Resumen Sistemas Operativos

(Debian-Deepin)

Menú:

1. Introducción. Lunix
   1. Linux vs Microsoft
   2. Uniti
2. Lenguajes de programación
3. Kernel
4. Otras diferencias.
5. Copyright y copyleft

INTRODUCCION

Objetivo manipular servidores y entender la comunicación de esos servidores.

En Linux (servidores) corren todos las páginas y aplicaciones web.

Todos los SO son multiusuarios, multitareas y portables. Tienen seguridad desde los 70.

El SO más usado en el mundo es Android (derivado de Linux)

Windows vs GNU Linux (botón de colores vs performance)

Linux no fue pensado para usuarios. Y maneja mediante línea de comandos.

En Linux podemos escoger los servicios que van a correr (memoria RAM y ciclos de reloj).

Esto hace que podamos aprovechar al máximo nuestro equipo

Es software libre.

Existe versiones (distros (**con sus propios kernel**)) pagas de Linux, que ofrecen con algunos servicios, aunque también ofrecen versiones libres.

Para instalar se pide password, lo hace más seguro.

Windows tiene una interfaz gráfica que hace que consuma más recursos.

Corren la mayoría de los juegos populares

Es famoso por el marketing y la sencilles de su uso

IBM busca un sistema operativo para su hardware y busca a Gary Kildall (se fue de viaje y no concreto la reunión)

Bill Gates, al ver la oportunidad, obtuvo un clon de CP/M modifico y vendió a IBM (MS-DOS “Windows”). En el acuerdo entre IBM, Bill gates se quedo con la facultad de poder vender el software a otras compañías.

Pilares de Microsoft : suite de ofimática y en carta.

60-70 SO UNIX es el predecesor de Linux, es una versión paga.

Fue escrita en ensamblador inicialmente, y se reescribió completamente en C

Nacen de la necesidad de administrar MainFrame (Grandes maquinas multiusuario, multitarea) controlar desde cualquier lugar por distintos usuarios en tiempo real.

LENGUAJES DE PROGRAMACION

Lenguaje maquina (Assembler): perecido a binario, pero con alias, esa abstracción permitió la aparición del lenguaje B y C (fines de los 60).

C COBOL Y BASIC son los primeros.

Son es un lenguaje especial con su propia sintaxis, reglas y palabras clave usado para dar instrucciones a la maquina y permiten el desarrollo del software.

Nives:

Lenguaje de bajo nivel:

* Lenguaje de maquina: binario
* Lenguaje de ensamblado: assembler

Lenguajes de alto nivel:(abstracción del lenguaje)

* Permiten el manejo de hardware: C

Leguajes compilados: son directa y completamente traducidos a lenguaje máquina para ser ejecutado en el hardware. (solo sirve para un sistema operativo y tipo de maquina)

Lenguajes interpretados: necesitan de un software intermedio(interprete) y es el encargado de ejecutar las instrucciones (el intérprete corre en varios so y corren sobre la marcha)

(React, angular, typescript, JavaScript) para freelance

PARADIGMAS

Programación orientada a objetos: todo es un objeto

**KERNEL**

Esta dentro del SO(cada uno tiene su propio kernel)

Es un intermedio entre el hardware y el sistema operativo, da un acceso seguro

Se encarga de manejar los drivers y las funciones. Permite el acceso a los periféricos que se conecten al equipo

Administra la memoria para que se utilice de manera eficiente por los programas.

Responsable de gestionar el tiempo del procesador que es usado por un programas y procesos.

Primer lanzamiento del kernel de Linux, se ha ido mejorando por las comunidades.

Otras diferencias entre SO

* Interprete de línea de comandos+
* Su interfaz gráfica. Linux cuenta con una interfaz de usuario básica (x Windows)
* Programas unitarios y librerías.

**Copyright (propiedad intelectual)**

Son leyes para cuidar a distintos autores de las copias, plagios o ventas ilegales de sus obras.

La única persona que tiene control del contenido son los **productores económicos.** Esto obliga pagar regalías o autorización.

Existe la posibilidad de dar derechos de copias a terceros o delegar completamente los derechos a otra persona o **empresa.**

Los genios o incomprable se separan. Caso MySQL, lo vendió y creo maría DB.

Copyright en software: al ser leyes antiguas no aplican, porque dentro del proceso de un programa este se copia una y otra vez dentro de las distintas memorias de nuestro ordenador.

Nace Copyleft: no tiene límite ni restricción…

RICHARD STALLMAN

Creo una licencia creo un proyecto GNU, con el fin de escribir un sistema operativo completo, basado en UNIX, pero sin restricción al código fuente. Creo GPL

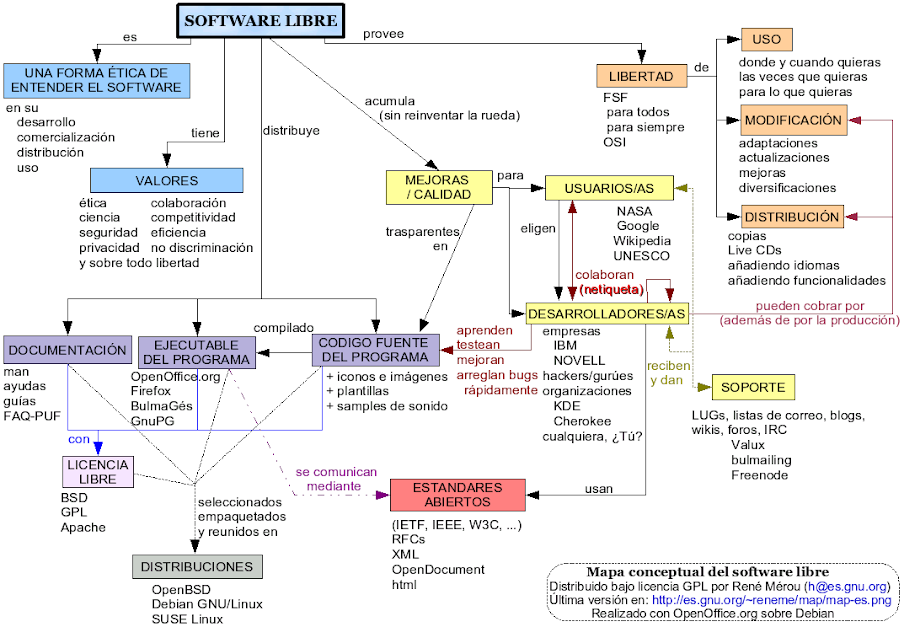
LICENCIA GLP: posibilita que nosotros podamos compartir el software.

FREE SOFTWARE FUNDATION: establecer una definición del software libre:

Pertenece a la sociedad, que la misma la use y mejore. Acceso al código fuente.

OPEN SOURSE INICIATIVE: Esta más basado en acceso a los softwares gratuitos (no tanto al código fuente, puede o no).

SOFTWARE LIBRE: Es el acceso al código fuente (no software gratis). **Las empresas cobran por el servicio no por el software.**



* Libertad de usarlo para cualquier propósito.
* Examinar el código fuente y modificarla
* Libertad para redistribuir el software o el nidificado.

**LICENCIAS DE SOFTWARE**

Las empresas invierten en un proyecto. Las licencias es una forma de comercializar y mantener esa comercialización. (inversión > idea intelectual esto pasa en la actualidad).

Una desventaja es que no nos permite modificar el producto o mejorarlo, aguantarnos con lo que nos vendieron.

Nacieron las iniciativas del software libre (mayor colaboración).

**CLOUD**

Ofrecer un servicio de computación atreves de internet, red interna, etc.

John McCarthy propuso la computadora como un servicio público.

* Privada: exclusivamente para uso empresarial.
* Publico: abierta a todo publico
* Hibrido.

NIVELES

SaaS: servicios en la nube, sin control ni conocimiento de la infraestructura. (Google doc)

PaaS: intermedio puede desarrollar sus propias aplicaciones. Sin preocuparse de los recursos y memoria, estos corren a cargo del proveedor. (hosting).

IaaS: es dar una computadora y el usuario decide que hacer con ella. El equipo tiene en control total, la empresa es responsable de las configuraciones, capacidad de memoria y/o procesador.

VENTAJAS

Respaldo en contra accidentes

Portabilidad

Prestación a nivel mundial

Actualizaciones automáticas y optimización de recursos IT

DESVENTAJAS

La disponibilidad de la información está sujeta a la disponibilidad de internet.

Disponibilidades de servicios especializados pueden tardar menses o años.

Seguridad: la información recorre distintos métodos o canales y cada uno de ellos representa un foco de inseguridad

Escalabilidad a largo plazo: a medida que nuestra estructura crece y si el proveedor no tiene un buen esquema de crecimiento puede llevar a la degradación del servicio.

**MEMORIAS**

**GESTORES DE PAQUETES**

Así como GitHub, con bodegas donde se encuentran alojadas los paquetes a instalar. No son .exe

Nos permiten acceder al código desde cualquier lugar.

Pueden ser: programas, archivos, módulos, bibliotecas, documentación.

TIPOS:

Estables: ya testeados y actualizados. Son los que los fabricantes nos recomiendan y aseguran funcionar al 100%

Prueba: Es la última versión salida recientemente, funcional pero falta de testing.

Inestable: aun sigue en desarrollo, con esto se espera que el testing lo haga el usuario corregir los bugs a la espera de su versión estable.

LISTA DE PAQUETES:

/etc/apt/sources.list

/etc/apt/sources.list.d/\*

BASE DE DATOS:

/var/lib/apt/lists/.

GESTOR DE PAQUETES **APT**

**(**El que vamos a usar nosotros**)(**usados para los derivados de debian**)**

|  |  |
| --- | --- |
| sudo apt-get update | actualiza la lista de paquetes disponibles para instalar |
| sudo apt-get upgrade | actualiza el SO y todos los paquetes instalados dentro del sistema operativo. |
| apt list --installed | lista de paquetes instalados. |
| sudo apt-get install mc  sudo apt-get --reinstall mc | se instala un paquete, en este caso mc  reinstala por errores. |
| **get-cache search mc** | **busca todo lo referido a mc dentro de nuestro cache** |
| sudo apt-get remove mc  sud apt-get --purge remove mc | elimina un paquete  ademas elimina todo los archivos q creo durante su utilización. |
| sudo apt-get clean  sudo apt-get autoclean  sudo apt-get autoremove | elimina el cache y los paquetes que bajamos de los repositorios  elimina paquetes viejos, no usados por actualizaciones.  borra las dependencias que viene luego de instalar el paquete |
| apt search terminator | busca dentro de la lista de paquetes las aplicaciones con el nombre terminator |
|  |  |
| 15) Instalar un paquete que tengamos en nuestra pc (un archivo .deb)  # dpkg -i virtualbox-5.1\_5.1.26-117224~Ubuntu~zesty\_amd64.deb  Instala directamente el paquete que bajamos. A diferencia del comando apt-get install  paquete, al usar el comando dpkg -i, no baja las dependencias necesarias. Por dicha razón,  tenemos que usar el comando apt-get -f install.  16) Listar los paqutes instalados  # dpkg -l | more  Con este comando podemos ver todos los paquetes que tenemos instalados en nuestro equipo  tanto sea por el comando apt-get install como también dpkg -i.  17) Ver contenido de los paquetes  # dpkg -L mc | more  Esto nos muestra todo el contenido del paquete que fue instalado, toda su estructura de  archivos.  18) Borrar un paquete  # dpkg -r mc  Borrar un paquete instalado mediante dpkg -i.  19) Buscar a qué paquete pertenece un comando.  # which ls  # dpkg -S /bin/ls  20) Configurar un paquete  # dpkg --configure paquete  21) Reconfigurar todo  # dpkg --configure -a  Esto nos permite reconfigurar los paquetes que quedaron sin configurar. | |

**SISTEMAS OPERATIVOOS SEGUNDA PARTE CURSADA**

**Procesos**

Son las instrucciones que están en la CPU listas para ser ejecutadas, en su estado actual: muerto, espera o activo. Y los recursos: memoria y su contenido.

Es un programa que se ejecuta en un determinado momento. Junto a una serie de procesos, sin intervención del usuario y hacen que el sistema sea utilizable.

**Proceso de arranque:** al prender la máquina, esta enciende sin tener la SO siquiera

* Primer proceso: BIOS. Empieza un proceso verificación de periféricos de entrada y salida, y almacenamiento. Si este procedimiento falla la maquina nos informa mediante pitidos la parte de la falla.
* Segunda parte bootloader: Este programa se encuentra guardado en el MBR,
* Tercera etapa: kernel: maneja la comunicación entre hardware y hardware, sofware y sofware y hardware y software. Lanza un primer proceso (init)(padre) y este larga todo los demas proceso.
* Cuarto proceso: Init. Es el primer proceso del sistema operativo. Suelta todos lo procesos (**hijos**) para que el sistema operativo este listo para el uso. Y cada proceso tiene una una dirección de memoria(ósea se pude ejecutar o borrar).
  + Procesos de usuario: cuando queremos performance y ejecutamos solo los que necesitamos, pero solo el dueño lo puede usar.
  + Procesos demonio: esta es cuando queremos una pc que venga lista para toda clase de usuarios pero es superlenta ya que tiene todos los procesos listos para su uso.

Swaper corre a la misma distancia init: traslada de la memoria a la memoria virtual.

Dentro del directorio /etc se encuentra el activo inittab, está la podemos modificar para definir que procesos se van a ejecutar al inicia del sistema operativo.

Manejo de procesos: las estructuras de datos de procesos contienen la información: donde se guarda tamaño, si está activo, espera o muerto, su prioridad, quien es el usuario grupo que tiene acceso.

**Atributos de los procesos**

PID (Process IDentiication): número de procesos si queremos pasarlo a background. Para ubicarlo y matarlo.

PPID: quien es el padre. **Si este es el PDI 1 entoces es el Init**

UID, GID (User IDentification, Group IDentidication): solo los números que indican al usuario que le pertenece y grupo. Solo el root o el usuario que lo creo pueden modificar el estado de operación de los procesos

EUID, EGID (Effective User IDentification, Effective Group IDentification): estos números identifican al usuario que ejecutan estos procesos.

Prioridad: esto determina que el que tiene mayor prioridad tendrá mas tiempo del procesador y en cada ciclo lo elegirá mas veces para ejecutarlo. Si no se indica este adopta el del padre.

**Creación y ejecución de un proceso**

**Estados del proceso**

Ejecutándose (R)

Durmiendo (S)

Intercambio (swapped): en la espera de algún recurso.

Zombi (Z): el proceso trata de finalizar su ejecución. Un lugar en la memoria de un proceso que no funciona. Un proceso que está a la espera de un proceso que ya no existe.

Parado (stopped): no puede ser ejecutado.

*cuando existe memoria y tiempo del procesador el proceso es ejecutado, cuando no el proceso pude ser mandado a dormir.*

Procesos dormidos: esperan a que ocurra un determinado evento. No consumen tiempo de proceso.

Procesos de intercambio: se encuentran en la memoria de intercambio (más lenta) cuando no hay espacio en la principal. Se crea una copia de los datos de la ram al disco duro sin transformarlo con el microprocesador para pasarlos la nuevamente a la ram de una manera más rápida.

Proceso zombi: no se puede ejecutar, sin espacio en la memoria, sino la memoria que espera el dato si existe

Proceso parado: no se pude ejecuta. Como parar (ctrl+z)

**Señales**

**…** lista de señales el el pdf.

**LVM(continuación de memorias)**

Permite agrandar o achicar el espacio de memoria de una carpeta mediante volúmenes lógicos.

Se pude asignar mas memoria sin la necesidad de apagar la maquina(en caliente) si el usuario asi lo desea.

RAID Y LVM

No es una alternativa al raid sino un complemento.

Ventajas de lvm: añade memoria dinámicamente, en caliente. En caso de falla de un disco permite la migración de datos de manera sencilla.

Ventajas RAID: mejora la performance de lectura y escritura. Hace backup en tiempo real.

**Protocolos**

Son una serie los pasos a seguir y en el tiempo estipulado. También cosas que no se deben hacer.

Es usada como vía de seguridad o como una red de comunicación.

**Modulo OSI (protocolos para redes)**

Es un modelo estándar y existen 7 niveles de capas y cada capa es un protocolo. Es un modelo de seguridad para que la comunicación (emisor y receptor) se realice en forma segura.

**Modelo Cliente-Servidor**

Es el intercambio de información cuando el cliente solicita la información y este servidor la transmite respetando los protocolos

Cliente -servidor

Request-Response

**Toquen:** deja la sección activa hasta que tiene una determinada vida.

**Algunos protocolos:**

HTTP: trabaja con el puesto 80, protocolo que permite la comunicación cliente-servidor a través de la red. Las request se realizan a través de una dirección URL, además es necesario establecer el método a usar: get, post, put, delete…

XML: Cumple la misma función que el JSON, pero es antigua

JSON: Forma de escribir un array, objeto o array de objetos y permiten ser exportado y usado en otros lenguajes etc.

FTP: es la forma de acceder a un host (disco). Tenemos acceso a ese archivo (carpeta), y usamos un programa para meterle archivos al host. (algo antigua).

SSH: A través de una autenticación. Es un acceso a una maquina virtual del cual témenos privilegios de host, pero es manejado por líneas de comandos, acceso a una consola de forma remota.

ARP: Tiene la función de acceder a la dirección MAC a través de su dirección IP estando en red. Lo puede solicitar un pc a otra, modem, switch, etc. Y estas son guardadas en una tabla MAC para agilizar la comunicación. Linux es mas segura para estos ataques con arp request