

ISO-Teoría 1er parcial

GNU/Linux GNU es un S.O. de código abierto, y gratuito iniciado en 1983, x Richard Stallman q el objetivo de crear un sistema operativo libre qe permita tener un control total del ordenador.

características de libre distribución y código abierto

personalizable

estable y seguro

casi seguro, multiusuario, multiprocesador y multitarea.

multipropósito

multiplataprima

Historia: Stallman se enfocó en crear una herramienta de software libre y gratuita pero fallaba el kernel. En 1991 Linus Torvalds creó el núcleo Linux el cual se convirtió en el S.O. GNU/Linux.

- **multitarea:** capacidad del SO de hacer varias tareas a la vez. GNU/Linux es multitarea. Gracias al planificador de tareas se puede ejecutar varios procesos paralelamente. El planificador asigna RR de HW a los <> procesos garantizando tiempo de procesador y memoria p/ q sea eficiente.

distribuciones: una distribución es una versión del SO q tiene el núcleo Linux, herramientas, utilidades y además software adicional seleccionado p/ d distribución. C/ distribución es única en términos de SW, apariencia, características, etc.

C/ distribución se diferencia en objetivos, enfoque y filosofía, además de distintos conjuntos de SW, paquetes preinstalados y forma de gestión y actualizac.

Debian distribución enfocada en la estabilidad, seguridad y libertad.

POSIX estándar que define una interfaz p/ el SO y las apps. Espera una serie de interfaces y comandos q deben estar presentes en todos los S.O. Incluyendo interfaces de gestión de procesos, archivos, directorios, terminales y señales.

componentes GNU/Linux - **kernel:** núcleo del SO q gestiona los RR del HW.

- **sistema GNU:** proporciona herramientas, utilidades básicas del sistema, como shell, compiladores, bibliotecas o programas del sistema.

- **sistema de archivos:** organiza y arrode a los datos y programas del sist.

estructura básica del SO GNU/Linux

/raíz del sistema: contiene todos los archivos y directorios del sistema y es el pto. de partida p/ el resto de las rutas.

/bin contiene archivos binarios o ejecutables p/ el funcionamiento del sistema como la shell o programas básicos.

/sbin binarios o ejecutables q administran el sistema.

/etc: archivos de configuración del sistema, de red, de usuarios, etc.

/dev: archivos de dispositivos del sistema como discos duros, USB, etc.

/home: directorio de los usuarios.

/usr: archivos del programa y bibliotecas o manuales.

/var: datos variables del sistema

/proc: sist. de arch. virtual q' guarda info de procesos, del kernel, etc.

/boot: archivos necesarios p/ arrancar el sistema, incluyendo gestor de arranque y kernel

/mnt: pto de montaje p/ sist. de arch. externos

KERNEL

Administra RA del HW, da una interfaz de programax, administra procesos del sist, administra la memoria y el sist. de archivos. La versin actual es la 5.16.0.

versiónado: antes de 24, el esquema de versionado tenía una numerax impar p/ versiones de desarrollo y nros. pares p/ versiones estables.
A partir de 26 la numerax cambia a 3 nros. Dado A.B.C, A indica una versión importante, B es una versión menor y C indica una correx de un cambio pequeño.

-Puede haber más de un Kernel en una misma máquina.

: GNU/Linux tiene un kernel monolítico híbrido. Con monolítico se refiere a que los fx se ejecutan en mismo espacio de memoria. e híbrido se refiere q' puede cargar y descargar funcionalidades.

SHELL

es una interfaz de línea de comandos p/ q' los usuarios interactúen c/ el SO. p/ gestionar arch, directorios o ejecutar scripts.

Los comandos de la shell no tienen una ubicación específica. Cuando se escribe un comando en la terminal, el shell busca el comando en cada directorio especificado en PATH.

Además la shell no forma parte del kernel pq shell es una interfaz p/ interactuar c/ el S.O.

Se puede definir tmb. un intérprete de comandos p/ c/u usuario. La definix se hace en /etc/passwd. El cual tiene info de c/u usuario.

FILE SYSTEM

forma en la q' un S.O organiza y administra los archivos y directorios del S.O.

file systems soportados en GNU/Linux

ext4: s.a x mejoró, c/g eficiencia y fiabilidad.

btrfs: características avanzadas como compresión, almacenamiento en RAID y verificación, correx de errores

XFS: sistema de archivos usado p/ servidores y sistemas empresariales.

NTFS: sistema de arch. nativo de Windows, GNU/Linux puede leer y escribir estas particiones

FAT32: sistema común usado en disp. externos como USB, memorias SD

FHS: Filesystem Hierarchy Standard, estándar q' establece la estructura de los directorios y archivos del sistema de archivos de un S.O./Unix.

PARTICIONES: división lógica de un disco en una o más partes separadas donde c/u tiene su propio file system. C/ partición se trata como una unidad separada x el S.O y pueden tener un tamaño y formato de sistema diferente.

Las particiones pueden ser:

- primarias: contienen un sistema de archivos y pueden ser montadas en el sist.
En una tabla de particiones, solo se pueden hacer 4 particiones primarias.
- extendidas: no tienen un sist. de arch., pero pueden dividirse en varias particiones lógicas.

En un disco puedo tener 4 primarias o 3 primarias y 1 extendida.
Las particiones permiten organizar la info del disco c/ > estabilidad.

identif/cas: se hace mediante una nomenclatura basada en el dispositivo físico al cual está asignado y el número de partic.

• Discos IDE/SATA: /dev/hdXn o /dev/sdXn, X = letra del dispositivo físico n= número de partic.

• Discos SCSI: /dev/sdXn, X = letra asignada al dispositivo SCSI, n = nro. de partic.

Particiones q/ instalar GNU/Linux se necesitarán al menos 2 particiones, una p/ la raiz (/) y otra p/ la memoria de intercambio (swap).

"gral" se identifica como "sda1" o "sda2" y el sistema de archivos puede ser ext4, ext3, xfs, btrfs, etc.

La memoria swap se usa p/ almacenar temporal" datos q' no caben en RAM. y se identifica como sda3

ARRANQUE

BIOS (Basic Input/Output System) es un pequeño programa q' se ejecuta inmediatamente desp. de encender el equipo. Realiza una serie de pruebas y comprobaciones del HW. Además carga el SO en RAM y proporciona una interfaz p/ configurar componentes del sistema.

UEFI (Unified Extensible Firmware Interface) firmware de arranque en vez de usar la BIOS tradicional. Prepara al HW y al sist. p/ el inicio del S.O.

Un firmware es un tipo de software incorporado en el HW de un dispositivo y controla su funcionamiento

mbr (Master Boot Record): sector de 512 bytes q' contiene el código de arranque y la tabla de particiones del disco. Tamb. contiene un pequeño programa llamado "gestor de arranque" o "bootloader" q' inicia el proceso de carga del S.O.

Una tabla de particiones es una estructura de datos q' está en el sector de arranque del disco duro y q' contiene informacion sobre cómo se divide y organiza el espacio de almacenamiento del disco en particiones.

MBR (Master Boot Code): código q' se ubica en el bootloader y se carga en el proceso de arranque. El MBR se carga en memoria y ejecuta el bootloader.

GPT (GUID Partition Table): tabla de partición usada p/ reemplazar al MBR de las formas tradicionales. GPT usa identificadores únicos p/ identificar las particiones en un disco duro, permitiendo la creación de un gran nro. de particiones.

Gestor de arranque: cargado en RAM p/ iniciar el S.O y otros programas p/q' el syst funnione.

Proceso de Bootstrap: proceso mediante el cual se carga el SO en RAM y prepara el equipo p/q' se use. Los pasos son:

- Encendido

- Post: el firmware (BIOS/UEFI) hace una autopregunta de encendido p/ verificar que el hardware de la computadora esté funcionando correcta'

- Inicio del bootloader: completado el post, el firmware carga el bootloader desde el sector de arranque del disco duro

- Selección del SO:

- Carga del Kernel

- Inicio del S.O

Proceso de parada/shutdown:

- el usuario decide q' quiere apagar el sist.

- se notifica a apps. y procesos en ejecución q' el sist. se va a apagar y los da tiempo p/ finalizar.

- servicios: se detienen los servicios y demonios ejecutándose. Los demonios son programas q' se ejecutan en segundo plano.

- desactivación de dispositivos:

- Kernel: el Kernel finaliza las tareas pendientes como cancelar interrupciones o tareas en ejecución.

- apagado.

Dual boot: capacidad de tener varios S.O instalados en un mismo dispositivo. Con GNU/Linux es posible.

System V: este tipo de S.O tiene el siguiente proceso de arranque

- POST: realiza una prueba de autodiagnóstico del HW p/ detectar errores.

- Carga el gestor de arranque: es el 1º programa ejecutado y carga el S.O en los S.O System V el gestor de arranque suele ser GRUB (Grand Unified Bootloader). Grub es un gestor de arranque q' carga el Kernel del S.O y lo transfiere el control p/q' inicie el S.O

- Carga del Kernel: el bootloader carga el Kernel en la memoria y lo inicia

- Ejecución de "init": el proceso init es el 1º proceso q' inicia desp. del Kernel y realiza la inicialización del sistema y gestiona los procesos y servicios del arranque. En System V, init es responsable de cargar <> niveles de ejecución y los servicios de cada

runlevels: clasifican al S.O en <> modos de funcionamiento, c/u c/u servicios y procesos específicos / el objetivo de controlar los servicios y procesos en función de lo situado en la q' se encuentren

- runlevel 0: apagado del sistema.
- runlevel 1: modo usuario único o de mantenimiento del sist, no tiene servicios de red o de múltiples usuarios.
- runlevel 2: modo múltiples usuarios sin servicio de red.
- runlevel 3: " c/u servicio de red y sin entorno gráfico
suele ser el q' se inicia la PC
- runlevel 4: no usado predeterminada, se puede personalizar
- runlevel 5: modo múltiples usuarios c/u servicios de red y entorno gráfico.
- runlevel 6: reinicio del sistema y cambiar.

archivo /etc/inittab: su finalidad es iniciar el sistema / <> runlevels. Almacena tab. informa sobre los procesos q' deben iniciarse en c/u runlevel.

Cada linea contiene

- id = identificador p/ la linea
- runlevel: indica nivel de runlevel
- action: especifica la acción a realizar. Incluyendo initdefault (establecer runlevel), sysinit (p/ proceso de inicialización del sistema), respawn (reinicia el proceso si termina inesperada) y once (ejecuta un proceso a la vez).
- process: nombre del proceso a ejecutar

En caso de querer cambiar el runlevel se usa el comando "init + nro de runlevel" pero no es un cambio permanente y al reiniciar el sist. se volverá al predeterminado.

script RC archivos de scripts usados p/ configurar y arrancar servicios y apps. durante el inicio del sist. c/u el objetivo de automatizar el proceso. y se almacenan en /etc/init.d

El programa "init" gestiona procesos y servicios, usa el archivo "inittab" el real específico de runlevel del sistema y los scripts q' deben ejecutarse p/ c/u runlevel. Los scripts RC tienen un nombre q' comienza c/u una letra "S" o "K" seguido x el nro. y nombre del script. y se ejecutan en un orden específico.

insserv comando q' configura las scripts de inicio del sist. y su orden. Incluye una ejecución eficiente y organizada de los scripts de inicio final administración y control.

Upstart: sistema de inicialización de servicios del sistema q' usa una arquitectura en eventos y tareas p/ gestionar el proceso de arranque.

SystemV vs. Upstart

	arquitectura	inicio paralelo	depurax	registro de eventos	compatibilidad
SystemV	arquitectura basada en scripts ejecutados secuencialmente	inicia los servicios secuencialmente	-	no registra eventos y se basa en scripts p/ ejecutar tareas específicas.	+ compatible c/ distribuciones viejas
Upstart	arquitectura basada en eventos	inicializa los servicios paralelamente	tiene herramientas de depuración y monitoreo	registra eventos y lo usa p/ desencadenar acciones	diseñado p/ ser más moderno y c/ rendimiento.

systemd: sistema inf. monolítico usado como proceso padre p/ iniciar y detener serv. del S.O. Alternativa a System V o Upstart.

activar un socket (systemd): característica q' permite q' los servicios se activen automáticamente cuando se recibe una solicitud de conexión de un socket stdio.
(Un socket es un canal de comunicación entre procesos).

cgroup: funcionalidad q' limita, controla y aísla RR usuarios x procesos o grupos de procesos.

ARCHIVOS

en GNU/Linux los archivos se identifican a través de su nombre y extensión, aunque no es necesario q' siempre tenga la extensión.

permisos: los permisos se dividen en propietario, grupo al q' pertenece y demás usuarios. y pueden leerlo, escribirlo o ejecutarlo (r/W/X)

0 (sin permisos) 1 (permiso ejecutable) 2 (W) 4 (X)

full path name: ruta completa q' especifica la ubicación de un archivo desde el directorio raíz "/"

relative path name: ruta de un archivo especificado en relación al directorio actual

USUARIOS

archivos q' almacenan info. de usuarios

/etc/passwd: info básica como nombre, UID, GID, directorio y shell

/etc/shadow: info sobre contraseñas encriptadas p/ usuarios

/etc/group: info sobre los grupos.

/etc/gshadow: info de contraseñas encriptadas p/ los grupos.

HOJA N°

FECHA

UID (User ID): nro único asignado al usuario p/ identificarlo de manera única

GUID (Group ID): nro p/ identificar el grupo en el sist.
no podrán coexistir 2 UIDs iguales.

usuario root: usuario q' administra el sistema, puede acceder y controlar el sistema y cualquier acción. Puede haber un único usuario de este tipo y tiene UID=0.

PROCESOS

(process identifier) y tamb un PPID (Parent Process Identifier) p/ saber qué proceso lo llamo. Otros atributos q' tienen los procesos son: estado de proceso, uso de CPU, prioridad, memoria usada y tiempo de ejecución.

ejecutar background: se está ejecutando en 2º plano y no interactúa directamente con el usuario. Si agrego & se ejecuta qd este modo.

"foreground": en 1º plano e interactúa mediante la terminal o sea q' espera una accion del usuario.

redirecciones - de entrada: toma de entrada un archivo en vez del teclado.

- de salida estandar: envia la salida de un comando a un archivo mediante >

NOTA