

Manejo de Sistemas Operativos

Laboratorio I

Versión 1.0

13 de marzo de 2015

Diego Saravia

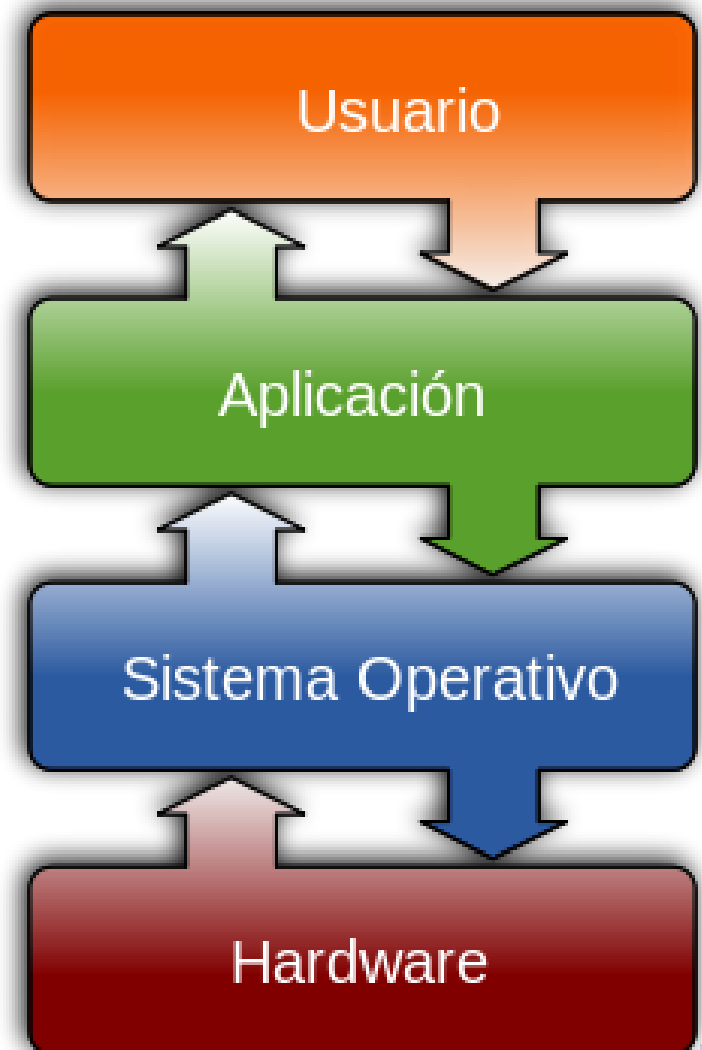
dsa@unsa.edu.ar

diego.saravia@gmail.com

Departamento de Física. Facultad de Ciencias Exactas.

Universidad Nacional de Salta (UNSa).

INENCO (UNSa – CONICET).



2.- Notas generales sobre el texto.

- ▶ Esto es un BORRADOR, ojo con los errores.
- ▶ Se ha puesto la palabra OPCIONAL, en las láminas cuyo contenido no será evaluado: en general demostraciones matemáticas o información complementaria. La palabra (no) en una lámina indicada como demostración, nos dice que se omitió la demostración, pero que contiene afirmaciones demostrables.
- ▶ Los tipos de láminas: nota e información, si aparecen, son equivalentes.
- ▶ La versión que no indica número es la inicial, correspondería 0.1.
- ▶ Críticas, correcciones, comentarios, y aportes, bienvenidos.

3.- Bibliografía

- 📄 Guía de (auto) aprendizaje en Software Libre. Currículo para la formación profesional. Diego Saravia. 2005. <http://softwarelibre.unsa.edu.ar/dsatex/cursos.dir/index.html>
- 📄 Curso de Introducción a GNU/Linux Historia, Filosofía, Instalación y Conceptos Básicos Sebastián D. Criado, Emiliano Gavilán
http://www.lugro.org.ar/sites/default/files/cursos/curso_intro/book1.html
<http://www.lugro.org.ar/sites/default/files/cursos/edt1.html>
- 📄 Sistema Operativo, Wikipedia.
http://es.wikipedia.org/wiki/Sistema_operativo
- 📄 GNU/Linux, Wikipedia. <http://es.wikipedia.org/?title=GNU/Linux>
- 📄 Windows, Wikipedia.
http://es.wikipedia.org/wiki/Microsoft_Windows
- 📄 Proyecto GNU. Wkipedia, [http://es.wikipedia.org/wiki/GNU Sitio](http://es.wikipedia.org/wiki/GNU_Sitio),
<https://www.gnu.org/home.es.html>

4.- Bibliografía

- 📄 Qué son y cómo se crean las particiones en Linux. Gabriela. <http://hipertextual.com/archivo/2013/10/particiones-en-linux/>
- 📄 Tanenbaum, A. (1992) Modern Operating Systems, Englewood Cliffs: Prentice-Hall
- 📄 Sistema de Archivos. Wikipedia.
http://es.wikipedia.org/wiki/Sistema_de_archivos
- 📄 El rol de las señales de interrupción. Linux Internals - Kernel y Arquitectura. Luciano Rojo. 13. <http://kernelinternals.blogspot.com.ar/2013/10/el-rol-de-las-senales-de-interrupcion.html>
- 📄 Organización del Computador. Máquina de Von Neumann [//dc.uba.ar/materias/oc1/2012/c1/descargas/C02-VonNeumannYGeneral.pdf](http://dc.uba.ar/materias/oc1/2012/c1/descargas/C02-VonNeumannYGeneral.pdf)

5.- Lista de páginas

- 1.- Título
- 2.- Notas generales sobre el texto.
- 3.- Bibliografía
- 4.- Bibliografía
- 5.- Lista de páginas
- 8.- CAPÍTULO 1: Objetivos**
- 9.- CAPÍTULO 2: Hardware, modelo de Von Neumann**
- 10.- Nota a1: hardware, modelo de Von Neumann
- 11.- Nota a2: hardware, Modelo de Von Neumann
- 12.- Nota a3: procesos e interrupciones
- 13.- Nota a3: interrupciones de hardware
- 14.- Nota b: dispositivos
- 15.- Nota c: arquitecturas de hardware
- 16.- CAPÍTULO 3: Software, Sistemas Operativos**
- 17.- Información k1: GNU/Linux
- 18.- Información k1: GNU/Linux
- 19.- Información k1: GNU/Linux
- 20.- Información k2: GNU/Linux

21.- Información k3: GNU/Linux

22.- Información l: Windows

23.- Sección a: tareas de un SO

24.- Información m1: gestión de procesos

25.- Información m1: gestión de procesos

26.- Información m2: gestión de recursos

27.- Información m3: gestión de usuarios

28.- Información m4: gestión de usuarios

29.- Información n: gestión de la memoria

30.- Información ñ1: gestión de dispositivos

31.- Información ñ2: gestión de dispositivos

32.- Información o: gestión de módulos

33.- Información p: gestión de archivos

34.- CAPÍTULO 4: Archivos: la gran abstracción

35.- Información q: archivos: propiamente dichos, procesos y dispositivos

36.- Información r: archivos: propiamente dichos, procesos y dispositivos

37.- Información r: archivos y directorios

38.- Información r: archivos de dispositivos

39.- Información s: rutas y nombre de archivos

40.- Información t: ejemplo de 'ruta' en GNU/Linux

41.- Información t: ejemplo de 'ruta' en un sistema Windows

42.- Información t: extensiones

43.- Información u: tipo de sistemas de archivos

44.- Información u: capacidades de los sistemas de archivos

45.- CAPÍTULO 5: El árbol estándar

46.- Nota d: el árbol genérico

47.- Nota e1: el árbol estándar

48.- Nota e2: el árbol estándar

49.- Nota e3: el árbol estándar

50.- Nota e4: el árbol estándar

51.- Nota e5: el árbol estándar

52.- Nota e6: el árbol estándar

53.- Nota e7: el árbol estándar

54.- Nota e8: el árbol estándar

55.- CAPÍTULO 6: Comandos y sus Intérpretes

56.- Nota f1: argumentos

57.- Nota f2: comandos compuestos

58.- Nota f3: ¿que es el shell?

59.- Nota g: ejemplo, comando date

60.- Nota g4: algunos Comandos

61.- Nota g5: redirección

62.- CAPÍTULO 7: Sistema gráfico X - escritorio, gestión de ventanas

63.- Nota h1: puntos básicos

64.- Nota h2: cliente – servidor, eventos

65.- Nota h1: entornos gráficos de escritorio

66.- Nota h2: KDE

67.- Nota h3: Gnome

68.- Nota h4: interfaz gráfica de usuario (Graphical shells), Unity

69.- Nota h5: Windows

70.- CAPÍTULO 8: Instalación de aplicaciones

71.- CAPÍTULO 9: Instalación

72.- Nota i: particionado

73.- CAPÍTULO 10: Práctico

8.- CAPÍTULO 1: Objetivos

Entender los aspectos esenciales involucrados en:

- ▶ el hardware, los sistemas operativos, y las aplicaciones,
- ▶ el sistema de archivos y la forma de denominarlos,
- ▶ la terminal de comandos,
- ▶ la interfaz gráfica,
- ▶ la instalación de sistemas operativos,
- ▶ la instalación de aplicaciones propias y de terceros en sistemas operativos;

a los efectos de poder manejar y manejarse en una computadora, tanto como usuario o como desarrollador de aplicaciones;

e ir realizando tareas concretas a medida que avanza la clase.

Duración de la actividad: 2 horas. Clase anterior: armado de computadoras.

Próxima: Aplicaciones de “oficina”.

9.- CAPÍTULO 2: Hardware, modelo de Von Neumann

Hardware: máquina física que ejecuta software.

Partes:

cpu “Unidad Central de Proceso”: elementos para procesar datos.

controlador de interrupciones: Combina varias fuentes de interrupciones sobre una o más líneas del CPU, mientras permite que los niveles de prioridad sean asignados a su/s salida/s.

memoria, (E/S): Retiene, memoriza o almacena datos informáticos durante algún intervalo de tiempo. Es una gran tabla de palabras o bytes que se referencian cada una mediante una dirección única. Tipos: RAM, ROM. La entrada–salida suele hacerse mediante dispositivos sobre posiciones de memoria.

Por el de direcciones el procesador indica a que posición se refiere.

10.- Nota a1: hardware, modelo de Von Neumann

buses: Conjuntos de cables que transportan la información. Tipos:

direcciones: transporta la dirección del dato o del periférico al que se quiere acceder,

control: selecciona la operación a realizar sobre el dato (principalmente lectura, escritura o modificación), interrupciones

datos: transporta los datos (32 bits, 64 bits).

11.- Nota a2: hardware, Modelo de Von Neumann

Modelo de von Neumann Bus del Sistema

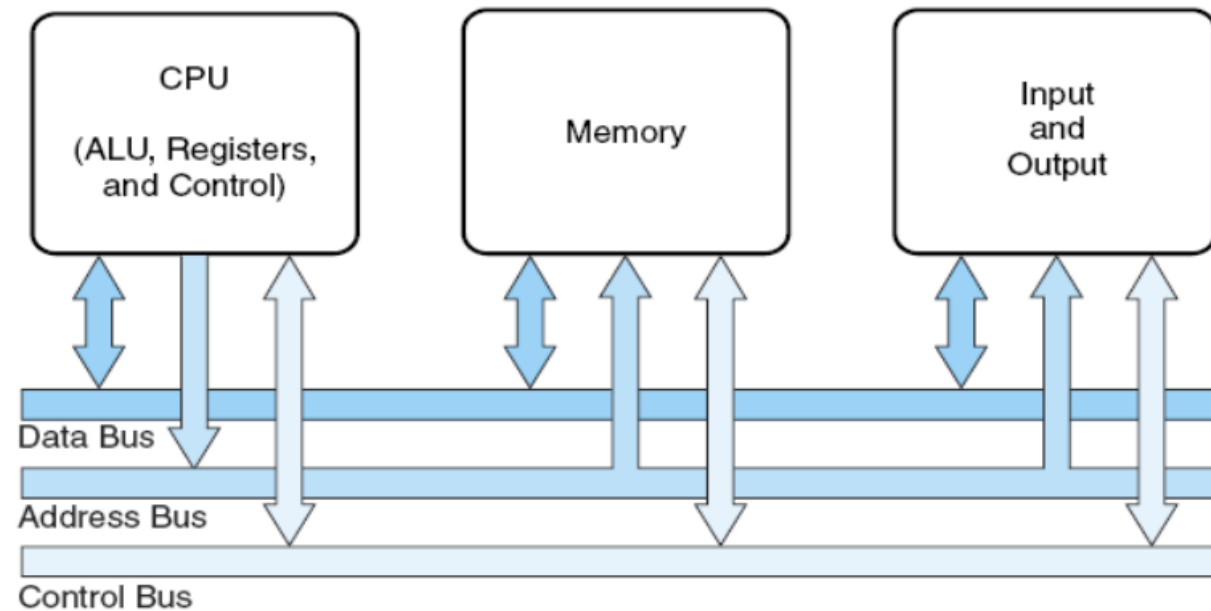


Figura 2: Buses Von Neumann

12.- Nota a3: procesos e interrupciones

La computadora se inicia leyendo una posición prefijada de memoria (ROM) y ejecutando el programa que allí se encuentra, iniciando un proceso, que puede iniciar a su vez otros procesos.

Habr  un proceso (programa, aplicaci n) ejecut ndose (o idle), que cada tanto es interrumpido por el hardware (IRQ). Por ejemplo el tocar una tecla o un tick del reloj, ante lo cual se ejecuta un software espec fico de esa interrupci n y al terminar retoma con el proceso original.

13.- Nota a3: interrupciones de hardware

Las señales de interrupción tienen como propósito proveer una manera al procesador de manejar código que esta fuera del flujo normal de ejecución. Cuando llega una señal de interrupción, el cpu tiene que parar lo que esta haciendo en ese momento y cambiar a una nueva actividad. Esto lo hace salvando el contenido de los registros y comenzando a ejecutar la interrupción.

14.- Nota b: dispositivos

Una computadora tiene periféricos, que se conectan como vimos en direcciones de memoria determinadas.

Generalmente realizan operaciones de entrada o salida, es decir para comunicarse o para almacenar información (comunicarse con el futuro y el pasado).

Ej.: impresoras, pantallas de texto, teclados, discos duros, etc.

15.- Nota c: arquitecturas de hardware

Unas 13 en el caso Debian. Nos interesan:

- ▶ i386,
- ▶ amd64,
- ▶ arm (android).

16.- CAPÍTULO 3: Software, Sistemas Operativos

Software: secuencia de instrucciones (dualismo: materia e información).

Un sistema operativo “SO” (Operating System “OS”):

- ▶ Es un conjunto de programas.
- ▶ Que gestiona los recursos de hardware y provee servicios a los demás programas (aplicaciones), en los que abstrae las especificidades del hardware.
- ▶ Se ejecuta en modo privilegiado respecto de los restantes programas (aunque puede que parte del mismo se ejecute en espacio de usuario).
- ▶ Junto con el manejo de memoria, el manejo de interrupciones es una de las tareas mas sensibles y complejas que tiene que manejar el núcleo. Las interrupciones, en este caso las asincrónicas, pueden aparecer en cualquier momento, y el kernel puede que quiera finalizar algo mas antes de atenderla. El objetivo del kernel es atenderla lo mas rápido posible sin dejar de procesar tanto como puede, es una tarea milimétrica.

17.- Información k1: GNU/Linux

En el caso del GNU/Linux:

- ▶ núcleo o kernel Linux.
- ▶ sistema GNU: por ejemplo administrar módulos de dispositivos.

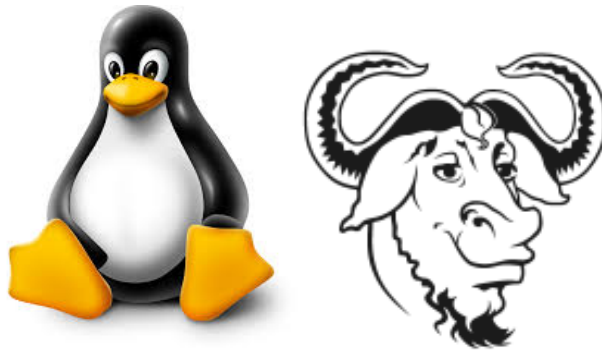


Figura 3: GNU y Linux, Daniel Colleti, Linus Torvalds, Mad Dog Hall

Las aplicaciones del SO suelen poder operar en espacio de núcleo, que no es limitado. El resto tiene asignado cierta memoria virtual en el espacio de usuario (userland) y del cual no puede salirse, por diseño del hardware.

18.- Información k1: GNU/Linux

Fuera del SO: aplicaciones o programas de sistema, que son aplicaciones de utilidad que se suministran con el SO pero no forman parte de él. Ofrecen un entorno útil para el desarrollo y ejecución de programas, siendo algunas de las tareas que realizan:

- ▶ Manipulación y modificación de archivos.
- ▶ Información del estado del sistema.
- ▶ Soporte a lenguajes de programación.
- ▶ Comunicaciones.

19.- Información k1: GNU/Linux

Distribuciones: Debian/Ubuntu, Red Hat/Fedora, OpenSUSE, de diferentes empresas y organizaciones sociales.

Fundamentalmente compiladoras y empaquetadoras de paquetes binarios como los deb y rpm.

Compatibilidad fuente (muy importante), no binaria, entre todas, y binaria en cada versión de cada distribución.

20.- Información k2: GNU/Linux



Figura 4: GNU/Linux, iconos

21.- Información k3: GNU/Linux

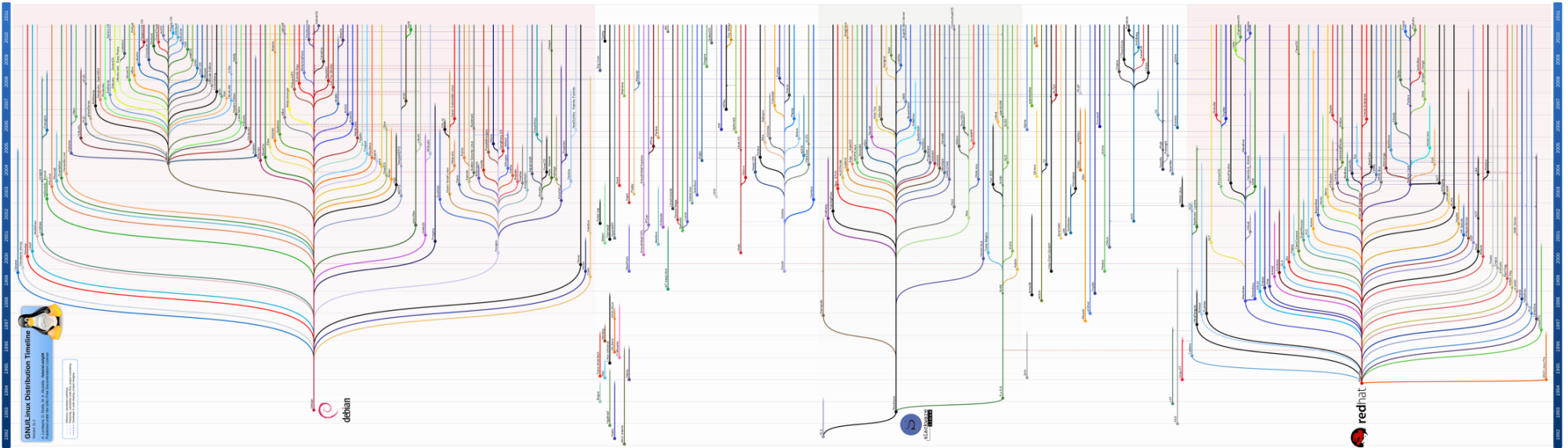


Figura 5: GNU/Linux, línea de tiempo

22.- Información I: Windows

En el caso de Windows: diversas versiones, todas de Microsoft.

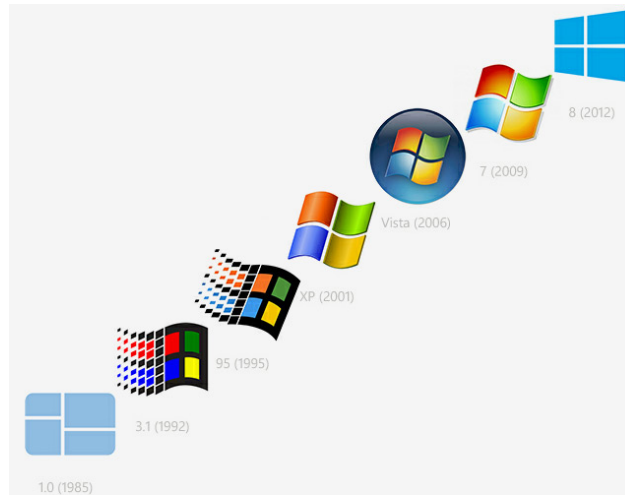


Figura 6: Versiones de Windows

23.- Sección a: tareas de un SO

24.- Información m1: gestión de procesos

Un proceso es un programa en ejecución que necesita recursos para realizar su tarea: tiempo de CPU, memoria, archivos y dispositivos de E/S.

Tanto GNU/Linux como Windows (hoy) son sistemas operativos multitarea, capaces de ejecutar varios procesos al “mismo tiempo”, sea secuencialmente o incluso en distintos procesadores o núcleos. Este tipo de SO normalmente asigna los recursos disponibles (CPU, memoria, periféricos) de forma alternada a los procesos que los solicitan.

El SO es el responsable de:

- ▶ Parar y reanudar procesos.
- ▶ Crear y destruir procesos. Vimos las interrupciones de hardware, también hay de software como por ejemplo, el enviarle una señal “kill” a una aplicación para terminarla. O excepciones, cuando por ejemplo se intenta dividir por cero. Generalmente, si no se fija otra política, el SO termina la aplicación ante este error.
- ▶ Ofrecer mecanismos para que los procesos puedan comunicarse y se sincronicen (señales, pipes, sockets, semáforos).

25.- Información m1: gestión de procesos

Implica una lista de tareas a realizar y fijarles prioridades.

Primero se ejecutan las de más alta prioridad. Una vez realizada la tarea se saca de la lista. Esto puede traer el problema que las tareas de baja prioridad nunca lleguen a ejecutarse y permanezcan en la lista para siempre. Para solucionar esto, se puede aumentar la prioridad de las tareas más antiguas.

Suele asignarse un número a cada proceso, llamado “pid” o “process id”.

26.- Información m2: gestión de recursos

El sistema operativo administra los recursos del sistema, como los ya vistos:

- ▶ La unidad central de procesamiento (donde está alojado él o los microprocesadores).
- ▶ La memoria principal (o de acceso directo) y el cache (o memoria secundaria).
- ▶ Los dispositivos de entrada y salida: teclados, pantallas, impresoras, discos duros, dvds, pendrives, etc.

Cada dispositivo suele tener su propio módulo de software que hay que activar cuando se usa.

27.- Información m3: gestión de usuarios

Tanto GNU/Linux como Windows (hoy) son sistemas operativos multiusuarios. Varios usuarios ejecutan simultáneamente sus programas, accediendo a la vez a los recursos de la computadora. Normalmente estos sistemas operativos utilizan métodos de protección de datos, de manera que un programa no pueda usar o cambiar los datos de otro usuario.

Al acceder a la computadora cada usuario debe indicar quien es y proporcionar una contraseña mediante el programa “login”.

Se puede recorrer las diferentes consolas de texto y gráficas con “ctrl alt f1/f7” para verificar lo anterior.

Todo esto implica la existencia de múltiples programas para administrar a los usuarios y sus derechos sobre diferentes recursos.

28.- Información m4: gestión de usuarios

Aspectos:

- ▶ seguridad, derechos o permisos (-rw),
- ▶ listas de control de acceso (ACLs),
- ▶ UGO (Usuario, Grupo, Otros, o por sus siglas en inglés, User, Group, Others),
- ▶ Capacidades granuladas,
- ▶ Atributos extendidos (ej.: sólo añadir al archivo pero no modificar, no modificar nunca, etc.).

El SO se encarga de:

- ▶ Distinguir entre uso autorizado y no autorizado.
- ▶ Especificar los controles de seguridad a realizar.
- ▶ Forzar el uso de estos mecanismos de protección.

29.- Información n: gestión de la memoria

El SO es el responsable de:

- ▶ Conocer qué partes de la memoria están siendo utilizadas y por quién.
- ▶ Decidir qué procesos se cargarán en memoria cuando haya espacio disponible.
- ▶ Asignar y reclamar espacio de memoria cuando sea necesario.

También existen mecanismos para administrar un “cache” de memoria en disco duro, por si se agota la memoria rápida. Generalmente se asigna durante la instalación una partición de “swap” del doble del tamaño de la memoria RAM.

30.- Información ñ1: gestión de dispositivos

Un sistema operativo moderno abstrae muchos de éstos periféricos en dos grandes tipos de dispositivos:

de caracteres: Un dispositivo de caracteres se comunica con la unidad central de proceso por medio de bytes individuales.

Ejemplos de dispositivos de caracteres son impresoras y teclados, que permiten el manejo de un solo byte a la vez.

de bloques: Se comunica con la unidad central de proceso en conjuntos indivisibles de tamaño fijo.

Un ejemplo de dispositivo de bloque es un disco duro, que está ordenado internamente en sectores que típicamente son de 512 bytes, y al escribir o al leer del disco es necesario transferir esa cantidad de información como conjunto.

Para recibir o enviar ese bloque se requiere especificar su dirección.

31.- Información ñ2: gestión de dispositivos

Como veremos en Unix, todo es un archivo. Se puede saber cuándo un archivo es para un dispositivo de carácter, de bloque u otros tipos mirando el primer carácter de la salida de `ls -l`.

- archivo común (una secuencia de bloques en un dispositivo de bloques).
- d** directorio, (archivo que contiene una lista de archivos).
- b** dispositivo de bloques.
- c** dispositivo de caracteres.
- l** un enlace a otro archivo (un nuevo nombre para algo que ya existe)
- p** cañería con nombre (named pipe) FIFO conectada a un programa
- s** un socket local conectado a un programa.

32.- Información o: gestión de módulos

Los sistemas operativos modernos, si bien no siempre se adecuan al concepto de microkernel, manejan muchos dispositivos con módulos.

Es decir el código para los miles de posibles dispositivos no está incorporado al binario del núcleo, sino que se puede incorporar en “vivo” el código específico para los dispositivos específicos que están presentes en cada momento.

Existen comandos especiales para ésta tarea.

33.- Información p: gestión de archivos

El SO es responsable de:

- ▶ Construir y eliminar archivos y directorios.
- ▶ Ofrecer funciones para manipular archivos y directorios.
- ▶ Establecer la correspondencia entre archivos y unidades de almacenamiento.
- ▶ Realizar copias de seguridad de archivos.

34.- CAPÍTULO 4: Archivos: la gran abstracción

Un sistema operativo abstrae cosas diferentes bajo un mismo concepto para simplificar.

Así una misma aplicación pueda acceder a entidades diferentes como si fuesen lo mismo.

Muchos dispositivos y entidades son manejadas por el SO bajo el concepto de archivo.

¿Que es un archivo?: es una secuencia de bits. Ceros y unos. Comienza y termina. A los efectos de su uso, en muchos casos pensamos los bits en palabras de 8 bits o bytes.

35.- Información q: archivos: propiamente dichos, procesos y dispositivos

Y los interpretamos con un código. El ASCII. Ver editor del mc

```
mc [dsa@dsa-fa]:~/svn/laboratorioc
/home/dsa/svn/laboratorioc/I-ducción al uso de Scilab.pdf 0x00000001e
00000000 20 20 20 20 20 20 20 20 20 55 6E 69 76 65 72 73 Univers
00000010 69 64 61 64 20 4E 61 63 69 6F 6E 61 6C 20 04 65 idad Nacional de
00000020 20 53 61 6C 74 61 0A 20 20 20 20 20 20 20 20 20 Salta.
00000030 20 46 61 63 75 6C 74 61 64 20 64 65 20 43 69 65 Facultad de Cie
00000040 6E 63 69 61 73 20 45 78 61 63 74 61 73 0A 0A 0A ncias Exactas...
00000050 0A 0A 49 4E 54 52 4F 44 55 43 43 49 C3 93 4E 20 ..INTRODUCCIÓ N
00000060 41 4C 20 55 53 4F 20 44 45 20 53 43 49 4C 41 42 AL USO DE SCILAB
00000070 0A 20 20 4C 65 6E 67 75 61 6A 65 20 6C 69 62 72 . Lenguaje libr
00000080 65 20 70 61 72 61 20 6C 61 20 70 72 6F 67 72 61 e para la progra
00000090 6D 61 63 69 C3 B3 6E 20 63 69 65 6E 74 C3 AD 66 mació n cienti f
000000A0 69 63 61 0A 0A 0A 20 20 20 20 20 20 20 20 20 20 20 20 41 ica... A
000000B0 75 74 6F 72 65 73 3A 0A 20 20 20 20 20 20 20 20 20 20 20 20 utores:.
000000C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 4D 69 67 Mig
000000D0 75 65 6C 20 43 6F 6E 64 6F 72 C3 AD 0A 20 20 20 20 uel Condori .
000000E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000F0 20 20 4D 61 72 69 65 6C 61 20 46 69 6E 65 74 74 Mariela Finett
00000100 69 0A 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 i.
00000110 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 Luis Sara
00000120 76 69 61 0A 0A 0A 0A 20 20 20 20 20 20 20 20 20 20 20 20 20 via.....
1Ayuda 2 3Salir 4Ascii 5Ir a 6Guarda 7Búsqueda 8Crudo 9Format 10Salir
```

Figura 7: Editando un archivo

Pero pueden ser cualquier conjunto de información relacionada, definida por sus creadores. Éstos almacenan programas (en código fuente y objeto) y datos tales como imágenes, textos, información de bases de datos, etc.

36.- Información r: archivos: propiamente dichos, procesos y dispositivos

Cualquier entidad representable de esta forma es abstraída por el SO como un archivo:

- ▶ archivos en el disco duro,
- ▶ teclado,
- ▶ pantalla en modo texto,
- ▶ impresora, etc
- ▶ el mismo disco duro es un archivo y cada una de sus particiones es un archivo.

37.- Información r: archivos y directorios

Un tipo especial de archivo es el directorio (o carpeta), que contiene una lista de archivos.

Así un archivo puede estar contenido en otro de dos formas diferentes (en una partición, y en un directorio).

Todo archivo esta en un árbol y existe un estándar que indica donde deben colocarse:

- ▶ dispositivos en /dev (habitualmente): los discos duros y sus particiones; los teclados y las pantallas.
- ▶ archivos de los discos duros en el resto del árbol.
- ▶ procesos en /proc

38.- Información r: archivos de dispositivos

Hay dos formas principales de que el núcleo se comunique con los procesos y los dispositivos (módulos). Una es a través de los ficheros de dispositivos (como los que están en el directorio `/dev`) y la otra es usar el sistema de ficheros proc. El propósito original de los ficheros de dispositivo es permitir a los procesos comunicarse con los controladores de dispositivos en el núcleo, y a través de ellos con los dispositivos físicos (módems, terminales, etc.).

Nota: Windows usa diferentes árboles para distintos dispositivos y los nombra con una letra seguida de dos puntos.

39.- Información s: rutas y nombre de archivos

Normalmente los archivos y carpetas se organizan jerárquicamente.

La estructura de directorios suele ser jerárquica, ramificada o “en árbol”, aunque en algún caso podría ser plana. En algunos sistemas de archivos los nombres de archivos son estructurados, con sintaxis especiales para extensiones de archivos y números de versión. En otros, los nombres de archivos son simplemente cadenas de texto y los metadatos de cada archivo son alojados separadamente.

En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada “ruta”, (path). La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una diagonal ('/') o diagonal invertida ('\') y puede terminar en el nombre de un archivo presente en la última rama del directorio. El punto representa el directorio “corriente” y los dos puntos el anterior. Una barra inicial indica la raíz del árbol, su ausencia que se parte del directorio corriente.

40.- Información t: ejemplo de 'ruta' en GNU/Linux

Así, por ejemplo, en un sistema tipo Unix como GNU/Linux, la ruta para la canción llamada “La canción.ogg” del usuario “Pedro” sería algo como:

“/home/Pedro/Mi música/La canción.ogg”

Donde: el primer '/' representa el directorio raíz donde está montado todo el sistema de archivos. 'home/Pedro/Mi música/' es la ruta del archivo. 'La canción.ogg' es el nombre del archivo.

Para programas no es buena idea usar espacios en nombres de archivos, dado que la “shel” usa el espacio para separar argumentos y comandos.

41.- Información t: ejemplo de 'ruta' en un sistema Windows

Un ejemplo análogo en un sistema de archivos de Windows (específicamente en Windows 8) se vería como:

`"C:\Users\Pedro\Music\canción.mp3"`

donde: "C:" es la unidad de almacenamiento en la que se encuentra el archivo.

`"\Users\Pedro\Music\"` es la ruta del archivo. "canción" es el nombre del archivo.

42.- Información t: extensiones

*.?, *.bat, *.exe, *.com, *.txt, *.sh, *.mp3

'*.mp3' es una extensión, este elemento, parte del nombre, es especialmente relevante en los sistemas Microsoft Windows, ya que sirve para identificar qué tipo de archivo es y la aplicación que está asociada con el archivo en cuestión, es decir, con qué programa se puede editar o reproducir el archivo. Sin embargo muchas veces Windows oculta la extensión del nombre y representa el tipo con un icono.

Asteriscos y preguntas para búsquedas. En los listados de archivos y caminos se pueden reemplazar letras por comodines y con esto se incluye a todos los que coinciden.

43.- Información u: tipo de sistemas de archivos

Un sistema de archivo de disco está diseñado para el almacenamiento de archivos en una unidad de disco, que puede estar conectada directa o indirectamente a la computadora.

Existen diferentes sistemas de archivos, es decir, existen diferentes formas de organizar la información que se almacena en las memorias (normalmente discos) de los ordenadores. Por ejemplo, existen los sistemas de archivos FAT, FAT32, ext3, NTFS, XFS, etc.

Desde el punto de vista del usuario estas diferencias pueden parecer insignificantes a primera vista, sin embargo, existen diferencias muy importantes. Por ejemplo, los sistemas de ficheros FAT32 y NTFS, que se utilizan fundamentalmente en sistemas operativos de Microsoft, tienen una gran diferencia para un usuario que utilice una base de datos con bastante información ya que el tamaño máximo de un fichero con un sistema de archivos FAT32 está limitado a 4 gigabytes, sin embargo, en un sistema NTFS el tamaño es considerablemente mayor.

Se establecen durante la instalación al formatear el disco.

44.- Información u: capacidades de los sistemas de archivos

- ▶ Mecanismo para evitar la fragmentación (GNU/Linux vs windows),
- ▶ Capacidad de enlaces simbólicos o duros,
- ▶ Integridad del sistema de archivos (Journaling),
- ▶ Soporte para archivos dispersos,
- ▶ Soporte para cuotas de discos,
- ▶ Soporte de crecimiento del sistema de archivos nativo.

45.- CAPÍTULO 5: El árbol estándar

Estructura del sistema de archivos de los GNU/Linux.

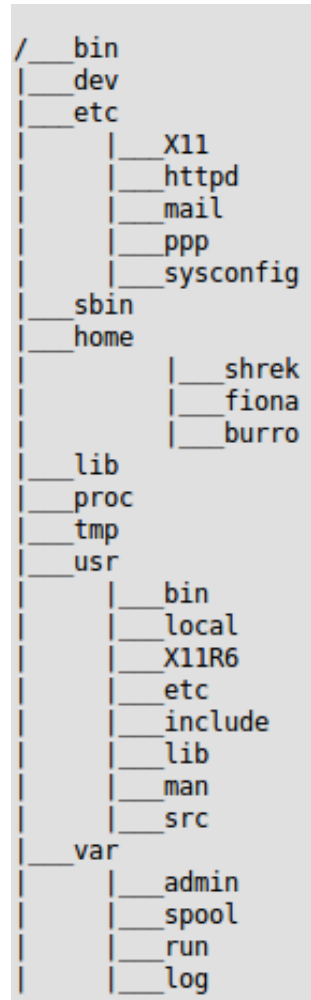


Figura 8: Sistema de archivos

46.- Nota d: el árbol genérico

Arboles Definir que sub-árbol al instalar un programa, para los:

- / imprescindibles para el arranque y conexión a red de la computadora,
- /usr/ solo lectura, montable vía red de un repositorio centralizado,
- /usr/local/ para aplicaciones no empaquetadas.

Aplicaciones Directorios para cada tipo de archivo de la aplicación:

- bin/ ejecutables para los usuarios
- sbin/ ejecutables para el superusuario y sistema
- etc/ configuración
- lib/ librerías binarias de los ejecutables
- include/ encabezados de las librerías
- share/ librerías de lenguajes interpretados o material de ayuda (man, info), compartibles entre arquitecturas diferentes
- src/ código fuente

Varios . dev, var, home, proc, tmp (varios), X11R6, srv

47.- Nota e1: el árbol estándar

- /bin** bin es la abreviación de binaries, o ejecutables. Es donde residen la mayoría de los programas esenciales del sistema. Use la orden `ls -F /bin` para listar los ficheros. Podrá ver algunas órdenes que reconocerá, como `cp`, `ls` y `mv`. Estos son los programas para estas órdenes. Cuando usa la orden `cp` está ejecutando el programa `/bin/cp`. Usando `ls -F` se verá que la mayoría (si no todos) los ficheros de `/bin` tienen un asterisco añadido al final de sus nombres. Esto indica que son ficheros ejecutables.
- /dev** El siguiente es `/dev`. Veamos de nuevo con `ls -F`. Los archivos en `/dev` son conocidos como controladores de dispositivo (device drivers) son usados para acceder a los dispositivos del sistema y recursos, como discos duros, módems, memoria, etc. Por ejemplo, de la misma forma que puede leer datos de un fichero, puede leerla desde la entrada del ratón leyendo `/dev/mouse`.

48.- Nota e2: el árbol estándar

- ▶ `/dev/console` hace referencia a la consola del sistema, es decir, al monitor conectado directamente a su sistema.
- ▶ Los dispositivos `/dev/ttyS*` y `/dev/cua*` son usados para acceder a los puertos serie. Por ejemplo, `/dev/ttyS0` hace referencia a “COM1” bajo MS-DOS. Los dispositivos `/dev/cua` son “callout”, los cuales son usados en conjunción con un módem.
- ▶ Los nombres de dispositivo que comienzan por `hd` acceden a los viejos discos duros ide. `/dev/hda` hace referencia a la totalidad del primer disco duro, mientras que `/dev/hda1` hace referencia a la primera partición en `/dev/hda`.
- ▶ Los nombres de dispositivo que comienzan con `sd` son dispositivos SCSI. Si tiene un disco duro SCSI, en lugar de acceder a él mediante `/dev/hda`, deberá acceder a `/dev/sda`. Las cintas SCSI son accedidas vía dispositivos `st` y los CD-ROM SCSI vía `sr`.

49.- Nota e3: el árbol estándar

- ▶ Los nombres que comienzan por lp acceden a los puertos paralelo. /dev/lp0 hace referencia a “LPT1” en el mundo MS-DOS.
- ▶ /dev/null es usado como “agujero negro”, cualquier dato enviado a este dispositivo desaparece. ¿Para qué puede ser útil esto? Bien, si desea suprimir la salida por pantalla de una orden, podría enviar la salida a /dev/null.
- ▶ Los nombres que comienzan por /dev/tty[0-9]* hacen referencia a “consolas virtuales” de su sistema (accesibles mediante las teclas Alt-F1, Alt-F2, etc). /dev/tty1 hace referencia a la primera VC, /dev/tty2 a la segunda, etc.
- ▶ Los nombres de dispositivo que comienzan con /dev/pty* son “pseudo-terminales”. Estos son usados para proporcionar un “terminal” a sesiones remotas. Por ejemplo, si su máquina está en una red, telnet de entrada usará uno de los dispositivos /dev/pty.

50.- Nota e4: el árbol estándar

- `/etc` contiene una serie de ficheros de configuración del sistema. Estos incluyen `/etc/passwd` (la base de datos de usuarios), `/etc/rc` (guiones de inicialización del sistema), etc.
- `/sbin` se usa para almacenar programas esenciales del sistema, que usará el administrador del sistema.
- `/home` contiene los directorios “home” de los usuarios. Por ejemplo, `/home/shrek` es el directorio del usuario shrek. En un sistema recién instalado, no habrá ningún usuario en este directorio.
- `/lib` contiene las imágenes de las librerías compartidas. Estos ficheros contienen código que compartirán muchos programas. En lugar de que cada programa contenga una copia propia de las rutinas compartidas, estas son guardadas en un lugar común, en `/lib`. Esto hace que los programas ejecutables sean menores y reduce el espacio usado en disco.

51.- Nota e5: el árbol estándar

- `/proc` es un “sistema de ficheros virtual”. Los ficheros que contiene realmente residen en memoria, no en un disco. Hacen referencia a varios procesos que corren en el sistema, y le permiten obtener información acerca de que programas y procesos están corriendo en un momento dado.
- `/tmp` Muchos programas tienen la necesidad de generar cierta información temporal y guardarla en un fichero temporal. El lugar habitual para esos ficheros es en `/tmp`.
- `/usr` Es un directorio muy importante. Contienen una serie de subdirectorios que contienen a su vez algunos de los más importantes y útiles programas y ficheros de configuración usados en el sistema.
Los directorios descritos arriba son esenciales para que el sistema esté operativo, pero la mayoría de las cosas que se encuentran en `/usr` son opcionales para el sistema.

52.- Nota e6: el árbol estándar

- `/usr/X11R6` contiene el sistema XWindow si se ha instalado. El sistema XWindow es un entorno gráfico grande y potente el cual proporciona un gran número de utilidades y programas gráficos, mostrados en “ventanas” en su pantalla.
- `/usr/bin` es el almacén real de programas del sistema Unix. Contiene la mayoría de los programas que no se encuentran en otras partes como `/bin`.
- `/usr/etc` Como `/etc` contiene diferentes ficheros de configuración y programas del sistema, `/usr/etc` contiene incluso más que el anterior. En general, los ficheros que se encuentran aquí no son esenciales para el sistema, a diferencia de los que se encuentran en `/etc`, que si lo son.
- `/usr/include` contiene los ficheros de cabecera para el compilador de C. Estos ficheros (la mayoría de los cuales terminan en `.h`, de “header”) declaran estructuras de datos, subrutinas y constantes usados en la escritura de programas en C.

53.- Nota e7: el árbol estándar

- `/usr/lib` contiene las librerías equivalentes “stub” y “static” a los ficheros encontrados en `/lib`. Al compilar un programa, este es “enlazado” con las librerías que se encuentran en aquí, las cuales dirigen al programa a buscar en `/lib` cuando necesita el código de la librería. Además, varios programas guardan ficheros de configuración en `/usr/lib`.
- `/usr/local` En general, los programas que se encuentran en `/usr/local` son específicos de su sistema.
- `/usr/man` Este directorio contiene las páginas de manual. Hay dos subdirectorios para cada página “sección” de las páginas (use la orden `man man` para más detalles). Por ejemplo, `/usr/man/man1` contiene los fuentes (es decir, los originales por formatear) de las páginas de manual de la sección 1, y `/usr/man/cat1` las páginas ya formateadas de la sección 1.
- `/usr/src` contiene el código fuente (programas por compilar) de varios programas de su sistema. El más importante es `/usr/src/linux`, el cual contiene el código fuente del Núcleo de Linux.

54.- Nota e8: el árbol estándar

`/var` contiene directorios que a menudo cambian su tamaño o tienden a crecer. Muchos de estos directorios solían residir en `/usr`, pero desde que estamos tratando de dejarlo relativamente inalterable, los directorios que cambian a menudo han sido llevados a `/var`. Algunos de estos directorios son:

`/var/log` contiene varios ficheros de interés para el administrador del sistema, específicamente históricos del sistema, los cuales recogen errores o problemas con el sistema. Otros ficheros guardan las sesiones de presentación en el sistema, así como los intentos fallidos.

`/var/spool` contiene ficheros van a ser pasados a otro programa. Por ejemplo, si su máquina está conectada a una red, el correo de llegada será almacenado en `/var/spool/mail` hasta que lo lea o lo borre. Artículos nuevos de las “news” tanto salientes como entrantes pueden encontrarse en `/var/spool/news`, etc.

`/var/mail` En algunas distribuciones como la RedHat, es donde se guardarán los emails que nos lleguen y mandemos.

55.- CAPÍTULO 6: Comandos y sus Intérpretes

Un “comando” invoca a

- ▶ un programa que se puede ejecutar,
- ▶ o un comando interno del “shell”.

Se hace:

- ▶ introduciendo su nombre en una terminal,
- ▶ clickeando en su icono en una ventana,
- ▶ la orden exec o similares de un programa,
- ▶ una interrupción.

56.- Nota f1: argumentos

Los argumentos son modificadores que cambian el comportamiento de un comando. Por ejemplo “ls -l”.

El argumento “-l” cambia el comportamiento del comando ls para listar información. En Unix/Linux, la mayoría de los comandos aceptan argumentos que modifican su comportamiento.

57.- Nota f2: comandos compuestos

Una de las características de los sistemas *nix es la capacidad de combinar comandos simples y complejos con el fin de obtener comandos compuestos. Un comando compuesto consiste en una lista de comandos simples y complejos separados por el carácter de punto y coma (;). Un ejemplo de un comando complejo es:

```
$ comando1 ; comando2 ; comando3 ; ...
```

En múltiples lenguajes de programación (Pascal, C, Perl, Ruby, etc), se utiliza el carácter de punto y coma para designar el final de la línea, y muchos programadores de shell scripts lo utilizan de la misma forma, mas como una cuestión de costumbre que funcional.

Hay otras formas de combinar comandos.

58.- Nota f3: ¿que es el shell?

Se utiliza un programa especial llamado shell para interpretar los comandos. El shell provee una interface al sistema y suele correr en una terminal. Obtiene la información del usuario y ejecuta programas en base a esa información. Cuando el programa finaliza su ejecución, despliega la salida del programa.

Por esta razón el shell es conocido como el interprete de comandos.

En GNU/Linux suele usarse un programa llamado “bash”, en Windows se le ha llamado “command.com” o “símbolo del sistema”, pero es muy limitado en funcionalidad. Para tener un intérprete de comandos moderno en Windows conviene instalar el programa CygWin.

El shell es un lenguaje de programación completo basado en comandos con variables de entorno, funciones y estructuras de control.

59.- Nota g: ejemplo, comando date

```
date [Enter] Fri Mar 19 21:34:59 CST 2004
```

Al introducir este comando, date despliega el nombre del día, mes, número de día transcurrido del mes, hora, zona horaria y año de el sistema.

Hay que hacer notar que después de ejecutar el comando, el sistema despliega el carácter \$ El carácter \$ indica el prompt (o línea de comando). Cuando se esta en el prompt, se puede teclear el nombre de un comando y presionar Enter. Esto ejecuta el comando que se tecleo. Mientras el comando esta siendo ejecutado, el prompt (\$) no es desplegado en la pantalla. Cuando el comando finaliza su ejecución, el prompt es desplegado nuevamente.

60.- Nota g4: algunos Comandos

- ▶ copy cp
- ▶ ren mv
- ▶ type cat
- ▶ md mkdir
- ▶ del rm
- ▶ dir ls
- ▶ rd rmdir
- ▶ cd cd

61.- Nota g5: redirección

redirecciones o cañerías: `>` `<` `>>` `<<` `2 >` `|`

así `ls > guardado` envía el listado al archivo guardado, `ls | sort` ordena la salida.

62.- CAPÍTULO 7: Sistema gráfico X - escritorio, gestión de ventanas

El sistema X Window es un sistema gráfico y distribuido. Es prácticamente el estándar de los entornos gráficos para usuarios de Unix, pero no es solo eso, es un completo sistema para redes que permite ejecutar aplicaciones X desde una computadora en la otra parte del globo de donde se encuentra el servidor, a través de Internet.

63.- Nota h1: puntos básicos

Una sesión de X esta compuesta por lo general de varias ventanas donde cada una de ellas ejecutan un programa aislado. Al igual que los programas en cualquier otro sistema, los programas que se ejecutan en X varían en cuanto a su funcionalidad.

Algunos necesitan de la interacción con el usuario y otros solo despliegan datos en la pantalla. A la ventana de fondo se la conoce como ventana raíz (root window). Las ventanas de aplicación se despliegan encima de la ventana raíz. Al igual que los procesos en Un*x estas ventanas están agrupadas o relacionadas de acuerdo a una jerarquía. Así es que la ventana raíz es el padre de todas las demás ventanas.

64.- Nota h2: cliente – servidor, eventos

X esta formado de dos partes: un servidor y un cliente. La funcionalidad básica es familiar a la forma en que trabajan todos los modelos cliente-servidor, en el sentido de que el servidor X tiene determinados recursos que proporciona al cliente. Debido a que el servidor X esta integrado a las redes TCP/IP las peticiones pueden venir de cualquier cliente y pueden ser solicitadas a cualquier servidor. Debido a que X, además de ser un programa específico, define un protocolo, los clientes pueden comunicarse con servidores X en otro tipo de plataforma. El servidor actúa como una interfaz entre los programas clientes y el hardware físico.

Cuando se proporcionan datos, ya sea por medio del teclado o el mouse, el servidor acepta la entrada y es responsable de pasarla al cliente. Esta información se pasa al cliente como un evento. Una tecla apretada o el movimiento del mouse generan un evento ante el cual reaccionará el cliente.

65.- Nota h1: entornos gráficos de escritorio

Existen entornos de escritorios, los cuales proveen un manejador de ventanas, utilidades varias y un framework de desarrollo que nos permitirán desarrollar aplicaciones que se integren al entorno perfectamente, compartiendo el look&feel del mismo y aprovechando servicios como arrastrar y soltar, servicio de impresión centralizado, un portapapeles común, etc. Algunos de estos entornos son KDE, y Gnome.

66.- Nota h2: KDE

KDE ofrece para GNU/Linux un entorno gráfico de configuración sencillo y con una apariencia homogénea entre las distintas aplicaciones. KDE es un proyecto de un grupo de programadores que se formó en 1996 y que siguen trabajando en el proyecto junto con muchos otros que fueron incorporándose. Ofrece su propio administrador de ventanas llamado kwin y además una serie de aplicaciones para hacerlo más cómodo como el konqueror que es un administrador de ficheros que permite trabajar con ellos en una forma muy gráfica a través de iconos. KDE está desarrollado con las bibliotecas gráficas Qt.

67.- Nota h3: Gnome

Gnome es un acrónimo de “GNU Network Object Model Environment”, entorno de trabajo en red orientado a objetos. Gnome forma parte del proyecto GNU. Gnome es un entorno gráfico (escritorio de trabajo) amigable que permite a los usuarios usar y configurar sus ordenadores de una forma sencilla. Gnome incluye un panel (para arrancar aplicaciones y presentar el estado de funcionamiento), un escritorio (donde se pueden situar los datos y las aplicaciones), un conjunto estándar de aplicaciones y herramientas de escritorio, y un conjunto de convenciones que facilitan la operación y consistencia de las aplicaciones entre sí. El gestor de sesiones de Gnome recuerda la configuración previa, de manera que una vez que haya configurado las cosas a su gusto, las mantendrá así. Soporta muchos idiomas, y puede añadir más sin cambiar el software. Soporta incluso varios protocolos de arrastrar y soltar para una máxima interoperabilidad con aplicaciones que no sean compatibles con Gnome.

68.- Nota h4: interfaz gráfica de usuario (Graphical shells), Unity

Las interfaces gráficas nos permiten manipular los programas basados en los GUI (interfaces gráficos de usuario). Permitiendo abrir, cerrar, mover o cambiar el tamaño de sus ventanas, así como cambiar de foco entre ventanas.

Pueden ser parte del entorno gráfico o separadas.

Unity es una interfaz de usuario para GNOME creada por Canonical para Ubuntu.

69.- Nota h5: Windows

Windows tiene una interfaz local, propia, monolítica e inseparable, que ha ido evolucionando en las sucesivas versiones del SO.

70.- CAPÍTULO 8: Instalación de aplicaciones

Existen dos formatos ampliamente usados en el universo del software libre. El formato deb, de Debian y Ubuntu y el rpm de Red Hat, usado por OpenSuse y muchas otras distribuciones.

Hay que tener en cuenta que no todos los paquetes que usen el mismo formato de paquete pueden ser instalados en una máquina en particular.

Deben estar contruidos para la misma distribución y versión, para tener compatibilidad binaria y corresponder las librerías que se requieren con la de los paquetes de los que dependen. Aquí el concepto clave es el de dependencia. Un paquete depende de otros y esto debe funcionar armónicamente.

Por ello cada distribución tiene repositorios de paquetes coherentes.

Esta es probablemente una de las contribuciones mas importantes del mundo del software libre, al software en general.

La actualizaciones del sistema operativo y sus aplicaciones se hace automáticamente a partir de los repositorios, mediante aplicaciones a estos fines. En el caso de Windows cada desarrollador de software se ocupa de su empaquetado de la forma que prefiera y no suele haber repositorios centrales.

71.- CAPÍTULO 9: Instalación

La instalación se hace en general desde un DVD o pen drive que porta un sistema GNU/Linux que funciona sin instalar (live), al prender la máquina. Una vez funcionando este sistema, se comienza particionando el o los discos. Luego se pueden instalar otros sistemas operativos en sus respectivas particiones. Y finalmente conviene instalar el sistema operativo GNU/Linux con el que comenzamos para configurar el programa que elige, al encender la computadora, cual sistema se va a usar.

72.- Nota i: particionado

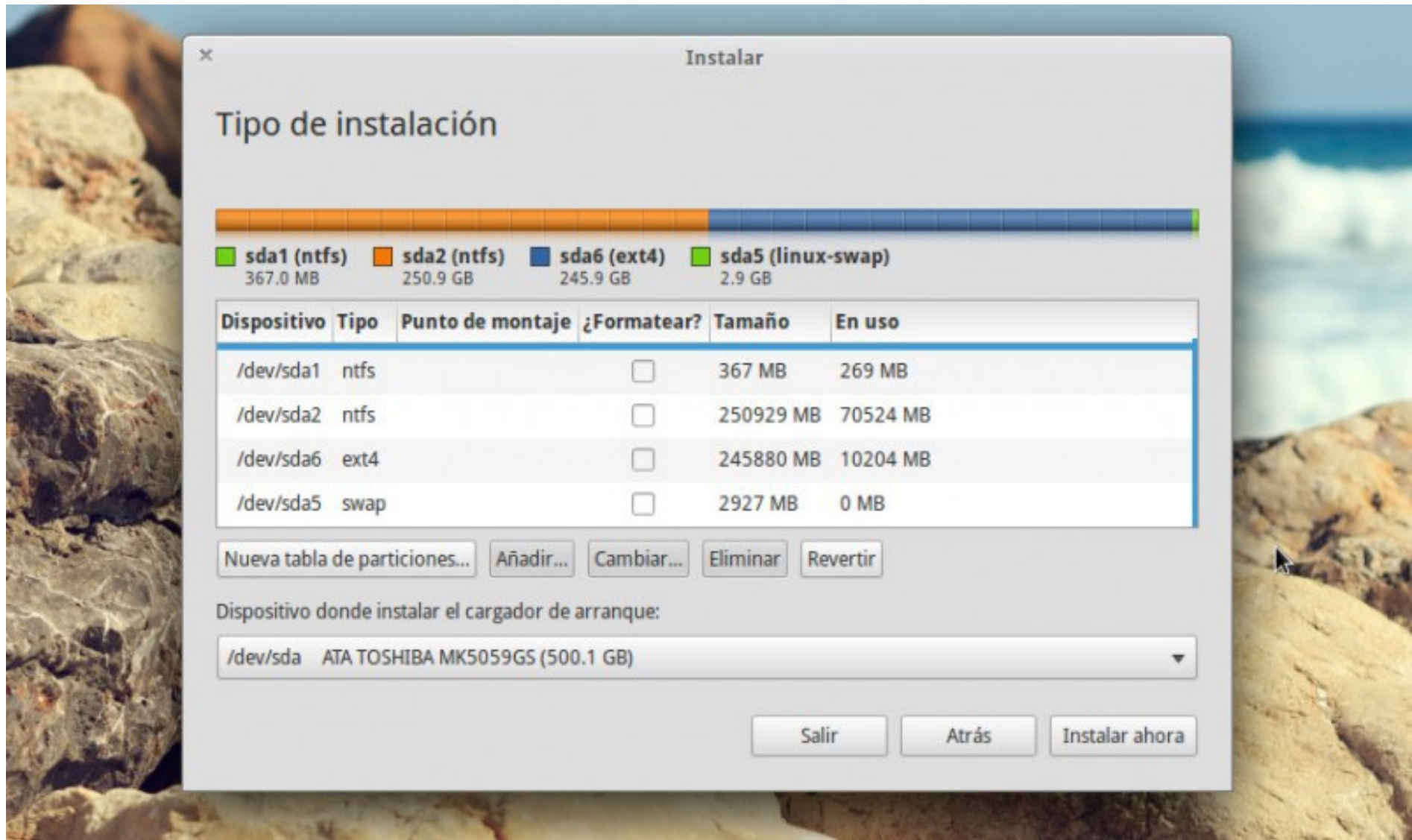


Figura 9: Particionado en la instalación

73.- CAPÍTULO 10: Práctico

- ▶ instalar Ubuntu y Windows en forma simultanea en una computadora.
- ▶ instalar Scilab y Cygwin en Windows, Scilab en Ubuntu
- ▶ realizar tareas sencillas en una terminal, camino en archivos, matar aplicaciones
- ▶ ejecutar comandos del sistema operativo desde Scilab