de propósito general que requiere de la existencia de un administrador de sistemas para garantizar su óptimo rendimiento. El Administrador de sistemas *UNIX/Linux* es uno de los perfiles profesionales informáticos más solicitados desde hace años. El libro pretende la formación de este tipo de técnico informático de forma que, al concluir el libro, se tengan las nociones avanzadas que todo administrador de sistemas *UNIX/Linux* debe conocer para desempeñar correctamente su misión.

• Los planteamientos teóricos del libro son referentes a *UNIX* en general, mientras las prácticas se realizan, mayoritariamente, sobre Solaris 10 de SUN y sistemas *Linux* en red (Ubuntu y Fedora Core).

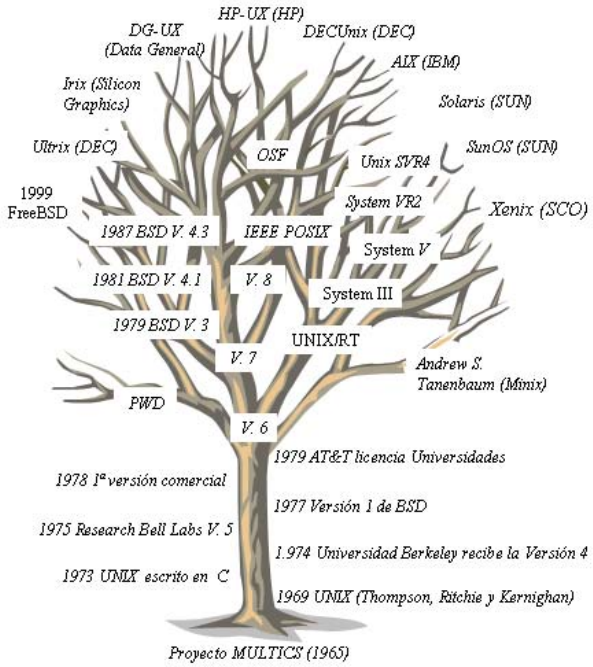
Administración Avanzada de Redes de Sistemas UNIX/Linux

Índice

Introducción, evolución y administración de los S.O. UNIX/Linux

- Orígenes, evolución y situación actual del S.O. UNIX.
- Características principales de UNIX.
- El fenómeno del S. O. Linux.
- La figura del Administrador de Sistemas.

Orígenes, evolución y situación actual del S.O. UNIX



Características principales de UNIX

• **Multitarea**:*(multitasking)* Ejecución de tareas simultáneamente en tiempo compartido.

  • *Multitarea prioritaria (tiempo compartido).* Cada proceso del sistema tiene garantizada la oportunidad de ejecutarse hasta que el S.O. da la oportunidad a otro proceso (*UNIX*, *Linux*, *Win-NT*, *Win95*).

  • *Multitarea cooperativa*. Los procesos trabajan hasta que acaban o voluntariamente dejan a otros procesos (*MS-DOS*, *MS-Windows 3.x*).

• **Multiusuario**: *Unix*, *Win-NT*, *VMS*, *MVS*, *Netware*, *OS-400*...

Administración Avanzada de Redes de Sistemas UNIX/Linux

Características principales de UNIX

• **Multiprocesador** y **multiplataforma**.

• **Terminales virtuales.**

• ***Shells* programables (***Bourne*, *C*, *Bash*, *Korn*...**).**

• **Redirección** de E/S y errores, ***pipes***, ***background***...

• **Independencia de dispositivos**:

Dispositivos=ficheros, se enlazan (controlador de dispositivos) con el *Kernel*.

Linux también, pero mayores problemas por reconocer menos dispositivos que *Unix* (“todo se andará”).

Características principales de UNIX

- **Sistema de ficheros**.
- ***Networking*: *TCP/IP*** (nativo) nace en *Unix*.
- **Portabilidad** del código entre distintos *UNIX*. Estandarización y ahorro de costes y fácil migración de unos fabricantes a otros. El argumento de los costes.
- **Sistema abierto** (*POSIX*) e interoperatividad.
- *Unix* **comercial**: *Solaris, DEC-Unix, SunOS, DG-UX, Irix, HP-UX, AIX, SCO Xenix, MAC-UX...*
- Mecanismos de **protección de memoria** entre procesos.

Características principales de UNIX

- **Herramientas** de administración del sistema.
- **Gestión de memoria compartida:** mayor velocidad y menor volumen de memoria ocupada.
- **Memoria virtual con paginación a disco (*swapping*).**
- **Memoria dinámica** para la carga de programas o para caché a disco, definible por el administrador.
- **Librerías** estáticas y dinámicas.
- Posibilidades de **auditoría** del código fuente.

Características principales de UNIX

- ***FreeBSD:***
  - S.O. para arquitecturas x86 compatibles (incluyendo Pentium® y Athlon™), amd64 compatibles (incluyendo Opteron™, Athlon 64 y EM64T), UltraSPARC®, IA-64, PC-98 y ARM.
  - Derivado de BSD, la versión de UNIX® desarrollada en la Universidad de California, Berkeley.
  - Licencia FreeBSD. Más restrictiva GPL y LGPL
  - BSD Family Tree: FreeBSD, NetBSD y OpenBSD
  - *http://www.freebsd.org*

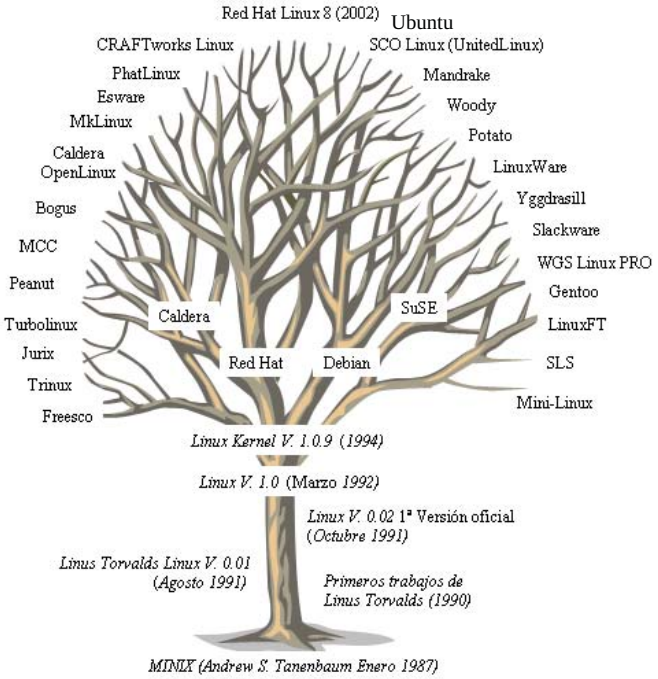
El fenómeno del S.O. Linux

- Linus Torvalds (1991) amplía *Minix* hasta convertirlo en una réplica perfecta de *Unix* para *PC*.
- Desde sus inicios incorpora bibliotecas de utilidades GNU y el sistema *GUI X-Window*.
- Sin derechos de autor y libre para todo el mundo. Aplicaciones *freeware*.
- *Es multi[usuario,tarea,media,plataforma]*, no depende de estrategias de marketing pero ¿tampoco? aporta servicio técnico.
- Evolución vertiginosa.

El fenómeno del S.O. Linux

- *Linux* ofrece libremente:
  - Compiladores (*C, C++, SmallTalk, Flex Bison*, etc.).
  - Bases de Datos.
  - Ofimática (OpenOffice).
  - Internet (navegadores, clientes correo, *news*, etc.).
  - Todo tipo de herramientas:
    - Administración.
    - Audio y video.
    - Seguridad.

Administración Avanzada de Redes de Sistemas UNIX/Linux



El fenómeno del S.O. Linux

• Distribuciones Linux (miles de usuarios trabajando en objetivo común):

• ***RedHat****:* Fácil de instalar y administrar. En punta. Gestión de paquetes con *RPM*. Soporte técnico.

• ***Slackware***: De las más veteranas en el ámbito comercial. La primera hasta aparecer *Red Hat*. Consume pocos recursos.

• ***Debian***: Quizás la más profesional (muy completa y pensada para informáticos) y mejorable para el uso del gran público.

Administración Avanzada de Redes de Sistemas UNIX/Linux

El fenómeno del S.O. Linux

• Distribuciones Linux (miles de usuarios trabajando en objetivo común):

• ***Caldera OpenLinux:*** Muy comercial y algo cara. Basada en RedHat enriquecida con paquetes como *Corel, Wordperfect, Wabi...*

• ***Yggdrasil***: Orientada a info. multimedia reconociendo muchos dispositivos. Live CD. Soporte técnico.

• ***MCC***: Versión 1.2+, última versión de Junio 95 (poco extendida).

El fenómeno del S.O. Linux

• Distribuciones Linux (miles de usuarios trabajando en objetivo común):

• ***Bogus:*** Versión 1.0.1 actualizada irregularmente (última Julio 94).

• ***SLS:*** Primera gran distribución de Linux. Decayendo (Nov. 94).

• ***Mini-Linux:*** Versión reducida para soporte sobre MS-DOS.

• ***Jurix:*** Corriente alemana bastante innovadora pero poco depurada.

Administración Avanzada de Redes de Sistemas UNIX/Linux

El fenómeno del S.O. Linux

• Distribuciones Linux (miles de usuarios trabajando en objetivo común):

• ***Craftworks Linux****:* Distribución comercial para Intel y AXP. RPM

• ***Linux Pro:*** Impulso para que Linux sea el único y verdadero S.O. Una de las mejores distribuciones.

• ***SuSE***: Similar a *Red Hat*. *RPM*, fácil de instalar, no permite copias con fines comerciales.

• ***Eurielec, Gentoo, Mandriva, LUCAS*** (*Linux* en castellano), etc...

El fenómeno del S.O. Linux

- Distribuciones Linux (miles de usuarios trabajando en objetivo común):
  - ***Knoppix:*** Popular versión LiveCD
  - ***Ubuntu:*** *oo-Bun-Tu "humanidad hacia otros".* Oct.'04
    - Completamente de fuente abierta con nucleo Linux.
    - Nueva versión cada seis meses. Basado en Debian.
    - Arquitecturas i386 (procesadores 386/486/Pentium(II/III/IV) y Athlon/Duron/Sempron), AMD64 (Athlon64, Opteron y los nuevos procesadores Intel de 64 bits), PowerPC (iBook/Powerbook, G4 y G5).
    - Gnome por defecto y Kubuntu opcional.

El fenómeno del S.O. Linux

- Versiones de Linux más actualizadas en Internet:
- ***News****: comp.os.linux.announce, comp.os.linux.answers comp.os.linux.setup*
- ***FAQs****:sunsite.unc.edu /pub/Linux/docs*
- ***HOWTOs****: ftp://nic.funet.fi/pub/Linux/doc*
- ***FTP****: ftp://sunsite.unc.edu/pub/linux*
  ftp://nic.funet.fi/pub/Linux
  ftp://ftp.rediris.es/pub/linux
  ftp://ftp.mcc.ac.uk/pub/linux

El fenómeno del S.O. Linux

- Versiones de Linux más actualizadas en Internet:
- ***WEB:*** *http://www.ubuntu.com/* *Pagina de Ubuntu.*
  http://www.linux.org.uk *Linux online*
  http://www.redhat.com *Red Hat*
  http://www.debian.com *Debian*

- Licencias GNU, software libre y software de código abierto.

Administración Avanzada de Redes de Sistemas UNIX/Linux

La figura del Administrador de Sistemas

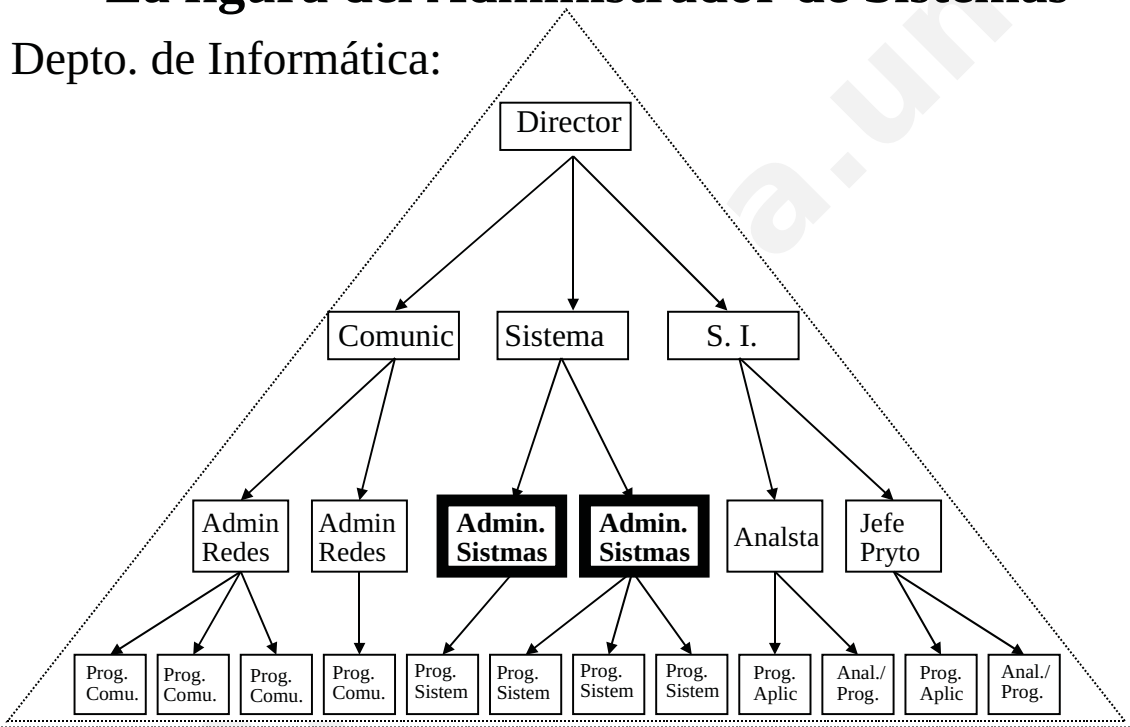
- Administración de usuarios (A,B,M).
- Configuración de dispositivos (HDs, CDs, FDs, impresoras,terminales, cintas, etc.).
- Garantizar la seguridad y correcto funcionamiento del sistema
- *Networking*.
- Coexistencia con otros sistemas.
- Labores de mantenimiento.
- El administrador debe conseguir que el sistema sea: Eficiente, seguro y fiable.
- Planificación: tecnológica y de inversiones.
- Coordinación: Dptos. desarrollo, comunicaciones, BD y SI.

La figura del Administrador de Sistemas

- Reparto equitativo de recursos.
- El *system manager* debe ser: diplomático, hábil, técnico, responsable, formador consejero y "bombero" muy a menudo.
- Responsabilidades y "juez y parte", Código deontológico.
- Aspectos legales y responsabilidades:
  - LSSI (34/2002 de 11 de Julio).
  - LOPD (15/1999 de 13 de Diciembre).
  - Ley Firma Electrónica (59/2003 de 19 de Diciembre).
  - Ley Gral. Telecos (32/2003, de 3 de Noviembre).
  - Código Penal.
  - Peritajes informáticos.

La figura del Administrador de Sistemas

- Depto. de Informática:



La figura del Administrador de Sistemas

- La labor de Administración del sistema depende de:
  - Equipos (sistemas y red).
  - Periferia.
  - Usuarios (Nº y características).
  - Aplicaciones.
  - Actividad de la organización.
  - Organigrama de la organización.

Administración Avanzada de Redes de Sistemas UNIX/Linux

La figura del Administrador de Sistemas

- Tareas concretas en UNIX/Linux:
  - Instalación y configuración del sistema: hardware del servidor, configuración RAID, características del SO, etc.
  - Gestión de usuarios y grupos: perfiles, permisos y NIS.
  - Sistema de archivos: particiones, permisos, cuotas, NFS.
  - Administración de la red: configuración de la red, de routers, archivos y dispositivos compartidos, Samba.
  - Servicios básicos de red: DNS, DHCP, de terminales.
  - Servicios Internet: Web, Correo.e, FTP, etc.

La figura del Administrador de Sistemas

- Tareas concretas en Unix/Linux:
  - Monitorización del sistema: *tunning* y monitorización del rendimiento de recursos y gestión de alertas del sistema.
  - Control de la seguridad del sistema.
  - Copias de seguridad: tipos de copias y restauraciones. Planificación de las copias.
  - Parcheo, compilación y personalización del kernel.
  - Parada y arranque del sistema.
  - Procesos y *daemons*.

Instalación

Instalación

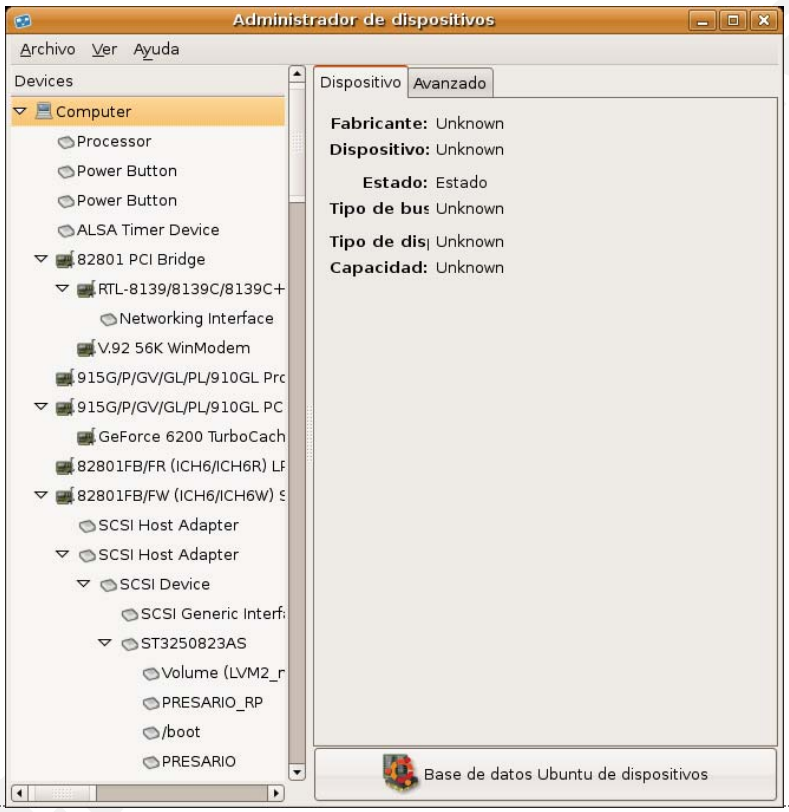
- Instalación del Sistema
- Pre-instalación (planificación)
- Dispositivos de tipo disco
- Etiquetas de particiones
- Áreas de Swap
- GRand Unified Bootloader (GRUB)
- Initial RAM Disk (initrd)
- Instalación gráfica de Ubuntu
- Instalación en modo texto de Ubuntu
- Particionado
- Arranque, coexistencia y parámetros del kernel

Pre-instalación (planificación)

- Antes de comenzar con la instalación del sistema en sí, debemos determinar con qué *hardware* se cuenta.
- A pesar de que el programa de instalación trate de detectar automáticamente el *hardware* del sistema, es posible que no pueda determinar algunos parámetros, especialmente si el equipo no es moderno.
- Si el sistema va a conectarse a una red, deberemos consultar al administrador cuáles son los parámetros de configuración
- Si en el sistema van a coexistir varios sistemas operativos, debemos conocer cuáles son y qué cantidad de disco es necesaria para ellos.

Determinar si el hardware es compatible

- Para conocer el hardware del equipo, en Ubuntu, se puede utilizar el Administrador de dispositivos.
- En ocasiones puede ser necesario utilizar otro sistema operativo para la identificación del hardware que no se detecte correctamente.
- Posibles problemas:
  - Tarjetas gráficas
  - Tarjetas Wireless
  - Winmodems
  - Dispositivos USB
  - Gestión energía, sensores, etc.
  - Y mucho más en equipos portátiles.
- https://wiki.ubuntu.com/HardwareSupport



Dónde instalar el sistema

- Si el equipo no tiene sistema operativo y en la planificación se decide que no va a tener ningún otro salvo el que se instale, la instalación sueles ser simple.
- Si no,
  - Crear espacio para el nuevo sistema:
    - Disminuir espacio ocupado por el anterior (ntfsresize, resize2fs, resize_reiserfs, volúmenes lógicos).
    - Utilizar espacio libre.
  - Determinar cuál será el gestor de arranque:
    - Cada sistema tiene el suyo propio
    - A veces es necesario utilizar alguno de forma específica

Dispositivos de tipo disco (Linux)

Para poder crear y modificar las tablas de particiones, debemos conocer cuál es el nombre del dispositivo que se corresponde con el disco duro que queremos modificar.

El nombre del dispositivo depende tanto del sistema operativo utilizado como del modo en el que esté conectado el disco duro.

El nombre del dispositivo se determina en función de:

- Si el disco duro es **IDE**, el nombre de dispositivo comienza por **hd**:
  - Puerto primario, disco maestro ⇒**hda**
  - Puerto primario, disco esclavo ⇒**hdb**
  - Puerto secundario, disco maestro ⇒**hdc**
  - Puerto secundario, disco esclavo ⇒**hdd**

- Si el disco es **SCSI** ó **SATA**, el nombre del dispositivo comienza por **sd**.
  - Para el ID 0, ⇒**sda** Para el ID 1, ⇒**sdb** ...

Dispositivos de tipo disco (Solaris)

En Solaris existen dos formas de acceder al espacio de almacenamiento:

- /dev/rdsk/*: Acceso en modo *raw*. La comunicación con el dispositivo se realiza directamente a través del *driver* y las lecturas/escrituras deben efectuarse de acuerdo con un tamaño de bloque.
- /dev/dsk/*: Acceso en modo *cooked*: La comunicación con el dispositivo se realiza a través de funciones de alto nivel. Las peticiones pueden almacenarse temporalmente en *búffers*, e incluso cambiar su orden y no existe un tamaño fijo para las lecturas/escrituras.

/dev/rdsk/c0t0d0s0

- c0: (**c**ontroller) indica el primer controlador.
- t0: (**t**arget) indica el primer dispositivo conectado al controlador.
- d0: (**d**evice) indica el primer LUN. Raramente se usa.
- s0: (**s**lice) indica el primer *slice* o partición.

Asignación de espacio en dispositivos de almacenamiento

- Identificados los discos del sistema, queda asignar espacio para el sistema operativo.
- En UNIX, distintas partes del sistema de archivos se almacenan en distintas partes del disco.
- Esta separación puede hacerse mediante:
  - Particiones
  - Gestión de volúmenes lógicos
- ¿Problemas de coexistencia entre varios sistemas operativos en un mismo equipo?
  - Más relacionados con los distintos tipos de sistemas de archivos utilizados.
  - Aunque selección del arranque también es fuente de problemas.

Etiquetas de particiones

- Las etiquetas de particiones sirven para describir las particiones de un dispositivo de almacenamiento.
- DOS partition table:
  - Máximo de cuatro particiones primarias.
  - Puede existir una partición extendida.
  - Dentro de partición extendida se pueden crear hasta 4 particiones lógicas.
- Solaris disk label (VTOC):
  - 8 “slices”.
  - 0: root, 1:swap, 6: usr
  - 2: representa el disco completo.
- ...

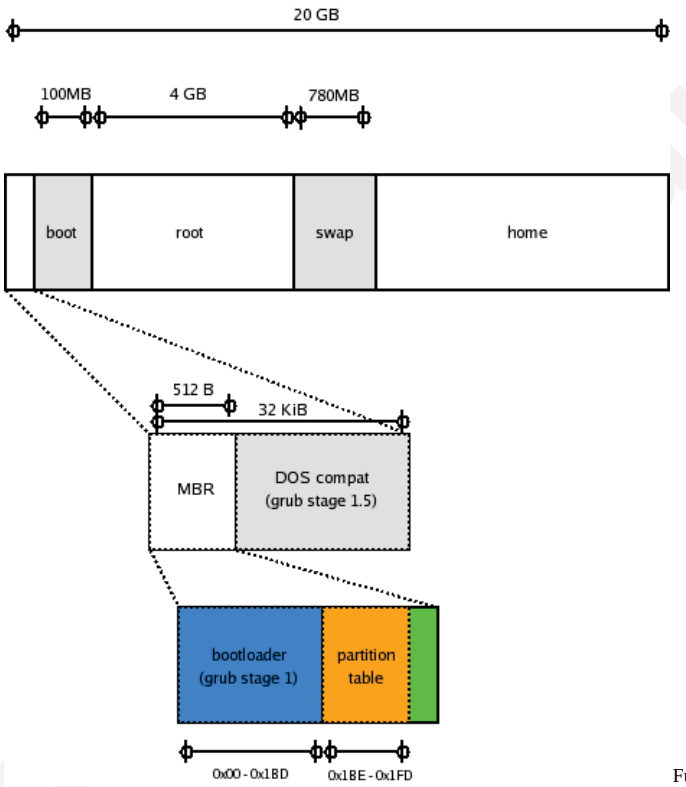
Denotación de particiones (Etiqueta DOS)

Una vez conocido el nombre del dispositivo que se corresponde con el disco duro, para designar una partición, simplemente añadiremos el número de partición al final del nombre del dispositivo. Es decir:

- La partición 1 del disco IDE conectado en el puerto primario como maestro es: `hda1`

Para poder crear más de cuatro particiones en un solo dispositivo, se debe crear una (sólo una) partición extendida.

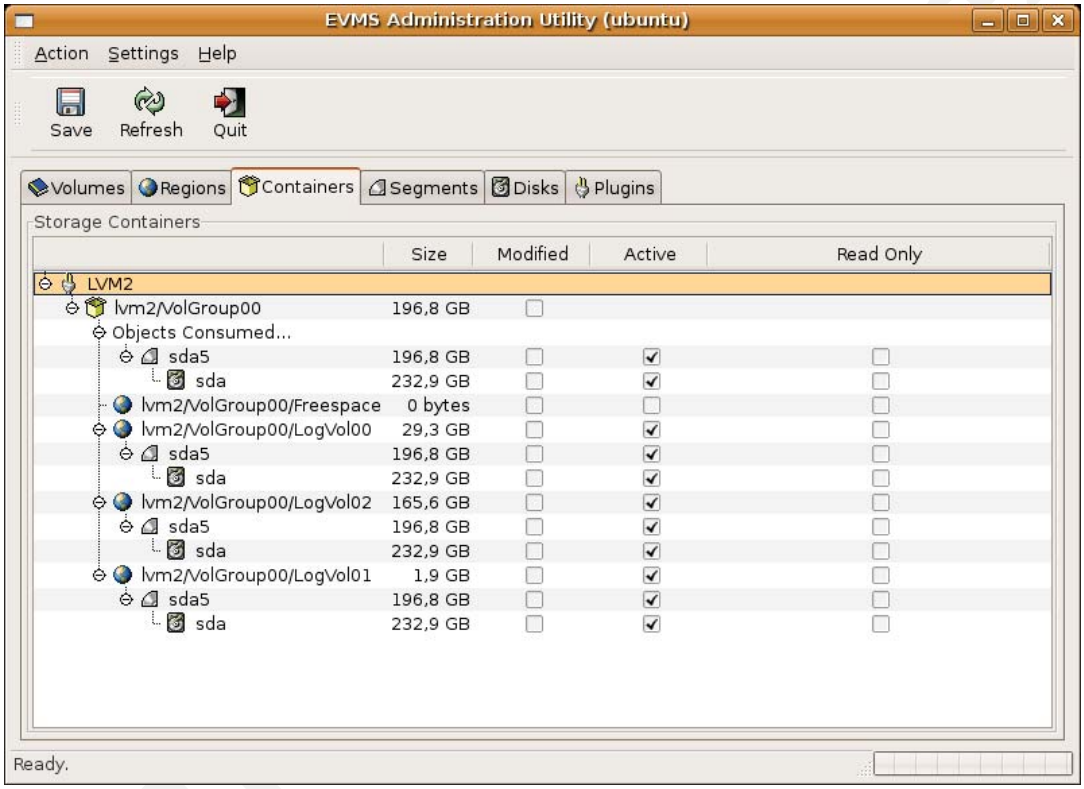
Sobre una partición extendida se pueden crear hasta cuatro particiones lógicas, que se numeran de 5 a 8 (`hda5..hda8`)



Fuente: http://www.pixelbeat.org/docs/disk/

Logical Volume Manager (LVM2) & Enterprise Volume Management System (EVMS)

- Si se usa, los sistemas de archivos residen en “volúmenes lógicos”
- ¿Por qué usarlo? Permite:
  - Aumentar o disminuir el tamaño de los VL.
  - Añadir, eliminar, unir y separar VL.
  - Crear, copiar, mover VL en distintos dispositivos físicos.
  - Crear snapshots: backups sin pasar a monousuario.
- ¿Por qué no usarlo?:
  - Recuperación del sistema más compleja.
  - No todos los gestores de arranque lo soportan.
  - Faltan GUI en algunas distribuciones Linux. Otras incorporan.



Área de swap

Salvo excepciones, el sistema requerirá que se designe un área para poder utilizarla en caso de que necesite realizar *swapping* a disco.

Tanto en *Linux* como en *Solaris* se permite que el área de *swap* se encuentre en una partición o en un fichero.

Determinación del tamaño del área de *swap*.

Convención: Doble del tamaño de la memoria principal ¿?

Consideraciones:

- Tamaño de disco y de memoria principal disponibles
- Número usuarios estimados y sus perfiles (Índice simultaneidad)
- Tamaño (KB) de aplicación más utilizada
- Número máximo de ficheros (búffers) abiertos

Partición de swap

La razón principal que nos llevará a elegir una partición como área de *swap* es que el acceso a una partición de *swap* es más rápido que a un fichero de *swap*, ya que evitamos, entre otros problemas, que el fichero de *swap* se fragmente.

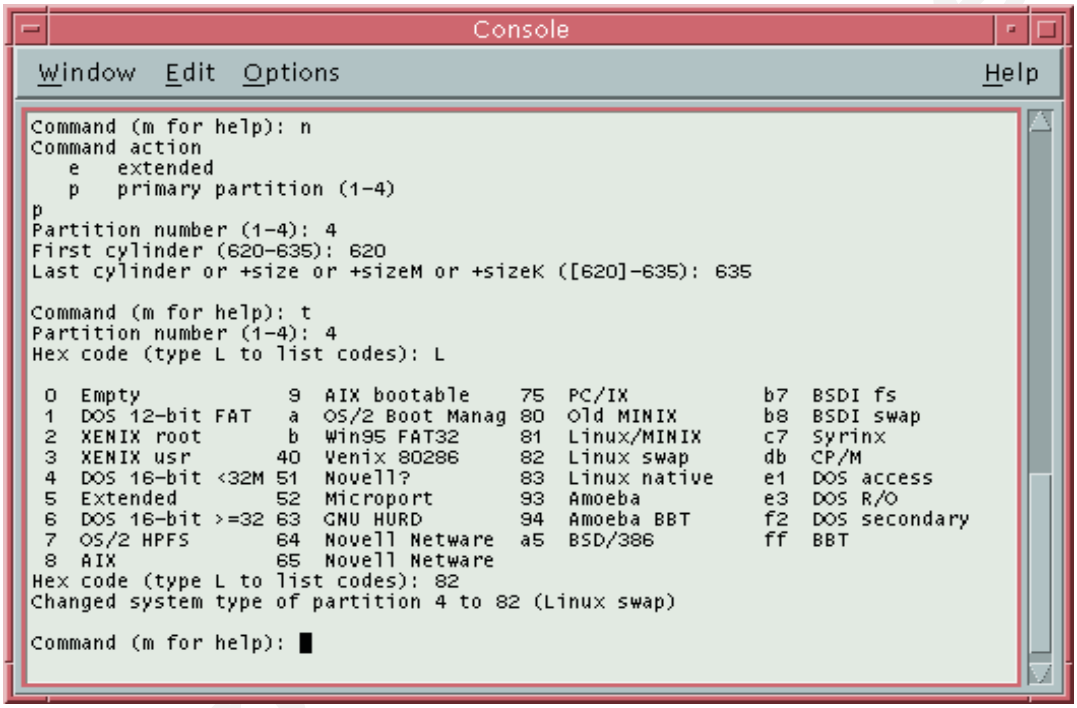
Puesto que tanto *Linux* como *Solaris* permiten la existencia de más de un área de *swap*, se puede utilizar una partición como área de *swap* y, en caso de que el tamaño calculado para el área resulte insuficiente, asignar un fichero de *swap* auxiliar, de forma **temporal**.

Creación de una partición de swap

Lo único que diferencia a una partición de *swap* del resto es su *ID*:

```
Disk /dev/hda: 128 heads, 63 sectors, 635 cylinders
Units = cylinders of 8064 * 512 bytes

   Device Boot    Begin     Start       End    Blocks   Id  System
/dev/hda1             1         1       254  1024096+  83  Linux native
/dev/hda2           255       255       508  1024128   83  Linux native
/dev/hda3           509       509       619   447552   83  Linux native
/dev/hda4           620       620       635    64512   82  Linux swap

Command (m for help): 
```

A continuación se indica el procedimiento para crear una partición de *swap*



Administración Avanzada de Redes de Sistemas UNIX/Linux

Fichero de swap

Para crear un fichero de *swap*, se debe ejecutar el siguiente comando:

```
#dd if=/dev/zero of=swapfile \
bs=1024 count=16384
```

Ruta y nombre del fichero de *swap*

Tamaño (en KB) del fichero de *swap*

Una vez creada la partición o el fichero de *swap*, tendremos que formatear el área, utilizando el siguiente comando:

```
#mkswap [nombre de la partición/fichero]
```

Y activarla, con el comando:

```
#swapon [nombre de la partición/fichero]
```

GRand Unified Bootloader (GRUB)

- Implementación de referencia de la especificación Multiboot.
- Configurable dinámicamente: carga su configuración durante el arranque. Interactivo.
- Stage 1, 1.5, 2
  - 1: (en MBR). Para acceder al stage 1.5.
  - 1.5: (en espacio de compatibilidad con DOS). Acceso al sistema de archivos para cargar stage 2.
  - 2: Contiene la mayoría de la lógica de GRUB.
- Interfaz de texto, gráfica y a través de puerto serie
- Presente en la mayoría de distribuciones Linux.
- Solaris lo incorpora desde la versión 10 1/06
- Instalación: grub-install (stage 1).

Administración Avanzada de Redes de Sistemas UNIX/Linux

GRand Unified Bootloader (GRUB)

- Comandos sólo en menú:
  - default: Set the default entry
  - fallback: Set the fallback entry
  - hiddenmenu: Hide the menu interface
  - timeout: Set the timeout
  - title: Start a menu entry
- Comandos generales:
  - hide: Hide a partition
  - password: Set a password for the menu interface
  - serial: Set up a serial device
  - unhide: Unhide a partition
- Comandos en línea y menú:
  - boot: Start up your operating system
  - chainloader: Chain-load another boot loader
  - initrd: Load an initrd
  - install: Install GRUB
  - kernel: Load a kernel
  - makeactive: Make a partition active
  - root: Set GRUB's root device
  - rootnoverify: Set GRUB's root device without mounting
  - savedefault: Save current entry as the default entry

GRand Unified Bootloader (GRUB)

- Denotación de dispositivo de arranque:
  - ¡¡Índice basado en 0!!
  - Dispositivos: hd (hard disk), fd (floppy disk), nd (network disk)
  - hd(0,0): primer disco, primera partición
  - Numeración de dispositivos según BIOS.
    - /dev/hda = hd(0,x) /dev/hdb = hd(1,x)
    - /dev/sda = hd(0,x) /dev/sdb = hd(1,x)


```
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,4)
#          kernel /vmlinuz-version ro root=/dev/hda7
#          initrd /initrd-version.img
default=0
timeout=5
splashimage=(hd0,4)/grub/hpblue.xpm.gz
hiddenmenu
title Fedora Core (2.6.19-1.2895_1.fc6.cubbi_suspend2)
        root (hd0,4)
        kernel /vmlinuz-2.6.19-1.2895_1.fc6.cubbi_suspend2 ro root=/dev/hda7
        initrd /initrd-2.6.19-1.2895_1.fc6.cubbi_suspend2.img
title Windows XP Home Edition
        rootnoverify (hd0,1)
        chainloader +1
        makeactive
        boot
savedefault
```

Initial RAM Disk (initrd)

- Es un sistema de archivos temporal utilizado durante el arranque.
- El gestor de arranque lo carga en memoria y se lo pasa al kernel.
- El kernel monta el initrd como el sistema de archivos raíz.
- Este sistema de archivos raíz será sustituído por el definitivo.
- Objetivos:
  - Principal: preparar el sistema para el montaje definitivo del sistema de archivos raíz.
  - Carga de módulos: LVM, RAID.
  - Arranque por red.
  - Particiones cifradas.
  - Incluso necesario para dispositivos ATA, SCSI ó USB.
- Dos aproximaciones:
  - initrd genérico (Fedora, Ubuntu)
  - initrd específico (creado para un equipo concreto durante la instalación)

Instación Gráfica de Ubuntu

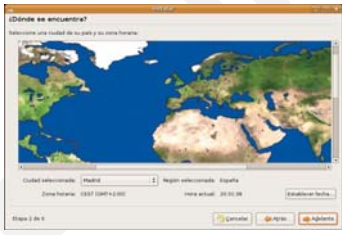
- Arranque del Cd Live

- Pulsar sobre el icono de instalación

- Seleccionamos el idioma

Instalación Gráfica de Ubuntu

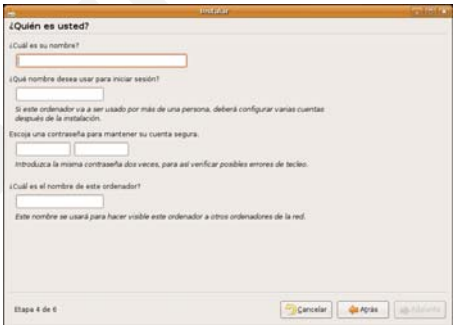
- Configuramos la zona horaria



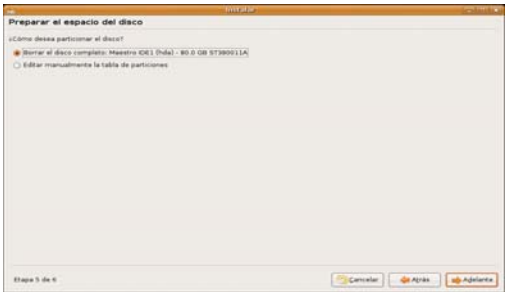
- Se elige la configuración de teclado

Instalación Gráfica de Ubuntu

- Se crean el primer usuario del sistema

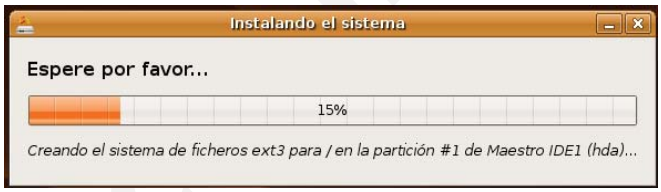


- Se crean las particiones

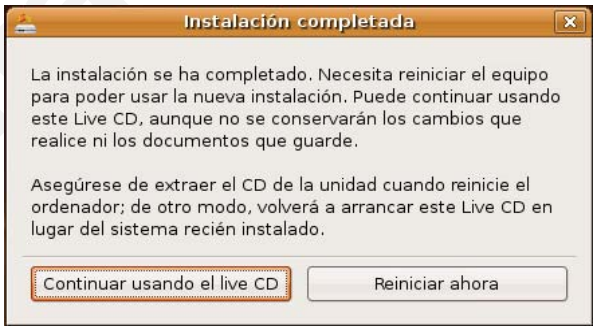


Instalación Gráfica de Ubuntu

- Comienza la instalación

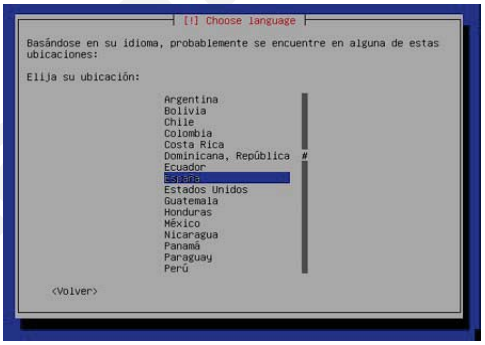


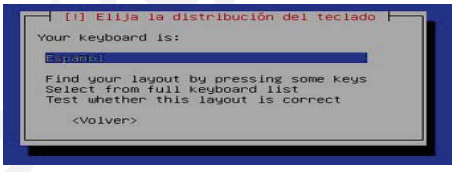
- Mensaje de finalización



Instalación Texto de Ubuntu

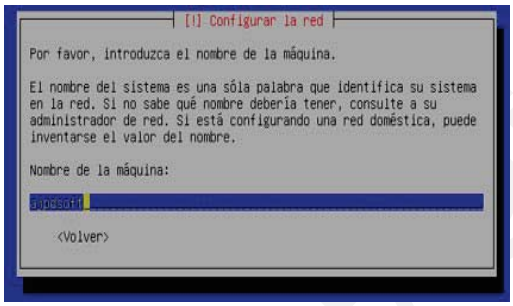
- Seleccionamos el idioma, localización y la distribución del teclado:

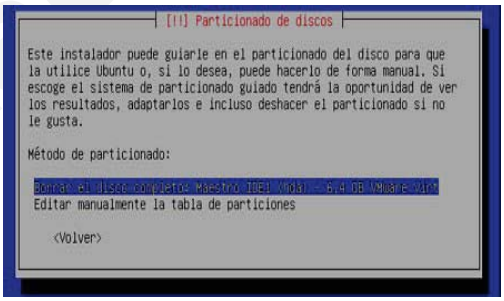




Instalación Texto de Ubuntu

- Tras cargar algunos módulos que van a ser usados en la instalación prosigue la configuración del equipo con el nombre de la máquina y el particionado de la misma

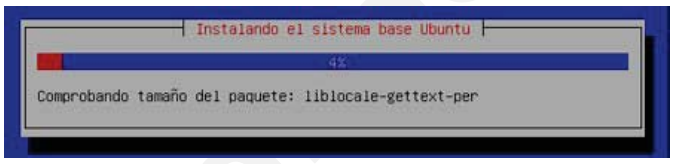




Administración Avanzada de Redes de Sistemas UNIX/Linux

Instalación Texto de Ubuntu

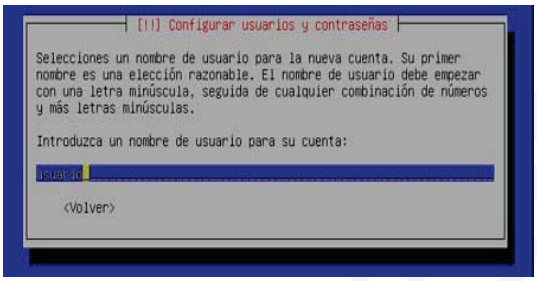
- Una vez que se han hecho las particiones, comienza el proceso de instalación del sistema base, en el cual se instalan los componentes básicos del sistema operativo y que permiten la ejecución del mismo.

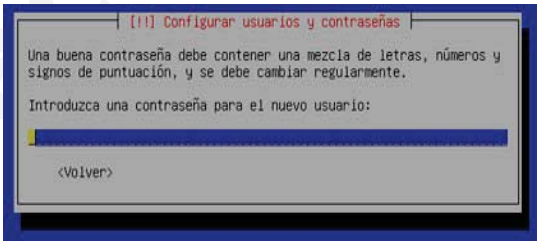


- Una vez instalado el sistema base se procede a la configuración del mismo, tanto de paquetes de idioma como de localización del sistema

Instalación Texto de Ubuntu

- Una vez hecha la configuración del sistema base, se crea el nuevo usuario del sistema dándole un **nombre de usuario y contraseña**.

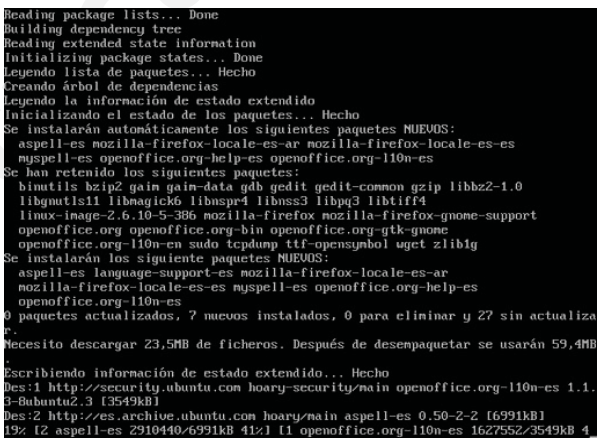




- Una vez creado el usuario, la instalación preparará la segunda fase y reiniciará el sistema

Instalación Texto de Ubuntu

- Una vez el equipo se ha reiniciado, se configura el servidor de vídeo y continúa con la instalación de los paquetes restantes del cd hasta finalizar la instalación.

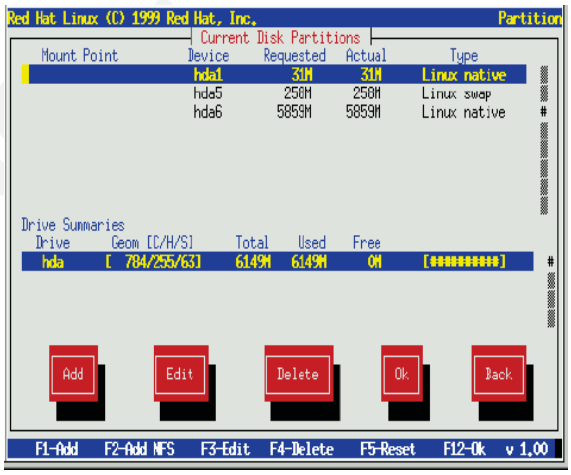


Herramientas de particionado de disco

- Existen varias herramientas para el particionado de disco dependiendo de la distribución que usemos y de si la instalación es en modo texto o en modo gráfico.
- Las más importantes son:
  - **Red Hat/Fedora:**
    - Disk Druid
  - **Ubuntu/Debian:**
    - Modo Texto: cfdisk
    - Modo Gráfico: qparted

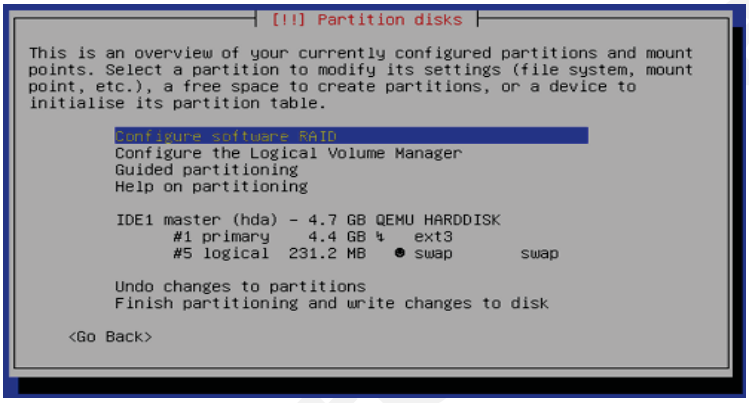
Herramientas de particionado de disco Red Hat/Fedora

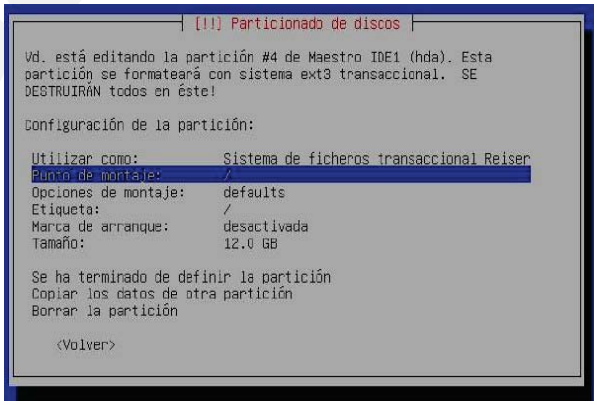
- Disk Druid



Herramientas de particionado de disco Ubuntu/Debian

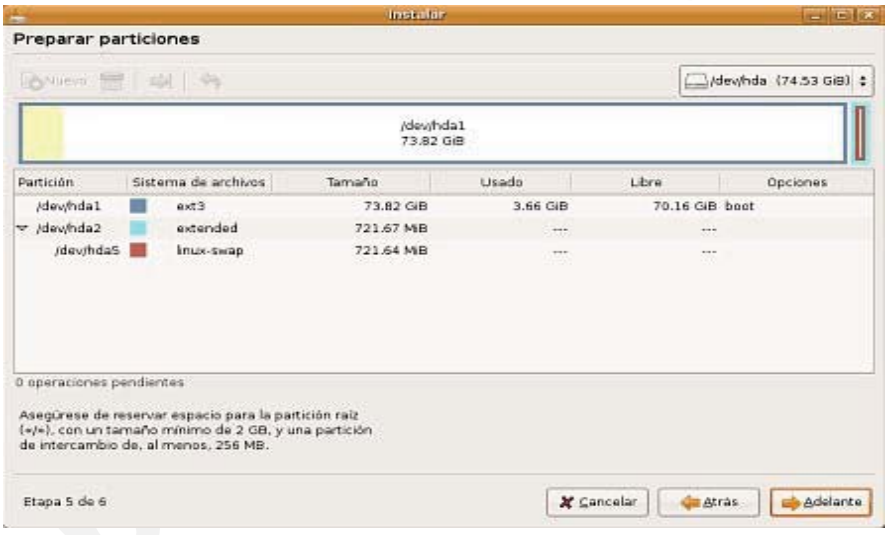
- CFDISK deriva de fdsik haciéndola más intuitiva al usuario usando los cursores de movimiento de tal forma que usando los cursores arriba y abajo se selecciona la partición y con izquierda y derecha la acción a realizar.





Herramientas de particionado de disco Ubuntu/Debian

- QTParted



Instalación del GRUB

- GRUB es el gestor de arranque usado por la mayoría de distribuciones Unix hoy en día y que substituye a LILO.
- La instalación del mismo se hace mediante el comando grub-install.
  - **/sbin/grub-install –root-directory=/boot /dev/hda**
- Con esta directiva se escribe en el MBR de /dev/hda, además estamos suponiendo que la partición de boot está en otra partición.
- Tomará por defecto la configuración almacenada en el fichero /boot/grub/grub.conf o menu.lst

Ejemplo de fichero configuración menu.lst

```
default=0
timeout=10
splashimage=(hd0,1)/grub/splash.xpm.gz
title Ubuntu (2.4.18-14)
        root (hd0,1)
        kernel /vmlinuz-2.4.18-14 ro root=/dev/hda3
        initrd /initrd-2.4.18-14.img
title Windows XP
        rootnoverify (hd0,0)
        chainloader +1
```

Parámetros del kernel

- Arranque en modo root mediante GRUB para la recuperación del sistema.
  - kernel /boot/vmlinuz-2.6.10-5-386 root=/dev/hda2 ro quiet splash rw init=/bin/bash
- **acpi** [on|off]: Activa o desactiva la deteccion de la carga o la conexion del ordenador al adaptador de corriente alterna,
- **apm** [on|off]: Activa o desactiva el módulo de suspensión.
- **mem**=MEMORIA: La usaremos cuando el kernel no sea capaz de detectar toda la memoria, forzándolo a usar la cantidad especificada como MEMORIA.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Parámetros del kernel

- **single**: Arrancará el sistema en single mode, sin lanzar los daemons, sin red... sólo lo básico. Muy útil para tareas de administración. Se inicia el proceso init y después nos pide la contraseña de root. Si pulsamos Crtl+d el proceso de arranque continuará en el modo establecido en el /etc/inittab
- **root=/dev/device**: Le dice al kernel qué dispositivo debe usar como sistema de archivos raíz. Por ejemplo, si arrancamos desde un Linux instalado en un pendrive usaríamos root=/dev/sda1
- **nousb:** El sistema arrancará el equipo sin activar los puertos de usb.

Parámetros del kernel

- **vga=ask**: Nos muestra durante el arranque los modos posibles de la gráfica, permitiéndonos elegir uno de forma interactiva.
- **ro** (read only): Con este parámetro ordenamos al kernel que monte el sistema de archivos raíz en modo de sólo lectura. Lo usaremos cuando queramos comprobar y reparar un sistema de archivos con fsck (no se debe hacer en modo lectura/escritura) o, por ejemplo, durante la investigación forense de una intrusión en un servidor.
- **rw** (read-write): Para montar el sistema de archivos en modo lectura/escritura. Es el modo por defecto.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Parámetros del kernel

- **panic**=N: Esta opción forzará un reinicio en N segundos en caso de un kernel panic. Útil si estamos probando un kernel que acabamos de compilar.
- **maxcpus**=N: Para indicar a un kernel SMP el número máximo de CPUs que debe usar.
- **debug**: Activa el kernel debugging. Útil sólo si estás programando un módulo de kernel y quieres encontrar problemas. Supongo que Linus Torvalds lo pasará directamente desde el cargador de arranque.
- **selinux** [0|1]: Activa (1) o desactiva (0) SELinux en el arranque.

Parámetros del kernel

- **raid**=/dev/mdN: Esta opción le dice al kernel cómo montar arrays RAID. Es de destacar que, si md está compilado dentro del kernel y no como módulo, las particiones de tipo 0xfd son detectadas y montadas "automágicamente" en RAID. Esta detección puede suprimirse usando el parámetro raid=noautodetect.
- **hdN**=noprobe: Para desactivar el dispositivo hdN. Recuerda que si desactivas en la BIOS, pongamos por caso, /dev/hdb, Linux lo seguirá detectando a no ser que uses este parámetro.
- **nopcmcia:** Desactiva la carga de controladores pcmcia.

Otras directivas de GRUB

- password: Añadir contraseña general. Las propiedades de contraseña permite limitar el acceso al shell de grub mediante contraseña y así impedir que se puedan realizar operaciones interactivas. El uso de contraseñas se indica en grub con:
  - `password [--md5] contraseña [fichero_configuración]`
- lock: bloquear arranques. En una máquina podemos tener varios sistemas operativos instalados, pero no queremos que todo el mundo pueda utilizarlos todos, queremos que para acceder a cierto sistema haya que introducir una contraseña.

Otras directivas de GRUB

- En ciertas ocasiones puede que interese modificar la asignación de particiones predeterminadas. Por ejemplo, Windows no se puede iniciar si no lo tenemos instalado en la primera partición del primer disco duro. Para estos casos usamos la orden map que modifica la asignación. Por ejemplo para que el sistema vea la segunda partición como si fuera la primera ejecutaríamos:
  - `grub> map (hd0) (hd1)`

Configuración arranque desde boot de Windows

- En el momento de instalación del GRUB se debe instalar este en el primer sector de la partición /boot. (nunca en el MBR) en la ubicación de la partición /boot en el disco duro.
- Una vez arrancado el sistema Linux se debe obtener los 512 bytes del sector de arranque de Linux para esto usamos el siguiente comando:
  - **dd if=/dev/particion_boot of=/destino/linux.bin bs=512 count=1**
- Una vez obtenido linux.bin, se debe copiar a C:\ con esto ya tenemos el sector de arranque en la partición de Windows.

Configuración arranque desde boot de Windows

- Una vez hecho esto debemos editar el archivo C:\boot.ini que es el fichero de configuración del arranque de Windows y agregar a este fichero la línea del arranque de Linux.
  - **c:\linux.bin="Linux"**
- Si no se puede modificar este archivo se han de cambiar los privilegios con el comando
  - C:\attrib -R -S -H boot.ini.
- Una vez editado volvemos a dejar los mismos privilegios:
  - C:\attrib +R +S +H boot.ini

Configuración y Administración

Configuración y Administración

- Niveles del sistema
- Cadena de arranque
- Administración de servicios
- Arranque y detención del sistema
- Fichero */etc/passwd*
- Fichero */etc/shadow*
- Política de grupos
- Gestión de usuarios
- Herramientas para la gestión de usuarios

Los niveles de ejecución del sistema (runlevels)

Aparecieron debido a la necesidad de diferenciar el funcionamiento del sistema dependiendo de las diferentes formas de mantenimiento que se llevan a cabo en el sistema.

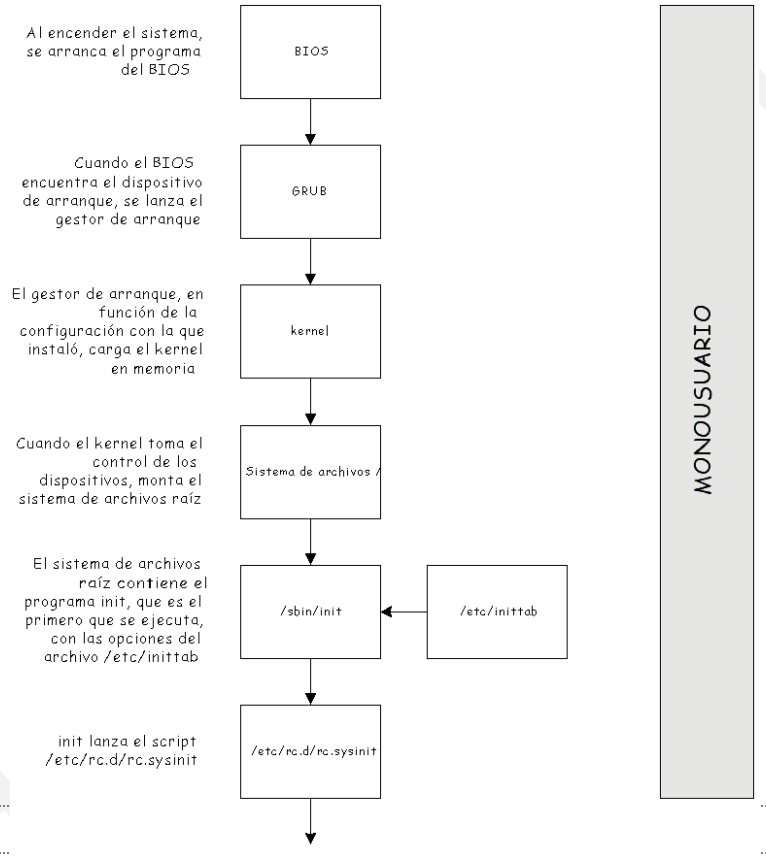
En el fichero `/etc/inittab` se reflejan los niveles del sistema.

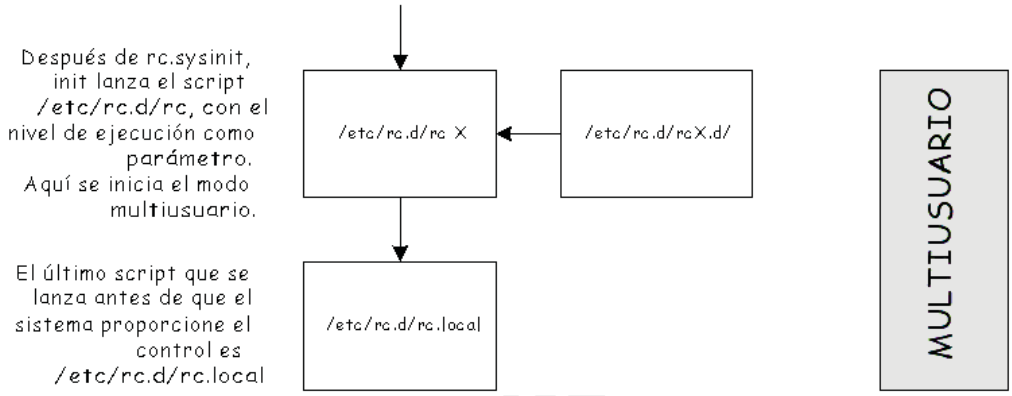
```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
```

- **Identificador**
- **Runlevels** en los que se ejecuta la acción
- **Acción**. Acción que se realiza
- **Proceso**. Proceso que se ejecuta

```
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

```
# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```



El script rc realiza lo siguiente:

- Detiene los servicios que no deben ejecutarse en ese nivel de ejecución. K00 primero en detenerse. K99 el último.
- Después, inicia los servicios que deben estar activos en ese nivel de ejecución. S00 primero en detenerse. S99 el último.

Administración de servicios

- Ubuntu:
  - Mantenimiento manual
  - update-rc.d
  - Herramienta gráfica (Sistema/Administración/Servicios)
  - Webmin
- Solaris:
  - Mantenimiento manual
  - Webmin
  - svcs: lista servicios
  - svcadm: permite activar y desactivar servicios.

Arranque y detención del sistema

Los sistemas UNIX requieren de un apagado ordenado. Sólo cuando el sistema esté listo para su apagado podrá procederse. De no ser así al proceder nuevamente a su encendido, posiblemente encontraríamos los sistemas de ficheros dañados.

La orden para proceder a un apagado ordenado es `shutdown`, y puede tomar los siguientes parámetros:

- `-r`. Reinicia el sistema tras el `shutdown`
- `-h`. Detiene el sistema tras el `shutdown`. Si no se especifica ninguno de estos dos parámetros, tras el `shutdown`, el sistema pasa a modo `monousuario`.
- `-f`. Realiza un `fastshutdown`, que consiste en no chequear los sistemas de ficheros al volver a arrancar el sistema.
- `-F`. Fuerza el chequeo de los sistemas de ficheros al arrancar el sistema.
- `-c`. Cancela un `shutdown` que, de otro modo, se ejecutaría

Después de incluir las opciones, indicamos a `shutdown` el tiempo que debe esperar antes de realizar el `shutdown`. Este parámetro se puede especificar como un tiempo absoluto (hh:mm), o relativo (+m). `now` es un alias para +0.

Fichero */etc/passwd*

El fichero ***/etc/passwd*** (*more*, *less* y *ls* para ver contenido y permisos 644, propiedad de *root* y el *GID* debe ser 0 (*root* o *system*)).

Nombre de cuenta de usuario para *login* en el sistema

Contraseña cifrada. Puede ser x *(/etc/shadow*)

Identificador numérico de usuario

username:pwd:UID:GID:Full_name:Home_Directory:tipo_shell

Identificador de grupo de usuario

Nombre real del usuario *GCOS*(datos personales)

Directorio de trabajo del usuario y acceso al sistema

Tipo de *shell (/bin/sh, /bin/csh, /bin/bash, /bin/ksh...*

Fichero */etc/shadow*

- */etc/passwd* almacena claves de acceso, lo cual es un elemento sensible del sistema (aunque estén cifradas).
- El mecanismo de *shadow passwords* permite extraer la contraseña cifrada del archivo público */etc/passwd* y llevarlo a */etc/shadow*, al que sólo pueden acceder el sistema y el administrador.
- Si el campo *password* de */etc/passwd* contiene una *x*, las contraseñas estarán en */etc/shadow* que contiene otra info de seguridad de cuentas y sólo legible por *root*.
- *#/usr/sbin/pwconv* crea */etc/shadow*. *#/usr/sbin/pwunconv desactiva el mecanismo de shadow passwords.*

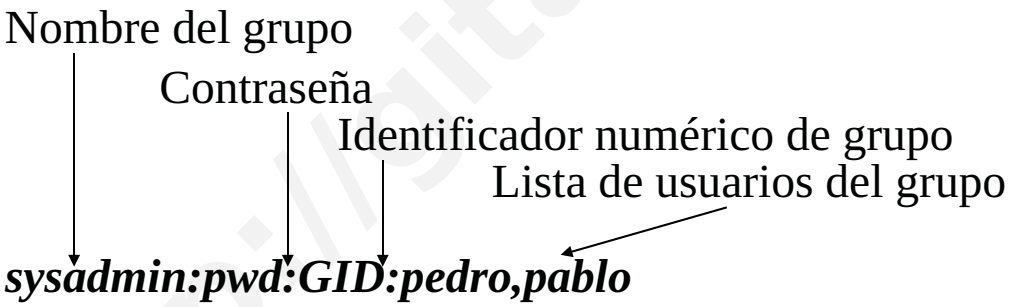
Formato de entradas del fichero */etc/shadow*:

username:34r.U9:ult_cambio:min:max:aviso:inactivo:expiración

ult_cambio: Días transcurridos desde el cambio de la contraseña.
min: número de días que deben pasar antes del cambio de *password*.
max: número de días de duración de la contraseña.
aviso: número de días de aviso de fin de validez de contraseña.
inactivo: días permitidos en inactivo antes de que cuenta se bloquee.
expiración: fecha absoluta de caducidad de la cuenta, alcanzada la cual no podrán realizarse más accesos al sistema.

Política de grupos

- Administrador puede clasificar usuarios en grupos, pudiendo establecer privilegios y restricciones para conjuntos de usuarios.
- El fichero /etc/group (*more*, *less* y *ls* para ver contenido y permisos):



Gestión de usuarios

- Alta manual de usuarios:

1- *#groupadd -g 200 curso (#vi /etc/groupadd*)
2- *#useradd -u 101 -g 200 -c "Cucufato" -d /export/home/alum1 \ -m -s /bin/csh alum1* (*#vi /etc/passwd*)
3- *#mkdir /export/home/alum1*
4- *#cd /etc/skel* Directorio "*skeleton*"
5- *#cp .* /export/home/alum1*
6- *#chown -R alum1 /export/home/alumn1*
7- *#chmod -R 744 /export/home/alum1*
8- *#passwd alumn1*
9- *#su alum1*
10- *login*

- Administración manual de usuarios:

#passwd (cambios de contraseñas. -s permite cambio de *shell*)

#su (switch user. Con el - se leen las vbles. entorno del usuario accedido)
#id (muestra *UID* y *GID* del usuario)
#chsh (cambio del tipo de *shell* en *Linux*, pide antes *pwd*)
#who -u (quien está dentro y desde cuando lee */var/run/utmp*)
#ps, #finger (lee *passwd, /var/run/utmp, .forward, .plan*

- Ficheros de */etc/default* relacionados con la gestión de usuarios (*Solaris*):

- /etc/default/su

SULOG=/var/adm/sulog: Fichero donde se registran los intentos de *su*. Si variable está comentada (#) *su* es rechazado

CONSOLE=/dev/console: Si la variable no está comentada todos los intentos de *su* a *root* son mostrados en la consola del sistema.

- /etc/default/passwd: Define las variables *MAXWEEKS*, *MINWEEKS* y *PASSLENGTH* generalizado a todos los usuarios del sistema. Cuando *MAXWEEKSy MINWEEKS* son 0, sólo los usuarios con valor en *MAX* y *MIN* de */etc/shadow* deben cambiar sus contraseñas según lo definido.

- /etc/default/login: Define variables de seguridad de acceso:
PASSREQ=YES : todos los usuarios deben tener contraseña.
CONSOLE=/dev/console: root sólo hará *login* desde consola.
CONSOLE= : No se permite *login* a *root*. Debe hacerse con *su*.
#CONSOLE : Se permite conexión como *root*

Herramientas para la gestión de usuarios

- *Solaris*:
  - *admintool* Herramienta gráfica de gestión de usuarios
  - Alta manual: *#useradd* y *#groupadd*
- *Linux (Red Hat)*:
  #adduser (interactivo), *#useradd* (parámetros), *#usermod*, *#groupmod*
  #newusers < fichero_usuarios
  #userdel username; #find home_directory -exec rm {} \;
  #groupadd username Usuario puede estar en varios grupos
  #groupdel username #groupmod -ng nuevo_g viejo_g

Administración Avanzada de Redes de Sistemas UNIX/Linux

#adduser

Enter login name for new account (^C to quit): ***alum1***
Full Name: ***Cuenta para curso***
GID [100]: ***200***
Checking for an available UID after 500.....
UID[508]: ***510***
Home Directory [/home/...]: ***/home/alum1***
Shell [/bin/bash]:
Password:
Is this correct? [y/n]:
Adding the files.....

- Proceso de login:

init →*getty* →ejecuta programa *login*

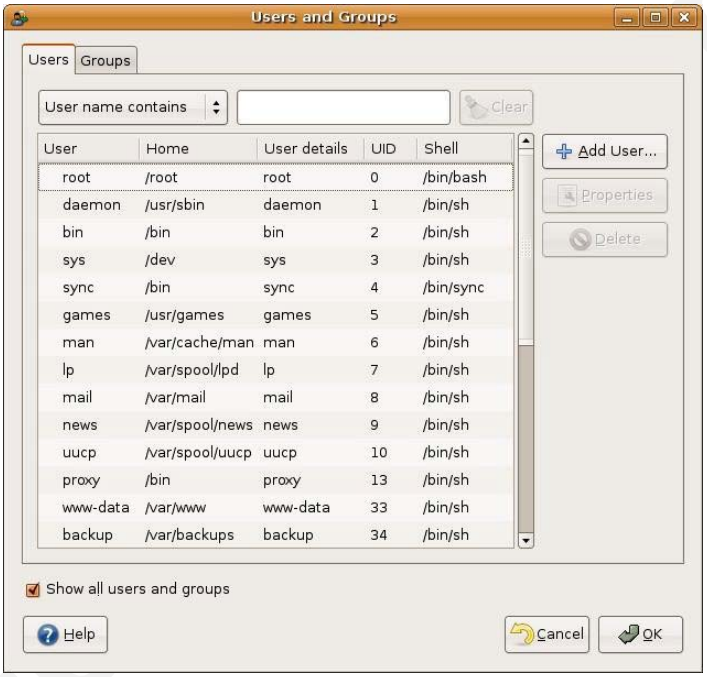
Si existe */etc/nologin* se impide acceso al sistema (*shutdown* lo crea). Si no existe *login* solicita *username* y *password* y comprueba con el contenido de */etc/passwd*. Si no concuerdan, *login* solicita nuevamente *username* y *password*.

Si todo concuerda:

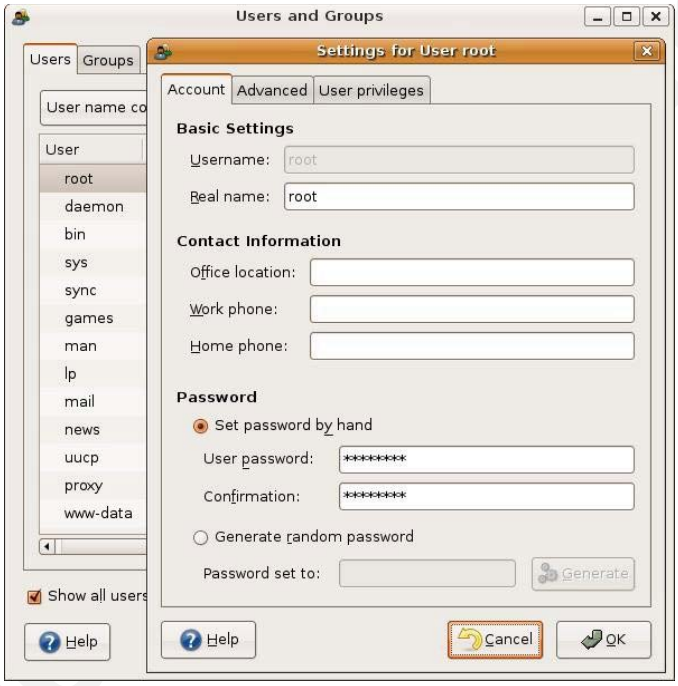
Se consulta *home directory*, *shell*, *UID*, *GID*, *GECOS*
Se visualiza */etc/motd* y lee correo (*touch .hushlogin* lo evita)
Se ejecutan ficheros *.login*, *.profile*, etc según tipo de *shell*

Intentos fallidos en *syslog*. Accesos en */var/log/wtmp*

- *System-Administration-Users and Groups*:

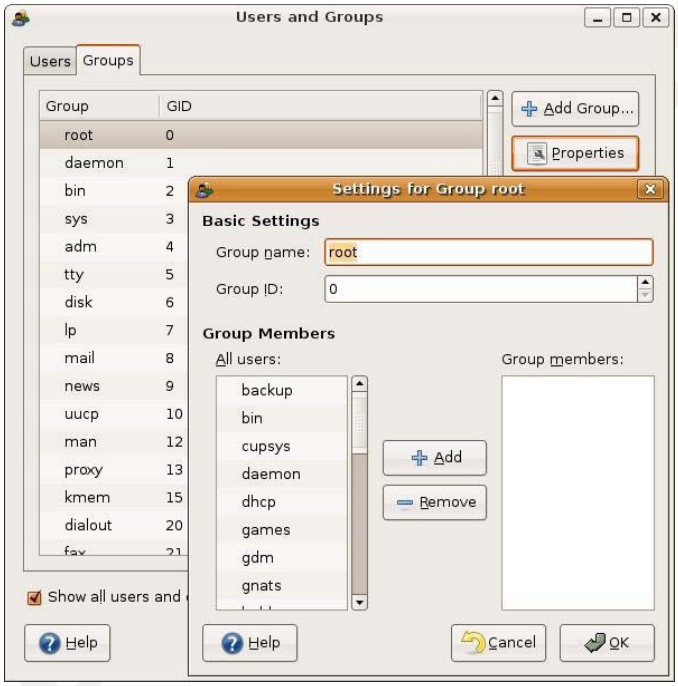


- *User Properties*:



Nº 93

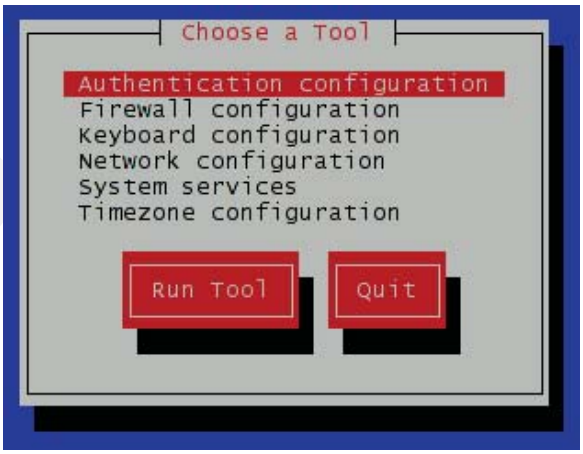
- *Groups:*



Nº 94

Herramientas de configuración y administración

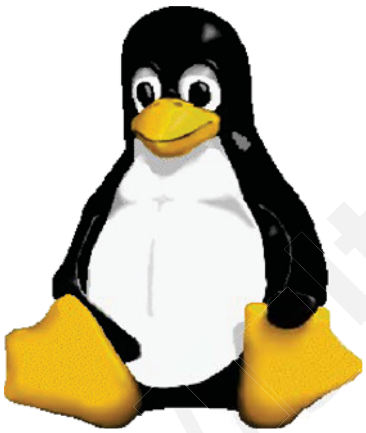
- Red Hat: *Setup*
  Es una agrupación de diversas herramientas de configuración y administración disponibles en modo texto. Cada una de las opciones o herramientas también puede ser invocada desde una shell del sistema.



- Ubuntu: *System-Administration*

- *Webmin*: Permite la administración remota del sistema a través de *http://host:10000*

Kernel de GNU/Linux y Dispositivos



Kernel de GNU/Linux y Dispositivos

- Descarga y parcheado del código fuente del kernel de Linux
- Configuración, compilación e instalación del kernel
- Dispositivos
- Carga de *driver* mediante módulos
- Nodos de dispositivo
- udev
- sysfs
- HAL

Kernel de GNU/Linux

- ¿Qué es el kernel de un sistema operativo?
  - De forma breve; porción de software que gestiona el hardware del computador y permite ejecutar aplicaciones que dependen de él para acceder a dicho hardware.
- ¿Por qué me debe interesar?
  - Porque como administrador hay ocasiones en que hay que ajustarlo para que todo funcione bien.

Kernel de GNU/Linux

- ¿Que tiene de especial el Kernel de GNU/Linux?
  - Multitarea, uso de memoria virtual, uso de librerías dinámicas, carga bajo demanda, P&P, APM, conectividad, diversos *filesystems* transparente al usuario...
  - Módulos: porciones dinámicas, parte del kernel, que se cargan o descargan sólo cuando se usan. Esto implica que no gastan memoria mientras no se usan, el kernel es más reducido, se mejora su mantenimiento, etcétera

Administración Avanzada de Redes de Sistemas UNIX/Linux

Kernel de GNU/Linux

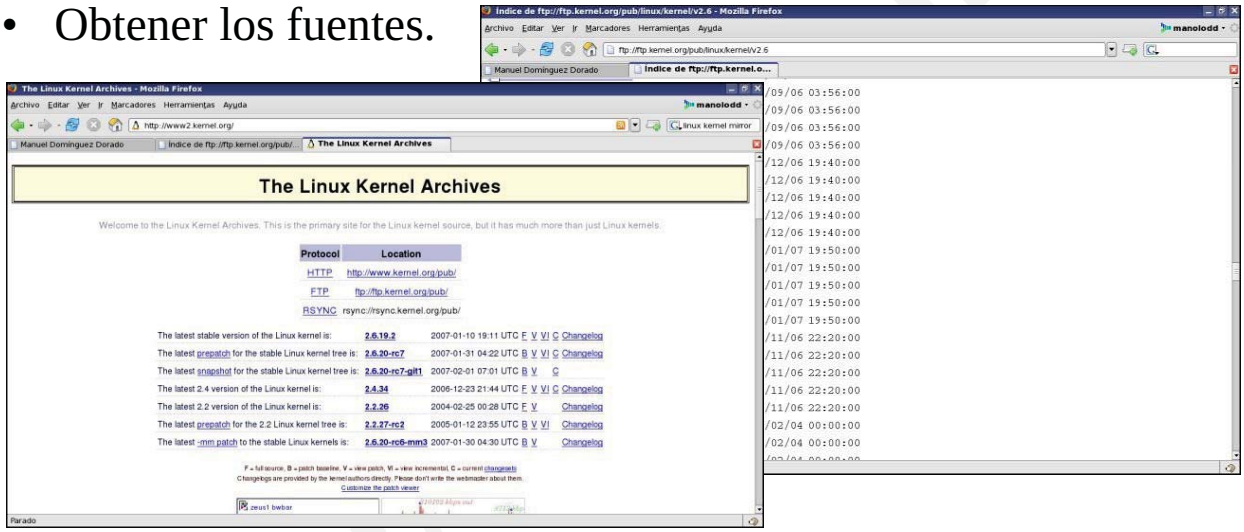
- ¿Que razones hay para compilar el Kernel de GNU/Linux?
  - Ajustar el kernel a los procesadores de la máquina.
  - Eliminar funcionalidades innecesarias.
  - Corrección de *bugs* existentes.
  - Soporte para hardware.
  - Se tiene una distribución que no se puede cambiar (entornos empresariales) y no actualiza el kernel.
- Todo administrador, tarde o temprano, compila el kernel.

Kernel de GNU/Linux

- ¿Qué pasos hay que seguir?
  - Descargar e instalar los fuentes.
  - Descargar o instalar los parches (si procede).
  - Preparar las librerías y utilidades necesarias.
  - Configurar las opciones de compilación.
  - Compilar.
  - Instalar los módulos.
  - Instalar el kernel.
  - Preparar el gestor de arranque.

Kernel de GNU/Linux

- Obtener los fuentes.



  - http://www2.kernel.org o bien, ftp://ftp.kernel.org

Kernel de GNU/Linux

- ¿Que significan los número de versión?
  - En una versión llamada A.B.C.D:
    - A: superversión. Sólo se modifica ante enormes cambios.
    - B: versión. Par=estable. Impar=versión de desarrollo.
    - C: revisión. Indica las mejoras añadidas a la versión.
    - D: subrevisión. Indica cambios ínfimos y muy puntuales.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Kernel de GNU/Linux

- En la Web del kernel veo el enlace linux-2.6.18.1
  - Entonces es la superversión 2, la versión 6, estable, con revisión 18 y una subrevisión.
  - Actualmente la última versión es 2.6.19.2, así que 2.6.18.1 es “relativamente” nueva.
- ¿Por que algunos enlaces se llaman patch-A.B.C.D?
  - Son parches. Permiten actualizar las fuentes de un kernel antiguo sin bajar los más de 40 MB que ocupa el nuevo.

Kernel de GNU/Linux

- Parche con la forma patch-A.B.C
  - Es incremental.
  - Sirve para pasar de linux-A.B.C-1 a linux-A.B.C
- Parche con la forma patch-A.B.C.D
  - NO es incremental.
  - Sirve para pasar de linux-A.B.C a linux-A.B.C.D

Administración Avanzada de Redes de Sistemas UNIX/Linux

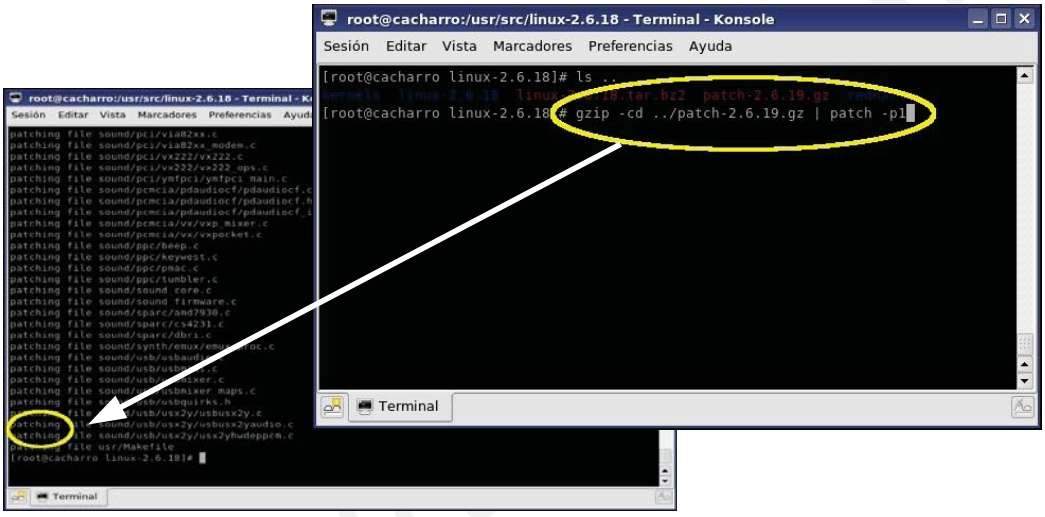
Kernel de GNU/Linux

- Supongamos que queremos instalar un nuevo kernel de cero.
- Hay que bajar fuentes. Por ejemplo linux-2.6.19.1.tar.bz2
- Se copian a **/usr/src**
- Se descomprimen y desempaquetan.
  - **tar xjfv linux-2.6.19.1.tar.bz2**
- Esto crea el directorio /usr/src/linux-2.6.19.1 con el código fuente completo del kernel.
- Ya están instalados los fuentes del nuevo núcleo.

Kernel de GNU/Linux

- Supongamos que tenemos instalado linux-2.6.18.tar.bz2 y queremos actualizarlo a linux-2.6.19.tar.bz2 con parches.
  - Descargamos patch-2.6.19.gz
  - Se copia a **/usr/src**
  - Se parchea el código fuente del núcleo que ya teníamos:
    - **gzip -cd ../patch-2.6.19.gz | patch -p1**
  - En este caso hay que hacerlo desde **dentro** del directorio /usr/src/linux-2.6.18, el que queremos parchear.

Kernel de GNU/Linux



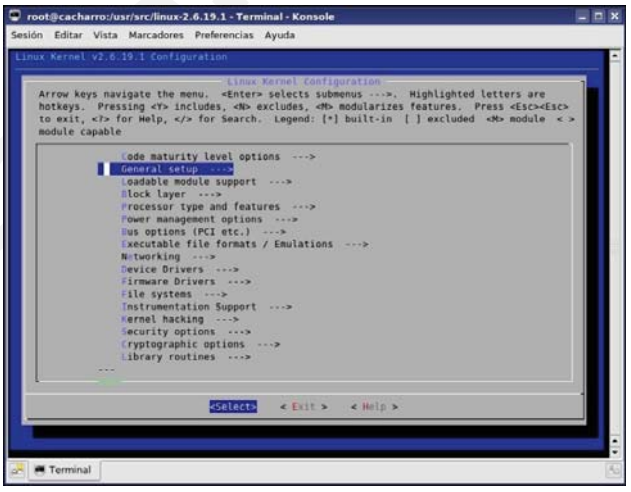
- Ya tenemos los fuentes de linux-2.6.19 instalados mediante un parche.

Kernel de GNU/Linux

- Todo listo para empezar ¿no?
  - Realmente no. Hay que comprobar que las librerías y utilidades necesarias para el proceso están instaladas.
  - Nuevas versiones del kernel pueden requerir nuevas versiones de estas librerías y utilidades.
  - Para ver lo que se necesita, leer siempre el fichero:
    - **/usr/src/linux-A.B.C.D/Documentation/Changes**
  - Hay que instalar todo lo que se indique, **SI SE PUEDE**.

Kernel de GNU/Linux

- Para configurar las opciones de compilación del kernel:
  - **cd /usr/src/linux-A.B.C.D**
  - **make mrproper**
  - **make menuconfig**
  - También, para lo último:
    - **make config**
    - **make xconfig**
    - **otras...**



Kernel de GNU/Linux

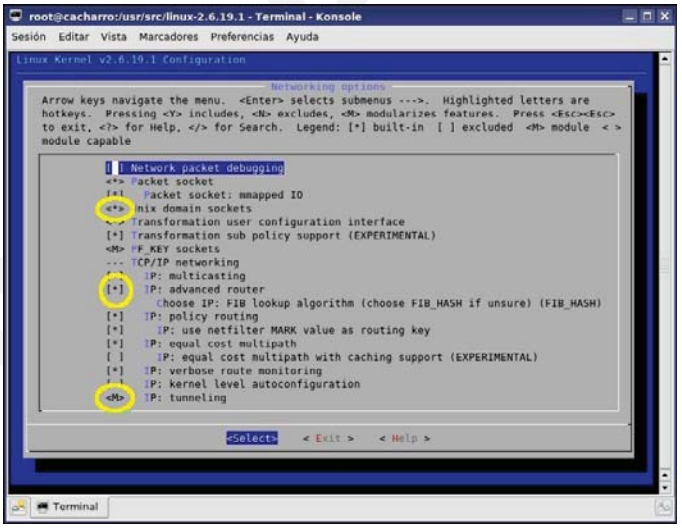
- Decidir qué opciones activar en el kernel y cómo, es una tarea compleja y larga. Se necesita:
  - Conocer bien el hardware que se tiene.
  - Leer con precaución la ayuda de cada opción. Sin prisas.
    - Por ejemplo, para muchas tarjetas Fast Ethernet hay que activar el módulo 8139too.
  - Documentarse bien sobre qué módulo sirve para qué hardware. Primero, saber que hardware se tiene instalado.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Kernel de GNU/Linux

- Para cada valor puede haber cuatro opciones:

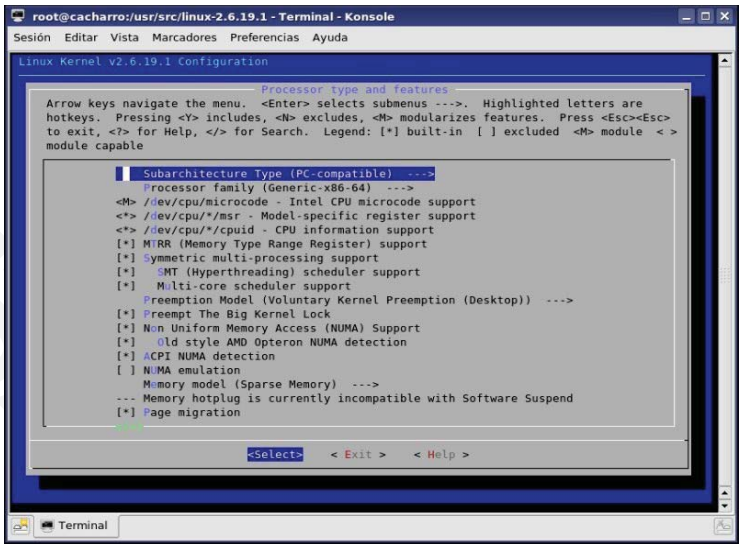
<*>, <M>, < >, [ ] y [*]



- <*> Incluir estático.
- <**M**> Incluir módulo.
- < > No incluir.
- **[ ]** No incluir.
- **[*]** Incluir estático.

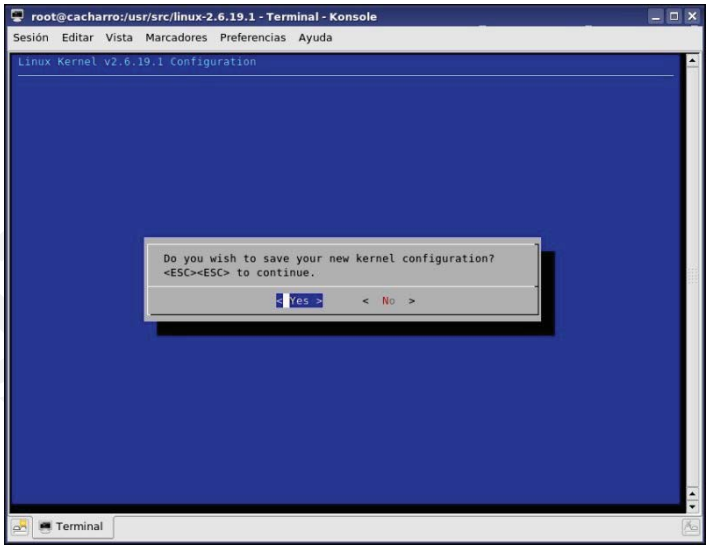
Kernel de GNU/Linux

- Seleccionar durante el proceso todo lo que se desee:
  - Procesador.
  - Multiprocesamiento.
  - Tecnologías de red.
  - Sistemas de fichero.
  - Soporte hardware.
  - Algoritmos.
  - Etcétera.



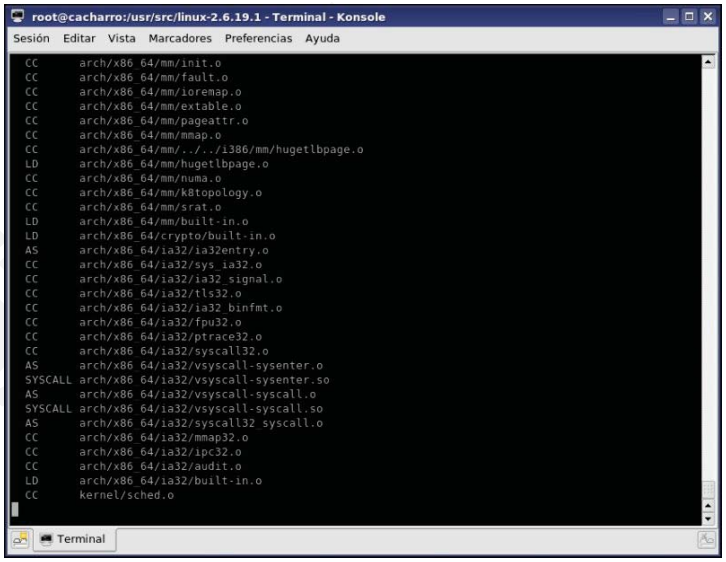
Kernel de GNU/Linux

- Al finalizar, salir y guardar la configuración cuando se pregunte.
  - Ésta, se almacena en un fichero oculto llamado **.config**
- Ya están configuradas las opciones de compilación.



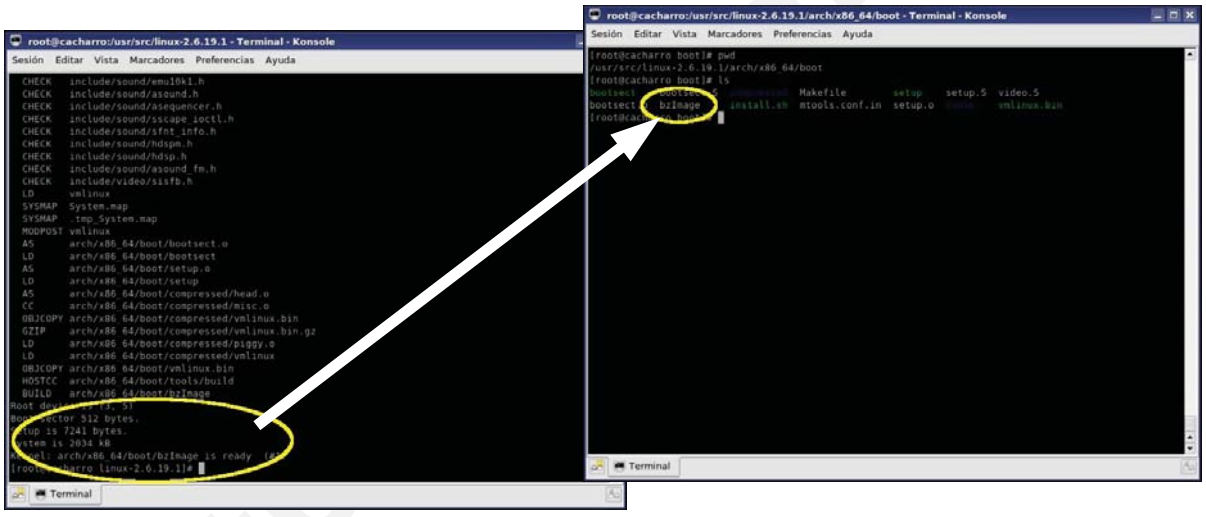
Kernel de GNU/Linux

- Una vez finalizada la configuración, se pasa a compilar el kernel.
  - **make bzImage**
- O, si se tienen más de un procesador, se obtiene mejor rendimiento con:
  - **make -j bzImage**



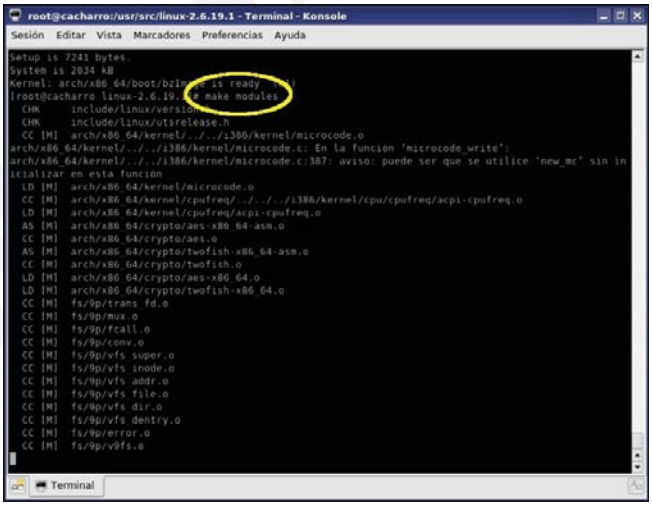
Kernel de GNU/Linux

- Al final (puede que 1 hora después), en **arch/XXX/boot/** se encuentra el kernel ya compilado, con nombre **bzImage**.



Kernel de GNU/Linux

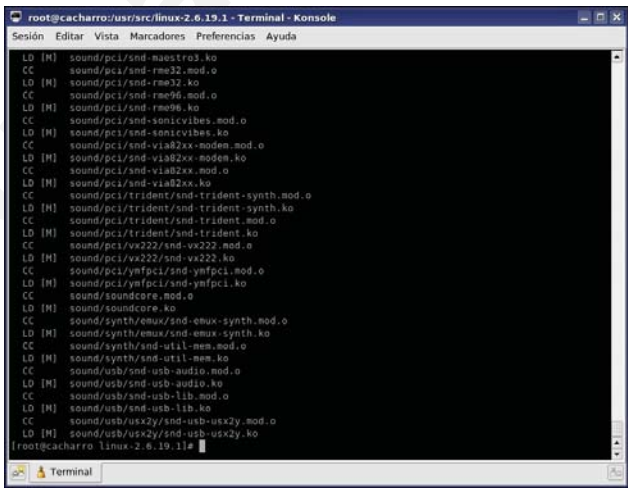
- Ya está compilada la parte estática del kernel. Ahora hay que compilar lo que se decidió que fuesen módulos.
  - **make modules**
- O, si se tienen más de un procesador, se obtiene mejor rendimiento con:
  - **make -j modules**



Administración Avanzada de Redes de Sistemas UNIX/Linux

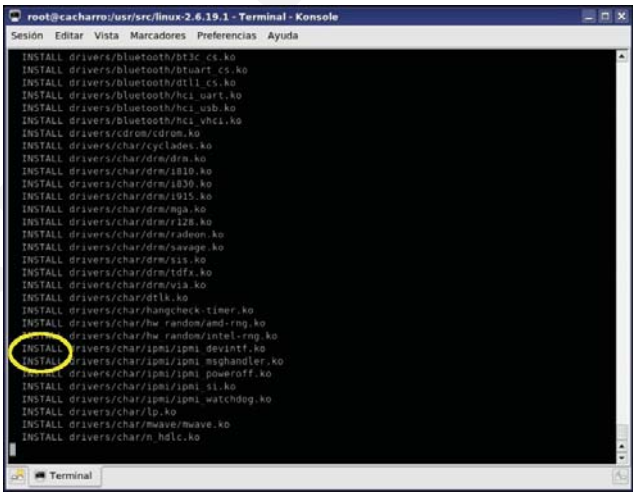
Kernel de GNU/Linux

- Al final (puede que otra hora después), se terminan de compilar todos los módulos. Pero aún queda instalarlos.
  - **make modules_install**
- O, si se tienen más de un procesador, se obtiene mejor rendimiento con:
  - **make -j modules_install**



Kernel de GNU/Linux

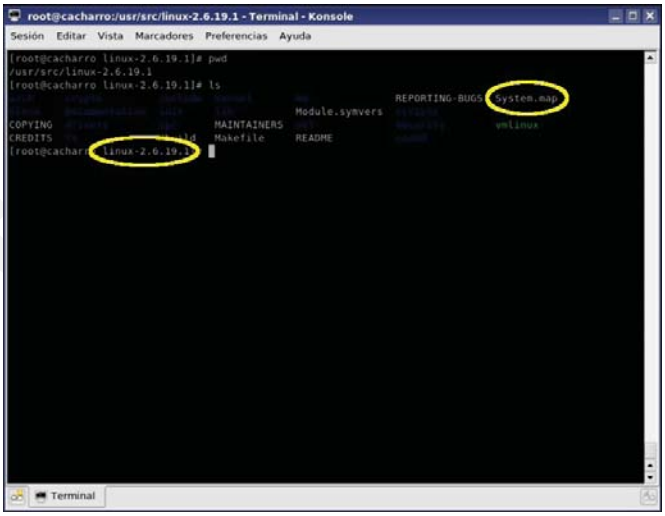
- La instalación de los módulos hace dos cosas:
  - Instala los módulos compilados bajo **/lib/modules/A.B.C.D**
  - Calcula qué módulos de penden de otros y lo guarda en:
    - **/lib/modules/A.B.C.D/modules.dep**



Administración Avanzada de Redes de Sistemas UNIX/Linux

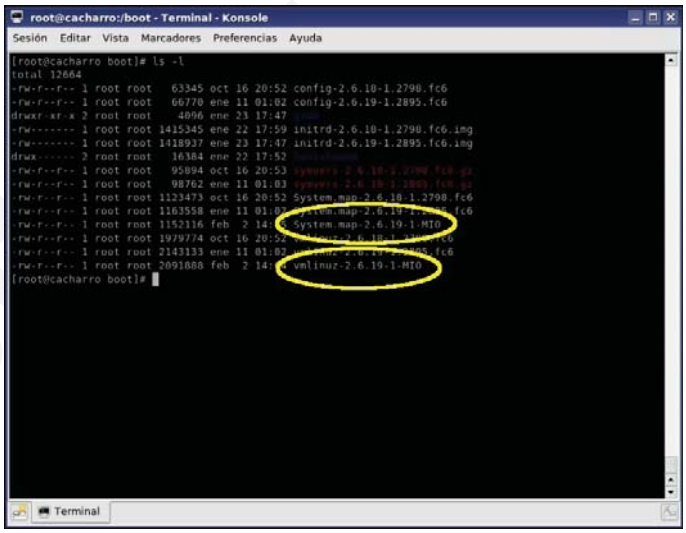
Kernel de GNU/Linux

- La compilación genera un archivo llamado **System.map**
  - Se encuentra en el directorio /usr/src/linux-A.B.C.D
  - Documenta todos los símbolos del kernel.
  - Es importante para el arranque.



GNU/Linux

- Para instalar el nuevo kernel, hay que copiar **bzImage** y **System.map** a **/boot**.



- Se debe cambiar el nombre para evitar errores:
  - **System.map-A.B.C.D**
  - **vmlinux-A.B.C.D**
- Para **System.map** y **bzImage**, respectivamente.

Kernel de GNU/Linux

- Por último, hay que configurar el gestor de arranque para que permita arrancar con el nuevo kernel.
- Basta con copiar y pegar la mismas líneas existentes para el kernel actual y cambiar el nombre del kernel a arrancar por el nuevo que hemos puesto en **/boot**.
- El fichero de configuración del gestor de arranque, si se usa GRUB, está en **/boot/grub/menu.lst**
- Si se usa LILO, está en **/etc/lilo.conf**
- El siguiente ejemplo es sólo para GRUB.

Kernel de GNU/Linux

- Ejemplo de contenido nuevo a añadir en el fichero **/boot/grub/menu.lst**

```
#######################################
title GNU/Linux (Nuevo kernel)
        root (hd0,2)
        kernel /vmlinuz-A.B.C.D ro root=LABEL=/1 rhgb quiet
#######################################
```

- **IMPORTANTE:** Se deben añadir las nuevas líneas, pero se deben dejar las antiguas para poder arrancar si algo va mal.

Kernel de GNU/Linux

- ¿Puede algo ir mal a partir de este momento?
  - Si, puede que no te arranque el kernel nuevo. Lo normal es que si esto pasa, muestre un mensaje con un error muy aclaratorio:

    “KERNEL PANIC”

  - Normalmente se debe a un error típico. Se ha compilado como módulo el sistema de ficheros o el dispositivo donde reside el núcleo. **SOLUCIÓN:** recompilar.

Dispositivos en Linux

- Un dispositivo sólo podrá ser usado si el kernel lo soporta o si existe un controlador capaz de controlarlo y si se configura apropiadamente para hacerlo.
- Las fuentes en C de cada versión del kernel cuentan con controladores para diversos dispositivos. Cuando se compila una versión, algunos de esos controladores pueden unirse con el kernel mismo (estáticamente), otros pueden dejarse como módulos para cargarse/descargarse cuando la parte estática del kernel este operando, otros pueden ser excluidos del proceso de compilación

Dispositivos en Linux

- Un módulo se refiere a un controlador de un dispositivos o servicio que puede cargarse o descargarse cuando el usuario o algún dispositivo lo solicita (p.e dinámicamente). Los módulos que se distribuyen con en el kernel están ubicados en el directorio /lib/modules/version, donde version es la versión de su kernel, organizados en directorios que indican el tipo de dispositivo o el propósito, por ejemplo fs - sistema de archivos, net - protocolos y hardware para redes.
- Para lograr configurar un dispositivo controlado por un módulo, puede emplear las herramientas del paquete modutils o **modconf**

Dispositivos en Linux

- **lsmod**
  - Lista los módulos cargados, de cada uno presenta nombre, tamaño, cuenta de usos y lista de módulos que lo usan (es equivalente a cat /proc/modules).
- **rmmod módulos**
  - Descarga uno o más módulos cargados, mientras estos no estén siendo usados. Con la opción -r intenta descargar recursivamente módulos de los cuales el módulo especificado dependa. El comando rmmod -a descarga todos los módulos que no estén siendo usados.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Dispositivos en Linux

- **insmod módulo [opciones]**
  - Trata de cargar el módulo especificado. Pueden pasarse opciones específicas para el módulo, a continuación del nombre con la sintaxis símbolo=valor (los símbolos posibles dependen del módulo)
- **depmod**
  - Un módulo puede requerir otros, es por esto que hay dependencias que deben respetarse al cargar y descargar módulos. depmod permite calcular tales dependencias entre varios módulos o entre todos los disponibles.

Dispositivos en Linux

- **modprobe módulo opciones**
  - Emplea la información de dependencias generada por depmod e información de /etc/modules.conf para cargar el módulo especificado, cargando antes todos los módulos de los cuales dependa.
- **modconf**
  - Permite listar, cargar y descargar módulos con menús.
- **update-modules**
  - Actualiza el archivo /etc/modules.conf a partir de la información de los archivos del directorio /etc/modutils

Nodos de dispositivo

- Los nodos de dispositivo (ó archivos de dispositivo) se refieren a dispositivos virtuales o físicos del sistema, tales como discos duros, tarjetas de video, pantalla o teclado. Un ejemplo de dispositivo virtual es la consola, representado por /dev/console.
- Existen dos tipos de dispositivos:
  - dispositivos de carácter
  - dispositivos de bloque

Nodos de dispositivo

- En un sistema Linux tradicional, en el directorio /dev hay nodos de dispositivo creados para cada dispositivo conocido, esté o no en el sistema. Se dice que es un conjunto de ficheros estático, ya que los nodos no cambian.
- Los archivos de dispositivo se mapean en los correspondientes dispositivos a través de dos números, conocidos como "números de dispositivo mayor y menor". El número de dispositivo mayor identifica el driver con el cual está asociado el archivo. El número de dispositivo menor identifica a qué dispositivo del tipo dado se refiere (número de unidad o instancia).

Administración Avanzada de Redes de Sistemas UNIX/Linux

Nodos de dispositivo

- Los archivos de dispositivo se crean con el comando mknod, cuya sintaxis es:
  - **mknod nombre tipo [mayor menor]**
- **nombre** es el nombre del archivo de dispositivo a ser creado, **tipo** es c para dispositivos tipo caracter o b para dispositivos modo bloque, y **mayor y menor** son los números de dispositivo mayor y menor.
- Muchos sistemas proveen un script llamado MAKEDEV (en /dev) para proveer automáticamente los argumentos a mknod para los dispositivos más comunes.

udev

- El modelo de gestión tradicional de dispositivos da algunos problemas:
  - el directorio /dev es enorme y difícil de manejar.
  - los números mayor y menor que se asocian a cada dispositivo se estaban acabando
  - los usuarios necesitan que cada dispositivo sea accesible de la misma manera.
  - los programas necesitan poder detectar cuándo se ha conectado o desconectado un dispositivo, y cuál es la entrada que se le ha asociado en /dev

Características de udev

- udev mantiene en /dev sólo las entradas correspondientes a los dispositivos que hay conectados al sistema. Así se soluciona el problema del /dev superpoblado.
- No se usa el número mayor y menor para reconocer a cada dispositivo.
- Permite dar un nombre fijo para cada dispositivo.
- Avisa mediante mensajes D-BUS para que cualquier programa del espacio de usuario pueda enterarse cuando un dispositivo se conecta o desconecta (esto es útil para HAL).

Características de udev

- udev hace que toda la política de nombres de dispositivo esté en espacio de usuario, y no en el kernel. Esto hace posible que un programa cualquiera pueda decidir el nombre de un dispositivo.
- udev respeta la forma de nombrar dispositivos definida en el LSB, aunque permite que los usuarios usen otros nombres.
- El proceso (udevd) ocupa poca memoria y no necesita ejecutarse siempre. Esto favorece a los sistemas embebidos y equipos poco potentes.

Funcionamiento de udev

- Enchufamos nuestro dispositivo
- El kernel se da cuenta de que algo está en marcha, detecta nuestro dispositivo y le pasa el control a hotplug1
- Hotplug examina el tipo de dispositivo y carga módulos si son necesarios para dejar el dispositivo listo para usar
- Hotplug notifica a udev que tiene un nuevo dispositivo conectado
- udev se da por enterado, y se va a la partición sysfs a buscar la identificación del dispositvo; estos datos son contrastados con una serie de reglas de nomenclatura y le asigna una entrada de dispositivo.
-
-

Creación de reglas para udev

- Para la creación de reglas de udev debemos irnos al directorio /etc/udev/rules.d/
- En este directorio podremos crear un archivo de reglas de la siguiente manera:
  - xx-nombre-descriptivo.rules
- Hay que decir que tenemos unas posibilidades enormes, podemos definir reglas en función de a qué bus se conecta un dispositivo, a qué puerto pci o usb, que lo llame en función de un programa que llamemos nosotros al conectar el disposito, que coincida con un determinado valor en las propiedades de sysfs, y etc.

Creación de reglas de udev

- BUS el tipo de bus al que se conecta.
- KERNEL nombre del kernel.
- SYSFS_fichero Coincidir el contenido del fichero de sysfs que definimos (sólo se pueden comprobar 5 archivos por norma) Atencion:en la release 018 se ha cambiado por SYSFS{fichero}.
- PROGRAM Llama a un programa externo y considera que la condición se cumple si el programa devuelve 'éxito'.

Creación de reglas de udev

- RESULT se usa para afinar más con los programas externos, pidiendo esta regla que la salida del programa anterior coincida con lo que hemos definido.
- NAME El nombre que le vamos a dar al dispositivo.
- SYMLINK En caso de que queramos crear un enlace simbólico al dispositivo; se pueden crear más de uno.
- Tanto NAME, como SYMLINK aceptan ``comodines'' de la que se enuncia ahora los dos principales (la lista completa con 'man udev'):

Administración Avanzada de Redes de Sistemas UNIX/Linux

Dispositivos en sysfs

- Para poder definir las reglas en udev es fundamental conocer las características de los dispositivos, pero para eso es necesario buscar cómo se identifican al sistema en sysfs.
- Para cada objeto añadido en el árbol del modelo de controladores se crea un directorio en sysfs. La relación padre/hijo se refleja con subdirectorios bajo /sys/devices/ (reflejando la capa física). El subdirectorio /sys/bus se puebla con enlaces simbólicos, reflejando el modo en el que los dispositivos pertenecen a diferentes buses. /sys/class muestra dispositivos agrupados de acuerdo a su clase, como por ejemplo red, mientras que /sys/block/ contiene los dispositivos de bloques.

Capa HAL

- Para hacer más sencilla la labor de los desarrolladores de aplicaciones que hacen uso de hardware, se ha creado la especificación HAL. El objetivo es ofrecer una API unificada de programación a los desarrolladores de aplicaciones de forma que puedan manejar de forma sencilla el hardware.
- Las labores principales en las que se centra HAL es en las de permitir a las aplicaciones descubrir y configurar el hardware que se encuentra en la máquina. Por ejemplo, una aplicación de gestión de fotos podrá preguntar a HAL como hablar con la cámara de fotos, o una aplicación de videoconferencia, cuál es el dispositivo de vídeo y el de audio.

Sistemas de Ficheros

Sistemas de Ficheros

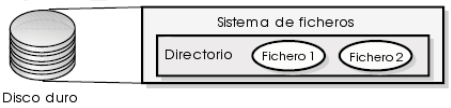
- Jerarquía del sistema de ficheros
- Tipos de sistemas de ficheros
- Puntos de montaje
- Gestión de sistemas de ficheros
- Enlaces “duros” y “blandos”
- Permisos
- Cuotas de disco

Introducción

- La gestión de los sistemas de ficheros UNIX es una de las tareas más importantes del administrador de sistemas.
- Responsable de asegurar que los usuarios tienen acceso a los ficheros que necesitan, que no son corruptos y son seguros.
- Administrar un sistema de archivos incluye muchas tareas:
  - Hacer que tanto los ficheros locales como los remotos estén disponibles para el usuario.
  - Monitorizar y gestionar los recursos de disco
  - Limitar el acceso a los datos. Confidencialidad.
  - Configurar nuevos dispositivos de almacenamiento.
  - ...

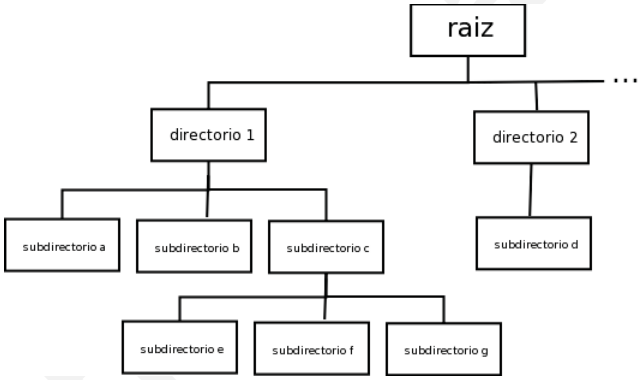
Introducción

- Sistema de archivos de UNIX
  - Una de las mejores ideas de este S.O
  - Estructura jerárquica, con forma de árbol, una única raíz.
- Para que un S.O pueda acceder a archivos y datos de un S.F, este tiene que estar admitido por el kernel.
- "Simplemente" una colección de objetos organizados para:
  - Conseguir acceso óptimo a los datos
  - Asegurar la integridad de los datos
  - Proporcionar manejabilidad y escalabilidad



Jerarquía del Sistema de Ficheros

- Todos los archivos, independientemente del dispositivo en el que se encuentren, están accesibles bajo una jerarquía de directorios con una única raíz (/).
- El directorio es el elemento que permite crear la jerarquía clásica de un sistema UNIX.



Jerarquía del Sistema de Ficheros

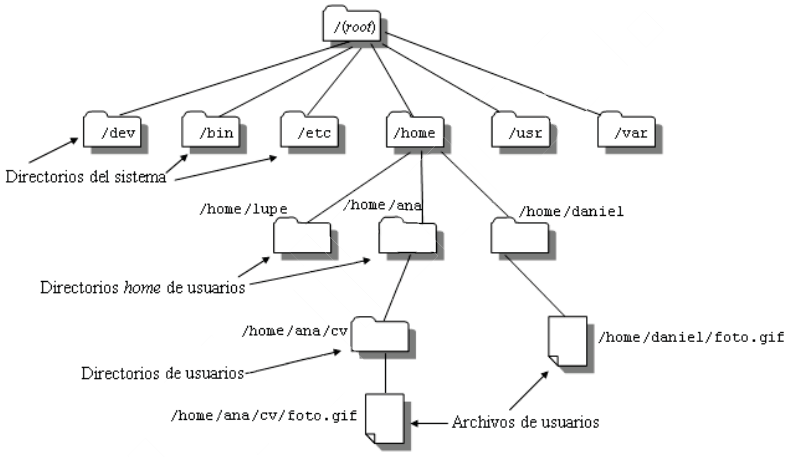
- Primeros UNIX tenían estructura de directorios distinta: directorios de usuarios en /usr, ejecutables en /etc...
- En los inicios de Linux, las distribuciones situaban cada fichero donde querían.
- Se necesitaba una estructura robusta. En 1994 se publicó *Linux Filesystem Standard Structure*.
- Evolución para abarcar todos los sistemas UNIX. *Filesystem Hierarchy Standard*.
- Este documento especifica dónde se encuentran los archivos de configuración, las aplicaciones, los directorios de trabajo, los archivos temporales, etc.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Jerarquía del Sistema de Ficheros

- Directorios definidos por el FHS: `bin boot dev etc home lib media mnt opt root sbin srv tmp usr var`.
- Distribuciones y administradores locales pueden usar mas directorios: `initrd lost+found proc sys`.
- Nada aparte de estos directorios, con la excepción de imágenes del kernel necesarias para el arranque del sistema (`vmlinuz, initrd.img...`). Sistemas antiguos.
- El árbol puede ser “roto” en varias partes.
- `/`, `/usr`, `/var` y `/home`.
- Su diseño permite que pueda estar repartido por red.

Jerarquía del Sistema de Ficheros



/bin

- Contiene comandos básicos de usuario.
- Estos tienen que ser los imprescindibles para arrancar en modo mono-usuario.
- El resto bajo `/usr/bin` .
- También incluye la shell y las órdenes que ejecutan indirectamente los scripts
- Requisitos mínimos: `cat chgrp chmod chown cp date dd df dmesg echo false hostname kill ln login ls mkdir mknod more mount mv ps pwd rm rmdir sed sh stty su sync true umount uname`

/boot

- Se sitúan los archivos utilizados por el cargador del *kernel*.
- Los ficheros exactos dependen del sistema operativo, de la arquitectura, de la distribución y del gestor de arranque.
- En Linux tendremos, por cada kernel que esté configurado en el arranque:
  - vmlinuz: Imagen del kernel a cargar.
  - initrd.img: Sistema de ficheros inicial, encargado de configurar y arrancar el sistema.
  - config: Fichero de configuración del kernel.
  - System.map: Símbolos del *kernel* y sus dir. en memoria.
  - vmlinux: Imagen del kernel como un ejecutable.

/dev

- Directorio que contiene los nodos de dispositivo pertenecientes al sistema.
- Todos los dispositivos del sistema (de caracteres y de bloques) deben estar en este directorio.
- *Script* encargado de crear otros nuevos. MAKEDEV.
- Dispositivos identificados con 2 números: *major y minor*.
- Lista de todos los dispositivos reconocidos por el kernel en: `Documentation/devices.txt`
- Algunos dispositivos no se refieren a *hardware* concreto: `mem, kmem, null, port, zero, full, random, urandom...`

/etc

- Directorio que contiene los archivos de configuración del sistema.
- Configuración general del sistema:
  - `exports, fstab, ftpusers, group, host.conf, hosts, inetd.conf, inittab, init.d, issue, motd, mtab, passwd, profile, protocols, rcN.d, resolv.conf, securetty,services, shadow, shells, skel, syslogd.conf...`
- Cada paquete puede tener un fichero de configuración o un directorio con varios ficheros (como `/etc/X11`).

/home

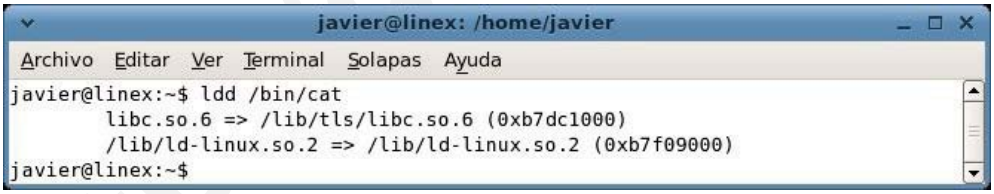
- Directorio que contiene los “directorios de inicio” de los usuarios del sistema, salvo el usuario.
- Normalmente, los directorios de inicio de cada usuario estarán en `/home/login`.
- Al crear una cuenta nueva, se copia en ella el contenido de `/etc/skel`.

/mnt

- Punto de montaje para sistemas de archivos temporales.
- Punto de montaje temporal, y sin que aparezca constancia alguna en el fichero de configuración `/etc/fstab`.

/lib

- Contiene las librerías que comparten los programas.
- En `/lib/modules` se encuentran los módulos disponibles para el *kernel*.
- Los ejecutables de `/bin` y `/sbin` solo pueden depender de bibliotecas dinámicas que estén en `/lib`.
- Se puede saber de qué bibliotecas dinámicas depende un determinado ejecutable con la orden `ldd`.



/opt

- Directorio usado para la instalación de software adicional.

/root

- Directorio *home* del superusuario *root*.
- `/home` puede pertenecer a un sistema de ficheros distinto, y el usuario *root* puede necesitar acceder a su directorio incluso cuando no estén montados todos los discos.

/sbin

- Utilidades y programas ejecutables dedicadas a la administración del sistema.
- Ordenes habituales: `fdisk fsck getty halt ifconfig init mkfs mkswap reboot route swapon swapoff`

/tmp

- Directorio con permiso de escritura para todos los usuarios.
- Cualquier programa o *script* puede usar este directorio para dejar ficheros temporales.
- Normalmente, se borra en cada reinicio del sistema.

Administración Avanzada de Redes de Sistemas UNIX/Linux

La jerarquía /usr

- Segundo directorio más importante del sistema, tras el raíz.
- Contiene datos que se pueden compartir entre máquinas, y todo su contenido ha de ser de solo lectura.
- Debe contener al menos los directorios `bin include lib local sbin share`.
- Opcionalmente `X11R6 games src`.

/usr/bin

- Programas de usuario, con excepción del sistema gráfico X.

/usr/include

- Archivos de cabecera (.h) de las librerías instaladas en el sistema. (`stdio.h, stdlib.h...`)
- Los ficheros *include* del kernel deben estar en el directorio `/usr/include/linux`

/usr/lib

- `/lib` contiene librerías básicas para el funcionamiento
- `/usr/lib,` librerías usadas por las aplicaciones.
- También bibliotecas estáticas y ficheros .so, necesarios para compilar programas.

/usr/local

- Jerarquía similar a la de `/`, `/usr` o `/opt`, pero para aplicaciones instaladas localmente por el administrador.
- Cometido similar a `/opt`.
- Debe contener únicamente los directorios `bin etc games include lib man sbin share src`.

/usr/sbin

- Aplicaciones ejecutadas normalmente con privilegios de superusuario
- No son necesarias para reparar el sistema o una emergencia.

/usr/share

- Se almacenan los archivos independientes de la plataforma. En `/usr` los dependientes de la arquitectura.
- No contiene binarios ejecutables, sino imágenes, sonidos, etc.

/usr/src

- El código fuente de cualquier paquete debe estar en este directorio.
- Este código fuente que debe usarse solo como referencia. `/usr/src` no es el lugar apropiado para compilar dicho código.

/usr/X11R6

- Sistema de ventanas X, Versión 11 Release 6
- Jerarquía reservada al sistema de ventanas X.
- Contiene los directorios:
  - bin → Ordenes
  - include → Ficheros *include* para compilar en C
  - lib → Bibliotecas (dinámicas y estáticas)
  - man → Páginas de manual

/var

- Contiene diferentes archivos de datos del sistema
  - correo electrónico (`/var/mail`)
  - archivos históricos (`/var/log`)
    - *messages*
    - *dmesg*
    - ...
- En general, datos que puedan sufrir modificaciones
- Subdirectorios: `cache games lib local lock log mail opt run spool tmp www`.

Administración Avanzada de Redes de Sistemas UNIX/Linux

/proc

- Representa una estructura virtual. `/proc` son un medio de enviar y recibir información directamente del kernel.
- Está implementado sobre memoria y no se guarda en disco.
- Los "archivos" son entradas que el núcleo genera automáticamente para obtener información del propio *kernel*
- El núcleo durante su arranque pone en funcionamiento un *pseudo-filesystem.*
- Existe un directorio por cada PID del sistema.
- Otros ficheros tienen información general: `cmdline cpuinfo devices filesystems interrupts ioports loadavg dma meminfo modules partitions swaps uptime version...`

Tipos de Sistemas de Ficheros

- Los que reconoce el kernel están en /proc/filesystems.
  - vfat: Usado a partir de Windows 95, compatible con MSDOS pero con posibilidad de nombres de fichero largos.
  - ntfs: A partir de Windows NT, añade características de seguridad (permisos, dueños, etc).
  - iso9660: Sistema de fichero utilizado en los CDs de datos.
  - ext2: Sistema de ficheros por excelencia en Linux.
  - ...

Tipos de Sistemas de Ficheros

- Con "*journal*":
  - ext3: Siguiente versión del ext2, idéntico pero con adición de *journal*. El más utilizado hoy en día.
  - reiserfs: Primer sistema de ficheros con journal para Linux; muy eficiente con directorios grandes.
  - jfs: Sistema de ficheros con journal de IBM.
  - xfs: Sistema de ficheros con journal creado por SGI para su plataforma IRIX
  - ext4: Anunciado el 10 de octubre de 2006 como una mejora compatible con ext3. Soporta volúmenes de hasta 1024 petabytes.

Puntos de montaje

- No todos los ficheros del árbol de directorios se encuentran en el mismo disco.
- Punto de montaje: directorio que pertenece a un disco (o partición) distinto.
- El kernel incluye información de todas las particiones y puntos de montaje asociados en el fichero `/proc/mounts` (el fichero `/etc/mtab` contiene prácticamente la misma información).
- También se pueden consultar los puntos de montaje junto con los discos o particiones que están montadas en ellos con las órdenes "`mount`" y "`df`".

Herramientas útiles

- `Mount`: Muestra las particiones, puntos de montaje, tipo de partición y opciones de cada una de ellas:
- `df`: Muestra cada una de las particiones con ficheros reales montadas en el sistema, el punto en el que está montada, su capacidad y su uso:

Montaje de sistemas de ficheros existentes

- Crear el directorio si no existe:

```
#mkdir /home/usuario/punto_montaje
```

- Hacer visible el sistema de ficheros bajo ese directorio:

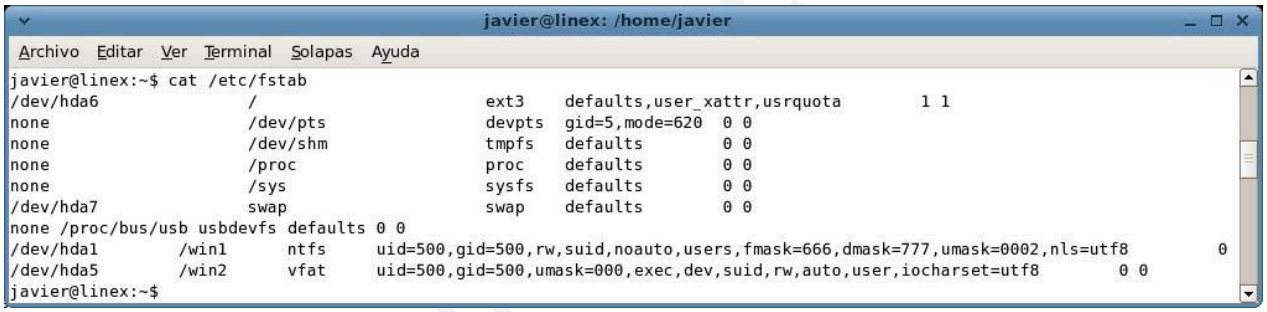
```
#mount -t ext2 -o rw /dev/hda3 /home/usuario/punto_montaje
```

- Si queremos desmontar (o hacer invisible) un sistema de ficheros que esté montado:

```
#umount /home/usuario/punto_montaje
```

/etc/fstab

- Automatización del montaje de sistemas de ficheros en el arranque.



Opciones de los sistemas de ficheros

- `defaults`: Ninguna opción especial.
- `rw`: Permisos de lectura y escritura.
- `ro`: Sólo lectura.
- `loop`: Permite montar un fichero en lugar de un dispositivo de bloques.
- `noauto`: No montar automáticamente con "`mount -a`" (esto es, en el inicio).
- `user`: Puede montarlo y desmontarlo un usuario distinto de root.
- Para otras opciones, consultar la página de manual de mount

Añadir nuevo S. F.

- Para añadir un disco nuevo al sistema, es necesario:
  - Preparar las particiones del disco
  - Crear el sistema de ficheros
  - Elegir (y crear) el punto de montaje
  - Montar el nuevo disco,
  - Modificar /etc/fstab para que se monte automáticamente en el arranqueu

Añadir nuevo S. F. Particiones de disco

- Preparar particiones
  - Uso de las órdenes "`fdisk`", "`cfdisk`" o "`sfdisk`" para modificar la tabla de particiones.
    - `fdisk`: Herramienta original, todo "a mano".

- Crear el sistema de ficheros.
  - Hay que elegir el tipo de sistema de ficheros más adecuado.
  - Creación del sistema de ficheros con la orden "mkfs":
    - `mkfs.ext3 /dev/hda3`

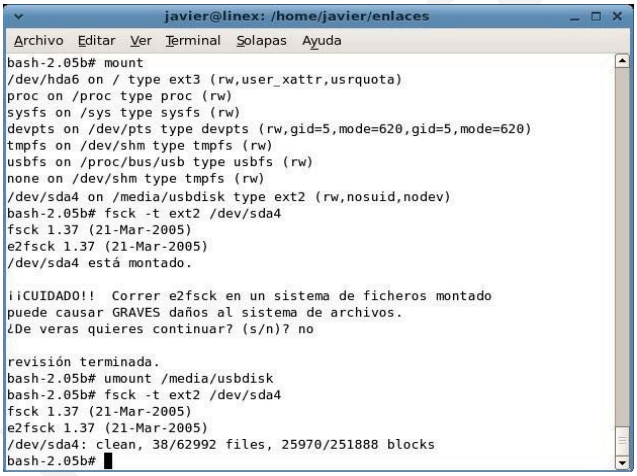
Administración Avanzada de Redes de Sistemas UNIX/Linux

Añadir nuevo S. F. Montaje

- En primer lugar, elegir el punto de montaje:
  - Puede ser uno de los directorios definidos: `/usr`, `/var`, `/home`, `/tmp`, `/var/tmp`, o crear otro, por ejemplo `/home/usuario/montaje`
  - Los permisos que tenga este directorio no son muy relevantes, ya que desaparecerán una vez montado el nuevo sistema de ficheros.
- Luego se monta el dispositivo sobre ese punto
  - `mount dispositivo punto_montaje`
- Por cada punto de montaje que tengamos en el sistema, es necesaria una línea nueva en el fichero `/etc/fstab`.

Integridad del Sistema de Ficheros

- Al arrancar se hace una comprobación de los S. F.
- Esta comprobación se realiza con la orden "`fsck`", y puede realizarse a mano (siempre con el disco sin montar)



Ficheros, directorios y algo más. Inodos

- Aparte de ficheros y directorios, existen otras estructuras de datos específicas que soportan y mantienen el S.F. o gestionan las tareas y rutinas.
- Tabla de i-nodos. Sigue la pista de los ficheros en el S.F.
  - El i-nodo es una estructura de datos que contiene muchos atributos del fichero (permisos, fechas, ubicación, pero no el nombre). (`ls`)
  - Cada inodo queda identificado por un número entero, único dentro del sistema de ficheros
  - De esta forma, un directorio es simplemente un archivo que contiene una lista de números de inodos y nombres de archivos.

Ficheros, directorios y algo más. Inodos.

- Dentro del inodo se guarda el número de links del archivo.
- Un link es una entrada en un directorio
- Un archivo puede tener muchos links. Son el mismo archivo con dos nombres diferentes.
- Si borramos uno de estos archivos, no se borrará el contenido del archivo, sino únicamente la entrada de directorio
- Se decrementa la cantidad de links al inodo.
- Solamente se eliminará el contenido del archivo cuando el número de links llegue a cero.
- A este tipo de links se los llama *hard links*

Administración Avanzada de Redes de Sistemas UNIX/Linux

Ficheros, directorios y algo más. Inodos.

- Existe otro tipo de links, llamado soft links o también *symlinks (Symbolic Link)*.
- Estos no comparten el inodo del otro archivo, ni aumentan la cantidad de *links* que tiene el otro archivo
- Se trata únicamente de un puntero a otro archivo, que se puede encontrar en cualquier parte del sistema de archivos.
- Muchos programas usados por administradores UNIX trabajan con el número de inodo para identificar archivos.
  - fsck

Ficheros, directorios y algo más. Superbloques.

- Aparte de ficheros y directorios, existen otras estructuras de datos específicas que soportan y mantienen el S.F. o gestionan las tareas y rutinas.
- Superbloque. Cada partición UNIX contiene un bloque especial que contiene información sobre el sistema de ficheros
  - Tipo de sistema de ficheros
  - Tamaño y fecha de modificación
  - Lista de bloques libres y asignados y el primer inodo, que apunta al directorio raíz (/).
- Está replicado por seguridad.

Enlaces simbólicos o “blandos”

- Es una entrada de directorio que contiene una ruta de acceso, apuntando hacia otra entrada de otro directorio.
- Si apunta a un directorio que no existe, el enlace está “roto”.
- Comando ln (*link*):

```
ln [opciones] origen [destino]
```

- Si no se especifica el destino, el enlace se creará en el directorio actual y con el mismo nombre que la entrada de directorio origen.

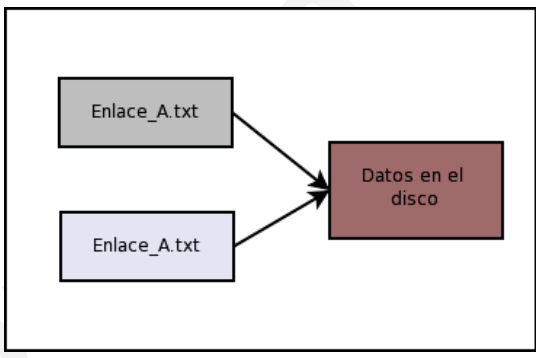
Enlaces “duros”

- Es más robusto pero menos flexible.
- Nunca puede estar “roto”.
  - No contiene ruta de acceso.
  - Apunta directamente a la zona del sistema de archivos en el que se encuentra el archivo referenciado.
  - Si el archivo original cambia, el enlace sigue referenciando a la zona que almacena el archivo.
  - Si el archivo original se elimina, el contenido del archivo sigue siendo apuntado por el enlace duro.
- Se comportan de forma idéntica al archivo desde el que se hace el enlace

Administración Avanzada de Redes de Sistemas UNIX/Linux

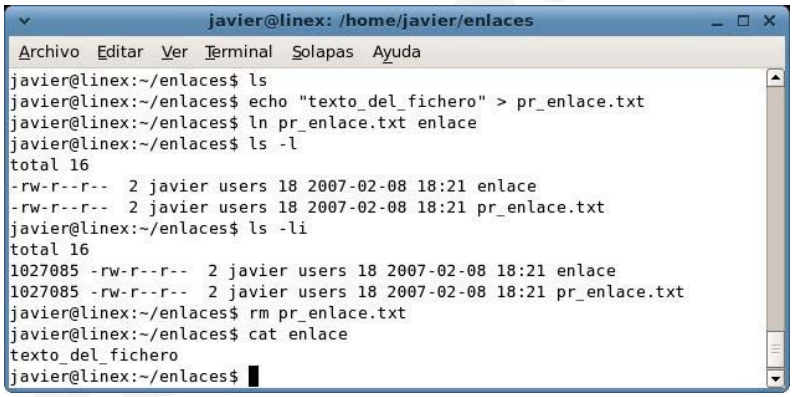
Enlaces “duros”

- El sistema de archivos guarda el número de referencias a la zona de almacenamiento del archivo.
- ls -l da el *número de referencias*. Este campo se incrementa por cada referencia y/o enlace duro.



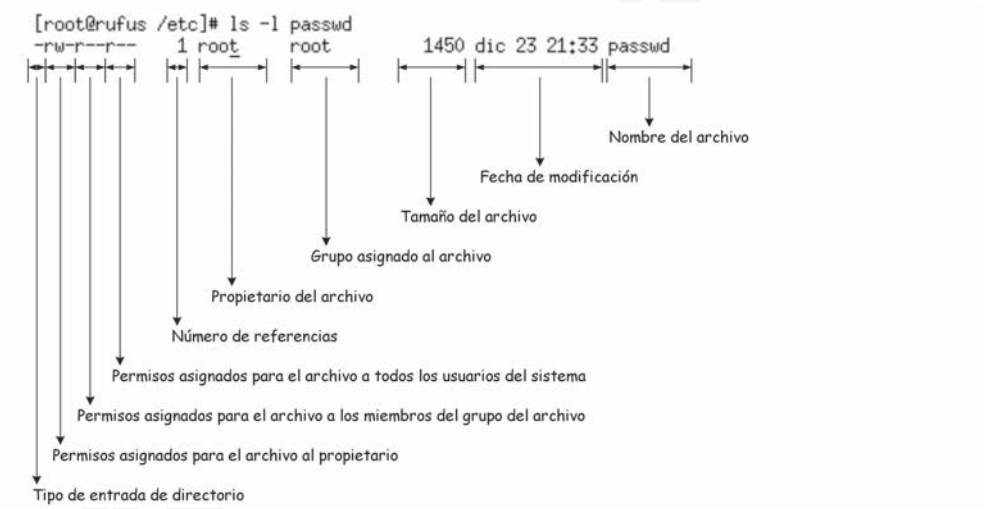
Enlaces “duros”

- El sistema de archivos guarda el número de referencias a la zona de almacenamiento del archivo.
- ls -l da el *número de referencias*. Este campo se incrementa por cada referencia y/o enlace duro.



Permisos

- Cada fichero o directorio puede disponer de la serie de permisos:



Permisos

- Establecer permisos:
  - chown: Modifica el propietario de archivos y directorios.
    - `chown [opciones] propietario archivo [archivo ...]`
  - chgrp: Modifica el grupo asignado a archivos y directorios.
    - `chgrp [opciones] grupo archivo [archivo...]`
  - chmod: Modifica los permisos del archivo en los tres niveles
    - `chmod [opciones] permisos archivo [archivo...]`

Cuotas de disco

- Las cuotas permiten al administrador establecer limitaciones de cantidad de almacenamiento
  - Inodos
  - Bloques de disco
- La idea es que los usuarios no pueden superar la limitación de espacio que se les imponga.
- Si hay más de un sistema de ficheros donde el usuario puede crear ficheros, la cuota se establecerá de forma separada para cada uno de ellos.
- Existen herramientas que facilitan la gestión de la política de cuotas

Cuotas de disco

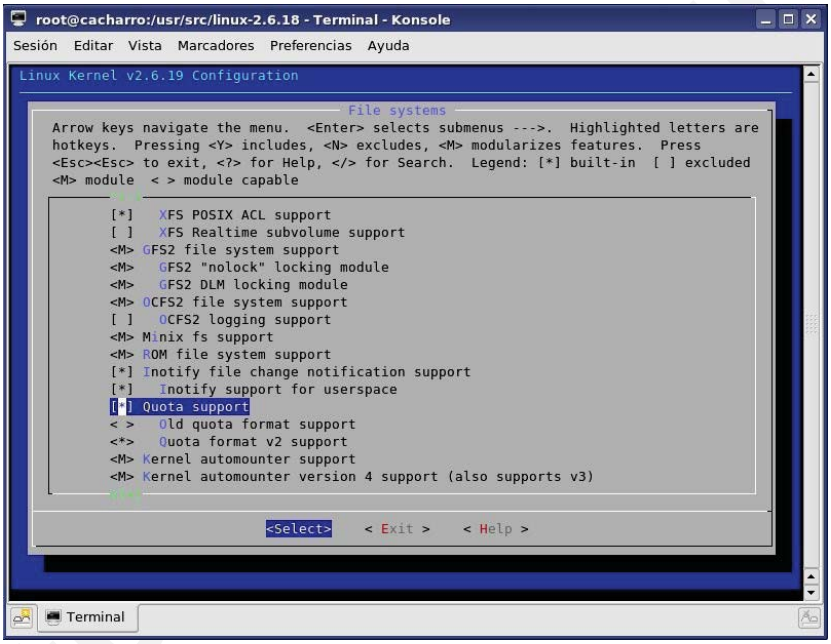
- En primer lugar debemos saber si nuestro núcleo las soporta.

```
javier@linex:~$ dmesg |grep quota
VFS: Disk quotas dquot_6.5.1
```

- Disponer de los paquetes necesarios. En Debian:
  - quota
  - quotatool
- Activarlas.
- Definirlas.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Cuotas de disco



Cuotas de disco. Activación

- Las cuotas se activan modificando el fichero `/etc/fstab` y añadiendo *usrquota* o *grpquota*

```
/dev/hda6  /   ext3    defaults,user_xattr,usrquota     1 1
/dev/hda7 swap swap    defaults          0 0
```

- El sistema se activa al reiniciar el sistema. En el siguiente arranque quotacheck generará los ficheros apropiados.

```
Checking quotas...
done
Turning on quotas
```

  - Si no, volver a montar el sistema de ficheros y crear los ficheros de forma manual.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Cuotas de disco. Asignar cuotas

- Asignar cuotas a un usuario concreto
  edquota -u nombre_usuario

```
Disk quotas for user javier (uid 1002):
  Filesystem   blocks     soft       hard       inodes    soft$
  /dev/hda6    65000      60000      65000       1037      0$
```

- Asignar cuotas a un grupo concreto
  edquota -u nombre_grupo

Cuotas de disco. Asignar cuotas

- "Periodo de gracia" de las cuotas.
  edquota -t

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem   Block grace period     Inode grace period
  /dev/hda6          7days                 7days
```

- Establecen el tiempo límite antes de que se fuerce al tamaño *soft limit*.

Cuotas de disco. Comandos útiles.

- Quotacheck
  - Utilizado para buscar el uso de un sistema de ficheros con cuotas activada.
  - Actualiza el fichero "aquota.user".
- Repquota
  - Resume la información de las cuotas para un S.F.

Cuotas de disco. Comandos útiles.

```
bash-2.05b# repquota -a
*** Report for user quotas on device /dev/hda6
Block grace time: 7days ; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      -- 4191236       0       0          142722    0     0
...
javier    +-   59836   60000   65000  6days     1029    0     0
#500      --     160       0       0              15    0     0
#1003     --   24292       0       0              70    0     0

bash-2.05b#
```

Cuotas de disco. Límites

```
javier    +-   59836   60000   65000  6days     1029    0     0
```

- Si vamos abarcando mas bloques:

```
javier@linex:~$ cp fichero.pdf fichero2.pdf
hda6: warning, user block quota exceeded.
javier    +-   60556   60000 65000  6days    1030      0     0
```

- Si seguimos creando más

```
javier    +-   64876   60000 65000  6days    1036      0     0

javier@linex:~$ cp fichero.pdf fich8.pdf
hda6: write failed, user block limit reached.
cp: escribiendo «fich8.pdf»: Se ha excedido la cuota de disco

javier    +-   65000   60000 65000  6days    1037      0     0
```

Cuotas de disco. “Periodo de gracia”

```
javier    +-   59836  60000  65000  6days    1029   0      0
```

- Superamos el límite *soft*. 10 minutos de periodo de gracia

```
javier@linex:~$ cp fichero.pdf fichero2.pdf
hda6: warning, user block quota exceeded.
javier    +-  60556  60000 65000  00:10   1030     0      0
...
javier    +-  60556  60000 65000   none    1030     0      0

javier@linex:~$ cp fichero.pdf fichero3.pdf
hda6: write failed, user block quota exceeded too long.
cp: no se puede crear el fichero regular «fichero3.pdf»: Se ha
   excedido la cuota de disco
```

Cuotas de disco. UBUNTU Live

- Seguimos la misma estrategia
  - Soporte del nucleo

```
javier@linex:~$ dmesg |grep quota
VFS: Disk quotas dquot_6.5.1
```

  - Paquetes necesarios.
    - quota (apt-get)
    - quotatool (dpkg)
  - Activarlas.
  - Definirlas.

Cuotas de disco. UBUNTU Live

- Haremos cuotas en un disco externo.
- Preparación del disco
  - Preparar la tabla de particiones
    fdisk /dev/sda4
  - Crear el sistema de ficheros y formatear
    mkfs.ext2 /dev/sda4
- */etc/fstab*

```
unionfs / unionfs rw 0 0
tmpfs /tmp tmpfs nosuid,nodev 0 0
/dev/hda7 swap swap defaults 0 0
/dev/sda4 /media/usbdisk ext2 rw,usrquota 0 0
```

Administración Avanzada de Redes de Sistemas UNIX/Linux

Cuotas de disco. UBUNTU Live. Activación.

```
root@ubuntu:~# quotaon -avug
quotaon: Cannot find quota file on /media/usbdisk [/dev/sda4]
   to turn quotas on/off.

root@ubuntu:/media/usbdisk# touch aquota.user
root@ubuntu:/media/usbdisk# chmod 600 aquota.user
root@ubuntu:/media/usbdisk# mount -o remount /media/usbdisk/
root@ubuntu:/media/usbdisk# quotacheck -avumg
quotacheck: WARNING -  Quotafile /media/usbdisk/aquota.user was
   probably truncated. Can't save quota settings.. .
quotacheck: Explorando /dev/sda4 [/media/usbdisk] quotacheck:
   Old group file not found. Usage will not be subst racted.
quotacheck: Comprobados 3 directorios y 2 archivos.
root@ubuntu:/media/usbdisk# quotaon -avug
/dev/sda4 [/media/usbdisk]: user quotas turned on
```

Networking

Administración Avanzada de Redes de Sistemas UNIX/Linux

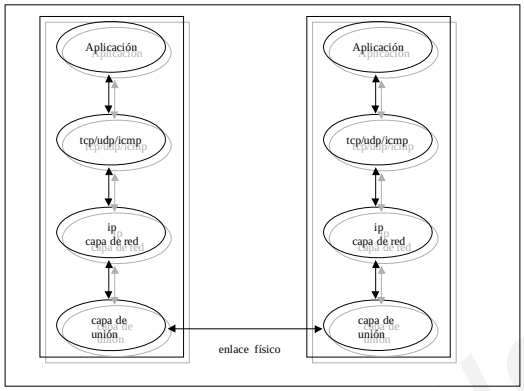
Networking

- Terminología *Networking*.
- OSI y TCP/IP.
- RFCs y protocolos.
- Direccionamiento IP.
- Segmentación y subredes.
- Planificación topológica de redes.
- Configuración de red TCP/IP (Ubuntu, Fedora y Solaris)
- IPv6.
- PPP y modem.
- Redes Wi-Fi.

Terminología Networking

- Dirección *Ethernet*: @ hardware (MAC) única de 48 bits expresados en 6 bloques de dos dígitos hexadecimales 8:0:10:b:34:3e (IEEE).
- Dirección IP: Dirección única de 32 bits divididos en 4 campos de 8 bits cada uno 158.49.98.12 (Internic).
- Loopback: interfaz que permite a un ordenador enviarse datagramas a sí misma. 127.0.0.1 es la @ de *loopback* por defecto.
- Datagrama: paquete de datos IP.
- Dirección broadcast: difusión a todos los nodos de la red.
- Backbone, segmentos, repeater, router, bridge, gateway, brouter, hub...

OSI y TCP/IP



Pila de protocolos TCP/IP

OSI	Internet	
Aplicación	Telnet FTP SMTP	NFS SNMP DNS
Presentación		
Sesión	TCP	UDP
Transporte		
Red	IP	
Enlace		
Físico		

Relación OSI TCP/IP

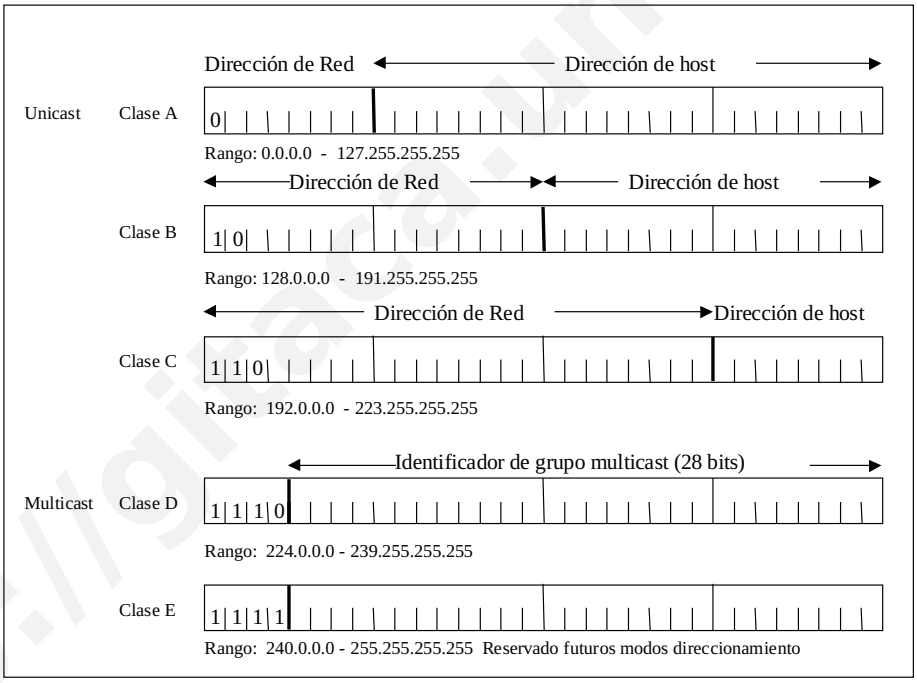
RFCs y Protocolos

- ***RFC791*** : Procolo IP.
- ***RFC950*** : Subred Internet.
- ***RFC1058*** : Routing.
- ***RFC1180*** : Tutorial TCP/IP.
- ***RFC1208*** : Glosario de redes.
- ***RFC1219*** : Asignación de @ subredes.

http://www.ietf.org/rfc.html

Direccionamiento IP

- Clase A: 127 redes de 16 Mega Nodos
- Clase B: 16.384 redes de 65.535 @ cada una
- Clase C: 2.097.152 redes de 254 nodos



Direccionamiento IPv4

Direccionamiento IP

- Direcciones especiales:
  - Direcciones de broadcast: xx.xx.xx.255
  - 127.0.0.1 loopback address (localhost)
  - Una @ de máquina no puede contener ningún cero (subred).
  - Dentro de cada clase de red se emplean direcciones para redes internas sin conexión directa a Internet:
    - Clase A: 10.0.0.0
    - Clase B: 172.16.0.0 a 172.31.0.0
    - Clase C: 192.168.0.0 a 192.168.255.0
  - Direcciones privadas sin uso en Internet.
  - Convención de xxx.xxx.xxx.1 para los routers

Segmentación y subredes

- División de una gran red lógica en pequeñas redes físicas.
- Justificación para la segmentación:
  - Sencillez en la administración de sistemas y redes.
  - Limitaciones eléctricas y topológicas
  - Separación de departamentos (workgroups)
  - Seguridad y aislamiento de tráficos de info sensible y pesada.
  - Subredes unidas a la red por routers, bridges, hubs, concentradores, brouters, gateways.
- Ejemplos:

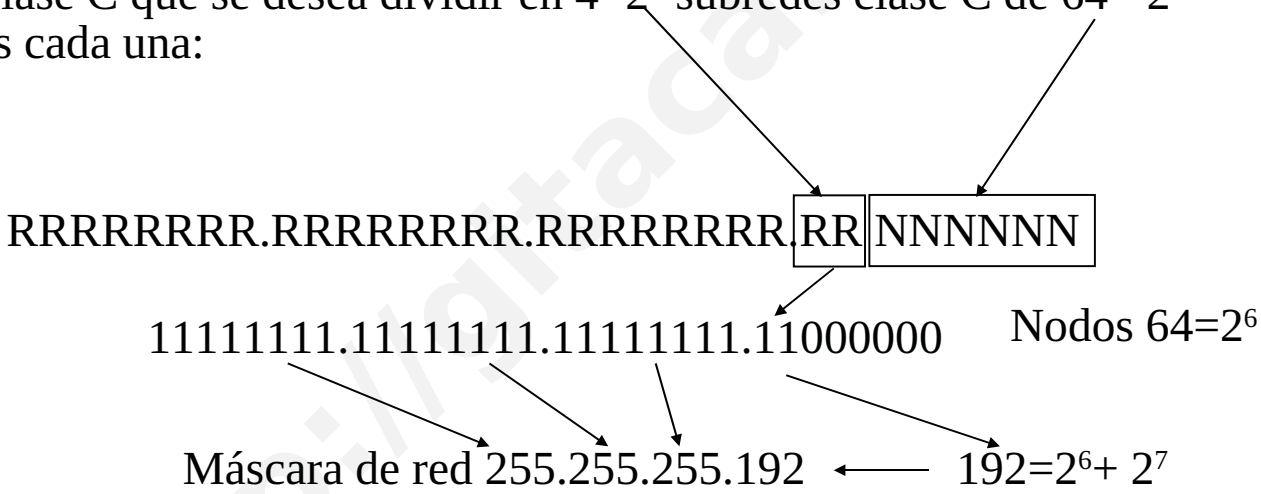
Subredes	*Nodos en la subred*	*Máscaras de red*
2	126	255.255.255.128 11111111.11111111.11111111.10000000
4	62	255.255.255.192 11111111.11111111.11111111.11000000
8	30	255.255.255.224 11111111.11111111.11111111.11100000
16	14	255.255.255.240 11111111.11111111.11111111.11110000
32	6	255.255.255.248 11111111.11111111.11111111.11111000
64	2	255.255.255.252 11111111.11111111.11111111.11111100

Subredes clase C y máscaras

- La máscara de subred se usa para la comunicación entre nodos de todas las subredes de una red principal.
- Máscaras de subredes:
  - Determinar dónde termina la dirección de red y dónde comienza la dirección de cada sistema principal (nodos).
  - Máscara de subred tiene a 1s los campos de red y a 0s los de nodo.

Ej.1: RRRRRRRR.RRRRRRRR.RRRRRRRR.NNNNNNNN

- Red clase C que se desea dividir en $4=2^2$ subredes clase C de $64=2^6$ nodos cada una:



Ej. 2:

11000000.10101000.00101010.00000000 ↔ 192.168.42.0 @Clase C

11111111.11111111.11111111.00000000 ↔ 255.255.255.0 Netmask

Para obtener $8=2^3$ subredes de $32=2^5$ nodos la máscara 255.255.255.224

11111111.11111111.11111111.11100000

255.255.255.224 ← $224=2^6+2^7+2^5$

Ej. 3:

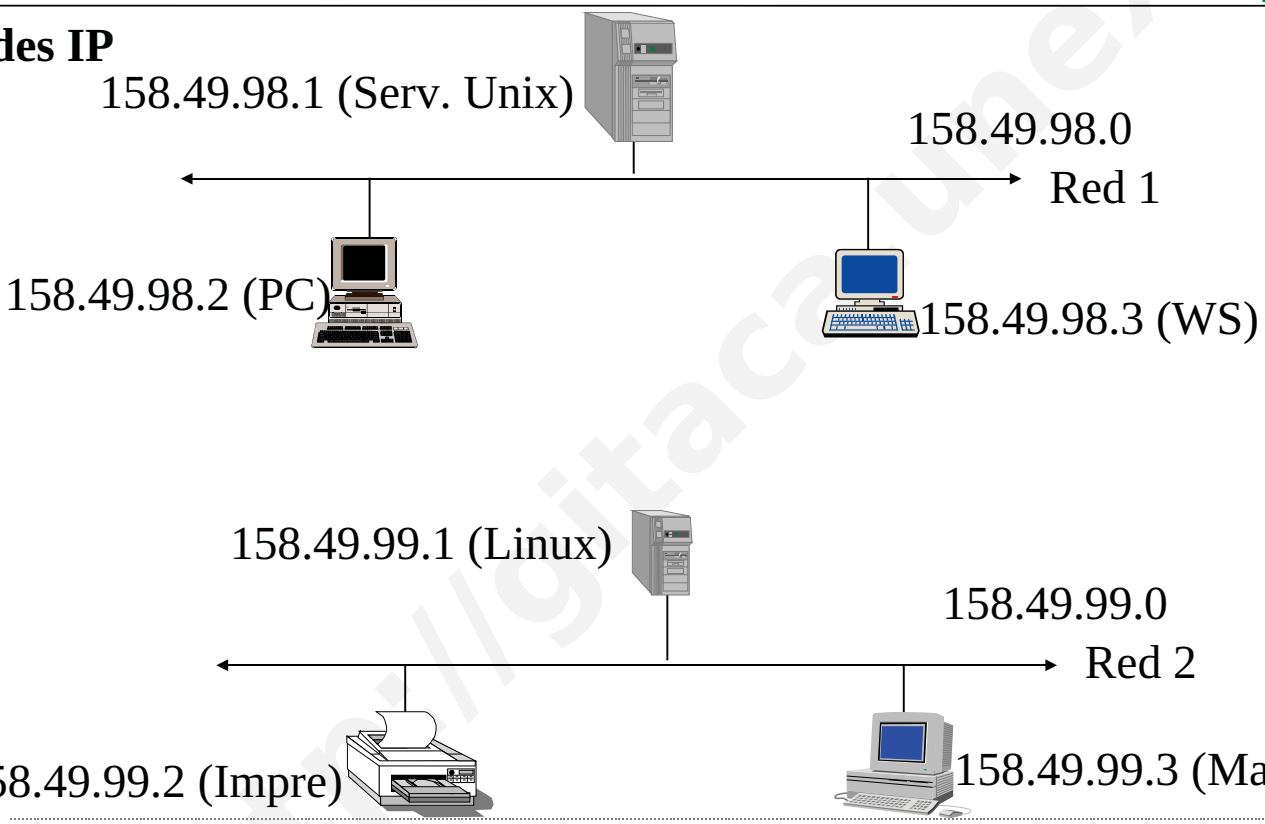
• Idea extensible a las máscaras de redes clase B y A

RRRRRRRR.RRRRRRRRNNNNNNNN.NNNNNNNN

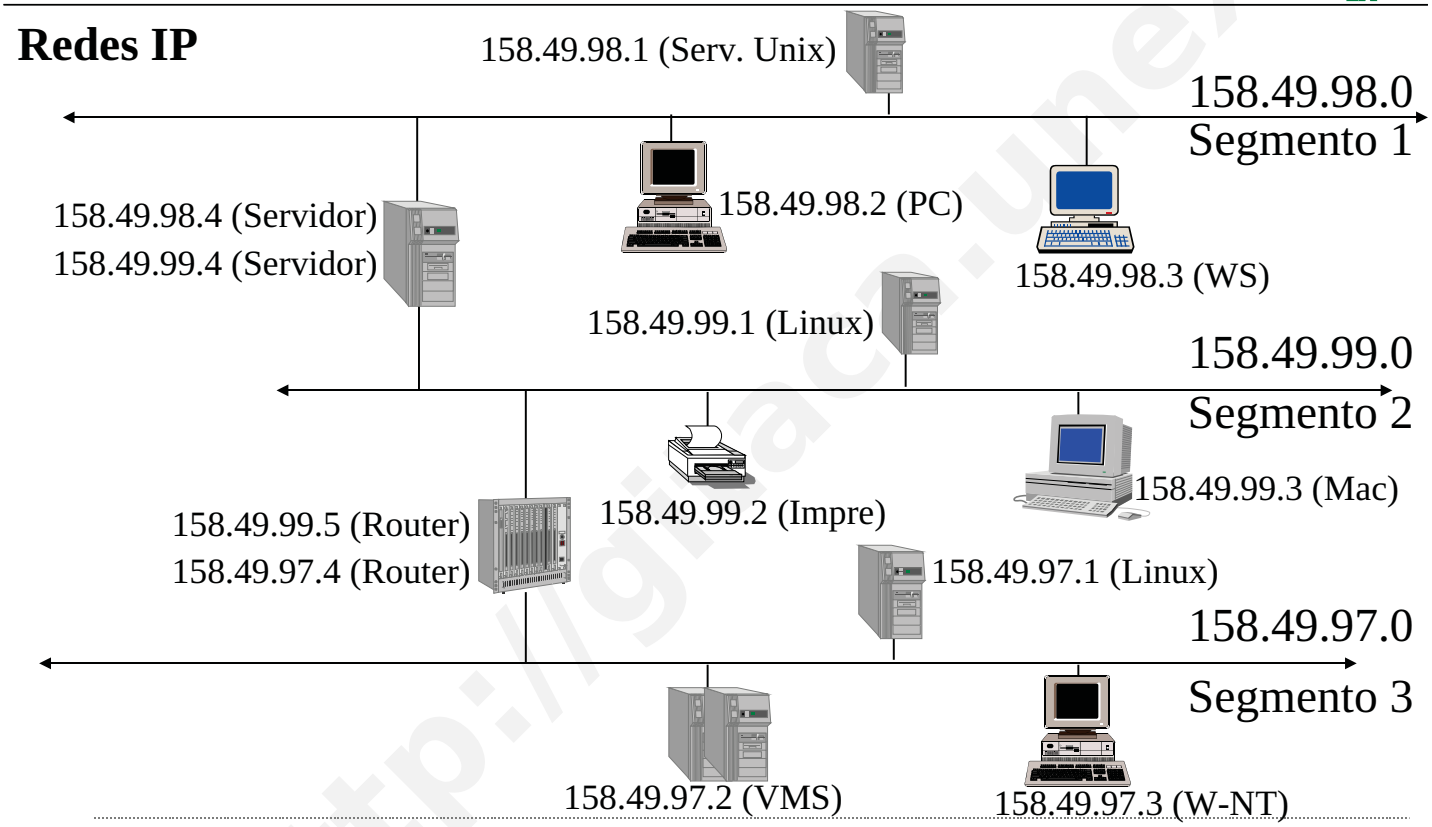
11111111.11111111.00000000.00000000

11111111.11111111.10000000.00000000 255.255.128.0

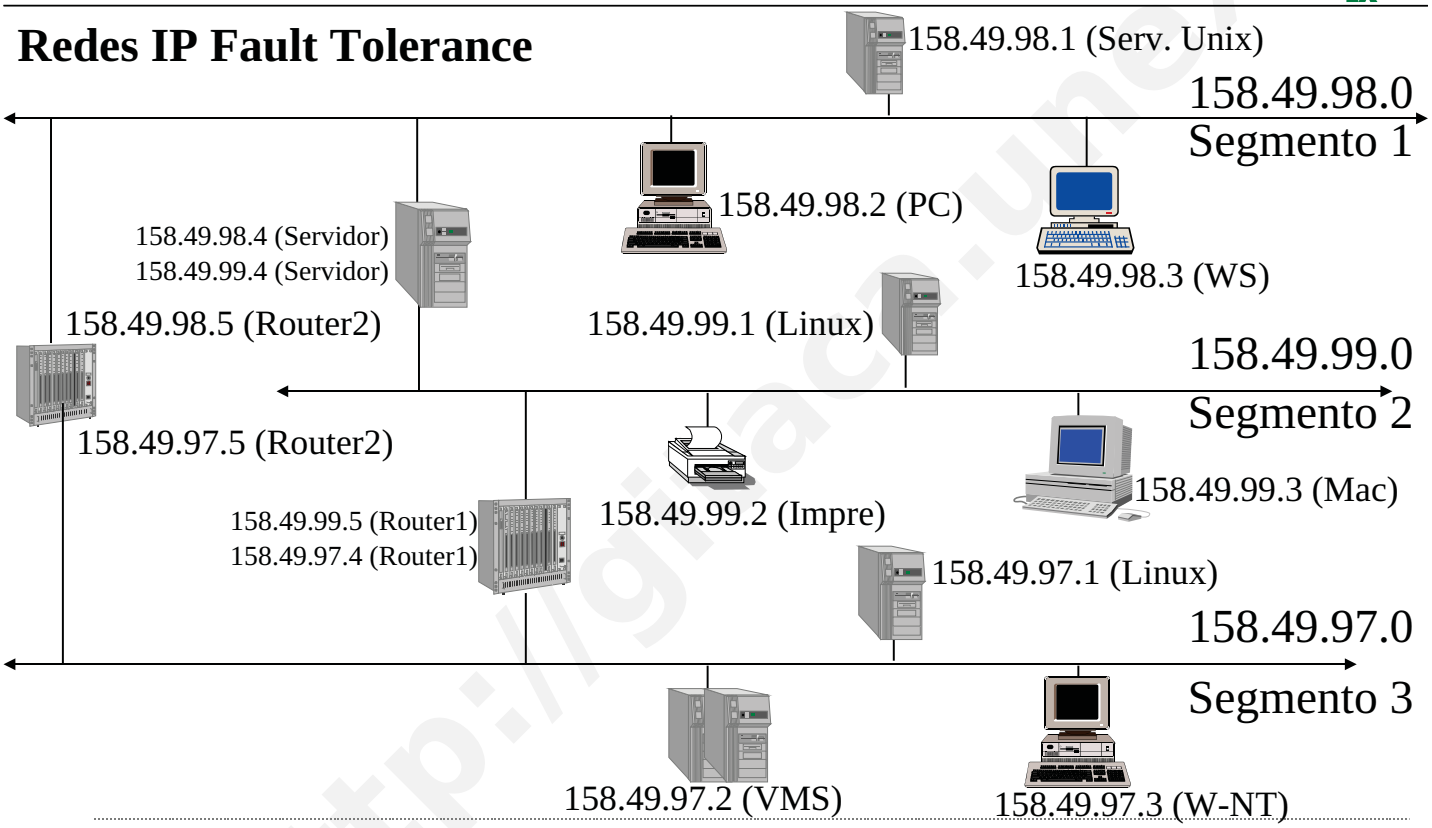
Redes IP



Redes IP



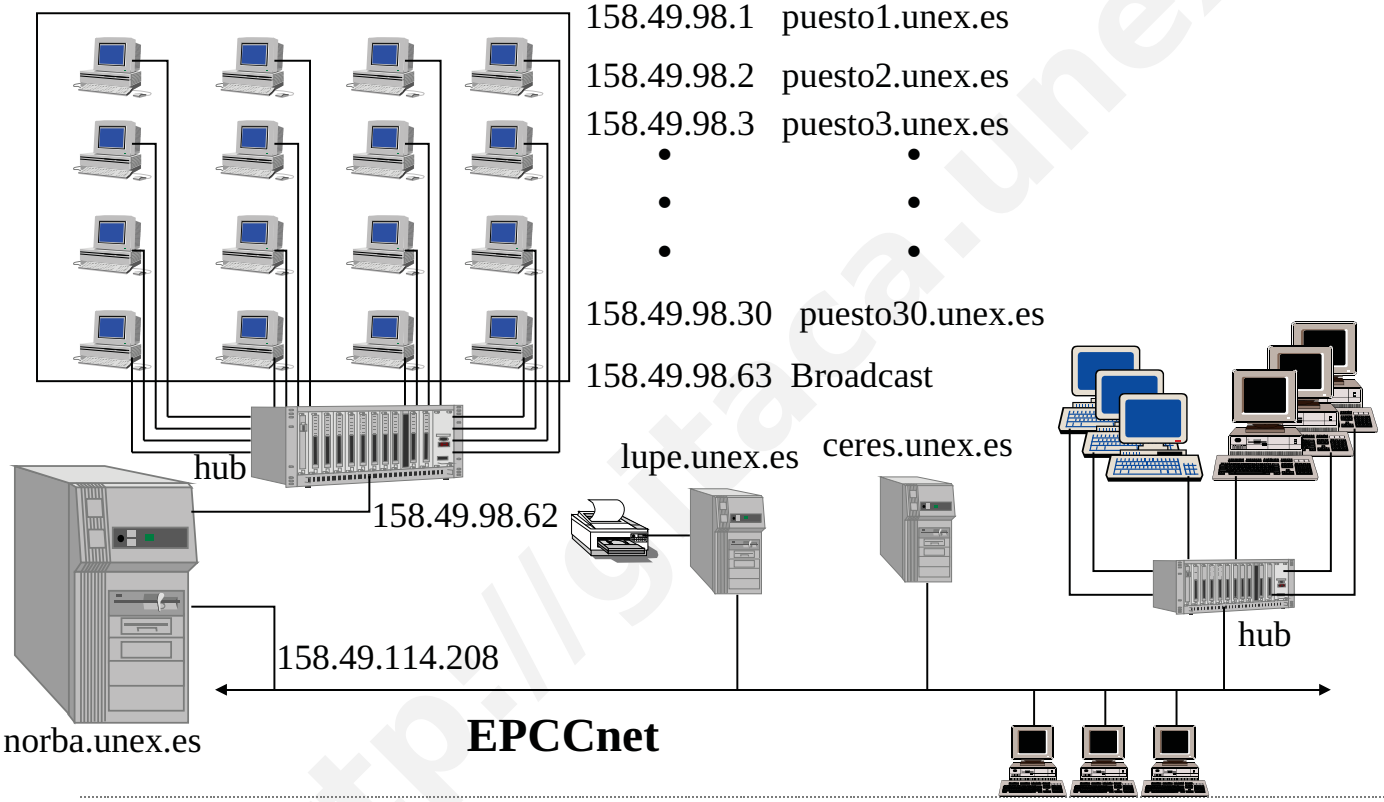
Redes IP Fault Tolerance



Planificación topológica de redes

- Uso actual y evolución futura.
- Deteminar las rutas de intercambio de información.
- Protocolos empleados (TCP/IP, IPX, DECNET, AppleTalk...).
- Gestión centralizada.
- Periferia compartida a traves de la red (discos, impresoras...).
- Sistemas conectados.
- Aplicaciones ejecutadas en red.
- Grupos de trabajo, redes virtuales.

Administración Avanzada de Redes de Sistemas UNIX/Linux



Configuración de red TCP/IP (Ubuntu, Fedora y Solaris)

a) Crear/modificar archivos de configuración de red TCP/IP *(/etc/~)*

b) Uso del comando *ifconfig* para configurar interfaces *Ethernet.*

c) Configuración de rutas *IP* (*route add*).

d) Supervisión y detección de problemas de red con *netstat.*

a) Archivos de configuración de *TCP/IP* en *Linux*:

- Fichero */etc/hosts* Asigna direcciones IP con nombres de sistemas.
- Alias.
- *127.0.0.1 localhost* (Dirección de bucle interno local *loopback*).
- */etc/hosts* y *DNS* Ver el fichero */etc/hosts* de *Linux* y *Norba.*

- */etc/networks* Para dar nombre a las subredes.
- */etc/nsswitch.conf* Para especificar cómo resuelve nombres el sistema.
- */etc/resolv.conf* Para indicar el servidor de nombres DNS del sistema.

b) Inicialización de interfaces de red *ifconfig*

- Informa al kernel de las interfaces de red (*loopback* y tarjetas Ethernet).
- Se usa para supervisar y reconfigurar interfaces de red:

 # ifconfig interfaz [-net -host[dirección opciones]]
#ifconfig lo 127.0.0.1
#ifconfig eth0 puesto1
#ifconfig eth0 puesto1 broadcast 158.49.98.63 netmask 255.255.255.192

c) Routing IP: Determina ruta de acceso de un datagrama desde la fuente hasta el destino a través de la red

#/sbin/route: gestiona la tabla de routing del kernel y define rutas estáticas a otros ordenadores o redes a traves de interfaces que se han configurado y activado con *ifconfig* (cadena de arranque).

#route add 127.0.0.1

#route add puesto1.unex.es

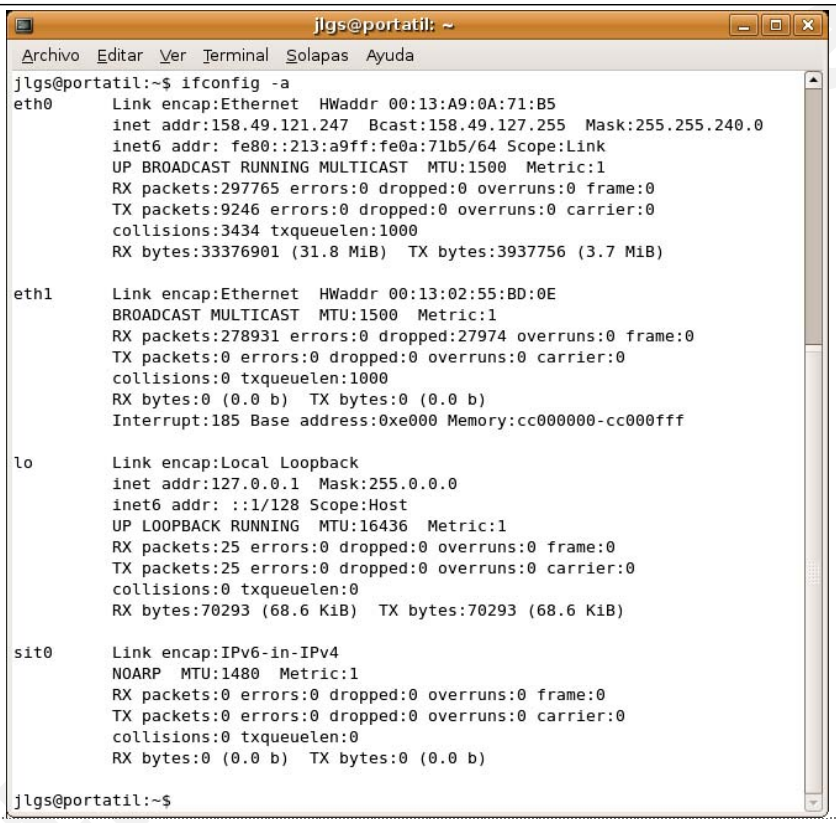
#route add puesto1.unex.es netmask 255.255.255.192

#route add -net 158.49.98.0

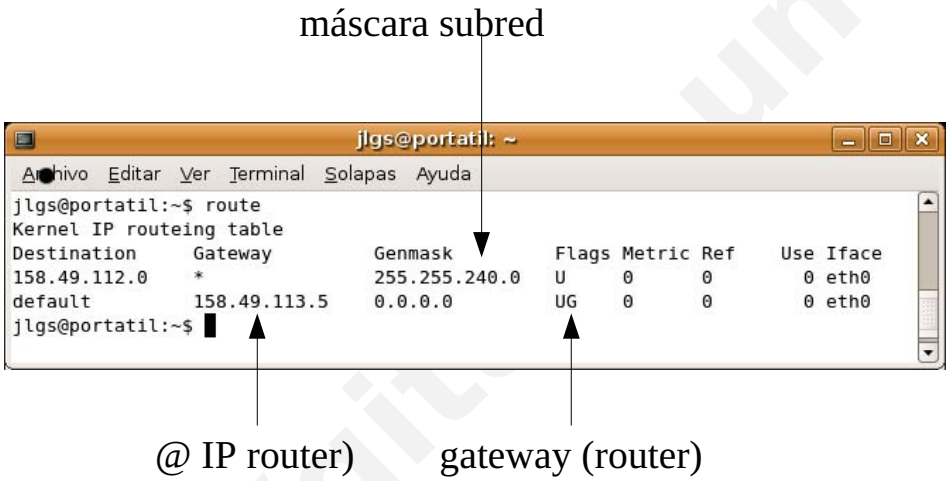
d) Supervisión y detección de problemas de red con *netstat*

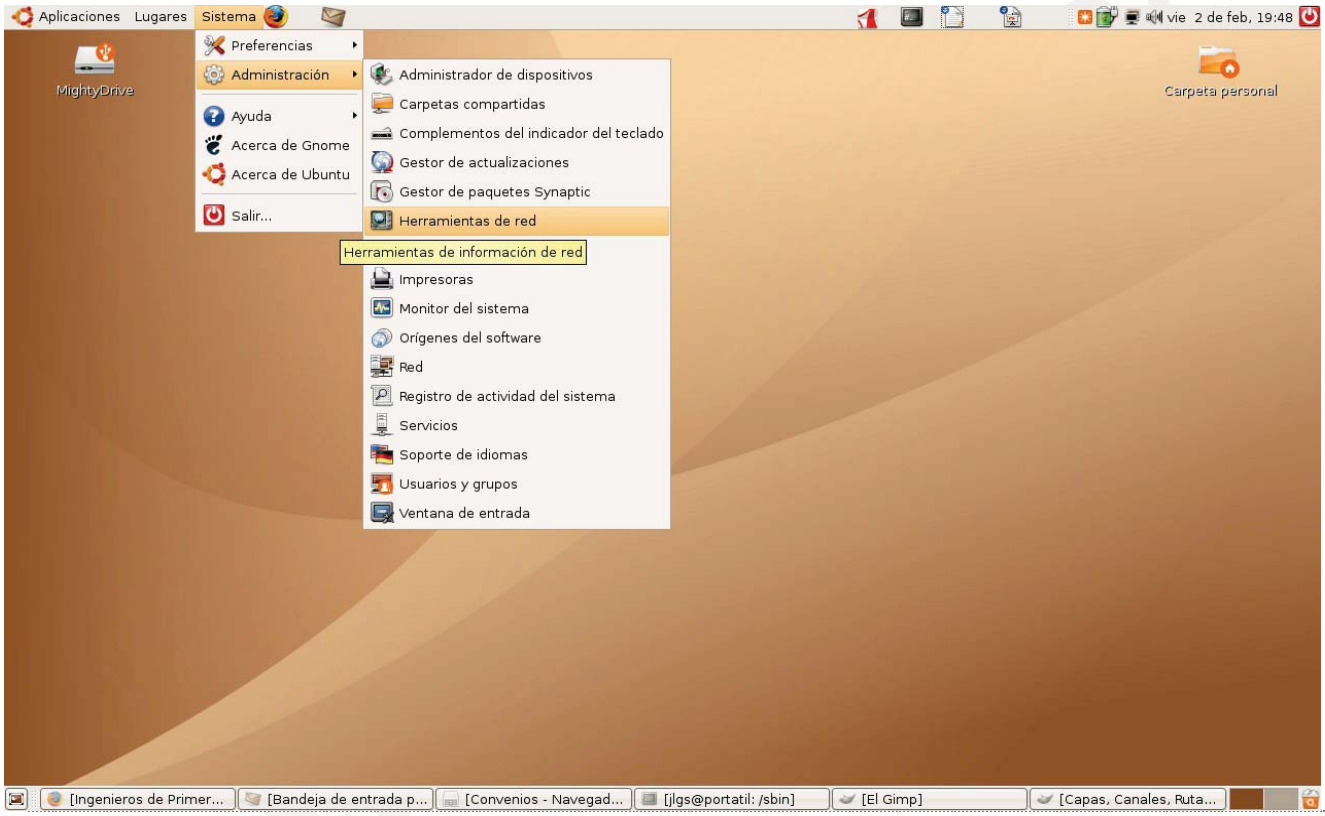
#netstat : Herramienta de supervisión de red. Muestra la tabla de routing; el estado de las conexiones y estadísticas de uso de cada interfaz de red.

Ubuntu:



Ubuntu:

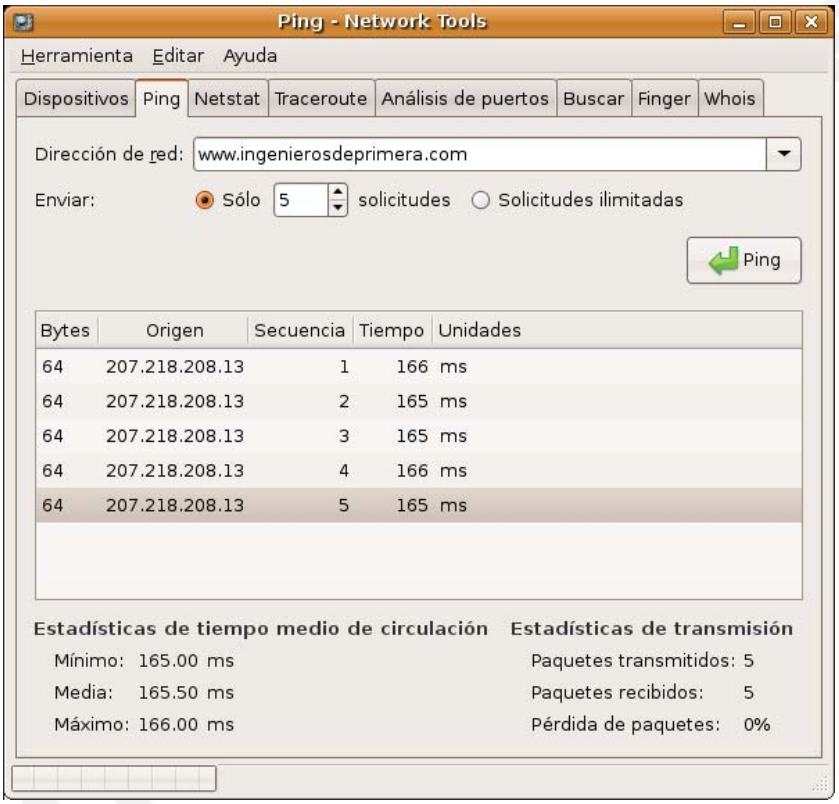

máscara subred
jlgs@portatil: ~
Archivo Editar Ver Terminal Solapas Ayuda
jlgs@portatil:~$ route
Kernel IP routeing table
Destination Gateway Genmask Flags Metric Ref Use Iface
158.49.112.0 * 255.255.240.0 U 0 0 0 eth0
default 158.49.113.5 0.0.0.0 UG 0 0 0 eth0
jlgs@portatil:~$
@ IP router)
gateway (router)


Aplicaciones Lugares Sistema
vie 2 de feb, 19:48
MightyDrive
Carpeta personal
Preferencias
Administración
Ayuda
Acerca de Gnome
Acerca de Ubuntu
Salir...
Administrador de dispositivos
Carpetas compartidas
Complementos del indicador del teclado
Gestor de actualizaciones
Gestor de paquetes Synaptic
Herramientas de red
Herramientas de información de red
Impresoras
Monitor del sistema
Orígenes del software
Red
Registro de actividad del sistema
Servicios
Soporte de idiomas
Usuarios y grupos
Ventana de entrada
[Ingenieros de Primer...
[Bandeja de entrada p...
[Convenios - Navegad...
[jlgs@portatil: /sbin]
[El Gimp]
[Capas, Canales, Ruta...

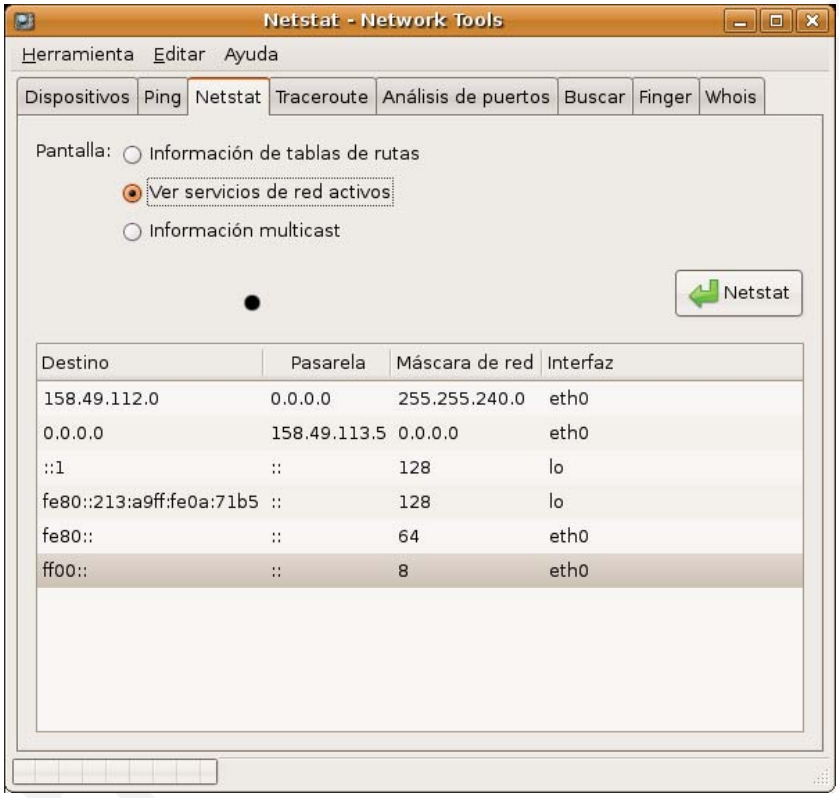
Ubuntu:

Administración Avanzada de Redes de Sistemas UNIX/Linux

Ubuntu:



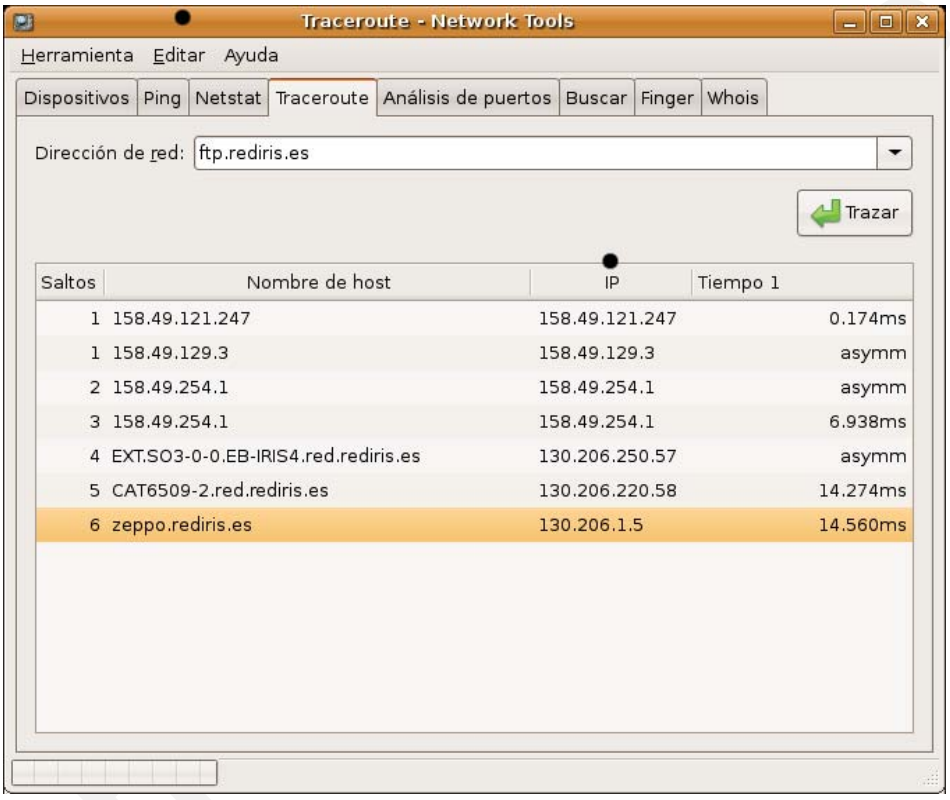
Ubuntu:



Nº 225

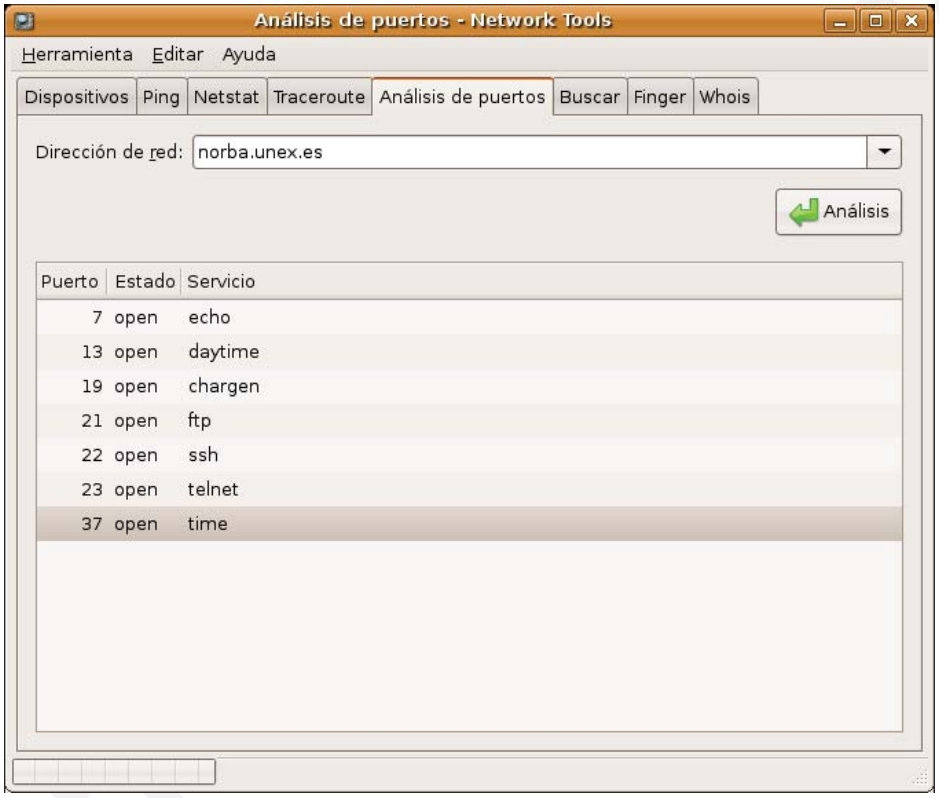
Administración Avanzada de Redes de Sistemas UNIX/Linux

Ubuntu:



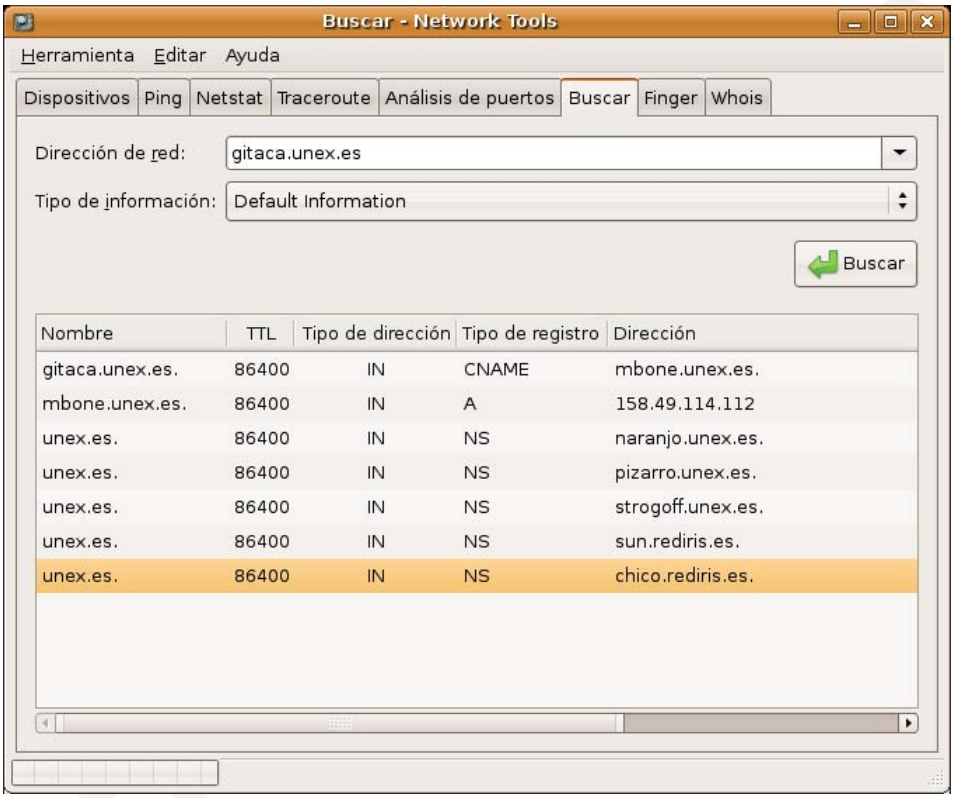
Nº 226

Ubuntu:



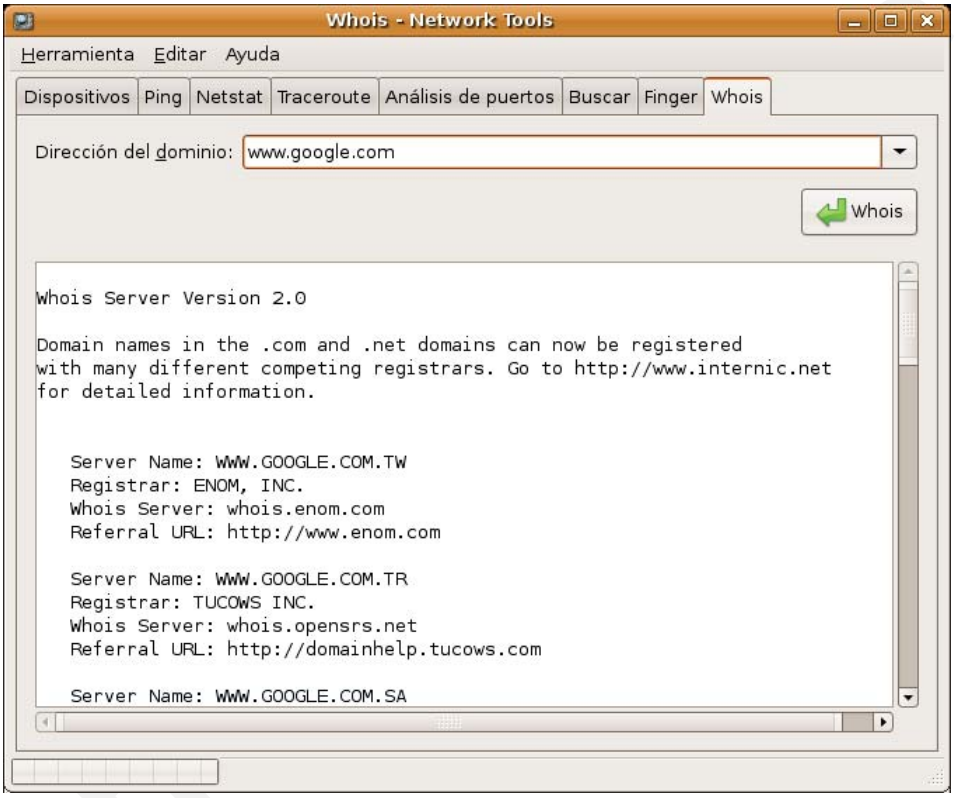
Nº 227

Ubuntu:



Nº 228

Ubuntu:



Nº 229

Administración Avanzada de Redes de Sistemas UNIX/Linux

Ubuntu:

Nº 230

Ubuntu:

Configuración de la red (RedHat Linux-Fedora)

- *ifconfig interfaz @IP* opciones Usado para instalar y configurar la tarjeta de red. Ejecutar *#ifconfig -a*

#ifconfig eth0 158.49.98.12 netmask 255.255.255.192 broadcast 158.49.98.63

- *#ifconfig* (sin parámetros muestra todas las tarjetas configuradas)
  - *lo* es el dispositivo de *loopback*.
  - Muestra: *@ip,MAC,Bcast,Mask*; paquetes transmitidos, errores en paquetes, paquetes descartados (*dropped*) y paquetes demasiado largos (*overruns*); idem. para recibidos.

Configuración de la red (RedHat Linux-Fedora)

• Parámetros opcionales de *ifconfig*:

• *up* activar interfaz para comenzar emisión y recepción.

• *down* desactiva interfaz de red.

• *ARP (Address Resolution Protocol*). Convierte la *@ hard* en IP. El protocolo *DHCP* lo emplea para localizar nodos sin usar *@ IP*. *ARP* no atraviesa *routers* y está activada por defecto (desactiva con -).

• *MTU* N (Maximum Transfer Unit). Por defecto 1500 (tamaño de cada paquete *Ethernet* expresado en número de bytes).

• *#route* : El router es el enlace de la subred con otras redes exteriores.
• *#route* (sin parámetros muestra la tabla de *routing* definida).
• *#route cmd type target_ip netmask gateway options:*
- *cmd=add,del* para añadir o eliminar una ruta
- *type=-net,-host* para ruta hacia red o hacia host
- *target_ip* es la *@IP* de la red o *host* para la que se va a crear la ruta. Si se usa *default* todos los paquetes sin ruta en la tabla irán a esta ruta
- *netmask* permite indicar la máscara de red para la que se crea la ruta
- *gateway* es la pasarela a usar para enviar los paquetes a *target_ip*, y suele ser la @ IP del *router*.

- *options=*

. *n*: usa @ numéricas en lugar nombres de nodo. Se usa cuando se ejecuta *route* sin *add* ni *del* para ver las rutas empleadas en una transm.
. *dev ethx* especifica la placa a la que irá un paquete de una ruta determinada.Usado en sistemas con varias tarjetas de red. Siempre al final.

#route del 158.49.98.13

Ejemplos:

#route add -net default gw 158.49.98.12 paquetes dirigidos al *router*

#route add -net 158.49.98.0 dev eth0 La red 98 se conecta con *eth0*

Tablas de *routing* con *#route*

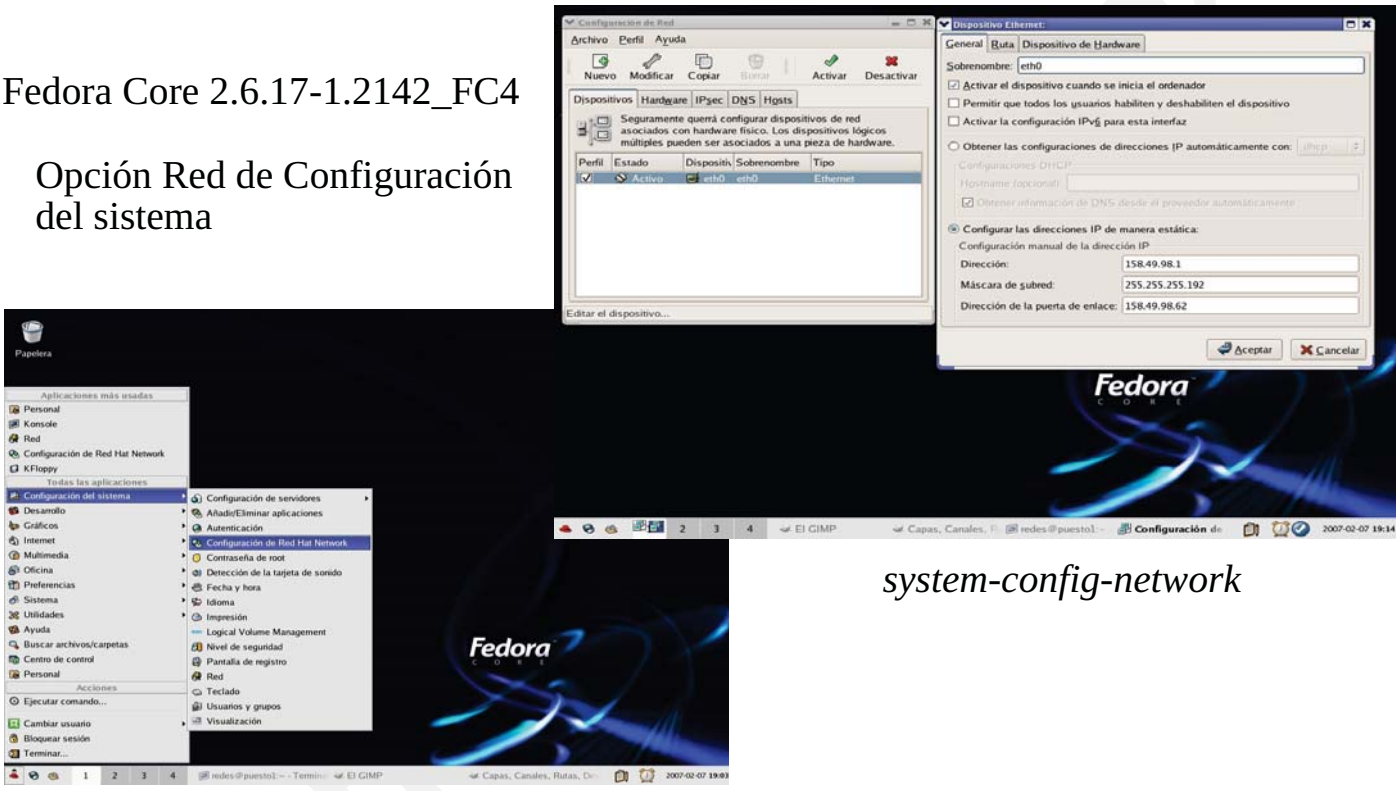
Destination Gateway Genmask Flags Metric Ref Use Iface

- *Destination:* red de destino o @ del nodo.
- *Gateway* : nodo pasarela (*router*) para llegar al destino. Si no hay pasarela definida se muestra *

- *Flags*: describe características de la ruta: U (activa); H (host) G (a través de gateway).
- *Metric*: métrica de cada ruta para asignación dinámica (defecto 0).
- *Ref* número de referencia de la ruta (no usado en *kernel Linux*).
- *Use* expresa el número de veces sistema ha usado la ruta.
- *Iface* interfaz de red que emplea esa ruta.

Fedora Core 2.6.17-1.2142_FC4

Opción Red de Configuración del sistema



system-config-network

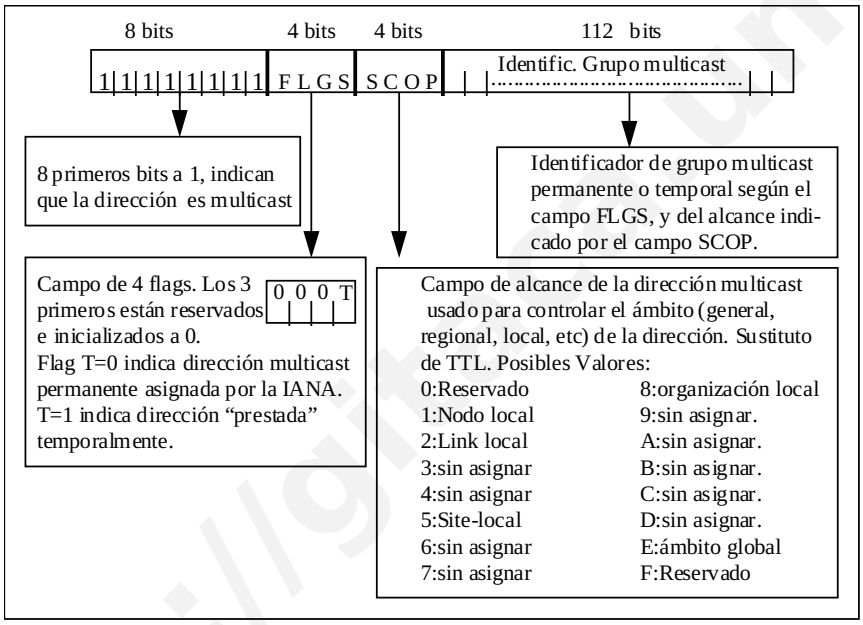
IPv6

- IP muriendo de éxito:
  - Direcciones se agotan.
  - Limitación memoria *routers*.

- Direcciones pasan de 32 a 128 bits: 667 trillones de @ por m^2 si la Tierra tiene 501.101 millones de Km^2. Cada electrón del Universo puede ya tener su nodo de Internet.

- Seguridad.
- *Jumbogramas*.
- QoS.
- Túneles *IPv4*.

- *IPv6*: Red Hat, Ubuntu, Solaris... ya operativo.
- *http://www-6bone.net/*



Direccionamiento IPv6

Conexión por módem

- ¿Conexión por módem para la administración de sistemas?
  - Administradores de sistemas “solos ante el peligro”.
  - Centros de Procesos de Datos (CPD) habitualmente sin conectividad IP hacia fuera.
  - A veces, mientras se administra en un CPD, es necesario consultar documentación existente en la Intranet de nuestra empresa.
  - Modo de conectarnos a nuestra Intranet:
    - Teléfono del CPD + módem del portátil + RAS de nuestra empresa, o
    - convencer al administrador del CPD para que abra acceso al exterior (¡buena suerte!)

Conexión por módem. Introducción.

- PPP (Point-to-Point Protocol) es un protocolo del nivel de enlace utilizado, entre otras cosas, para encapsular tráfico IP punto a punto entre el módem y el servidor de acceso.

7	Aplicación
6	Presentación
5	Sesión
4	Transporte
3	Red
2	Enlace
1	Físico

Arquitectura del Modelo OSI

7	Protocolos de niveles superiores
6	
5	
4	TCP / UDP
3	IP
2	PPP
1	Físico (Conexión serie, módem, RDSI,...)

Arquitectura de PPP

Conexión por módem. Introducción.

- La conexión por módem se establece en varios pasos:
  - Establecimiento de la conexión física (nivel 1)
  - Establecimiento del enlace (nivel 2)
  - Autenticación
  - Obtención de direcciones IP y DNS. (nivel 3)
- Una vez obtenida las direcciones, la conectividad se basa en la pila TCP/IP sobre PPP.
- pppd es el *daemon* encargado de gestionar el proceso de conexión PPP en sistemas UNIX/Linux.

Conexión por módem. Configuración gráfica.

- Todas las distribuciones tienen herramientas gráficas de conexión por módem.
  - *kppp, gdial, gppp, system-config-network, network-admin, etc*.
- Vamos a realizar la configuración gráfica de la conexión por módem en Ubuntu, usando *network-admin*.

Conexión por módem. Configuración gráfica.

- Paso 1: Pantalla inicial de *network-admin*

Conexión por módem. Configuración gráfica.

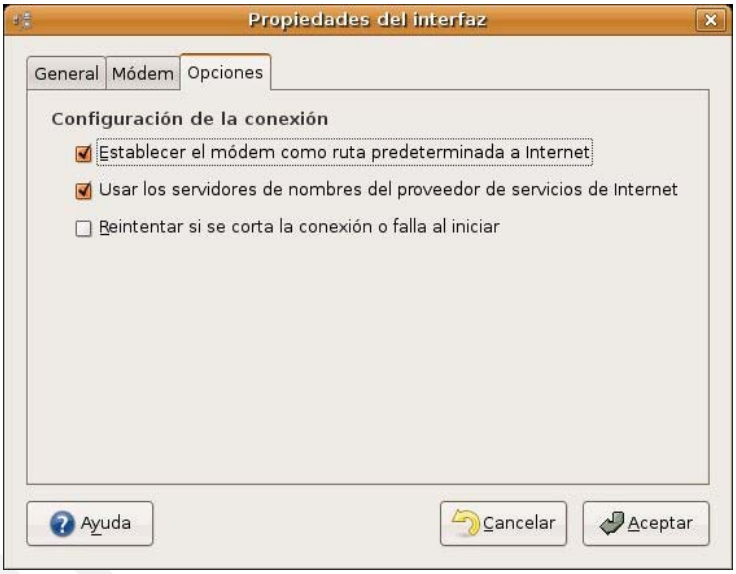
- Paso 2: Número de teléfono y datos de usuario

Conexión por módem. Configuración gráfica.

- Paso 3: Parámetros del módem

Conexión por módem. Configuración gráfica.

- Paso 4: Opciones adicionales



Conexión por módem. Configuración gráfica.

- Paso 5: Conexión.

Conexión por módem. Configuración gráfica.

- Interfaz ppp:

 # ifconfig

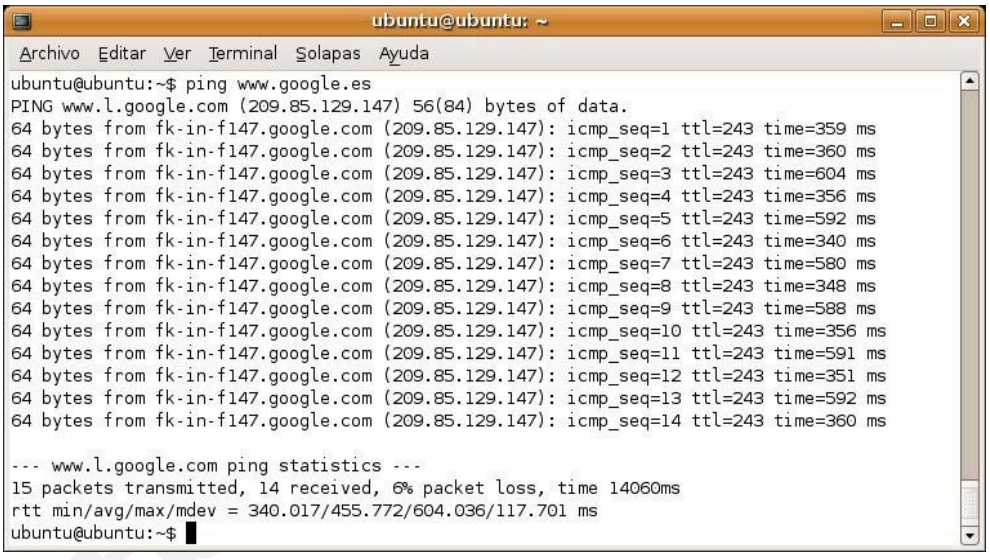
Conexión por módem. Configuración gráfica.

- Interfaz ppp como ruta por defecto:

 # route

Conexión por módem. Configuración gráfica.

- Prueba de *Ping* sobre la conexión ppp



Conexión por módem. Configuración manual.

- ¿Por qué la configuración manual?
  - Generalmente, la administración de servidores UNIX/Linux se realiza por red mediante SSH y sin entorno gráfico.
- Pasos a seguir:
  - Crear el fichero de opciones de ppp.
  - Crear el *script* de la cadena de conexión.
  - Crear el fichero de autenticación CHAP (*Challenge-Handshake Authentication Protocol*)
  - Levantar el *daemon* pppd

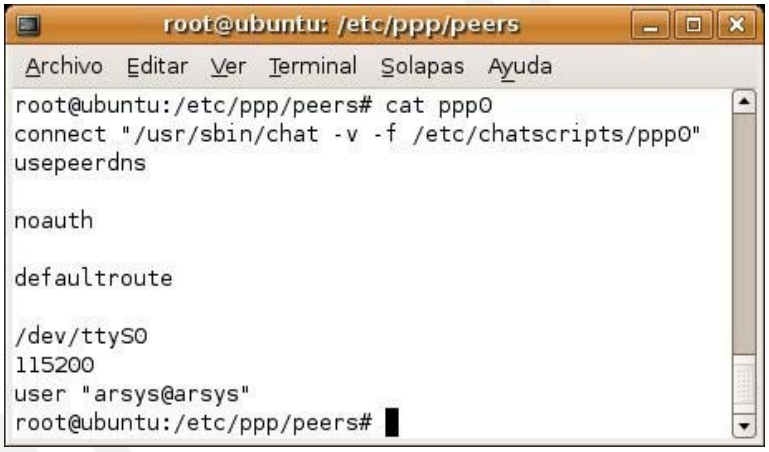
Administración Avanzada de Redes de Sistemas UNIX/Linux

Conexión por módem. Configuración manual.

- Fichero de opciones: En este archivo se encuentran los parámetros que controlan la negociación entre el *daemon* pppd y el RAS remoto.
  - **connect**: Especifica el fichero donde se encuentra la cadena de conexión
  - **usepeerdns**: Solicita al extremo remoto hasta 2 direcciones de servidores DNS.
  - **noauth**: Hace que el servidor remoto no tenga que autenticarse.
  - **defaultroute**: Añade a la tabla de rutas, una entrada para la conexión PPP y establece la ruta por defecto a través del RAS que actúa como *gateway*.

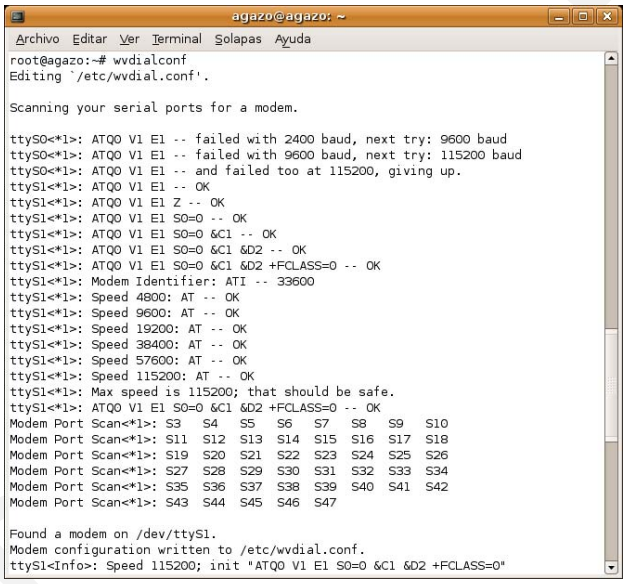
Conexión por módem. Configuración manual.

- ***dispositivo****: Dispositivo asociado al módem*
- ***velocidad_solicitada***: en baudios/seg.
- **user:** nombre de usuario para autenticarse



Conexión por módem. Configuración manual.

- Detectar el dispositivo y velocidad: *wvdialconf*



Conexión por módem. Configuración manual.

- *Script* de la cadena de conexión.
  - Este fichero contiene el diálogo que se tiene que llevar a cabo con el módem local, para realizar la conexión física
  - PPPD gobierna el diálogo, enviando al módem las órdenes que aparecen en el archivo.
  - Cada línea del fichero representa el comando AT que el módem local debe enviar al remoto, y la respuesta que se espera recibir del mismo.
  - La secuencia de líneas describen el diálogo necesario para realizar la conexión.

Conexión por módem. Configuración manual.

- *Script* de la cadena de conexión:

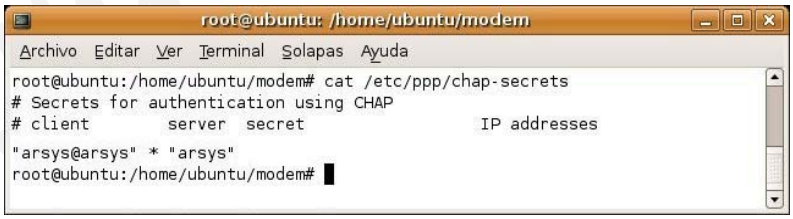
Conexión por módem. Configuración manual.

- Fichero de autenticación CHAP:
  - Almacena los pares usuario-contraseña para la autenticación ante el servidor de acceso remoto.
  - Se debe llamar *chap-secrets*
  - Debe estar en */etc/ppp*
  - Su propietario debe ser *root*
  - Debe tener permisos *rw- --- --- (600)*
  - Formato:

Administración Avanzada de Redes de Sistemas UNIX/Linux

Conexión por módem. Configuración manual.

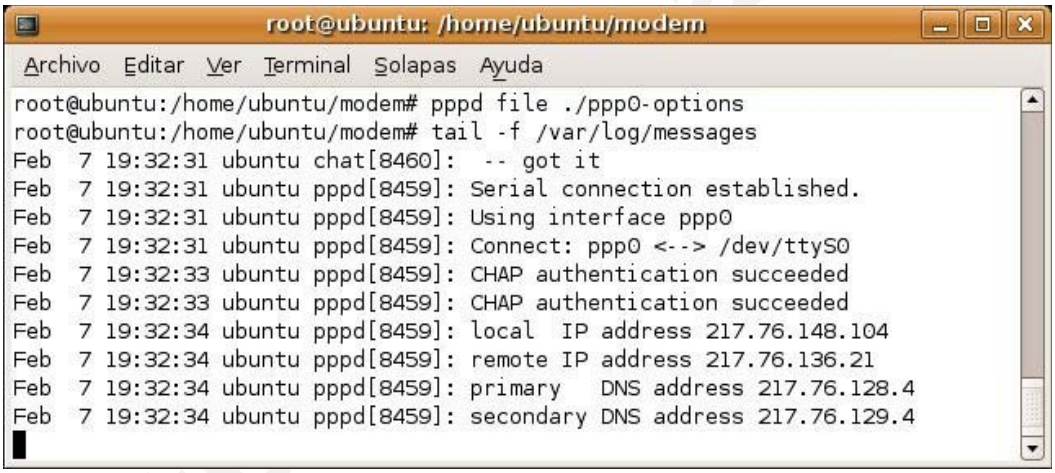
- Fichero de autenticación CHAP:
  - Almacena los pares usuario-contraseña para la autenticación ante el servidor de acceso remoto.
  - Se debe llamar *chap-secrets*
  - Debe estar en */etc/ppp*
  - Su propietario debe ser *root*
  - Debe tener permisos *rw- --- --- (600)*
  - Formato:



Conexión por módem. Configuración manual.

- Conexión:

```
#pppd file fichero_opciones
```



Conexión por módem. Configuración manual.

- Desconexión (cualquiera de estas opciones):
  - `killall pppd`
  - `kill -9 PID_del_proceso_pppd`
  - `/etc/init.d/pppd stop`

Conexión por Wi-Fi. Introducción.

- Wi-Fi no está determinado por el cableado.
  - Hay que indicar a cual de las redes Wi-Fi (si hay más de una) nos vamos a conectar.

- Red identificada principalmente por:
  - Dirección HW del punto de acceso.
  - ESSID *(case sensitive)*
  - Encriptación (ninguna, WEP, WPA)

Conexión por Wi-Fi. Configuración gráfica.

- Wi-Fi ha tenido problemas en Linux.
  - Retraso de implementación de IEEE 802.11en el *kernel*, hace que el modo de conexión no sea homogéneo para todos los interfaces.
  - Drivers
- Existen herramientas gráficas para la mayoría de las distribuciones.
  - *Wireless Assistant, Network-admin, etc.*
- Vamos a realizar la configuración gráfica de la conexión Wi-Fi en Ubuntu, usando *network-admin.*

Conexión por Wi-Fi. Configuración gráfica.

- Paso 1: Pantalla inicial de *Network-Admin*

Conexión por Wi-Fi. Configuración gráfica.

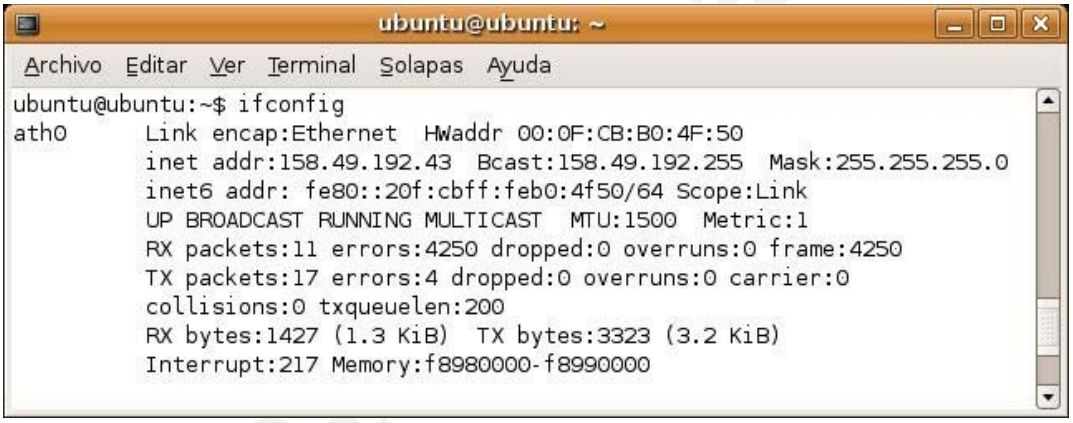
- Paso 2: Selección de la red.

Conexión por Wi-Fi. Configuración gráfica.

- Paso 3: Conexión a la red Wi-Fi

Conexión por Wi-Fi. Configuración gráfica.

- Interfaz inalámbrica



Conexión por Wi-Fi. Configuración manual.

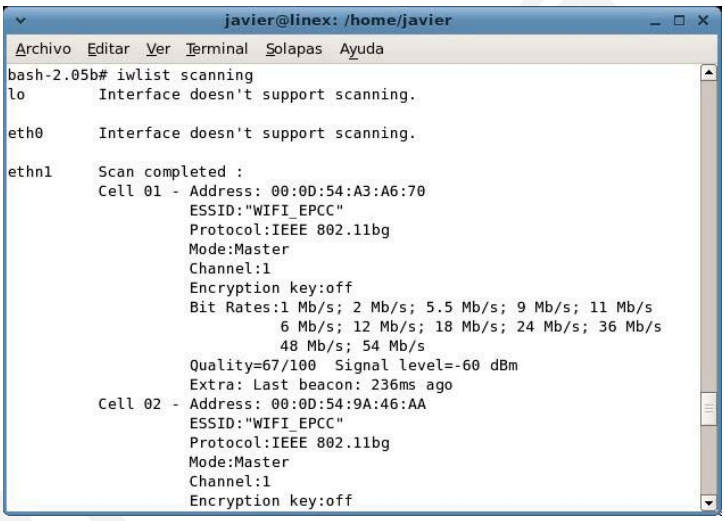
- ¿Por qué la configuración manual?
  - Generalmente, la administración de servidores UNIX/Linux se realiza por red mediante SSH y sin entorno gráfico.
- Pasos a seguir:
  - Detectar las redes inalámbricas.
  - Asociar la interfaz al punto de acceso de la red deseada.
  - Configurar el interfaz (IP, mascara, broadcast), la tabla de rutas, y */etc/resolv.conf* con los valores adecuados o,
  - Solicitar direcciones IP y DNS vía DHCP.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Conexión por Wi-Fi. Configuración manual.

- Paso 1: Detectar las redes inalámbricas.

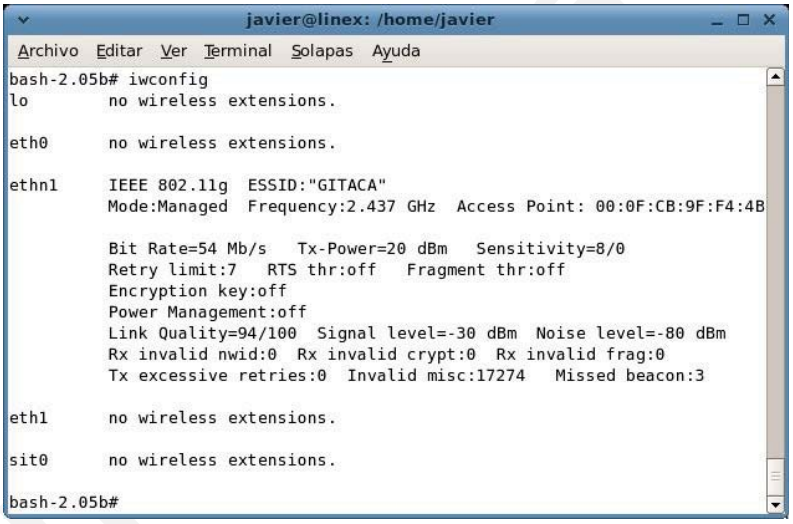
```
#iwlist scanning
```



Conexión por Wi-Fi. Configuración manual.

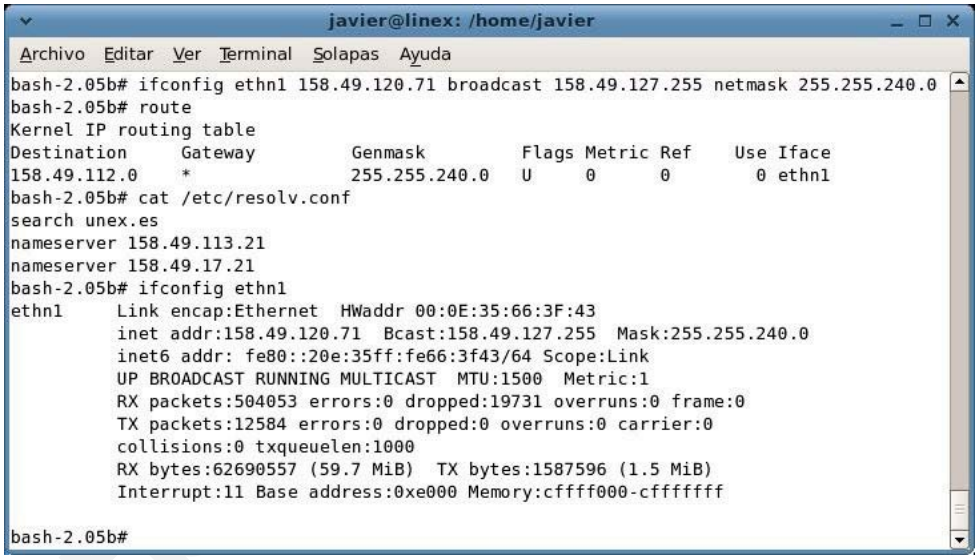
- Paso 2: Asociar interfaz al punto de acceso.

`#iwconfig` *`ifaz`* `essid` *`"essid_red"`*



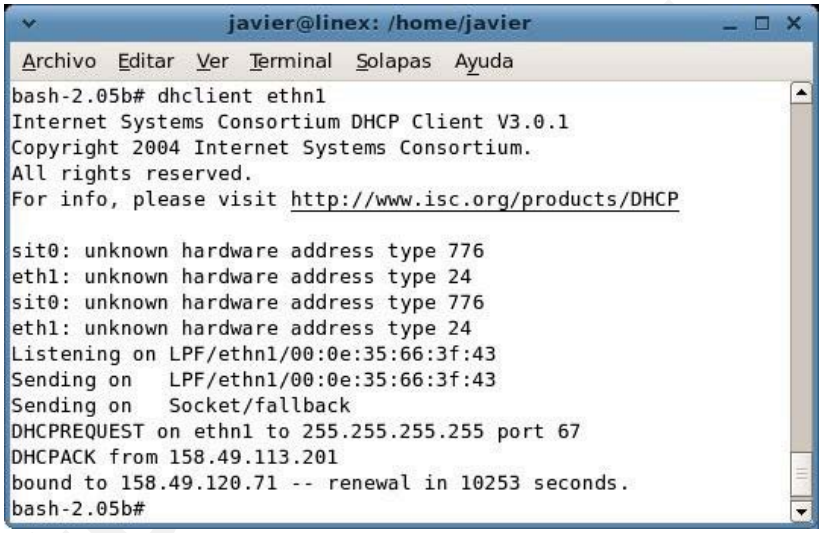
Conexión por Wi-Fi. Configuración manual.

- Paso 3: Configurar el interfaz manualmente

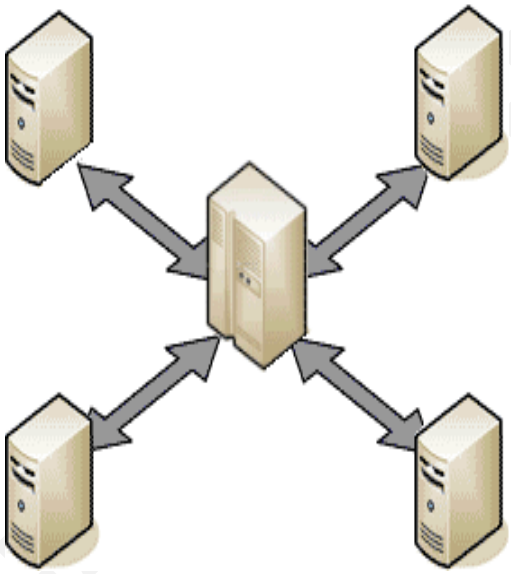


Conexión por Wi-Fi. Configuración manual.

- Paso 3: Configurar el interfaz por DHCP



Servicios de Red



Servicios de Red

- Clientes de Internet
- Acceso a sistemas de archivos remotos
- Servicio NFS
- Servicio DNS
- Servicio HTTP

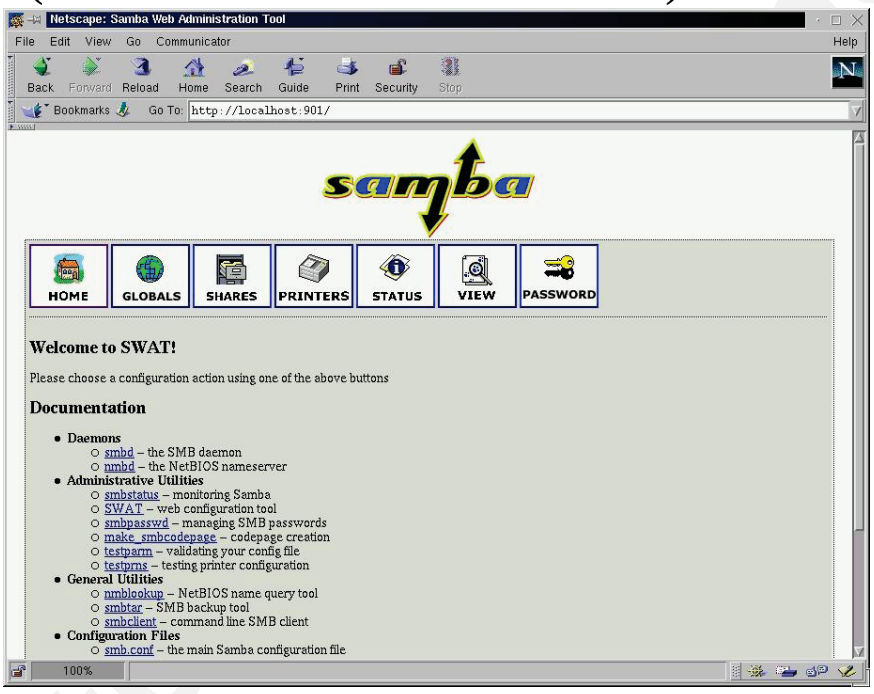
Clientes Internet

- Telnet (*Telecommunications Network*): permite el acceso remoto a los recursos de un equipo multiusuario.FTP (*File Transfer Protocol*): transferencia remota de ficheros entre ordenadores (*ftp, gFTP*).
- Correo electrónico y lectores de noticias: *mail, Elm, Pine, Kmail, Mozilla Mail, Evolution, xrn*).
- Chat y mensajería instantánea: mantener conversaciones escritas entre equipos remotos (*kSirc, X-Chat, Gaim*).
- Navegación Web: puede integrar todos los clientes Internet en un solo programa, incluido acceso a WWW (*World Wide Web*): *Lynx, Mozilla, Konqueror*

Samba

- Colección de programas que implementan SMB (*Server Message Block*). Permite configurar un sistema *Linux* para interactuar en una red *Microsoft* (compartir archivos, directorios e impresoras, autentificar usuarios y resolver nombres WINS).
- Incluye los *daemon smbd* y *nmbd*:
  - *smbd*: compartición de ficheros e impresoras y autenticación de clientes.
  - *nmbd*: responsable de la resolución de nombres WINS.
- Samba dispone de muchos parámetros de configuración. Su gestión se simplifica con SWAT a través de web.

- SWAT (*Samba Web Administration Tool*):



NFS (Network File System)

- Permite a varios sistemas *Unix* compartir entre sí sistemas de ficheros. El *server FTP* exporta sistemas de ficheros a los que acceden los clientes como locales.
- *NFS* se basa en los protocolos *RPC* (*Remote Procedure Calls*) pensado para comunicar dos nodos entre sí; *TCP/IP* y *eXternal Data Representation* (*XDR*).
- Instalación de *NFS* en *RedHat*, basado en los programas:
  - *rpc.portmapper*: Centraliza las peticiones de acceso desde otros ordenadores delegando *NFS* a los *daemons NFS*.
  - *rpc.nfsd*: *daemon* encargado de atender las peticiones *NFS* en los servidores *NFS*. Suelen activarse 4 de este tipo.

- *rpc.mountd: daemon* encargado en los servidores *NFS* de montar y desmontar sistemas de ficheros. Suelen activarse 4 *daemons*.

#rpcinfo -p : muestra los programas *RPC* instalados en local.
#rpcinfo -p nodo : muestra los programas instalados remotos.
#/etc/rc.d/init.d/nfs start : Arranca el servidor *NFS*.
#/etc/rc.d/init.d/nfs stop : Detiene el servidor *NFS*.

- Configuración de *NFS (Linux y SunOS)*:

a) */etc/exports*. Fichero que contiene los dirs. a compartir y los derechos de acceso de cada usuario. Se configura en el servidor. Los permisos de acceso son: *rw* (lectura y escritura), *ro* (sólo lectura) y *no_root-squash* (reconocer y confiar en *root* desde clientes)

#Ejemplo /etc/exports
/export/home lupe (rw) puesto2(ro) puesto5(rw)
#Exporta dirs de software
/export/usr/local lupe (rw,no_root_squash)
puesto(ro,no_root_squash)

- Una vez configurado exports en el servidor, se debe avisar a sus *daemons rpc.nfsd* y *rpc.mountd* con el comando
  #exportfs -a
- Se pueden comprobar los sistemas de ficheros exportados, mediante el comando *#exportfs*, sin parámetros.

b) Montar los sistemas de ficheros a exportar:

#mount norba:/export/home /home

/home debe existir en el ordenador local antes de montar en el sistema de ficheros exportado. A *mount* pueden pasárselo con -o los parámetros *rw* (lectura y escritura), *ro* (sólo lectura), *bg* (*background*, si el montado falla, reintenta hasta conseguirlo), *intr* (montaje interrumpible si se están esperando otras operaciones), *soft* (Ops. *NFS* son hard por defecto, *soft* permite que los clientes *NFS* devuelvan un error al servidor tras *retrans* intentos), *retrans* (Nº de intentos de transmisión a un sistema de archivos que se ha montado con opción *soft*).

#mount -o rw,bg,intr,soft,retrans=4 norba:/export/home /home
#umount /home

Para poder desmontar es necesario que el sist. fich. no se esté usando y todos los archivos deben estar cerrados. Esto se garantiza con:

1.- umount -f (obliga al sist. fich. a desmontarse. Peligroso)

2.- usar programa *lsoft* para saber qué usuarios están usando el sist. fich. y los ficheros abiertos. Esperar a que no quede nadie.

3.- Poner el sistema en *single user* y desmontar el sist. fich. Es el método más lento y trabajoso, pero el más seguro para los usuarios.

c) Archivo */etc/fstab*. Se configura en los clientes para indicar la lista con todas las particiones que hay que montar durante el arranque y los directorios en los que hay que montarlas.


```
#Ejemplo de /etc/fstab
/dev/hda1                /          ext2     rw          0  0
/dev/hda2                swap       swap
norba:/export/home       /homenfs     rw,bg,intr,soft    0  0
```

- Dispositivo a montar
- Punto de montaje del sistema de ficheros.
- Tipo de sistema de ficheros. Para los locales es *ext2*
- Parámetros que se pasan al comando *mount*
- Parámetro usado por *dump* para saber si hay que hacer copia de un sistema de ficheros.
- Usado por *fsck* para saber el orden en que hay que chequear discos.

- En *fstab* se indican todos los sistemas de ficheros que se deseen montar automáticamente al botar. Los sist. ficheros nuevos cuando el sistema está botado, se montan a mano.
- La partición *swap* no se monta con *mount* sino con *swapon -a* partición.
- Un servidor de *NFS* puede ser a su vez cliente de otro servidor *NFS (a*tención a los montajes cruzados que provocan problemas de *boot)*.

  #nfstat (*Solaris*)

  -*c* (muestra información relativa al cliente *NFS*).
  -*m (*presenta estadísticas para cada sistema de ficheros).
  -*s* (muestra información relativa al servidor *NFS*).
  -*z* (reinicia estadísticas).

Domain Name Service (DNS)

- Inicialmente fichero *hosts.txt* copiado por *FTP*. Ineficiencia
- *DNS* es un sistema distribuido de información sobre la Red.
- Conceptos: Servicio de nombres, espacio de nombres, dominio, servidor de nombres, “resolver” de nombres.
- Servidores Maestros o servidores de nombres autorizados mantienen información relativa a la zona.

  •Serv. primario: Almacena en disco la información de la zona que es la que se gestiona y delega a otros servidores de la zona.

  •Serv. secundario: Mantiene una copia de la zona que es periódicamente actualizada desde el principal.
- Un servidor puede ser principal para unas zonas y secundario para otras.

Domain Name Service (DNS)

- *Caching*: La caché del servidor recibe información y la almacena según el alcance de *TTL (TimeToLive*). Por esto, todos los servidores de nombres lo son también de *caching*.
- *in.named: es el daemon ejecutado por todos los servidores de DNS.*También se le llama *BIND* (*Berkeley Internet Name Domain*). El *daemon* sólo se arranca en el *boot* si existe y ejecuta el fichero *in.named*
- */etc/named.conf: fichero de arranque que define si el servidor es principal* secundario o *caché*. Además especifica las zonas sobre las que el servidor tiene autoridad y cuáles son los ficheros de datos. Se lee cuando se lanza *daemon in.named* desde */etc/init.d/inetsvc*.

/etc/bind/named.conf (Ubuntu)

```
// If you are just adding zones, please do that in /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
// be authoritative for the localhost forward and reverse zones
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
```

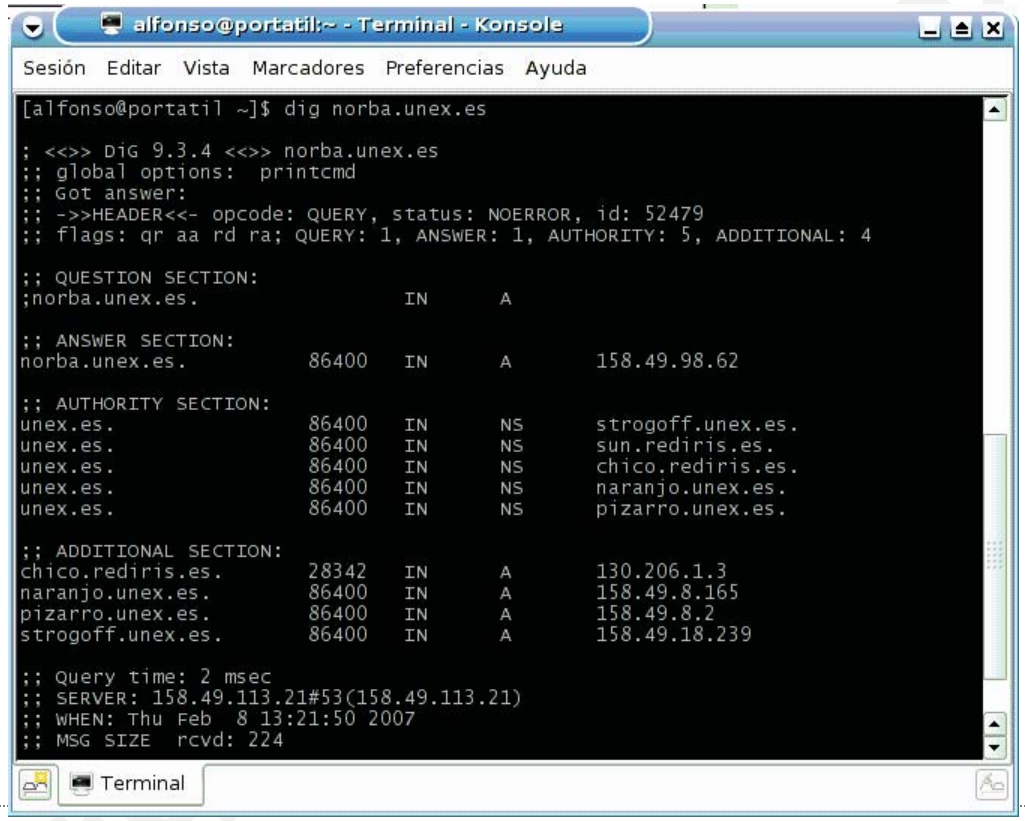
Clientes DNS

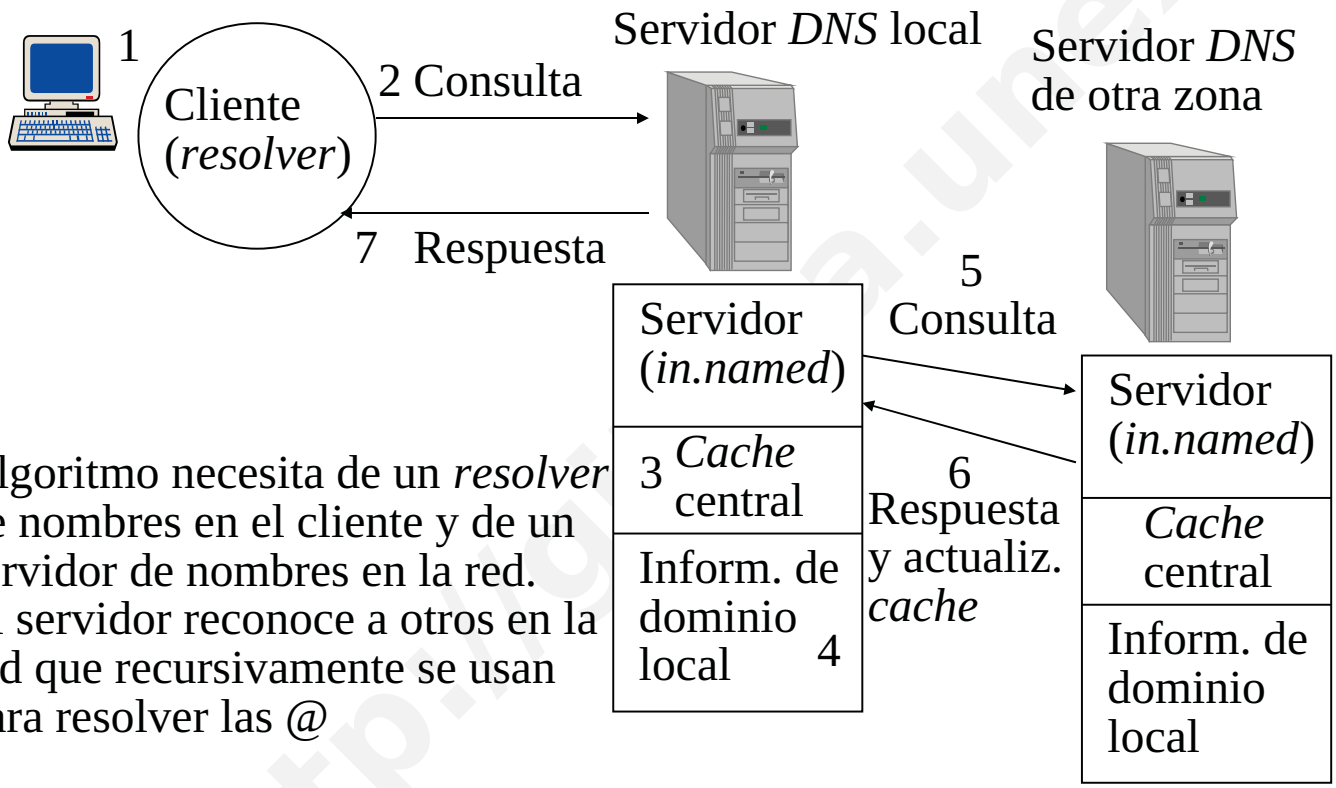
- Ficheros de configuración de los clientes de *DNS*:

  /etc/resolv.conf: especifica servidores de DNS y dominios de búsqueda.

  /etc/nsswitch.conf: especifica el orden en el que ciertos servicios (no sólo de nombres de hosts) van a resolverse.

- Clientes:
  - nslookup (obsoleto)
  - dig





Algoritmo necesita de un *resolver* de nombres en el cliente y de un servidor de nombres en la red.
El servidor reconoce a otros en la red que recursivamente se usan para resolver las @

• Arranque servidor *DNS*:
#in.named
Modificar cadena de arranque si es necesario
#nslookup - 158.49.113.21 Comprobar el servidor de nombres.

• Configuración clientes *DNS* (necesitan acceso a la BD de *DNS*):
- */etc/resolv.conf*: usado por el resolver para saber el nombre de dominio y para localizar la @IP del servidor de nombres. Crearlo con *vi*
- */etc/nsswitch*.conf editarlo con vi para configurar el sistema para que use el programa resolver de *DNS*. Editar la línea:
  hosts: dns files
- Comprobar que el cliente funciona con:
  #nslookup
  >ls .com ó >maquina ó >help
  #ping maquina

Hyper Text Transfer Protocol (HTTP)

- Cada servidor web dispone de al menos un proceso que escucha en el puerto TCP 80 (*well-know*), esperando peticiones por parte de clientes.
- El protocolo que define las solicitudes y las respuestas *legales* se denomina HTTP.
- Podemos considerar una petición HTTP originada desde el URL: http://www.w3.org/hypertext/WWW/TheProject.html

–El navegador determina el URL.

–El navegador solicita la dirección IP de www.w3.org al servidor de DNS.

–El servidor de DNS contesta con 18.23.0.23.

–El navegador establece una conexión TCP sobre el puerto 80 con 18.23.0.23.

–A continuación, el navegador emite un mensaje GET /hypertext/WWW/TheProject.html

–El servidor www.w3.org envía el archivo TheProject.html

–Se cierra la conexión TCP.

–El navegador presenta todo el texto de TheProject.html

–El navegador trae y presenta todas las imágenes de TheProject.html

Administración Avanzada de Redes de Sistemas UNIX/Linux

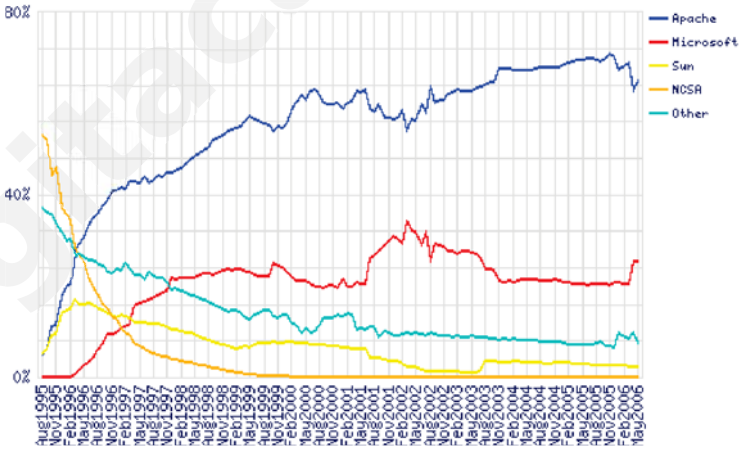
Hyper Text Transfer Protocol (HTTP)

- Pero HTTP no es sólo un protocolo para transferencia de archivos.
- A través de HTTP se pueden utilizar los siguientes métodos:
  - GET: solicita leer un recurso.
  - HEAD: solicita leer la cabecera de un recurso.
  - PUT: solicita almacenar un archivo.
  - POST: añade datos a un recurso.
  - DELETE: elimina un archivo.
  - LINK: conecta dos recursos existentes.
  - UNLINK: rompe una conexión existente entre dos recursos

"100": Continue
"101": Switching Protocols
"200": OK
"201": Created
"202": Accepted
"203": Non-Authoritative Information
"204": No Content
"205": Reset Content
"206": Partial Content
"300": Multiple Choices
"301": Moved Permanently
"302": Found
"303": See Other
"304": Not Modified
"305": Use Proxy
"307": Temporary Redirect
"400": Bad Request
"401": Unauthorized
"402": Payment Required
"403": Forbidden
"404": Not Found
"405": Method Not Allowed
"406": Not Acceptable
"407": Proxy Authentication Required
"408": Request Time-out
"409": Conflict
"410": Gone
"411": Length Required
"412": Precondition Failed
"413": Request Entity Too Large
"414": Request-URI Too Large
"415": Unsupported Media Type
"416": Requested range not satisfiable
"417": Expectation Failed
"500": Internal Server Error
"501": Not Implemented
"502": Bad Gateway
"503": Service Unavailable
"504": Gateway Time-out
"505": HTTP Version not supported

Servicio HTTP en UNIX/Linux

- Diverso software de servicio HTTP sobre UNIX/Linux
- Mayor competencia en el pasado.
- Actualmente, Apache se considera **EL** servidor HTTP sobre UNIX/Linux.
- RedHat llama a este servicio `httpd` (HTTP daemon)



Apache HTTP Server

- Uno de los proyectos de la Apache Software Foundation (+80 en total)
- Parte en 1.994, del código de NCSA httpd
- Un año después aparece Apache HTTP Server 1.0
- Contribuciones de desarrolladores y webmasters. Amplísima base de usuarios
- Actualmente se estima una cuota de mercado de 64,76% (junio de 2006)

Apache en Solaris

- Incluído en distribución estándar
- Apache 1.x / Apache 2.x
- Configuración en:
  - /etc/apache - /etc/apache2
    - access.conf
    - httpd.conf
    - magic
    - mime.types
    - srm.conf

Apache en Ubuntu

- Apache 1.x: /etc/apache
  - access.conf
  - httpd.conf
  - mime.types (symlink)
  - modules.conf
  - srm.conf
  - **conf.d**
- Apache 2.x: /etc/apache2
  - apache2.conf
  - httpd.conf
  - magic
  - ports.conf
  - **conf.d**
  - **mods-available**
  - **mods-enabled**
  - **sites-available**
  - **sites-enabled**

mod_deflate

- Compresión de archivos en tránsito.
- Si el navegador lo soporta.
- Ver cabeceras HTTP y cuerpo.
- No activo por defecto (/etc/apache2/mod_deflate)
- Activar módulos en Ubuntu/Debian: **`a2enmod`**
- Distinto en otras distribuciones Linux.

Seguridad

Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad

- Política de seguridad.
- Seguridad de la Información y Comunicaciones.
- Seguridad en sistemas UNIX/Linux.
- ssh (Secure Shell).
- GnuPG (Gnu Pretty Good).

Política de Seguridad

• Pilares de la seguridad informática:
  • Integridad de la información propia y personal.
  • Confidencialidad para ojos no autorizados.
  • Disponibilidad total de acceso a la información propia.

• Política de la segurida debe identificar:
  • Riesgos potenciales.
  • Forma de adelantarse a su aparición.
  • Tareas a realizar cuando aparecen los incidentes.

• Reglas, mecanismos y servicios para garantizar nivel de seguridad adecuado.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Política de Seguridad

• Debe proteger:
  1. La información valiosa contenida en los S. I. que:
     - Si es accedida indebidamente pierde la *confidencialidad*.
     - Si es alterada sin privilegios se incumple la *integridad*.
     - Si no está al acceso de sus propietarios no garantía de *disponibilidad*.
  2. Recursos de los S. I. :
     - Destrucción parcial o total.
     - Monopolización (CPU, discos, memoria...).
     - Accesos no autorizados.
     - Sabotaje.
  3. Reputación:
     - Personal; grupo; depto.; área funcional; organización.

Política de Seguridad

• Debe concretar:
1. ¿Por qué se establece?.
2. ¿Cuánto cuesta? y ¿Cuánto vale?.
3. ¿Quién es el responsable, cuál es su autoridad y presupuesto?.

• Una PSI es un conjunto de requisitos definidos por los responsables de sistemas que indica, en términos generales, qué está y qué no está permitido en el área de seguridad durante la operación del sistema.

• RFC 1244 define la PSI como una declaración de intenciones de alto nivel que cubre la seguridad de los sistemas informáticos y que proporciona las bases para definir y delimitar las responsabilidades para las diversas actuaciones técnicas y organizativas.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Política de Seguridad

• Se refleja en una serie de normas, reglamentos y protocolos a seguir, donde se definen las medidas a tomar para proteger la seguridad del sistema, pero, ante todo, una PSI es una forma de comunicarse con los usuarios.

• La seguridad está relacionada con personas y debe:
  • Ser holística (cubrir todos los aspectos relacionados con la misma). Proteger puertas y ventanas.
  • Adecuada a las necesidades y recursos (caja fuerte para los bolígrafos).
  • Atemporal: el tiempo no debe influir en su eficacia y eficiencia.
  • Definir estrategias y criterios generales.

Seguridad de la Información y Comunicaciones

- Debilidades seguridad:
  - Sistemas operativos.
  - Protocolos redes y comunicaciones.
  - y usuarios confiados.

- Suma atención, cuidado y seguimiento diario.

- Soluciones:
  - Sistemas operativos reforzados (nivel C2).
  - Comunicaciones seguras (IPv6, cifrado, firewalls...).
  - Usuarios concienciados (formación, inversión).

Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad de la Información y Comunicaciones

Seguridad informática: protege ordenador y lo relacionado con él (acceso físico, cableados, periferia, etc.). ***Seguridad de la información***.

- Intenciones de los accesos no autorizados:
  - Obtención de información valiosa.
  - Destruir información.
  - Pura curiosidad.
  - Demostrar que se ha conseguido franquear las barreras.
- Otros peligros menos espectaculares pero más habituales:
  - Errores humanos.
  - Inexistencia de ***política de seguridad***.
  - Contraseñas compartidas.
  - Desconocimiento....

Seguridad de la Información y Comunicaciones

• Amenaza: en un entorno informático es cualquier elemento que compromete el sistema.

• Las amenazas pueden analizarse: antes, durante o después del ataque. Estos mecanismos conforman políticas que garantizan la seguridad del sistema informático:

• La prevención (antes): como mecanismo que aumenta la seguridad (o fiabilidad) de un sistema durante su funcionamiento normal. Ej. Cifrado de la info antes de transmitirla.

• La detección (durante): Mecanismo orientado a revelar violaciones a la seguridad. Suelen ser programas de auditoría.

Seguridad de la Información y Comunicaciones

• La recuperación (después) mecanismos que se aplican, cuando la violación del sistema ya se ha detectado para restaurarlo a su funcionamiento normal (recuperación desde copias de seguridad).

• Preguntas que debe resolver un Admin. sistemas ante un problema de seguridad:

• ¿Cuánto tardará la amenaza en superar la “solución planteada”?
• ¿Cómo se detecta e identifica a tiempo la amenaza?
• ¿Cómo se neutraliza?

Seguridad de la Información y Comunicaciones

• Riesgo: la proximidad o posibilidad de daño sobre un bien. Actos naturales, errores u omisiones humanas y actos intencionados. Cada riesgo debe atacarse:

  • Minimizando la posibilidad de su ocurrencia.
  • Reduciendo al mínimo el perjuicio producido, si no ha podido evitarse que ocurriera.
  • Diseño de métodos para la más rápida recuperación de los daños experimentados
  • Corrección de las medidas de seguridad en función de la experiencia recogida.

Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad de la Información y Comunicaciones

• Para garantizar que un sistema sea fiable se debe conocer:
  - Qué se quiere proteger.
  - De quién se quiere proteger.
  - Cómo se puede lograr técnica y legalmente.
  - Formular estrategias de seguridad para disminuir o anular riesgos.

• Conocer y comprender la seguridad ayuda a llevar a cabo análisis sobre:
  - Los riesgos, vulnerabilidades, amenazas y contramedidas.
  - Evaluar las ventajas o desventajas de la situación.
  - Decidir medidas técnicas y tácticas metodológicas, físicas e informáticas en base a las necesidades de seguridad.

Seguridad de la Información y Comunicaciones

• De quién hay que protegerse:Intruso o atacante es cualquier persona que accede (o lo intenta) sin autorización a un sistema ajeno, sea de forma intencionada o no. Tipos de intrusos:

- Clase A: el 80%. Son la base, los nuevos intrusos que bajan programas de Internet y “juegan” para probar.
- Clase B: 12 % y los más peligrosos. Saben compilar programas pero no programar. Prueban programas, testean vulnerabilidades y acceden por ellas.
- Clase C: el 5% que sabe, conoce y define sus objetivos. Buscan todos los accesos remotos e intentar acceder.
- Clase D: 3 % restante que cuando entran a determinados sistemas saben lo que buscan.

Seguridad de la Información y Comunicaciones

• Qué hay que proteger:

- El hardware.
- El software (SO, aplicaciones y utilidades)
- Los datos o conjunto de informaciones lógicas que maneja el software y el hardware: archivos, documentos, bases de datos, etc . Son los más importantes.
- Fungibles: toner, cintas magnéticas, papel, discos, CDs, etc.

• Importante entender que no existe el 100% de seguridad esperado o deseable.

Seguridad en sistemas UNIX/Linux

- */etc/passwd* vigilar permisos y activar *shadowing*.

- *suid* y *gid* (ejecución de programas como si se fuese su propietario): */usr/bin/passwd* tiene activado *suid* pues */etc/passwd* es legible por todos, pero escribible sólo por *root*. *gid*, lo mismo, pero para grupos:

#chmod u+s archivo ó *#chmod 2xxx fichero*
#chown g+s archivo ó *#chmod 4xxx fichero*
-rwsrws--x 1 usu grupo 30 Feb 12:34:45 fichero

#find / -perm -200 -o -perm -400 -print

Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad en sistemas UNIX/Linux

- Contraseña de longitud mínima de 5 caracteres, por defecto.

- Editar archivo */etc/login.defs* y modificar, como poco,:
  PASS_MIN_LEN 8
  PASS_MAX_DAYS 60
  FAIL_DELAY 5

- */etc/pam.d* (Pluggable Authentication Modules for Linux): sistema de librerías que manipula las tareas de autentificación de aplicaciones o servicios en el sistema.

- */etc/profile* : *TMOUT=3000 (")*

Seguridad en sistemas UNIX/Linux

- Atentos a mensaje de *lastlogin*.
- Cuenta root sólo en consola de servidores: */etc/securetty*.
- Impedir comando *$su root* a usuarios concretos (en */etc/pam.d*).
- Si la seguridad es un requerimiento atención a NFS (evitar el acceso de escritura).
- Atención a *xinetd* para arrancar sólo los daemons necesarios.
- Configurar (firewall) tcp_wrappers con */etc/hosts.allow* y */etc/hosts.deny*
- Privilegios de */etc/init.d/* 700* para que sólo root tenga acceso.
- */etc/issue* y */etc/issue.net* aportan demasiada info.
- Editar archivo */etc/sysctl.conf*:
  - *net.ipv4.icmp_echo_ignore_all = 1*
  - *net.ipv4.tcp_syncookies = 1*
  - *net.ipv4.icmp_echo_ignore_broadcast = 1*

Seguridad en sistemas UNIX/Linux

- *net.ipv4.conf.all.rp_filter = 1* . Activa protección contra IP spoofing
- *net.ipv4.conf.all.log_martians = 1*. Log de paquetes spoofed, redir.
- Encontrar archivos *.rhosts* en el sistema: *$find / -name .rhosts*
- Archivos en */etc/security* : permiten, limitan o controlan determinados accesos.

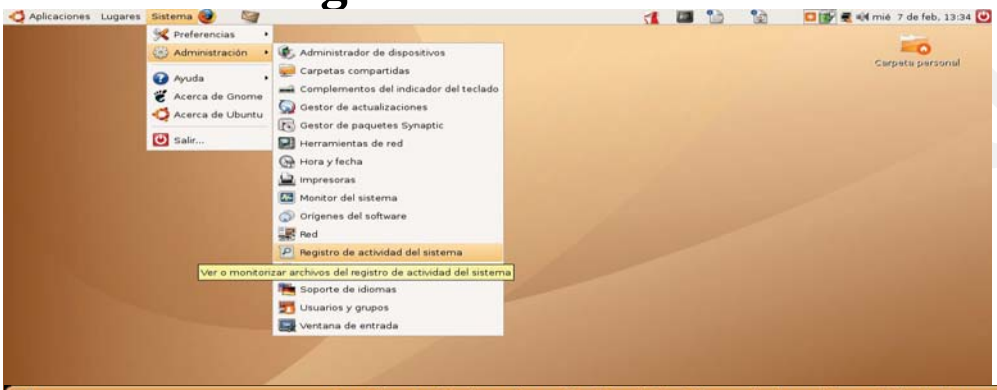
Seguridad en sistemas UNIX/Linux

```
jlgs@portatil: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jlgs@portatil:~$ ps -aux
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1632   536 ?        Ss   08:27   0:01 /sbin/init spla
root         2  0.0  0.0      0     0 ?        SN   08:27   0:00 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        S    08:27   0:00 [watchdog/0]
root         4  0.0  0.0      0     0 ?        S<   08:27   0:00 [events/0]
root         5  0.0  0.0      0     0 ?        S<   08:27   0:00 [khelper]
root         6  0.0  0.0      0     0 ?        S<   08:27   0:00 [kthread]
root         8  0.0  0.0      0     0 ?        S<   08:27   0:00 [kblockd/0]
root         9  0.0  0.0      0     0 ?        S<   08:27   0:00 [kacpid]
```

```
jlgs@portatil: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jlgs@portatil:~$ who -a
          system boot  2007-02-07 08:27
          `run-level' 2 2007-02-07 08:27                   último=
LOGIN     tty1         2007-02-07 08:27              3945 id=1
LOGIN     tty2         2007-02-07 08:27              3946 id=2
LOGIN     tty3         2007-02-07 08:27              3947 id=3
LOGIN     tty4         2007-02-07 08:27              3948 id=4
LOGIN     tty5         2007-02-07 08:27              3949 id=5
LOGIN     tty6         2007-02-07 08:27              3950 id=6
jlgs     ? :0          2007-02-07 08:28   ?          4708
jlgs     + pts/0       2007-02-07 13:39   .         12976 (:0.0)
root     + pts/1       2007-02-07 13:45   .         13203 (:0.0)
jlgs@portatil:~$
```

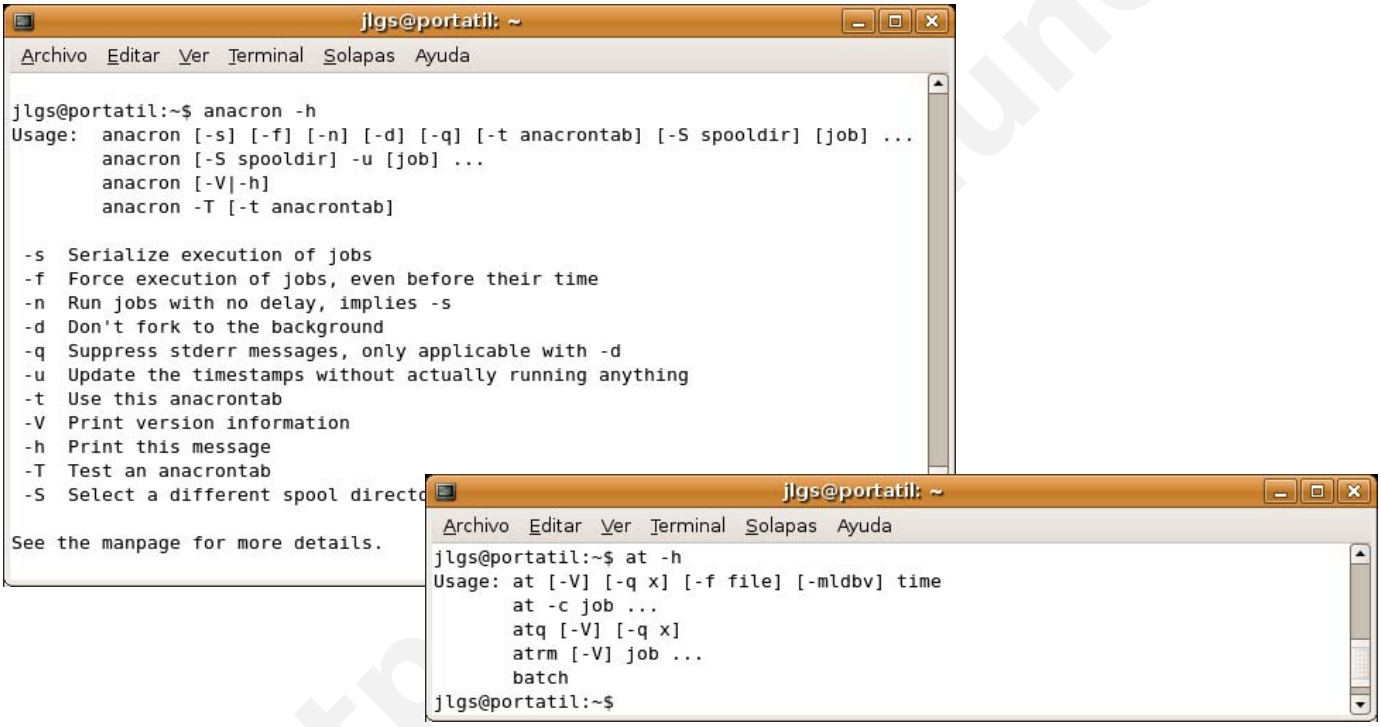
Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad en sistemas UNIX/Linux




```
/var/log/auth.log (monitorizado) - Visor del registro del sistema
Archivo  Editar  Vista  Ayuda
/var/log
  auth.log
  daemon.log
febrero  2007
lun mar mié jue vie sáb dom
29  30  31  1   2   3   4
5   6   7   8   9   10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  1   2   3   4
5   6   7   8   9   10  11
05/02/07
    Feb  5 09:17:01 localhost CRON[5938]: (pam_unix) session opened for user root by (uid=0
    Feb  5 09:17:01 localhost CRON[5938]: (pam_unix) session closed for user root
    Feb  5 09:52:33 localhost su[6937]: unknown configuration item `FAIL_DELAY'
    Feb  5 09:52:33 localhost su[6937]: Successful su for jlgs by root
    Feb  5 09:52:33 localhost su[6937]: + ??? root:jlgs
    Feb  5 09:52:33 localhost su[6937]: (pam_unix) session opened for user jlgs by (uid=0)
    Feb  5 09:52:34 localhost su[6937]: (pam_unix) session closed for user jlgs
82 líneas (6,6 Kib) - Última actualización: Wed Feb  7 13:17:01 2007
```

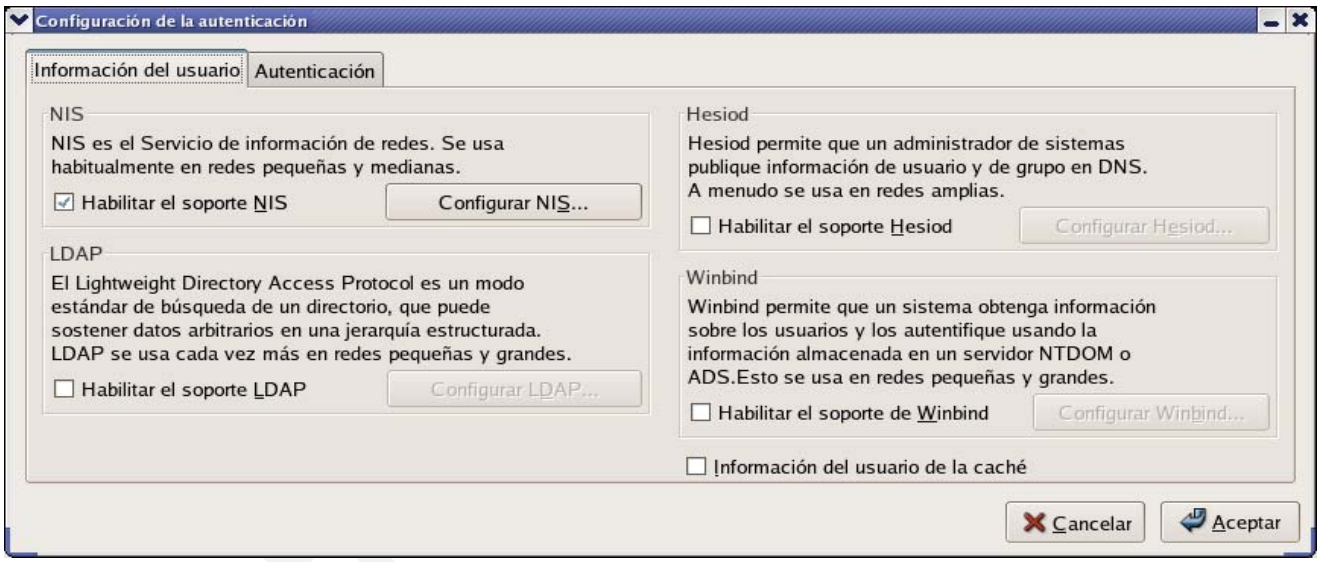
Seguridad en sistemas UNIX/Linux



Seguridad en sistemas UNIX/Linux

Fedora Core 2.6.17-1.2142_FC4

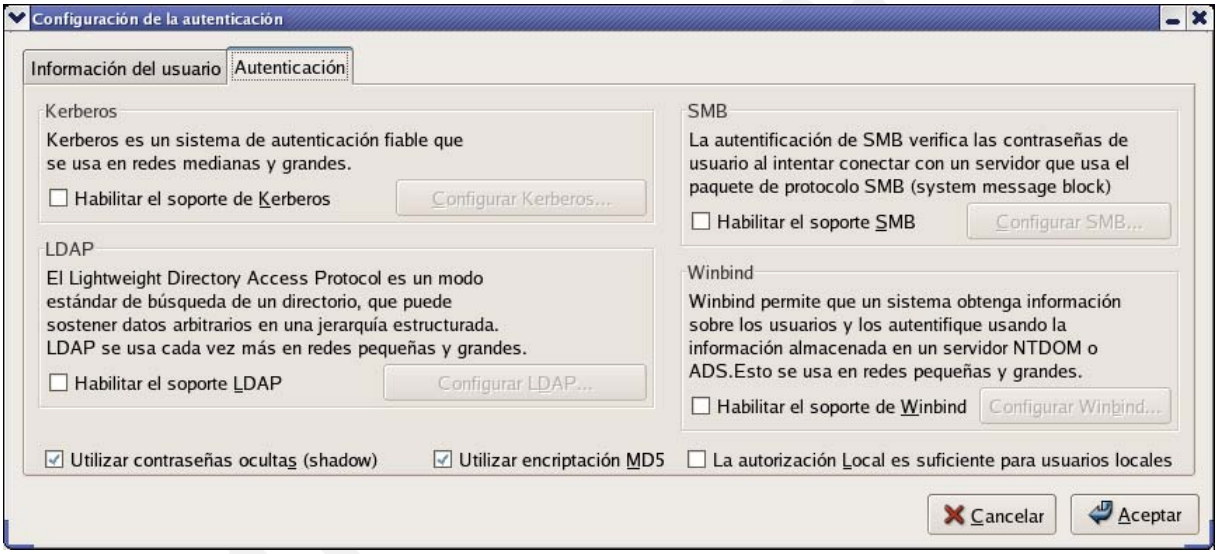
system-config-authentication



Seguridad en sistemas UNIX/Linux

Fedora Core 2.6.17-1.2142_FC4

system-config-authentication



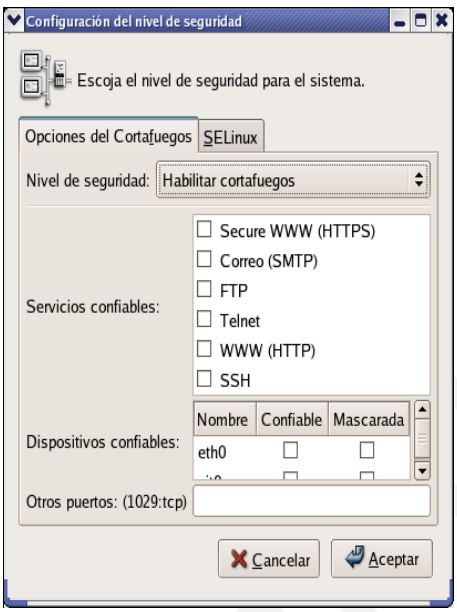
Nº 321

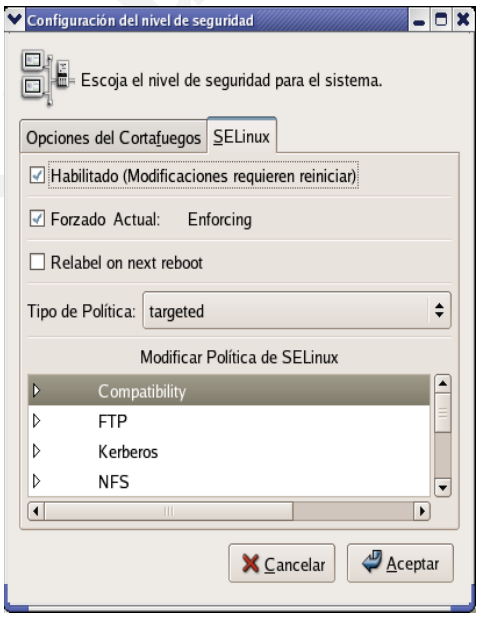
Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad en sistemas UNIX/Linux

Fedora Core 2.6.17-1.2142_FC4

system-config-securitylevel

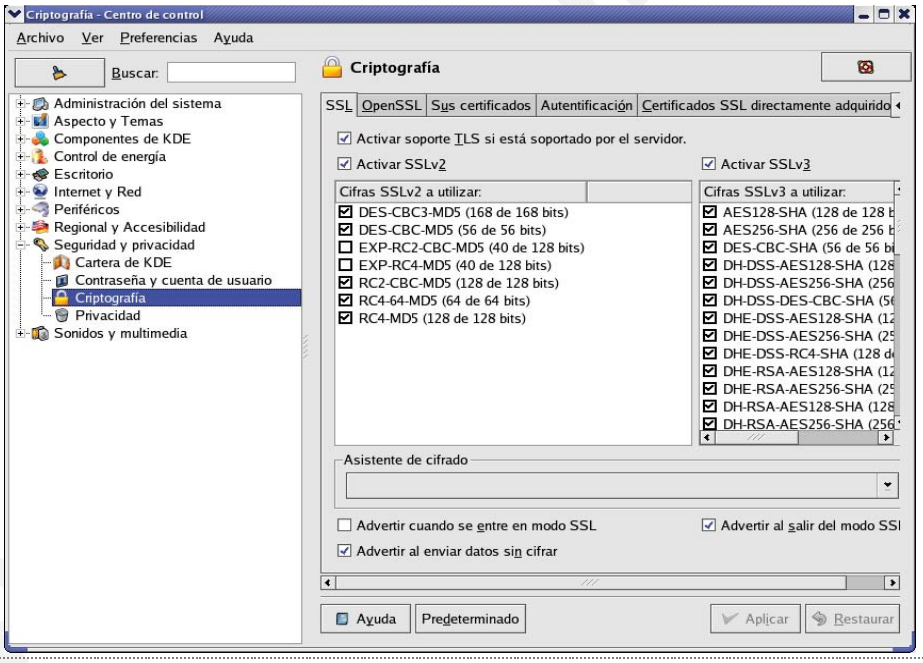




Nº 322

Seguridad en sistemas UNIX/Linux

Fedora Core 2.6.17-1.2142_FC4



Administración Avanzada de Redes de Sistemas UNIX/Linux

Seguridad en sistemas UNIX/Linux

- *iptables* (evolución de *ipchains*):
  - *packet filtering*: tipo de firewall en el kernel de Linux.
  - un paquete llega a un sistema sólo si las reglas del cortafuegos lo permiten. Puede ser filtrado por: tipo, @origen, @destino o número de puerto.
  - usadas para configurar, mantener e inspeccionar las tablas de las reglas del filtrado de paquetes.
  - pueden definirse varias tablas y cada una puede tener varias cadenas que son una lista de reglas.
  - cada regla especifica qué hacer con un paquete que coincide con un patrón de filtrado.

SSH (Secure SHell)

- ¿Qué es SSH?
  - SSH (Secure Shell) es un conjunto de aplicaciones encaminadas a asegurar las operaciones remotas entre sistemas.
  - Sustituye telnet, rcp, ftp... por aplicaciones que utilizan un sistema de claves asimétricas para cifrar las comunicaciones. Permite *tunneling* para otras aplicaciones.
  - Para GNU/Linux existe la versión OpenSSH desarrollada por el proyecto OpenBSD.

Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- Siempre he usado telnet, ftp, rcp... y nunca he tenido problemas. ¿Por qué debería cambiar a SSH?
  - Has tenido suerte. Cualquier *sniffer* (tcpdump, wireshark...) puede capturar en cuestión de segundos las claves de usuario utilizadas mediante estas técnicas (en texto plano y legible). SSH lo evita.
- ¿De donde lo descargo?
  - http://www.openssh.org

SSH (Secure SHell)

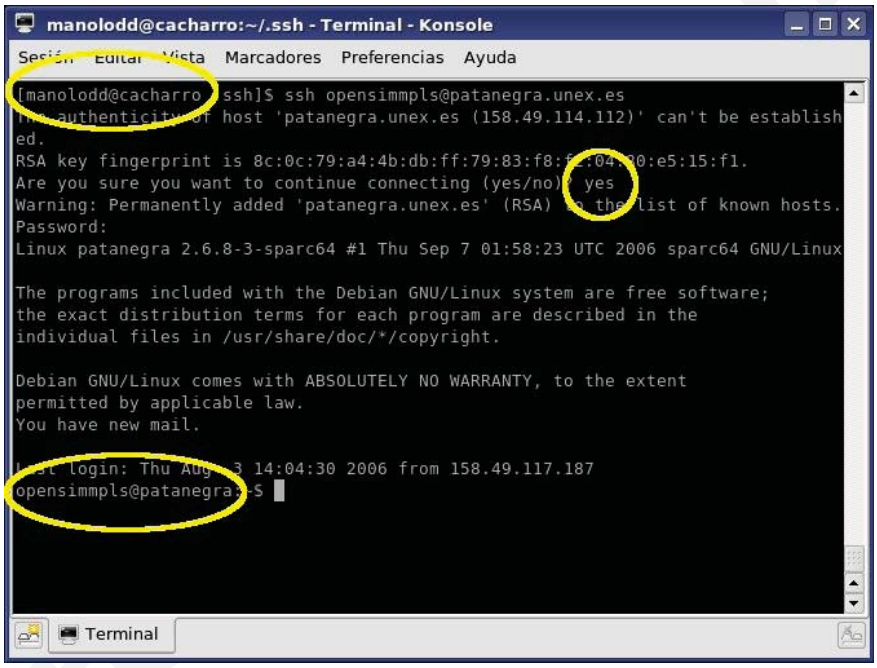
- El *toolkit* está compuesto por:
  - **ssh**: sustituto seguro de telnet.
  - **scp**: sustituto seguro de rcp.
  - **sftp**: sustituto seguro de ftp.
  - **sshd**: Servidor. Se instala en el host remoto.
  - **ssh-keygen**: utilidad para generar claves pública/privada.
  - **sftp-server**: subsistema servidor de FTP seguro.
  - **ssh-add**, **ssh-agent**, **ssh-keysign**: herramientas adicionales para la gestión de claves pública/privada y automatización de autenticación.

Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- **ssh**: Permite conectarse remotamente a una máquina en modo *shell*. También permite la ejecución remota de comandos.
  - ssh usuario@host.dominio comando
  - **usuario**: usuario dado de alta en host.dominio.
  - **host.dominio**: dirección o nombre de la máquina remota.
  - **comando**: comando a ejecutar en el servidor remoto. Debe estar allí. Si no se especifica, en lugar de ejecutar un comando remoto, hace *login* contra la máquina remota.

SSH (Secure SHell)



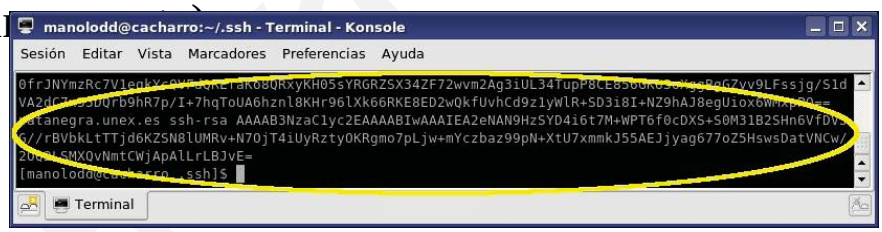
Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- En la primera conexión *ssh* aun *host* remoto concreto:
  - Se produce un intercambio de claves entre el cliente y el servidor.
  - Nuestro cliente *ssh* nos pregunta si deseamos confiar en ese *host* remoto con esa clave.
  - Si no aceptamos, la sesión *ssh* finaliza.
  - Si aceptamos, la sesión *ssh* se llevará a efecto y se nos pedirá el *password* del usuario en la máquina remota.
  - No se nos pedirá más si queremos confiar en el *host* remoto puesto que ya hemos dicho que si.

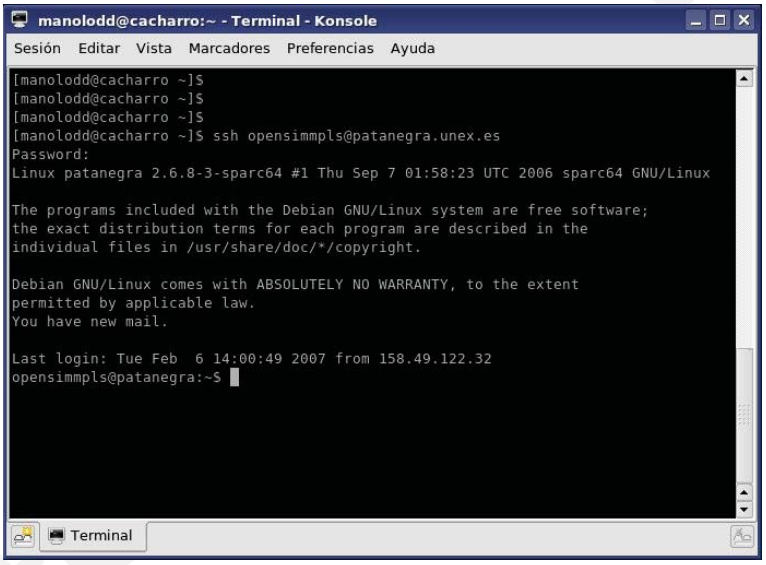
SSH (Secure SHell)

- La clave del servidor remoto, así como su nombre o dirección se almacenan en el fichero **~/.ssh/known_hosts**. Es por eso que ya no se pregunta más veces si confiamos en ese *host*.
- El fichero **~/.ssh/known_hosts** es válido para todas las herramientas del *toolkit*, por ello, si usamos *scp* o *sftp* con el mismo usuario y equipo remoto, se confiará por defecto en él (aunque se seguirá pidiendo la *password* de usuario de la máqui



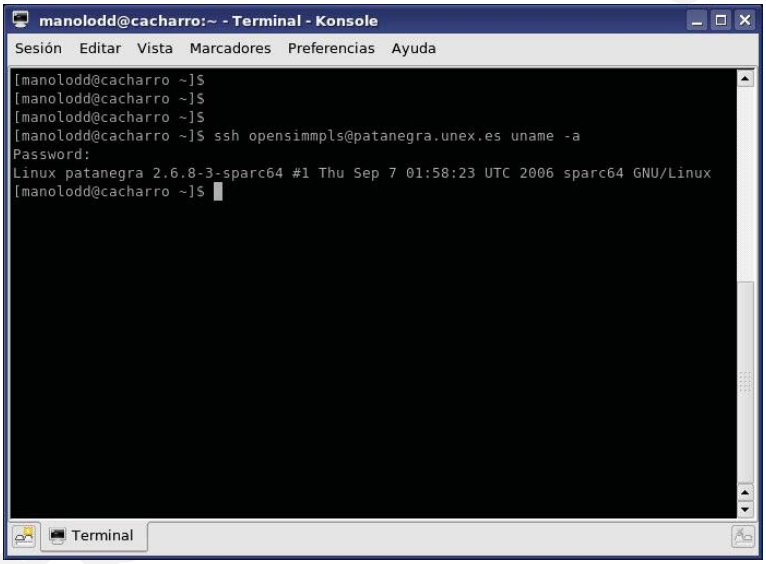
SSH (Secure SHell)

- Ejemplo: uso de *ssh* para conexión *shell* remota.



SSH (Secure SHell)

- Ejemplo: uso de *ssh* para ejecución de comando remoto.



SSH (Secure SHell)

- **scp:** es una utilidad que permite copiar ficheros de forma remota hacia o desde un servidor remoto.

  - **scp fichero usuario@host.dominio:dir**

  - **fichero**: fichero a copiar al *host* remoto.
  - **usuario**: usuario dado de alta en host.dominio.
  - **host.dominio**: dirección o nombre de la máquina remota.
  - **dir**: directorio existente en el *host* destino, especificado a partir del $HOME del usuario en aquella máquina.

SSH (Secure SHell)

- **scp:** es una utilidad que permite copiar ficheros de forma remota hacia o desde un servidor remoto.

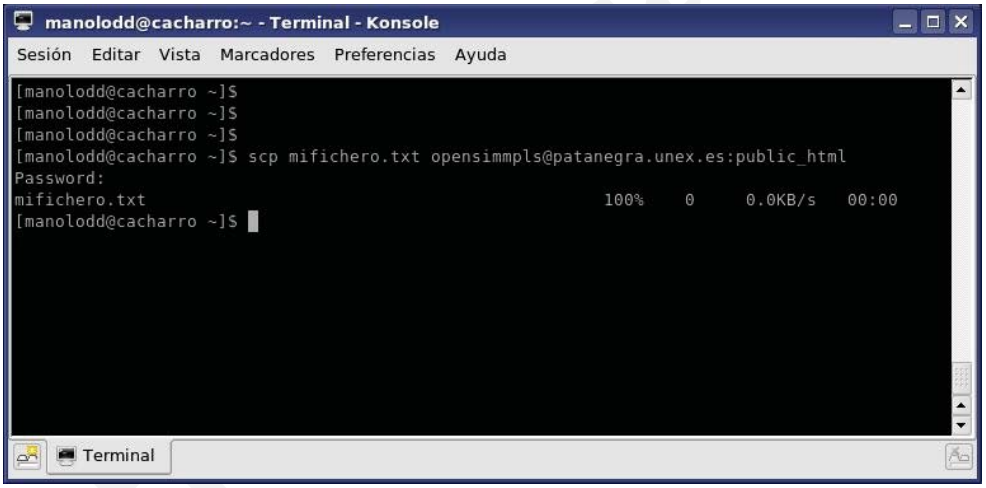
  - **scp usuario@host.dominio:dir/fichero dirlocal**

  - **usuario**: usuario dado de alta en host.dominio.
  - **host.dominio**: dirección o nombre de la máquina remota.
  - **dir**: directorio existente en el *host* destino, especificado a partir del $HOME del usuario en aquella máquina.
  - **fichero**: fichero a copiar del *host* remoto.
  - **dirlocal**: directorio local destino de los ficheros copiados.

Administración Avanzada de Redes de Sistemas UNIX/Linux

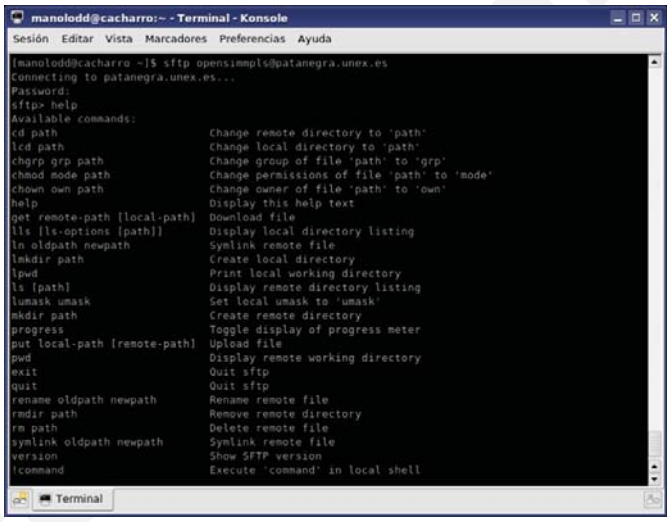
SSH (Secure SHell)

- Con el parámetro **-r**, se pueden copiar directorios completos tanto en uno como en otro sentido (especificar un directorio en lugar de un fichero)



SSH (Secure SHell)

- **sftp**: sustituye al cliente *ftp*. El protocolo es distinto aunque la forma de trabajo de cara al cliente es similar.



SSH (Secure SHell)

- **sshd**: Es el servidor *ssh*. Actúa de servidor para *scp*, *ssh* y *sftp* (se encarga de llamar a *sftp-server* cuando hace falta).
- Se configura en **/etc/ssh/sshd_config**:
  - **HostKey**: Clave privada de RSA o DSA del host. Normalmente **/etc/ssh/ssh_host_rsa_key**.
  - **PubkeyAuthentication**: Si es **yes**, entonces se permite la autenticación de usuarios mediante clave pública.
  - **AuthorizedKeysFile**: Lugar donde están almacenadas las claves públicas de los usuarios que se pueden conectar al sistema. Normalmente **%h/.ssh/authorized_keys** (%h = directorio *home* de cada usuario).

SSH (Secure SHell)

- Se configura en **/etc/ssh/sshd_config**:
  - **PasswordAuthentication**: Si el valor de esta opción es **yes**, se permite la autenticación de usuarios mediante contraseñas (del sistema).
  - **Subsystem sftp /usr/libexec/openssh/sftp-server**: si esta línea aparece así en el fichero, *sshd* llamará a *sftp-server* cuando sea necesario.

Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- Autenticación mediante claves públicas.
  - Método mejorado de autenticación ante un servidor SSH.
  - La seguridad recae en el usuario: protección de su clave privada.
  - Permite la selección del algoritmo criptográfico.
  - Se establecen relaciones de confianza.
  - Bien usado, permite la administración **segura** y **desatendida** de equipos de forma remota.

SSH (Secure SHell)

- ¿Cómo puedo autenticarme ante un *host* remoto mediante clave pública?
  - Lo primero es crear un par de claves pública/privada:
    - **cd ~/.ssh**
    - **ssh-keygen -t dsa**
  - Pedirá el fichero donde almacenarla. Por defecto, **~/.ssh/id_dsa**

Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- ¿Cómo puedo autenticarme ante un *host* remoto mediante clave pública?
  - Luego pedirá la frase (importante no olvidar) que puede ser tan compleja como se desee, o se dejarse en blanco.
  - Al final en **~/.ssh** genera dos ficheros
    - **id_dsa**: clave privada.
    - **id_dsa.pub:** clave pública.

SSH (Secure SHell)

- ¿Cómo puedo autenticarme ante un *host* remoto mediante clave pública?
  - Hay que copiar la clave pública al *host* remoto.
  - Hay que conectarse al host remoto (estaría bien con ssh) y una vez allí:
    - **cat id_dsa.pub >> ~/.ssh/authorized_keys**
  - Lo que añadirá nuestra clave pública a la lista de autorizados del *host* remoto.

Administración Avanzada de Redes de Sistemas UNIX/Linux

SSH (Secure SHell)

- ¿Cómo puedo autenticarme ante un *host* remoto mediante clave pública?
  - Ahora, aún en el servidor remoto, comprobamos que los permisos son los adecuados:
    - **chmod 700 ~/.ssh**
    - **chmod 644 ~/.ssh/authorized_keys**
  - Ya está, la próxima conexión nos pedirá la frase de la clave pública, y no la *password* de la máquina remota.

SSH (Secure SHell)

- ¡Siempre pide la frase. Esto no es DESATENDIDO.!

- Se pide la frase si se especificó al crear las claves pública/privada. Si no, no (y conserva toda la seguridad).
- Para usar *ssh* en tareas desatendidas:

  – Ejecución de *scripts* remotos no interactivos con *ssh*.
  – Copias de seguridad remotas con *scp*.
  – Otras...

  – Es necesario no poner frase a la clave.

SSH (Secure SHell)

SSH (Secure SHell)

- En autenticación mediante clave pública/privada, con una frase establecida, puede interesarnos que no se nos pida la clave permanentemente. *ssh-agent* debe controlar el terminal.
  - **ssh-agent bash**
  - **ssh-add**
- Entonces se nos pedirá la frase. Y no se nos pedirá más mientras no cerremos el terminal; independientemente del número de veces que nos conectemos al equipo remoto.

Administración Avanzada de Redes de Sistemas UNIX/Linux

GnuPG (GNU Privacy Guard)

- Clon libre de PGP *(Pretty Good Privacy)*.
- En la versión 8, PGP pasó a ser código cerrado. Se inició a partir de entonces *GnuPG* (GPG).
- Es un sistema de seguridad que utiliza claves asimétricas para su funcionamiento.
- El usuario tiene un par de claves pública/privadas.
- La clave privada es secreta, personal e intransferible.
- La clave pública debe estar disponible para cualquiera con el que deseemos “funcionar” mediante GPG.
- Se basa en relaciones de confianza (No existe C.A.)

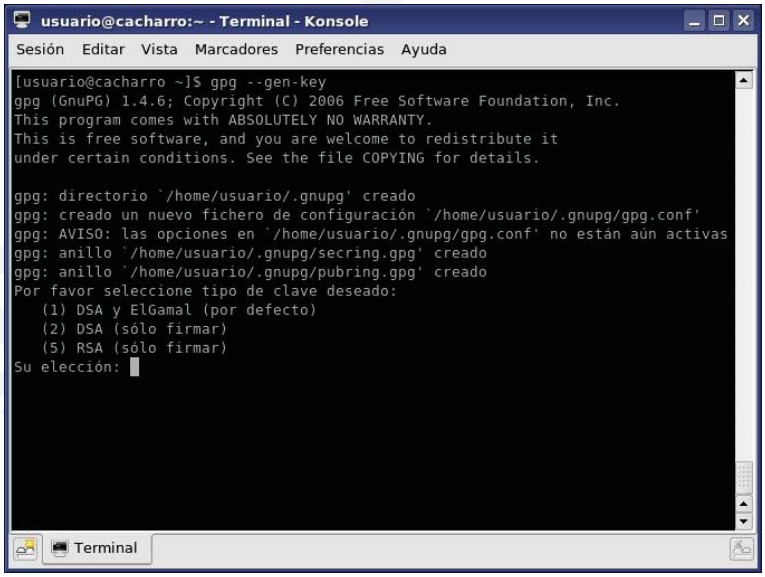
GnuPG (GNU Privacy Guard)

- ¿Dónde se obtiene GPG?
  - http://www.gnupg.org
- ¿Que puedo hacer con GPG?
  - **Firmar**. No repudio, integridad.
  - **Cifrar**. Intimidad.
  - **Verificar**. No repudio, integridad.
  - **Descifrar**. Intimidad.

Administración Avanzada de Redes de Sistemas UNIX/Linux

GnuPG (GNU Privacy Guard)

- Primer paso. Generar el par de claves.
  - **gpg –gen-key**



- Pasos:
  - Elegir algoritmo.
  - Elegir tamaño.
  - Elegir caducidad.
  - Datos personales.
  - Especificar frase.

GnuPG (GNU Privacy Guard)

- Las claves se generan en **~/.gnupg**
  - **gpg.conf**: fichero de configuración de GnuPG.
  - **pubring.gpg**: Llavero. Contiene las claves públicas. Por ahora, sólo la nuestra recién generada.
  - **secring.gpg**: Llavero. Contiene las claves privadas. Por ahora sólo la recién generada. Generalmente no contendrá más.

GnuPG (GNU Privacy Guard)

- Operaciones más comunes para gestión de llaveros.
  - **gpg --list-public-keys**: Lista las claves públicas.
  - **gpg --list-secret-keys**: Lista las claves privadas.
  - **gpg --fingerprint**: Muestra la huella de una clave.
  - **gpg --export**: Exporta la clave pública de un par.
  - **gpg --import**: Importa una clave pública de un fichero.
  - **gpg --delete-key**: Elimina una clave pública del llavero.

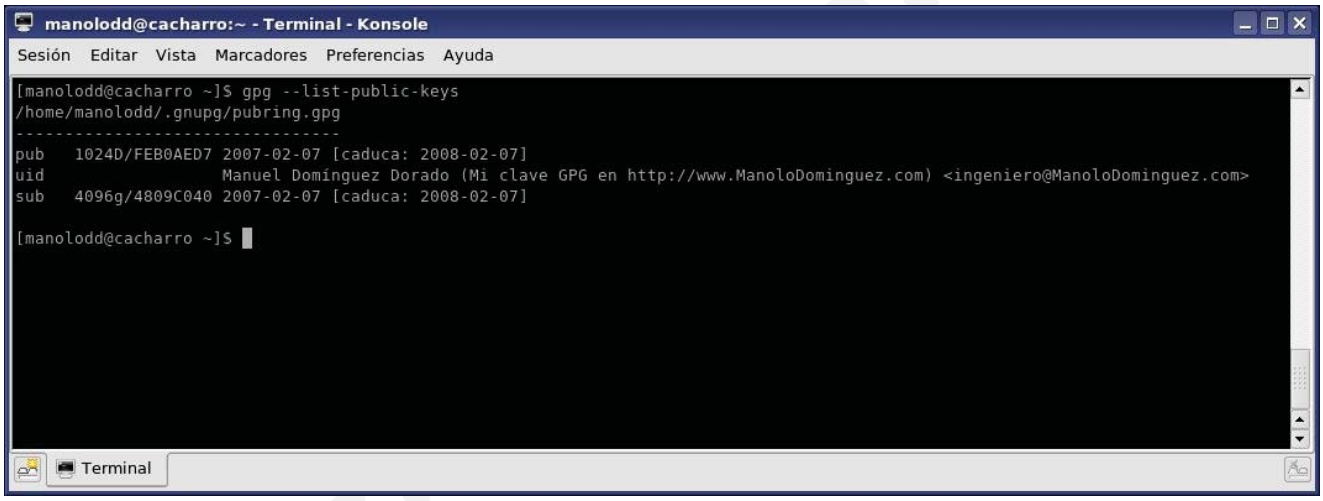
GnuPG (GNU Privacy Guard)

- Operaciones más comunes mediante *gpg*.
  - **gpg --sign**: firma un fichero.
  - **gpg --encript**: cifra un fichero.
  - **gpg --verify**: comprueba la firma de un fichero firmado.
  - **gpg --decrypt**: descifra un fichero cifrado.

Administración Avanzada de Redes de Sistemas UNIX/Linux

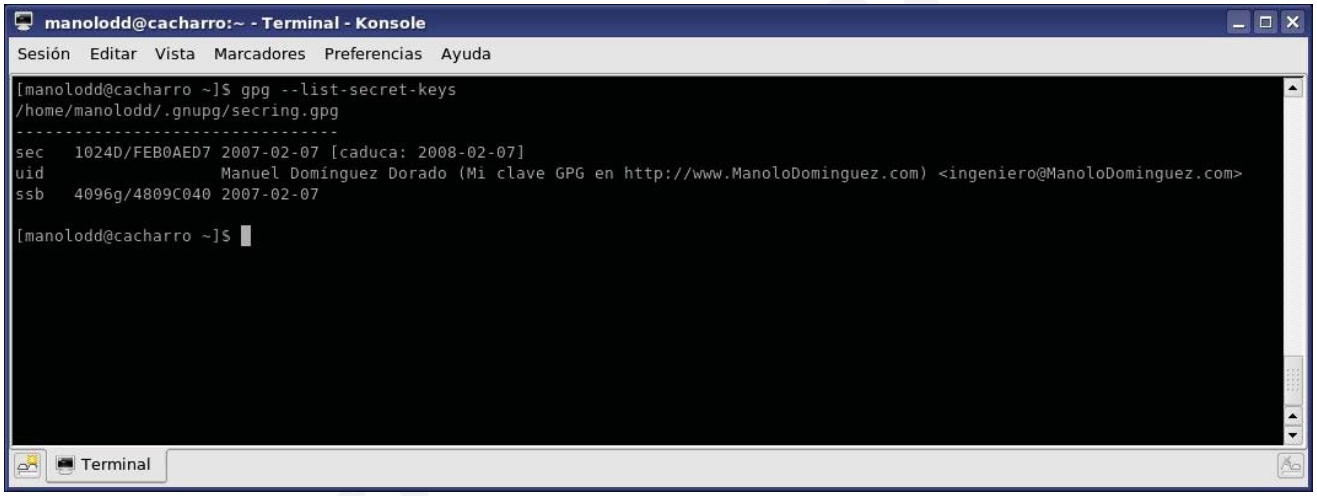
GnuPG (GNU Privacy Guard)

- **gpg --list-public-keys**



GnuPG (GNU Privacy Guard)

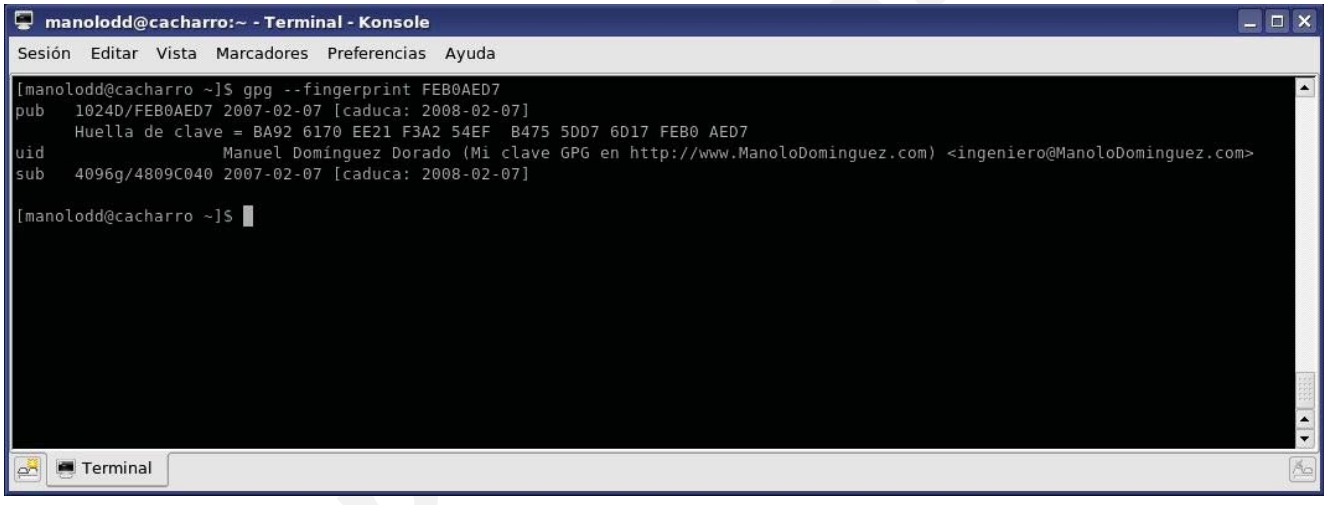
- **gpg --list-secret-keys**



Administración Avanzada de Redes de Sistemas UNIX/Linux

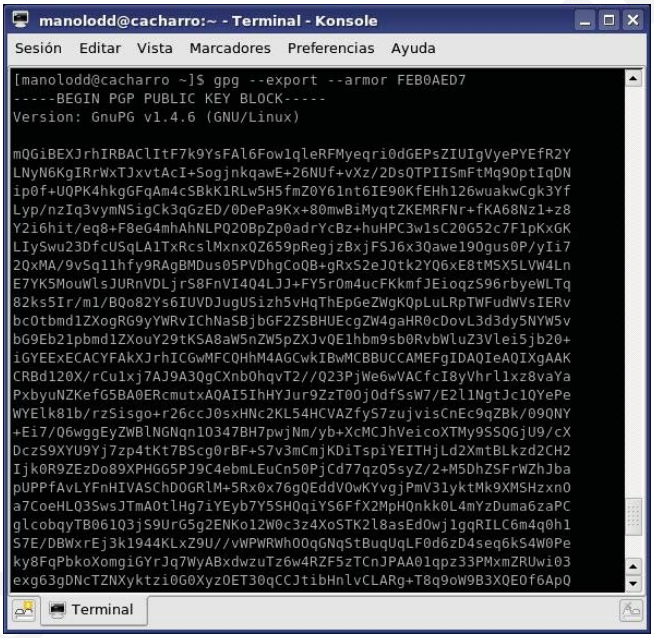
GnuPG (GNU Privacy Guard)

- **gpg --fingerprint**



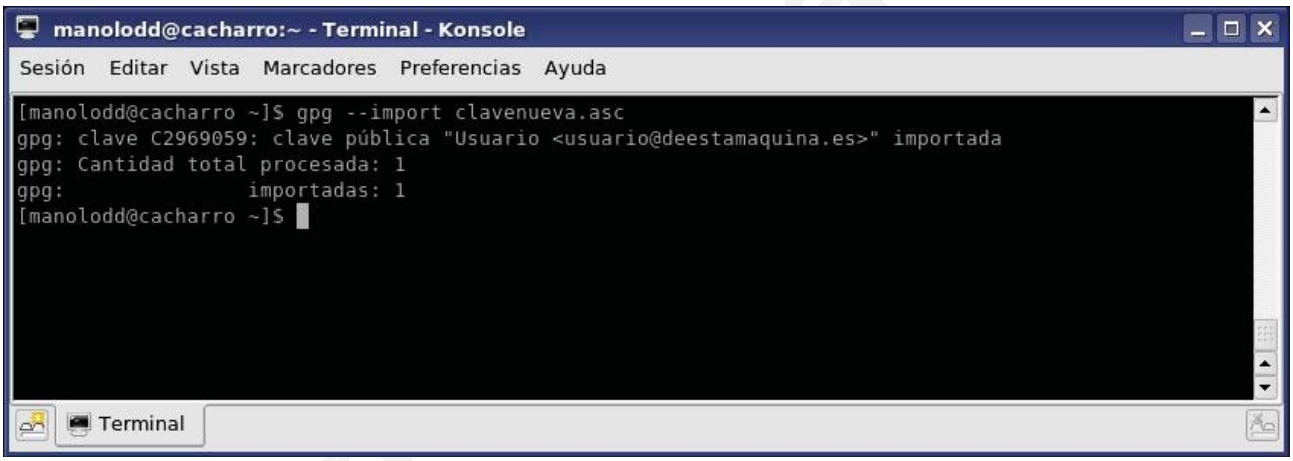
GnuPG (GNU Privacy Guard)

- **gpg --export**



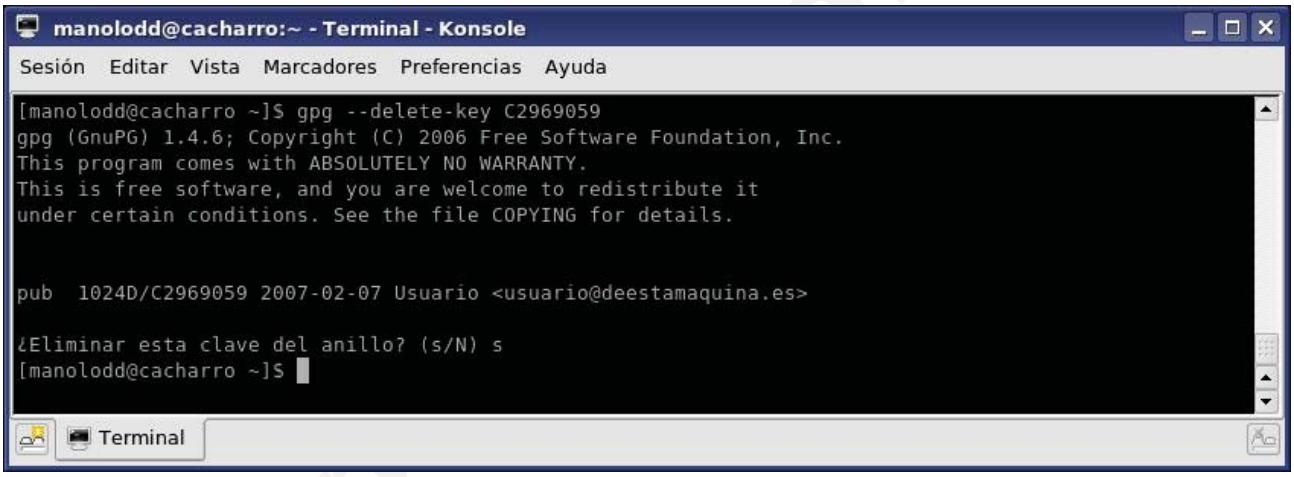
GnuPG (GNU Privacy Guard)

- **gpg --import**



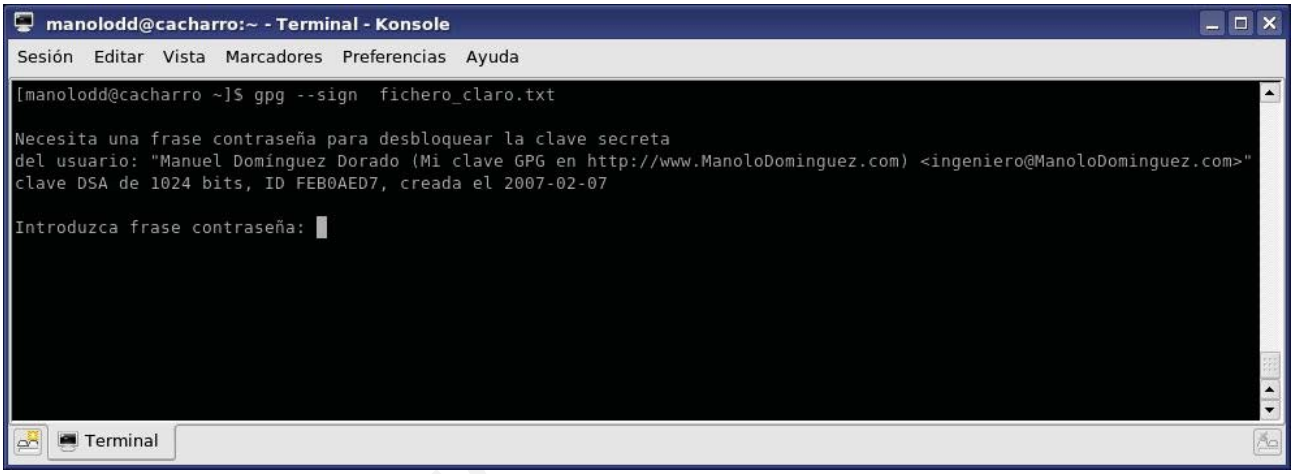
GnuPG (GNU Privacy Guard)

- **gpg --delete-key**



GnuPG (GNU Privacy Guard)

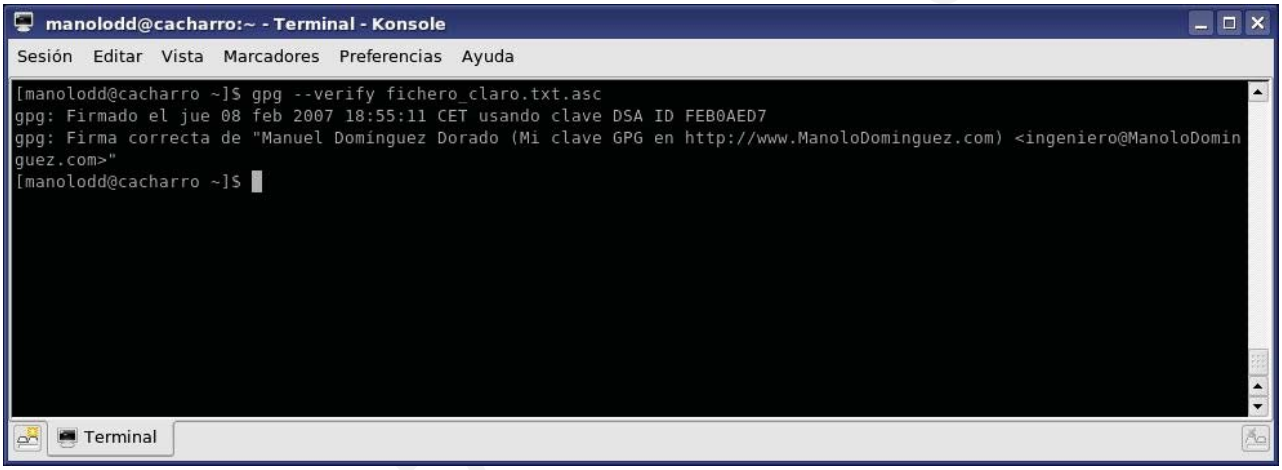
- **gpg --sign**



- Esto produce un fichero *.gpg o *.asc, que es el fichero firmado.

GnuPG (GNU Privacy Guard)

- **gpg --verify**

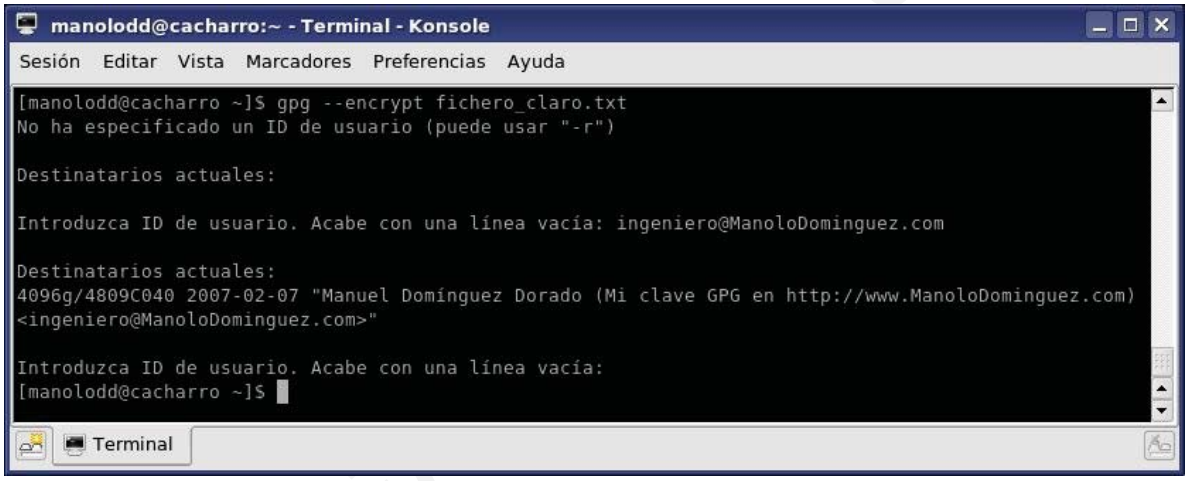


- El comando se hace sobre el fichero firmado (*.gpg o *.asc).

Administración Avanzada de Redes de Sistemas UNIX/Linux

GnuPG (GNU Privacy Guard)

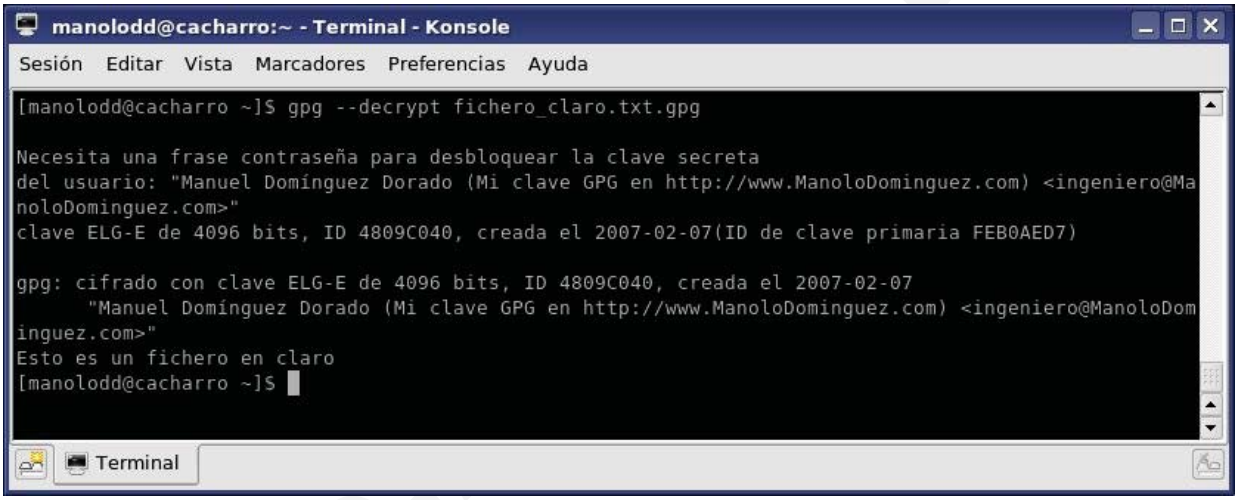
- **gpg --encrypt**



- Esto produce un fichero *.gpg o *.asc, que es el fichero cifrado.

GnuPG (GNU Privacy Guard)

- **gpg --decrypt**



- El comando se hace sobre el fichero cifrado (*.gpg o *.asc).

Mantenimiento Preventivo

Mantenimiento Preventivo

- Monitorización del sistema
- Tareas programadas
- Copias de seguridad del sistema
- Actualizaciones del sistema
- Analizadores de logs

Tarea / Proceso

- En este punto tendremos que empezar a determinar que es un proceso y una tarea.
  - Un programa se transformaba en proceso en el momento en que este se ejecutaba y estaba en memoria. Además del nombre que el proceso recibe, que es el nombre del programa que esta corriendo, recibe también un número identificativo llamado PID (process ID, o ID de proceso).
  - Una tarea se refiere al proceso que esta ejecutandose en un momento determinado.

Comando ps

- ps nos devuelve la lista de procesos que se está ejecuntado
  - PID TTY TIME CMD
  - 11229 pts/0 00:00:00 su
  - 11230 pts/0 00:00:00 bash
  - 13961 pts/0 00:00:00 ps
- Los parámetros que nos devuelve por defetco es PID que es el identificador de proceso; TTY identifica la consola donde se esta ejecutando el proceso; TIME nos indica la cantidad de tiempo total que el proceso se ha estado ejecutando y CMD que es el comando en cuestión

Administración Avanzada de Redes de Sistemas UNIX/Linux

Comando free

- El comando free muestra la utilización de la memoria del sistema.
  - total used free shared buffers cached
  - Mem: 507632 497368 10264 0 9104 202632
  - -/+ buffers/cache: 285632 222000
  - Swap: 979924 73240 906684
- La fila Mem: muestra la utilización de la memoria física; la fila Swap: muestra la utilización del swap del sistema; -/+ buffers/cache: muestra la cantidad de memoria actualmente dedicada a las memorias intermedias del sistema.

Comando top

- Mientras que free muestra solamente información relacionada con la memoria, el comando top hace un poquito de todo. Utilización del CPU, estadísticas de procesos, utilización de memoria — top lo monitoriza todo. Además, a diferencia de free, el comportamiento predeterminado de top es el de ejecutarse de forma continua
- La pantalla se divide en dos secciones. La parte superior contiene información relacionada con el estatus general del sistema. La sección de abajo muestra estadísticas a nivel de procesos. Es posible cambiar lo que se muestra mientras top se ejecuta. Por ejemplo, por defecto top muestra procesos activos y ociosos.

Comando vmstat

- Con vmstat, es posible obtener una vista general de los procesos, memoria, swap, E/S, sistema y actividad de CPU.
- La primera línea divide los campos en seis categorías, incluyendo procesos, memoria, swap, E/S, sistema y estadísticas relacionadas al CPU. La segunda línea identifica aún más los contenidos de cada campo, haciendo más fácil escanear datos para ver estadísticas específicas.

Campos vmstat

- Los campos relacionados a procesos:
  - r — El número de procesos ejecutables esperando para acceder al CPU
  - b — El número de procesos en un estado dormido contínuo
- Los campos relacionados a la memoria:
  - swpd — La cantidad de memoria utilizada
  - free — La cantidad de memoria libre
  - buff — Memoria utilizada por las memorias intermedias
  - cache — La cantidad de memoria utilizada como caché

Administración Avanzada de Redes de Sistemas UNIX/Linux

Campos vmstat

- Los campos relacionados a swap son:
  - si: La cantidad de memoria intercambiada desde el disco
  - so: La cantidad de memoria intercambiada hacia el disco
- Los campos relacionados con E/S son:
  - bi : Los bloques enviados a un dispositivo de bloques
  - bo : Los bloques recibidos desde un dispositivo de bloques
- Los campos relacionados al sistema son:
  - in : El número de interrupciones por segundo
  - cs : El número de cambios de contexto por segundo

Campos vmstat

- Los campos relacionados al CPU son:
  - us — El porcentaje de tiempo que el CPU ejecutó código de nivel del usuario
  - sy — El porcentaje de tiempo que el CPU ejecutó código de nivel del sistema
  - id — El porcentaje de tiempo que el CPU estaba desocupado
  - wa — Esperas de E/S

vmstat

- Cuando se ejecuta vmstat sin opciones, solamente se muestra una línea. Esta línea contiene promedios, calculados desde la última vez que se arrancó el sistema.
- Sin embargo, la mayoría de los administradores de sistemas no confían en los datos en esta línea, pues los tiempos en que fueron recopilados varían. En su lugar, la mayoría de los administradores tomas ventaja de la habilidad de vmstat de mostrar repetidamente datos de la utilización de recursos en intervalos establecidos. Por ejemplo, el comando vmstat 1 muestra una nueva línea de utilización de datos cada segundo.

Munin

- Munin es un programa que monitoriza el servidor en el que esté montado.
- Crea rapidamente graficos para cada aspecto de los servidores (promedio de carga, uso de memoria, uso del CPU, trafico a traves de las ethx, todo MySQL, etc.) sin demasiadas configuraciones.
- De instalación sencilla, únicamente se han de poner el comando **apt-get install munin munin-node** para comenzar la instalación.

Administración Avanzada de Redes de Sistemas UNIX/Linux

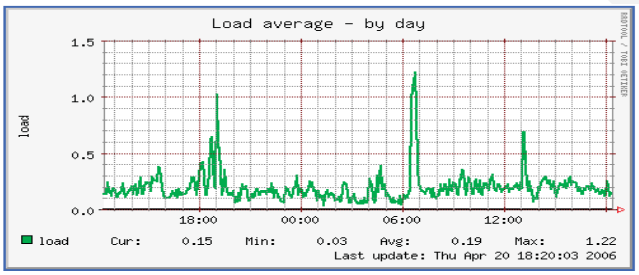
Munin

- Una vez arrancado se pueden configurar opciones de guardado de información y gráficas mediante la edición del archvo **/etc/munin/munin.conf**
- Se le pueden aplicar métodos de protección para que nadie más que nosotros pueda ver la información usando un fichero de .httaccess como el siguiente

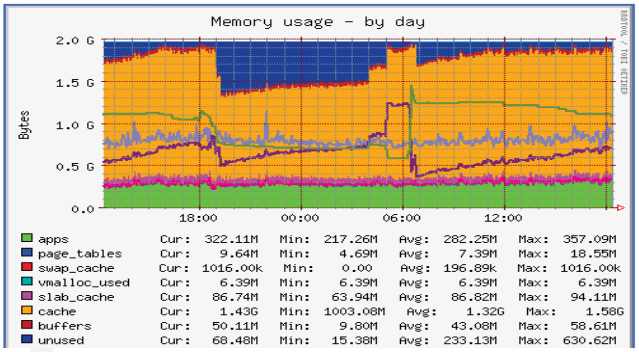
Munin

  - AuthType Basic
  - AuthName "Members Only"
  - AuthUserFile /var/www/www.example.com/.htpasswd
  - <limit GET PUT POST>
  - require valid-user
  - </limit>
- Una vez hecho esto creamos el password para el usuario admin:
  - **htpasswd -c /var/www/www.example.com/.htpasswd admin**

Munin Gráficas



:: Memory usage



Herramientas de backup

- Herramientas clásicas en línea de comandos.
- Libres: Amanda, BRU, DAR, etc.
- Copias de seguridad:
  - ¿Qué incluir?
  - ¿Cómo hacerlas?
  - ¿Cuándo hacerlas?
  - ¿En dónde hacerlas?

Tipos de copias de seguridad

- *full dump* (copia un sistema de ficheros completo).
- *incremental* (sólo se copian los ficheros que cambiaron desde el último backup realizado).
- *selectiva*, eligiendo los ficheros a copiar.
- *multivolumen* en más de un disquete, cinta, dat, streamer, cdrom, dvd.

Herramientas en línea de comandos para backup

- Herramientas genéricas: *cpio, tar y dd*.
  - *tar* (tape archive, permite n ficheros de una jerarquía de directorios.
  - *cpio* (copy in/copy out, crea un fichero en disco o cinta de uno o varios ficheros entrados desde teclado. Copia jerarquías de directorios).
  - *dd* (se usa para convertir y copiar ficheros con diversos formatos).
- Comandos Solaris para backups: *ufsdump/ufsrestore*
  - *ufsdump/ufsrestore* (para copiar sistemas de ficheros completos, o directorios y ficheros individuales).
- Comandos Linux para backups: *dump/restore*

Tareas programadas (cron)

- cron / anacron
  - /etc/crontab
  - /etc/cron.d
  - /etc/cron.hourly
  - /etc/cron.daily
  - /etc/cron.weekly
  - /etc/cron.monthly

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

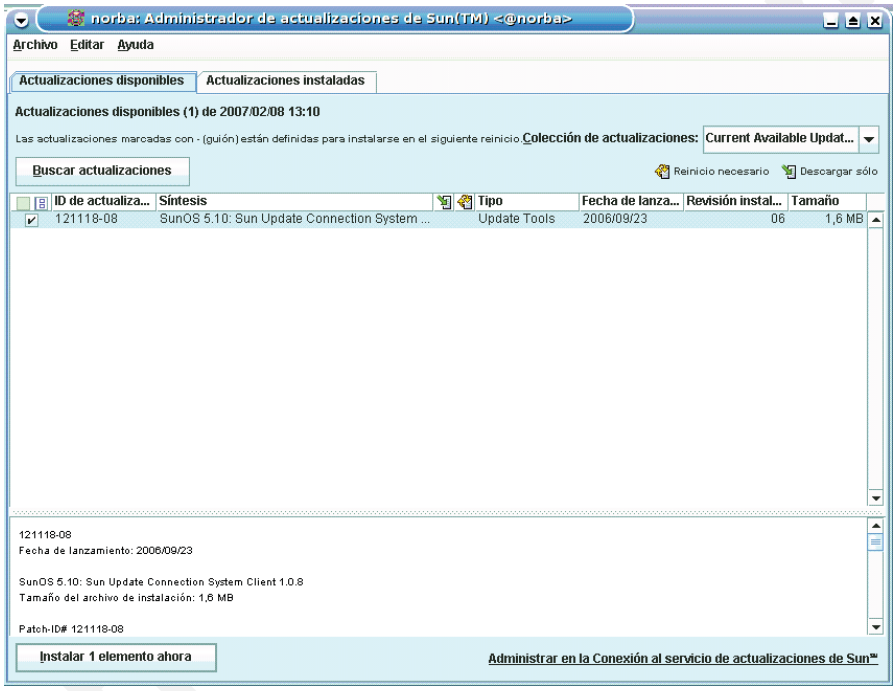
Actualizaciones de seguridad

- Sin caer en “versionitis”, es importante mantener actualizados los sistemas.
- Objetivo principal: evitar vulverabilidades.
- ¿Actualización manual?
- ¿Aviso de actualizaciones disponibles?
- Actualizaciones desatendidas. Problemas:
  - Actualización de kernel o módulos.
  - Actualización con dependencias.
  - Archivos de configuración.

cron-apt

- Permite actualizaciones desatendidas.
- No tiene interfaz gráfico. Configuración en /etc/cron-apt
- ¿Nocturnas?
- Configuración de parámetros de actualización.
- Envío de informes por correo electrónico.
- Disponible para Debian/Ubuntu

Solaris Update Manager



Analizadores de logs

- Permiten, al administrador de sistemas, tener una idea de cómo están funcionando los equipos.
- Resumen la generalmente profusa información de los log del sistema.
- Parametrizables.
- ¿Envío diario?

logwatch

```
################### Logwatch 7.2.1 (01/18/06) ####################
       Processing Initiated: Thu Feb  8 04:06:52 2007
       Date Range Processed: yesterday
                             ( 2007-Feb-07 )
                             Period is day.
     Detail Level of Output: 0
             Type of Output: unformatted
          Logfiles for Host: gateway
 ##################################################################

--------------------- MailScanner Begin ------------------------


MailScanner Status:
       60 messages Scanned by MailScanner
       823.8 Total KB
       19 Spam messages detected by MailScanner
              19 Spam messages with action(s) spam@localhost,forwa
       60 Messages delivered by MailScanner

**Unmatched Entries**
   Expired 1 records from the SpamAssassin cache : 16 Time(s)
   Expired 2 records from the SpamAssassin cache : 11 Time(s)
   Connected to SpamAssassin cache database : 7 Time(s)
   Using SpamAssassin results cache : 7 Time(s)
   Expired 5 records from the SpamAssassin cache : 2 Time(s)
   Expired 6 records from the SpamAssassin cache : 1 Time(s)
   MailScanner child dying after Bayes rebuild : 1 Time(s)
   Expired 3 records from the SpamAssassin cache : 1 Time(s)
   Expired 4 records from the SpamAssassin cache : 1 Time(s)

50 Ignored Lines

---------------------- MailScanner End -------------------------
```

```
--------------------- SSHD Begin ------------------------


Refused incoming connections:
      ::ffff:201.245.93.82 (::ffff:201.245.93.82): 35 Time(s)
      ::ffff:59.74.112.9 (::ffff:59.74.112.9): 2 Time(s)
      ::ffff:75.34.107.76 (::ffff:75.34.107.76): 2 Time(s)
      ::ffff:88.191.11.107 (::ffff:88.191.11.107): 1 Time(s)

---------------------- SSHD End ------------------------


--------------------- XNTPD Begin ------------------------


Time Reset 2 times (total: -0.067222 s  average: -0.033611 s)

Total synchronizations 3 (hosts: 2)

---------------------- XNTPD End ------------------------


--------------------- Disk Space Begin ------------------------

Filesystem                            Size  Used Avail Use%  Mounted on
/dev/hda1                             111G  105G  4.7G  96%  /export
/dev/hde2                              20G   11G  8.8G  54%  /
/dev/hde1                              98M   12M   85M  12%  /boot


---------------------- Disk Space End ------------------------


###################### Logwatch End #########################
```

Nº 387