



Manual de Administración de entorno y servicios en Canaima GNU/Linux

Caracas, Octubre de 2009

Créditos y licencia



© 2008-2009 Centro Nacional de Tecnologías de
Información

© 2008-2009 ONUVA Integración de Sistemas

Este documento se distribuye al público como *documentación y conocimiento libre* bajo los términos de la Licencia Pública General GNU, que puede obtener en la dirección Web:

<http://www.gnu.org/copyleft/gpl.html>

Convenciones tipográficas

*Texto enfatizado, anglicismos, **texto resaltado**, comandos, salidas, paquetes o contenido de archivos.*



Indica información muy importante con respecto al contenido



Indica comandos, salidas en pantalla o contenido de archivos



Indica los pasos de un procedimiento



Contenido

| | |
|--|----|
| Créditos y licencia..... | 2 |
| Convenciones tipográficas..... | 2 |
| UNIDAD I: INTRODUCCIÓN A CANAIMA GNU/LINUX..... | 14 |
| Tema 1: Sistema operativo GNU/Linux..... | 14 |
| Tema 2: Distribuciones GNU/Linux..... | 15 |
| Componentes de software..... | 15 |
| Administrador de paquetes RPM..... | 16 |
| Tema 3: Distribución Canaima GNU/Linux..... | 17 |
| Historia y motivación..... | 17 |
| Características diferenciales..... | 18 |
| Escenarios de aplicación..... | 18 |
| Tema 4: Plataformas colaborativas..... | 19 |
| Sitios Web oficiales..... | 19 |
| Elementos colaborativos de Canaima GNU/Linux..... | 19 |
| Foro..... | 19 |
| Wiki..... | 19 |
| Listas de correo..... | 20 |
| Sistema de manejo de versiones..... | 20 |
| Sistema de construcción de paquetes..... | 21 |
| UNIDAD II: INSTALACIÓN DE GNU/LINUX - DISTRIBUCIÓN CANAIMA. | 22 |
| Tema 1: Consideraciones previas a las instalación..... | 22 |
| Tema 2: Obteniendo información del hardware. | 23 |
| Tema 3: Medios de instalación. | 24 |
| Tema 4: Requisitos mínimos de hardware. | 25 |
| Requisitos de Memoria y Espacio en Disco Duro..... | 25 |
| Dispositivos de red..... | 26 |
| Tema 5: Nomenclatura para discos y particiones..... | 27 |
| Tema 6: Esquema de Particiones..... | 28 |



| | |
|---|----|
| Estructura de directorios en Canaima GNU/Linux. | 28 |
| Esquema básico..... | 30 |
| Esquema avanzado. | 30 |
| Tema 7: Gestor de Arranque..... | 32 |
| Tema 8: Proceso de Instalación de Canaima GNU/Linux..... | 33 |
| Preparando su sistema para la instalación..... | 33 |
| Iniciando la instalación | 34 |
| Durante la instalación. | 35 |
| Configuración de la red | 36 |
| Particionamiento..... | 36 |
| Finalizando la instalación. | 37 |
| UNIDAD III: Interpretador de comandos (SHELL)..... | 38 |
| Tema 1: El SHELL de Canaima GNU/Linux..... | 38 |
| Tema 2: Presentación del SHELL - Indicador del sistema..... | 38 |
| ¿Qué es un prompt?..... | 39 |
| PROMPT de usuarios..... | 39 |
| PROMPT del administrador..... | 40 |
| Tema 3: Instrucciones al SHELL..... | 41 |
| Tema 4: Entorno de funcionamiento del SHELL..... | 43 |
| Variables de Entorno y Configuraciones..... | 43 |
| Variables de entorno..... | 44 |
| Tema 5: Flujos de entrada y salida en el SHELL..... | 46 |
| Entrada-salida Estándar..... | 46 |
| Redirecciones..... | 46 |
| Tuberías de comunicación..... | 48 |
| Salida de errores. | 49 |
| Tema 6: Historial de comandos..... | 50 |
| ¿Qué es un historial?..... | 50 |
| Tema 7: Procesos en el sistema..... | 51 |
| Propiedades de un proceso..... | 51 |



| | |
|---|----|
| Estado de un proceso..... | 51 |
| Prioridad de un proceso..... | 52 |
| Gestión de procesos..... | 52 |
| Comando ps..... | 52 |
| Comando top..... | 53 |
| Comando kill..... | 53 |
| Comando bg..... | 55 |
| Comando fg..... | 55 |
| Comunicación entre procesos..... | 55 |
| Tema 8: Tareas comunes en el SHELL..... | 58 |
| Paginación de archivos..... | 58 |
| El comando more..... | 58 |
| El comando less..... | 59 |
| Manipulación de archivos y directorios..... | 60 |
| Comando cp..... | 60 |
| Comando mv..... | 61 |
| Comando rm..... | 62 |
| Comando touch..... | 63 |
| Comando cat..... | 64 |
| Comando ls..... | 64 |
| Comando cd..... | 66 |
| Comando mkdir..... | 66 |
| Monitorización del sistema..... | 67 |
| Comando top..... | 67 |
| Comando uname | 69 |
| Comando uptime..... | 70 |
| Comando time..... | 70 |
| Comando vmstat..... | 71 |
| Comando free:..... | 73 |
| Comando df:..... | 73 |



| | |
|--|----|
| Comando du..... | 74 |
| Tema 9: Programación en SHELL..... | 75 |
| Descripción de elementos de programación..... | 75 |
| Variables..... | 75 |
| Línea de comandos..... | 75 |
| Parámetros..... | 75 |
| La salida de los programas..... | 76 |
| Operación Aritmética..... | 76 |
| Manejo de parámetros..... | 76 |
| Manejo de variables..... | 77 |
| Operaciones aritméticas..... | 79 |
| Interactuando con archivos..... | 80 |
| UNIDAD IV: Gestión de usuarios y grupos..... | 82 |
| Tema 1: Gestión de usuarios. | 82 |
| Creación de cuentas de usuario. | 86 |
| Crear un usuario manualmente..... | 86 |
| Modificación de cuentas de usuarios..... | 87 |
| Comando usermod..... | 87 |
| Eliminación de cuentas de usuario..... | 89 |
| Consulta de información de cuentas de usuario..... | 90 |
| Bases de datos usuarios y passwords..... | 91 |
| Tema 2: Gestión de grupos de usuarios. | 94 |
| Creación de grupos de usuarios..... | 94 |
| Addgroup..... | 94 |
| Modificación de grupos de usuarios..... | 94 |
| Comando newgrp..... | 94 |
| Comando chgrp..... | 95 |
| Eliminación de grupos de usuarios..... | 95 |
| Consulta de información de grupos..... | 96 |
| Directorios personales..... | 97 |



| | |
|---|-----|
| Tema 3: Administrador del sistema..... | 98 |
| Características del administrador. | 98 |
| Suplantación de identidad..... | 100 |
| Tema 4: Grupos y usuarios especiales en el sistema..... | 104 |
| Usuarios especiales preexistentes..... | 104 |
| Grupos especiales preexistentes..... | 105 |
| UNIDAD V: Gestión de almacenamiento y sistema de archivos..... | 106 |
| Tema 1: Dispositivos de almacenamiento en sistemas GNU/Linux..... | 106 |
| Dispositivos IDE en Linux..... | 107 |
| Dispositivos SCSI y SATA en Linux..... | 108 |
| Tema 2: Particiones de disco..... | 109 |
| Tabla de particiones..... | 109 |
| Tipos de particiones..... | 110 |
| Particiones primarias y lógicas..... | 110 |
| Creación de particiones..... | 111 |
| Tema 3: Manejo de volúmenes lógicos..... | 113 |
| Conceptualización y arquitectura del esquema de almacenamiento..... | 113 |
| Volúmenes físicos..... | 113 |
| Grupos de volúmenes..... | 115 |
| Volúmenes lógicos..... | 115 |
| Operaciones comunes con volúmenes lógicos. | 116 |
| Respaldo con volúmenes lógicos..... | 117 |
| Copia instantánea de volúmenes..... | 117 |
| Tema 4: Sistemas de archivos. | 119 |
| Descripción de sistemas de archivos comunes. | 119 |
| Creación de sistemas de archivos. | 121 |
| Creación de una partición | 121 |
| Manipulación de sistemas de archivos. | 124 |
| Redimensionamiento de una partición..... | 124 |
| Montaje de sistemas de archivos. | 125 |



| | |
|---|-----|
| Archivo /etc/fstab..... | 128 |
| Tema 5: Cuotas de disco. | 130 |
| Activación de cuotas en un punto de montaje..... | 131 |
| Manipulación de cuotas. | 134 |
| Edquota..... | 136 |
| Cuota absoluta..... | 137 |
| Cuota de gracia. | 137 |
| Aplicando cuotas masivamente..... | 138 |
| Comprobaciones..... | 139 |
| ¿Qué es una cuota de disco?..... | 141 |
| Tema 6. Permisos sobre el sistema de archivos..... | 142 |
| Umask..... | 142 |
| Permisos básicos..... | 143 |
| Notación de las permisologías en sistemas tipo GNU/Linux..... | 144 |
| Notación simbólica..... | 144 |
| Notación octal..... | 145 |
| Suid (o bit setuid)..... | 146 |
| Gid (o bit setgid)..... | 147 |
| Bit pegajoso..... | 147 |
| Chmod..... | 148 |
| Uso de chmod..... | 148 |
| Opciones de chmod | 148 |
| Modos en chmod..... | 149 |
| Ejemplos..... | 150 |
| Permisos basados en listas de control de acceso ACL..... | 152 |
| UNIDAD VI: Fundamentos de Redes TCP/IP en GNU/Linux..... | 153 |
| Tema 1: Configuración de interfaces de red..... | 153 |
| Generalidades..... | 153 |
| Comando ifconfig..... | 154 |
| Configuraciones estáticas al inicio del sistema..... | 158 |



| | |
|--|-----|
| Configuraciones adicionales para interfaces wifi..... | 159 |
| Configuraciones automáticas al inicio del sistema..... | 160 |
| Múltiples interfaces de red..... | 160 |
| Interfaces virtuales..... | 161 |
| Tema 2: Integración del sistema en un entorno de red..... | 163 |
| Utilizando el servicio DNS..... | 163 |
| Tema 3: SSH..... | 164 |
| Iniciando ssh..... | 164 |
| Servidor SSH..... | 164 |
| Mecanismos de autenticación y opciones de configuración..... | 165 |
| Cliente SSH..... | 166 |
| El comando SCP..... | 167 |
| Conexiones SSH reversa..... | 168 |
| Transferencia de archivos con SFTP..... | 169 |
| Tema 4: Servicio VNC..... | 171 |
| UNIDAD VII: Instalación de paquetes de software..... | 174 |
| Tema 1: Sistema de empaquetado APT..... | 174 |
| Uso e instalación de paquetes..... | 174 |
| Almacén de paquetes..... | 176 |
| Tema 2: Configurando el sistema APT..... | 178 |
| Definiendo repositorios y versiones..... | 178 |
| Configurando el comportamiento del sistema APT..... | 179 |
| Tema 3: Servicio de proxy/cache APT..... | 181 |
| Apt-cacher-ng..... | 181 |
| Configuración del servicio..... | 182 |
| Configuración del cliente..... | 183 |
| UNIDAD VIII: Servicios de impresión con CUPS..... | 185 |
| Tema 1: Introducción a CUPS..... | 185 |
| Funcionamiento..... | 185 |
| Instalación a través del sistema de empaquetado..... | 186 |



| | |
|--|-----|
| Elementos de configuración del servicio..... | 186 |
| Tema 2: Interfaz Web de administración..... | 188 |
| Gestión de Impresoras..... | 193 |
| Administración de colas..... | 195 |
| Tema 3: Integración con servidores SMB/CIFS. (Samba)..... | 197 |
| UNIDAD IX: Servicio de almacenamiento remoto/compartido con NFS..... | 198 |
| Tema 1: Sistema de archivos de red NFS..... | 198 |
| Funcionamiento..... | 198 |
| Ventajas y Desventajas..... | 199 |
| Ventajas..... | 199 |
| Desventajas..... | 199 |
| Tema 2: Implementando un servidor NFS..... | 200 |
| El Servidor..... | 201 |
| Tema 3: Utilizando NFS a través del cliente integrado..... | 205 |
| UNIDAD X: Servicio SMB/CIFS con Samba..... | 207 |
| Tema 1: Introducción a las redes basadas en SMB/CIFS..... | 207 |
| Funcionalidades y Virtudes..... | 207 |
| Tema 2: Implementando un servidor Samba..... | 208 |
| Instalación del servidor OpenLDAP..... | 209 |
| Instalación de Herramientas y Librerías Adicionales..... | 209 |
| Manipulación de Archivos de Configuración..... | 210 |
| Smb.conf..... | 211 |
| smbldap.conf..... | 216 |
| smbldap_bind.conf..... | 220 |
| Archivo slapd.conf..... | 221 |
| Parámetros Globales..... | 221 |
| Definición de etiquetas..... | 222 |
| Archivo ldap.conf..... | 227 |
| Archivo nsswitch.conf..... | 228 |
| Archivos pam-ldap.conf y libnss-ldap.conf..... | 229 |



| | |
|---|-----|
| Tema 3: Administración del Controlador de Dominio..... | 231 |
| Labores Comunes de Administración..... | 231 |
| Creación, Modificación y/o Eliminación de Cuentas de Usuario..... | 231 |
| Creación, Modificación y/o Eliminación de Grupos de Usuarios..... | 233 |
| Creación, Modificación y/o Eliminación de Cuentas de Maquinas del Dominio..... | 235 |
| UNIDAD XI: Interactuando con el Kernel LINUX..... | 236 |
| Tema 1: Definición de kernel..... | 236 |
| Tipo de kernel | 236 |
| Versionado del kernel..... | 237 |
| Núcleos precompilados..... | 239 |
| Tema 2: Obteniendo un nuevo kernel..... | 240 |
| Obteniendo las fuentes de un kernel estándar..... | 240 |
| Obteniendo las fuentes de un kernel Canaima GNU/Linux..... | 241 |
| Tema 3: Configurando el nuevo kernel..... | 242 |
| Proceso de configuración..... | 243 |
| Tema 4: Instalando el nuevo kernel | 245 |
| UNIDAD XII: Introducción a la administración de servicios basados en Canaima GNU/Linux. | |
| | 246 |
| Tema 1: Servicios de correo electrónico..... | 246 |
| Tema 2: Sistema de Resolución de Nombres (DNS)..... | 247 |
| Elementos de un Sistema de nombres de dominio..... | 247 |
| Tema 3: Servicios de Directorio basados en LDAP..... | 249 |
| Atributos LDAP | 249 |
| Tema 4: Respaldo y Recuperación..... | 251 |
| Elementos para mantener la información segura. | 251 |
| Tema 5: Seguridad de la Información..... | 253 |
| Elementos de seguridad lógica..... | 253 |
| Elementos de la seguridad física..... | 254 |
| Tema 6: Redes privadas virtuales..... | 255 |
| Usos comunes de las Redes Privadas Virtuales:..... | 255 |



| | |
|--|-----|
| UNIDAD XIII: Apéndice I. Editor de archivos VIM..... | 257 |
| Tema 1: Introducción a VIM..... | 257 |
| Tema 2: La tecla ESC..... | 259 |
| Tema 3: Algunas consideraciones sobre el texto..... | 260 |
| Borrando texto..... | 262 |
| Modo edición..... | 262 |
| Tema 4: Otros comandos útiles..... | 265 |
| Repeticiones de comandos..... | 266 |
| Ejemplos:..... | 266 |
| UNIDAD XIV: Apéndice II. Sistema X.org..... | 267 |
| Tema 1: El sistema X.Org..... | 267 |
| Tema 2: X-Windows..... | 268 |
| Tema 3: Modos VESA..... | 269 |
| Tema 4: Reconfigurar servidor gráfico X.org..... | 270 |
| Tema 5: Las secciones de xorg.conf..... | 271 |
| Sección “Modules”..... | 271 |
| Sección “ServerFlags”..... | 271 |
| Sección “Monitor”..... | 272 |
| Sección “Device”..... | 273 |
| Sección “Screen”..... | 274 |
| Sección “Input Device”..... | 275 |
| Sección “Files”..... | 276 |
| Tema 6: Sesiones..... | 277 |
| Inicio de Sesiones desde Terminales..... | 277 |
| Inicio de sesiones a través de la red..... | 278 |
| Anexos:..... | 279 |
| Ejercicio Propuesto N#1: | 279 |
| Solución del ejercicio..... | 280 |
| Ejercicio Propuesto N#2: | 287 |
| Referencias..... | 288 |



Ficha descriptiva

| | | | |
|---------------------------|--|-----------------|-------------|
| Curso | Manual de Administración de entorno y servicios en Canaima GNU/Linux. | | |
| Modalidad | A distancia. | Duración | 10 semanas. |
| Dirigido a | Público y comunidad en general, así como personal docente, técnico y estudiantil de Colegios Universitarios y Politécnicos. | | |
| Requisitos previos | <p>Nociones básicas en el manejo de:</p> <ul style="list-style-type: none"> • Permisos y ACL POSIX. • Redes en GNU/Linux. • Gestión de usuarios y permisos bajo Linux. • Manejo de servicios SysV. • Gestión de procesos POSIX. • Herramientas de paginación y visualización de texto. • Conocimiento en respaldo GNU/Linux. • Conocimiento en LDAP GNU/Linux. • Conocimiento en DNS GNU/Linux. • Conocimiento en VPN GNU/Linux. | | |
| Objetivo del curso | Desarrollo de destrezas para el manejo y administración de sistemas y servicios en entornos basados en el sistema operativo Canaima GNU/Linux. | | |



UNIDAD I: INTRODUCCIÓN A CANAIMA GNU/LINUX

Tema 1: Sistema operativo GNU/Linux

GNU/Linux es un poderoso y sumamente versátil sistema operativo con licencia libre y que implementa el estándar POSIX (acrónimo de: *Portable Operating System Interface*, que se traduce como Interfaz de Sistema Operativo Portátil). Fue inicialmente creado en 1991 por Linus Torvalds, siendo entonces un estudiante de la Universidad de Helsinki, Finlandia. En 1992, el núcleo (*kernel*) Linux (de la autoría original de Torvalds, ahora extendido por una multitud de desarrolladores a nivel mundial) fue combinado con el sistema GNU¹. El Sistema Operativo formado por esta combinación se conoce como GNU/Linux.

¿Qué es el *kernel*?

El *kernel* o núcleo del sistema operativo es el programa principal del sistema operativo que se encarga de la comunicación entre el hardware y el software, construyendo una capa de abstracción sencilla que facilita la interacción de los programas con la memoria, procesador y demás hardware del computador; Asimismo, el *kernel* se encarga de la comunicación, administración, creación y destrucción de procesos a bajo nivel.

1 <http://www.gnu.org/home.es.html>



Tema 2: Distribuciones GNU/Linux

El sistema operativo GNU/Linux, al ser software libre, carece de un solo ente que lo controle en un sentido “comercial”, lo que le otorga una flexibilidad inmensa que permite su rápida y fácil adaptación para entornos de trabajo específicos. En este sentido, Un sistema GNU/Linux se distribuye en forma de múltiples distribuciones, es decir, un conjunto de aplicaciones reunidas para permitir la instalación sencilla del sistema que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre, o como se dijo anteriormente, incorporar aplicaciones o controladores propietarios.

Existen también, las *meta-distribuciones*, estas son distribuciones que están a su vez basadas en otras distribuciones y su propósito principal ha sido mejorar la integración del sistema operativo a un grupo común de usuarios, un buen ejemplo de esto es una meta-distribución que ya incluya paquetes y herramientas para escritura y corrección ortográfica en un idioma específico mientras que en la distribución en la que esta se encuentra basada solo los provee de forma opcional.

Canaima GNU/Linux, mejor conocida como Canaima, es una de las múltiples meta-distribuciones que hoy en día se puede encontrar en la red trabajando con el núcleo Linux y un conjunto de las herramientas del proyecto GNU.

Componentes de software

Como se mencionó anteriormente, las distribuciones GNU/Linux están formadas por conjuntos diferentes de software, diferentes distribuciones ofrecen métodos diversos para la instalación, remoción y actualización de software. En el argot de las distribuciones, a los componentes de software se les llama “paquetes” ya que el software viene empaquetado con todas las instrucciones y procedimientos para su integración con



la distribución a la que está destinada.

Aunque existen una diversidad de administradores de paquetes, la mayoría de las distribuciones usan alguno de los dos mas populares, a saber:

- Administrador de paquetes RPM²
- Administrador de paquetes APT³

Administrador de paquetes RPM

El formato de paquetes utilizado por el administrador de paquetes RPM es el formato RPM, los archivos en este formato usualmente tienen la extensión *.rpm* y la herramienta principal con la que se maneja la instalación, desinstalación y actualización de software en las distribuciones que utilizan este formato de paquetes es el comando *rpm*, mediante el cual se hace la gestión de paquetes por medio de la consola. Asimismo, existen diversas herramientas gráficas fáciles de utilizar que cumplen el mismo fin y que utilizan, a bajo nivel las características proporcionadas por *rpm*.

² *Red Hat Package Manager*, Administrador de paquetes de Red Hat, or sus siglas en inglés.

³ *Herramienta avanzada de empaquetado*, por sus siglas en inglés.



Tema 3: Distribución Canaima GNU/Linux.

En términos generales, Canaima GNU/Linux es una distribución de software libre y estándares abiertos basada en el sistema de paquetes APT dirigida a usuarios finales venezolanos y desarrollada en concordancia con el marco legal vigente en Venezuela

Canaima, como ya se comentó, utiliza el sistema de paquetes APT, posiblemente el mecanismo de distribución gestionada de software de mayor difusión a nivel internacional. Es utilizado por distribuciones de impacto global como Debian y Ubuntu, y es la base de centenares de distribuciones incluyendo algunas utilizadas por OEMs en equipos portátiles y de escritorio, así como para servidores.

En ese sentido, una de las partes más importantes de Canaima es su repositorio, que cuenta con tres (3) ramas con niveles de servicio diferenciados y la posibilidad de sincronizar sus paquetes de software con los repositorios de la rama de pruebas de Debian. Así mismo, es posible incluir nuevos paquetes de software en los repositorios en un momento dado. Los repositorios son autocontenidos.

Canaima se distribuye en distintos medios, que incluyen el repositorio, el instalador en formato DVD para arquitecturas i386, amd64 y powerpc, el LiveDVD para múltiples arquitecturas⁴ con su instalador integrado y el instalador para dispositivos USB.

Historia y motivación

Canaima es la distribución GNU/Linux venezolana basada en Debian que surge como una solución para cubrir las necesidades informáticas de los usuarios finales de la Administración Pública Nacional (APN) Venezolana y para dar cumplimiento al decreto presidencial N° 3.390 sobre el uso de tecnologías libres en la APN.

4 Este LiveDVD multi-arquitectura está disponible solamente para las arquitecturas i386 y amd64



Canaima es una de las distribuciones GNU/Linux más usadas en Venezuela a raíz de su incorporación en escuelas públicas, siendo utilizada en proyectos de gran escala como "Canaima", proyecto que busca dotar a más de 1.250 mil niños y niñas en edad escolar con computadores Canaima, y su caso de uso ha sido presentado en Congresos Internacionales sobre el uso de estándares abiertos, a pesar de que su reciente desarrollo ha sido utilizada en el flisol donde ha sido instalada en los equipos de muchos usuarios.

Características diferenciales

Su base de software es pequeña y sustentable, sobre todo cuando se le compara con otros proyectos como Debian o Ubuntu, contando con menos de mil quinientos (1500) paquetes binarios de software. A la fecha se mantiene como premisa la compatibilidad binaria con, al menos, Debian y Ubuntu.

Escenarios de aplicación

Canaima, aunque fue primero pensada como una plataforma de escritorio libre en concordancia de las necesidades de la administración pública nacional Venezolana, desde su misma inserción ha crecido constantemente para ser aplicada en diferentes ámbitos tecnológicos dentro de las instituciones y hogares Venezolanos, en tal sentido, Canaima es tanto una distribución orientada al escritorio, como una distribución orientada a su uso como plataforma de servidor, soportando para ello las populares arquitecturas de hardware donde estos funcionan, así como en plataformas de hardware para servidores de muy alto rendimiento y escalabilidad.



Tema 4: Plataformas colaborativas.

Sitios Web oficiales.

Sitios oficiales de la distribución Canaima GNU/Linux:

<http://canaima.softwarelibre.gob.ve/cms/>

<http://wiki.canaima.softwarelibre.gob.ve/>

<http://www.cnti.gob.ve/>

Elementos colaborativos de Canaima GNU/Linux.

Foro

En el encontrarás diversos foros de mensajes, de opinión o de discusión que sirven de ayuda y de soporte a apreciaciones, valoraciones y/o aportes en línea. La dirección es: http://canaima.softwarelibre.gob.ve:8080/canaima_cms/mensajes/en-construccion.

Wiki

Si estás en la búsqueda de documentación referida a procesos de instalación, configuración y uso de GNU/Linux Canaima, así como de los paquetes adicionales que están disponibles, puedes consultar nuestra documentación, donde la comunidad y los organismos participantes en su desarrollo crean, editan y refinan la información de manera comprensible para el usuario. La dirección es la siguiente:

<http://canaima.softwarelibre.gob.ve/wiki/index.php/Portada>



Listas de correo

El correo electrónico es una de los medios más usados para discutir diversos temas. Las listas de correo electrónico son un uso especial del correo electrónico que permite la distribución masiva de información entre múltiples usuarios de Internet a la misma vez, es por ello que la mayoría de las conversaciones entre desarrolladores y usuarios de GNU/Linux Canaima se llevan a cabo a través de las siguientes listas:

- Lista para información general sobre GNU/Linux Canaima: discusion@canaima.softwarelibre.gob.ve
- Lista para aclarar inquietudes y resolver problemas con GNU/Linux Canaima: soporte@canaima.softwarelibre.gob.ve
- Lista para discutir fallas, problemas o hacer la solicitud de nuevas funcionalidades en la plataforma colaborativa de GNU/Linux Canaima y sus servicios servicios@canaima.softwarelibre.gob.ve

Sistema de manejo de versiones

Subversión es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversión es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.



Sistema de construcción de paquetes.

Canaima es una distribución de propósito general, por lo que no ha sido diseñada para cubrir las necesidades de cada una de las personas u organizaciones que deseen hacer uso de este sistema operativo en sus plataformas tecnológicas.

Es por ello que en casi todos los casos los usuarios de Canaima querrán derivar sus propias distribuciones, versiones o *sabores* a partir de Canaima GNU/Linux con la finalidad de cumplir con un objetivo de negocios particular.

El sistema de construcción de paquetes en Canaima se realizó a través de los siguientes pasos:

- Se agregan los paquetes binarios de software en el instalador
- Remover paquetes binarios de software del instalador
- Agregar, remover o modificar la preconfiguración de Debconf en el instalador
- Agregar, remover o modificar la postconfiguración en el instalador
- Cambiar la preconfiguración del manejador de ventanas GNOME
- Cambiar el estilo visual
- Cambiar el perfil predeterminado de nuevos usuarios
- Crear un repositorio parcial para uso institucional
- Agregar o cambiar llaves PGP para el sistema de paquetes
- Importar paquetes binarios en formato RPM de otras distribuciones

Todos estos pasos se pueden encontrar explicados de manera mucho más detallada en el instructivo que está en:

<http://canaima.proyectos.onuva.com/descargas/canaima-manual/06/Manual-Canaima.odt>



UNIDAD II: INSTALACIÓN DE GNU/LINUX - DISTRIBUCIÓN CANAIMA.

Tema 1: Consideraciones previas a las instalación.

A continuación se describen los pasos a seguir durante el proceso de instalación de cualquier distribución GNU/Linux:

- Realizar una copia de seguridad de los datos o documentación existente en el disco duro donde se planea realizar la instalación.
- Reunir información sobre el sistema, así como toda la documentación que se necesite antes de iniciar la instalación.
- Crear un espacio particionable en el disco duro para la instalación del sistema operativo, de ser necesario.
- Localizar y/o descargar el programa instalador, así como los archivos de cualquier controlador especializado que la computadora donde se va a instalar el sistema necesite.
- Instalar los archivos de arranque (la mayoría de los usuarios de CD pueden arrancar desde uno de éstos).
- Arrancar el sistema de instalación.
- Elegir el idioma para la instalación.
- Activar la conexión de red, si está disponible.
- Crear y montar las particiones en las que se instalará el sistema operativo.
- Esperar a la descarga/instalación/configuración automática del sistema base.
- Instalar el gestor de arranque.



Tema 2: Obteniendo información del hardware.

En la mayoría de los casos, el instalador detecta automáticamente el hardware del computador donde se instala el sistema. Sin embargo, es posible que esto no suceda, si es este el caso, se debe estar preparado. Por lo tanto, se recomienda estar familiarizado con el hardware de la máquina antes de la instalación. En este sentido, se debe obtener la información del hardware de la computadora, para esto se pueden utilizar:

- Los manuales que vienen con cada pieza de hardware.
- Las pantallas de configuración de la BIOS del computador. Estas pueden verse cuando se enciende la máquina y se presiona una combinación de teclas (verificar el manual para saber la combinación, la mayoría de las veces se utiliza la tecla Supr).
- Las cajas y cubiertas de cada pieza de hardware.
- Órdenes del sistema o herramientas de otros sistemas operativos, incluyendo las capturas de pantallas de los gestores de archivos. Esta fuente de información es especialmente útil para obtener información sobre la memoria RAM y el espacio disponible en el disco duro.
- El administrador de sistemas o proveedor de servicio de Internet puede ofrecer información necesaria para configurar la red y el correo electrónico, esto si se sistema está conectado a alguna red durante todo el día. Por ejemplo, si utiliza una conexión Ethernet o equivalente, pero no si tiene una conexión PPP (Protocolo Punto a Punto).



Tema 3: Medios de instalación.

En esta sección se puede determinar los diferentes tipos de medios que se usan para instalar el sistema operativo GNU/Linux.

- **CD-ROM/DVD-ROM:** existe soporte para la instalación basada en CD-ROM para algunas arquitecturas o para propósitos de recuperación del sistema.
- **Dispositivo de memoria USB:** son utilizados para gestionar (instalar y cuando sea necesario recuperar el sistema) servidores y en los casos de sistemas pequeños que no tienen espacio para unidades innecesarias.
- **Red:** se utiliza durante la instalación para recuperar archivos. El que se utilice la red o no, depende del mecanismo de instalación que se escoja y de las respuestas dadas a algunas preguntas que se realizarán durante la instalación. Este sistema de instalación puede utilizar la mayor parte de las conexiones de red a través tanto de HTTP como FTP. También se puede arrancar el sistema de instalación a través de la red.



Tema 4: Requisitos mínimos de hardware.

Requisitos de Memoria y Espacio en Disco Duro

En ciertas ocasiones nos puede interesar conocer cuáles son los requisitos de hardware que necesitan una u otra distro para su instalación. Usualmente estos datos varían mucho entre distribuciones, por lo que se ha recopilado la información y con ella se ha creado una tabla comparativa que permite decidir cuál distribución funcionará mejor en la computadora donde se desea instalar el sistema.

En nuestro caso nos interesa las distribuciones Debian y Canaima, por lo tanto, la información es la siguiente:

Debian 4.0:

- **Procesador:** Intel Pentium 1-4, AMD Duron, Celeron, Athlon, Sempron u Opteron.
- **RAM:** Mínimo 16 MB para modo texto, 64 MB interfaz gráfica / Recomendado: 128 MB.
- **Espacio en Disco Duro:** Mínimo 450 MB / Recomendado 4 GB.

Debian 5.0:

- **Procesador:** Intel Pentium 1-4, AMD Duron, Celeron, Athlon, Sempron u Opteron.
- **RAM:** Mínimo 32 MB para modo texto, 194 MB interfaz gráfica Recomendado: 256 MB.
- **Espacio en Disco Duro:** Mínimo 500 MB / Recomendado 3 GB.

Canaima:

- **Procesador:** Basado en Intel x86 i386, mínimo Pentium III.



- RAM: Mínimo 64 MB / Recomendado 512 MB.
- Espacio en Disco Duro: Mínimo 5 GB.

Dispositivos de red

Casi cualquier tarjeta de interfaz de red (NIC) soportada por el núcleo Linux debería también ser soportada por el sistema de instalación, los controladores modulares deberían ser cargados automáticamente. Esto incluye la mayoría de tarjetas PCI y PCMCIA. Muchas tarjetas ISA antiguas son soportadas también.



Tema 5: Nomenclatura para discos y particiones

En el diseño tradicional UNIX, todo es un fichero y los discos se nombran mediante su fichero de dispositivo: IDE, SCSI y USB.

- IDE : /dev/hda Disco Maestro en canal IDE 0, /dev/hdb Disco Esclavo en canal IDE 0, /dev/hdc Disco Maestro en canal IDE 1, /dev/hdb Disco Esclavo en canal IDE 1.
- SCSI y USB: /dev/sda, /dev/sdb, entre otros.

Las particiones de un disco se nombran mediante el nombre de dispositivo y el número de partición:

- Primarias: /dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4.
- Lógicas: /dev/hda5 en adelante.



Tema 6: Esquema de Particiones

El particionamiento es la creación de divisiones lógicas en un disco duro que permite aplicar el formato lógico de un sistema operativo específico. Cada partición aparece ante el sistema como si fuese un disco independiente.

Un disco duro puede tener un máximo de 4 particiones primarias, porque la información de la tabla de particiones reside (junto con el código de arranque) en el MASTER BOOT RECORD (MBR): el sector 0 del disco. Sin embargo, una de las particiones primarias puede ser designada como partición extendida y ser subdividida en un número ilimitado de particiones lógicas.

GNU/Linux puede ser instalado en cualquier tipo de partición y suele numerar las particiones primarias de un disco desde 1 a 4 reservando los números 5 y superior para las particiones lógicas.

Es usual que en los sistemas GNU/Linux se creen hasta 3 particiones: la principal representado por el símbolo / la cual contiene todo el software del Sistema Operativo, una segunda para el directorio home que contiene las configuraciones de usuario y una tercera llamada swap para la memoria virtual temporal que es utilizada en casos de sobrecarga de trabajo, esto para un esquema simple y efectivo. Si el usuario es avanzado puede necesitar particiones separadas para aplicaciones, archivos temporales, entre otros. Por ejemplo: /usr para el directorio de aplicaciones, /var para el directorio de logs y otros archivos de tamaño variable, /tmp para directorio de archivos temporales y /opt para directorio de software comercial específico.

Estructura de directorios en Canaima GNU/Linux.

La estructura de directorios se distribuye de la siguiente manera:



| Directorio | Descripción |
|------------|---|
| / | La raíz, que contiene los directorios principales |
| /bin | Contiene archivos ejecutables fundamentales del sistema, utilizados por todos los usuarios (como por ejemplo los comandos ls, rm, cp, chmod, mount, etc.). |
| /boot | Contiene los archivos que permiten que Linux se inicie |
| /dev | Contiene los puntos de entrada para los periféricos |
| /etc | Contiene los comandos y los archivos que el administrador del sistema necesita (archivos passwd , group , inittab , ld.so.conf , lilo.conf , etc.) |
| /home | Directorio personal del usuario |
| /lib | Contiene bibliotecas compartidas que son fundamentales para el sistema durante su inicio |
| /mnt | Contiene puntos de montaje de particiones temporales (CD-ROM, disquete, etc.) |
| /opt | Contiene paquetes de aplicaciones suplementarias |
| /root | Directorio del administrador de raíz |
| /sbin | Contiene los sistemas binarios fundamentales (por ejemplo, el comando adduser) |
| /tmp | Contiene archivos temporales |
| /usr | Jerarquía secundaria |
| /var | Contiene datos variables |



Esquema básico.

El mejor momento para el particionado es durante la instalación, al momento de realizar esta deberá seleccionar la siguiente opción para obtener un particionado básico:

Guiado - Utilizar todo el disco: recomendado para novatos, nos propone de forma automática y utilizando todo el disco el particionamiento apropiado, al seleccionar esta opción tendremos otro menú para escoger entre:

- Todos los ficheros en una partición: creando solo 2 particiones, una para el área de intercambio o swap y la otra para el sistema de ficheros de raíz o barra (/) de donde se crean los demás directorios y ficheros del sistema.
- Separar la partición /home: esta opción nos permite separar la partición /home de la partición barra (/), esto tiene la ventaja de que los directorios y archivos de los usuarios quedan separados en esta partición y a la hora de que por algún motivo tengamos que rehacer el sistema los datos de los usuarios quedarán en una partición aparte y no tendremos necesidad de formatearla ya que únicamente trabajaremos con la partición barra (/).

Esquema avanzado.

Consiste básicamente en separar las particiones /home, /usr, /var, /tmp y /opt para obtener una distribución y manejo del espacio mucho más eficiente.

Las opciones de particionado avanzado que se muestran durante la instalación son las siguientes:

Guiado - utilizar el disco completo y configura LVM: al igual que el anterior propone, de forma automática y utilizando todo el disco, el particionamiento adecuado y, además, permite configurar LVM (Logical Volume Management). LVM permite agrupar



discos físicos en grupos virtuales de discos y posteriormente crear particiones o volúmenes lógicos.

Guiado - utilizar el disco completo y configura LVM cifrado: igual que el anterior y, además, cifrando los datos.

Manual: Particionamiento completamente manual, es recomendable para usuarios avanzados. Deberemos crear todas las unidades necesarias (/ , swap, ext3, etc) manualmente.



Tema 7: Gestor de Arranque

Un gestor de arranque es un programa que se carga en el momento de arrancar el computador y permite elegir qué sistema operativo, de entre los que haya instalados en el disco duro, se quiere iniciar.

Conceptualmente todos los gestores funcionan de la siguiente manera: primero la BIOS del computador debe leer el código de arranque del MBR (sector 0 del disco). Para ello se debe configurar la BIOS para que pueda arrancar del disco que se quiere. La BIOS solo sabe arrancar el programa que se encuentra en el MBR, dicho programa es el gestor de arranque, en su primera etapa y a su vez sabe a qué particiones tiene que ir a leer para continuar con la carga de la siguiente etapa, y de ahí ofrecer un menú para que el usuario seleccione uno u otro sistema operativo.

Uno de los gestores más flexibles y el que se ha convertido en estándar es GRUB (Grand Unified Bootloader) el cual es un gestor de arranque múltiple que se usa comúnmente para iniciar dos o más sistemas operativos instalados en un mismo computador. GRUB viene preinstalado en la mayoría de las distribuciones de GNU/Linux modernas, entre ellas Debian y sus derivadas.



Tema 8: Proceso de Instalación de Canaima GNU/Linux

Preparando su sistema para la instalación.

Antes de empezar a instalar Canaima GNU/Linux tome en cuenta las siguientes previsiones:

- Respalde toda la información sensible de su computadora en un medio de almacenamiento seguro. Recomendamos utilizar distintos tipos de medios de almacenamiento (CD's, DVD's, memorias Flash, discos duros externos)
- Si desea conservar otro sistema operativo en el mismo disco duro de su computadora, debe preparar un esquema de particionado para aplicarlo con el Instalador de Canaima GNU/Linux.
- Es recomendable, pero no necesario, instalar Canaima GNU/Linux con una conexión no restringida a Internet.
- El Equipo de Desarrollo de Canaima GNU/Linux no presta soporte a instalaciones en máquinas virtuales.
- La instalación y uso de Canaima GNU/Linux se hace a su propio riesgo, y el producto se distribuye como está, sin ningún tipo de garantías.
- Su computadora debe contar con una unidad lectora de CD o DVD interna o externa para poder instalar Canaima GNU/Linux.
- Es recomendable contar con al menos 384 MB. de memoria RAM para instalar y utilizar Canaima GNU/Linux. de forma satisfactoria.



- Es recomendable contar con al menos 5 GB. de espacio en el disco duro de su computadora para instalar y utilizar Canaima GNU/Linux. de forma satisfactoria.

Iniciando la instalación

Para iniciar la instalación de Canaima GNU/Linux, introduzca el CD o DVD de instalación en su unidad de CD o DVD y reinicie su equipo con el disco introducido en la unidad. La configuración de su computadora puede requerir que modifique la BIOS o presione alguna tecla para poder iniciar el sistema desde el CD o DVD.

En breves instantes aparecerá la pantalla de bienvenida del Instalador de Canaima GNU/Linux y podrá presionar ENTER o esperar diez (10) segundos para que inicie la instalación. También podrá iniciar la instalación en los siguientes modos especiales:

- 1.** Modo a prueba de fallos: es una versión del Instalador que no utiliza elementos gráficos y está diseñada para funcionar incluso en computadoras con tarjetas gráficas que no puedan iniciar el instalador tradicional. Se comporta de igual forma que el instalador gráfico.
- 2.** Modo experto: es una versión del Instalador que hace más preguntas para personalizar aún más la experiencia de instalación; sin embargo, podrá encontrar preguntas no documentadas en este manual.
- 3.** Modo de rescate: es una versión del Instalador diseñada para acceder temporalmente a un sistema instalado en el disco duro pero que por alguna razón no puede ser iniciado.



Durante la instalación.

El instalador de Canaima GNU/Linux le hará algunas preguntas sobre el sistema que está instalando. El Equipo de Desarrollo de Canaima GNU/Linux ha preparado la instalación para que sea lo más sencilla posible. Por favor, preste atención a las preguntas que le hace el instalador para que su sistema esté correctamente ajustado:

1. Mapa de teclado: seleccione el mapa de teclado que utilizará el sistema operativo. En Venezuela, usualmente encontrará teclados con mapa de teclado Español, Latinoamericano o Inglés estadounidense.
2. Configuración de la red: si no está conectado a una red o la red a la que está conectado no dispone de autoconfiguración con DHCP⁵, el instalador le preguntará algunos datos sobre la red. Vea el capítulo correspondiente.
3. Particionamiento: el instalador siempre le preguntará como desea particionar su sistema. Vea el capítulo correspondiente.
4. Datos del administrador: el instalador le preguntará la contraseña del administrador dos (2) veces, para confirmar. El nombre del usuario administrador es root.
5. Datos del usuario: el instalador creará un usuario no privilegiado por usted. Debe introducir su nombre completo, un nombre corto de usuario y la contraseña del usuario dos (2) veces, para confirmar.

Respondiendo a estas cinco (5) preguntas, podrá tener su sistema Canaima GNU/Linux instalado y listo para funcionar.

5 *Protocolo de configuración automática de máquina*, por sus siglas en inglés



Configuración de la red .

Si su red soporta autoconfiguración con DHCP, el instalador no le hará ninguna pregunta sobre la configuración de la red. Si no está conectado a ninguna red o si su red no soporta el protocolo anteriormente citado, el instalador le preguntará lo siguiente:

- Nombre de equipo: introduzca un nombre corto para su computadora.
- Nombre de dominio: introduzca el nombre de su dominio DNS; si no lo tiene, puede dejarlo en blanco.
- Dirección IP: introduzca una dirección IP válida para su computadora.
- Máscara de red: introduzca la máscara de red.
- Pasarela de enlace: introduzca la pasarela de enlace; en algunas ocasiones el instalador intentará autocalcularla.
- Servidores DNS: introduzca el o los servidores DNS de su red.

Estos valores pueden ser provistos por el administrador de su red. El Equipo de Desarrollo de Canaima GNU/Linux no puede proveer esta información. También puede optar por no configurar la red si no está conectado a ninguna red; para hacer esto seleccione la opción No configurar la red en este momento.

Particionamiento.

Su sistema Canaima GNU/Linux necesita al menos dos (2) particiones, o secciones de su disco duro, para poder funcionar. Una corresponde a la memoria virtual, conocida como memoria de intercambio o memoria swap, y otra al sistema operativo y datos personales.



Si usted no tiene otro sistema operativo instalado en su computadora y no tiene necesidades especiales de particionamiento, recomendamos que elija la opción de Particionado automático, Utilizar todo el disco, y Todos los ficheros en una partición ya que ésta es la opción más sencilla y directa para instalar Canaima GNU/Linux. De otra forma, seleccione Particionado manual y siga las instrucciones del instalador para particionar su disco.

Una vez seleccionado el esquema de particionamiento, el Instalador le preguntará si está seguro de aplicar los cambios y luego formateará las particiones seleccionadas.

Finalizando la instalación.

Una vez finalizados todos los pasos de la instalación, el disco (CD o DVD) será automáticamente expulsado de la unidad y su computadora se reiniciará automáticamente. Retire el disco de la unidad y permita que la computadora arranque normalmente. Al cabo de pocos segundos verá en pantalla el gestor de arranque que le muestra dos opciones de inicio, la predeterminada arrancará en cinco (5) segundos y una opción de rescate o modo "single-user".

Luego de unos instantes, su nuevo sistema operativo Canaima GNU/Linux arrancará y podrá iniciar sesión con el usuario y clave definidas durante la instalación de Canaima GNU/Linux.



UNIDAD III: Interpretador de comandos (SHELL).

Tema 1: El SHELL de Canaima GNU/Linux

El intérprete de comandos es la interfaz entre el usuario y el sistema operativo; por esta razón, se le da el nombre en inglés shell, que significa caparazón. Por lo tanto, la shell actúa como un intermediario entre el sistema operativo y el usuario gracias a líneas de comando que este último introduce. Su función es la de leer la línea de comandos, interpretar su significado, llevar a cabo el comando y después arrojar el resultado por medio de las salidas.

La shell es un archivo ejecutable que debe interpretar los comandos, transmitirlos al sistema y arrojar el resultado. Existen varios shells. La más común es sh (llamada Bourne shell), bash (Bourne again shell), csh (C Shell), Tcsh (Tenex C shell), ksh (Korn shell) y zsh (Zero shell). Generalmente, sus nombres coinciden con el nombre del ejecutable.



Cada usuario tiene una shell predeterminada, la cual se activará cuando se abra un indicador del comando. La shell predeterminada se especifica en el archivo de configuración `/etc/passwd` en el último campo de la línea que corresponde al usuario. Es posible cambiar de shell durante una sesión. Para esto, solo se debe ejecutar el archivo correspondiente. Por ejemplo: `/bin/bash`.

Tema 2: Presentación del SHELL - Indicador del sistema

La shell se inicia al leer su configuración completa (en un archivo del directorio `/etc/`) y después al leer la configuración propia del usuario (en un archivo oculto cuyo



nombre comienza con un punto y que se ubica en el directorio básico del usuario, es decir /home/user_name/.configuration_file). A continuación, aparece el siguiente indicador llamado prompt en inglés:

equipo:/directorio/actual\$

De manera predeterminada, para la mayoría de las shells, el indicador consiste en el nombre del equipo, seguido de dos puntos (:), el directorio actual y después un carácter que indica el tipo de usuario conectado. Si el carácter es \$ especifica un usuario normal, si es # especifica un usuario administrador, llamado root.

¿Qué es un prompt?

Prompt es el carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la espera de órdenes. Éste puede variar dependiendo del intérprete de comandos y suele ser configurable.

PROMPT de usuarios

El prompt de usuario, depende de root para tareas de administración del sistema y configuración de dispositivos. Tiene un catálogo de comandos a su disposición mucho más limitado que el que podríamos encontrar en un prompt de administrador. Solo tiene permitido el trabajo con ficheros propios o navegación entre directorios del sistema de archivos, sin derecho a modificarlos si no posee permisos sobre ellos. El prompt de usuario en Canaima se identifica por tener el símbolo \$ al final del directorio en donde nos encontremos:

usuario@nombre-computador:~\$



PROMPT del administrador

El usuario root, es el administrador del sistema. Este asigna permisos a otros usuarios, crea grupos, usuarios, etc. En resumen tiene todos los derechos de sobre el sistema operativo, es por esto que no es recomendable usar continuamente este usuario para el día a día del uso del sistema, ya que por un descuido o comando mal efectuado, podemos causar graves daños al sistema o perder información importante para nosotros. El prompt de root viene identificado con el símbolo **#** al final de la ruta donde nos encontremos, de la siguiente manera:

nombre-computador:/home/usuario#



Tema 3: Instrucciones al SHELL.

Algunos comandos básicos para el manejo de ficheros y directorios, crear y borrar directorios; listar, copiar, renombrar y borrar archivos son los siguientes:

| Comando | Descripción |
|---------|--|
| ls | Listar archivos y directorios |
| cp | Copiar archivos y directorios |
| pwd | Mostrar el nombre del directorio de trabajo actual |
| cd | Cambiar de directorio |
| sort | Ordenar ficheros |
| mkdir | Crear directorios |
| touch | Crear o actualizar ficheros |
| rm | Borrar archivos y/o directorios |
| rmdir | Borrar directorios vacíos |
| mv | Mover o renombrar archivos |
| more | Muestra ficheros página a página |
| less | Muestra Ficheros página a página |
| cat | Mostrar ficheros de forma continua |
| head | Ver el inicio de un archivo |



| | |
|------|--|
| tail | Ver las últimas líneas de un archivo |
| find | Buscar archivos |
| grep | Buscar el patrón pasado como argumento en uno o más archivos |
| wc | Calcular la cantidad de cadenas y palabras en archivos |
| ln | Crea enlace entre ficheros |



Tema 4: Entorno de funcionamiento del SHELL.

Variables de Entorno y Configuraciones

Las variables de entorno y configuraciones son aquellas que tienen un significado propio para la shell o algún otro programa. Ciertos programas leen el contenido de las variables de entorno para modificar su comportamiento, entre ellos la propia shell. Entre las variables de entorno más importantes se pueden citar:

- PATH, indica la ruta de búsqueda de programas ejecutables. Está constituida por una lista de directorios separados por dos puntos (:). El directorio actual, de forma predeterminada, no viene incluida en PATH.
- PS1, especifica el indicador del sistema. Lo habitual es que PS1 sea el símbolo \$ para usuarios normales y # para usuario root.
- PS2, especifica el indicador secundario del sistema. Aparece cuando no se ha completado una orden. LANG, especifica el lenguaje que se aplica al usuario; para español se utiliza es.
- LC_ALL, contiene el idioma y se utiliza para usar los valores locales como mensajes del sistema, símbolo monetario, formato de fecha, formato de números decimales y otras características.
- TERM, almacena el tipo de terminal desde el que se está trabajando.
- EDITOR, especifica el editor por omisión del sistema. Lo habitual en los sistema Unix es que el editor por omisión sea vi.
- DISPLAY, especifica qué equipo muestra la salida que se efectúa en modo gráfico. Ese equipo deberá tener un servidor gráfico.
- LD_LIBRARY_PATH, se utiliza para definir rutas alternativas de búsqueda para bibliotecas de funciones del sistema.
- PWD, contiene el directorio de trabajo efectivo.


- Last, información sobre los últimos usuarios que han usado el sistema.

Con la orden `env` se puede comprobar el valor de las variables de entorno del sistema. Para modificarlas basta asignarle un nuevo valor.


Variables de entorno

Las variables de entorno tienen la funcionalidad de configurar ciertos aspectos del entorno del intérprete de comandos y otros programas, que pueden cambiar con el tiempo. Estas variables se establecen cuando se abre una sesión, y la mayoría son configuradas por los *scripts* de inicio del intérprete de comandos.

Aunque se pueden establecer nombres de variables con minúsculas, por costumbre se utilizan nombres en mayúsculas, el comando para establecer las variables de entorno se llama **export**, y se utiliza de la siguiente forma:

| | |
|---|--------------------------------------|
|  | <code>\$export VARIABLE=valor</code> |
|---|--------------------------------------|

Para ver el contenido de una variable, se puede usar el comando **echo** de la siguiente manera:

| | |
|---|------------------------------|
|  | <code>echo \$VARIABLE</code> |
|---|------------------------------|

Para eliminar una variable, se utiliza el comando interno del intérprete `bash`, llamado `unset` pasándole como parámetro el nombre de la variable.



Es importante notar que una vez que se sale de una sesión, las variables establecidas se pierden. Es por eso que si se necesita disponer de variables específicas cada vez que se abra una sesión en GNU/Linux, es imprescindible agregar dichas configuraciones a los archivos de inicio del intérprete de comandos.

Otro uso común de estas variables es en los *scripts*, programas hechos en el lenguaje del intérprete; las variables de entorno son de gran ayuda para establecer configuraciones fácilmente cambiables en dichos programas.



Tema 5: Flujos de entrada y salida en el SHELL.

Entrada-salida Estándar

Una vez que se ejecuta un comando, se crea un proceso. Este proceso abre tres flujos:

- 1) stdin, denominado entrada estándar, en cuyo caso el proceso lee los datos de entrada. De manera predeterminada, stdin se refiere al teclado. stdin se identifica con el número 0.
- 2) stdout, denominado salida estándar, en cuyo caso el proceso escribe los datos de salida. De manera predeterminada, stdout se refiere a la pantalla. stdout se identifica con el número 1.
- 3) stderr, denominado error estándar, en cuyo caso el proceso escribe los mensajes del error. De manera predeterminada, stderr se refiere a la pantalla. stderr se identifica con el número 2.


Por lo tanto, de manera predeterminada, cada vez que se ejecuta un programa, los datos se leen desde el teclado y el programa envía su salida y sus errores a la pantalla. Sin embargo, también es posible leer datos desde cualquier dispositivo de entrada, incluso desde un archivo, y enviar la salida a un dispositivo de visualización, un archivo, entre otros.

Redirecciones


Como cualquier sistema Unix, Linux posee mecanismos que permiten redirigir la entrada-salida estándar a archivos.

Por lo tanto, si se usa el carácter ">", se puede redirigir la salida estándar de un


comando que se encuentra a la izquierda a un archivo que se encuentra a la derecha:

| | |
|---|--|
|  | <pre>\$ls -al /home/jf/ > toto.txt echo "Toto" > /etc/miarchivodeconfiguración</pre> |
|---|--|


El siguiente comando equivale a una copia de los archivos:

| | |
|---|----------------------------------|
|  | <pre>\$cat toto > toto2</pre> |
|---|----------------------------------|


El propósito de la redirección ">" es el de crear un archivo nuevo. En el caso de que un archivo ya exista con el mismo nombre, se lo debe eliminar. El siguiente comando simplemente crea un archivo vacío:

| | |
|---|---------------------------|
|  | <pre>\$> archivo</pre> |
|---|---------------------------|

El uso del carácter doble ">>" permite agregar la salida estándar al archivo, es decir, permite agregar la salida después del archivo sin eliminarlo. De manera similar, el carácter "<" indica una redirección de la entrada estándar. El siguiente comando envía el contenido del archivo *toto.txt* con el comando *cat*, cuyo único propósito es mostrar el contenido en la salida estándar (el ejemplo no es útil, pero es instructivo):

| | |
|---|--------------------------------|
|  | <pre>\$cat < toto.txt</pre> |
|---|--------------------------------|

Por último, el uso de la redirección "<<" permite la lectura, en la entrada estándar, hasta que se encuentre la cadena ubicada a la derecha. En el siguiente ejemplo, se lee la entrada estándar hasta que se encuentra la palabra STOP. Después, se muestra el resultado:

| | |
|---|----------------------------------|
|  | <code>\$cat << STOP</code> |
|---|----------------------------------|

Tuberías de comunicación

Las tuberías (en inglés "*pipes*") (literalmente "tuberías") son mecanismos de comunicación específicos para todos los sistemas UNIX. Una tubería, simbolizada por una barra vertical (carácter "|"), permite asignar la salida estándar de un comando a la entrada estándar de otro, de la misma forma en que una tubería permite la comunicación entre la entrada estándar de un comando y la salida estándar de otro.

En el siguiente ejemplo, la salida estándar del comando *ls -a* se envía al programa *sort*, el cual debe extraer el resultado en orden alfabético.

| | |
|---|----------------------------|
|  | <code>ls -al sort</code> |
|---|----------------------------|

Esto permite conectar una cierta cantidad de comandos a través de sucesivas tuberías. En el siguiente ejemplo, el comando muestra todos los archivos del directorio actual, selecciona las líneas que contienen la palabra "zip" (utilizando el comando *grep*) y cuenta la cantidad total de líneas:



```
ls -l | grep zip | wc -l
```

Salida de errores.

Si quisiéramos realizar un listado de un directorio y, en caso de producirse un error, este fuese redirigido a un archivo, haremos lo siguiente:



```
$ ls /bin 2>/tmp/error.ls
```

Esta simple redirección solo tendrá efecto sobre el error estándar (stderr) o como también se denomina, *descriptor de archivo nº 2*. Con esta redirección los posibles errores serían redirigidos al archivo /tmp/error.ls. Si quisiéramos dividir tanto la salida por pantalla como el error en dos archivos separados podemos hacerlo de esta manera:



```
ls /bin 1>/tmp/salida 2>/tmp/error.ls
```



Tema 6: Historial de comandos.

¿Qué es un historial?

El shell del sistema mantiene un historial de los últimos comandos ejecutados, mediante el comando `history` podemos acceder dicho historial de comandos e incluso interactuar con él. Usando el comando `history` nos muestra una lista de los últimos 500 comandos ejecutados; Asimismo, se puede borrar el historial de comandos ejecutando:



```
history -c
```



Tema 7: Procesos en el sistema.

El shell utiliza el kernel para la ejecución de procesos, los cuales quedan bajo su control. Es posible definir un proceso como un programa en ejecución. Ya que UNIX es multitarea, utiliza una serie de métodos de tiempo compartido en los cuales parece que hay varios programas ejecutándose a la vez, cuando en realidad lo que hay son intervalos de tiempo cedidos a cada uno de ellos según un complejo esquema de prioridades.

Propiedades de un proceso

Básicamente, un proceso tiene las siguientes propiedades:

- Un número identificador, (Process ID o PID), identificador de proceso, es necesario para referirse a un proceso en concreto de los varios que se encuentran en ejecución.
- Un PPID (Identificador del Proceso Padre), es el número que indica qué proceso creó al proceso en cuestión.

Estado de un proceso

Hay momentos en los que un proceso sigue existiendo en el sistema, pero en realidad no están realizando algo, quizás porque pueden estar esperando a que una señal le sea enviada para volverse activo, o a un usuario le puede interesar detenerlo o pausarlo bajo determinadas circunstancias. Los estados más importantes son dormido (S), y en ejecución (R).



Prioridad de un proceso

Para empezar, se pueden ver las tareas y las subtareas en una estructura anidada mediante el comando `ps tree` es decir, permite visualizar un árbol de procesos. Asimismo, `ps tree -p` muestra entre paréntesis el número identificador (PID) de los procesos, algo muy importante cuando se quiere pasar de actuar de forma pasiva a interactuar con los procesos, cosa que normalmente se hace señalando sus PIDs. Aunque dicha información estructurada resulta interesante, existen dos comandos muy conocidos que muestran una cantidad ingente de información sobre los procesos

Gestión de procesos

Comando `ps`

Muestra una lista de los procesos en ejecución. Las opciones más habituales son: `ps u` → que muestra los procesos que pertenecen al usuario actual, `ps aux` → muestra información detallada de todos los procesos en ejecución. Algunos de los campos más importantes mostrados por `ps` son:

- USER - usuario dueño del proceso.
- PID - número identificador del proceso.
- %CPU - porcentaje de uso del microprocesador por parte de este proceso.
- %MEM - porcentaje de la memoria principal usada por el proceso.
- VSZ - tamaño virtual del proceso (lo que ocuparía en la memoria principal si todo él estuviera cargado, pero en la práctica en la memoria principal solo se mantiene la parte que necesita procesarse en el momento).
- RSS - tamaño del proceso en la memoria principal del sistema (generalmente son



KBytes, cuando no lo sea, se indicará con una M detrás del tamaño).

- TTY - número de terminal (consola) desde el que el proceso fue lanzado. Si no aparece, probablemente se ejecutó durante el arranque del sistema.
- STAT - estado del proceso.
- START - cuándo fue iniciado el proceso.
- TIME - el tiempo de CPU (procesador) que ha usado el proceso.
- COMMAND - el comando que inició el proceso.

Comando top

Es la versión interactiva de ps, y tiene algunas utilidades interesantes añadidas. Si se ejecuta en una terminal y sin opciones, aparecerá arriba información del sistema: usuarios, hora, información del tiempo de funcionamiento de la máquina, número de procesos, uso de CPU, uso de memoria y uso del swap y a continuación muestra una lista de procesos similar a la que se muestra con ps, la diferencia entre ambos radica en que ésta se actualiza periódicamente, permitiendo ver la evolución del estado de los procesos.

Con top, se tiene dos posibilidades de especificar opciones, bien en línea de comandos en el shell, o bien interactivamente (mientras está en ejecución y sin salir de él). La página del manual de top es también muy buena, con descripciones detalladas de los campos y de las opciones, tanto de línea de comandos como interactivas.

Comando kill

Este comando sirve para matar o anular procesos indeseados. Se debe tener en



cuenta que cada proceso lleva su usuario y por tanto solo él (o el superusuario) pueden anularlo. Normalmente, si los programas que componen el grupo de procesos son civilizados, al morir el padre mueren todos ellos siempre y cuando el padre haya sido señalado adecuadamente. Para ello, se emplea el comando \$kill - <número_señal> PID, siendo PID el número del proceso o del grupo de procesos. Los números de señales (número_señal) utilizados con más frecuencia son:

| Número | Descripción |
|--------|--|
| 15 | TERM o terminación, se manda para que el proceso cancele ordenadamente todos sus recursos y termine. |
| 1 | HUP ⁶ , normalmente utilizado en procesos de servicios para que hagan una nueva lectura de sus archivos de configuración. |
| 2 | Interrupción |
| 3 | Salir (QUIT) |
| 5 | TRAP. Señal para hacer que el programa depurador que supervisa el proceso informe sobre los puntos de chequeo del mismo. |
| 9 | (KILL) La más enérgica de todas las señales, esta evita que los procesos mueran ordenadamente. El proceso que la recibe finaliza inmediatamente. |

⁶ ó HANGUP: *colgar* en inglés.



Comando bg.

Nos permite reanudar un proceso y que este se ejecute en segundo plano, permitiéndonos hacer uso del prompt mientras dicho proceso avanza.

Comando fg

Este comando nos sirve para reanudar un proceso que se haya detenido y lo envía al primer plano nuevamente para que continúe su ejecución.

Comunicación entre procesos

Los procesos en UNIX no comparten memoria, ni siquiera los padres con sus hijos. Por tanto, hay que establecer algún mecanismo en caso de que se quiera comunicar información entre procesos concurrentes. El sistema operativo UNIX define tres clases de herramientas de comunicación entre procesos (IPC): los semáforos, la memoria compartida y los mensajes.

El tipo de llamadas al sistema para estos IPCs es análogo al de los semáforos: existen sendas funciones `shmget` y `msgget` para crear o enlazarse a un segmento de memoria compartida o a una cola de mensajes, respectivamente. Para alterar propiedades de estos IPCs, incluyendo su borrado, están las funciones `shmctl` y `msgctl`. Para enviar o recibir mensajes, se utilizan las funciones `msgsnd` y `msgrcv`. En este apartado se describirán brevemente algunas llamadas al sistema disponibles para el uso de las IPCs dentro de la programación en C.



Existen 4 formas de comunicación entre procesos en Linux:

1. A través de variables de entorno:

Solo es posible de padres a hijos. Este hace una llamada a un intérprete para ejecutar un comando. El proceso espera a que finalice la ejecución de la subrutina y devuelve la salida del programa ejecutado.

2. Mediante una señal:

Solo indica que algo ha ocurrido y solo lleva como información de un número de señal.

3. Mediante entrada salida:

Es la forma más corriente a nivel de shell. Ejem: el operador pipe '|' que conecta dos procesos.

4. Mediante técnicas IPC u otras:

El resto del conjunto de mecanismos IPC (semáforos, memoria compartida y cola de mensajes) poseen una serie de características comunes a todos ellos, que se pueden resumir de forma básica en los siguientes puntos:

- Una estructura con información acerca de qué se está haciendo con dicho mecanismo.
- Una estructura que define los permisos de los usuarios y grupos de usuarios que pueden acceder al mecanismo IPC.
- Una clave de acceso o llave.
- Un conjunto de funciones que permitirán realizar un control sobre el mecanismo en cuestión. Este conjunto de funciones se puede dividir en tres



grupos:

- La familia get, para crear o buscar un mecanismo.
- La familia ctl, para realizar operaciones de control y suprimir mecanismos.
- Un conjunto de funciones particulares a cada mecanismo (msgsnd, shmat, etc.).

Tema 8: Tareas comunes en el SHELL

Paginación de archivos

Los comandos de paginación, o paginadores, son usados para leer archivos de texto. El más famoso tal vez sea el comando `more`, debido a su uso en el DOS.

El comando `more`

`More` nos muestra el archivo en pantalla. Su sintaxis es la siguiente:



`more [archivo...]`

Este comando es un programa interactivo, es por lo que no se hablará de argumentos sino de comandos:


| Comando | Descripción |
|---------|--|
| ENTER | Pulsando la tecla ENTER se va avanzando de línea en línea. |
| ESPACIO | Si se oprime la barra espaciadora, less avanzará un número de líneas igual al número de líneas por pantalla que posea |

| | |
|---|---------------------------------|
| | la terminal que se esté usando. |
| q | Salir del programa. |

El comando less

Este comando es de mucha utilidad; su función es paginar texto en pantalla. Muchas veces ocurre que cuando se ejecuta algún comando, la salida del mismo aporta demasiada información como para que se pueda leer en la pantalla del monitor. Entonces se puede redireccionar esta salida a less para que permita al usuario leer sin mayores problemas, pudiendo avanzar o retroceder en el texto con las flechas de cursor del teclado. También se utiliza para visualizar archivos de texto almacenados en disco.

La idea de less proviene del paginador antes explicado, llamado more, un clásico en los UNIX y en sistemas DOS. El comando **more** no era lo suficientemente amigable, por eso hicieron less. Su sintaxis es la siguiente:

| | |
|---|--------------------------------|
|  | <code>less [archivo...]</code> |
|---|--------------------------------|

El comando less es un programa interactivo, es por lo que no se hablará de argumentos sino de comandos:

| Comando | Descripción |
|---------|--|
| ESPACIO | Si se oprime la barra espaciadora, less avanzará un número de líneas igual al número de líneas por pantalla que posea la terminal que se esté usando. |
| ENTER | Pulsando la tecla ENTER se va avanzando de línea en línea. |



| | |
|-------|---|
| G | Ir al final del texto. |
| g | Ir al inicio del texto |
| / | Ingresar una palabra a ser buscada avanzando dentro del texto. |
| ? | Ingresar una palabra a ser buscada retrocediendo dentro del texto. |
| n | Ir a la siguiente ocurrencia de la búsqueda |
| AvPag | Avanzar una pantalla de texto. |
| RePag | Retroceder una pantalla de texto. |
| v | Cargar el editor de texto en el lugar donde se encuentre el usuario dentro del archivo. El editor que normalmente se utiliza es el vi , el cual se explica en Introducción al editor de textos vi . |
| q | Salir del programa. |
| r | Repintar la pantalla. Útil cuando se está visualizando un archivo que ha sido modificado por otro programa. |

Manipulación de archivos y directorios

El objeto principal en el uso de los sistemas operativos son los archivos. Los comandos más usados para manipularlos serán explicados a continuación.

Comando cp

Se utiliza para copiar archivos, su sintaxis es la siguiente:

| | |
|---|--|
|  | <code>cp <opciones> archivo-origen directorio-destino</code> |
|---|--|

Entre las opciones más relevantes, se tienen:

| Opción | Descripción |
|--------|---|
| -f | Borrar los archivos de destino ya existentes. |
| -d | Copiar los enlaces simbólicos tal cual son, en lugar de copiar los archivos a los que apuntan. |
| -p | Preservar los permisos, el usuario y el grupo del archivo a copiar. |
| -R | Copiar directorios recursivamente. |
| -a | Equivalente a utilizar las opciones -dpR. |
| -u | No copia un archivo si en el destino ya existe tal archivo, y éste tiene la fecha de modificación igual o mas reciente. |
| -v | Da información en pantalla sobre los archivos que se van copiando. |

Comando mv

Este comando se usa tanto para mover archivos, como para renombrarlos algo que, al fin de cuentas, es una manera de mover archivos; su sintaxis es la siguiente:

| | |
|---|---|
|  | <code>mv <opción> origen destino</code> |
|---|---|


Si el último argumento, destino, es un directorio existente, mv mueve cada uno de los otros archivos a destino.

Algunas opciones de este comando son:

| Opción | Descripción |
|--------|--|
| -f | Borrar los archivos de destino existentes sin preguntar al usuario. |
| -i | Lo contrario de -f; pregunta por cada archivo a sobrescribir antes de hacerlo. |
| -v | Muestra el nombre de cada archivo a ser movido. |

Comando rm

He aquí un comando que debe ser utilizado con cautela, se trata de rm, que se utiliza para borrar archivos o directorios, su sintaxis es:

| | |
|---|---------------------------------------|
|  | <code>rm [opciones] archivo...</code> |
|---|---------------------------------------|

Se debe *siempre* pensar dos veces lo que se está haciendo antes de ejecutar este comando. Es importante tomar en cuenta que este es un comando peligroso pero más aún cuando se está administrando un equipo que da servicios a varios usuarios, un tecleo en falso hace que fácilmente se pierdan datos importantes. Sus opciones más utilizadas son:



| Opción | Descripción |
|--------|--|
| - f | No imprimir mensajes de error, ni pedir al usuario una confirmación por cada archivo que se vaya a borrar. |
| - r | Borrar los contenidos de directorios recursivamente. |
| - v | Muestra el nombre de cada archivo eliminado. |

Comando touch

Este comando se utiliza para cambiar la fecha de acceso y/o modificación a un archivo. Su sintaxis es la que sigue:


| | |
|---|---|
|  | <code>touch [opción...] archivo...</code> |
|---|---|

Si el argumento archivo corresponde al nombre de un archivo que no existe, a menos que se le diga, **touch** creará el archivo con dicho nombre y sin ningún contenido. Sus opciones de mayor importancia son:

| Opción | Descripción |
|--------|--|
| - a | Cambia solamente el tiempo de acceso. |
| - c | No crear archivos que no existían antes. |

Comando cat

Se utiliza para concatenar archivos y mostrarlos por la salida estándar (normalmente la pantalla). Su sintaxis es muy simple:

| | |
|---|--|
|  | <code>cat [opción] [archivo]...</code> |
|---|--|


Donde archivo puede ser uno o más archivos. Si no se especifica este segundo parámetro, **cat** tomará la entrada de la entrada estándar (normalmente el teclado).

Sus opciones más comunes son:

| Opción | Descripción |
|--------|--|
| -n | Numera todas las líneas de salida. |
| -b | Numera aquellas líneas de salida que no estén en blanco. |

Comando ls

Quizás uno de los comandos de mayor utilización, sirve para listar archivos. Su sintaxis es la siguiente:

| | |
|---|---|
|  | <code>ls [opciones] [archivo...]</code> |
|---|---|




Si se ejecuta ls sin argumentos, dará como resultado un listado de todos los archivos (incluyendo directorios) del directorio donde el usuario está posicionado. Sus opciones son:

| Opción | Descripción |
|--------|--|
| -a | Lista todos los archivos, incluyendo aquellos que comienzan con un «.» |
| -d | Lista el nombre del directorio en vez de los archivos contenidos en él. |
| -l | Lista los archivos con mucho más detalle, especificando para cada archivo sus permisos, el número de enlaces rígidos, el nombre del propietario, el grupo al que pertenece, el tamaño en bytes y la fecha de la última modificación. |
| -r | Invierte el orden de listado de los archivos. |
| -s | Muestra el tamaño de cada archivo en bloques de 1024 bytes a la izquierda del nombre. |
| -h | Muestra los tamaños de archivo en términos de kilobytes, megabytes, etc. |
| -t | Lista los archivos ordenados por el tiempo de modificación en vez de ordenarlos alfabéticamente. |
| -a | Lista todos los archivos excepto el «.» y el «..». |
| -R | Lista los contenidos de todos los directorios recursivamente. |
| -S | Ordena el listado por el tamaño de los archivos |



Comando cd


Este comando se usa para cambiar de directorio. Generalmente cuando el usuario inicia una sesión en GNU/Linux, el directorio donde comienza es su directorio personal. Desde ahí uno puede moverse a los diferentes directorios donde se tenga acceso usando este comando. Su sintaxis es la siguiente:

| | |
|---|------------------------------------|
|  | <code>cd <directorio></code> |
|---|------------------------------------|

Éste es un comando interno del intérprete.

Comando mkdir

Este comando es bastante simple; su finalidad es la creación de directorios, y su sintaxis es la que sigue:

| | |
|---|---|
|  | <code>mkdir [opciones] directorio...</code> |
|---|---|

Sus opciones son las que se listan a continuación:

| Comando | Descripción |
|---------|--|
| -m | Establece los permisos de los directorios creados. |
| -p | Crea los directorios padre que falten para cada argumento directorio |


Monitorización del sistema

Comando top

Sirve para saber que procesos hay en ejecución y cuanta memoria consumen, se actualiza cada tres segundos.

La primera línea de información es la salida del comando uptime.

El comando top simultáneamente cumple las funciones de ps y kill. El comando top se usa para mostrar los procesos que más consumen CPU, proporciona una visión continuada de la actividad del procesador en tiempo real. El comando top muestra un listado de las actividades que hacen un uso más intensivo de la CPU, uso de memoria y tiempo de ejecución, la pantalla se actualiza cada 5 segundos de forma predeterminada. Su sintaxis es:

| | |
|---|----------------|
|  | top [opciones] |
|---|----------------|

Sus opciones se ejecutan mientras top muestra información, y son las siguientes:



| Comando | Descripción |
|---------|---|
| d | El comando top especifica el intervalo entre actualizaciones de la pantalla. |
| qq | El comando top redibuja la pantalla sin intercambio de actualización, si el que ejecuta el programa es el superusuario, top se ejecuta con la prioridad más alta posible. |
| S | Especifica el modo acumulativo, cada proceso se lista con el tiempo de CPU que él , así como sus procesos hijos muertos, han consumido. |
| i | Arranca el comando top descartando cualquier proceso inactivo o zombie. |
| -s | Ejecución del comando top en modo seguro. |

El comando top muestra una variada información sobre el estado del procesador:

uptime: el comando top muestra el tiempo que el sistema a estado activo y las tres medias de carga para el sistema (número medio de procesos listos para ejecutarse en los últimos 1, 5 y 15 segundos).

processes: número total de procesos ejecutándose cuando la última actualización.

CPU states: porcentaje de tiempo de CPU en modo de usuario, en modo de sistema, en tareas con la prioridad alterada por el comando nice y el tiempo de inactividad.

Mem: datos sobre el empleo de memoria.

Swap: datos sobre el espacio de trasiego, incluyendo el total, el utilizado y el libre.

PID: el identificador (ID) de proceso (PID) de cada tarea arrojado por el comando top.

PPID: el identificador del proceso padre de cada tarea

UID: el identificador de usuario del propietario de cada tarea

USER: el nombre de usuario del propietario de cada tarea.

PRI: prioridad de cada tarea.

RSS: cantidad de memoria física utilizada por la tarea según el comando top.

SHARE: memoria compartida empleada por la tarea.

STAT: estado de la tarea (S- durmiente, R-ejecución, T-parado o trazado).

TIME: el comando top muestra el tiempo total de CPU que la tarea a utilizado desde comienzo.

Comando uname

El comando uname es el encargado de mostrar información del sistema. Este comando imprime o muestra la información de la máquina y el sistema operativo instalado en la misma. Su sintaxis es la siguiente:



uname [opciones]

Sus opciones son:



| Opción | Descripción |
|--------|---|
| -m | Opción para mostrar el tipo de máquina. |
| -n | Con esta opción se puede visualizar el hostname del nodo de red de la máquina. |
| -r | se utiliza con el comando uname para mostrar la versión del sistema operativo. |
| -s | Muestra el nombre del sistema operativo. |
| -v | Opción del comando uname que permite mostrar la fecha de compilación del sistema operativo. |
| -a | Muestra toda la información anterior. |

Comando uptime

Refleja la hora del sistema, el tiempo que lleva en funcionamiento y el número de usuarios. Muestra también la carga media del sistema durante el último minuto, los últimos cinco y los últimos diez minutos.

Un sistema en operación normal debe mostrar una carga igual o inferior a 3, aunque se deben tener en cuenta la configuración y el tipo de programas en ejecución.

Comando time

Time mide el tiempo de ejecución de un programa. Toma la medida del tiempo



durante el que se ha estado ejecutando la aplicación (real), y de los tiempos de ejecución de código en modo usuario (user) y en modo supervisor (sys), como resultado de llamadas del programa al sistema operativo, de modo que:

$$\text{Tiempo de espera} = \text{real} - \text{user} - \text{sys}$$

Comando vmstat

Muestra información relativa al sistema de memoria, incluyendo datos sobre la memoria física y virtual, la actividad de intercambio entre memoria y disco (swapping), transferencias con el disco, interrupciones, cambios de contexto y utilización del procesador. La sintaxis de esta orden es: vmstat t n. Donde t indica el tiempo transcurrido, normalmente en segundos, entre dos muestras consecutivas, y n es el número de muestras.

La primera línea de información es irrelevante, ya que se calculan desde el instante en el que se inició el sistema hasta el momento actual.

La información de este comando es:

| Comando | Descripción |
|---------|---|
| r | procesos en espera de ser ejecutados. |
| b | procesos durmiendo ininterrumpidamente. |
| ww | procesos intercambiados. |
| swpd | memoria virtual en uso. |



| | |
|-------|---|
| free | memoria física libre. |
| buff | memoria usada como buffer. |
| cache | memoria usada como caché. |
| si | memoria intercambiada desde disco (KB/s) |
| so | memoria intercambiada hacia disco. (KB/s) |
| bi | bloques de memoria por segundo enviados a disco. |
| bo | bloques de memoria por segundo enviados desde disco . |
| in | interrupciones por segundo. |
| cs | cambios de contexto por segundo. |
| us | uso del procesador ejecutando código de usuario. |
| sy | uso del procesador ejecutando código de sistema. |
| id | porcentaje de tiempo con el procesador ocioso. |

Se puede usar con algunos modificadores:

| Comando | Descripción |
|---------|--|
| -a | aporta información acerca de la memoria activa e inactiva. |
| -f | da el número de tareas creadas desde el arranque. |
| -d | da estadísticas del uso de los discos. |



Comando free:

Permite obtener información del estado de la memoria del sistema, tanto de la memoria física como de la del archivo de intercambio. Presenta el valor de la memoria total disponible, la que se encuentra en uso y libre, la memoria compartida que se encuentra en uso, número de buffers utilizados y tamaño de la caché.

Puede utilizarse para capturar información de forma periódica: free s t, donde t indica el tiempo entre muestreos consecutivos.

Comando df:

Permite monitorizar el sistema de archivos.

Muestra la cantidad de espacio disponible en cada unidad montada del sistema de archivos. El espacio se muestra en bloques de 1K por defecto.

Algunos de sus modificadores son:

| Comando | Descripción |
|---------|---|
| -h | mejora la legibilidad al usar unidades más grandes. |
| -l | muestra solo las unidades locales. |



Comando du

Du muestra la capacidad ocupada por un directorio concreto. Algunos de sus modificadores son:

| Comando | Descripción |
|---------|------------------------------------|
| -h | mejora la legibilidad. |
| -all | muestra solo las unidades locales. |



Tema 9: Programación en SHELL.

La programación en shell se basa en el uso de las herramientas del sistema, y el UNIX es un sistema operativo (UNIX y sus clones) que cuenta con bastantes herramientas de proceso y filtrado de textos y de control de procesos, entre otras. Por ello, permite automatizar procesos repetitivos, que hechos a mano serían engorrosos y lentos.

Descripción de elementos de programación

Variables

Una variable es un contenedor. Consta de un identificador que la distingue de otra (su nombre) y de un contenido. La relación entre variable y contenido es de equivalencia.

Línea de comandos

Cuando se ejecuta nuestro programa en shell hay una serie de variables que siempre estarán disponibles, entre ellas las que nos permiten acceder a los distintos argumentos con los que fue ejecutado nuestro script.

Parámetros

Dentro del script, los parámetros recibidos pueden referenciarse con \$1, \$2, \$3, ..., \$9, siendo \$0 el nombre del propio programa. Debido a que se los reconoce por su ubicación, se llaman parámetros posicionales.



La salida de los programas

Cuando se ejecuta un programa, un comando UNIX es un programa, podemos, a parte de redirigir su entrada y su salida, recoger el resultado de su ejecución y su salida. El resultado es un valor numérico, por lo general cero si todo ha ido bien, y distinto de cero si ha habido alguna clase de error. La salida del programa es lo que obtendríamos en **stdin** y **stdout**.

Operación Aritmética

Las operaciones aritméticas permiten manejar los datos para producir resultados de expresiones. Varios circuitos se fabrican y diseñan para tales propósitos, los más típicos son los de: suma, resta, multiplicación y división otros menos conocidos pero menos frecuentes son el resultado de uno o más de ellos, por ejemplo los contadores, el circuito contador es básicamente un circuito cuya operación principal es la suma, con la particularidad de que al llegar al tope deseado, retorna al valor inicial.

Manejo de parámetros

Los parámetros son variables normales, que tienen los nombres \$1, \$2 ... \$9. Aunque se pueden dar más de nueve parámetros a un guión para el intérprete de órdenes, solo se puede acceder de forma directa a los nueve primeros. La orden shift permite desplazar los parámetros de sitio, de tal forma que sean accesibles los que estén más allá del noveno, con el inconveniente de no poder acceder a los primeros. El

funcionamiento de shift es el siguiente:

Supongamos que tenemos como parámetros \$1=-o, \$2=foo y bar, por llamar al guión (suponiendo que el nombre del guión es compila) así:

| | |
|---|---------------------------------|
|  | <code>compila -o foo bar</code> |
|---|---------------------------------|

Lo que queremos es quitarnos de en medio las opciones, después de haberlas procesado, de tal forma que el tercer parámetro (bar) se quede como primero. Lo que haremos, entonces, es llamar dos veces a shift, o llamar a shift con el parámetro 2. Teniendo este código:

| | |
|---|-----------------------------------|
|  | <code>shift 2 echo \$1</code> |
|---|-----------------------------------|

Y suponiendo la llamada anterior, el resultado por pantalla sería bar.

Las variables \$#, \$*, \$0 nos permiten saber el número de parámetros pasados al guión, la ristra entera de todos los parámetros pasados, y el nombre del programa que se ha llamado.


Manejo de variables

Por lo general las variables en shell no tienen tipos asociados y se definen de la siguiente forma:



identificador = contenido


Ejemplos:

| | |
|---|---|
|  | <pre># i vale 1 i=1 # I vale echo I=echo # msg vale Hola mundo! msg="Hola mundo!"</pre> |
|---|---|

Debemos tener cuidado si dejamos espacios entre el = y el identificador o el valor, el shell creará que son comandos a ejecutar y no la asignación de una variable.

Para acceder al contenido de una variable empleamos \$ delante de su identificador:
\$identificador


Ejemplos:

| | |
|---|--|
|  | <pre>\$ i=1 ; echo \$i \$ msg="Mola mundo!" ; echo \$msg \$ fu=echo; \$fu goo!</pre> |
|---|--|

Cuando empleamos **\$identificador** el shell busca el valor almacenado en la variable asociada a ese identificador y lo utiliza para reemplazar esa ocurrencia de **\$identificador**.

Operaciones aritméticas


Para que el shell evalúe una operación aritmética y no la tome como argumentos de un comando, por ejemplo:

| | |
|---|------------------------|
|  | <pre>\$ echo 1+1</pre> |
|---|------------------------|

Si queremos que sustituya la operación por su valor emplearemos:

`$((expresión))` evalúa la expresión aritmética y reemplaza el bloque por el resultado

Ejemplo:

| | |
|---|------------------------------|
|  | <pre>\$ echo \$((1+1))</pre> |
|---|------------------------------|


Algunos operadores aritméticos soportados:

- + suma
- * multiplicación
- resta
- / división entera
- % resto de la división entera


() agrupar operaciones

Interactuando con archivos

Es cómodo poder retener una lista larga de comandos en un archivo, y ejecutarlos todos de una sola vez solo invocando el nombre del archivo. Crearemos el archivo `misdatos.sh` con las siguientes líneas:

| | |
|---|---|
|  | <pre># misdatos.sh # muestra datos relativos al usuario que lo invoca # echo "MIS DATOS." echo " Nombre: "\$LOGNAME echo "Directorio: "\$HOME echo -n "Fecha: " date echo # fin misdatos.sh</pre> |
|---|---|

El símbolo `#` indica comentario. Para poder ejecutar los comandos contenidos en este archivo, es preciso dar al mismo permisos de ejecución:

| | |
|---|-------------------------------------|
|  | <pre>\$chmod ug+x misdatos.sh</pre> |
|---|-------------------------------------|

La invocación (ejecución) del archivo puede realizarse dando el nombre de archivo como argumento a `bash`.



```
$bash misdatos.sh
```

Invocándolo directamente como un comando:



```
$misdatos.sh
```

Puede requerirse indicar una vía absoluta o relativa, o referirse al directorio actual:



```
./misdatos.sh
```

Esto si el directorio actual no está contenido en la variable PATH.



UNIDAD IV: Gestión de usuarios y grupos.

Tema 1: Gestión de usuarios.

Linux es un sistema multiusuario y permite que varios usuarios puedan acceder, incluso simultáneamente. Cada usuario podrá tener su configuración y sus archivos independientes.

Los grupos permiten asignar permisos de archivos y directorios a muchos usuarios de una vez.

A un grupo pueden pertenecer varios usuarios y un usuario puede pertenecer a varios grupos. Un usuario tiene asignado un grupo principal o por defecto.

El Superusuario dentro del entorno de GNU/Linux, es aquel usuario que posee todos los privilegios dentro del sistema, es capaz de realizar cualquier operación dentro del sistema, es equivalente al usuario administrador dentro de los sistemas Microsoft Windows. Además de entrar en el login del sistema como root, hay dos formas para ampliar los privilegios de un usuario y adquirir los de root. Los dos programas para hacer esto son su y sudo.

El comando su hace que un usuario que se haya identificado con su propia cuenta pueda cambiar su uid al de root. Por supuesto debe saber el password del root.

El comando sudo, en este caso no es necesario que el usuario conozca la contraseña de root.

Este programa permite que un usuario pueda ejecutar determinados comandos con privilegios de root. Estos usuarios y los comandos permitidos para él deben de estar en el



archivo /etc/sudoers.

Por ejemplo para que el usuario carlos pueda hacer un shutdown del sistema debe haber una entrada en el archivo sudoers como: carlos /sbin/shutdown -[rh] now.

Comandos más utilizados:

- Comando who: información sobre los usuarios que usan el sistema en este momento.
- Comando finger: información sobre el usuario usuario.
- Comando adduser: registra y crea una cuenta de usuario.

En ese momento, no solo se creará la cuenta del usuario sino también su directorio de trabajo, un nuevo grupo de trabajo que se llamará igual que el usuario y añadirá una serie de archivos de configuración al directorio de trabajo del nuevo usuario:



```
root@defuser:/home# adduser luis
Adding user luis...
Adding new group luis (1000).
Adding new user luis (1000) with group luis
Creating home directory /home/luis
Copying files from /etc/skel
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for luis
Enter the new value, or press return for the
default
Full Name []:
Room Number []:
Work Phone []:
```



Home Phone []:
Other []:
Is the information correct?
[y/n] y

En ese momento, el usuario ya puede trabajar en el sistema. Otro comando utilizado es `deluser`, el cual borra la cuenta de usuario usuario. Este comando no elimina automáticamente el directorio de trabajo del usuario.



```
root@defuser:/home# deluser luis
Removing user luis...
done.
```

Una vez realizado este proceso, es responsabilidad del administrador decidir si elimina el directorio de trabajo del antiguo usuario.

Comando `passwd`: cambia la clave de acceso para el usuario actual. `root` puede cambiar la clave de cualquier usuario con `passwd usuario`. `# passwd` Víctor.

La base de datos básica de usuarios en un sistema Unix es un archivo de texto `/etc/passwd` (llamado el archivo de contraseñas), que lista todos los nombres de usuarios válidos y su información asociada.

El archivo tiene una línea por usuario, y está dividido en siete campos delimitados cada uno por dos puntos, estos campos son:

- nombre de usuario.
- contraseña, de modo encriptado.

- Identificación (Id) de número de usuario.
- Identificación (Id) de número de grupo
- Nombre completo u otra información descriptiva de la cuenta.
- Directorio Inicio (directorio principal del usuario).
- Interprete de comandos (programa a ejecutar al ingresar al sistema).

Cualquier usuario del sistema puede leer el archivo de contraseñas, para por ejemplo conocer el nombre de otro usuario del mismo. Esto significa que la contraseña (el segundo campo) esta también disponible para todos. El archivo de contraseñas encripta las contraseñas, así que en teoría no hay problema, pero dicho encriptado puede ser quebrado, sobre todo si dicha contraseña es débil. Por lo tanto no es buena idea tener las contraseñas en el archivo de contraseñas.



Muchos sistemas GNU/Linux tienen contraseñas sombra. Esto es una alternativa en la manera de almacenar las contraseñas: las claves encriptadas se guardan en un archivo separado `/etc/shadow` que solo puede ser leído por el administrador del sistema. Así el archivo `/etc/passwd` solo contiene un marcador especial en ese segundo campo. Cualquier programa que necesite verificar un usuario o uid, pueden también acceder al archivo `shadow/sombra`. Significa también que programas normales que solo usan otros campos del archivo de contraseñas, no pueden acceder a las contraseñas. Paralelamente también existe `/etc/gshadow` para cierta información según grupos.



Creación de cuentas de usuario.

Crear un usuario manualmente

Para crear una nueva cuenta de forma manual, sigue estos pasos:



1. Editar `/etc/passwd` con `vi` y agregar una nueva línea por cada nueva cuenta. Teniendo cuidado con la sintaxis. No se debe editar directamente con un editor, se debe usar `vi` que bloquea el archivo, así otros comandos no tratarán de actualizarlo al mismo tiempo. Se debería hacer que el campo de la contraseña sea `*`, de esta forma es imposible ingresar al sistema.
2. Similarmente, edite `/etc/group` con `vi`, si necesita crear también un grupo.
3. Cree el directorio Inicio del usuario con el comando `mkdir`.
4. Copie los archivos de `/etc/skel` al nuevo directorio creado.
5. Corrija la pertenencia del dueño y permisos con los comandos `chown` y `chmod`. La opción `-R` es muy útil. Los permisos correctos varían un poco de un sitio a otro, pero generalmente los siguientes comandos harán lo correcto:
`cd /home/nuevo-nombre-de-usuario`
`chown -R nombre-de-usuario.group .`
`chmod -R go=u,go-w . chmod go= .`
6. Asigne una contraseña con el comando `passwd`

Después de asignar la contraseña del usuario en el ultimo paso, la cuenta funcionara. No debería configurar esto hasta que todo lo demás este hecho, de otra



manera el usuario puede inadvertidamente ingresar al sistema mientras copias los archivos de configuración de su entorno de trabajo.

A veces es necesario crear cuentas falsas que no son usadas por personas. Por ejemplo, para configurar un servidor FTP anónimo (así cualquiera podrá acceder a los archivos por él, sin tener que conseguir una cuenta de usuario en el sistema primero) podría crear una cuenta llamada "ftp". En esos casos, usualmente no es necesario asignar una contraseña (el ultimo paso de arriba). Verdaderamente, es mejor no hacerlo, para que nadie puede usar la cuenta, a menos que primero sea root/cuenta administrador, y así convertirse en cualquier usuario.

Modificación de cuentas de usuarios.


Comando usermod

Como su nombre lo indica, usermod permite modificar o actualizar un usuario o cuenta ya existente. Sus opciones más comunes o importantes son las siguientes:


| Opción | Descripción |
|--------|--|
| -c | añade o modifica el comentario, campo 5 de /etc/passwd |
| -d | modifica el directorio de trabajo o home del usuario, campo 6 de /etc/passwd |
| -e | cambia o establece la fecha de expiración de la cuenta, formato AAAA-MM-DD, campo 8 de /etc/shadow |
| -g | cambia el número de grupo principal del usuario (GID), campo 4 de |

| | |
|----|--|
| | /etc/passwd |
| -G | establece otros grupos a los que puede pertenecer el usuario, separados por comas. |
| -l | cambia el login o nombre del usuario, campo 1 de /etc/passwd y de /etc/shadow |
| -L | bloque la cuenta del usuario, no permitiéndole que ingrese al sistema. No borra ni cambia nada del usuario, solo lo deshabilita. |
| -s | cambia el shell por defecto del usuario cuando ingrese al sistema. |
| -u | cambia el UID del usuario. |
| -U | desbloquea una cuenta previamente bloqueada con la opción -L. |

Si quisiéramos cambiar el nombre de usuario de 'pablo' a 'pablito':

| | |
|---|----------------------------|
|  | # usermod -l pablito pablo |
|---|----------------------------|

Esto también cambiará el nombre del directorio de inicio o HOME en /home, pero si no fuese así, entonces:

| | |
|---|------------------------------------|
|  | # usermod -d /home/pablito pablito |
|---|------------------------------------|

Otros cambios o modificaciones en la misma cuenta:



```
# usermod -c "supervisor de area" -s /bin/ksh -g  
505 pablito
```

Lo anterior modifica el comentario de la cuenta, su shell por defecto y su grupo principal de usuario quedó establecido al GID 505 y todo esto se aplicó al usuario 'pablito' que como se observa debe ser el último argumento del comando.

Por otro lado si queremos bloquear el usuario 'pablito' para asegurarnos de que nadie use su cuenta:



```
# usermod -L pablito
```

Eliminación de cuentas de usuario.

Como su nombre lo indica, userdel elimina una cuenta del sistema, userdel puede ser invocado de tres maneras:




```
# userdel pablo
```


Sin opciones elimina la cuenta del usuario de /etc/passwd y de /etc/shadow, pero no elimina su directorio de trabajo ni archivos contenidos en el mismo, esta es la mejor



opción, ya que elimina la cuenta pero no la información de la misma.

| | |
|---|-------------------------------|
|  | <pre># userdel -r pablo</pre> |
|---|-------------------------------|

Al igual que lo anterior elimina la cuenta totalmente, pero con la opción -r además elimina su directorio de trabajo y archivos y directorios contenidos en el mismo, así como su buzón de correo, si es que están configuradas las opciones de correo. La cuenta no se podrá eliminar si el usuario esta autenticado en el sistema al momento de ejecutar el comando.

| | |
|--|-------------------------------|
|  | <pre># userdel -f pablo</pre> |
|--|-------------------------------|

La opción -f es igual que la opción -r, elimina todo lo del usuario, cuenta, directorios y archivos del usuario, pero además lo hace sin importar si el usuario esta actualmente en el sistema trabajando. Es una opción muy radical, además de que podría causar inestabilidad en el sistema, así que hay que usarla solo en casos muy extremos.

Consulta de información de cuentas de usuario.

La información sobre todos los usuarios de un sistema Linux se guarda en el fichero */etc/passwd*, donde cada línea del fichero es un registro único con la información existente sobre el usuario.


Bases de datos usuarios y passwords.

UNIX utiliza un programa llamado crypt(3) para encryptar las passwords, este programa se basa en el algoritmo de encryptacion DES (Estándar de Encryptacion de Datos).

Las passwords pueden tener como máximo 11 caracteres pero una vez encriptadas tienen 13, esto es debido a que los dos primeros caracteres son la semilla y se generan a partir del contador del reloj en el momento en el que se asigna la contraseña, esta semilla consta de 12bits, esto quiere decir que puede ser un valor de entre 0 y 4095. Así para cada contraseña posible hay 4095 formas diferentes de encriptación.

En el fichero /etc/passwd se guardan el nombre de usuario, nombre real, información de identificación, y si el sistema no esta en shadow el passwd encryptada del usuario.

Un ejemplo:

| | |
|---|---|
|  | <code>root:WLR103VcgThbH,31AB:0:0:root:/root:/bin/bash</code> |
|---|---|

Donde WLR103VcgThbH es la clave encryptada y WL es la semilla. El número 3 es el máximo de vida del passwd, después de tres semanas el usuario deberá cambiarla. El número 1 indica cuántas semanas deben pasar para que el usuario pueda cambiar su password. AB Indica cuándo se cambió por última vez la clave, (Indica el número de semanas que han pasado desde 1/1/1970).

Estos datos están escritos en base 64. Si por ejemplo root:...,31AB:0:0:root:/root:/bin/bash , indica que el usuario tendrá que cambiar su clave la próxima vez que se conecte al sistema, para cambiar el passwd basta con teclear:



```
$passwd -n <mínimo_de_días> -x <máximo_de_días>  
usuario
```

Donde mínimo_de_días es el número de días que han de pasar para que el usuario puede cambiar su passwd, máximo_de_días es el número de días en el que expirará la clave. Para deshabilitar la expiración basta con que el valor de tiempo sea -1. Por ejemplo en el caso del usuario Invitado:



```
$passwd -x 30 Invitado
```

El usuario Invitado tendrá que cambiar su clave cada mes.

Si en el campo de la clave aparece un NP, (NO password) la cuenta no tienen password, si por el contrario aparece LK (Locked password) significa que la cuenta esta bloqueada.

Para buscar las cuentas sin passwd:



```
#egrep 'NP|::' /etc/passwd
```

Para buscar por ejemplo info de un usuario en el fichero passwd:



```
#egrep 'usuario1|usuario2' /etc/passwd
```

El número 0 es el identificador de Usuario (uid). El identificador de usuario permite determinar, entre otras cosas, qué ficheros son propiedad de dicho usuario, cuáles son los que se pueden ejecutar, a qué zonas tiene acceso un usuario, etc. ; Con esto se quiere decir que UNIX/Linux no identifica de quien es cada cosa por el nombre de usuario sino por el UID. UID es un entero sin signo de 16 bits, el cual va de 0 a 65535. Los UID's menores de 9 son habitualmente usados por funciones del sistema como ftp. El UID 0 corresponde a el superusuario (root), los UID para los demás usuarios suelen empezar a partir del número 100.

El siguiente campo pertenece al Identificador de Grupo, en este caso root pertenece al grupo 0. A los usuarios que van a pertenecer al mismo grupo se les asigna el mismo gid.

El gid puede ir de 0 hasta 60000, El 0 corresponde al grupo de superusuario. Los usuarios que están en el mismo grupo tienen el mismo nivel de privilegios para aquellos elementos que pertenezcan al grupo.




Tema 2: Gestión de grupos de usuarios.

Creación de grupos de usuarios

Addgroup

Con addgroup realizamos la creación de usuarios en el sistema. La forma de hacerlo es:


| | |
|---|---|
|  | <pre>root@defuser:/home# addgroup usuarios Adding group usuarios (105)... Done.</pre> |
|---|---|

El número 105 nos indica que ese es el identificador numérico que se le asigna al nuevo grupo en el momento de su creación.

Modificación de grupos de usuarios.

Comando newgrp

El comando newgrp permite a un usuario cambiar su grupo primario hacia otro al cual también pertenezca. Su uso se hace de la siguiente manera:

| | |
|---|---------------------------|
|  | <pre># newgrp grupo</pre> |
|---|---------------------------|

Comando chgrp

La orden chgrp sirve para cambiar el grupo propietario de un archivo, su sintaxis es:



```
chgrp [opciones] [grupo] [archivo]
```

La orden chown sirve para cambiar el propietario de un archivo, pero también admite la utilización de uid y gid.

Por ejemplo:



```
chown 500:100 /bin/bash
```

Cambia el propietario del archivo al usuario con uid 500 y al grupo gid 100. Se recomienda que /bin/chown solo pueda ser ejecutado por los usuarios con gid0.

Eliminación de grupos de usuarios.

Utilizando el comando delgroup, se pueden borrar los grupos pertenecientes al sistema. La eliminación de un grupo se hace de esta forma:



```
root@defuser:/home# delgroup usuarios
Removing group usuarios...
Done.
```

¿Qué puede pasar si tratamos de eliminar un grupo inexistente? El sistema nos avisará con el siguiente mensaje:



```
root@defuser:/home# delgroup usuarios
/usr/sbin/delgroup: `usuarios' does
not exist.
```

Consulta de información de grupos.

En el archivo /etc/group esta la lista de cada grupo, un ejemplo del /etc/group es:



```
Administ:*:0:root,Charli---
Programadores:*:10:Juan,Pedro,Antonio---
Usuarios:*:100:
```

Donde Administ Es el nombre del grupo, * Es el password del grupo y en este caso no hay passwd. Root y Charli serían los miembros del grupo.

Pueden haber un usuario que pertenezca a varios grupos, el usuario podrá entrar en todos aquellos grupos donde esté su cuenta, por defecto al entrar en el sistema entrará en el grupo al que corresponda a su gid.



Directorios personales.

En un sistema GNU/Linux, cada usuario dispone de su propio directorio personal donde puede guardar los documentos creados por él con los distintos programas. Este directorio personal puede ser de acceso exclusivo para cada usuario, por lo que ningún otro usuario podrá entrar en él y visualizar el contenido de los archivos que contiene (aunque esto depende del nivel de seguridad seleccionado durante la instalación del sistema, ya que lo normal suele ser que el resto de usuarios puedan entrar en él y ver el contenido de algunos archivos y subdirectorios, pero no modificarlo).

Los directorios personales están ubicados en */home/<nombre de usuario>*. El usuario *root* sí dispone de los permisos suficientes para acceder a los directorios personales del resto de usuarios.



Tema 3: Administrador del sistema.

Características del administrador.

A continuación se citan algunos de los puntos más importantes que un administrador Linux debe conocer al momento de realizar sus tareas:

- **Características de Linux**
 - Sistema Operativo Basado en los paradigmas UNIX
 - Multitud de herramientas por consola que se complementan.
- **Componentes: Procesos**
 - Las distintas tareas que están realizándose en la máquina se denominan procesos.
 - Como administradores debemos gestionar los trabajos que realiza nuestro sistema.
 - Podemos verlos y analizarlos (ps, top), darlos prioridades (nice), pararlos (kill).
 - Podemos saber de quien es el trabajo, su estado actual, sus prioridad, cuanto lleva, sus recursos.
- **Componentes: CPU y Memoria**
 - Recursos principales por los que “luchan” los procesos.
 - Linux soporta múltiples procesadores y múltiples arquitecturas
 - Dispone de memoria virtual
 - El administrador debe conocer la utilización que se está haciendo de estos recursos.
- **Componentes: S. Ficheros**
 - Organización de directorios estándar FHS⁷.

⁷ *Filesystem Hierarchy Standard*, Siglas en inglés de: Jerarquía estándar del sistema de archivos.



- Amplio soporte de sistema de ficheros. Capa virtual VFS⁸.
- Cualquier sistema de archivos lo montaremos sobre el árbol de directorios
- Para tener información de los sistemas tenemos los comandos `df` y `du`.
- Podemos establecer cuotas (quotaon) del sistema a los usuarios, con `edquota`, verlas con `quotacheck`.
- Dispone de permisología para archivos basadas en unix y acl.
- **Componentes: Núcleo**
 - Componente principal de sistema, encargado de relacionar todas las partes de sistema, desde el hardware hasta las características que queremos para nuestro sistema.
 - Es un núcleo monolítico con partes modulares. Que comprenden funcionalidades adicionales y controladores de hardware.
 - Linux permite al administrador compilar el núcleo a partir del código fuente para obtener el máximo rendimiento y flexibilidad de su sistema.
- **Componentes: Gestión de Usuarios**
 - Nos podemos cambiar de un usuario a otro con el comando `su`.
 - Los usuarios se agrupan en grupos. Para ver nuestros grupos con `id`.
 - Los procesos y ficheros pertenecen a los usuarios.
 - Un usuario que todo lo puede, `root`. Nosotros los administradores.
 - El administrador puede crear usuarios (`useradd`), borrarlos (`userdel`), cambiar la clave (`passwd`).
 - La lista de usuarios se encuentra en `/etc/passwd`.
- **Componentes: Auditoría y Logs**
 - Todo lo que va ocurriendo en el sistema va a quedar apuntado.
 - Como administradores podemos revisar que ha estado pasando en nuestro sistema.
 - Los principales servicios y el núcleo dejan sus incidencias en `/var/logs`.
 - El registro de usuarios se puede ver con `last` y `lastlog`. Los actuales con `w`, `who` y `users`.

8 Virtual File System.



- **Componentes: Distribuciones**

- Una distribución es el conjunto de aplicaciones con las que se nos presenta un sistema Linux para su instalación, actualización y correcciones del software.
- El administrador decide la distribución en instalación y puede instalar el software que desea en su máquina.
- Las distribuciones dan la posibilidad de instalar aplicaciones de administración mucho más complejas (webmin, por ejemplo).

- **Servicios**

- Los servicios son procesos que están disponibles en el sistema para la realización de tareas esenciales típicas de los sistemas.
- El administrador decide cuáles son los servicios que va a disponer el sistema.
- Existen servicios para la gestión local de la máquina, ya sea del sistema operativo (cron, syslog, ...) o hardware (apmd, udev, ...)

- **Otras tareas de Administrador**


- Gestión de recursos
- Optimización y personalización del sistema
- Auditoría del sistema
- Seguridad y permisos
- Backup y Restauración
- Administración de Red

Suplantación de identidad.

El comando sudo es una utilidad presente en cualquier sistema Linux que permite a cualquier usuario ejecutar alguno, algunos o todos los comandos como si fuera root, previa autorización de este, habilitando a dicho usuario la capacidad de ejecutar comandos como si fuera el superusuario.

Resulta realmente útil cuando utilizamos nuestro usuario regular (diferente e root) pero necesitamos puntualmente ejecutar algún comando de administración. El uso de sudo deviene en una delegación de permisos más segura para que un usuario regular del sistema utilice privilegios de superusuario de vez en cuando, sobre todo evitando su cambio directo a este nivel de privilegios utilizando el comando su. Por otra parte, permitirá que usuarios que no sean root puedan ejecutar comandos que en principio solo este usuario puede ejecutar sin que tengan que tener acceso a un control total del equipo o a la contraseña del superusuario. Por ejemplo, sería posible delegar la capacidad de crear usuarios a un usuario que no fuera root habilitándole la posibilidad de ejecutar ese comando a través de sudo.


Si lo hemos dejado preparado, podremos ejecutar ciertos comandos (o todos) directamente si tener que convertirnos en root, simplemente anteponiendo el comando sudo delante del comando que queramos ejecutar. Supongamos que quiero crear un usuario nuevo

| | |
|---|--|
|  | <code>usuario@maquina:/home/usuario\$ sudo adduser fulano</code> |
|---|--|

Existen dos modos de funcionamiento para sudo:

- Uno solicitará la contraseña (la del usuario) cada vez que se utilice sudo (para evitar que quién consiga mi usuario habitual esté realmente consiguiendo como consecuencia permisos de root)
- Otro modo en el que se puede evitar que pida contraseña por lo que resulta más cómodo porque los comandos se ejecutan directamente.

Por último, es necesario saber cómo se configura sudo. Tendremos que hacernos root y ejecutar el comando visudo, que nos dará acceso a editar el fichero de configuración de sudo, ubicado en /etc/sudoers donde podremos indicar que usuarios y en que máquinas tienen acceso a que permisos y se solicitará a estos su contraseña cada vez que invoquen algún comando a través de sudo. Se trata de editar líneas siguiendo este formato:

| | |
|---|--|
|  | <pre>root@maquina:/root# visudo</pre> <p><i># Nos abrirá un editor de texto donde podremos agregar entradas en esta forma:</i></p> <pre>mengano ALL = (ALL) ALL perencejo ALL = NOPASSWD: ALL fulano ALL = (ALL) adduser</pre> |
|---|--|

En el primer caso, el usuario mengano podrá ejecutar desde cualquier máquina (ALL =), como root ((ALL)) cualquier comando (ALL) a través de sudo. En el segundo caso, al usuario perencejo no se le solicitará contraseña (NOPASSWD:) y podrá ejecutar cualquier comando. Y, por último, el usuario fulano solo tendrá acceso al comando adduser.

En vez de indicar entre paréntesis la palabra ALL, podremos indicar el nombre de algún usuario de manera que en vez de ejecutar los comandos de esa entrada como usuario, lo hagan como algún otro usuario del sistema. En el siguiente ejemplo, el usuario fulano podrá ejecutar, a través de sudo, el comando mkdir como si lo hiciera siendo el



usuario mengano:



```
root@maquina:/root# visudo
```

```
fulano ALL = (mengano) mkdir
```



Tema 4: Grupos y usuarios especiales en el sistema.

Los grupos en Linux son un mecanismo para gestionar una serie de usuarios del sistema. Todos los usuarios de Linux tienen un ID de usuario, un ID de grupo y un número de identificación único llamado userid (UID) y respectivamente un grupo (GID). Los grupos pueden ser asignados lógicamente junto a los usuarios para establecer una relación común de seguridad, privilegios y el derechos de acceso. Esta viene a ser la fundación de la seguridad y acceso en Linux.

Usuarios especiales preexistentes.

Algunos son: bin, daemon, adm, lp, sync, shutdown, mail, operator, squid, apache, etc. Sus características son las siguientes:

- Se les llama también cuentas del sistema.
- No tiene todos los privilegios del usuario root, pero dependiendo de la cuenta asumen distintos privilegios de root. Esto para proteger al sistema de posibles formas de vulnerar la seguridad.
- No tienen contraseñas pues son cuentas que no están diseñadas para que se inicien sesiones con ellas.
- También se les conoce como cuentas "sin login" (nologin).
- Se crean, por lo general de forma automática al momento de la instalación de la distribución Linux o de la aplicación que usa la cuenta en cuestión.
- Generalmente se les asigna un UID entre 1 y 100 (definido en el archivo `/etc/login.defs`)



Grupos especiales preexistentes.

Algunos de ellos son: root, bin, daemon, sys, adm, tty, disk, lp, mem, kmem, wheel, mail, man, floppy, named, rpm, xfs, cdrom, ftp, lock, sshd, nobody users, etc. Esta es solo una lista parcial de los grupos predeterminados. También habrá un conjunto predeterminado identificadores de usuarios miembros los cuales, estarán asociados a la mayoría de los grupos.



UNIDAD V: Gestión de almacenamiento y sistema de archivos.

Tema 1: Dispositivos de almacenamiento en sistemas GNU/Linux.

UNIX, y por lo tanto GNU/Linux, reconocen dos tipos de dispositivos: dispositivos de bloques de acceso aleatorio (tales como discos), y dispositivos de caracteres (tales como cintas y líneas seriales), algunos de estos últimos pueden ser de acceso secuencial y algunos de acceso aleatorio. Cada dispositivo soportado en GNU/Linux es representado en el sistema de archivos como un archivo de dispositivo.

Cuando se lee o escribe sobre un archivo de dispositivo, los datos van o vienen desde el dispositivo que este representa. De esta manera no se necesitan programas especiales (y no se necesitan ningún método especial de programación, como descubrir interrupciones o escudriñar puertos seriales) para acceder a los dispositivos. Por ejemplo, para enviar un archivo a la impresora, puede simplemente ejecutar: `$ cat nombre_de_archivo > /dev/lp0` y el contenido del archivo es impreso (en este caso, el archivo debe estar en un formato que la impresora comprenda). Nótese, que no es una buena idea tener a varias personas realizando un `cat` de sus archivos a la impresora al mismo tiempo. Generalmente se utiliza un programa especial para enviar los archivos a que sean impresos (usualmente el servicio de impresión ya instalado en el sistema). Estos programas se aseguran de que solo un archivo esté siendo impreso en un momento dado, y automáticamente envía archivos a la impresora en cuanto se finalice la impresión del archivo previo. Algo similar puede ser necesario para la mayoría de los dispositivos. De hecho, raramente los archivos de dispositivos son utilizados directamente por los usuarios del sistema.

Ya que los archivos de dispositivos se muestran como archivos en el sistema (en el directorio `/dev`), es fácil ver cuales de ellos existen, utilizando `ls` o algún otro comando



similar. En la salida del comando `ls -l`, la primera columna indica el tipo de archivo y sus permisos. Por ejemplo, observe la salida al inspeccionar un archivo de dispositivo de un puerto serial:

| | |
|---|---|
|  | <pre>\$ ls -l /dev/ttyS0 crw-rw-r-- 1 root dialout 4, 64 Aug 19 18:56 /dev/ttyS0</pre> |
|---|---|

El primer carácter en la primera columna arriba, es decir “c” en `crw-rw-rw-`, le indica el tipo de archivo, en este caso un dispositivo de caracteres. Para archivos comunes el primer carácter es “-”, para directorios es “d” y para dispositivos de bloques es “b”. Nótese que usualmente podrían existir archivos de dispositivos que no necesariamente están presentes en el sistema. Sin embargo, existe un programa que se encarga de la creación y gestión automática de los archivos de dispositivo llamado `udev`⁹, que solo crea los explícitamente necesarios para que el sistema funcione, añadiendo también, los archivos dispositivo correspondientes al hardware presente en el sistema.

Dispositivos IDE en Linux

Cada disco rígido es representado por un archivo de dispositivo separado. Los archivos de dispositivos para los discos rígidos IDE son `/dev/hda`, `/dev/hdb`, `/dev/hdc`, y `/dev/hdd`, respectivamente.

⁹ <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>



Dispositivos SCSI y SATA en Linux

Los archivos de dispositivos para los discos rígidos SCSI/SATA son `/dev/sda`, `/dev/sdb`, etc.

Para las unidades ópticas, `udev` normalmente crea `/dev/sr0`, `/dev/sr1` y así sucesivamente por cuantas unidades ópticas existan en el sistema.



Tema 2: Particiones de disco.

Un disco duro puede dividirse en varias partes, normalmente llamadas particiones. Cada partición funciona como si fuera un espacio aislado del disco duro independiente. La idea es que si solo se tiene un disco, y se quieren tener, digamos, dos sistemas operativos en él, se pueda dividir el disco en dos particiones. Cada sistema operativo utilizará su propia partición tal y como se desea, y, a menos de que lo deseemos, no tocará la otra. De esta forma los dos sistemas operativos pueden coexistir pacíficamente en el mismo disco duro. Sin particiones se tendría que comprar un disco duro para cada sistema operativo.

Los disquetes generalmente no se particionan. No hay ninguna razón técnica para ello, pero dado que son tan pequeños, particionarlos sería útil solo en extrañas ocasiones, así como los CD-ROM tampoco se suelen particionar, ya que es más fácil utilizarlos como un disco grande, y raramente existe la necesidad de tener varios sistemas operativos en uno de ellos.

Tabla de particiones.

La tabla de particiones en los sistemas basados en procesadores Intel x86 está alojada en el MBR (del inglés Master Boot Record) a partir del byte 446 del sector de arranque y ocupa 64 bytes, conteniendo 4 registros de 16 bytes, los cuales definen la particiones primarias. En ellos se almacena toda la información básica sobre la partición: si es arrancable, si no lo es, el formato, el tamaño y el sector de inicio.



Tipos de particiones.

Particiones primarias y lógicas

El esquema original de particionamiento para discos duros de PC permitía solamente cuatro particiones. Esto rápidamente se volvió demasiado escaso para la vida real, en parte porque algunas personas querían más de cuatro sistemas operativos (Linux, MS-DOS, OS/2, Minix, FreeBSD, NetBSD, o Windows/NT,mpor nombrar algunos), pero principalmente porque algunas veces es buena idea tener varias particiones para un sistema operativo. Por ejemplo, el espacio swap (memoria virtual) está generalmente mejor colocado para Linux en su propia partición en lugar de la partición principal por cuestiones de rapidez.

Para superar este problema de diseño, se inventaron las particiones extendidas. Este truco permite particionar una partición primaria en sub-particiones. Esta partición primaria subdividida es la partición extendida; las sub-particiones son las particiones lógicas. Se comportan como particiones primarias, pero se crean de diferente manera. No existen diferencias de rendimiento entre ellas.

La estructura de particiones de un disco duro debe parecerse a la que aparece en la siguiente Figura, “A sample hard disk partitioning.”. El disco se divide en tres particiones primarias, la segunda de las cuales se divide en dos particiones lógicas. Una parte del disco no está particionada. El disco como un todo y cada partición primaria tienen un sector de arranque.



Creación de particiones

Existen muchos programas para crear y eliminar particiones. La mayoría de sistemas operativos tienen el suyo propio, y es buena idea utilizar el propio con cada sistema operativo, por si se diera el caso que haga algo inusual que los otros no puedan hacer. Muchos de estos programas se llaman fdisk, incluido el de Linux, o variaciones de esa palabra. Los detalles de uso del fdisk de Linux se dan en su página de manual. El comando cfdisk es similar a fdisk, pero tiene una interfaz más amigable.

Anteriormente cuando se utilizaban discos EIDE, la partición de arranque (la partición con los archivos de imagen del núcleo arrancable) debía estar completamente definida en los primeros 1024 cilindros del disco. Esto era así por una limitación implícita de los BIOS antiguos, sin embargo, esta limitación no existe en cualquier computador moderno (posterior a 1999).

Cada partición debe tener un número par de sectores, puesto que el sistema de archivos Linux utiliza un tamaño de bloque de 1 kilobyte, es decir, dos sectores. Un número impar de sectores provocará que el último no pueda utilizarse. Esto no es ningún problema, pero resulta feo, y algunas versiones de fdisk avisarán de ello.

Cambiar el tamaño de una partición requiere que primero se realice una copia de seguridad de todo lo que quiera salvar de esa partición (a ser posible de todo el disco, por si acaso), borrar la partición, crear una nueva, y entonces restaurarlo todo a la nueva partición. Si la partición crece, puede necesitar ajustar los tamaños (y guardar y restaurar) las particiones adjuntas también.

Como cambiar una partición es un proceso que debe hacerse con cautela, es preferible establecer las particiones correctamente al principio, o tener un rápido y fácil de utilizar sistema de copia de seguridad. Si está instalando desde un medio que no requiere demasiada intervención humana (digamos, desde un CD-ROM), es generalmente



más fácil probar con diferentes configuraciones al principio. Como no tiene todavía datos que guardar, no es tan costoso el modificar particiones varias veces.

Es importante resaltar que existe un programa muy útil para Linux llamado ntfs-resize que pertenece al proyecto Linux-ntfs. Puede redimensionar de manera segura y no destructiva particiones de tipo NTFS. Además soporta todas las versiones de NTFS y trabaja aún cuando el sistema de archivos se encuentre fragmentado.



Tema 3: Manejo de volúmenes lógicos.

Cuando se crean particiones para un sistema **GNU/Linux**, se hace un cálculo aproximado del tamaño que vamos a necesitar, pero si por alguna casualidad necesitamos aún más, tendremos que reparticionar el disco, lo cual es un trabajo engorroso.

También se puede presentar el caso en el que necesitemos particiones muy grandes. Los enlaces simbólicos podrían solventar un poco el problema, pero con el tiempo se convertirán en problema que nos hará la vida imposible.

Conceptualización y arquitectura del esquema de almacenamiento.

Volúmenes físicos.

Son discos duros, particiones o cualquier otro dispositivo como un volumen RAID. El acrónimo RAID¹⁰ significa: conjunto de discos redundantes independientes/baratos. La idea que hay detrás de RAID es sencilla, se pueden utilizar varios discos físicos para:

- Guardar la información de forma redundante en ellos, lo que puede proporcionar tolerancia a fallos y/o un mejor rendimiento según la forma en la que se realice
- Obtener un volumen lógico con la capacidad de varios discos físicos independientes

El soporte RAID se puede obtener por software, siendo en este caso el SO el que gestiona los dispositivos físicos. O bien mediante hardware, de manera que una

10 *Redundant array of independent/inexpensive disks*

controladora se encarga de controlar varios discos para presentar al sistema informático un volumen unificado RAID. Para las opciones de intercambio de discos en caliente es necesario soporte desde el hardware.

En Canaima GNU/Linux se incluyen herramientas para la gestión y creación de volúmenes físicos y lógicos para una administración sencilla de particiones fáciles de modificar a través del paquete `lvm2`. Esto permite una gestión efectiva y flexible del espacio que ocupa nuestro sistema.

Para usar LVM solo son necesarias dos etapas:

- Instalar el paquete, o bien utilizar un medio de instalación del SO Linux que soporte LVM
- Crear uno o más volúmenes físicos para nuestros grupos de volúmenes y volúmenes lógicos

Si lo crea al instalar su sistema, el asistente de instalación lo ayudará a crearlo de manera fácil y rápida. Ahora bien, para crearlo con las herramientas de consola, puede usar el siguiente comando (nótese que deberá particionar el disco duro destino con una partición tipo LVM, *tipo hexadecimal: 0x8e* con `cfdisk/fdisk`)




```
# pvcreate /dev/sd(letra de disco)(número de partición)
```

Ejemplo: Crear un volumen físico en el segundo disco duro SATA en su tercera partición:

```
# pvcreate /dev/sdb3
```

Grupos de volúmenes.

Podemos definirlo como una colección de volúmenes lógicos y volúmenes físicos. Cuando tenemos todos nuestros discos preparados estos deben ser asignados a un Grupo de Volumen. Esto lo podemos realizar con la utilidad `vgcreate` de la siguiente manera:

| | |
|---|---|
|  | <code>#vgcreate <nombre> <Volumen_Físico>[<Volumen_Físico>...]</code> |
|---|---|

Al crear un grupo de volumen este se genera en el directorio `/dev` un nuevo directorio con el nombre asignado al grupo volumen, por lo tanto es importante que el nombre de este grupo volumen sea algo que también pueda ser el nombre de un directorio, igual se recomienda que sea algo corto y simple.

Volúmenes lógicos.

Es el equivalente a una partición en un sistema tradicional. Se ve como un dispositivo de bloques que puede contener un sistema de archivos. Para administrar un volumen lógico podemos usar la herramienta LVM, este es un gestor de volúmenes lógicos para el núcleo Linux inspirado en VVM.

No incluye funciones RAID1 o RAID5 (más si de RAID0) por lo que normalmente se utiliza encima de volúmenes RAID.

Algunas de las características de LVM son:



- Redimensionado de volúmenes lógicos
- Redimensionado de grupos de volúmenes
- Instantáneas de solo lectura con LVM1, o lectura/escritura con LVM2
- RAID 0 de volúmenes lógicos

Operaciones comunes con volúmenes lógicos.

Una de las características más importantes de trabajar con volúmenes lógicos es que estos pueden ser redimensionados. Cuando se redimensionan volúmenes lógicos (LV) es necesario redimensionar también el sistema de archivos que contienen. Si no se quiere perder información es muy importante no tener un volumen lógico menor que el sistema de archivos que contiene.

Al extender el volumen lógico:

Primero se extiende el volumen lógico y luego el sistema de archivos contenido

Al reducir el volumen lógico:

Primero se reduce el sistema de archivos y luego el volumen lógico

Sobre los sistemas de archivos, debe tenerse en cuenta que:

EXT2/3/4: Puede extenderse o reducirse, pero mientras se realiza la operación el sistema de archivos no debe estar montado.

ReiserFS: Puede extenderse o reducirse (por el inicio o el final) y el sistema de archivos puede estar montado o desmontado.

XFS y JFS: Sólo es posible extender estos sistemas de archivos, y solo es posible hacerlo si el sistema de archivos está montado. Además hay que referirse al sistema de



archivos por su punto de montaje, en lugar de hacerlo por el nombre del dispositivo.

Respaldo con volúmenes lógicos.

La mayor ventaja de LVM es que proporciona la capacidad de tomar una copia instantánea de volumen (snapshot) de cualquier volumen lógico

Copia instantánea de volúmenes

Esta solución solo funciona si se ha creado la partición con LVM. Una copia de volumen instantánea es un tipo especial de volumen que presenta todos los datos existentes en el volumen al momento de que instantánea fue creada. Esto significa que se pueden hacer copias de ese volumen sin tener que preocuparse acerca de los datos que se puedan estar modificando mientras que la copia de seguridad se está realizando, y así no se tiene que suspender el acceso al volumen de datos cuando la copia de seguridad está en proceso.

Teniendo espacio sobrante en el grupo de volúmenes donde hay el disco del cual queremos hacer una instantánea se debería ejecutar el siguiente comando:



```
#lvcreate -L16G -s -n nombre.snapshot  
/dev/volumegroup/disco
```

En el siguiente ejemplo crearemos una instantánea llamada *var.snap* del disco *var* del grupo de volúmenes con nombre: *local*.



```
#lvcreate -L16G -s -n var.snap /dev/local/var
```

Luego fácilmente podemos hacer copias de nuestra instantánea de la siguiente manera:



```
#dd if=/dev/local/var/var.snap  
of=/backups/backups.img bs=1024
```

Y posteriormente eliminar la instantánea anteriormente creada:



```
#lvremove /dev/local/var/var.snap
```

Tema 4: Sistemas de archivos.

Un sistema de archivos se crea, esto es, se inicia, con el comando mkfs. Existen en realidad programas separados para cada tipo de sistemas de archivos. mkfs es únicamente una careta que ejecuta el programa apropiado dependiendo del tipo de sistemas de archivos deseado. El tipo se selecciona con la opción -t fstype.

Los programas a los que -t fstype llama tienen líneas de comando ligeramente diferentes. Las opciones más comunes e importantes se resumen más abajo.

| Opción | Descripción |
|--------|--|
| -t | fstype, selecciona el tipo de sistema de archivos. |
| -c | busca bloques defectuosos e inicia la lista de bloques defectuosos en consonancia. |
| -l | filename, lee la lista inicial de bloques defectuosos del archivo dado. |

Descripción de sistemas de archivos comunes.

Los sistemas de archivos indican el modo en que se gestionan los archivos dentro de las particiones. Según su complejidad tienen características como previsión de apagones, posibilidad de recuperar datos, indexación para búsquedas rápidas, reducción de la fragmentación para agilizar la lectura de los datos, etc. Hay varios tipos, normalmente ligados a sistemas operativos concretos. A continuación se listan los más representativos:



| Tipo | Descripción |
|---------------------|--|
| fat32 o vfat | Es el sistema de archivos tradicional de MS-DOS y las primeras versiones de Windows®. Por esta razón, es considerado como un sistema universal, aunque padece de una gran fragmentación, no es posible crear archivos de un tamaño mayor a 4 GB y no soporta nombres de archivo en codificaciones diferentes a las soportadas por el SO para el cual fue creado. |
| ntfs | Es un sistema de archivos moderno de Windows®, con auditoría y soporte a múltiples codificaciones de caracteres. |
| ext2 | Hasta hace unos años era el sistema estándar de Linux. Tiene una fragmentación bajísima, aunque sufre de bajo rendimiento con archivos de gran tamaño. |
| ext3 | Es la versión mejorada de ext2, con previsión de pérdida de datos por fallos del disco o apagones. Es compatible con el sistema de archivos ext2. Actualmente es el más difundido dentro de la comunidad GNU/Linux y considerado el estándar de facto. |
| ext4 | Es el sistema de archivos de última generación para Linux. Organiza los archivos de tal modo que se agilizan mucho las operaciones de disco. |
| swap | Sistema de archivos de intercambio (o memoria virtual) de los sistemas operativos basados en Linux. |


Creación de sistemas de archivos.



Creación de una partición

Antes de crear una partición, arranque en modo de rescate (o desmonte cualquier partición en el dispositivo y elimine cualquier espacio swap).

Inicie parted, donde /dev/sdb es el dispositivo en el que se crea la partición:


| | |
|---|-------------------------------|
|  | <code>#parted /dev/sdb</code> |
|---|-------------------------------|

Visualice la tabla de particiones actual para determinar si hay suficiente espacio libre con el comando print

Si no hay suficiente espacio libre, puede cambiar el tamaño de partición ya existente.

Ahora desde la tabla de particiones, determinaremos los puntos de comienzo y fin de la nueva partición y qué tipo de partición debe ser. Puede tener solamente cuatro particiones primarias (sin partición extendida) en un dispositivo. Si necesita más de cuatro particiones, puede tener tres particiones primarias, una partición extendida y varias particiones lógicas dentro de la extendida.

Por ejemplo, para crear una partición primaria con un sistema de archivos ext3 desde 1024 megabytes hasta 2048 megabytes en un disco duro, escribimos el siguiente comando:

| | |
|---|--|
|  | |
|---|--|



```
#mkpart primary ext3 1024 2048
```

Los cambios se harán efectivos tan pronto como presionemos Enter, por tanto debemos revisar bien el comando antes de ejecutarlo.

Después de crear la partición, usaremos el comando print para confirmar que está en la tabla de particiones con el tipo de partición, tipo de sistema de archivos y tamaño correctos. Debemos recordar también el número minor de la nueva partición, de modo que se pueda etiquetar. También deberíamos poder visualizar la salida de :



```
#cat /proc/partitions
```

Para asegurarnos de que el kernel reconoce la nueva partición.

Hasta ahora la partición no tiene todavía un sistema de archivos. Ahora crearemos el sistema de archivos:



```
#mkfs.ext3 /dev/sdb1
```

Al formatear la partición se destruirán permanentemente los datos que existan en la partición.

A continuación, asignaremos una etiqueta a la partición. Por ejemplo, si la nueva



partición es /dev/sda1 y queremos etiquetarla /trabajo, hacemos:



```
#e2label /dev/sdb1 /trabajo
```

Posteriormente como usuario root, crearemos un punto de montaje:



```
#mkdir /trabajo
```

Como root, editaremos el archivo /etc/fstab para incluir la nueva partición. La nueva línea debe ser parecida a la siguiente:




```
LABEL=/trabajo /trabajo ext3 defaults  
1 2
```

La primera columna debe contener LABEL= seguida de la etiqueta que usted dio a la partición. La segunda columna debe contener el punto de montaje para la nueva partición y la columna siguiente debería ser el tipo de sistema de archivo (por ejemplo, ext3 o swap).

Si la cuarta columna es la palabra defaults, la partición se montará en el momento



de arranque. Para montar la partición sin arrancar de nuevo, como root, escriba el comando:


| | |
|---|-----------------------------|
|  | <code>mount /trabajo</code> |
|---|-----------------------------|

Manipulación de sistemas de archivos.

Redimensionamiento de una partición

Antes de cambiar el tamaño a una partición, arranque en modo de rescate (o desmonte cualquier partición en el dispositivo y elimine cualquier espacio swap en el dispositivo).

Arrancaremos parted, donde /dev/sdb es el dispositivo en el cual se redimensionará la partición:


| | |
|---|-------------------------------|
|  | <code>#parted /dev/sdb</code> |
|---|-------------------------------|

Visualicemos la tabla de particiones actual para determinar el número minor¹¹ de la partición que se quiere redimensionar, así como los puntos de comienzo y fin para la partición con el comando print

¹¹ Número de partición dentro del dispositivo de bloques

Es importante acotar que el espacio usado de la partición que se quiere redimensionar no puede ser mayor que el nuevo tamaño.

Para redimensionar la partición, use el comando `resize` seguido del número minor de la partición, el lugar comienzo y fin en megabytes. Por ejemplo:

| | |
|---|---------------------------------|
|  | <code>resize 1 1024 2048</code> |
|---|---------------------------------|

Después de cambiar el tamaño a la partición, se usa el comando `print` para confirmar que se ha cambiado el tamaño de la partición correctamente, que es el tipo de partición y de sistema de archivos correcto.

Después de reiniciar el sistema el modo normal, usamos el comando `df` para asegurarnos que la partición fue montada y que es reconocida con el nuevo tamaño.

Montaje de sistemas de archivos.

Ya se ha visto que Linux accede a los dispositivos mediante archivos (archivos bajo `/dev`), y, por este motivo, en Linux no hay el concepto de unidades, ya que todo está bajo el directorio principal o una rama de este. En Linux no se accede a la primera disquetera mediante la orden `A:` como en DOS sino que hay que montarla.

De este modo, tenemos dos conceptos nuevos:

- montar: informarle al núcleo que se utilizará un determinado dispositivo con un

determinado sistema de archivos y estará en un directorio especificado.

- Desmontar: informarle al núcleo que se ha dejado de utilizar un determinado dispositivo.

En la siguiente tabla se muestran los sistemas de archivos de uso más común en Linux:

| TIPO | DESCRIPCIÓN |
|-------------|---|
| ext3 o ext2 | Sistema de archivos de Linux. |
| vfat | Sistema de archivos de Windows 9X (nombres largos). |
| iso9660 | Sistema de archivos de CD-ROM. |
| nfs | Sistema de archivos compartido por red ("exportado"). |

Para montar un determinado sistema de archivos de un dispositivo, se utiliza el comando mount. La sintaxis es la siguiente: `mount -t sistema_archivos dispositivo directorio [-o opciones]`, donde:

- *sistema_archivos*, puede ser cualquiera de los que aparece en la tabla anterior.
- *dispositivo*, puede ser cualquier dispositivo del directorio /dev o, en el caso de nfs, un directorio de otro PC.
- *directorio*, es el directorio donde estará el contenido del dispositivo.
- *opciones* pueden ser cualquiera de la tabla siguiente, en el caso de no poner ninguna opción, mount utilizará las opciones por defecto

| OPCIÓN | DESCRIPCIÓN |
|--------|-----------------------|
| rw | Lectura/escritura. |
| ro | Sólo lectura. |
| exec | Se permite ejecución. |

| | |
|-------|--|
| user | Los usuarios pueden "montar"/"desmontar". |
| suid | Tiene efecto los identificadores de propietario y del grupo. |
| auto | Se puede montar automáticamente. |
| async | Modo asíncrono. |
| sync | Modo síncrono. |
| dev | Supone que es un dispositivo de caracteres o bloques. |

Una vez montado el dispositivo, si no se va a volver utilizar se puede desmontarlo con el comando `umount` con la siguiente sintaxis: `# umount directorio`

Siempre, después de utilizar un dispositivo hay que desmontarlo, para que asegurar que cualquier cambio diferido se escriba al dispositivo. Un ejemplo de ello, es el hecho de que, una memoria Flash a la que se ha escrito una cantidad considerable de datos podría perderlos si no se le desmonta apropiadamente (ya que tanto el núcleo del sistema operativo como el mismo hardware encola las operaciones de escritura para que el proceso termine más rápido mientras los datos se escriben en segundo plano para no causar demoras notables a la sesión del usuario).

Se muestran algunos ejemplos:



Disquete de DOS

```
#mount -t msdos /dev/fd0 /mnt/floppy -o  
rw,noexec
```

```
#umount /mnt/floppy
```

Disquete de Windows 9X

```
#mount -t vfat /dev/fd0 /mnt/floppy -o user,rw  
#umount /mnt/floppy
```

CD-ROM

```
#mount -t iso9660 /dev/cdrom /mnt/cdrom -o ro  
#umount /mnt/cdrom
```



Directorio exportado de host2 por NFS

```
#mount -t nfs host2:/tmp /mnt/host2
#umount /mnt/host2
```

Archivo /etc/fstab

Lista los sistemas de archivos montados automáticamente en el arranque del sistema por el comando `mount -a`. En Linux, este archivo también contiene información acerca de áreas de intercambio que se activan a través del comando especial `swapon`

El primer campo, llamado *fs_spec*, describe el dispositivo especial de bloque o sistema de archivos remoto a ser montado.

El segundo campo, *fs_file*, describe el punto de montaje para el sistema de archivos. Para particiones de intercambio (tipo: swap), este campo debe decir "none".

El tercer campo, *vfs_vfstype*, describe el tipo del sistema de archivos. Y puede contener cualquier sistema de archivos que el núcleo soporte y sea automáticamente montable, entre ellos: minix, ext2, ext3, Raiserfs, msdos, hpfs, iso9660, nfs, swap.

Si *vfs_vfstype* tiene el valor "ignore", la entrada se ignorará. Esto es útil para ver aquellas particiones del disco que no están siendo usadas.

El cuarto campo, llamado *fs_mntops*, describe las opciones de montaje asociadas con el sistema de archivos. Es una lista de opciones separadas por comas. Contiene como mínimo el tipo de sistema de archivos y otras opciones apropiadas para el tipo del sistema de archivos.



Las distintas opciones para sistemas de archivos locales están documentadas en la página del manual de mount . Las opciones específicas para cada uno de los sistemas de archivos están listadas en su respectiva página del manual.

El quinto campo, *fs_freq*, lo utiliza el comando dump para determinar qué sistemas de archivos necesitan ser volcados para respaldo. Si el quinto campo está vacío, se asume que el sistema de archivos no necesita ser volcado.

El sexto campo, *fs_passno*, lo usa el programa fsck¹² para determinar el orden en el cual se van a chequear los sistemas de archivos cuando el sistema arranca. El sistema de archivos raíz debería llevar *fs_passno* igual a 1, y otros sistemas de archivos deberían llevar *fs_passno* igual a 2. Los sistemas de archivos que residan en un mismo disco serán comprobados de manera secuencial, pero aquellos sistemas de archivos que se encuentren en diferentes discos serán comprobados de forma paralela. Si el sexto campo no está presente o tiene un valor igual a 0, fsck asumirá que los sistemas de archivos no necesitan ser chequeados.

La forma apropiada de leer los registros de fstab es usando las rutinas getmntent().

¹² *Filesystem Check*, Chequeo del sistema de archivos, por su acrónimo.



Tema 5: Cuotas de disco.

Una cuota de disco es la serie de parámetros y limitaciones que se asignan para el uso sobre ciertos recursos. Existen básicamente dos tipos de cuotas:

- Cuota de uso o de bloques: limita la cantidad de espacio en disco que puede ser utilizado.
- Cuota de archivos o i-nodos¹³: limita la cantidad de archivos y carpetas que pueden ser creadas.

Asimismo, usualmente se establecen dos niveles en la administración de las cuotas:

- Cuota suave: este nivel se establece como un nivel de advertencia, en donde el usuario (o grupos) son informados de que están próximos a alcanzar su límite.
- Cuota dura: este nivel es el límite de capacidad asignado al usuario o grupo, aún cuando se puede establecer un período de gracia (usualmente 7 días) en los que se pueden exceder estas cuotas.

Las cuotas se definen por usuario o por grupos, y son controladas por el administrador del sistema. Estas cuotas se establecen para que los usuarios no hagan uso desmedido de los recursos que se les proporciona, optimizando estos y reduciendo los costos asociados al despilfarro de capacidad.

Los sistemas de archivos manejados por Linux y Unix generalmente soportan la implementación de cuotas de disco. Entre estos encontramos a ext2, ext3, ext4, reiserfs, ufs, etc. Los sistemas de archivos msdos (fat12, fat16, fat32/vfat) No soportan la implementación de cuotas, a diferencia de ntfs.

¹³Estructura de datos básica del sistema de archivos que almacena datos sobre un archivo, directorio o socket.

Regularmente las cuotas se asignan por cada sistema de archivos, de manera independiente. Dependiendo del tipo de gestión que desee implementarse, se habilitarán o no las cuotas en distintos sistemas de archivos.

Activación de cuotas en un punto de montaje

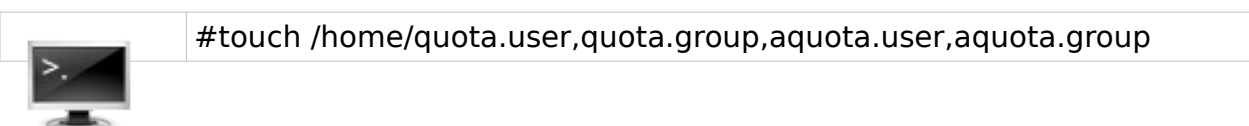
Para habilitar el soporte para cuotas en un sistema de archivos determinado, deberemos asegurarnos de que ningún proceso esté haciendo uso de dicho recurso. Para asegurarnos de que este requisito sea cubierto, deberemos iniciar nuestro sistema en nivel de ejecución monousuario.



Aún cuando podemos cambiarnos del actual nivel de ejecución hacia el nivel monousuario (init 1), esto no nos asegura que no haya algún proceso haciendo uso del recurso. Es por ello que se recomienda reiniciar el equipo e iniciarlo en dicho nivel de ejecución.

Por cada sistema de archivos al que se desee implementar la cuota, se deberán crear 4 archivos: quota.user, quota.group, aquota.user, aquota.group. Estos archivos serán los responsables de administrar las cuotas y llevar los índices de los archivos de los usuarios y grupos.

Suponiendo que tenemos 3 distintos sistemas de archivos a los que queremos aplicar las cuotas (/home, /var, /deptos), deberemos crear los 4 archivos arriba mencionados en cada punto de montaje:



```
#touch /var/quota.user,quota.group,aquota.user,aquota.group  
#touch /deptos/quota.user,quota.group,aquota.user,aquota.group
```

Si queremos optimizar la ejecución de dichos comandos, y se tiene un poco más de pericia sobre el manejo de bash, podremos resumir en una línea como sigue:



```
# touch /{home,var,deptos}/{,a}quota.{user.group}
```

En el archivo `/etc/fstab`, se debe especificar el montaje de dichos sistemas de archivos con las opciones de cuotas de usuario y/o cuotas de grupo:

- Cuota de uso o de bloques: limita la cantidad de espacio en disco que puede ser utilizado.
- Cuota de archivos o inodos: limita la cantidad de archivos y carpetas que pueden ser creadas.

Asimismo, usualmente se establecen dos niveles en la administración de las cuotas:



```
/dev/sda3          /home ext3 defaults,usrquota 1 1  
LABEL=/var        /var ext3 defaults,usrquota,grpquota 1 1  
/dev/mapper/VG01/LV01 /deptos ext3 defaults,grpquota 1 1
```

Debemos volver a montar dichos sistemas de archivos:



```
# mount -o remount /home  
# mount -o remount /var
```

mount -o remount /deptos

Una vez hecho esto, verificamos las opciones de montaje mediante el comando mount. Los archivos que generamos, en este momento no son mas que simples archivos en blanco, tenemos que convertirlos a un formato de cuotas:



quotacheck -ugav

| Comando | Descripción |
|---------|--|
| u | Activa las cuotas de usuarios |
| g | Activa las cuotas de grupos |
| a | Verifica la creación de cuotas en todos los sistemas de archivos con soporte para estas |
| v | Muestra una salida detallada de la ejecución del mandato. Es usual ver que el sistema nos envía un mensaje de advertencia cuando ejecutamos este mandato por primera vez, ya que se están generando los índices. |

Activamos las cuotas en los puntos de montaje especificados:



quotaon /home
quotaon /var
quotaon /deptos

Listo. Con esto, ya tenemos capacidad para la administración de cuotas de disco en nuestro sistema.

Manipulación de cuotas.



1. Debe iniciarse el sistema en nivel de corrida 1 (mono usuario), ya que **se requiere no haya procesos activos** utilizando contenido de la partición a la cual se le aplicará la cuota de disco.

2. Durante la instalación, debió asignarse una partición dedicada para, por mencionar un ejemplo, los directorios /var y /home.

3. Con la finalidad de añadir el soporte para cuotas en las particiones anteriormente mencionadas, se debe añadir en el fichero **/etc/fstab** los parámetros **usrquota** y **grpquota** a las líneas que definen la configuración de las particiones /var y /home:



```
LABEL=/var /var ext3 defaults,usrquota,grpquota 1  
2  
LABEL=/home /home ext3  
defaults,usrquota,grpquota 1 2
```

4. Debe remontar las particiones para que surtan efecto los cambios:
`#mount -o remount /var`

```
#mount -o remount /home
```

5. Se deben crear los ficheros `aquota.user`, `aquota.group`, `quota.user` y `quota.group`, los cuales se utilizarán en adelante para almacenar la información y estado de las cuotas en cada partición.

```
#cd /var
```

```
#touch aquota.user aquota.group quota.user quota.group
```

```
#cd /home
```

```
#touch aquota.user aquota.group quota.user quota.group
```

```
#quotacheck -avug
```



6. La primera vez que se ejecuta el mandato anterior es normal marque advertencias refiriéndose a posibles ficheros truncados que en realidad no eran otra cosa sino ficheros de texto simple vacíos a los cuales se les acaba de convertir en formato binario. Si se ejecuta de nuevo `quotacheck - avug`, no deberá mostrar advertencia alguna.

7. Para activar las cuotas de disco recién configuradas, solo bastará ejecutar:

```
#quotaon /var
```

```
#quotaon /home
```



8. Vaya al nivel de corrida 3 a fin de aplicar cuota de disco a algunos usuarios.


```
#init 3
```

Edquota

Es importante conocer que significa cada columna mostrada por edquota.


- **Blocks(Bloques):** Corresponde a la cantidad de bloques de 1 Kb que está utilizando el usuario.
- **Inodes (Inodos):** Corresponde al número de ficheros que está utilizando el usuario. Un inodo (también conocido como Index Node) es un apuntador hacia sectores específicos de disco duro en los cuales se encuentra la información de un fichero. Contiene además la información acerca de permisos de acceso así como los usuarios y grupos a los cuales pertenece el fichero.
- **Soft (Límite de gracia):** Límite de bloques de 1 KB que el usuario puede utilizar y que puede rebasar hasta que sea excedido el período de gracia (de modo predeterminado son 7 días).
- **Hard (Límite absoluto):** Límite que no puede ser rebasado por el usuario bajo circunstancia alguna.

Asignar cuotas de disco a cualquier usuario o grupo solo hará falta utilizar edquota citando el nombre del usuario al cual se le quiere aplicar:

| | |
|---|------------------------------|
|  | <code>#edquota fulano</code> |
|---|------------------------------|


Lo anterior deberá devolver algo como lo siguiente a través de vi u otro editor de texto simple:



| | | | | | |
|---|--|--------|------|------|--------|
|  | Disk quotas for user fulano (uid 501): | | | | |
| | Filesystem | blocks | soft | hard | inodes |
| | soft | hard | | | |
| | /dev/hda7 | 0 | 0 | 0 | 0 |
| | 0 | 0 | | | |
| | /dev/hda5 | 24 | 0 | 0 | 10 |

Cuota absoluta

Suponiendo que se quiere asignar una cuota de disco de 6 MB para el usuario «fulano» en en /dev/hda7 y /dev/hda5, se utilizaría lo siguiente:

| | | | | | | | |
|---|--|--------|------|------|--------|------|------|
|  | Disk quotas for user fulano (uid 501): | | | | | | |
| | Filesystem | blocks | soft | hard | inodes | soft | hard |
| | /dev/hda7 | 0 | 0 | 6144 | 0 | 0 | 0 |
| | /dev/hda5 | 24 | 0 | 6144 | 10 | 0 | 0 |

El usuario siempre podrá rebasar una cuota de gracia pero nunca una cuota absoluta.

Cuota de gracia.

El sistema tiene de modo predeterminado un período de gracia de 7 días que se puede modificar con el mandato `edquota -t`, donde se puede establecer un nuevo período de gracia por días, horas, minutos o segundos.



```
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/hdb7                7days                    7days
/dev/hdb5                7days                    7days
```

La cuota de gracia establece los límites de bloques o i-nodos que un usuario tiene en una partición. Cuando el usuario excede el límite establecido por la cuota de gracia, el sistema advierte al usuario que se ha excedido la cuota del disco sin embargo permite al usuario continuar escribiendo hasta que transcurre el tiempo establecido por el período de gracia, tras el cual al usuario se le impide continuar escribiendo sobre la partición. Suponiendo que quiere asignar una cuota de gracia de 6 MB en /dev/hda7 y /dev/hda5, la cual podrá ser excedida hasta por 7 días, se utilizaría lo siguiente:



```
Disk quotas for user fulano (uid 501):
Filesystem      blocks      soft      hard      inodes      soft
hard
/dev/hda7                0      6144                0                0
0
/dev/hda5             24      6144                0              10                0
0
```

Aplicando cuotas masivamente

Si se quiere que todo aplique para los usuarios existentes, a partir de UID 510, por ejemplo, suponiendo que tiene al usuario "pepito" como molde (note por favor el acento grave en el mandato justo antes de awk, no es una comilla ni apostrofe):



```
edquota -p pepito `awk -F: '$3 > 510 {print $1}' /etc/passwd`
```

Comprobaciones

Utilice el mandato edquota con el usuario fulano.



```
#edquota fulano
```

Asigne al usuario «fulano» una cuota de disco de 50 MB en todas las particiones con cuota de disco habilitada:



```
Disk quotas for user fulano (uid 501):
```

| Filesystem | blocks | soft | hard | inodes | soft |
|------------|--------|------|-------|--------|------|
| hard | | | | | |
| /dev/hda7 | 0 | 0 | 51200 | 0 | 0 |
| /dev/hda5 | 24 | 0 | 51200 | 10 | 0 |

Desde otra terminal acceda hacia el sistema como el usuario fulano y ejecute el mandato quota y observe con detenimiento la salida:



```
Disk quotas for user fulano (uid 501):
```

| Filesystem | blocks | quota | limit | grace | files | quota | limit |
|------------|--------|-------|-------|-------|-------|-------|-------|
| /dev/hda7 | 0 | 0 | 51200 | | 1 | 0 | 0 |
| /dev/hda5 | 24 | 0 | 51200 | | 10 | 0 | 0 |

Realice una copia del directorio `/usr/lib` como el subdirectorio `~/prueba-cuotas` dentro de su directorio de inicio:



```
#cp -r /usr/lib ~/prueba-cuotas
```

Notará que llegará un momento en el que el sistema indicará que ya no es posible continuar copiando contenido dentro de `~/prueba-cuotas` debido a que se ha agotado el espacio asignable al usuario en la partición.

Utilice de nuevo el mandato `quota` y observe con detenimiento la salida, en donde aparecerá un asterisco justo junto a la cantidad en la columna de bloques bloques, el cual indica que se ha excedido la cuota del disco:



Disk quotas for user fulano (uid 501):

| Filesystem | blocks | quota | limit | grace | files | quota | limit |
|------------|--------|-------|-------|-------|-------|-------|-------|
| /dev/hda7 | 0 | 0 | 51200 | | 1 | 0 | 0 |
| /dev/hda5 | 51200* | 0 | 51200 | | 7439 | 0 | 0 |

Para poder volver a escribir sobre la partición, es necesario liberar espacio. Elimine por completo el directorio `~/prueba-cuotas` y vuelva a utilizar el mandato `quota`:



```
#rm -fr ~/prueba-cuotas  
#quota
```



¿Qué es una cuota de disco?

Además de controlar el espacio en disco usado por el sistema, el almacenamiento en disco se puede restringir mediante la implementación de cuotas de disco y de esta manera el administrador es notificado antes de que un usuario consuma mucho espacio en disco o que una partición se llene.

Las cuotas se pueden configurar para usuarios individuales o para grupos. Este tipo de flexibilidad hace posible darle a cada usuario una pequeña porción del disco para que maneje sus archivos personales (tales como correo o informes), mientras que se le permite tener más espacio para manejar los proyectos en los que estén trabajando o cuotas más grandes (asumiendo que a los proyectos se les da sus propios grupos). Además, se puede configurar las cuotas no solo para que controlen el número de bloques de disco pero también el número de inodes. Debido a que los inodes son usados para contener información relacionada a los archivos, esto permite controlar el número de archivos que pueden ser creados.



Tema 6. Permisos sobre el sistema de archivos.

Los permisos de los sistemas de archivos en los sistemas basados en GNU/Linux, así como en los sistemas de filosofía Unix se manejan en tres clases bien distintas. Las clases son conocidas como usuario (*user*), grupo (*group*) y otros (*others*). En efecto, como puede apreciarse, la permisología sobre los archivos en este tipo de sistemas son una forma simplificada de las listas de control de acceso (ACLs).

Cuando se crea un nuevo archivo en esta familia de sistemas, sus permisos son determinados a través de la función *umask* del proceso que lo creó.

Umask

Umask¹⁴ es tanto un comando como una función en los ambientes POSIX¹⁵ que define **la máscara del modo de creación de archivos** del proceso en curso. La máscara del modo de creación de archivos (también conocida en si misma como la “umask”) limita los modos de permisología para archivos y directorios que son subsecuentemente creados por el proceso. Cuando un intérprete de comandos u otro programa está creando un archivo o un directorio, especifica la permisología que será entregada, el sistema operativo entonces elimina de esos permisos aquellos que la máscara del modo de creación de archivo no permite.

14 Forma abreviada en inglés: *user mask*, que significa: “máscara de usuario”

15 Siglas en inglés de: Portable Operating System Interface for Unix, una serie de estándares para sistemas basados en UNIX y sus similares, estándares que Canaima GNU/Linux cumple.



Permisos básicos

Existen tres permisologías específicas básicas en los sistemas GNU/Linux que aplican a cada una de las clases mencionadas anteriormente:

- La permisología de lectura (*read*), la cual concede los permisos para la lectura de un archivo. Cuando se aplica a un directorio, esta permisología concede la posibilidad de leer los nombres de los archivos contenidos en él (mas no obtener información adicional sobre los mismos, como: tipo, tamaño, permisología y dueño, entre otras)
- La permisología de escritura (*write*), que concede los permisos para modificar un archivo. Cuando es aplicada a un directorio, esta permisología da a el usuario o al proceso el derecho de modificar las entradas del directorio. Esto incluye creación, borrado y renombrado de archivos.
- La permisología de ejecución (*execute*), la cual concede los permisos para ejecutar un archivo. Esta permisología debe ser asignada a los binarios ejecutables (por ejemplo, un programa en C++ compilado) o para los scripts del intérprete de comandos (un programa en Perl, por ejemplo) de manera de permitir que el sistema operativo lo ejecute. Cuando esta permisología es asignada a un directorio, concede la posibilidad de recorrer en forma transversal su árbol para tener acceso a los archivos o subdirectorios contenidos en este, más no ver los archivos en el directorio (a menos de que la permisología de lectura esté activa).

Cuando una permisología específica no está asignada, los derechos que otorgaría son denegados. A diferencia de los sistemas que basan sus permisologías en listas de control de acceso, las permisologías de un sistema basado en GNU/Linux no son heredables, quiere decir, por ejemplo, que los archivos creados dentro de un directorio no necesariamente compartirán los mismos permisos que el directorio donde se encuentran.



Estas permisologías a ser asignadas son determinadas por las máscaras de modo de creación de archivos o umasks.

Notación de las permisologías en sistemas tipo GNU/Linux

Notación simbólica

Existen una diversidad de métodos por las cuales los esquemas de permisos son representados. La forma más común es la notación simbólica.

Cada clase de permisos está representado por una dupla de tres caracteres. El primer conjunto representa a la clase de usuario, o propietario. El segundo conjunto representa al grupo de usuarios. El tercero y último representa la clase de acceso por otros usuarios.

Cada uno de estos tres caracteres representan los permisos de lectura, escritura y ejecución, respectivamente:

- **r**: si el bit de lectura está activado, de lo contrario será “-”
- **w**: si el bit de escritura está activado, de lo contrario será “-”
- **x**: si el bit de ejecución está activado, de lo contrario será “-”

Los siguientes son algunos ejemplos de la notación simbólica:

- ➔ **-rwxr-xr-x** para un archivo regular cuya clase de usuario tiene permisos completos sobre el archivo, mientras que sus clase de grupo y otros solo tienen permisos de lectura y ejecución.
- ➔ **crw-rw-r--** para un archivo especial de caracteres cuyas clases de usuario y



grupo tienen permisos de lectura y escritura, mientras que su clase otros solo tiene permisos de lectura.

- dr-x----- para un directorio cuya clase de usuario tiene permisos de lectura y ejecución y las otras clases no tienen ningún tipo de permisología asignada.
- dr-xr-xr-x para un directorio donde todas las clases tienen permisos de lectura y ejecución.

Notación octal

Otro método común para representar la permisología es la notación octal. La notación octal consiste de un valor de base 8 de tres o cuatro dígitos.

Cuando se utiliza notación octal de tres dígitos, cada numeral representa un componente diferente del conjunto de permisologías: clase de usuario, clase de grupo y clase otros, respectivamente.

Cada uno de estos dígitos es una suma de sus bits componentes. Como resultado, cada bit específico añade a la suma cuando esta es representada como un numeral:

- El bit de lectura suma 4 al total (número binario 100),
- El bit de escritura suma 2 al total (número binario 010)
- El bit de ejecución suma 1 al total (número binario 001).

Estos valores nunca producen combinaciones ambiguas; cada suma representa un conjunto específico de permisos.

Algunos ejemplos de la notación simbólica dados en notación octal:

- "-rwxr-xr-x" sería representado como 755 en octal de tres dígitos



- "-rw-rw-r--" sería representado como 664 en octal de tres dígitos
- "-r-x-----" sería representado como 500 en octal de tres dígitos

A continuación un sumario de los significados para valores de dígito octales individuales:

- 0 --- sin permisos
- 1 --x ejecución
- 2 -w- escritura
- 3 -wx escritura y ejecución
- 4 r-- lectura
- 5 r-x lectura y ejecución
- 6 rw- lectura y escritura
- 7 rwx lectura, escritura y ejecución

Permisos especiales: suid, guid y bit pegajoso

Suid (o bit setuid)

El bit suid sobre los permisos se coloca generalmente en un ejecutable. Cuando un archivo que tiene este permiso asignado se ejecuta, el proceso resultante asumirá la identificación de usuario (o *ID de usuario*) efectiva dada a la clase de usuario. Un ejemplo típico en un sistema GNU/Linux es cuando se desea cambiar una clave de usuario: ningún usuario debería poder modificar el archivo `/etc/passwd` directamente. La única forma de poder modificarlo debería ser a través del comando correspondiente, que necesariamente tendrá que tener asignado el bit setuid. Es decir, el comando `/usr/bin/passwd` ejecutado por un usuario se ejecutará como si lo hubiese invocado el



superusuario, de manera de que este tenga la posibilidad de modificar el archivo `/etc/passwd`.

Gid (o bit setgid)

Realiza la misma operación que **suid** aunque aplicado a la identificación de grupo (*o ID de grupo*) si el setgid le es asignado a un directorio, cualquier archivo nuevo o directorio creado debajo de ese directorio heredará el grupo de ese directorio, a diferencia del comportamiento por defecto, que es usar el grupo primario del usuario efectivo al asignar el grupo de archivos nuevos y directorios. (es decir, que por defecto los permisos no son heredables).

Bit pegajoso

El comportamiento típico del bit pegajoso (inglés: *sticky bit*) en archivos ejecutables obliga al núcleo a retener la imagen del proceso resultante luego de su terminación. Originalmente, esta era una característica para ahorrar memoria, pero hoy en día, existen una multitud de mejores técnicas para lidiar con esa necesidad, que en gran medida fue sobrepasada en los núcleos Linux más recientes, por lo que, en la actualidad no se lo suele utilizar más para los archivos ejecutables, ahora bien, en un directorio, el bit pegajoso evita que los usuarios renombren, muevan o borren los archivos que allí se encuentran, cuando estos no pertenezcan a ellos mismos, inclusive si tienen permisos de grupo que les podrían permitirselo. Únicamente el propietario del directorio y el superusuario quedan exentos de esta restricción.

Chmod

El comando `chmod`¹⁶ (*change mode*) se utiliza para cambiar la permisología sobre archivos y directorios, de cualquier índole, este comando forma parte de las herramientas básicas de cualquier sistema GNU/Linux, y como tal, su uso es uniforme entre sistemas que cumplen la norma POSIX.

Uso de chmod



```
chmod [opciones] modo archivo (o directorio)
```

Opciones de chmod

| Opción | Descripción |
|--------|--|
| -R | Cambia permisos de forma descendente o recursiva en un directorio dado. |
| -c | Muestra que ficheros han cambiado recientemente en una ubicación dada |
| -f | No muestra errores de ficheros o directorios que no se hayan podido cambiar |
| -v | Modo prolijo, donde se da una descripción detallada de los mensajes generados por el proceso |

¹⁶ Acrónimo en inglés de: cambiar modo



Modos en chmod

En el modo se especifica que clase de permisología se cambiará y los permisos que se le asignarán. El modo puede darse de forma simbólica, o bien, de forma octal.

El formado para el modo simbólico es '[ugoa...][[+|=][rwxXstugo...][...][,...]'. Se pueden realizar múltiples operaciones simbólicas a la vez, separando cada una por coma.

La combinación de las letras "ugoa" controla la clase de permisología que se cambiará sobre el archivo o directorio: el del usuario que es propietario del archivo (u), el del grupo (g), otros usuarios (o) o, todas las clases de permisología (a). Si no se especifica ninguna de forma explícita, el comportamiento por defecto es afectar todas las clases (es decir, como si se especificara "a"), sin embargo, los bits que están colocados en la máscara de modo no serán afectados (setuid, setgid, etc.).

El operador *suma* ("+") causa que los permisos seleccionados sean añadidos a los permisos ya existentes para el archivo, el operador *resta* ("-") causa que sean eliminadas y el operador *igual* ("=") causa que sean los únicos permisos que le sean asignados al archivo sobre el cual se está operando.

Las letras "rwxXstugo" seleccionan los nuevos permisos que tendrán las clases de usuario: lectura (r), escritura (w), ejecución (o navegar por el directorio) (x), ejecutar solamente si el archivo es un directorio o bien, si posee ya permisos de ejecución en alguna de las clases establecer setuid o setgid (s), asignar bit pegajoso (t).

Asimismo, chmod puede operar en modo octal, este modo numérico consta de uno a cuatro dígitos octales (0-7), que se derivan de la suma octal como se explicó anteriormente. De omitirse alguno, se presume que está precedido por ceros. Si se desean asignar permisos especiales, debe recordarse que se utilizarán los 4 dígitos completos asignando al principio un número cuatro (4) para asignar setuid, (2) para

asignar setguid y el número uno (1) si se desea asignar el bit pegajoso. Si se especifican solo tres dígitos, se asume operación sobre el propietario del archivo, el grupo y otros usuarios, en ese orden, respectivamente.



El comando `chmod` jamás cambia los permisos de enlaces simbólicos; sin embargo, esto no representa un problema en virtud de que jamás se utilizan los permisos de los enlaces simbólicos. Si se aplica el mandato `chmod` sobre un enlace simbólico, se cambiará el permiso del fichero o directorio hacia el cual apunta. Cuando se aplica `chmod` de forma recursiva en un directorio, este ignora los enlaces simbólicos que pudiera encontrar en el recorrido del árbol de subdirectorios.

Ejemplos



```
# Asigna permisos de lectura, escritura y  
ejecución para la clase "otros" a todos los  
archivos de la carpeta
```

```
chmod o=rwx *
```

```
# Asigna todos los permisos a todos los usuarios  
para el archivo pepe.txt
```

```
chmod a=rwx pepe.txt
```

```
# Quita todos los permisos para los usuarios del  
grupo y para cualquier otro usuario, es decir,
```



para la clase "otros".

```
chmod go= *
```

Da todos los permisos al dueño del fichero, a los del grupo del dueño le asigna permisos de lectura y escritura y a los otros usuarios les quita todos los permisos.

```
chmod u=rwx,g=rw,o= *
```

Esto quita todos los permisos a todos los tipos de usuario.

```
chmod a-wrx *
```

Este comando asigna permisos de lectura a todos los usuarios y permisos de escritura al dueño del archivo y el grupo del dueño.

```
chmod a+r,gu+w *
```

Asigna todos los permisos al archivo archivo.txt

```
chmod 777 archivo.txt
```



```
# Asigna permisos de lectura y escritura,  
(excluyendo ejecución) a todos los archivos y  
directorios del directorio donde ejecutamos el  
comando. (modo octal)
```

```
chmod 0666 *
```

```
# Esto da permisos a todos los archivos y  
directorios del directorio donde se invoca el  
comando y de todos los directorios que cuelgan de  
él. Los permisos asignados son de lectura a todos  
los usuarios, de escritura solo al dueño del  
archivo y de ejecución a nadie. (modo octal)
```

```
chmod -R 0644 *
```

Permisos basados en listas de control de acceso ACL

Este tipo de permisologías sobre sistemas de archivos son aplicables solamente en el caso de que se configure un servidor NFS¹⁷ (*Network File System*). Estas permisologías serán cubiertas en el desarrollo de la unidad IX de este documento.

17 Sistema de archivos de red, por sus siglas en inglés.



UNIDAD VI: Fundamentos de Redes TCP/IP en GNU/Linux.

Tema 1: Configuración de interfaces de red

A pesar de que el núcleo del sistema operativo Linux controla todo el acceso a la red, el proceso de configurar un computador con GNU/Linux instalado para hacer uso de la misma involucra más que solamente el núcleo. En esta unidad estaremos cubriendo algunos tópicos relativos a la configuración, en resumen, veremos como configurar la pila TCP/IP para utilizar bien sean direcciones estáticas o configuraciones automáticas a través del protocolo DHCP, además de explorar las características especiales de configuración que se aplican a las interfaces de red inalámbricas.

Generalidades

El primer paso en la configuración de una tarjeta de red bajo GNU/Linux es cargar los controladores apropiados, para ventaja del usuario y administrador, Canaima GNU/Linux así como casi todas las distribuciones actuales de GNU/Linux, incluyen una amplia gama de controladores para tarjetas de red que evitan la necesidad de configurar a bajo nivel su activación y funcionamiento en el entorno operativo. En la mayoría de los casos, no necesitará pasar ninguna opción especial para el controlador de manera que funcione apropiadamente.

Comando *ifconfig*

La sintaxis básica de *ifconfig* es la siguiente:


| | |
|---|---|
|  | <code>ifconfig [interfaz] [opciones]</code> |
|---|---|

El programa se comporta de una manera diferente dependiendo de las opciones que se le den. En general, *ifconfig* puede hacer varias cosas:

Si es utilizado sin parámetros, *ifconfig* retornará el estado de todas las interfaces de red que se encuentran activas. Usado de esta manera, *ifconfig* es una útil herramienta de diagnóstico. Si se le da un solo nombre de interfaz (como por ejemplo *eth0* o *tr1*), *ifconfig* retornará información únicamente sobre esa. De nuevo, es una herramienta útil de diagnóstico.


Si se le dan opciones a un nombre de interfaz, *ifconfig* modificará la operación de la interfaz en concordancia a las especificaciones. Lo más común es que sean opciones que activen o desactiven la interfaz. Si se está utilizando *ifconfig* para configurar una interfaz, le importarán más las opciones que pueden pasarse a la utilidad. Aunque la página del manual describe con lujo de detalles todas las opciones que acepta *ifconfig*, las más importantes son las siguientes:

`ifconfig interfaz up dirección`: Esta opción activa la interfaz y asocia la dirección IP especificada a la interfaz. Si el comando no incluye una opción de máscara de subred, `ifconfig` asignará la máscara dependiendo de la clase de la dirección. En la mayoría de los casos, puede omitirse la palabra “up”; `ifconfig` asume automáticamente que debe activar la interfaz al asignarle una dirección IP.

| | |
|---|---------------------------------------|
|  | <pre>ifconfig eth0 192.168.20.1</pre> |
|---|---------------------------------------|

`Ifconfig interfaz down`: Esta opción es lo opuesto a “up”; cierra la interfaz, desactivándola.

`ifconfig interfaz netmask máscara`: Define la máscara de subred de la interfaz, lo que determina cuales bits de la dirección IP corresponden a una dirección de red y cuales identifican un computador específico en la red. Si se omite esta opción, `ifconfig` definirá la máscara a un valor por defecto en concordancia con la clase anteriormente definida por la dirección IP configurada.


| | |
|---|--|
|  | <pre>ifconfig eth0 netmask 255.255.0.0</pre> |
|---|--|

`ifconfig interfaz [-]promisc`: Normalmente, una tarjeta de red solo acepta aquellos paquetes que están dirigidos a ella, o a todos los sistemas en su segmento de

red. Esta opción habilita (promisc) or deshabilita (-promisc) el modo promiscuo, donde la tarjeta captura aquellos paquetes que atraviesan su segmento de red, sin que estos sean específicamente de su segmento de red. El modo promiscuo es necesario para los capturadores de paquetes, los cuales pueden ser usados como herramientas de diagnóstico de red. (Los crackers también usan los capturadores de paquetes para obtener contraseñas que son enviadas sin encriptación.) Algunos programas pueden habilitar el modo promiscuo por sí mismos. El comportamiento por defecto es activar la interfaz en un modo no promiscuo.

`ifconfig interfaz mtu número`: Con esta orden se define el tamaño máximo de transferencia (MTU) de una interfaz, lo que viene a ser el tamaño máximo de los paquetes a bajo nivel. Para redes tipo Ethernet, el valor MTU normalmente es 1500, pero puede configurarse a otro valor de ser necesario. (Algunos enrutadores y protocolos utilizan un valor MTU más bajo, lo que podría reducir el rendimiento si el MTU de la interfaz que se encuentra configurando es muy alto, ya que los paquetes más grandes tendrán que ser divididos y enviados en pedazos más pequeños.)

`ifconfig interfaz add dirección/[bits de la máscara o máscara]`: Esta opción es equivalente a “up” y “netmask”, solo que también funciona con direcciones IP versión 6. Un ejemplo puede ser:

| | |
|---|---|
|  | <pre>ifconfig eth0 add 192.168.20.2</pre> |
|---|---|

Lo que añadirá una dirección IP para eth0 con una máscara igual a la máscara que posee la dirección IP original de la interfaz

`ifconfig interfaz del dirección/[bits de la máscara o máscara]`: Esta



opción es exactamente lo opuesto a la opción “add” explicada anteriormente.


`ifconfig interfaz media tipo de medio`: Algunas tarjetas de red incluyen dos o más puertos (por ejemplo, tarjetas de red con un conector UTP RJ-45 y un conector BNC para 10Base-2). Se puede especificar el conector a utilizar con esta opción, como por ejemplo: `media 10base-2`. Consulte la documentación del controlador para los detalles de que valores acepta en “tipo de medio”.

`ifconfig interfaz hw clase dirección física`: Esta opción le permite cambiar la dirección física del adaptador. Podría necesitar cambiarlo si reemplazó la tarjeta de red pero desea utilizar la misma dirección física antigua para seguir recibiendo la misma concesión de un servidor DHCP. A veces, también, los fabricantes cometen una equivocación y sacan al mercado un lote de tarjetas de red con direcciones físicas idénticas (lo que puede causar problemas de recepción de paquetes en una red con varias tarjetas de red con la misma dirección física). Esta opción requiere dos parámetros: la *clase* (que puede ser “ether” para Ethernet, “ARCnet” para ARCnet o “ax25” para AX.25, entre otras) y la *dirección física* (o dirección MAC). Esta función opera con muchas, pero no todas, las tarjetas de red.


`ifconfig interfaz txqueulen tamaño`: Define el tamaño de la cola de transmisión, que es el número de paquetes que la interfaz intentará encolar para su envío al mismo tiempo. El valor por defecto para los dispositivos Ethernet es 100, que usualmente es suficiente. Bajar el valor de este parámetro en conexiones lentas puede mejorar un poco el rendimiento de aplicaciones interactivas a través de la red (por ejemplo una sesión SSH).

Configuraciones estáticas al inicio del sistema.

Supongamos que desea configurar una interfaz Ethernet que tiene una dirección IP fija 192.168.0.123. Esta dirección comienza con 192.168.0 por lo tanto debe estar en una LAN. Supongamos además que 192.168.0.1 es la dirección de la puerta de enlace de la LAN a Internet. Edite `/etc/network/interfaces` de modo que incluya un fragmento como el siguiente:

| | |
|---|--|
|  | <pre>root@maquina:~# editor /etc/network/interfaces # para que la interfaz se active automáticamente al # iniciar el sistema auto eth0 # para que la interfaz se configure con una # dirección IP (versión 4) estática iface eth0 inet static address 192.168.0.123 netmask 255.255.255.0 gateway 192.168.0.1</pre> |
|---|--|

Si tiene instalado el paquete `resolvconf` puede añadir líneas para especificar la información relativa al DNS. Por ejemplo:

| | |
|---|--|
|  | <pre># editor /etc/network/interfaces iface eth0 inet static address 192.168.0.123 netmask 255.255.255.0 gateway 192.168.0.1 dns-search midominio.org dns-nameservers 195.238.2.21 195.238.2.22</pre> |
|---|--|

Luego que se activa la interfaz, los argumentos de las opciones dns-search y dns-nameservers quedan disponibles para resolvconf para su inclusión en /etc/resolv.conf. El argumento midominio.org de la opción dns-search corresponde al argumento de la opción search en resolv.conf. Los argumentos 195.238.2.21 y 195.238.2.22 de la opción dns-nameservers corresponde a los argumentos de las opciones nameserver en resolv.conf. Otras opciones reconocidas son dns-domain y dns-sortlist.

Configuraciones adicionales para interfaces wifi

El paquete wireless-tools incluye el script /etc/network/if-pre-up.d/wireless-tools que permite configurar hardware Wi-Fi (802.11a/b/g) antes que se active la interfaz.

Para cada parámetro posible del comando iwconfig puede incluir una opción en /etc/network/interfaces con un nombre como el del parámetro con el prefijo “wireless-”. Por ejemplo, para fijar el ESSID de eth0 en “miessid” y la clave de cifrado en 123456789e antes de activar wlan0 usando DHCP, edite el /etc/network/interfaces de modo que incluya un fragmento como el siguiente :



```
# editor /etc/network/interfaces  
  
iface wlan0 inet dhcp  
wireless-essid miessid  
wireless-key 123456789e
```

Configuraciones automáticas al inicio del sistema

Supongamos ahora que desea configurar una interfaz Ethernet que obtiene su dirección IP automáticamente cada vez que ud. inicia su sistema GNU/Linux. Edite `/etc/network/interfaces` de modo que incluya un fragmento como el siguiente:



```
root@maquina:~# editor /etc/network/interfaces

# para que la interfaz se active automáticamente al
# iniciar el sistema

auto eth0

# para que la interfaz se configure con una
# dirección IP (versión 4) dinámica otorgada a través
# de un servidor DHCP local.

iface eth0 inet dhcp
```

Múltiples interfaces de red

A fin de facilitar la configuración de la red, Canaima y las distribuciones basadas en Debian proporcionan una herramienta estándar de configuración de red de alto nivel que consiste en los programas `ifup`, `ifdown` y el archivo `/etc/network/interfaces`. Si elige utilizar `ifupdown` para realizar la configuración de su red, entonces no debería usar los comandos de bajo nivel. `ifupdown` se programó bajo la suposición que solo iba a ser

utilizado para configurar y desconfigurar las interfaces de red.

Para actualizar la configuración de la interfaz haga lo siguiente:



Bajamos la interfaz eth0

```
root@maquina:~# ifdown eth0
```

Editamos el archivo de configuración de las interfaces de red a nuestro gusto

```
root@maquina:~# editor /etc/network/interfaces
```

Finalmente, activamos la interfaz o interfaces definidas con anterioridad usando ifup

```
root@maquina:~# ifup eth0
```

Interfaces virtuales

Usando interfaces virtuales puede configurar una única tarjeta Ethernet para que sea la interfaz de distintas subredes IP. Por ejemplo, supongamos que su máquina se encuentra en una red LAN 192.168.0.x/24. Desea conectar la máquina a Internet usando una dirección IP pública proporcionada con DHCP usando su tarjeta Ethernet existente. Edite /etc/network/interfaces de modo que incluya un fragmento similar al siguiente:



```
root@maquina:~# editor /etc/network/interfaces

iface eth0 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255

iface eth0:0 inet dhcp
```

La interfaz eth0:0 es una interfaz virtual. Al activarse también lo hará su padre eth0.



Tema 2: Integración del sistema en un entorno de red

Utilizando el servicio DNS

La configuración relevante de los servidores de nombre se hace, como se mencionó con anterioridad en el archivo: `/etc/resolv.conf`. En este archivo se definen las direcciones IP que atienden solicitudes de resolución de nombres.

Este archivo solo debe ser manipulado en caso de tener una configuración de red que comprenda direcciones estáticas, ya que el sistema no las actualizará de forma automática. Si su configuración de red comprende el uso de DHCP, el cliente DHCP se encargará de manejar cualquier cambio a este archivo.

Si desea más información, refiérase al manual: “Sistemas de Resolución de Nombres en Canaima GNU/Linux”.



Tema 3: SSH

SSH¹⁸ (*Secure SHell*) es una implementación segura de sesiones de intérprete de comando remotas. SSH es la manera segura de comunicarse a través de Internet para la administración de un computador ejecutando GNU/Linux. La implementación más popular de SSH consta de una variante llamada OpenSSH¹⁹. La cual incluye una colección de herramientas para realizar conexiones a través de este protocolo.

Iniciando ssh

Para comenzar, instale el cliente y el servidor OpenSSH. (en Canaima GNU/Linux tanto el cliente ssh como el servidor vienen instalados por defecto)



```
# aptitude install ssh openssh-server
```

Servidor SSH

El servidor ssh, al funcionar en un computador con GNU/Linux, permite la conexión de clientes a un intérprete de comandos en el sistema anfitrión. Por otro lado, sus características de encriptación y autenticación de varias vías le dan mucha flexibilidad cuando se trata de controlar el acceso de los clientes con métodos de autenticación que van desde la simple solicitud de contraseñas, hasta el bloqueo de clientes desconocidos a

¹⁸ Intérprete o “shell” seguro, por sus siglas en inglés.

¹⁹ [Http://www.openssh.org](http://www.openssh.org)



la red donde opera.

Mecanismos de autenticación y opciones de configuración

Las configuraciones del servidor ssh, como sus mecanismos de autenticación y el puerto donde este opera pueden cambiarse editando el archivo `/etc/ssh/sshd_config`. Explicamos a continuación los más importantes:

- **Port:** Define el puerto donde el servicio funcionará. El puerto estándar de ssh es el puerto 22 TCP.
- **Protocol:** Define la versión de protocolo que se acepta por defecto, el valor por defecto es 2, ya que actualmente, es poco común conseguir un cliente ssh que no soporte esta versión del protocolo.
- **UsePAM:** Este parámetro determina si el servidor ssh permitirá la entrada a usuarios válidos del sistema donde se está ejecutando, de estar establecido a “no” la autenticación del servidor deberá configurarse para utilizar otra base de datos de usuarios y contraseñas, o bien, un método de autenticación por validación de llave pública.
- **PubkeyAuthentication:** autenticación del usuario basada en una clave pública
- **Hostbased Authentication:** autenticación basada en `.rhosts` o `/etc/hosts.equiv` combinada con la autenticación de la clave pública de la máquina cliente (desactivada).
- **Password Authentication:** autenticación basada en contraseña.
- **Challenge Response Authentication:** autenticación basada en challenge response.
- `/etc/ssh/sshd_config` es el archivo de configuración más importante, las entradas son:
- **Host:** Restringe las siguientes declaraciones (up to the next Host keyword) siendo nada mas los host que declaremos aquí los que se les dará una llave de

autenticación ssh para entablar comunicación.

- Protocol: especifica la versión del protocolo SSH. Valor predeterminado “2,1”.
- Preferred Authentications: especifica el método de autenticación para el cliente SSH2. Por defecto: “hostbased,publickey,keyboard-interactive,password”.
- ForwardX11: desactivado por defecto. Se puede no tener en cuenta mediante la opción “-X” de la línea de comandos
- Otro archivo de interés: /etc/ssh/sshd_config: valores predeterminados del servidor SSH. Las entradas más importantes son:
- ListenAddress: especifica las direcciones locales que sshd debe escuchar. Se permiten múltiples opciones.
- AllowTcpForwarding: desactivado por defecto.
- X11Forwarding: desactivado por defecto.

Cliente SSH

Mediante el uso del cliente ssh es que en práctica, se hace uso de las bondades de esta herramienta administrativa. El cliente ssh se invoca en cualquier consola del sistema por su nombre

Lo siguiente iniciará una conexión ssh desde un cliente a un servidor en la máquina miservidor.ssh.com



```
usuario@maquina$ ssh nombre_usuario@miservidor.ssh.com  
password: (digite su contraseña)  
nombre_usuario@miservidor.ssh.com:$
```



Como puede verse, se obtiene una sesión remota en un intérprete de comandos en otro computador.

El cliente y servidor ssh resulta de especial importancia para la administración de sistemas GNU/Linux remotos, aunque se encuentren próximos al administrador. Ya que permiten una gran flexibilidad y comodidad para la administración y operación de los ambientes productivos.

El comando SCP

Parte del cliente ssh es el comando scp²⁰ (*secure copy*) que permite la copia segura de archivos entre computadores.

Por ejemplo, deseamos copiar el archivo “pepe.txt” desde el directorio /tmp de una máquina al home del usuario “sutano” en la máquina con dirección “mi.ssh.net”.



```
$ scp /tmp/pepe sutano@mi.ssh.net:/home/sutano/  
sutano@mi.ssh.net's password:  
pepe.txt 100%|*****| 50165
```

De nuevo, se le pedirá una clave. La orden scp muestra el progreso de la copia por omisión. Puede copiar un archivo desde un host remoto con la misma facilidad; simplemente especificando su nombre de host y ruta como origen y la ruta local como destino. También se puede copiar un archivo desde un host remoto a otro host remoto, pero habitualmente no necesitará hacer eso, porque todos los datos viajan a través de su host.

²⁰ Siglas en inglés de copia segura, *Secure copy*, s

Conexiones SSH reversa

Las conexiones ssh reversas son útiles cuando no es posible acceder directamente a un servidor ssh desde fuera de un segmento de red específico bien sea por que está bloqueado por un cortafuegos, o por que se encuentra en una NAT inaccesible, sin embargo nos es posible, desde ese computador, tener acceso a otro computador que ejecuta ssh. En este caso, podemos establecer un túnel de dos vías entre ambos computadores para acceder a aquel que de otra forma no podríamos alcanzar. Usualmente, el establecimiento de un túnel reverso ssh implica que el usuario que va a conectarse al sistema antes inaccesible, lo hace conectándose localmente al túnel que se crea en la máquina destino para este propósito.

Pongamos un ejemplo: digamos que nos encontramos en una máquina de nombre “anaconda” con dirección 172.31.5.11. La cual no tiene conexión directa con internet y al intentar conectarme a esta desde allí me es imposible. Teniendo acceso a la máquina y verificando que pueda conectarse con mi servidor de nombre “tuneles” en 207.100.50.11 establezco un túnel en esta última máquina para poder conectarme:



comenzamos indicándole en que puertos debería configurarse el túnel ssh, por defecto le indicamos que queremos hacer un túnel que llega al puerto 22 (puerto por defecto de ssh) en la máquina anaconda donde un usuario en la máquina tuneles pueda conectarse a anaconda a pesar de que está no es accesible desde el Internet, por el puerto 2000, desde la máquina local.



```
usuario@anaconda$ ssh -nNT -R 22:localhost:2000
usuario@207.100.50.11
```

A continuación, en la máquina túneles me conecto a anaconda por el túnel. Especificándole al cliente ssh el puerto

```
usuario@tuneles$ ssh usuario@localhost -p 2000
```

Transferencia de archivos con SFTP.

Aunque suele pensarse que SFTP es FTP sobre el protocolo SSH, en realidad no es así, SFTP es una de las tantas herramientas y sub-protocolos derivados de SSH.

Provee todas las ventajas del antiguo modo pasivo FTP (como resumir descargas, pausar y comprobación, por citar algunas) además de la encriptación nativa del protocolo en el cual se basa, SSH.

Sftp suele usarse para conexiones semi-persistentes seguras con algún servidor de archivos y su operación es bastante similar al antiguo ftp.



```
$ sftp usuario@maquinaremota
password:
sftp>
```

Como se ve en el ejemplo anterior, la invocación de sftp es bastante parecida a la invocación de ssh, sin embargo sftp nos deja en su propio intérprete para que realicemos las operaciones de copiar, descargar o buscar archivos.



Las operaciones más importantes en el intérprete de comandos de sftp se resumen a continuación:

- `get ruta remota [ruta local]`: `get` es utilizado para descargar archivos, se le puede dar una ruta completa al archivo donde se encuentra en el servidor o bien si está en la misma carpeta donde comenzamos la sesión de sftp (por defecto sftp inicia sesión en la carpeta `home` del usuario) se le da el nombre del archivo. Opcionalmente se le puede indicar a `get` donde colocar el archivo en la máquina local, siendo esta ruta relativa al directorio donde se inició la sesión sftp.
- `ls`: lista los archivos y directorios que se encuentran en la carpeta actual remota.
- `lls`: lista los archivos y directorios que se encuentran en la carpeta actual local.
- `put ruta local [ruta remota]`: `put` es utilizado para subir archivos, se le puede dar una ruta completa al archivo donde se encuentra en la máquina cliente o bien si está en la misma carpeta donde comenzamos la sesión de sftp (por defecto sftp inicia sesión en la carpeta `home` del usuario) se le dá el nombre del archivo. Opcionalmente se le puede indicar a `put` donde colocar el archivo en la máquina remota, siendo esta ruta relativa al directorio donde se inició la sesión sftp.

Tema 4: Servicio VNC.

Hay muchos servidores VNC, pero explicaremos el uso de x11vnc porque es el más sencillo de utilizar. La mayoría de servidores VNC requieren un display de las X particular y, aunque ofrecen un escritorio remoto, lo que hacen es iniciar una nueva sesión gráfica en vez de ofrecer acceso a una sesión ya existente. Con x11vnc podremos permitir el acceso a una sesión X ya existente de una forma sencilla.

Instalamos el paquete para el Servidor:



```
# aptitude install x11vnc
```

Arrancando el servidor

Para arrancar el servidor, abriremos una consola y escribiremos el comando x11vnc. Esto nos iniciará un servidor básico, sin contraseña, que permite el acceso a todo el mundo y que una vez ha desconectado el cliente, se cierra.

A continuación, veremos los parámetros que podemos pasarle al inicio, para configurar el servidor de una forma más razonable:

- bg: Nos inicia el servidor en segundo plano. Para poder cerrar la consola y que siga en marcha.
- passwd: Establece la contraseña que se pedirá a los clientes al conectar.

- gui: Inicia una pequeña interfaz gráfica del lado del servidor.

Sabiendo estos parámetros, podríamos iniciar el servidor de VNC de esta manera, para que nos aparezca su ventana de configuración:



```
$ x11vnc -bg -gui -passwd contraseña
```

Esto iniciará el servidor VNC y nos abrirá la pantalla de configuración, en la que podremos configurar opciones avanzadas del servidor.

Por el lado del cliente utilizaremos tighvncviewer, una implementación de cliente VNC muy robusta y multiplataforma. Para instalar el paquete cliente, se utiliza la siguiente orden:



```
# aptitude install xtightvncviewer
```

Se puede ejecutar xtightvncviewer desde el enlace del menú que nos crea en nuestro computador cliente, sin embargo, también puede ejecutarse a conveniencia desde la consola:



```
$xtightvncviewer
```

Cuando esté ejecutándose, nos solicitará el nombre o dirección IP del host con el



que deseamos conectarnos. Una vez establecida la conexión, nos preguntará por la contraseña previamente definida cuando configuramos y ejecutamos el servidor. Esto es igualmente válido para servidores VNC a los cuales tengamos acceso pero no necesariamente hayan sido instalados por nosotros.



UNIDAD VII: Instalación de paquetes de software

Tema 1: Sistema de empaquetado APT

APT²¹, es una interfaz de usuario libre que trabaja en conjunto con librerías de base para el control, instalación y desinstalación de software en la distribución Debian GNU/Linux y sus variantes, como Canaima GNU/Linux. APT simplifica el proceso de la administración del software en sistemas tipo Unix automatizando la descarga, configuración e instalación de paquetes de software, bien sea desde archivos binarios o compilando código fuente.

Uso e instalación de paquetes

No existe un único programa "apt"; apt en sí mismo es el nombre del paquete que contiene la variedad de herramientas (y las librerías base de funcionamiento) que dan soporte a sus características. Las herramientas más comunes que se instalan por defecto en un sistema GNU/Linux que utilice este sistema de manejo de paquetes son apt-get, apt-cache y aptitude.

APT podría considerarse también como una interfaz a dpkg, siendo incluso una interfaz más amigable a dselect. Mientras que dpkg realiza acciones en paquetes puntuales, las herramientas apt manejan las relaciones (especialmente las dependencias) entre ellos, así, como la obtención y manejo de decisiones de versionamiento de alto nivel.

21 Siglas en inglés de *Advanced Packaging Tool* o "Herramienta avanzada de empaquetado".



Una de las grandes características de APT es la forma en que ejecuta `dpkg`; realiza un ordenamiento topológico de los paquetes a ser instalados o removidos y llama a `dpkg` en la mejor secuencia posible. En algunos casos utiliza la opción “`--force`” (forzar). Sin embargo, solo lo hace en caso de que le sea imposible calcular la forma de evitar la razón por la cual cierta acción solo puede ser realizada en forma forzosa.

Una directiva de instalación va seguida de uno o más paquetes que se desean instalar. Cada nombre de paquete está definido solo con la porción de nombre que lo involucra, no un nombre completamente calificado. (por ejemplo, en un sistema Canaima GNU/Linux se daría el argumento: `canaima-estilo-visual`, más no: `canaima-estilo-visual_1.99.4-4_all.deb`). Notablemente, cualquier paquete que contenga dependencias en otro paquete o paquetes también será descargado e instalado. Esta es una característica original que distinguía al manejo de paquetes basado en `apt` de otros sistemas, puesto que así se evitaban conflictos entre librerías y software de igual funcionalidad.

Otra característica que distingue a `apt` es la descarga de los paquetes de repositorio remotos. Los cuales se configuran en un archivo de definición de fuentes de repositorios (`/etc/apt/sources.list`), donde además de definirse el lugar de descarga y obtención de paquetes provee la información de los paquetes que no están instalados en el sistema.

Los comandos más utilizados para instalar paquetes son:

`aptitude install nombre de paquete`. Este comando instala el paquete dado, y sus dependencias, de ser requerida la instalación de paquetes adicionales.

`aptitude update`: Este comando actualiza las listas de paquetes y software disponible desde los repositorios configurados.



aptitude upgrade: es utilizado para instalar la versión más reciente de todos los paquetes actualmente instalados en el sistema, según estén en el repositorio configurado. Por ningún motivo se desinstalan los paquetes que estén actualmente instalados en el sistema. Solo serán actualizados a sus últimas versiones.

aptitude dist-upgrade: Además de cumplir la misma función de “upgrade”, maneja de forma inteligente cualquier cambio de dependencia en las nuevas versiones. Lo que significa que intentará actualizar los paquetes más importantes de ser necesario, a expensas de otros de menor relevancia. Es también necesario hacer la actualización por este método si se cambia de versión de la distribución en uso. (quiere decir que hay muchos paquetes a actualizar y deseamos que se instalen con una resolución de conflictos adecuada).

Almacén de paquetes

El almacén de paquetes local, (usualmente ubicado en cualquier sistema que utilice el manejador de paquetes APT bajo el directorio: /var/cache/apt/archives) a diferencia de los repositorios (inclusive aquellos que pueden estar en un medio físico accesible localmente, como un CD o un DVD) solo contiene los paquetes que se han instalado o descargado históricamente en el sistema. Es una idea sencilla que permite reinstalar cualquier paquete previamente instalado bien sea por algún incidente con el sistema de archivos donde están instalados los paquetes o por que se desea devolverlos a una configuración original.

Sin embargo, este almacén puede limpiarse, que en la mayoría de los casos y con un sistema bastante longevo podría estar consumiendo espacio de forma innecesaria.



Esta limpieza se realiza con el comando:



```
# aptitude clean
```

También puede limpiarse del almacén local aquellos paquetes que no es posible descargar del repositorio, bien sea por que se actualizaron en el mismo o por que se cambió la configuración de los repositorios. Esto permitiría solo borrar aquellos paquetes que no solo no pueden ser instalados en el sistema, si no que permitiría que se sigan manteniendo en el almacén local aquellos paquetes que podríamos necesitar reinstalar eventualmente de manera de reducir la cantidad de descarga de paquetes. La limpieza parcial del almacén local puede hacerse con el siguiente comando:



```
# aptitude autoclean
```



Tema 2: Configurando el sistema APT

Definiendo repositorios y versiones

El sistema apt incluye herramientas que permiten actualizar un gran número de paquetes simultáneamente, como ya se explicó.

La definición de repositorios se hace en el archivo `/etc/apt/sources.list` en este archivo, se conservan línea por línea, las direcciones de los repositorios a utilizar, las ramas y tipos de paquetes soportadas por ese repositorio.

El formato de cada línea, es el siguiente:

- *Tipo de repositorio*: puede ser “deb” si es un repositorio de paquetes binarios solamente, o “deb-src” si es un repositorio para descargar paquetes de código fuente.
- *Dirección o URL de descarga*: Una URL en la forma: `http://direccion/directorio`, nótese que apt soporta solamente los protocolos http o ftp para descargar paquetes.
- *Rama de la distribución*: una palabra clave con la rama o versión de la distribución y los tipos de paquetes que serán descargados.
- *Tipos de paquetes*: aquí se definen las secciones de paquetes que podrán ser descargados desde este repositorio. No todos los repositorios disponen todos los tipos de paquetes, así que deberá consultar al administrador del repositorio cuales son los tipos de paquetes que pueden descargarse.

Un ejemplo de un repositorio de paquetes binarios que resida en el computador

mi.repositorio.com de la distribución pruebas y que soporte las secciones de paquetes: “libres” y “no-libres” respectivamente sería:



```
# editor /etc/apt/sources.list
```

```
deb http://mi.repositorio.com/ pruebas libres no-libres
```

Tomando también el ejemplo de los repositorios por defecto de Canaima GNU/Linux:



```
# editor /etc/apt/sources.list
```

```
# Repositorios en línea
```

```
deb http://repositorio.canaima.softwarelibre.gob.ve/ estable  
usuarios
```

```
deb http://universo.canaima.softwarelibre.gob.ve/ lenny main  
contrib non-free
```

```
deb http://seguridad.canaima.softwarelibre.gob.ve/  
seguridad usuarios
```

Configurando el comportamiento del sistema APT

Por defecto, APT instala los paquetes de la versión más reciente que está disponible desde los repositorios, sin embargo, APT también puede configurarse para preferir una versión anterior o bien, una versión específica de un repositorio o rama específica. Esto se hace a través de la característica “*pinning*²²” de APT, esto permite al administrador de un sistema evitar la actualización o instalación de algún paquete de software en particular, que podría entrar en conflicto con configuraciones ya establecidas

²² Una traducción vaga del inglés, sería: *marcado*.



o simplemente por razones de seguridad.

Para configurar el *pinning* es necesario modificar el archivo `/etc/apt/preferences`, en donde se encuentran las configuraciones de las preferencias de APT.

Un ejemplo del *pinning* que viene por defecto en Canaima es el siguiente:



```
# editor /etc/apt/preferences
```

```
Package: *
```

```
Pin: release a=estable
```

```
Pin-Priority: 700
```

```
Package: *
```

```
Pin: release o=Debian
```

```
Pin-Priority: 50
```

Este último ejemplo muestra claramente que se prefiere cualquier paquete para la rama “estable” de Canaima GNU/Linux que de la distribución Debian GNU/Linux (ya que Canaima es 100% compatible con los paquetes de esta distribución, al estar basada en ella). Por lo que si un mismo paquete existe en ambas distribuciones, se preferirá instalar la versión que viene con Canaima GNU/Linux que la que viene por defecto en Debian GNU/Linux, esto se determina con la prioridad asignada con el valor “Pin-Priority”.



Tema 3: Servicio de proxy/cache APT

Se puede tener, para efectos prácticos, un sistema que haga las funciones de un repositorio remoto, pero que se encuentre en nuestra red local. Esto, con el propósito de evitar que en un ambiente de múltiples computadores con el sistema operativo Canaima GNU/Linux todas deban descargar directamente de internet las actualizaciones y paquetes de software nuevo. En vez de eso se descarga una sola vez desde el internet desde un computador dispuesto para tal fin que ejecuta bien sea Canaima GNU/Linux o una distribución basada en Debian, instalándole *apt-cacher-ng*²³

Apt-cacher-ng

Apt-cacher-ng es un proxy caché especialmente diseñado para ser utilizado como intermediario entre repositorios de software para distribuciones Linux, enfocándose primariamente en Debian (y sus derivados) aunque no está limitado a estas distribuciones.

Su funcionalidad básica y más importante es la de mantener una copia local de los paquetes descargados para su instalación en uno o más clientes; En ese sentido, ahorra ancho de banda cuando se requiere que múltiples computadores que utilizan los mismos repositorios de software descarguen, de forma independiente los paquetes de software a instalar, manteniendo una sola copia que entonces es utilizada por todos los clientes que requieren descargar uno o múltiples paquetes de un mismo repositorio.

Su instalación es bastante sencilla:



```
# aptitude install apt-cacher-ng
```

²³ <http://www.unix-ag.uni-kl.de/~bloch/acng/>

Configuración del servicio

La configuración del servicio apt-cacher-ng es totalmente automática, siendo las únicas opciones relevantes que podrían cambiarse (según su necesidad) son: el puerto donde opera y la dirección IP donde aceptará conexión de los clientes.

Para cambiar el puerto donde opera el servicio solo es necesario editar el archivo: /etc/apt-cacher-ng/acng.conf:



```
# editor /etc/apt-cacher-ng/acng.conf
```

*#El puerto por defecto del servicio es 3142, podemos
cambiarlo modificando la opción "Port"*

```
Port:puerto
```

Si desea que apt-cacher-ng solo acepte conexiones en una dirección IP particular del servidor, podrá cambiar la opción BindAddress del servicio: (la configuración por defecto es que el servicio escuche en todas las direcciones IP que tiene asignadas), en este parámetro se define colocando las diferentes direcciones IP separadas por espacio.



```
# editor /etc/apt-cacher-ng/acng.conf
```

*#El servicio escucha por defecto en todas las direcciones
IP del computador, para cambiar esta configuración,
editamos el parámetro "BindAddress"*

```
BindAddress:direcciones IP separadas por espacio
```

Configuración del cliente

La configuración de los clientes de apt-cacher-ng es sumamente sencilla, pueden aplicarse dos estrategias. (pero nunca ambas al mismo tiempo); A saber, se puede configurar los repositorios accesibles al cliente anteponiendo la dirección del servidor donde se ejecuta apt-cacher-ng. En el siguiente ejemplo asumimos que nuestro servidor apt-cacher-ng se ubica en la dirección IP 172.16.31.8 y que está escuchando conexiones en el puerto 3142:



```
# editor /etc/apt/sources.list

# Cambiamos la definición original de nuestro repositorio
configurado actualmente:

deb http://repositorio.canaima.softwarelibre.gob.ve/ estable
usuarios

# Por:

deb
http://172.16.31.8:3142/repositorio.canaima.softwarelibre.gob.ve/
estable usuarios
```

La otra técnica de configuración involucra informarle al subsistema APT que existe un proxy para conectarse a los repositorios. En el siguiente ejemplo (conservando el contexto), se configura esta técnica alternativa, de la siguiente manera:



```
# editor /etc/apt/apt.conf.d/02proxy
```

```
# Configuramos las preferencias de APT para dirigirlo al proxy-  
caché de paquetes:
```

```
Acquire::http { Proxy "http://172.16.31.8:3142"; };
```




UNIDAD VIII: Servicios de impresión con CUPS

Tema 1: Introducción a CUPS

Funcionamiento

El Sistema de impresión común de Unix CUPS²⁴ es un sistema de impresión modular para sistemas operativos de tipo Unix que permite que un computador actúe como servidor de impresión. Un computador que ejecuta CUPS actúa como un servidor que puede aceptar tareas de impresión desde otros computadores clientes, los procesa y los envía al servidor de impresión apropiado.

CUPS está compuesto por una cola de impresión con su planificador, un sistema de filtros que convierte datos para imprimir hacia formatos que la impresora conozca, y un sistema de soporte que envía los datos al dispositivo de impresión. CUPS utiliza el protocolo IPP(Internet Printing Protocol) como base para el manejo de tareas de impresión y de colas de impresión. También provee los comandos tradicionales de línea de comandos de impresión de los sistemas Unix, junto a un soporte de operaciones bajo el protocolo server message block (SMB). Utilizado por clientes Microsoft Windows.

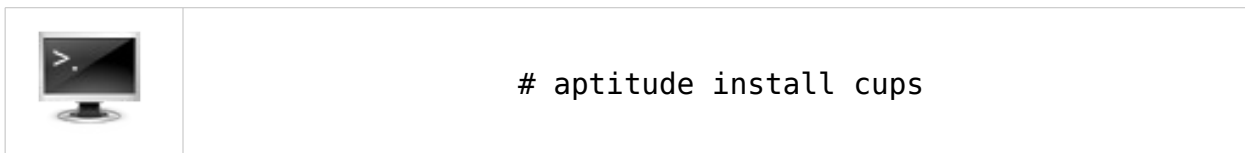
Los controladores de dispositivos de impresión que CUPS provee pueden ser configurados utilizando archivos de texto con el formato PPD²⁵ (Descripción de impresoras PostScript) de Adobe Systems.

24 Siglas de *Common Unix Printing System* en inglés. Que significa: sistema común de impresión de UNIX.

25 Siglas de *PostScript Printer Definition* en inglés

Instalación a través del sistema de empaquetado

Para instalar el servidor cups y sus herramientas asociadas, solo es necesario teclear la siguiente orden:



Elementos de configuración del servicio

Filtros, controladores, PPDs ... El flujo de impresión se compone de varios elementos que debemos conocer para comprender el funcionamiento de CUPS. El proceso general es el siguiente: la aplicación suministra un archivo en Postscript, PDF, texto plano o mapa de bits (JPG, PNG, etc.). El contenido de este archivo va al sistema de impresión, que entonces observa a cual impresora debe enviarlo. Una vez decidido, busca las características de la impresora para saber si es necesaria alguna transformación: hay impresoras que pueden recibir Postscript sin traducción, y en esos casos el sistema de impresión solo pasa lo que a su vez le proporcionó la aplicación, convirtiéndolo a Postscript antes si no viniera ya en ese formato. Como lo normal (para un usuario doméstico) es que la impresora no “hable” Postscript, el sistema de impresión tiene que buscar una forma de convertir los datos al lenguaje nativo de ésta. Los programas encargados de esto son los filtros. Estos filtros pueden recibir como entrada los datos de la aplicación, o necesitar un paso intermedio en el que se convierten a una representación gráfica llamada *raster*.

Asimismo, los filtros cumplen otras funciones aparte de convertir un archivo *raster*



a lenguaje nativo de la impresora: pueden reducir el tipo de letra, poner varias páginas “virtuales” por página “real”, etc. Estos filtros conocen los lenguajes nativos de las impresoras siempre que sus controladores estén instalados.

El sistema de impresión por defecto de la gran mayoría de las distribuciones GNU/Linux es CUPS, Canaima GNU/Linux también lo incluye de manera predeterminada. Aparte de algunos filtros que trae CUPS de serie (y que podemos ver bajo el directorio /usr/lib/cups/filter), Ghostscript, Foomatic o los filtros que proporcione el fabricante de nuestra impresora. Ghostscript trae controladores para muchas impresoras, y Foomatic también; aunque también podemos usar controladores externos, como los proporcionados por Gutenprint o los filtros y controladores para las impresoras HP, disponibles al instalar los paquetes hpijs y hpijs-ppds

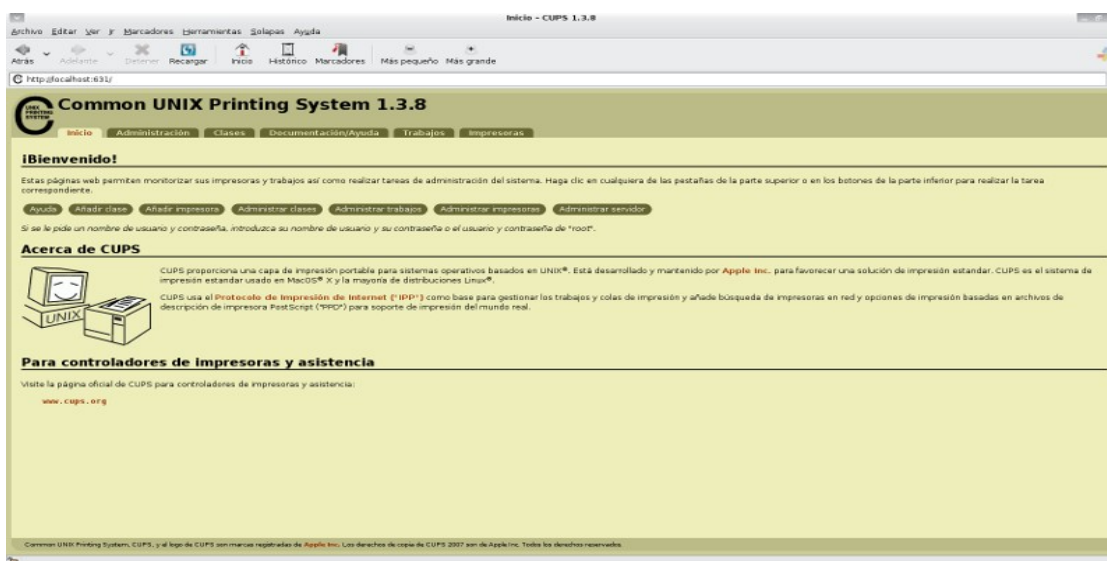
La configuración de cups, está almacenada alrededor de varios archivos en /etc/cups, por lo general no será necesario modificarla directamente, ya que la interfaz web de administración, además de hacerlo de una forma más amigable, chequea automáticamente su buen estado y la corrige de ser necesario para omitir parámetros mal ingresados. En este sentido CUPS es un servicio que se sana automáticamente, para evitar cualquier problema de configuración que evite su correcta operación.



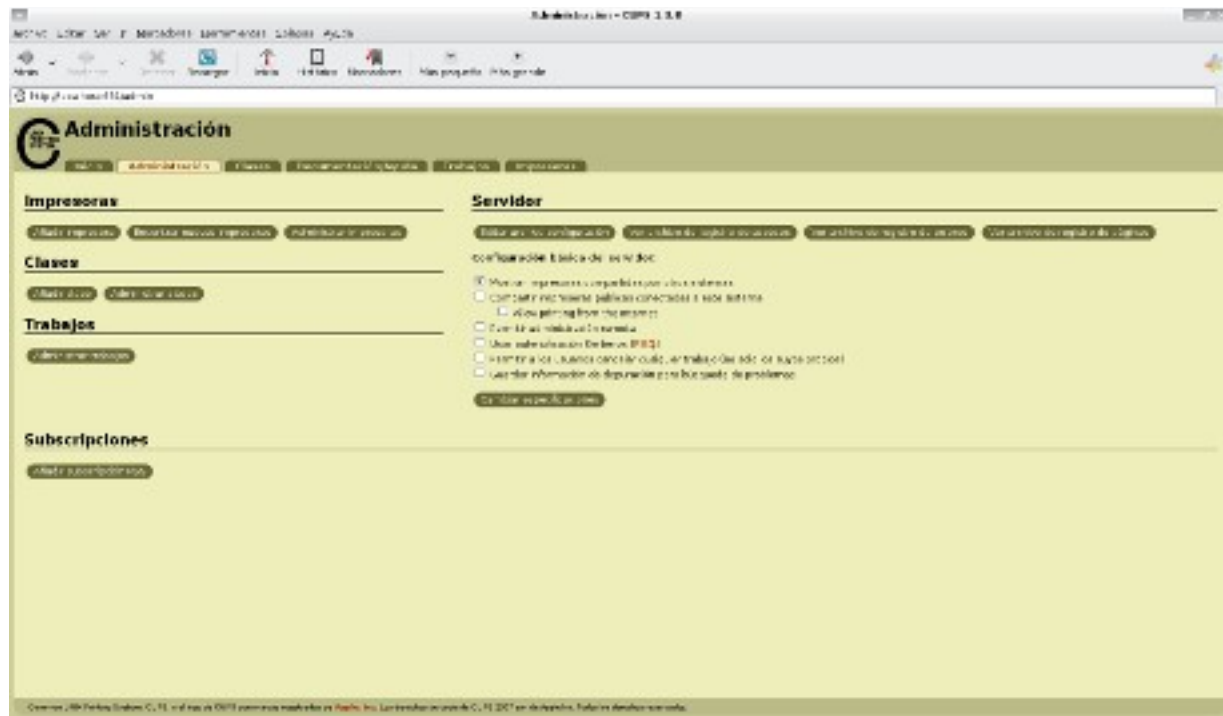
Tema 2: Interfaz Web de administración

Como se dijo con anterioridad, la administración de CUPS las haremos a través de su interfaz web. A este interfaz se puede acceder lanzando el navegador desde el mismo equipo en el que está CUPS, y apuntándolo a `http://localhost:631` (por defecto el interfaz solo funciona en la misma máquina, es decir localhost). Aunque por defecto se nos proporciona el diálogo de bienvenida, para acceder a las funciones avanzadas necesitaremos un usuario y contraseña para poder hacer cambios, por defecto el usuario “root” tiene permisos completos para el uso y modificación de las opciones de CUPS.

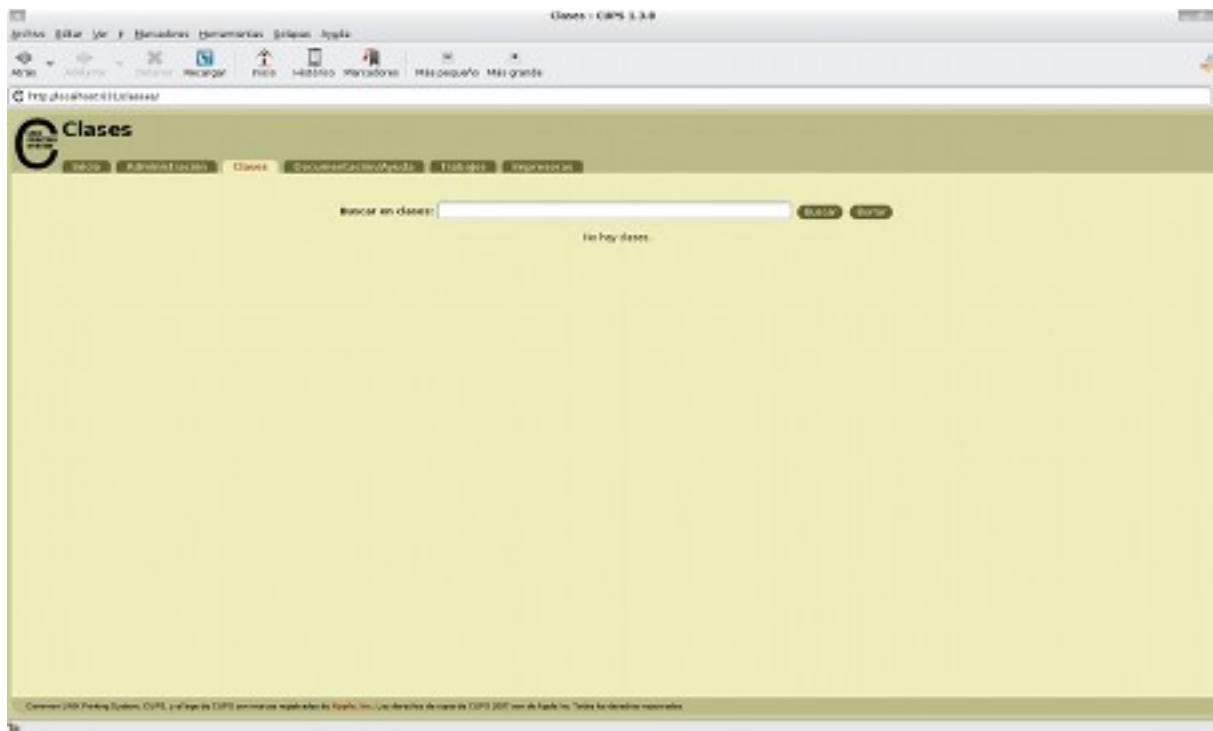
La interfaz web de administración es bastante intuitiva, dividiendo su funcionalidad en diferentes pestañas a las cuales se puede acceder de forma individual:



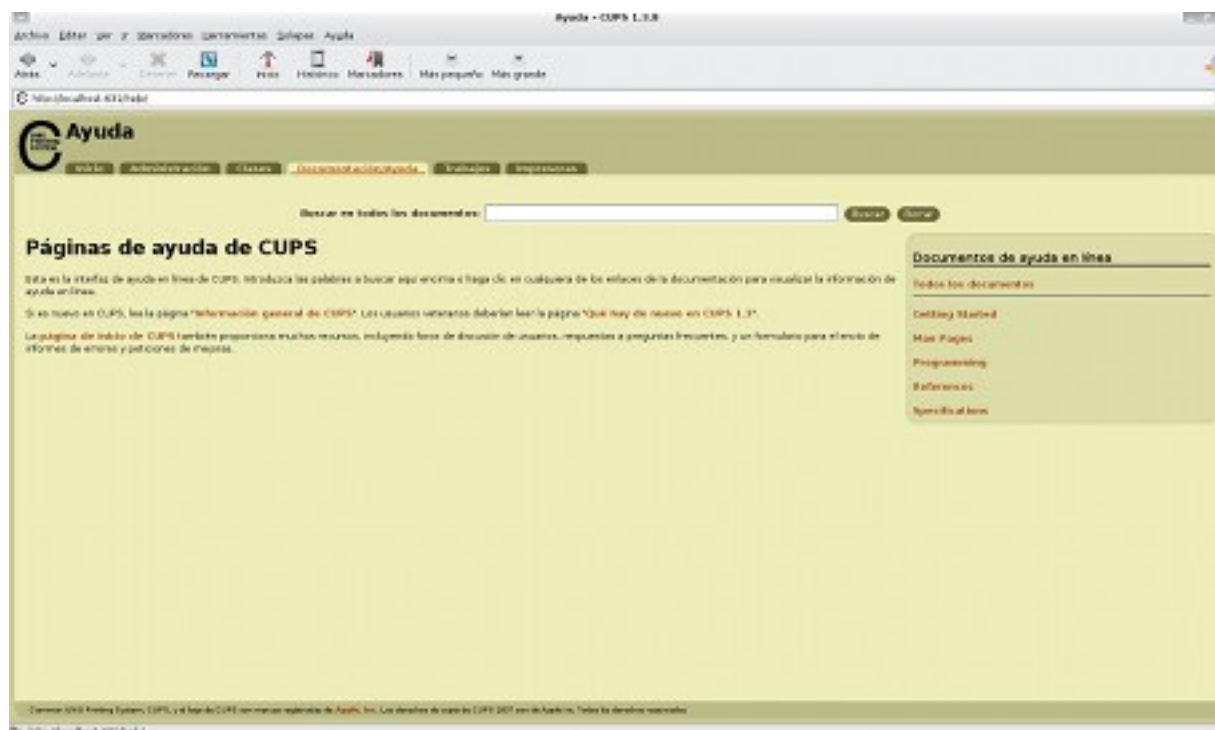
Inicio: Es la pantalla de bienvenida de la interfaz web del CUPS, desde ella se puede acceder a funciones comunes de administración del sistema de impresión y de las impresoras.



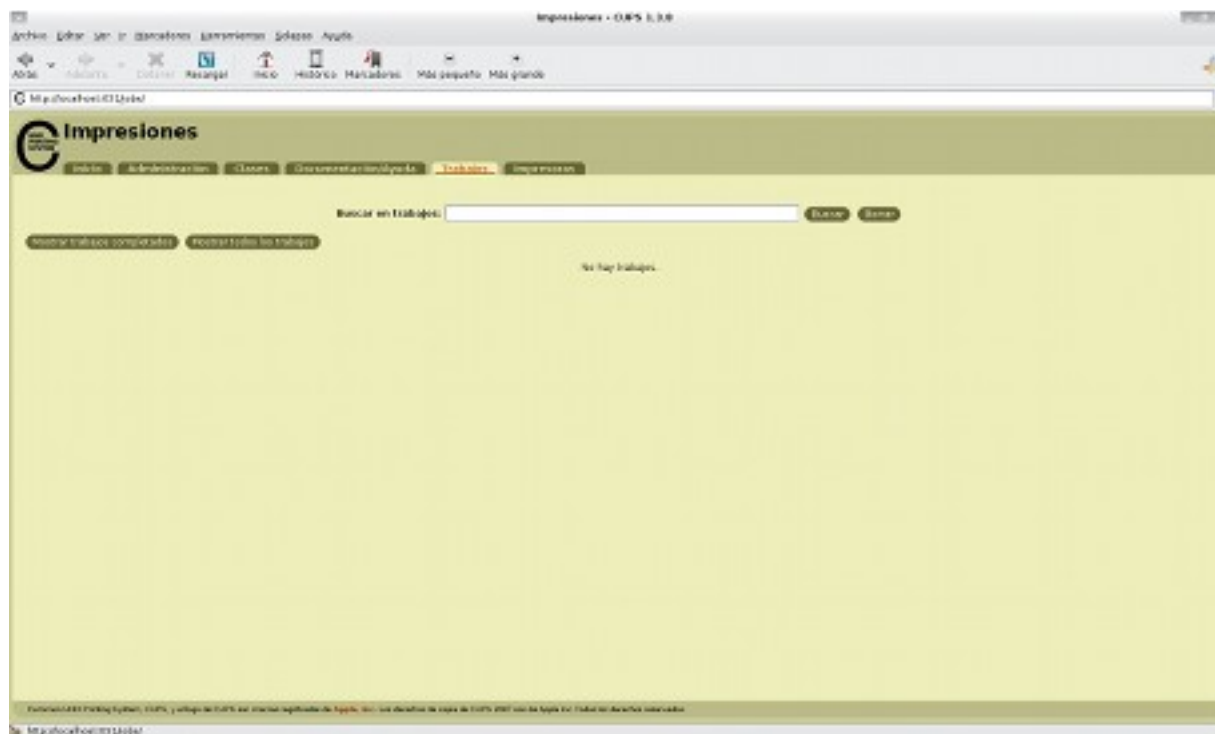
Administración: Comprende las funciones administrativas de CUPS, desde esta pestaña podemos hacer casi cualquier tarea administrativa sobre el sistema de impresión como añadir una impresora, añadir una clase, administrar los trabajos de impresión, ver los registros de acceso, entre otras.



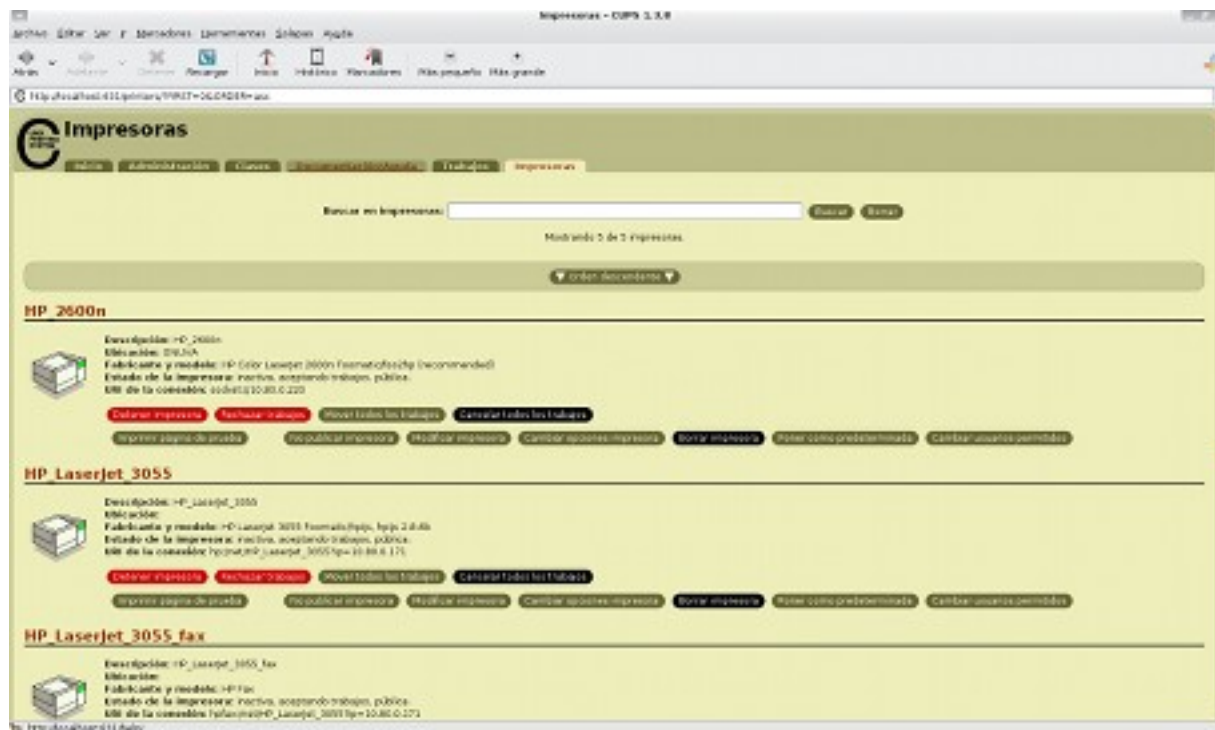
Clases: esta pestaña nos permite visualizar, buscar o definir las clases. Las clases en CUPS son grupos de impresoras del sistema con políticas definidas. De forma predeterminada CUPS no define ninguna clase y no son necesarias para su correcta operación.



Documentación/Ayuda: Una pestaña desde donde podrá acceder a la vasta documentación del sistema CUPS.



Trabajos: En esta pestaña podremos ver todos los trabajos que está gestionando el sistema CUPS actualmente, además, podremos ver los trabajos que están ejecutándose en otras impresoras compartidas por otros sistemas CUPS en la red (si el administrador del sistema CUPS remoto lo permite, claro está), también podremos gestionarlos; pausarlos, reiniciarlos o cancelarlos, siempre que tengamos derechos administrativos.



Impresoras: La pestaña más importante de esta interfaz. Desde ella podremos ver las impresoras que están instaladas, su estado, y opciones en general como tipo de papel, calidad de impresión, página de cubierta y etc. Desde aquí podremos también desinstalarlas o detenerlas.

Gestión de Impresoras

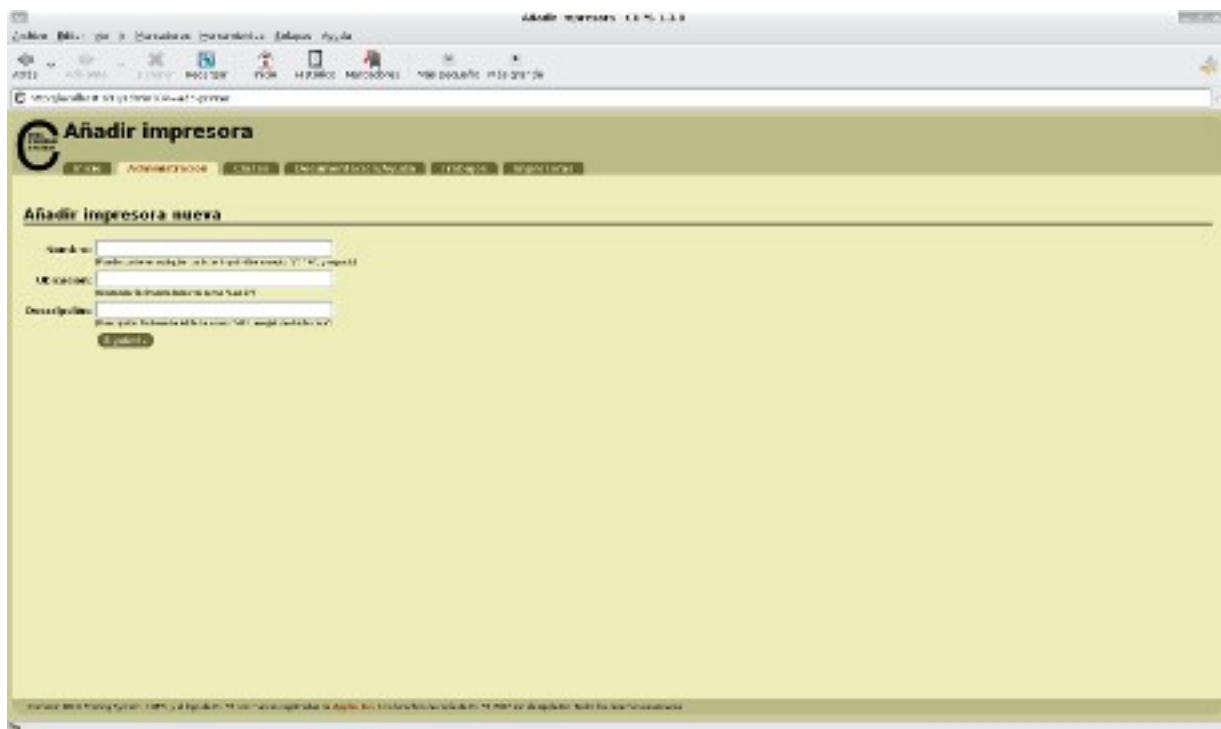
Añadir impresoras en CUPS es bastante sencillo, se puede hacer por la interfaz web del CUPS, o permitirle al sistema operativo que se encargue automáticamente de su adición y configuración (normalmente, Canaima se encargará de instalar y poner en funcionamiento las impresoras conectadas).

Para añadir una impresora desde la interfaz web del CUPS solo navegue a la misma



usando <http://localhost:631> y pulse el botón: “Añadir impresora”

Se presentará un asistente que le permitirá añadir la impresora, Ud. debe definir, el nombre, la ubicación y una descripción breve (estos dos últimos parámetros son opcionales)



Luego de definir el nombre, ubicación y descripción, se le da al botón “Siguiente”, CUPS nos preguntará entonces el tipo de conexión de la impresora. Si esta se encuentra conectada directamente al equipo por el puerto USB o el puerto paralelo, podrá ver entre las opciones el nombre del dispositivo y este será seleccionable desde la lista desplegable que presentará. Otras opciones comunes son: una impresora de red en otro equipo CUPS: *internet printing protocol (ipp)*, impresora de red HP: *Appsocket / Jetdirect* y una impresora compartida por un equipo con SO Windows: *Windows printer vía Samba*. Luego de elegir el tipo de conexión, haga clic en “Siguiente”.



Si la impresora que agregó es de red o está compartida por otro sistema, el siguiente paso le preguntará su ubicación en la red. Normalmente bastará con colocar la dirección IP o el nombre de máquina y nombre de la impresora en el formato que especifica el campo. Si es una impresora compartida por un sistema Windows, deberá colocar la dirección en formato SMB, como: `//servidor/impresora`. Si la cola de impresión Windows requiere autenticación y el usuario válido difiere del que usa para iniciar sesión en su sistema, deberá colocar: `//usuario:contraseña@servidor/impresora`. En caso de que la impresora sea local, este paso no se mostrará.

A continuación se le preguntará el fabricante de la impresora. Una lista desplegable mostrará los fabricantes. Seleccione aquel que corresponda con su impresora y haga clic en “Siguiente”, para ir a la selección del modelo específico. También tiene la opción de cargar un archivo PPD específico para la impresora. Esto solo se hace si el fabricante no está listado y Ud. posee dicho archivo.

Al definir el modelo de su impresora el asistente habrá finalizado la adición de la misma. Puede cambiar sus opciones específicas navegando hasta la pestaña impresoras y haciendo clic en “cambiar opciones de impresora”.

Opcionalmente, puede compartir la impresora con otros sistemas haciéndola pública, esto se hace con facilidad, desde la pestaña de impresoras, haciendo clic en la opción “Publicar impresora” para cada impresora que Ud. desee compartir con otros sistemas.

Administración de colas

Además de la adición y configuración de impresoras, podemos gestionar las colas



de impresión del CUPS. Esto se realiza de forma sencilla desde la interfaz web de CUPS, en la pestaña “Trabajos”. Allí podemos ver los trabajos completados y/o los trabajos en proceso. Los cuales podemos pausar, reanudar, cancelar o mover de cola. Nótese que algunos tipos de impresora no permiten mover sus trabajos a otras ya que los *rasters*



Tema 3: Integración con servidores SMB/CIFS. (Samba)

La integración con el servicio Samba²⁶ es transparente al hacer la instalación del mismo en el sistema. Samba solo exportará, creando un recurso compartido para sus clientes, las impresoras que están explícitamente compartidas por CUPS. Si desea definir autenticación para las colas, deberá modificar el recurso compartido correspondiente a la impresora en la configuración de Samba.

²⁶ <http://www.samba.org>, un servicio para integración y operación de redes tipo Windows 98/NT/2000/XP/Vista/7 en sistemas tipo Unix y GNU/Linux.



UNIDAD IX: Servicio de almacenamiento remoto/compartido con NFS

Tema 1: Sistema de archivos de red NFS.

El Sistema de archivos de red NFS²⁷ es probablemente el servicio de red más prominente que usa RPC²⁸. Permite acceder a archivos en anfitriones remotos exactamente en la misma manera que se accedería si fueran locales. Una mezcla de soporte en el núcleo y demonios en espacio de usuario en el lado del cliente, junto con un servidor NFS en el lado del servidor, hace esto posible. Este acceso a los archivos es completamente transparente al cliente y funciona con varias clases de servidores y arquitecturas anfitrionas.

NFS es ubicuo e interoperable entre los sistemas POSIX, quiere decir que tanto en GNU/Linux como en sistemas tipo Unix y en los sistemas directamente basados en él, es posible utilizar NFS para servicios de archivo.

Funcionamiento

NFS utiliza una arquitectura cliente/servidor estándar. La porción del servidor consiste en discos que contienen sistemas de archivos compartidos que son visibles a los clientes de una red. Este proceso por medio del cual un servidor publica sus sistemas de archivo a la red es llamado *exportación*. La porción del servidor NFS también se encarga del bloqueo de archivos y del manejo de cuotas.

²⁷ Siglas en inglés de: *Network Filesystem*

²⁸ Llamado a procedimiento remoto, Siglas en inglés de: *Remote Procedure Call*



Para los clientes solo es necesario montar dichos sistemas de archivos exportados, como si estuviesen montando cualquier otro sistema de archivos en su máquina local.

Ventajas y Desventajas

Ventajas

- NFS posee administración centralizada.
- Si se utiliza con un medio de autenticación distribuido será muy sencillo actualizar los privilegios de los usuarios.
- Ya que cualquier sistema de archivos exportado puede montarse de forma transparente en el cliente, se pueden mantener en red datos y aplicaciones para ahorrar espacio en disco en el cliente.
- Provee un sistema más homogéneo para que los usuarios puedan, en un entorno de red, utilizar sus datos, aplicaciones y personalizaciones desde cualquier computador.

Desventajas

- Es muy sensible a la congestión de la red.
- Sufre de pérdida de rendimiento considerable cuando algún proceso del servidor hace un uso intensivo del disco duro donde reside el sistema de archivos compartido.
- Tiene problemas de seguridad inherentes, ya que su diseño asume que la red es confiable.

Tema 2: Implementando un servidor NFS

NFS funciona de la siguiente manera: primero, un cliente intenta montar un directorio de un anfitrión remoto en un directorio local justo de la misma manera que si fuese un dispositivo físico. Sin embargo, la sintaxis usada para especificar el directorio remoto es diferente. Por ejemplo, para montar /home desde el anfitrión maquina2 bajo el directorio /users en un computador de nombre maquina1, el administrador escribiría la siguiente orden:



```
root@maquina1# mount -t nfs maquina2:/home /users
```

mount tratará de conectar con el demonio remoto a través del servicio mountd (el demonio de NFS) de maquina2 por RPC. El servidor verificará si maquina1 tiene permiso para montar el directorio en cuestión, en cuyo caso, devuelve un descriptor de archivo.

Este descriptor será usado en todas las peticiones subsecuentes que se hagan sobre los archivos bajo /users. Cuando alguien accede a un archivo sobre NFS, el núcleo manda una llamada de RPC a mountd en la máquina servidor. Esta llamada toma el descriptor de archivo, el nombre del archivo a acceder y los identificadores de usuario y grupo del usuario como parámetros. Éstos se usan en la determinación de los derechos de acceso al archivo especificado. Para prevenir que usuarios no autorizados lean o modifiquen archivos, los identificadores de usuario y grupo deben ser iguales en ambos anfitriones... En la mayoría de las implementaciones de Unix, la funcionalidad NFS de cliente y servidor se implementan como demonios a nivel de núcleo que arrancan desde el espacio de usuario al arrancar la máquina. Éstos son los Demonios NFS (rpc.nfsd) en el anfitrión servidor, y Block I/O Daemon (biodev) en el anfitrión cliente.



Para mejorar el rendimiento, el servidor NFS realiza las operaciones de E/S usando prelectura y postescritura asíncrona; también, varios demonios mountd usualmente se ejecutan concurrentemente. La implementación actual de NFS de Linux es un poco diferente del NFS clásico en la que el código de servidor se ejecuta enteramente en espacio de usuario, así que ejecutar múltiples copias simultáneamente es más complicado. La implementación actual de mountd ofrece una característica experimental que permite soporte limitado para múltiples servidores.

El Servidor

El primer paso, como ya hemos visto, es instalar todos los paquetes que den soporte al servidor NFS:



```
# aptitude install nfs-kernel-server
```

Esto instalará todas las dependencias y paquetes necesarios para el funcionamiento del servicio NFS.


El primer servicio importante que se inicia, de no tenerlo ya instalado es portmapper. Este servicio es quien maneja las conexiones RPC de los clientes NFS. Para iniciarlo, de no estarlo una vez instalado el paquete del servidor NFS, ejecute:



```
# invoke-rc.d portmapper start
```


Para ver el estatus del servicio portmapper se puede usar el comando `rpcinfo` que muestra los servicios RPC que se están ejecutando, usando la opción “-p” para especificar la máquina que deseamos supervisar. Si no se provee el argumento de máquina, se asume que se ejecutará contra la máquina local.

Ejemplo:

| | | | | | | |
|---|---------------------------|------|-------|--------|------------|--|
|  | # <code>rpcinfo -p</code> | | | | | |
| | programa | vers | proto | puerto | | |
| | 100000 | 2 | tcp | 111 | portmapper | |
| | 100000 | 2 | udp | 111 | portmapper | |
| | 100024 | 1 | udp | 37592 | status | |
| | 100024 | 1 | tcp | 58672 | status | |

Antes de que NFS se inicie por sí mismo, debe ser configurado. Existe un único archivo de configuración que se llama `/etc/exports`. Cada línea muestra la ruta exportada seguido de una lista de clientes a los que se permite el acceso. Se pueden añadir opciones al final de cada nombre de cliente o dirección permitida.

Un ejemplo de el archivo `/etc/exports` puede ser el siguiente:

| | | | |
|---|-------------------------|--|--|
|  | <code>/usr/local</code> | | <code>192.168.0.0/255.255.255.0(ro)</code> |
| | <code>/home</code> | | <code>192.168.0.0/255.255.255.0(rw)</code> |

En este ejemplo se exportan los directorios `/usr/local` (en modo solo lectura) y `/home` (en modo lectura y escritura) a todos los computadores dentro de la red `192.168.0.0/255.255.255.0`

Como se puede ver se aceptan tipos de nombres de cliente como dirección IP o una subred. Aunque también puede especificarse: nombre de la máquina, caracteres comodín en un nombre de dominio (por ejemplo: máquina*.midominio.org), un netgroup (@grupo) si se usa NIS o LDAP, entre otras.

Algunas de las opciones de permisología más importantes son las siguientes:

- *rw* (lectura/escritura): el cliente puede leer y escribir en el sistema exportado.
- *ro* (solo lectura): el cliente solo puede leer el sistema exportado.
- *root_squash* : es preferible que un usuario root del cliente no pueda escribir con permisos de root. Para impedirlo, UID/GID 0 (p.e. root) en el lado del cliente se traduce en el usuario nobody. Esta opción está activada por defecto, pero se puede cancelar con *no_root_squash*.
- *all_squash* : todos los clientes que acceden al sistema exportado utilizan el UID/GID de nobody.
- *anonuid*, *anongid*: el usuario nobody ahora usa los UID y GID definidos por estas opciones.

Ahora debemos que iniciar el servidor NFS. Comprobamos nuevamente que todo está funcionando con el comando `rpcinfo`. Incluso podemos inicializar el servidor para los protocolos.



```
# invoke-rc.d nfs-kernel-server start
```

Si mientras está en operación, hacemos cambios en el archivo de configuración `/etc/exports`, debemos sincronizar esos cambios en el servicio. El comando `exportfs`



transmite esta información a nuestros servidores: La opción `-r` sincroniza el archivo `/etc/mtab` con el archivo `/etc/exports`. La opción `-v` muestra juntos todos los sistemas de archivos exportados junto con sus opciones. Después de ponerse en marcha el servidor NFS, los siguientes archivos contienen información importante:

- `/var/lib/nfs/rmtab`: cada línea muestra el nombre del cliente y el sistema de archivos importado desde este servidor.
- `/var/lib/nfs/etab`: el archivo `/etc/exports` solo contiene una lista de peticiones. `etab` está creado por `exportfs`. Contiene en cada línea información detallada sobre las opciones usadas cuando se exporta un sistema de archivos a un solo cliente. Es el archivo de referencia usado por `mountd` cuando es arrancado
- `/proc/fs/nfs/exports` contiene la lista de clientes conocida por el núcleo.
- `/var/lib/nfs/xtab`: se usa por precisión cuando `etab` contiene nombres de clientes y grupos de máquinas con comodines. Este archivo solo contiene nombres explícitos de máquinas.

Cuando un cliente quiere acceder a un sistema de archivos, empieza haciendo una petición a `mountd`. Entonces se busca en `etab` si la petición está disponible. Se comprueba el núcleo para saber si el cliente tiene permitida la petición (comprobando `/etc/hosts.{allow, deny}`, reglas de cortafuegos, ...). El núcleo utiliza `exportfs` para la comprobación, permitiendo actualizar el archivo `/var/lib/nfs/etab`. Si, en este archivo, el sistema exportado tiene permitido ser exportado al grupo al que pertenece el cliente, entonces `mountd` informa al núcleo que actualice `xtab` con este nuevo cliente y su dirección.

Tema 3: Utilizando NFS a través del cliente integrado

El acceso al sistema de archivos exportado por NFS está controlado directamente por el núcleo. Éste tiene que haber sido compilado para soportar NFS. El archivo `/proc/filesystems` contiene una lista con todos los sistemas de archivos soportados directamente por el núcleo. Entonces, lo único que tiene que hacer es decir al núcleo que quiere acceder a un sistema exportado por NFS.

El comando `mount` permite acceder a diferentes sistemas de archivos. Informa al núcleo que está disponible un nuevo sistema de archivos indicando su tipo, su dispositivo y su punto de montaje. Como se explicó antes se puede usar la opción `-t` para indicar el tipo del sistema de archivos a usar. Para NFS, escribimos: “`-t nfs`”.

`mount` tiene sus propias opciones para NFS. Por ejemplo, se pueden utilizar las opciones `rsiz` y `wsiz` para cambiar el tamaño de los bloques para lectura o escritura. Puede combinar opciones específicas de NFS con opciones más generales como `intr`, `noexec` o `nosuid`. La página de manual `mount` muestra todas esas opciones.

Ilustraremos su utilización con un ejemplo.

Supongamos que la máquina `canaima` tiene un servidor NFS y exporta su directorio `/usr/local`. Cuando quiera acceder desde la máquina `tepuy`, tendrá que montar el directorio exportado de `canaima` a `tepuy`:



```
root@tepuy# mount -t nfs -o nosuid,hard,intr \
canaima:/usr/local /usr/local
```



El comando indica que estamos montando un sistema de archivos NFS (-t nfs), con las opciones *nosuid*, *hard* e *intr*. Los dos últimos argumentos son los más interesantes. El primero de ellos especifica el dispositivo a montar. Y el último el punto de montaje.

La opción *nosuid* evita que se puedan asignar los bits *setgid* o *setuid* sobre el sistema de archivos montado.

La opción *hard*, específica del montaje de sistemas de archivos NFS, indica que de haber algún fallo temporal contactando al servidor, el cliente reintentará infinitamente de contactarlo de nuevo (esto podría reducir el rendimiento del computador cliente)

Por último, la opción *intr* le indica al servidor que el cliente aceptará señales de interrupción sobre los procesos que estén utilizando los archivos, esto para garantizar una operación atómica que en caso de fallos de red evite la corrupción de los datos.

Como se pudo comprobar la sintaxis de `mount` para el caso de los sistemas de archivo NFS es distinta de la línea `mount` habitual, donde se especifica dispositivo y directorio. Aquí se especifica `servidor:directorio_exportado` en vez de dispositivo. El último argumento indica la localización del sistema de archivos en la parte cliente;

Siguiendo el último ejemplo, logramos compartir el directorio `/usr/local` de “canaima” con “tepuy” y así podemos evitar el tener que instalar programas en `/usr/local` más de una vez. Si queremos hacer de esta configuración algo permanente solo necesitamos modificar `/etc/fstab`, en el cliente.



UNIDAD X: Servicio SMB/CIFS con Samba

Tema 1: Introducción a las redes basadas en SMB/CIFS

Samba es una suite de código abierto que provee servicios de archivo e impresión a clientes SMB²⁹/CIFS³⁰ y permite la interoperabilidad entre servidores Linux/Unix y clientes Windows

Funcionalidades y Virtudes

Samba es un servicio con una gran cantidad de funcionalidades, la siguiente es un compendio de las funcionalidades más importantes de este paquete de software.

- Puede funcionar como controlador de dominio para redes Windows y Linux
- Gestiona impresoras y archivos compartidos para redes Windows y Linux
- Gestiona usuarios, grupos y políticas para impresoras, archivos y recursos compartidos
- Soporta una multitud de modelos de autenticación diferentes
- Es interoperable con servidores de plataformas propietarias y puede funcionar como controlador de dominio esclavo para esas redes.
- Navegación de recursos compartidos en redes mixtas
- Sincronización de credenciales
- Soporte WINS
- Controles de acceso al sistema de archivos

²⁹ *Server Message Block*, Protocolo de red de bloques de mensaje de servidor. Utilizado por redes Microsoft Windows

³⁰ *Common Internet File System*. Sistema de archivos común de internet. Un protocolo que está basado en el protocolo SMB y es compatible con él. Agregando funcionalidades.



Tema 2: Implementando un servidor Samba

En este tema cubriremos una instalación integrada con LDAP cubriendo así una funcionalidad característica de Samba y la forma en la cual más se lo suele utilizar.

Las herramientas necesarias para proporcionar el servicio unificado de controlador de dominio y directorio LDAP pueden ser instaladas fácilmente a través del sistema de empaquetado que ofrece la distribución Canaima GNU/Linux . Es necesario instalar diversos paquetes de software, comenzaremos con la instalación de Samba:



```
# aptitude install samba smbclient
```

Durante el proceso de instalación se creará un conjunto de archivos ubicados en diferentes partes de la estructura del sistema. De estos archivos se debe hacer mayor énfasis en los nuevos archivos de configuración para una de las herramientas centrales: Samba; sin dejar a un lado los archivos de configuración preexistentes y que tienen que ver con la configuración global de los mecanismos de autenticación del sistema y de búsqueda de nombres (PAM y NSS respectivamente). El conglomerado de archivos que se deben manipular para Samba, se muestra a continuación:



```
/etc/samba/smb.conf
```


Instalación del servidor OpenLDAP

Para la instalación del servidor OpenLDAP se utiliza el siguiente comando.



```
# aptitude install slapd ldap-utils
```

En consecuencia de la instalación algunos directorios son agregados y es importante conocerlos para lograr el efectivo manejo de estos:



```
/etc/ldap/slapd.conf  
/etc/ldap/ldap.conf
```

Instalación de Herramientas y Librerías Adicionales

Una herramienta adicional a la hora de integrar un controlador de dominio, implementado con Samba, y un directorio LDAP implementado con OpenLDAP es el paquete smbldap-tools. El mismo contiene una serie de scripts útiles para administración y gestión de cuentas de usuario y grupos (tanto Unix/Linux como Samba) que se encuentren almacenados en un directorio LDAP. Antes de usar estas utilidades es necesario realizar la instalación de ella, del siguiente modo:



```
# aptitude install libpam-ldap libnss-ldap smbldap-  
tools
```



Manipulación de Archivos de Configuración

El primer paso al configurar un servidor Samba que trabaje de manera colaborativa con un Directorio LDAP (OpenLDAP en este caso) en calidad de backend, es la edición de ciertos archivos de configuración vinculados a cada una de estas herramientas, con la finalidad de que se cumplan los requerimientos que son importantes para que ambos trabajen unificados, es decir, establecer los parámetros que serán utilizados para garantizar el intercambio de información entre estos servicios.

El archivo de configuración principal para Samba es smb.conf, allí se establecen por un lado todos los parámetros globales del servidor y que principalmente definen su modo de operación, nivel de participación en la red (PDC o BDC), backend de usuarios, entre otros. Por otra parte, se establecen los parámetros relacionados con los recursos que desean ser compartidos, además de las restricciones para los mismos.

Es importante destacar que este archivo se inicia con una sección del tipo [global] y al mismo tiempo contiene un total de cuatro (4) secciones especiales llamadas [homes], [print\$], [printers], [netlogon], respectivamente. La sección [global] es la más importante de todas ya que contiene una serie de parámetros que Samba usará para definir el comportamiento de todo el servidor, backend de usuarios y contraseñas, así como algunos parámetros asociados al comportamiento de todos los recursos establecidos como secciones adicionales y que a su vez se publican como recursos compartidos SMB/CIFS.

| SECCIONES | DESCRIPCIÓN |
|------------|---|
| [homes] | Le permite a los usuarios remotos acceder a sus documentos personales ubicados en el servidor <i>Samba</i> . Es por ello, que debe tener en cuenta que es lo que se necesita y cual el objetivo primordial de la utilidad de este servicio, a fin de poder establecer la permisología de acceso más adecuada a cada uno de los recursos de red. |
| [print\$] | Recurso compartido invisible para los clientes que contiene los recursos internos de impresión del sistema donde se aloja Samba |
| [printers] | Impresoras compartidas |
| [netlogon] | Directorio de login de los usuarios que hacen uso del sistema Samba. |

Smb.conf

El archivo **smb.conf** contiene los siguientes parámetros de configuración

por defecto:



1. Parámetros Globales:

```
[global]
```

```
workgroup = PRUEBA.NET
```

```
netbios name = maquina01
```

```
server string = Servidor "%h" - Controlador de  
Dominio Principal (Implementado con Samba %v)
```



```
dns proxy = no
wins support = yes
wins server = localhost
name resolve order = wins host bcast
interfaces = 172.16.2.226/23 127.0.0.0/8
hosts allow = 172.16.2.0/255.255.255.0 127.0.0.0/8
bind interfaces only = yes
log file = /var/log/samba/log.%m
    max log size = 1000
    syslog only = no
    syslog = 2
    log level = 0 auth:5 passdb:5
    security = user
    encrypt passwords = true
    server signing = auto
    pam password change = yes
    socket options = TCP_NODELAY SO_RCVBUF=8192
SO_SNDBUF=8192
```

2. Parámetros de controlador de Dominio:

```
domain logons = yes
domain master = yes
preferred master = yes
os level = 254
logon path =
```



```
logon script =
logon drive = F:
logon home =
time server = yes
guest account = guest
passwd program = /usr/sbin/smbldap-passwd %u
                passwd      chat      =
*Introduzca\ssu\sueva\s*\scontraseña*          %n\n
*Repita\ssu\sueva\s*\scontraseña:*              %n\n
*Contraseña\sactualizada\satisfactoriamente* .
add user script = /usr/sbin/smbldap-useradd -a -m
%u
add machine script = /usr/sbin/smbldap-useradd -w
-c "Estacion de trabajo %U - Dominio PRUEBA.NET" %u
add group script = /usr/sbin/smbldap-groupadd -a
%g
add user to group script = /usr/sbin/smbldap-
groupmod -m %u %g
set primary group script = /usr/sbin/smbldap-
usermod -g %g %u
delete user script = /usr/sbin/smbldap-userdel
"%u"
delete group script = /usr/sbin/smbldap-groupdel
"%g"
delete user from group script =
/usr/sbin/smbldap-groupmod -x %u %g
```



3. Parámetros de conexión a LDAP:

```
passdb backend = ldapsam:"ldap://localhost
ldap://ldap.prueba.net"
ldap passwd sync = yes
ldap suffix = dc=prueba,dc=net
ldap admin dn = cn=admin,ou=Dominio
Samba,dc=prueba,dc=net
ldap replication sleep = 3000
ldap group suffix = ou=Grupos,ou=Dominio Samba
ldap user suffix = ou=Usuarios,ou=Dominio Samba
ldap machine suffix = ou=Equipos,ou=Dominio
Samba
ldap idmap suffix = ou=Idmap,ou=Dominio Samba
ldap delete dn = yes
obey pam restrictions = yes
```

4. Recursos de Impresión

```
load printers = yes
printing = cups
printcap name = cups

[printers]
comment = All Printers
path = /var/spool/samba
```



```
printable = Yes  
guest ok = Yes  
read only = Yes  
browseable = Yes  
create mask = 0700
```

```
[print$]  
comment = Printer Drivers  
path = /var/lib/samba/printers  
browseable = yes  
read only = yes  
guest ok = no  
write list = root smbadmin @"Administradores de  
Impresion" @"Domain Admins"
```

5. Recursos Compartidos

Definición del recurso compartido asociado a las carpetas personales de cada usuario

```
[homes]  
comment = Documentos personales del usuario %U  
path = /home/%U  
browseable = no  
read only = no  
create mask = 0700  
directory mask = 0700
```

```
valid users = %S  
# Definición del recurso compartido especial  
"netlogon"; empleado por los clientes Windows  
durante el inicio de sesión en el dominio
```

```
[netlogon]  
comment = Network Logon Service  
path = /srv/archivos/netlogon  
guest ok = Yes  
read only = no  
share modes = no  
browseable = no
```

A continuación se describe la estructura que deben poseer los archivos
/etc/smbldap-tools/smbldap.conf y /etc/smbldap-
tools/smbldap_bind.conf respectivamente:

smbldap.conf



#Configuración General

#Identificador de seguridad base (SID) del dominio

SID="S-1-5-21-2195632384-0123456789-9876543210"



*# Nombre del dominio al cual esta vinculado el
servidor Samba*

sambaDomain="PRUEBA.NET"

Configuración LDAP

#

Servidor LDAP esclavo

slaveLDAP="127.0.0.1"

Puerto LDAP de escucha en el servidor esclavo

slavePort="389"

*# Servidor LDAP maestro (necesario para operaciones
de escritura)*

masterLDAP="127.0.0.1"

Puerto LDAP de escucha en el servidor maestro

masterPort="389"

Sufijo LDAP base

suffix="dc=prueba,dc=net"



*# Estructura del arbol LDAP donde son almacenados
los usuarios*

```
usersdn="ou=Usuarios,ou=Dominio Samba,{suffix}"
```

*# Estructura del arbol LDAP donde son almacenados
los computadores y equipos*

```
computersdn="ou=Equipos,ou=Dominio Samba,{suffix}"
```

*# Estructura del arbol LDAP donde son almacenados
los grupos* groupsdn="ou=Grupos,ou=Dominio Samba,{suffix}"

Parámetros adicionales

#

```
ldapTLS="0"
```

```
verify="none"
```

```
idmapdn="ou=Idmap,ou=Dominio Samba,{suffix}"
```

```
sambaUnixIdPooldn="sambaDomainName=prueba.net,{suffix}"
```

```
scope="sub"
```

```
hash_encrypt="SSHA"
```



#

Configuración Unix/Linux

#

userLoginShell="/bin/false"

userHome="/home/%U"

userHomeDirectoryMode="700"

userGecos="Cuenta de Usuario/Maquina de prueba"

defaultUserGid="3000"

defaultComputerGid="3001"

skeletonDir="/etc/skel"

defaultMaxPasswordAge="60"

#

Configuración Samba

#

*Ruta UNC indicando la ubicación de la unidad
(drive) remota de documentos de usuario*

userSmbHome="//%L/%U"

*Ruta UNC indicando la ubicación de los perfiles
móviles de usuario (Cadena vacía para su
desactivación)*

userProfile=""



Etiqueta de unidad mapeada por defecto para los documentos remotos de los usuarios

userHomeDrive="F:"

Nombre del script netlogon por defecto para los usuarios

userScript=""

Dominio agregado por defecto en el atributo LDAP "mail" de los usuarios, cuando el comando "smbldap-useradd -M" es empleado

mailDomain="prueba.net"

smbldap_bind.conf



Definición de credenciales para acceso tanto al servidor LDAP maestro como esclavo

slaveDN="cn=smbadmin,ou=Dominio

Samba,dc=prueba,dc=net"

slavePw="unaclave"

masterDN="cn=smbadmin,ou=Dominio

Samba,dc=prueba,dc=net"

masterPw="unaclave"



Por otro lado, se deben establecer configuraciones en los archivos vinculados con el servidor OpenLDAP, para lograr la interacción con el servicio de Controlador de Dominio y así cumplir con la integración de manera correcta.

Archivo slapd.conf

En este archivo se manejan todos los parámetros que definen el funcionamiento de OpenLDAP. Los cuales se encuentran establecidos de la siguiente manera.

Parámetros Globales

Los parámetros dentro de esta sección afectan el funcionamiento de todo el Servidor OpenLDAP. Cualquier definición antes de un parámetro database es considerado global, cabe mencionarse que los valores de parámetros globales pueden ser contrarrestados al nivel de bases de datos, esto es por ejemplo, si se define el parámetro access globalmente, es posible alterar el valor de este parámetro en "X" base de datos y el resto de las bases de datos permanecerán con el valor global.

Definición de etiquetas

- include: Este parámetro indica otros archivos de configuración utilizados por el Servidor OpenLDAP.
- pidfile: Contiene el número de proceso asignado al servidor LDAP durante el arranque.



Directivas Globales:

*# Definición de Esquemas y Clases de Objetos
(ObjectClass)*

```
allow bind_v2
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/samba.schema
include /etc/ldap/schema/amavis.schema
include /etc/ldap/schema/postfix.schema
```

*# Especificacion del lugar donde sera almacenado el
archivo PID.*

```
pidfile /var/run/slapd/slapd.pid
```

*# Lista de argumentos que son pasados al servidor
durante su arranque*



argsfile /var/run/slapd/slapd.args

*# Nivel de detalle en los registros de sucesos
(logs)*

loglevel 64 256

*# Donde se almacenan los módulos cargados
dinámicamente*

modulepath /usr/lib/ldap

Instrucciones para cargar los módulos necesarios

moduleload back_hdb

moduleload unique

moduleload syncprov

*# Definición del número máximo de elementos
retornados en un proceso de búsqueda*

sizelimit 2000

*# Cantidad actual de CPU's que son usados para el
proceso de indexado*

tool-threads 8



Opciones asociadas al uso de SSL/TLS

```
TLSCACertificateFile /etc/ssl/cert/ca.crt
TLSCertificateFile    /etc/ssl/certs/ldap.crt
TLSCertificateKeyFile /etc/ssl/certs/ldap.key
TLSVerifyClient       demand
```

2. Parámetros por Base de Datos:

Directivas específicas para la base de datos #1: tipo hdb

```
backend          hdb
```

Directivas específicas que aplican solo a esta base de datos

```
database         hdb
```

La entrada base en el directorio gestionado por esta base de datos

```
suffix           "dc=prueba,dc=net"
```

Lugar donde se almacenan físicamente los archivos # de esta base de datos



```
directory      "/var/lib/ldap/prueba.net"

# Credenciales del administrador principal de dicho
# backend

rootdn         "cn=pruebaadmin,dc=prueba,dc=net"
rootpw         {SSHA}wmI5xP89u1Hwgs3519QSD0h2r6X8LIaH

# Parámetros de optimización de la base de datos
# Tamaño de cache para la base de datos

dbconfig set_cachesize 0 33554432 0

# Número de objetos que pueden ser bloqueados al
# mismo tiempo

dbconfig set_lk_max_objects 1500
# Número de bloqueos (Tanto solicitados como
# otorgados)

dbconfig set_lk_max_locks 1500

# Número de bloqueadores

dbconfig set_lk_max_lockers 1500

# Opciones de indexado en la base de datos #1: bdb
```



```

index      objectClass      eq
index      uid,cn,sn,givenName,mail,memberUid
           eq,sub
index      title      sub
index      ou      eq
index      gidNumber eq
index      uidNumber eq
index      sambaSID                      eq
index      sambaPrimaryGroupSID      eq
index      sambaDomainName            eq
index      entryCSN,entryUUID eq

```

*# Activación de la característica que permite
guardar las fechas en la cual un objeto de la base
de datos fue modificado por ultima vez*

```

lastmod      on
checkpoint   512 30

```

*3. Políticas de acceso al directorio (Listas de
control de acceso):*

```

access to
attrs=uidNumber,gidNumber,sambaNextRid,sambaNextUse
rRid,sambaNextGroupRid  by
dn.children="ou=Administradores,ou=Usuarios,dc=prue

```

```
ba,dc=net" write
```

```
by dn="cn=smbadmin,ou=Dominio  
Samba,dc=prueba,dc=net" write
```

Archivo ldap.conf



```
# Definición de datos básicos del servicio de  
directorío LDAP  
# Datos empleados por utilidades del paquete ldap-  
tools  
BASE dc=prueba,dc=net  
URI ldap://localhost
```

Configuración PAM

Podrá observarse la configuración de este servicio para este caso de forma puntual. Si ud desea ahondar en materia acerca de la configuración de PAM, puede remitirse al manual de herramientas de aseguramiento del sistema operativo, disponible en capacitacion.softwarelibre.gob.ve



Archivo nsswitch.conf



Definición de bases de datos para búsquedas de usuario

```
passwd:                                compat [SUCCESS=return] ldap  
[UNAVAIL=return]
```

Definición de bases de datos para búsquedas de grupos

```
group:                                compat [SUCCESS=return] ldap  
[UNAVAIL=return]
```

Definición de bases de datos para búsquedas de contraseñas

```
shadow:                                compat
```

Definición de bases de datos para búsquedas de nombres de hosts

```
hosts:                                files mdns4_minimal  
[NOTFOUND=return] dns mdns4
```

Definición de bases de datos para búsquedas de subredes

```
networks:                             files  
protocols:                             db files  
services:                             db files
```



```
ethers:      db files
rpc:         db files
netgroup:    nis
```

Algunos de parámetros más importantes y que por lo general se modifican en los archivos de integración NSS-LDAP (`libnss-ldap.conf`) y PAM-LDAP (`pam-ldap.conf`) respectivamente se muestran a continuación. Si bien esto es solo un fragmento de la información encontrada en dichos archivos, el resto de los parámetros se encuentran comentados indicando su valor por defecto.

Archivos `pam-ldap.conf` y `libnss-ldap.conf`



```
# El valor DN (distinguished name) base para las  
búsquedas en el directorio LDAP
```

```
host localhost
base dc=hidro,dc=net
```

```
# Cadena URI especificando el servidor LDAP al cual  
conectarse
```

```
uri ldap://127.0.0.1/
```

```
# Versión del protocolo LDAP a usar (el valor por  
defecto es la versión 3)
```



ldap_versión 3

#

Parámetros por defecto

#

scope sub

timelimit 300

bind_policy hard

idle_timelimit 3600

nss_paged_results yes

pagesize 500

nss_reconnect_tries 1

nss_reconnect_sleeptime 1

nss_reconnect_maxconntries 1



Tema 3: Administración del Controlador de Dominio

Labores Comunes de Administración

A continuación se mencionan algunas de las actividades de configuración que se realizan con más frecuencia a la hora de usar un servicio de controlador de dominio integrado con un directorio LDAP. Para el proyecto se escogió como herramienta de administración predilecta a la herramienta web phpldapadmin así que todas las actividades aquí descritas pueden ser realizadas con mayor facilidad a través de la misma. Mientras tanto, para efectos didácticos, en los párrafos próximos de este documento las actividades específicas de administración serán realizadas a través de la consola del sistema y empleando los scripts provistos en el paquete smbldap-tools.

Creación, Modificación y/o Eliminación de Cuentas de Usuario

En este caso se indican comandos para tratar con cuentas de usuario POSIX así como cuentas de usuarios SAMBA.



*# Creación de cuenta posix (solo usada para sistemas
Unix/Linux)*

```
maquina01:/etc# smbldap-useradd -m <login usuario>
```

*# Creación de cuenta samba (solo usada para sistemas
Windows)*



```
maquina01:/etc# smbldap-useradd -a -m -c "<nombre  
completo usuario>" <login usuario>
```

Eliminación de cuentas

```
maquina01:/etc# smbldap-userdel <login usuario>
```

*# Modificación de diversos atributos a una cuenta de
usuario SAMBA Modificación del HomePath (Unidad de
acceso a) de un usuario documentos personales*

```
maquina01:/etc# smbldap-usermod --sambaHomePath <ruta  
remota> <login usuario>
```

Activar una cuenta existente como cuenta Samba/Windows

```
maquina01:/etc# smbldap-usermod -a <login usuario>
```

*# Bloquear/Desbloquear la contraseña UNIX del usuario
(bloquear cuenta UNIX). Impide inicio de sesión en
servicio Linux diferentes al inicio de sesión en
estaciones de trabajo*

```
maquina01:/etc# smbldap-usermod -L <login usuario>  
cosrv01:/etc# smbldap-usermod -U <login usuario>
```

*# Bloquear/Desbloquear una cuenta de usuario del dominio
(impide inicio de sesión en las estaciones de trabajo)*


```
maquina01:/etc# smbldap-usermod -I <login usuario>  
maquina01:/etc# smbldap-usermod -J <login usuario>
```

Configuración de Contraseña de Usuario.



Establece nueva contraseña (actualiza contraseña UNIX y Samba)

```
maquina01:/etc# smbldap-passwd <login usuario>
```

Activar la solicitud de cambio de contraseña en el próximo inicio de sesión

```
maquina01:/etc# smbldap-usermod --sambaPwdMustChange 1  
<login usuario>
```

Cambiar la fecha de expiración de la cuenta de usuario

```
maquina01:/etc# smbldap-usermod --sambaExpire <YYYY-MM-DD HH:MM:SS>
```

Creación, Modificación y/o Eliminación de Grupos de Usuarios

En este caso se indican comandos para tratar con grupos de usuario POSIX así como grupos de usuarios SAMBA.



Creación de un nuevo grupo

```
maquina01:/etc# smbldap-groupadd -a <nombre grupo>
```

*# Modificación de usuarios dentro de un grupo Agregación
de usuarios nuevos al grupo*

```
maquina01:/etc# smbldap-groupmod -m  
<usuario1,usuario2,...,usuarioN> <nombre grupo>
```

Eliminación de usuarios del grupo

```
maquina01:/etc# smbldap-groupmod -x  
<usuario1,usuario2,...,usuarioN> <nombre grupo>
```

*# Modificación remota de las membresías de usuario/grupo
en los grupos locales de las estaciones de trabajo.*

```
maquina01:/etc# net rpc group addmem "Adminsitradores"  
"PRUEBA.NET\Soporte Helpdesk" -U smbadmin -S  
<estacion_trabajo_remota>
```



Creación, Modificación y/o Eliminación de Cuentas de Maquinas del Dominio

En este caso se indican comandos para tratar con cuentas de maquinas POSIX así como cuentas SAMBA.



Creación de una nueva cuenta SAMBA de maquina

```
maquina01:/etc# smbldap-useradd -w <nombre de maquina>
```

Eliminación de una cuenta SAMBA de maquina

```
cosrv01:/etc# smbldap-userdel <nombre de maquina>$
```

Habilitación/Deshabilitación de una cuenta SAMBA de maquina

```
maquina01:/etc# smbldap-usermod -I <nombre de maquina>
```

```
maquina01:/etc# smbldap-usermod -J <nombre de maquina>
```



UNIDAD XI: Interactuando con el Kernel LINUX

Tema 1: Definición de kernel

El *kernel* es el núcleo del sistema operativo. El actúa como un intérprete entre el usuario y el hardware. El kernel controla el acceso a los recursos de hardware de la computadora y determina como compartir estos recursos de una manera equitativa. El incluye los drivers del hardware, sistema de archivos, redes, manejo de memoria y administración de los procesos.

Virtualmente, el *kernel* puede ser configurado y optimizado para cualquier entorno, a través de recopilación del kernel mismo. Podrías querer recompilar el kernel para incluir drivers para hardware específico o para actualizar drivers para la corrección de errores o para incluir nuevas características.

El Kernel es de los primeros software a ejecutarse en un computador. Una vez el kernel ha terminado su iniciación, hace un llamado al proceso init (llamado el proceso padre de todos los procesos). El kernel provee todas las funcionalidades básicas a los programas así como el manejo de los recursos del sistema: hardware, procesos, memoria, I/O y sistema de archivos. La funcionalidad del kernel es mejorada sumándole/removiéndole código compilado llamado módulos o manejadores (drivers) de dispositivos.

Tipo de kernel

El kernel de Linux es un proyecto activo con un desarrollo continuado. En este proceso existen dos ramificaciones que viajan en paralelo. La primera es la denominada



versión estable del kernel y su intención es para producción solamente y no investigación. La otra es la versión de desarrollo y es donde los desarrolladores denominados prueban y analizan propuestas de mejoras y cambios importantes. Casi siempre es inestable, con problemas y características incompletas. Puedes reconocer el tipo de kernel por sus números de versión.

Versionado del kernel

El kernel linux ha tenido tres esquemas de numeración diferentes con los años, siempre respetando el formato: *A.B.C.(D)*. A saber:

Versiones tempranas:

- La primera versión del kernel fue la 0.01. A esta le siguieron las versiones 0.02, 0.03, 0.10, 0.11, 0.12 (esta fue la primera versión basada en la licencia GPL), 0.95, 0.96, 0.97, 0.98, 0.99 y luego la 1.0.40. Desde la 0.95 en adelante existieron múltiples parches y versiones diferentes.

El esquema antiguo (luego de la 1.0 y antes de la versión 2.6):

- El primer número o número **A** denota la versión del kernel. Rara vez es cambiado, y solo cambia cuando hay grandes cambios en el código y modificaciones importantes de concepto. Solo ha cambiado dos veces en la historia del núcleo Linux: en 1994 (versión 1.0) y en 1996 (versión 2.0).
- El segundo número, **B**, denota una revisión mayor del kernel.
 - El kernel utilizaba el método tradicional de par e impar para numerar el sistema. Quería decir que las versiones impares eran consideradas inestables, mientras que las pares eran consideradas listas para su uso en



entornos de producción.

- El número **C** indica la revisión menor del kernel. Este número era cambiado cuando se aplicaban parches de seguridad, correcciones de bugs, nuevas características o controladores eran implementados en el kernel.

Luego de la publicación de la versión 2.6.0 se descubrió que una ciclo de publicación mas corto sería beneficioso. Desde entonces:

- *A* y *B* son irrelevantes
- *C* es la versión del kernel
- *D* solo refleja correcciones de bugs o de seguridad a la versión definida por *C* (todo el desarrollo sucede en candidatos de publicación con sufijo '*rc*')

Un número **D** ocurrió la primera vez cuando un grave error, que requirió arreglo inmediato se encontró en el código pertinente a NFS de la versión 2.6.8. Sin embargo, no hubieron suficientes cambios relevantes para legitimizar la salida de otra versión menor (que podría haber sido 2.6.9). Así que entonces, 2.6.8.1 fue publicado, únicamente con el arreglo a ese error. Desde la versión 2.6.11, esta práctica fue adoptada como la nueva política de versionamiento. Algunas correcciones de bugs y parches de seguridad son migrados también a las versiones anteriores y manejados con una versión que incluye la letra *D*. El desarrollo solo ocurre entre versiones que llegan al número *C*.

Por otro lado, algunas veces después del número de versión habrán ciertas letras, como por ejemplo '*rc1*' o '*mm2*'. '*rc*' se refiere a candidato para publicación y significa que es una versión previa no oficial. Otras letras usualmente (pero no siempre) las iniciales de una persona o desarrollador. Esto indica que el kernel en cuestión es de desarrollo y corresponde a una de estas ramas. Por ejemplo, las letras '*ck*' pertenecen a aquellos kernels inestables de prueba hechos por Con Kolivas,



Núcleos precompilados

En general Canaima y las distribuciones basadas en Debian pueden obtener núcleos pre-compilados listos para instalar desde algunos de los repositorios de Debian. Un ejemplo notable es el repositorio llamado *backports*³¹, En este repositorio se encuentran versiones más nuevas del núcleo linux listas para instalar en distribuciones como Canaima y algunas versiones estables de distribuciones que estén basadas en Debian.

³¹ [Http://www.backports.org](http://www.backports.org)



Tema 2: Obteniendo un nuevo kernel.

Obteniendo las fuentes de un kernel estándar.

Antes de poder compilar el kernel, primero debes desempacar el código fuente en el sitio adecuado y preparar el directorio fuente. Para obtener los fuentes de un kernel hay diferentes mecanismo o lugares para encontrarlos. Una opción es vía ftp anónimo en *ftp.kernel.org*. En Debian 5 otra opción es mediante los paquetes de la distribución. Lo más habitual es que vengan empaquetados en un único fichero de nombre linux-x.y.z.tar.gz o bien con sufijo bz2 lo cuál indica que han sido comprimidos con bzip2. Lo habitual es realizar la descompresión de dicho archivo de código fuente en el directorio: /usr/src/linux



Si el paquete está en formato tar.gz:

```
# tar -xvzf linux-x.y.z.tar.gz -C /usr/src
```

Si el paquete está en formato tar.bz2:

```
# tar -xvjf linux-x.y.z.tar.bz2 -C /usr/src
```




Obteniendo las fuentes de un kernel Canaima GNU/Linux.

En el caso de que el equipo de desarrollo de Canaima realice una actualización al núcleo o tenga disponible núcleos para descargar, bien sea pre-compilados o en forma de código fuente siempre podrán descargarse desde <http://descargas.canaima.softwarelibre.gob.ve/desarrollo>. Es desde este lugar donde el equipo de desarrollo de Canaima pone a disposición del público paquetes fuentes del núcleo del sistema operativo. Recuerde que, de ser paquetes de código fuente del núcleo, deberá descomprimirlos en el directorio

Tema 3: Configurando el nuevo kernel.

Este es el paso más delicado en todo el proceso de compilación de un nuevo kernel ya que podría ocurrir que incluso no pudiera arrancarse el sistema si no se selecciona alguna opción fundamental para el mismo.

Para configurar el kernel, se hace práctico utilizar la interfaz de menú de consola que provee la misma herramienta de configuración del núcleo. Esta solo podrá ser accedida si instala el siguiente paquete:



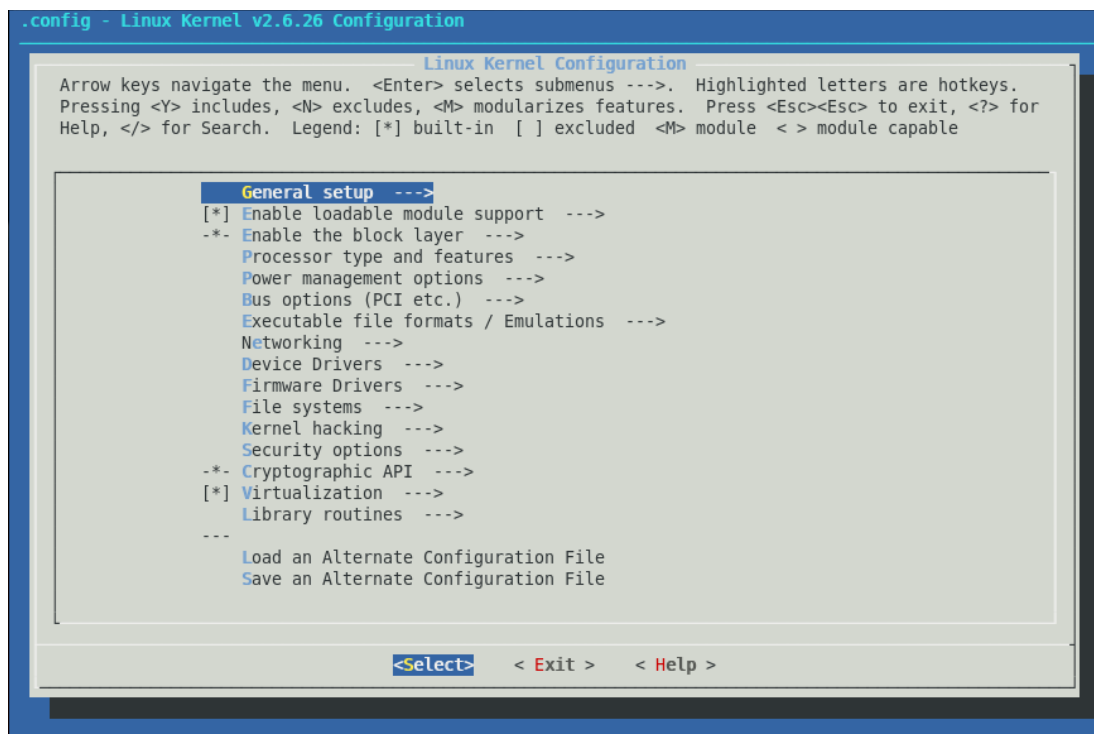
```
root@maquina:# aptitude install libncurses5w-dev  
libncurses5-dev
```

Una vez que haya instalado el paquete, debe situarse con nivel de superusuario en el directorio `/usr/src/linux/` donde se encuentra toda la estructura de directorios y ficheros que contienen los fuentes del kernel, donde a continuación puede ejecutar el comando:



```
root@maquina:/usr/src/linux# make menuconfig
```

Esto nos desplegará la interfaz de menú de consola para configurar el kernel.



Proceso de configuración

Al desplegarse la interfaz de configuración del kernel, deberemos navegar a cada una de las secciones del mismo y responder bien sea con un con si (y), con no (n), con la opción (m) de módulo o bien dejar la opción sin responder, Para cada una de las secciones que se encuentran en el menú de configuración.

Al responder sí (y), solicitamos que dicha funcionalidad sea compilada como parte integral del kernel. Si en cambio respondemos con módulo (m). El proceso de compilación del kernel construirá esta funcionalidad o controlador como un módulo aparte del kernel, lo que nos permitirá cargarlo y descargarlo en caliente cuando nuestro sistema esté funcionando con este núcleo.



Aunque es un proceso relativamente largo, esta interfaz nos puede mostrar información acerca de cada una de las funcionalidades que estamos activando. A veces también la activación de cierta funcionalidad desactiva otras, así que documente bien la configuración que requiere personalizar de manera de evitar desactivar alguna funcionalidad crítica de su sistema o el sistema podría quedar en un estado inútil, hasta el punto de no lograr arrancar del todo.

Tema 4: Instalando el nuevo kernel

Luego de terminar con la configuración, solo resta compilar el código fuente:



```
root@maquina:/usr/src/linux# make
```

Si todo sale bien y la compilación no retorna errores, nos toca instalar los módulos del sistema



```
root@maquina:/usr/src/linux# make modules-install
```

A continuación instalaremos el kernel de forma automática:



```
root@maquina:/usr/src/linux# make install
```

Con esto finaliza la configuración e instalación del núcleo. Para poder comenzar a usarlo, solo resta que reinicie el sistema y seleccione el núcleo que recién compiló, ya que tendrá una opción visible para ejecutarlo desde el menú del cargador de arranque.



UNIDAD XII: Introducción a la administración de servicios basados en Canaima GNU/Linux.

Tema 1: Servicios de correo electrónico

En sus comienzos el envío de correos electrónicos fue relativamente simple, y generalmente consistía en mover archivos desde un servidor a otro. Con el desarrollo de la red, el correo evoluciona a la compleja aplicación que conocemos hoy en día.

SMTP³² es el estándar de facto para el envío y recepción de correo electrónico que actualmente es utilizado en la Internet. Formalmente fue definido en el RFC³³ 821 y luego modificado en el RFC 1123.

El protocolo es ampliamente utilizado hoy en día y es formalmente denominado ESMTP³⁴, el cual es definido en el RFC 2821.



Para el manejo de buzones de correo electrónico se utilizan otros estándares de facto como POP3 e IMAP. Para estudiar su implementación en entornos basados en Canaima GNU/Linux puede referirse al manual del administrador correspondiente, en el cual se explica con lujo de detalles la implementación de dicho servicio en este entorno.

³² *Protocolo simple de transferencia de correos*, por sus siglas en inglés

³³ *Solicitud de comentarios*, por sus siglas en inglés

³⁴ *SMTP extendido*, por sus siglas en inglés



Tema 2: Sistema de Resolución de Nombres (DNS)

El *Domain Name System* (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. Anteriormente se alojaba un archivo llamado HOSTS que contenía todos los nombres de dominio conocidos, pero el crecimiento masivo de la red hizo que este procedimiento no resultara práctico y entonces se cuenta hoy en día con esta base de datos distribuida que almacena gran cantidad de información.

El DNS³⁵ es un sistema de nombres que permite traducir de nombres de dominio a direcciones IP y viceversa. Aunque la comunicación en Internet solo funciona en base a direcciones IP, el DNS permite que usemos nombres de dominio que además de ser más simples de recordar, permiten una abstracción útil para los usuarios en general

Elementos de un Sistema de nombres de dominio

Sistema que, al recibir una petición para resolver un nombre o una dirección, consulta su base de datos interna de equipos conocidos u otro sistema DNS para entregar bien sea la dirección IP de un dominio, o, si se le provee la dirección IP conocida, este devuelve el nombre de equipo y dominio al que pertenece.

35 Sistema de Nombre de Dominio, por sus siglas en inglés. (Domain Name System)



Son programas que se ejecutan en la computadora del usuario y que generan peticiones DNS de resolución de nombres a un servidor DNS. Por ej: ¿qué dirección IP corresponde a www.venezuela.com?

La configuración de equipos cliente DNS suele implicar la ejecución de las siguientes tareas administrativas:

- Configurar en el computador cliente los nombres para cada computador en la red.
- Configurar un sufijo DNS principal para el computador. El sufijo DNS principal del equipo es el nombre del dominio del cual este es miembro.
- Identificar los servidores DNS para los clientes y así realizar la consulta de resolución de nombres de forma rápida. Puede configurar los servidores DNS preferidos y alternativos. Los servidores DNS alternativos o suplentes se consultan cuando el preferido no contesta.



Si desea estudiar en profundidad la implementación de servicios DNS en entornos basados en Canaima GNU/Linux, puede referirse al manual de administración de servicios de resolución de nombres que cubre con detalles el proceso de instalación y configuración del mismo.



Tema 3: Servicios de Directorio basados en LDAP

Un servicio de directorio es una aplicación o un conjunto de aplicaciones que almacena y organiza la información de los usuarios de una red de computadores, permitiendo a los administradores gestionar el acceso de usuarios a los recursos sobre dicha red. Además, los servicios de directorio actúan como una capa de abstracción entre los usuarios y los recursos compartidos.

Los directorios tienden a contener información descriptiva basada en atributos y tienen capacidades de filtrado muy sofisticada. Los directorios generalmente no soportan transacciones complicadas ni esquemas de vuelta atrás (*roll back*) como los que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones. Las actualizaciones de los directorios son normalmente cambios simples y estos están optimizados mayormente para la rápida lectura de atributos, más que para su escritura constante.

Atributos LDAP

Al utilizar LDAP se puede consolidar información para toda una organización dentro de un repositorio central. Por ejemplo, en vez de administrar listas de usuarios para cada grupo dentro de una organización, puede usar LDAP como directorio central, accesible desde cualquier parte de la red. Puesto que LDAP soporta TLS y SSL, los datos confidenciales se pueden proteger de los curiosos.

LDAP también soporta un número de bases de datos (*backends*) en las que se almacena la información. Esto permite que los administradores tengan la flexibilidad para



desplegar la base de datos más indicada, para el tipo de información. LDAP tiene APIs³⁶ bien definidas, existe un número de aplicaciones acreditadas para LDAP, estas están aumentando en cantidad y calidad, las hay en distintos lenguajes de programación, tales como C, C++, Java, Perl, PHP, entre otros.



Puede, a partir del Manual correspondiente a los servicios de Directorio en Software Libre, particularmente para Canaima GNU/Linux estudiar a fondo su implementación y funcionamiento.

³⁶ Interfaces de programación de aplicaciones, por sus siglas en inglés



Tema 4: Respaldo y Recuperación

Las técnicas empleadas tradicionalmente en las organizaciones para diseñar e implementar las políticas de respaldos y recuperación de datos han evolucionado en los últimos años. Las principales causas han sido la evolución de la capacidad y funcionalidad de los grandes sistemas de almacenamiento, y la ya consolidada tecnología de redes de almacenamiento SAN³⁷. Esto, unido al considerable descenso en el coste de los productos de hardware, ha derivado en un aumento de implantaciones de estos entornos en pequeñas y medianas empresas.

Elementos para mantener la información segura.

La seguridad de la información comprende distintos elementos para mantener la información segura, estos requieren de un funcionamiento acoplado para lograr el éxito de los mismos. Existen básicamente tres áreas que componen las bases de la seguridad de la información para que ésta funcione de manera adecuada; estos elementos son: *confidencialidad*, *integridad* y *disponibilidad*, cada uno de estos requiere de diferentes herramientas y metodología para proteger la información en cada una de sus áreas.

- La **confidencialidad**, se refiere a mantener los datos de una manera que no permita que puedan ser vistos por personas no autorizadas. Esta información que es confidencial en su organización, pueden ser planes estratégicos, información financiera, información personal entre otras. Estos datos no solo deben ser protegidos de agentes externos, también se debe planificar una política que internamente mantenga la información a la vista de quien está autorizada para acceder a ella.

³⁷ Siglas en inglés de *Storage Area Network*, Red de almacenamiento de datos

- La **integridad**, es el factor que permite garantizar que la información no pueda ser cambiada o eliminada por personas no autorizadas, también hace referencia a que las personas que están autorizadas no realicen los cambios sin la debida aprobación. Mantener los datos sincronizados de manera adecuada entre los sistemas es llamado también integridad de los datos.
- **Disponibilidad**, ya que tener la información segura no es útil si la misma no se puede obtener al momento de necesitarla. Con el crecimiento exponencial de Internet ya no solo existe la preocupación de mantener la información segura de personas sin autorización, sino también que los que están autorizados puedan acceder a ella. El solo hecho de no poder tener una información al momento de requerirla, puede ser tan grave como el no tenerla. El elemento de disponibilidad también incluye la preparación ante los desastres que puedan ocurrir, y la habilidad de poder recuperarse rápidamente ante los mismos.



En este sentido, pueden usarse los elementos que se describen en profundidad en el manual de respaldo y recuperación en Canaima GNU/Linux para una comprensión completa del tema, así como ejemplos de implementación de una infraestructura de respaldo y recuperación con software libre de estándares abiertos.



Tema 5: Seguridad de la Información

Además de los temas que abarca el respaldo y recuperación de la información. Es importante hacer énfasis en otros temas basados en las premisas de integridad, disponibilidad y confidencialidad, como la seguridad lógica y física.

Elementos de seguridad lógica

Algunos de los elementos de la seguridad lógica son:

- **Identificador de usuarios:** identificador de usuario, también conocido como nombre de usuario o *login*, es un identificador único de la persona utilizado para acceder tanto a equipos locales como a repositorios de datos en red, como cuentas de correo electrónico y recursos de red compartidos.

Estos identificadores están basados en cadenas de caracteres alfanuméricos y son asignados de manera individual a cada usuario, los más conocidos son el nombre de usuario para acceder al equipo y la dirección del correo electrónico.

- **Autenticación:** es el proceso en el cual un sistema, un computador o una red intenta confirmar la identidad de un usuario. La confirmación de identidades es esencial para el establecimiento de controles de acceso, lo cual otorgará dependiendo del usuario, privilegios en los sistemas de archivos o red.

Identificación de doble vía: esta involucra que tanto el usuario como el sistema o la red, se intercambian identificadores compartidos para determinar que ambos son quienes dicen ser. Esto generalmente se realiza utilizando claves públicas. El mecanismo consiste en que el usuario envía su clave pública, el servidor determina que la conoce y reenvía al usuario su propia clave pública, el equipo del usuario verifica que esta es la clave del servidor y se establece la



comunicación entre ambos.

Elementos de la seguridad física

En el proceso del diseño, instalación e implementación de políticas de seguridad de información, la seguridad física debe ser el primer punto a tomar en cuenta.

No en vano, desde la antigüedad la seguridad física siempre ha sido una premisa en seguridad: las murallas y castillos son ejemplo de esto.

La **seguridad física** se puede dividir en tres elementos:

- Obstáculos para frustrar, detener o retardar posibles ataques.
- Alarmas y sistemas de detección de intrusos, guardias de seguridad, circuito de cámaras y monitores para que los atacantes sean identificados.
- Respuestas para repeler, capturar o frustrar atacantes al momento de detectar el mismo.

En un diseño de seguridad física bien implementado, estos elementos deben estar presentes complementándose unos a otros.

Para una comprensión completa de las necesidades de seguridad lógica y física en entornos basados en Canaima GNU/Linux, puede remitirse al manual que con este propósito se ha realizado, en el que encontrará información más completa y podrá hacer implementaciones de ejemplo que le ayudarán a tener un entendimiento profundo de estas previsiones.



Tema 6: Redes privadas virtuales

Las redes privadas virtuales o VPN³⁸, como son popularmente conocidas, se definen, en esencia, como una extensión de una red local y privada que utiliza como medio de enlace una red pública como por ejemplo: Internet. Es también posible utilizar otros tipos de conexión que involucren redes de área amplia, WAN (En inglés: *Wide Area Networks*) tales como: Relé de tramas (*Frame Relay*), enlaces en modo de transferencia asíncrona (ATM: *Asynchronous Transfer mode*) o Líneas de suscriptor digitales de alta tasa de bits (HDSL: *High bit rate Digital Subscriber Line*).

Este método permite enlazar dos o más redes simulando una única red privada, permitiendo así la comunicación entre computadores como si estuviesen en la misma red física; Asimismo, un usuario remoto se puede conectar individualmente a una LAN utilizando una conexión VPN, donde, en forma segura, puede utilizar aplicaciones no accesibles desde Internet para enviar datos, o actualizar información confidencial que necesita protegerse y enviarse por canales cifrados.

Usos comunes de las Redes Privadas Virtuales:

- Conexión entre diversos puntos de una organización a través de Internet.
- Conexiones de trabajadores domésticos o de campo con direcciones IP dinámica.
- Soluciones extranet para clientes u organizaciones asociadas con los cuales se necesita intercambiar cierta información en forma confidencial y privada, aunque no tengan acceso pleno al resto de la red interna de la organización.

Con todo esto, la implementación de redes privadas virtuales bajo Canaima

38 Siglas en inglés de: Virtual Private Networks



GNU/Linux se explica en detalle en el manual de VPN en entornos GNU/Linux (particularmente Canaima) con ejemplos que puede seguir el alumno para una comprensión profunda de su funcionamiento.



UNIDAD XIII: Apéndice I. Editor de archivos VIM.

Tema 1: Introducción a VIM

Vi (visual editor) es un editor de texto realizado originalmente por William Joy para la versión de UNIX de la universidad de Berkeley: BSD. Posteriormente este potentísimo editor se incorporó al System V de AT&T convirtiéndose en herramienta estándar.

VIM (vi Improved) es un moderno editor de GNU compatible con vi pero con muchas funcionalidades añadidas, entre otras deshacer multinivel, multiventanas, multibufes, coloreado de sintaxis, autocompletado de nombres de archivos, ayuda en línea, apertura de directorios.

```
VIM - VI Mejorado

versión 7.1.314
por Bram Moolenaar et al.
Vim es código abierto y se puede distribuir libremente

¡Patrocine el desarrollo de Vim!
escriba :help sponsor<Intro> para más información

escriba «:q<Intro>» para salir
escriba «:help<Intro> o <F1>» para obtener ayuda
type :help version7<Enter> for version info

0,0-1 Todo
```



El éxito de vi se debe a varios factores. El primero es que vi es un editor pensado para utilizar en pantalla completa; hay que tener en cuenta que antes los editores de texto solo mostraban la línea que estabas editando. De hecho vi está basado en el editor de líneas ex y se puede alternar entre un editor y otro mientras se trabaja o ejecutar comandos de ex en vi.

El segundo valor que popularizó vi es que todos sus comandos se realizan con el teclado alfanumérico permitiendo su uso indiscriminado entre terminales con distintas configuraciones. Así por ejemplo para mover el cursor no se utilizaban las flechas del teclado sino las letras h(izq)-j (abj)-k(arr)-l(der), aunque hoy día permite los dos sistemas. De nuevo hay que tener en cuenta que hasta el System V no existía apenas compatibilidad entre los distintos UNIX.

Hoy día los principales valores de vi son su universalidad (disponible en todos los sistemas UNIX yGNU), su eficiencia (es ligero y usa poca memoria) y sobre todo su potencia. Esta potencia viene de la distinción entre modo comando y modo edición y en la versatilidad de los comandos; todos, por sencillos que parezcan, pueden combinarse con repeticiones, marcas, rangos de líneas y/o expresiones regulares e ir complicándose cuanto se desee, con gran versatilidad.

vi, como veremos, es un editor con una curva de aprendizaje algo inclinada, pero fácil de utilizar. A lo largo del texto empezaremos con versiones sencillas de los comandos e iremos complicando los ejemplos como muestra de lo que se puede hacer. En las distribuciones GNU/Linux derivadas de Debian (Canaima, Ubuntu, entre otros.), para activar todas las funcionalidades de vim, hay que instalarlo mediante el siguiente comando:



```
# aptitude install vim
```



Tema 2: La tecla ESC

El principal uso de la tecla ESC es finalizar el modo edición para volver al modo comando. Además se utiliza para cancelar un comando a mitad. En todo caso con dos pulsaciones de la tecla ESC nos encontraremos en modo comando y sin ningún comando a medias. Si ya se encuentra en modo comando sin ningún comando que cancelar y pulsa ESC, vim le avisará emitiendo un breve pitido.



Tema 3: Algunas consideraciones sobre el texto

VIM distingue entre:

- Palabras → caracteres separados por espacios, tabuladores o saltos de línea.
- Expresiones → igual a las palabras pero considerando caracteres especiales (man vim para más información).
- Sentencias → caracteres entre los signos '.' (punto), '?' (interrogación) y '!' (exclamación) o saltos de línea.
- Líneas → texto entre dos saltos de línea (INTRO).
- Párrafos → texto separado por líneas en blanco.
- Además el final de archivo también termina todas ellas.

Principalmente los comandos de vim trabajan con palabras y líneas. Es importante tener en cuenta que una línea de texto puede ocupar más que el ancho de pantalla. En este caso vim nos la mostrará en varias líneas de pantalla, pero seguirá considerándose una única línea de texto.



vim

Pulse ':set number' para que le muestre los números de línea.

*Pulse '1G' para situar el cursor al principio del texto.
Pulse 'i' y escriba:*

Esta es una línea larga de texto que ocupa más que el ancho de la pantalla, por lo que el editor me la mostrará en varias líneas de pantalla, siendo solo una



línea de texto.

[Enter]

[Esc]

El texto queda como sigue:



1 Esta es una línea larga de texto que ocupa más que el ancho de la pantalla, por lo que el editor me la mostrará en varias líneas de pantalla, siendo solo una línea de texto.

2 Aquel que luche con monstruos

3 que procure no convertirse en un monstruo él mismo;

4 que tenga cuidado aquel que mire al abismo,

5 pues el abismo le devolverá la mirada.

6 F. Nietzsche

Teclee '1G' para situarse al principio del texto y después pulse '^G' (Ctrl+g). El editor mostrará: "prueba" line 1 of 6 -16%--

También puede pulsar varias veces '+' y '-' para comprobar que el nuevo texto es una sola línea de texto, aunque se muestre en dos líneas de pantalla.

Pulse ':set nonumber' para que desaparezcan los números de línea.



Borrando texto

Estando en modo comando, el modo simple de borrar es utilizar:

- x → borra el carácter indicado por el cursor
- X → borra el carácter anterior al indicado por el cursor
- D → borra desde el carácter indicado por el cursor hasta el final de línea
- dd → borra la línea indicada por el cursor

El comando 'd' (delete) borra desde el carácter indicado por el cursor hasta lo indicado por el siguiente comando de desplazamiento. Así:

- dw -- borra hasta el comienzo de palabra siguiente
- db -- borra hasta el comienzo de palabra anterior
- dxG -- borra hasta la línea x del texto, siendo x un número de línea
- dG -- borra hasta la última línea del texto
- d\$ -- borra hasta el final de línea (equivale a 'D')

Estando al principio del texto pruebe a borrar algunos caracteres y un par de palabras. Por último borre toda la línea con 'dd'.

Modo edición

Hasta ahora solo se ha visto el modo más sencillo de insertar texto con el comando 'i'. Este comando permite insertar delante del texto indicado por el cursor. Sin embargo existen multitud de comandos para cambiar al modo edición, distinguiendo entre modo inserción (el texto que escribimos se añade al existente) y modo reemplazo (el texto que

escribimos sustituye a texto existente). Los comandos principales son:

- a (append) → permite insertar texto detrás del texto indicado por el cursor.
- I (Insert) → permite insertar texto al inicio de la línea indicada por el cursor.
- A (Append) → permite insertar texto al final de la línea indicada por el cursor.
- o → añade una línea después de la actual y permite insertar texto.
- O → añade una línea antes de la actual y permite insertar texto.
- r (replace) → sustituye el carácter indicado por el cursor por otro carácter. (No hace falta pulsar ESC para volver al modo comando).
- R (Replace) → permite sobrescribir el texto a partir del carácter indicado por el cursor.
- s (substitute) → sustituye el carácter indicado por el curso por el texto introducido.
- S (Substitute) → sustituye la línea indicada por el cursor por el texto introducido.
- C (Change) -- sustituye desde el carácter indicado por el cursor hasta el final de línea.

El comando 'c' (change) cambia el texto desde el carácter indicado por el cursor hasta lo indicado por el siguiente comando de desplazamiento. Así:

- cw → cambia hasta el comienzo de palabra siguiente
- cb → cambia hasta el comienzo de palabra anterior
- cxG → cambia hasta la línea x del texto, siendo x un número de línea
- cG → cambia hasta la última línea del texto
- c\$ → cambia hasta el final de línea (equivale a 'C')



Teclee 'lG' para situarse al principio del texto y después pulse 'o' y escriba 'puede salir escaldado' y pulse ESC. Después pulse 'O' y escriba 'cuidado porque'



*y pulse ESC. Ahora pulse 'I' y escriba 'tenga mucho' y
ESC. El texto habrá quedado como sigue:*

Aquel que luche con monstruos
tenga mucho cuidado porque
puede salir escaldado
que procure no convertirse en un monstruo él mismo;
que tenga cuidado aquel que mire al abismo,
pues el abismo le devolverá la mirada.
F. Nietzsche



Tema 4: Otros comandos útiles

- Uno de los comandos más útiles en cualquier editor es el comando 'deshacer' (undo).

En vim el comando 'u' deshace el último cambio. Si se vuelve a pulsar 'u' deshace la acción de deshacer, es decir pulsar 'uu' hace que el texto quede igual.

- Pulse 'u' y desaparecerá 'tenga mucho' en la segunda línea. Vuelva a pulsar 'u' varias veces para comprobar su efecto.
- Vim tiene 'deshacer multinivel' lo que permite, según la configuración, que al pulsar varias veces 'u' se deshagan los últimos cambios que se han ido realizando.
- El comando 'J' elimina el salto de línea de la línea actual. De este modo une la línea siguiente al final de la línea actual.
- El comando ':w' escribe (guarda) el texto actual. También se puede utilizar ':w nombre_archivo2' con lo que guardará el texto en el archivo nombre_archivo2 (aunque seguiremos trabajando con nombre_archivo). ':2,5w nombre_archivo2' guarda las líneas 2 a 5 en nombre_archivo2.
- También se puede combinar el comando ':w' con el comando ':q' escribiendo ':wq'. El comando ':x' equivale a ':wq' (guardar los cambios y salir).
- El comando ':r nombre_archivo3' lee el archivo nombre_archivo3 y lo escribe después de la línea indicada por el cursor. ':5r nombre_archivo3' lo inserta en la línea 5.
- ^g (Ctrl+g) -- muestra información sobre el archivo, así como el número de línea (muy útil para desplazarse a una línea copiar desde donde estamos hasta la línea x, etc.)
- ^l (Control+l (ele)) -- redibuja la pantalla. Muy útil cuando el terminal hace que el scroll del texto monte las líneas y ya no sabes si lo que ves es realmente lo que pone



Repeticiones de comandos

Una de las posibilidades más versátiles de vim es la repetición de cualquiera de sus comandos. Hay dos modos básicos de hacerlo.

- A posteriori: . (punto) → repite el último comando de modificación del texto. Este último comando puede haber sido cambiar una palabra por una frase, añadir un texto a final de línea, borrar un párrafo, insertar una tabulación.
- A priori, se puede pulsar un número antes de ejecutar un comando y éste se repetirá tantas veces como hayamos indicado.

Ejemplos:

5+, nos llevará al inicio de la 5a línea después de la indicada por el cursor.

3Dd, borrará 3 líneas.

3J, unirá tres líneas.

8llaESC, insertará 8 veces la palabra la (insertará 'lalalalalalala').

2s(texto)ESC, sustituye dos caracteres por el texto introducido.

2cw(texto)ESC, sustituye dos palabras por el texto introducido.



UNIDAD XIV: Apéndice II. Sistema X.org

Tema 1: El sistema X.Org.

X es el componente de los sistemas Unix encargado de mostrar la información gráfica, en particular, de dibujar los iconos, fondos y ventanas en las que se ejecutan las aplicaciones y es totalmente independiente del sistema operativo.

El sistema de ventanas X distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor. El servidor provee servicios para acceder a la pantalla, teclado y ratón (determina la resolución de la pantalla y la profundidad de color, mueve el cursor del ratón alrededor de la pantalla, entre otras acciones) mientras que los clientes son las aplicaciones que utilizan estos recursos para la interacción con el usuario. De este modo, mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

X.Org es una implementación libre del sistema gráfico de ventanas X (también conocido como X11) que surgió como una bifurcación de Xfree86 después de un cambio de licencia que muchos consideran incompatible con la Licencia Pública General (GPL), esta ha sido adoptada por la mayoría de las distribuciones GNU/Linux.



Tema 2: X-Windows

UNIX y GNU/Linux no incorporan la interfaz gráfica de usuario dentro del núcleo, en su lugar, es implementada por programas a nivel de usuario. Esto se aplica tanto a entornos gráficos como al modo texto. Esta disposición hace que el sistema sea más flexible, pero tiene la desventaja de que, al ser simple, implementar una interfaz de usuario diferente para cada programa, dificulta el aprendizaje del sistema.

El entorno gráfico principalmente utilizado con GNU/Linux se llama Sistema X-Windows (X para abreviar X11). X tampoco implementa por sí mismo una interfaz de usuario, sino solo un sistema de ventanas. Es decir, las herramientas bases con las cuales se puede construir una interfaz gráfica de usuario. Algunos administradores de ventanas populares son: FVWM, ICEWM, BLACKBOX Y WINDOW MAKER, METACITY. Existen también dos populares administradores de escritorios, KDE y GNOME.



Tema 3: Modos VESA.

VESA (Video Electronics Standards Association - Asociación para estándares electrónicos y de video) es una asociación internacional de fabricantes de electrónica. Fue fundada por NEC en los años 80 del siglo XX con el objetivo inicial de desarrollar pantallas de vídeo con una resolución común de 800x600 píxeles. Desde entonces, la VESA ha realizado otros estándares relacionados con funcionalidades de vídeo en periféricos de los IBM PC y compatibles, como conectores, BIOS o características de la frecuencia, transmisión y sincronización de la imagen.

Los modos VESA más típicos son: hexadecimal y decimal.

Hexadecimal

| Colores | 640×480 | 800×600 | 1024×768 | 1280×1024 | 1600×1200 |
|------------------|---------|---------|----------|-----------|-----------|
| 256 (8 bits) | 0×0301 | 0×0303 | 0×0305 | 0×0307 | 0×031C |
| 32,768 (15 bits) | 0×0310 | 0×0313 | 0×0316 | 0×0319 | 0×031D |
| 65,536 (16 bits) | 0×0311 | 0×0314 | 0×0317 | 0×031A | 0×031E |
| 16.8M (24 bits) | 0×0312 | 0×0315 | 0×0318 | 0×031B | 0×031F |

Decimal

| Colores | 640×480 | 800×600 | 1024×768 | 1280×1024 | 1600×1200 |
|------------------|---------|---------|----------|-----------|-----------|
| 256 (8 bits) | 769 | 771 | 773 | 775 | 796 |
| 32,768 (15 bits) | 784 | 787 | 790 | 793 | 797 |
| 65,536 (16 bits) | 85 | 788 | 791 | 794 | 798 |
| 16.8M (24 bits) | 86 | 789 | 792 | 795 | 799 |



Tema 4: Reconfigurar servidor gráfico X.org

Si por alguna razón después de realizar la instalación del sistema operativo se necesita configurar el servidor gráfico de nuevo, por ejemplo, al no haber detectado en la instalación la resolución correcta del monitor, o en algún momento se cambia el monitor de la computadora y los parámetros que tienen configurados en el antiguo no funcionan con el nuevo, existe un script de configuración que ayuda a la reconfiguración del Servidor X sin necesidad de estar retocando a mano el archivo `/etc/X11/xorg.conf`.

Para invocar el script se recomienda que se inicie sesión como usuario root en una consola de texto (CTRL+ALT+F1) y seguir los pasos a continuación:

Antes de ejecutar el script, si ya se tiene configurado el servidor X, es recomendable que se realice una copia de seguridad del archivo `xorg.conf` de la siguiente manera:



```
# cp /etc/X11/xorg.conf /root
```

Una vez hecha la copia de seguridad, se ejecuta el script de configuración:
`dpkg-reconfigure xserver-xorg`



```
#dpkg-reconfigure xserver-xorg
```



Tema 5: Las secciones de xorg.conf

El archivo `/etc/X11/xorg.conf` contiene la configuración de X.Org y está dividido en secciones. Cada sección empieza con la instrucción `<Section>`, seguido por el nombre de la sección entre comillas y siempre termina con `<EndSection>`.

Sección “Modules”

La sección de módulos se utiliza para especificar que módulos deberían cargarse al inicio del servidor X. Normalmente esto es automáticamente determinado por el servidor xorg y muy rara vez se encuentra en el archivo de configuración. Esta sección es opcional, así como cualquier parámetro que en esta sección se configure.

Sección “ServerFlags”

En esta sección que es opcional se configuran opciones generales misceláneas del servidor xorg. Algunas de las opciones más importantes que pueden definirse aquí son:

- "DontZap" "booleano": Si el valor está configurado a "true" (verdadero) esta configuración evita el uso de la combinación de teclas Ctrl-Alt-Backspace para matar al servidor xorg.
- "DontZoom" "booleano": Si este valor es "true" no podrá cambiar la resolución en caliente con la combinación de teclas Ctrl-Alt-Keypad-Signo "+" y Ctrl-Alt-Keypad-Signo "-".
- "IgnoreABI" "booleano": Si este valor es "true" los módulos del servidor X

que no coincidan en versión con aquella del servidor serán cargados pese a las advertencias. (podría causar inestabilidad en el funcionamiento del servidor xorg)

Sección "Monitor"

Define las propiedades del monitor. Las especificaciones del Sync Horizontal definen cuánto ancho de banda puede soportar el monitor y es especificado en kilohertz. Esto ayuda a identificar qué resolución es capaz de soportar el monitor. El Refresco Vertical dice cuantas veces por segundo el monitor puede refrescar las imágenes. Estas dos especificaciones pueden ser definidas en rango de valores que los monitores pueden soportar. Se recomienda revisar las especificaciones en el manual del monitor o buscar las especificaciones usando unas de las herramientas de configuración disponibles.

Otros parámetros que se encuentran en la sección Monitor son los de los modes. Se tiene dos maneras de especificarlos: el primero, usar la directriz ModeLine y especificar todos los números en una línea. El segundo, usar la subsección Mode, especificando los parámetros con el uso de marcados (tags). En ambas, éstos parámetros le comunican al Servidor X qué frecuencias y posicionamiento usar para cada resolución.

Un ejemplo de la sección "Monitor"



Section "Monitor"

| | |
|------------|-----------------------|
| Identifier | "Failsafe Monitor" |
| Vendorname | "AOC" |
| Modelname | "AOC SPECTRUM 4V,4VA" |
| Identifier | "Failsafe Monitor" |



```

Vendorname      "AOC"
Modelname        "AOC SPECTRUM 4V,4VA,4Vlr & 4VlrA, 4Vn,
4VnA"
Horizsync        30.0-50.0
Vertrefresh      50.0-100.0
modeline         "800x600@56" 36.0 800 824 896 1024 600 601 603
625 +hsync              +vsync
modeline         "800x600@72" 50.0 800 856 976 1040 600 637 643
666 +hsync              +vsync
modeline         "800x600@60" 40.0 800 840 968 1056 600 601 605
628 +hsync              +vsync
modeline         "1024x768@60" 65.0 1024 1048 1184 1344 768 771
777 806 -vsync          -hsync
Gamma            1.0
EndSection        4VlrA, 4Vn, 4VnA"
Horizsync        30.0-50.0
Vertrefresh      50.0-100.0
EndSection

```

Sección "Device"

Especifica los parámetros de la tarjeta de video. En la misma se puede especificar el chipset que el adaptador utiliza, cuota de RAM de video que tiene, la velocidad que puede usar y cualquier opción disponible para el driver asociado con el chipset utilizable. En la mayoría de los casos, no se necesita invocar éstos parametros; ya que el servidor debe detectarlos.



Section "Device"

| | |
|------------|-------------------|
| Identifier | "Failsafe Device" |
| Boardname | "vesa" |
| Busid | "PCI:1:0:0" |
| Driver | "vesa" |
| Screen | 0 |

EndSection

Si por alguna razón el servidor no puede detectarlo correctamente, se pueden ingresar los parámetros correctos en esta sección. También, se debe revisar la documentación del Xorg.

Sección "Screen"

Unifica toda la información necesaria desde las otras secciones. Se puede tener más de una sección de Device o Monitor en el archivo, pero solo los listados en la sección Screen serán los utilizados, esta es la razón por la que cada sección incluye un identificador. De igual manera la sección screen especifica cuál módulo usar, la resolución y la intensidad del color.



Section "Screen"

| | |
|--------------|--------------------|
| Identifier | "Default Screen" |
| Device | "Failsafe Device" |
| Monitor | "Failsafe Monitor" |
| Defaultdepth | 24 |



SubSection "Display"

```

Depth      24
Virtual 1024    768
Modes       "800x600@72"    "640x480@72"
"800x600@75" "640x480@60"  "800x600@60"  "832x624@75"
"1024x768@60" "1024x768@43"
EndSubSection
EndSection

```

Sección "Input Device"

Permite definir el protocolo que se va a usar para comunicarse con el ratón. Los protocolos del ratón incluyen PS/2, IMPS/2, Microsoft, y Logitech. Para todo lo que va desde el puerto PS/2, se usa /dev/psaux como el dispositivo. Para los ratones seriales, /dev/ttySO para el COM1, /dev/ttySI para el COM2, y así sucesivamente. Muchas distribuciones permiten usar /dev/ratón sin importar qué tipo de ratón se esté usando. En la sección Pointer, se puede especificar algunas opciones, como lo es emular el botón del medio haciendo uso del izquierdo y el derecho simultáneamente.



Section "InputDevice"

```

Identifier    "Configured Ratón"
Driver        "ratón"
Option        "CorePointer"
Option        "Device"          "/dev/input/mice"
Option        "Protocol"        "ImPS/2"
Option        "ZAxisMapping"    "4 5"
Option        "Emulate3Buttons" "true"
EndSection

```



Sección "Files"

Informa al servidor de X dónde encontrar módulos de servidor, la base de datos de color RGB y archivos de tipografías. Esta opción es para usuarios avanzados. En la gran mayoría de los casos, se debería dejar activada.



Tema 6: Sesiones

Una sesión es la duración de una conexión empleando una capa de sesión de un protocolo de red, o la duración de una conexión entre un usuario y un servidor, generalmente involucrando el intercambio de múltiples paquetes de datos entre la computadora del usuario y el servidor. Típicamente es el tiempo que transcurre entre que un usuario se identifica en un sistema y, bien por falta de actividad, bien por desconexión voluntaria, el sistema deja de recordarle, la sesión le permite a un usuario, por ejemplo, estar conectado a los foros durante un tiempo determinado sin tener que volver a identificarse. A continuación se detalla los tipos de sesiones:

Inicio de Sesiones desde Terminales

El inicio de sesiones desde terminales (a través de líneas serie) y la consola (cuando no se está ejecutando X-Windows) es suministrado por el programa `getty`. `init` inicia una instancia independiente de `getty` por cada terminal en el que está permitido iniciar sesiones.

`Getty` lee el nombre de usuario y ejecuta el programa `login`, el cual se encarga de leer la `password`. Si el nombre de usuario y la `password` son correctas, `login` ejecuta el intérprete de comandos. Al finalizar el intérprete de comandos (en el caso en que, por ejemplo, el usuario finaliza su sesión; o cuando `login` finaliza debido a que no concuerdan el nombre de usuario y la `password`), `init` se entera de este suceso e inicia una nueva instancia de `getty`. El núcleo no tiene noción sobre los inicios de sesiones, esto es gestionado totalmente por los programas del sistema.



Inicio de sesiones a través de la red

Es tipo de sesión funciona de un modo un poco diferente al inicio de sesiones normales. Existe una línea serie física separada para cada terminal a través de la cual es posible iniciar sesión. Por cada persona iniciando una sesión a través de la red existe una conexión de red virtual, y puede haber cualquier número (no hay límite). Por lo tanto, no es posible ejecutar getty por separado por cada conexión virtual posible. Existen también varias maneras diferentes de iniciar una sesión a través de la red, las principales en redes TCP/IP son ssh, telnet y rlogin.

Los inicios de sesión a través de la red tienen, en lugar de una cantidad enorme de getty's, un servicio individual por tipo de inicio de sesión (telnet y rlogin tienen servicios separados) que escucha todos los intentos de inicio de sesión entrantes. Cuando el servicio advierte un intento de inicio de sesión, inicia una nueva instancia de si mismo para atender la petición individual; la instancia original continúa atenta a otros posibles intentos. La nueva instancia trabaja de manera similar a getty.



Anexos:

Ejercicio Propuesto N#1:

Su unidad productiva es contratada para realizar una implementación sencilla de un servidor de archivos basado en Samba en el Ministerio del Poder Popular para la Agricultura y Tierras, con las siguientes premisas:

- La red del ministerio está comprendida por la red privada de rango 172.31.0.0/23
- El servidor donde se implementará Samba tiene por nombre: SRVDOMINIO y tiene como dirección IP: 172.31.1.5
- Debe ser un controlador de dominio y autenticar a los usuarios que se conecten al mismo.
- El grupo de trabajo de los equipos que estarán haciendo uso del servidor es: MAT.GOB.VE
- El servidor tiene instalado previamente los servicios de sincronización de hora y debe proveerlos para el ambiente mixto de estaciones de trabajo que estarán conectándose al mismo.
- El servidor compartirá todas las impresoras que tiene conectadas a través del servidor de impresión CUPS.
- El servidor solo compartirá un directorio personal correspondiente a cada usuario.



Solución del ejercicio



Instalación de samba

```
aptitude install samba samba-common winbind
```

Editaremos el archivo /etc/samba/smb.conf

```
editor /etc/samba/smb.conf
```

Definiremos los parámetros globales:

```
[global]
```

Grupo de trabajo

```
workgroup = MAT.GOB.VE
```

Nombre del equipo NetBIOS

```
netbios name = SRVDOMINIO
```

Comentario que se mostrará a los cliente sobre el servidor

```
server string = Servidor "%h" - Controlador de  
Dominio Principal (Implementado con Samba %v)
```

Si este servidor deberá actuar como intermediario para resolución de nombres

```
dns proxy = no
```

Soporte para servicios de Nombre Windows

```
wins support = yes
```

Ubicación del servidor de Nombre Windows



```
wins server = localhost
# Orden de preferencia de la resolución de nombres
name resolve order = wins host bcast
# Interfaces donde escuchará el servicio de archivos
interfaces = eth0 172.31.1.5/23 127.0.0.0/8
# Redes permitidas para este controlador de dominio
hosts allow = 172.31.0.0/255.255.254.0 127.0.0.0/8
# Forzar a que solo se utilicen las interfaces antes
definidas
bind interfaces only = yes
# Archivo de registro de eventos del servicio
log file = /var/log/samba/log.%m
# Tamaño máximo del archivo de registro (en líneas)
max log size = 1000
# Utilización exclusiva de los servicios propios de
registro de eventos del Sistema Operativo
syslog only = no
# Servicio (número) del registro de eventos del sistema a
contactar para el registro de eventos propio del servicio
Samba
syslog = 2
# Nivel de registro (0: ninguno, 1-5: reportar fallos de
baja a mediana intensidad)
log level = 5
# Modelo de autenticación (user: verificar usuario contra
base de datos de usuarios local, domain: verificar
usuarios contra un repositorio de autenticación externo,
p. ej. OpenLDAP)
```



```
security = user
# Si deben encriptarse las contraseñas
encrypt passwords = true
# Si debe aceptarse autenticación de usuarios
server signing = auto
# Permitir el cambio de contraseñas de los usuarios
mediante la librería PAM
pam password change = yes
# Opciones de red (tamaño del paquete y búfer de recepción
de datos)
socket options = TCP_NODELAY SO_RCVBUF=8192
SO_SNDBUF=8192
# Permitir la validación de usuarios como controlador de
dominio
domain logons = yes
# Si es el examinador principal para la red antes
definida.
domain master = yes
# Si es el examinador principal preferido para la red
antes definida.
preferred master = yes
# Relevancia de este servidor con respecto a otros
servidores Samba o Windows/SMB dentro de la Red (254:
Nivel máximo)
os level = 254
# Ruta de login
logon path =
# Script de inicio de sesión
```



```
logon script = scripts\DomainUsers.cmd
# Letra de unidad principal (solo válido para clientes
windows)
logon drive = F:
# ¿Será este servidor un servidor de hora también?
time server = yes
# Cuenta de invitado del dominio
guest account = guest
# Recursos compartidos de impresión, Cargar impresoras
load printers = yes
# Servicio utilizado para manejar la impresión
printing = cups
# Servicio de cola de impresión
printcap name = cups
# Recurso principal de impresoras compartidas, por defecto
compartirá todas las impresoras presentes en el sistema
[printers]
# Comentario del recurso
comment = Todas las impresoras
# Ruta a donde enviará la cola
path = /var/spool/samba
# ¿Es posible imprimir a este recurso?
printable = Yes
# ¿Se permite el acceso al invitado del dominio?
guest ok = Yes
# ¿Es el recurso de solo lectura?
read only = No
# ¿Los clientes del servidor podrán navegar este recurso?
```



```
        browseable = Yes
# Máscara de creación de los archivos
        create mask = 0700
# Recurso compartido oculto para conexión a impresoras y
distribución de controladores
        [print$]
# Comentario
        comment = Controladores de impresoras
# Ruta donde estarán los archivos de controladores
        path = /var/lib/samba/printers
        browseable = yes
        read only = yes
        guest ok = no
# Usuarios a los que se permite escribir a este recurso
(debe estar definido "read only = yes")
        write list = root smbadmin @"Administradores de
Impresion" @"Domain Admins"
# Definición del recurso compartido asociado a las
carpetas personales de cada usuario
        [homes]
        comment = Datos personales del usuario %U
        path = /home/%U
        browseable = no
        read only = no
        create mask = 0700
# Máscara de directorio para el recurso compartido
        directory mask = 0700
# Usuarios a quienes se les permite acceder a esta carpeta
```



(%S: El mismo usuario que se ha conectado al recurso)

valid users = %S

*# Definicion del recurso compartido especial "netlogon";
empleado por los clientes Windows durante el inicio de#
sesion en el dominio (este recurso normalmente se incluye
de forma predefinida para facilitar la interoperabilidad
entre sistemas operativos)*

[netlogon]

comment = Network Logon Service

path = /var/lib/samba/netlogon

guest ok = Yes

read only = no

*# Forzar a los modos de acceso por usuario de recursos
compartidos*

share modes = no

browseable = no

*# Luego de de haber definido nuestros parámetros de
/etc/samba/smb.conf procedemos a reiniciar el servicio.
Debemos recordar que los usuarios que validemos en el
dominio deberán haber sido previamente creados en el
sistema.*

#invoke-rc.d samba restart

*# Ya que también hemos definido a nuestro servidor como
servidor WINS es necesario reiniciar el servicio winbind*



#invoke-rc.d winbind restart

Con esto, finaliza nuestra implementación de los requerimientos del servicio Samba para el MAT. Podemos agregar máquinas y usuarios al dominio navegando al mismo y conectándonos con nuestro usuario y contraseña al mismo. (Recuerde que para agregar máquinas al dominio requeriremos del usuario administrador o "root" como cuenta con permisos para añadir cuentas de máquina)



Ejercicio Propuesto N#2:

Integre un servidor samba similar al que se instaló previamente en el ejercicio anterior a un directorio LDAP. Recuerde que es necesaria la configuración de smbldap para que sirva como ayudante en la autenticación y creación de usuarios en el directorio.



Referencias

- Free Software Foundation, Inc.(2009), “**sistema operativo GNU/Linux**”. Disponible en: <http://www.gnu.org/home.es.html>
- OpenSSH (2009), “OpenSSH 5.3/5.3p”. Disponible en: <Http://www.openssh.org>
- Samba (2009), “Opening Windows to a Wider World”. Disponible: <http://www.samba.org/>
- Udev (2004), “Udev provides a dynamic /dev directory”. Disponible en: <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>
- Unix-ag (2009), “Apt-Cacher NG”. Disponible en: <http://www.unix-ag.uni-kl.de/~bloch/acng/>



Glosario de Términos

- **ACLs (Access Control List/Lista de Control de Acceso):** es una tabla que le dice a un sistema los derechos de acceso que cada usuario posee para un objeto determinado, como directorios, ficheros, puertos, etc. Técnicas para limitar el acceso a los recursos según la información de autenticidad y las normas de acceso.
- **ACL estándar:** donde solo se especifica una dirección de origen.
- **ACL extendida:** donde cuya sintaxis aparece el protocolo y una dirección de origen y de destino.
- **Apache:** es un software libre servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras. Presenta, entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Ha sido desde Abril de 1996 el servidor HTTP más usado.
- **APT (Advanced Packaging Tool):** es un sistema de gestión de paquetes creado por el proyecto Debian. APT simplifica en gran medida la instalación y eliminación de programas en los sistemas GNU/Linux; no existe un programa apt en sí mismo, sino que APT es una biblioteca de funciones C++ que se emplea por varios programas de línea de comandos para distribuir paquetes. En especial, apt-get y apt-cache.
- **ARPANET: (Advanced Research Projects Agency Network/Red de la Agencia de Proyectos de Investigación Avanzada):** es una red militar Norteamericana a través de líneas telefónicas, de la que posteriormente derivó Internet.
- **Authentication Header (AH):** protege la carga útil IP y todos los campos de la cabecera de un datagrama IP excepto los campos mutantes, es decir, aquellos que pueden ser alterados en el tránsito, además de resguardar opcionalmente contra



ataques de repetición utilizando la técnica de ventana deslizante y descartando paquetes viejos.

- **Autenticación:** confirma el origen/destino de la información, es decir, corrobora que los interlocutores son quienes dicen ser.
- **Autorización:** se da normalmente en un contexto de autenticación previa. Se refiere a un mecanismo que permite que el usuario pueda acceder a servicios o realizar distintas actividades conforme a su identidad.
- **ATM (Asynchronous Transfer Mode/Modo de Transferencia Asíncrona):** es una tecnología de telecomunicación, desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.
- **Bacula:** es una colección de herramientas de respaldo muy amplia, capaces de cubrir eficientemente las necesidades de respaldo de equipos bajo redes IP. Se basa en una arquitectura cliente/servidor que resulta muy eficaz y fácil de manejar, dada la amplia gama de funciones y características que brinda; copiar y restaurar ficheros dañados o perdidos. Además, debido a su desarrollo y estructura modular, Bacula se adapta tanto al uso personal como profesional, para parques de computadores muy grandes.
- **BIND (Berkeley Internet Name Domain, anteriormente: Berkeley Internet Name Daemon):** es la implementación del estándar DNS de uso más habitual en la Internet, especialmente en los sistemas tipo Unix, en los cuales es un estándar de facto.
- **BIND9:** es una nueva versión de BIND. Fue escrita desde cero en parte para superar las dificultades subyacentes de arquitectura que estaban presentes con anterioridad, para auditar el código en las primeras versiones de BIND y también para incorporar DNSSEC. BIND 9 incluye entre otras características importantes: TSIG, notificación DNS, nsupdate, IPv6, rndc flush, vistas, procesamiento en paralelo, y una arquitectura mejorada en cuanto a portabilidad. Es comúnmente usado en sistemas GNU/Linux.



- **C++:** es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.
- **Cache:** es todo duplicado de una información original que se almacena en un lugar de acceso más rápido que el original.
- **Canaima:** es una distribución GNU/Linux Venezolana basada en Debian que surge como una solución para cubrir las necesidades ofimáticas de los usuarios finales de la Administración Pública Nacional (APN) y para dar cumplimiento al decreto presidencial Nro. 3.390 sobre el uso de Tecnologías Libres.
- **Capas de seguridad:** el uso de un enfoque por capas al planificar la estrategia de seguridad, lo que busca garantizar que el atacante que penetre en una de las capas de defensa será detenido en la capa siguiente.
- **Cliente DNS:** son programas que se ejecutan en la computadora del usuario y que generan peticiones DNS de resolución de nombres a un servidor DNS.
- **Comando "su" (Switch User/ "Super User"/ Super Usuario):** permite abrir una sesión con el ID (ID identificador) de otro usuario, o de iniciar un shell de conexión con el nuevo ID.
- **Comando sudo:** es un comando de Unix que viene de "su do", que significa en inglés "do something as the supervisor" ('hacer algo como administrador'), y permite darle acceso a un usuario a ciertos comandos de administrador, sin tener que usar la clave o acceder al sistema como root.
- **Confidencialidad:** término que hace referencia al control de la información, es decir, permite tener acceso a la información solo por parte de las personas autorizadas.



- **Control de acceso:** se refiere a un mecanismo que en función de la identificación ya autenticada, permite acceder a datos o recursos.
- **Cpio:** es el nombre de una utilidad binaria tanto como del formato asociado a ésta, .cpio . Este tipo de archivo fue inicialmente creado para el almacenamiento de copias de seguridad en cintas magnéticas de una forma contigua, y tiene un funcionamiento muy parecido al formato tar. Más específicamente, un archivo CPIO consiste en una serie de ficheros y directorios tanto como los encabezados utilizados por GNU cpio para extraer el archivo, así como encabezados extra como el nombre, fecha de creación, permisos y propietario de cada fichero y directorio. Es de notar que aunque la extensión.cpio se asocia comúnmente con este tipo de fichero de archivado, no es necesario que tenga esa extensión, pues UNIX no requiere una extensión para manejar un fichero sino que más que nada sirve para la identificación rápida de éste por parte del usuario.
- **Datagramas:** es un fragmento de paquete que es enviado con la suficiente información como para que la red pueda simplemente encaminar el fragmento hacia el equipo terminal de datos receptor, de manera independiente a los fragmentos restantes.
- **DAP (Directory Access Protocol):** es un estándar dentro de las redes de computadoras que ha sido promulgado por la ITU-T y por la ISO en 1998, para el acceso de un servicio de directorio X.500.
- **Debian Project:** es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre precompilado y empaquetado, en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos.
- **DHCP (Dynamic Host Configuration Protocol / Protocolo de Configuración Dinámica de Servidores):** permite asignar IPs de forma dinámica, e indica servidores de nombre de dominios y gateways desde un servidor a todos los clientes que se la pidan.



- **Direcciones IP:** es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI.
- **Directorio:** es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica.
- **Disponibilidad:** garantiza que la información estará disponible para las personas autorizadas cuando lo necesiten.
- **Distribución:** es una recopilación de programas y ficheros (paquetes), organizados y preparados para su instalación en las diferentes arquitecturas de hardware disponibles en el mercado, las cuales se pueden obtener a través de Internet, o adquiriendo los CD de las mismas.
- **DNS (Domain Name System):** es un sistema de nombres que permite traducir de nombres de dominio a direcciones IP y viceversa.
- **DNSO (Domain Name Supporting Organization):** es un cuerpo asesor que aconseja el ICANN el Consejo de Administración en la política referente al Sistema de Nombre de dominio.
- **DNSSEC (DNS Security Extensions):** es una suite de datos específicos IETF para asegurar ciertas clases de información proporcionada por el Sistema de Nombre de Dominio (DNS), como cuando es usado sobre el Protocolo de Internet (IP). Es un juego de extensiones a DNS que proveen a clientes DNS.
- **Dominio:** nombre básico de un conjunto de dispositivos y computadores dentro de una red, los equipos o dispositivos que lo componen cada uno posee un nombre perteneciente a ese dominio, que lo hace más fácil de recordar en vez de utilizar direcciones numéricas para acceder a los mismos.
- **DoS (Denial Of Service / Denegación de Servicio):** se refiere al incidente en el cual un usuario o una organización se ven privados de un recurso que normalmente podrían usar. Habitualmente, la pérdida del servicio supone la

indisponibilidad de un determinado servicio de red, como el correo electrónico, o la pérdida temporal de toda la conectividad y todos los servicios de red.

- **Dovecot:** es un servidor de IMAP y POP3 de código abierto para sistemas GNU/Linux / UNIX-like, escrito fundamentalmente pensando en seguridad. Fue desarrollado por Timo Sirainen, y apunta principalmente a ser un servidor de correo de código abierto ligero, rápido, fácil de instalar y por sobre todo seguro.
- **ESMTP (Enhanced Simple Mail Transfer Protocol):** es una definición de extensiones de protocolo para el estándar SMTP, cuyo formato de extensión fue definido en el RFC 1869 en 1995. Este RFC estableció una estructura para todas las extensiones existentes y futuras con el fin de producir una manera consistente y manejable por la cual los clientes y servidores SMTP puedan ser identificados y los servidores SMTP puedan señalar las extensiones soportadas a los clientes conectados.
- **ESP (Encapsulating Security Payload):** este protocolo proporciona autenticidad de origen, integridad y protección de confidencialidad de un paquete. ESP también soporta configuraciones de solo cifrado y solo autenticación.
- **Estado de inseguridad activo:** se refiere a la falta de conocimiento del usuario acerca de las funciones del sistema, algunas de las cuales pueden ser dañinas para el sistema (por ejemplo, no desactivar los servicios de red que el usuario no necesita).
- **Estado de inseguridad pasivo:** se refiere a la falta de conocimiento de las medidas de seguridad disponibles (por ejemplo, cuando el administrador o usuario de un sistema no conocen los dispositivos de seguridad con los que cuentan).
- **Eudora:** es un cliente de correo electrónico elaborado para las plataformas MS Windows y Apple Mac OS X; así como para computadoras de mano basadas en Palm OS, con posibilidad en este último caso, de sincronizar los mensajes con una computadora de sobremesa. Pese a disponer de una cuota de mercado limitada, sus usuarios acostumbran a ser fieles y confiar en la robustez y sencillez de este

software.

- **Frame Relay:** consiste en una forma simplificada de tecnología de conmutación de paquetes que transmite una variedad de tamaños de tramas o marcos ("*frames*") para datos, perfecto para la transmisión de grandes cantidades de datos. Esta técnica se utiliza para un servicio de transmisión de voz y datos a alta velocidad que permite la interconexión de redes de área local separadas geográficamente a un coste menor.
- **FreeBSD:** es un sistema operativo libre para computadoras basado en las CPU de arquitectura Intel, incluyendo procesadores 386, 486 (versiones SX y DX), y Pentium.
- **FQDN (Fully Qualified Domain Name):** es un nombre que incluye el nombre de la computadora y el nombre del dominio asociado a ese equipo. La longitud máxima permitida para un FQDN es 255 caracteres (bytes), con una restricción adicional a 63 bytes por etiqueta dentro de un nombre de dominio. Las etiquetas FQDN se restringen a un juego de caracteres limitado: letras A-Z de ASCII, los dígitos, y el carácter «-», y no distinguen mayúsculas de minúsculas.
- **Gestión de claves:** antes de que el tráfico sea enviado/recibido, cada router/cortafuegos/servidor debe ser capaz de verificar la identidad de su interlocutor.
- **GPL (General Public License / Licencia Pública General):** la Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.
- **Header (Cabecera):** es la primera parte de un paquete de datos que contiene información sobre las características de este.
- **HDSL (High bit rate Digital Subscriber Line / Línea Digital de Abonado de alta velocidad):** es un sistema de transmisión de datos de alta velocidad que



utiliza dos pares trenzados; obteniendo velocidades superiores al Megabit en ambos sentidos.

- **Horde Webmail:** gestor de corre que permite la redacción de mensajes en html o texto plano con adjuntos, búsqueda de mensajes, etiquetado de mensajes, reglas de filtrado, cifrado PGP, organizar mensajes en carpetas o crear carpetas virtuales para la organización dinámica del correo.
- **Host:** un host o anfitrión es un computador que funciona como el punto de inicio y final de las transferencias de datos.; más comúnmente descrito como el lugar donde reside un sitio web. Un host de Internet tiene una dirección de Internet única (dirección IP) y un nombre de dominio único o nombre de host.
- **HTML (HyperText Markup Language / Lenguaje de Marcas de Hipertexto):** es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- **Icedove:** es un cliente de correo electrónico, grupos de noticias y RSS de código abierto exclusivamente destinado a distribuciones Linux basadas en Debian GNU/Linux.
- **IDLE:** hace referencia a la inactividad de un usuario en IRC.
- **IETF (Internet Engineering Task Force/Grupo de Trabajo en Ingeniería de Internet):** es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad. Fue creada en EE.UU en 1986.
- **IKE (Internet key exchange):** es un protocolo usado para establecer una asociación de seguridad (SA) en el protocolo Ipsec. Supone una alternativa al intercambio manual de claves, y su objetivo es la negociación de una Asociación de Seguridad para IPSEC, permitiendo además especificar el tiempo de vida de la sesión IPSEC, autenticación dinámica de otras máquinas, entre otras.



- **IMAP (Internet Message Access Protocol):** es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.
- **Imposibilidad de repudio:** es una forma de garantizar que el emisor de un mensaje no podrá posteriormente negar haberlo enviado, mientras que el receptor no podrá negar haberlo recibido.
- **Integridad:** término que hace referencia a que la información esté completa y sea auténtica.
- **IP (Internet Protocol):** el protocolo de comunicaciones IP permite que redes grandes y geográficamente diversas de computadoras, se comuniquen con otras rápida y económicamente a partir de una variedad de eslabones físicos.
- **IPsec (Internet Protocol security):** es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP), autenticado y/o cifrando cada paquete IP en un flujo de datos. IPsec incluye también protocolos para el establecimiento de claves de cifrado.
- **IPv4:** es la versión 4 del Protocolo IP (Internet Protocol). Esta fue la primera versión del protocolo que se implementó extensamente, y forma la base de Internet. IPv4 usa direcciones de 32 bits, limitándola a $2^{32} = 4.294.967.296$ direcciones únicas, muchas de las cuales están dedicadas a redes locales (LANs).
- **IPv6:** es una nueva versión de IP (Internet Protocol) y está destinado a sustituir a IPv4, cuyo límite en el número de direcciones de red admisibles está empezando a restringir el crecimiento de Internet y su uso; pero el nuevo estándar mejorará el servicio globalmente; por ejemplo, proporcionará a futuras celdas telefónicas y dispositivos móviles con sus direcciones propias y permanentes.
- **IPX (Internetwork Packet Exchange / Intercambio de paquetes interred):** es un protocolo de comunicaciones NetWare que se utiliza para encaminar mensajes de un nodo a otro. Los paquetes IPX incluyen direcciones de redes y

pueden enviarse de una red a otra.

- **IRC (Internet Relay Chat):** es un protocolo de comunicación en tiempo real basado en texto, que permite debates en grupo o entre dos personas. Se diferencia de la mensajería instantánea, porque los usuarios no deben acceder a establecer la comunicación de antemano; de tal forma, que todos los usuarios que se encuentran en un canal pueden comunicarse entre sí, aunque no hayan tenido ningún contacto anterior.
- **ISC BIND:** el nombre completo original del servidor BIND9 desarrollado por la Internet Systems Consortium.
- **ISP (Internet Service Provider/Proveedor de Servicios de Internet):** es una empresa dedicada a conectar a Internet a los usuarios, o las distintas redes que tengan, y a dar el mantenimiento necesario para que el acceso funcione correctamente. También ofrece servicios relacionados, como alojamiento web, registro de dominios, entre otros.
- **ITU-T (Sector de Normalización de las Telecomunicaciones de la UIT):** es el órgano permanente de la Unión Internacional de Telecomunicaciones (UIT), que estudia los aspectos técnicos, de explotación y tarifarios y publica normativa sobre los mismos, con vista a la normalización de las telecomunicaciones a nivel mundial.
- **LAN (Local Área Network):** es la interconexión de varias computadoras y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de hasta 200 metros; su aplicación más extendida es la interconexión de computadoras personales y estaciones de trabajo en oficinas, fábricas, etc., para compartir recursos e intercambiar datos y aplicaciones.
- **LDAP (Lightweight Directory Access Protocol/Protocolo Ligero de Acceso a Directorios):** es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido, para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su



sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

- **L2F (Layer 2 Forwarding):** fue diseñado por Cisco para establecer túneles de tráfico desde usuarios remotos hasta sus sedes corporativas. La principal diferencia entre PPTP y L2F es que, como el establecimiento de túneles de L2F no depende de IP, es capaz de trabajar directamente con otros medios, tales como Frame Relay o ATM. L2F utiliza el protocolo PPP para la autenticación del usuario remoto, pero también implementa otros sistemas de autenticación como TACACS+ (*Terminal Access Controller Access Control System*) y RADIUS (*Remote Authentication Dial-In User Service*).
- **LSB (Linux Standard Base o Base Estándar para Linux):** es un proyecto conjunto de varias distribuciones de Linux bajo la estructura organizativa del Free Standards Group (grupo de estándares libre) con el objeto de crear y normalizar la estructura interna de los sistemas operativos derivados de Linux.
- **L2TP (Layer 2 Tunneling Protocol):** fue creado para corregir las deficiencias entre los protocolos PPTP y L2F, para establecerse como un estándar aprobado por el IETF. L2TP utiliza PPP para proporcionar acceso telefónico, que puede ser dirigido a través de un túnel por Internet hasta un punto determinado. L2TP define su propio protocolo de establecimiento de túneles, basado en L2F.
- **LUN (Logical Unit Number):** es una dirección para una unidad de disco duro y por extensión, el disco en sí mismo. El término es originario del protocolo SCSI como una forma de diferenciar unidades de disco individuales dentro de un bus SCSI tal que un array de discos.
- **Mozilla Application Suite:** es un navegador web y una plataforma de desarrollo libre y de código abierto para la WWW.
- **NAT (Network Address Translation/Traducción de Dirección de Red):** es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes, que se asignan mutuamente direcciones incompatibles. Consiste en convertir en



tiempo real las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes, para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo.

- **NETBIOS (Network Basic Input/Output System):** es una especificación de interfaz para acceso a servicios de red, es decir, una capa de software desarrollado para enlazar un sistema operativo de red con hardware específico.
- **Newfs:** crea un sistema de archivos nuevo con nombres de archivo cortos (-S) o largos (-L).
- **Nombres de dominio:** son direcciones nemotécnicas o alias que identifican un sitio de Internet.
- **NSS (Network Switching Subsystem/Subsistema de Conmutación de Red):** es el componente que realiza las funciones de portar y administrar las comunicaciones entre teléfonos móviles y la Red de Conmutada de Telefonía Pública (PSTN) para una red GSM.
- **Nsupdate:** es una red de mantenimiento multiuso usada por administradores de redes para pedir el nombre del servidor de una zona DNS y actualizar su data. El nombre del servidor puede ser local o a un dominio o con una autenticación apropiada, y un permiso dado por dnssec un servidor de Internet.
- **OpenVPN:** ofrece conectividad punto-a-punto con validación jerárquica de usuarios y host conectados remotamente, resulta una muy buena opción en tecnologías Wi-Fi, y soporta una amplia configuración (como por ejemplo balanceo de cargas, entre otras). Está publicado bajo la licencia GPL, de software libre.
- **OSI (Open Source Initiative):** es una organización dedicada a la promoción del código abierto.
- **OSI (Open System Interconnection):** es el modelo de referencia de Interconexión de Sistemas Abiertos. Este modelo de red descriptivo fue creado por



la Organización Internacional para la Estandarización lanzado en 1984. Es decir, siendo un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

- **PAM (Pluggable Authentication Modules/Módulos enchufables de autenticación):** es un sistema de autenticación que controla el acceso a RHL.
- **PHP:** es un lenguaje de programación interpretado, y diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.
- **Pipe:** tubería o flujo de datos entre programas, se denota normalmente con el carácter "|" y se utiliza para enviar la salida de un comando a uno siguiente que se ejecutará procesando o trabajando sobre la salida del anterior.
- **Protocolos Criptográficos:** también llamado protocolo criptográfico o protocolo de cifrado, es un protocolo abstracto o concreto que realiza funciones relacionadas con la seguridad, aplicando métodos criptográficos.
- **POP3 (Post-Office Protocol/Protocolo de Oficina de Correos):** es un protocolo que permite recuperar el correo desde una máquina distinta a la que lo recibe y que es el más utilizado en las conexiones habituales por módem o RDSI a un proveedor (ya sea mediante PPP o SLIP).
- **PPP (Point-to-point Protocol/Protocolo Punto a Punto):** permite establecer una comunicación a nivel de enlace entre dos computadoras. Generalmente, se utiliza para establecer la conexión a Internet de un particular con su proveedor de acceso a través de un módem telefónico. Ocasionalmente también es utilizado sobre conexiones de banda ancha
- **PPTP (Point to Point Tunneling Protocol):** es un protocolo de red creado por



Microsoft que permite la realización de transferencias seguras desde clientes remotos a servidores emplazados en redes privadas, empleando para ello tanto líneas telefónicas conmutadas como Internet.

- **Proxy:** hace referencia a un programa o dispositivo que realiza una acción en representación de otro. Su finalidad más habitual es la de servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando solo se puede disponer de un único equipo conectado, esto es, una única dirección IP.
- **RADIUS (Remote Authentication Dial-In User Service):** es un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP.
- **Redes gigabit:** es una ampliación del estándar Ethenet (concretamente la versión 802.3ab y 802.3z del IEEE) que consigue una capacidad de transmisión de 1 gigabit por segundo, correspondientes a unos 1000 megabits por segundo de rendimiento contra unos 100 de Fast Ethenet (También llamado 100-Base/T).
- **Repositorio de datos:** es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en el tiempo.
- **Respaldos diferenciales:** son similares a los respaldos incrementales en que ambos solamente copian archivos que han sido modificados. Sin embargo, los respaldos diferenciales son *acumulativos* — en otras palabras, con un respaldo diferencial, una vez que un archivo ha sido modificado continua siendo incluido en todos los respaldos diferenciales subsecuentes (hasta el próximo respaldo completo).
- **Respaldos incrementales:** son utilizados en conjunto con respaldos regulares completos (por ejemplo, un respaldo semanal completo, con respaldos incrementales diarios). Los respaldos incrementales primero revisan para ver si la fecha de modificación de un archivo es más reciente que la fecha de su último



respaldo. Si no lo es, significa que el archivo no ha sido modificado desde su último respaldo y por tanto se puede saltar esta vez. Por otro lado, si la fecha de modificación es más reciente, el archivo ha sido modificado y se debería copiar.

- **RFC (Request For Comments/Petición de Comentarios):** son una serie de notas sobre Internet que comenzaron a publicarse en 1969. Cada una de ellas individualmente es un documento cuyo contenido es una propuesta oficial para un nuevo protocolo de la red Internet (originalmente de ARPANET), que se explica con todo detalle para que en caso de ser aceptado pueda ser implementado sin ambigüedades.
- **RSA:** el sistema criptográfico con clave pública RSA es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye (en forma autenticada preferentemente), y otra privada, la cual es guardada en secreto por su propietario.
- **RSH:** protocolo que permite que un usuario ejecute instrucciones en un sistema remoto sin tener que conectarse al sistema.
- **RST (Flag):** es un bit que se encuentra en el campo del código en el protocolo TCP, y se utiliza para reiniciar la conexión. Un ejemplo práctico de utilización es el que realiza un servidor cuando le llega un paquete a un puerto no válido: este responde con el RST activado.
- **Samba:** es una implementación libre del protocolo de archivos compartidos de Microsoft Windows, para sistemas de tipo UNIX. De esta forma, es posible que computadores con GNU/Linux, Mac OS X o Unix en general se vean como servidores o actúen como clientes en redes de Windows. Samba también permite validar usuarios haciendo de Controlador Principal de Dominio (PDC), como miembro de dominio e incluso como un dominio Active Directory para redes basadas en Windows; aparte de ser capaz de servir colas de impresión, directorios compartidos y autenticar con su propio archivo de usuarios.



- **SAN (Storage Area Network):** es una red específica dedicada a la tarea de transporte de datos para almacenamiento y recuperación. SAN arquitecturas son alternativas al almacenamiento de datos en los discos directamente a los servidores o de almacenamiento de datos en Network Attached Storage (NAS) de dispositivos que están conectados a través de redes de propósito general.
- **SASL (Simple Authentication and Security Layer/Capa de Seguridad y Autenticación Simple):** es un framework para autenticación y autorización en protocolos de internet. Separa los mecanismos de autenticación de los protocolos de la aplicación permitiendo, en teoría, a cualquier protocolo de aplicación que use SASL usar cualquier mecanismo de autenticación soportado por SASL.
- **Seguridad física:** consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e informaciones confidenciales.³⁹
- **Seguridad de la información:** es una disciplina integral que nos permite, principalmente, resguardar los datos e información de agentes externos no autorizados.
- **Seguridad lógica:** consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos, permitiendo que solo las personas autorizadas accedan a ellos.⁴⁰
- **Servidor DNS:** sistema que, al recibir una petición para resolver un nombre o una dirección, consulta su base de datos interna de equipos conocidos u otro sistema DNS para entregar bien sea la dirección IP de un dominio, o, si se le provee la dirección IP conocida, este devuelve el nombre de equipo y dominio al que pertenece.

39 HUERTA, Antonio Villalón. "Seguridad en Unix y Redes". Versión 1.2 Digital - Open Publication License v.10 o Later. 2 de Octubre de 2000. <http://www.kriptopolis.org>

40 <http://www.segu-info.com.ar/logica/seguridadlogica.htm>



- **Servidores DNS autoritarios:** hacen visibles y accesibles los servicios de red para los usuarios.
- **SGML (Standard Generalized Markup Language / Estándar de Lenguaje de Marcado Generalizado):** consiste en un sistema para la organización y etiquetado de documentos. El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.
- **Shell:** intérprete de comandos, el intérprete de comandos es un programa a través del cual un usuario ejecuta instrucciones sobre datos y funciones que residen en el sistema operativo, estas operaciones pueden ir desde mover o copiar archivos, hasta cambios en la política de prioridad de los procesos que se ejecutan en el sistema operativo, esta enorme flexibilidad es característica de los sistemas operativos basados en UNIX, como GNU/Linux. También es llamado consola o "concha".
- **Sistema Operativo:** es un software que administra y controla las actividades, y recursos de la computadora. Comprende todos aquellos paquetes que le permiten al computador funcionar como un conjunto de herramientas e intérpretes de comandos.
- **SMTP (Simple Mail Transfer Protocol/Protocolo Simple de Transferencia de Correo):** es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, entre otros.). Está definido en el RFC 2821 y es un estándar oficial de Internet.
- **SSL (Secure Sockets Layer/Protocolo de Capa de Conexión Segura):** proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptográficos.
- **SSL/TLS (Secure Sockets Layer/Protocolo de Capa de Conexión Segura y Transport Layer Security/Seguridad de la Capa de Transporte):** son



protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

- **Subdominio:** es un subgrupo o subclasificación del nombre de dominio, el cual es definido con fines administrativos u organizativos, que podría considerarse como un dominio de segundo nivel. Normalmente es una serie de caracteres o palabra que se escriben antes del dominio. En Internet se podría decir que el subdominio se utiliza para referirse a una dirección web, que trabaja como un anexo (o sitio relacionado) de un dominio principal.
- **SOA (Service Oriented Architecture / Arquitectura Orientada a Servicios):** es un software que administra y controla las actividades, y recursos de la computadora. Comprende todos aquellos paquetes que le permiten al computador funcionar como un conjunto de herramientas e intérpretes de comandos.
- **SOAP (Simple Object Access Protocol):** es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- **SOAP Header (Cabeceras SOAP):** es una clase especial de bajo nivel para pasar o devolver cabeceras SOAP. Es simplemente un contenedor de datos y no tiene métodos especiales aparte de su constructor.
- **SYN:** es un byte de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales ISN de una conexión en el procedimiento de establecimiento de tres fases (*3 way handshake*).
- **SYN/ACK (Acknowledge/ Reconocido):** se usa para sincronizar los números de secuencia en tres tipos de segmentos: petición de conexión, confirmación de conexión (con ACK activo) y la recepción de la confirmación (con ACK activo).
- **TACACS (Terminal Access Controller Access Control System/Sistema de Control de Acceso Mediante Control del Acceso desde Terminales):** es un protocolo de autenticación remota que se usa para comunicarse con un servidor de autenticación comúnmente usado en redes Unix. TACACS permite a un servidor de



acceso remoto comunicarse con un servidor de autenticación para determinar si el usuario tiene acceso a la red.

- **Tar:** se refiere a un formato de archivos ampliamente usado en entornos UNIX, identificados con la extensión tar. Además hace referencia al programa para la manipulación de archivos que es estándar en estos entornos. El formato fue diseñado para almacenar archivos de una forma conveniente en cintas magnéticas y de allí proviene su nombre, que proviene de "Tape ARchiver" (en inglés: archivador en cinta). Debido a este origen el formato está preparado para ser procesado linealmente, no contando con manera de extraer un miembro sin recorrer todo el archivo hasta encontrarlo.
- **TCP/IP (Transfer Control Protocol / Internet Protocol):** conjunto de protocolos definidos por catedráticos en el proyecto ARPANet del Departamento de Defensa de Estados Unidos, para la red universitaria Internet en los años setenta. Esta familia de protocolos es un estándar para el intercambio de comunicaciones entre computadores.
- **Telnet (TELecommunication NETwork):** es el nombre de un protocolo de red que sirve para acceder mediante una red a otra máquina, para manejarla remotamente como si estuviéramos sentados delante de ella.
- **TLD (Top Level Domain):** son los nombres en lo alto de la jerarquía de los DNS. Aparecen en los nombres de dominio, como "net" en "www.example.net". Los administradores del "dominio de la raíz" o "zona de la raíz" ("root domain" or "root zone") controlan lo que los TLDs sean reconocidos por los DNS. Los TLDs comúnmente usados incluyen a .com, .net, .edu, .jp, .de, etc.
- **TSIG (Transaction SIGnature):** usa llaves o claves secretas compartidas y germinador de una sola vía para proveer un significado seguro criptografado, para identificar cada punto final de una conexión así como el estar permitido a hacer o responder a la actualización DNS.
- **TTL (Time To Live):** es el tiempo que un paquete permanece activo en una red.



Hay un número TTL en cada header de paquete IP, y a medida que un paquete pasa por cada router o enrutador, lo reduce por 1 este número. Si el paquete llega a 0, los routers o enrutadores no seguirán reenviando el paquete.

- **Tunneling:** técnica que consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel dentro de una red de computadoras.
- **UDP (User Datagram Protocol):** es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red, sin que se haya establecido previamente una conexión; ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.
- **UNIX:** es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.
- **VPN (Virtual Private Networks/Red Privada Virtual):** es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.
- **WAN (Wide Area Networks/Red de Área Amplia):** es un tipo de red de computadoras capaz de cubrir distancias desde unos 100km hasta unos 1000 km, dando el servicio a un país o un continente.
- **XML (Extensible Markup Language / Lenguaje de Marcas Ampliable):** es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Consiste en una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.
- **X.500:** es un conjunto de estándares de redes de computadoras de la ITU-T sobre servicios de directorio, entendidos estos como bases de datos de direcciones electrónicas (o de otros tipos). Los protocolos definidos por X.500 incluyen,



protocolo de acceso al directorio (DAP), el protocolo de sistema de directorio, el protocolo de ocultación de información de directorio, y el protocolo de gestión de enlaces operativos de directorio.

- **Zonas DNS:** permiten al servidor maestro o primario cargar la información de una zona.