Alberto188
www.wuolah.com/student/Alberto188
190

resument1-3.pdf

PAS-ResumenTeoria

2º Programación y Administración de Sistemas

Grado en Ingeniería Informática

Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba

Tema 1: Introducción a la administración de sistemas

- El departamento de Informática se encarga de mantener y gestionar el Sistema Informático. Funciones:

  -. Administración de Servidores
  -. Administración de la Red
  -. Administración de los Datos
  -. Administración de la Web
  -. Administración de la Seguridad
  -. Desarrollo de un Software Específico

El responsable de informática, hace de enlace entre las necesidades de la empresa y el trabajo que se lleva a cabo en el departamento.

- Un administrador de sistemas es la persona que tiene la responsabilidad de implementar, configurar, mantener, monitorear, documentar y asegurar el correcto funcionamiento de un sistema informático, o algún aspecto de éste.

  -. Estrategia al realizar una tarea:

1. Planear, haciendo un estudio detallado de los pasos que hay que realizar.
2. Hacer los cambios reversibles, haciendo copia de seguridad del sistema o de los ficheros de configuración a modificar.
3. Realizar cambios de manera incremental, probandolos si fuese posible.
4. Probarlo múltiples veces antes de hacerlo público.
5. Conocer realmente cómo trabajan las cosas.

- ¿Qué es software libre y que no lo es?

  -. Software libre: es aquél que concede cuatro libertades a sus usuarios.

1. Libertad 0: La libertad de usar el programa con cualquier propósito.
2. Libertad 1: La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades.
3. Libertad 2: La libertad de distribuir copias.
4. Libertad 3: La libertad de mejorar el programa ,y hacer públicas las mejoras a los demás de modo que toda la comunidad se beneficie.

CONGRESO INTERNACIONAL

DESAFÍOS A LA SEGURIDAD GLOBAL

INTELIGENCIA, TERRORISMO Y AMENAZAS HÍBRIDAS

17 - 18 JUNIO

LUGAR

Centro Superior de Estudios de la **Defensa Nacional CESEDEN**

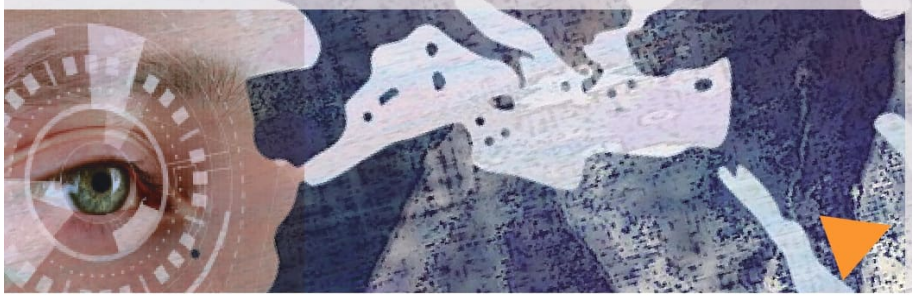
Pº de la Castellana, 61 Madrid

MÁS INFORMACIÓN

eventos@iniseg.es

912 141 926

www.iniseg.es



Organiza:

INISEG

Instituto Internacional de Estudios en Seguridad Global

PEGASO Università Telematica

INSTITUCIONES COLABORADORAS

UNIVERSIDADES QUE PARTICIPAN

OTROS COLABORADORES

Sigue las novedades del congreso **#CIS_INISEG**

-. No es software libre:

1. Software de dominio público: es aquél que no tiene copyright, por qué el autor ha renunciado a ellos.
2. Software semilibre: Es aquel que proporciona las mismas libertades que el software libre, siempre y cuando sea sin ánimo de lucro.
3. Freeware, existen muchos programas que se pueden descargar gratuitamente de Internet, pero no son libres.
4. Shareware, programas que el autor permite usar pero condicionando su uso.
5. Software con fuentes, se entrega el código fuente, pero no permite su modificación y redistribución sin condiciones excesivamente restrictivas.

- Ventajas y desventajas del software libre:

-. Ventajas:

1. Libertad de estudio, uso, redistribución y modificación.
2. Independencia tecnológica al no atarse a ningún proveedor en particular.
3. Ausencia de secretismo tecnológico y de patentes.
4. Fiabilidad y rendimiento.
5. Formatos estándar.
6. Métodos simples y unificados de gestión.
7. Inmensa variedad de soluciones muy maduras.
8. Sistemas potencialmente más seguros.
9. Aspectos económicos.

-. Desventajas:

1. Necesidad de una formación especializada.
2. Ausencia de interfaces visuales, ya que suelen ser privativas (no suelen contar con ella o bien la pierden).
3. No siempre hay para todo tipo de Hardware.
4. Hay un mayor mercado laboral, pero en campos diferentes a la administración de sistemas.

- Superusuario o administrador, es el usuario que tiene todos los privilegios sobre cualquier fichero, instrucción y orden del sistema. Comandos:

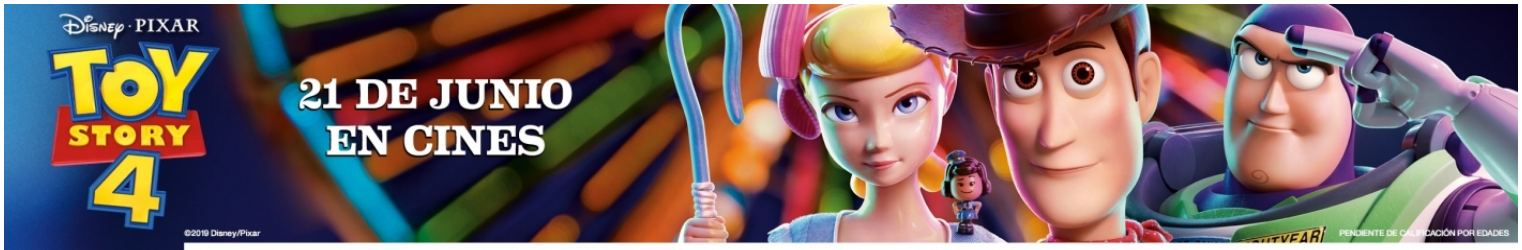
1. su -> nos permite al poner la contraseña, entrar con privilegios de administrador, usando el nombre de *root.*
2. sudo -> permite a otros usuarios ejecutar órdenes como si fuesen administradores.

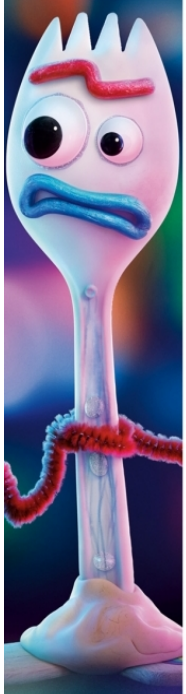
-. Algunos ficheros interesantes:

1. /etc/sudoers -> es el fichero de configuración para usuarios sudo.
2. sudo orden -> pide contraseña del usuario.
3. visudo -> orden para modificar el fichero de configuración */etc/sudoers*.

- Comandos para comunicarse con otros usuarios:

1. write: enviar un mensaje a un usuario.
2. talk: conversar con un usuario.
3. mesg [y/n]: habilitar o deshabilitar la llegada de mensajes al terminal.
4. wall: mandar un mensaje a todo los usuarios del sistema.





Tema 2: Organización de un sistema operativo tipo GNU/ Linux

- En GNU/ Linux todo son ficheros (en caso de no ser un fichero es un proceso), estos tienen una estructura jerárquica de directorios, conocida como sistema de archivos:

1. "/" -> Directorio raíz, puede estar compuesto por varias particiones pertenecientes a varios dispositivos.

- Sistema de ficheros, guarda los ficheros del sistema, no tiene unidades y se organiza de manera jerárquica en directorios.

-. Los nodos-i:

1. Aunque a nivel lógico el sistema de archivos parece un árbol, en realidad los ficheros se almacenan desorganizados por el disco duro. Un fichero puede tener sectores a lo largo de toda la superficie.
2. Los nodos-i, son metadatos (archivos de información) sobre los que nos proporcionan información sobre su tamaño, permisos… Además cada fichero posee un nodo-i.

- Propietarios y permisos:

-. Propietarios: cada fichero tiene dos propietarios: usuario y grupo, y podemos usar dos comandos.

1. chown <usuario> <fichero> -> cambia el usuario propietario del fichero.
2. chgrp <grupo> <fichero> -> cambia el grupo propietario del fichero.

-. Gestión de el acceso a ficheros, se gestiona de la siguiente forma.

1. r -> Ver el contenido del fichero, en un directorio te muestra el contenido.
2. w -> Modificar el contenido del fichero, sirve para crear/eliminar ficheros.
3. x -> Ejecutar el fichero, entrar en el directorio.

-. Es establecen independiente para el usuario propietario (u), usuarios del grupo propietario (g) y resto de usuarios (o). Para establecer permisos para un fichero o directorio, usamos los siguientes comandos:

1. chmod -R u+rwx, go-rwx <directorio>
2. chmod u+x <fichero>

Los permisos base para directorios son 777, para ficheros son 666.

-. Permisos especiales: el comando ‘ls’ representa como una ‘s’ en el sexto bits (s -> g+x , S -> g-x). Para ejecutables se usa para un cambio del dominio a nivel de grupo. Para directorios, cuando se crea un fichero dentro sus permisos de grupo del fichero son iguales a los del directorio. Otro permiso especial seria t (sticky bit), se puede usar para permitir que cualquiera pueda escribir y modificar sobre un fichero o directorio, pero sólo el usuario root pueda eliminarlo o renombrarlo.

- Máscara de permisos: Los permisos se deciden aplicando una máscara de permisos a los permisos base (se consultan/modifican usando el comando ‘umask’). Indica los permisos que están restringidos, indica con 1 aquellos bits que deben ser 0 en la cadena de permisos.

- Tipos de ficheros:

1. Directorio (d): son ficheros que contienen enlaces a otros ficheros.
2. Especial de bloque (b): fichero especial para interactuar con un dispositivo basado en bloques.
3. Especial de carácter (c): fichero especial para interactuar con un dispositivo basado en caracteres.
4. Named Pipes (p): tubería tipo FIFO con nombre.
5. Socket (s): como los pipes pero con comunicación duplex, es decir es capaz de enviar y recibir información de manera simultánea.
6. Enlace físico.
7. Enlace simbólico.

- Enlaces: Archivos especiales que permiten que varios nombres se asocian a un único e idéntico archivo. Ayuda a asegurar la coherencia y ahorrar espacios en el disco. Tipos de enlaces:

1. Enlaces físicos (ln <archivo-real> <enlace-físico>), representar un nombre alternativo para un archivo. Además, si eliminamos el enlace físico no eliminamos el archivo real, a noser que sea el único enlace físico. Esto no se puede realizar con directorios.
2. Enlaces simbólicos (ln -s <archivo-real> <enlace-simbólico>), es un puntero virtual al fichero real. Se crea un fichero de texto que contiene la ruta del archivo al que apunta, si se elimina el fichero simbólico no se elimina el fichero original.

- Procesos: son programas en ejecución. Información básica:

1. PID -> Identificador del proceso.
2. PPID -> Identificador del proceso padre.
3. Nice Number -> Prioridad asignada al ejecutar el proceso.
4. TTY -> Terminal en el que se está ejecutando.
5. RUID -> Identificador del usuario real del que lo ejecutó.
6. EUID -> Identificador del usuario efectivo, si ocurre o se realiza un cambio de dominio se refleja aquí (permiso suid).
7. RGID -> Identificador del grupo real, el grupo de usuarios que lo ejecutó.
8. EGID -> Identificador del grupo efectivo, si hay cambio de dominio se refleja aquí (permiso sgid).

-. Tipos de procesos: para ver los atributos de un proceso usamos 'ps -fl pid'.

1. Interactivos: cuando hay alguien conectado al sistema que los inicia.
2. Encolados: son procesos que se mandan a un buffer para ser ejecutados (en una fecha concreta o cuando la carga del sistema sea baja).
3. Demonios: programas ejecutados en segundo plano durante el arranque, estos esperan de forma continua un determinado evento.

-. Dispositivos, estos se representan y usan como ficheros.

1. Ficheros especiales de caracteres: representan a dispositivos de caracteres (cinta magnetica, puerto paralelo, puerto serie …).
2. Ficheros especiales de bloques: representan a dispositivos en bloques (disquete, partición de un disco duro o un pendrive).
3. Escribir/leer en un dispositivo se convierte en escribir/leer en el fichero correspondiente a dicho dispositivo.

- Tipos de contenido en un directorio:

1. Estáticos: contienen archivos binarios, bibliotecas, documentación y otros ficheros que no cambian sin intervención del administrador. Pueden estar archivos de solo lectura ,y no necesitan copias de seguridad muy a menudo en comparación con los dinámicos.
2. Dinámicos: contienen ficheros que no son estáticos. Se encuentran en dispositivos de lectura-escritura, necesitan que se hagan copias de seguridad a menudo.
3. Compartibles: contiene ficheros que se pueden encontrar en un ordenador y usarse en otro.
4. No compartibles: contiene ficheros que no podemos usar en distintas máquinas.

- Estructura genérica del sistema de ficheros:

1. /bin -> ficheros ejecutables básicos compartidos (mv, cp…).
2. /dev -> ficheros especiales de dispositivos.
3. /etc -> la mayoría de ficheros de configuración locales del sistema (sólo archivos de texto).
4. /root -> directorio 'Home' del administrador.
5. /sbin -> ficheros ejecutables que normalmente sólo el administrador puede ejecutar.
6. /home -> los directorios del trabajo de los usuarios.
7. / lost+found -> contiene "referencias" a los ficheros marcados como erróneos al chequear el sistema de ficheros.
8. /lib -> librerías necesarias para ejecutar los archivos.
9. /proc y /sys -> sistema de ficheros virtuales, contienen información sobre procesos, núcleo, módulos cargados, dispositivos, sucesos ….
10. /tmp -> ficheros temporales tienen permiso 't' activo.
11. /var -> ficheros variable (colas de datos de impresión…).
12. /boot -> núcleo y ficheros necesarios para cargar el núcleo y ficheros de configuración del gestor de arranque.
13. /mnt, /mount o /media -> montaje de otros sistemas de ficheros, disquetes, cdroms….
14. /upt -> paquetes de aplicación estáticas, estas no son actualizables.

-. usr: contiene subdirectorios de solo lectura, no deben ser específicos de la máquina que los usa.

1. /usr/bin -> ficheros ejecutables por todos los usuarios.
2. /usr/sbin -> ficheros ejecutables de administración.
3. /usr/include -> ficheros cabecera, de cabecera estándar para compilación.
4. /usr/lib -> librerías binarias.
5. /usr/local -> software local específico.
6. /usr/share -> datos compartidos (independencia de la arquitectura: imágenes, ficheros de texto…).
7. /usr/src -> código fuente, como el del Kernel

-. Comandos ordenados según el tipo de contenido de un fichero:

1. Estáticos: /bin, /sbin, /opt, /boot, /usr….
2. Dinámicos: /var/mail, /var/spool, /var/run, /var/lock, /home….
3. Compartibles: /usr/bin, /opt….
4. No compartibles: /etc, /boot, /var/run, /var/lock….

Tema 3: Arranque y parada del sistema

- Iniciador de ROM:

-. Al arrancar el ordenador se produce una señal eléctrica (RESET), iniciando todos los registros a valores por defecto. La memoria ROM contiene el software de configuración del hardware del sistema. Esta tiene 3 funciones que cumplir:

1. Comprobar el sistema, detectando características y comprobando su funcionamiento.
2. Leer y almacenar en memoria el programa cargador del SO.
3. Pasar el control al cargador del SO, saltando a la dirección de memoria donde lo ha almacenado. Se encuentra en los primeros sectores del disco y tiene un tamaña prefijado, el sector prefijado se conoce como Master Boot Record.

- Núcleo del SO: Realiza una comprobación del hardware del sistema. Se prepara a sí mismo para ejecutar el sistema inicializando sus tablas internas, creando estructuras de datos necesarias, etc…. Crea un proceso Init y le pasa el control.

El núcleo de Linux es cargado inicialmente en memoria, y permanece de manera residente durante el funcionamiento del sistema, controlando la ejecución del resto del software.

- Initrd: las características del arranque pueden implicar que el medio desde el que se carga el núcleo provenga de un sistema de ficheros concreto o incluso desde la red. Se necesitarán módulos específicos, alojados en initrd, el programa cargador le dice la posición del initrd. Funcionamiento:

1. El núcleo carga primero initrd.
2. Utilizando Initrd, se cargan los módulos necesarios.
3. El núcleo de manera seguida continuará con el proceso de arranque.

- Init: termina el proceso de arranque, dejando el sistema en modo multiusuario. Con ello está preparado para que los usuarios trabajen en él. Usa una serie de scripts que lo indican las acciones a realizar. Tareas realizadas por el proceso Init:

1. Chequea los sistemas de ficheros.
2. Monta los sistemas de ficheros permanentes.
3. Activa las áreas de swapping o intercambio.
4. Activa los demonios y a red.
5. Limpia los sistemas de ficheros (borra los directorios temporales).
6. Habilita el login a los usuarios del sistema.

- Grub (Grand Unified Bootloader): se instala en el Master Boot Record (MBR) y hace las funciones de Master Boot Program (MBP, Programa cargador). Pregunta que SO arranca:

1. Si es Linux -> Carga el núcleo solicitado y le pasa el control para que el arranque continúe.
2. Si es Windows -> pasa el control de Windows que realiza su arranque.

-. Con Grub 2.0: tenemos un archivo fundamental de configuración: /boot/grub/grub.c
Es archivo se genera a partir de del comando sudo update -grub2, utilizando todos los scripts incluidos en la carpeta /etc/grub.d . Los contenidos de la carpeta son los siguientes:

1. .../00_header: Cabeceras que no se suelen modificar.
2. .../05_debian_theme: Aspecto visual del menú.
3. .../10_linux: Este archivo contiene comandos y scripts que se encargan del Kernel de Linux de la partición principal.
4. .../20_*: Aplicaciones de third party, son aplicaciones creadas libremente para cualquier tipo de plataforma.
5. .../30_OS_proper: este archivo contiene comandos y scripts que se encargan del Kernel de Linux.

Son por tanto cuatro secciones, los cambios que realicemos en una sección no repercuten en otra.

-. Fichero /etc/default /grub:

1. GRUB_DEFAULT=0: es la entrada por defecto para el arranque, si ponemos saved será seleccionada por el administrador.
2. GRUB_SAVEDEFAULT=true: la entrada por defecto es siempre la última seleccionada.
3. GRUB_HIDDEN_TIMEOUT=0:
   a) Muestra una pantalla en negro o una imagen, durante el número de segundos indicado, antes del menú de arranque.
   b) Suele no usarse cuando hay múltiples sistemas.
   c) Es 0 cuando solo hay Linux.

4. GRUB_HIDDEN_TIMEOUT_QUIET=true: sin cuenta atrás.
5. GRUB_TIMEOUT_= 10: Número de segundos hasta seleccionar la entrada por defecto.
6. GRUB_CMDLINE_LINUX=”opciones”: pasar opciones de arranque al Kernel de Linux (modo normal o modo recuperacion).
7. GRUB_CMDLINE_LINUX_DEFAULT=”quiet splash”: pasar opciones de arranque al Kernel Linux (modo normal).
8. GRUB_TERMINAL=console: desactiva el modo gráfico.
9. GRUB_BADRAM=”...”: Deshabilita el uso de algunas direcciones de memoria.

10. GRUB_DISTRIBUTOR='sb_release -i -s 2> /dev/hull || echo Debian': obtener el nombre de la distribución.
11. GRUB_DISABLE_UUID="true": no utiliza el UUID del dispositivo raíz.
12. GRUB_GFXMODE=640x480: selecciona manualmente la resolución del menú.
13. GRUB_INIT_TUNE= " 480 440 1 ": hacer beep antes del menú de inicio.
14. GRUB_BACKGROUND: imagen de fondo.

Para reinstalar GRUB se usa: sudo grub_install /dev/sda.

-. GRUB permite durante la selección del sistema operativo:

1. Editar las entradas, al no ser los cambios permanentes, sirve para probar.
2. Consola Interactiva de GRUB, pulsar la tecla c. Permite ejecutar comandos para arreglar el arranque (seleccionar otro initrd, cargar módulos…).
3. Terminología de GRUB, enumerando los dispositivos según reconozca la BIOS empezando en cero.

- Modo monousuario: estado del sistema definido para realizar tareas administrativas y de mantenimiento que requieren un control completo y no compartido. Sólo realiza el montaje del sistema de ficheros raíz (/), los otros sistemas de ficheros están disponibles pero no están montados. Se puede acceder a todo el sistema ,pero:

1. Muy pocos demonios están en ejecución solo los necesarios.
2. Muchas utilidades no están activas.
3. Solo las órdenes del sistema de ficheros están disponibles, aunque si /usr esta en otra partición, no está montado.

Para entrar en modo monousuario el proceso Init crea el shell por defecto (/bin/sh) como usuario root. Antes ejecuta la orden /sbin/sulogin, que pide la contraseña de root para dejar entrar al sistema.

- Pasos para el proceso de arranque:

1. Chequea el sistema de ficheros raíz con fsck.
2. Monta el sistema de ficheros raíz en modo lectura-escritura.
3. Chequea el resto de sistema de ficheros con fsck.
4. Monta el resto de sistemas de ficheros.
5. Activa las particiones de intercambio: swapon -a.
6. Activa las cuotas de disco: quotacheck -a y quotaon -a.
7. Lanza los procesos servidores o demonios: crond, atd, cupsd, syslogd…
8. Activa la red.
9. Lanza los demonios de red: xinetd, apache2, nagiosd, sshd…
10. Limpia los sistemas de ficheros: /tmp,etc.
11. Permite a los usuarios entrar.

-. Niveles de ejecución en GNU/Linux: El sistema operativo puede estar en distintos niveles de ejecución. Los niveles son:

1. Nivel 0: Sistema apagado.
2. Nivel 1: Modo monousuario, rescue o troubleshooting.
3. Nivel 2: Modo multiusuario sin funciones red.
4. Nivel 3: Modo multiusuario con funciones red y terminales de texto.
5. Nivel 4: Sin uso, se puede definir por el administrador.
6. Nivel 5: Modo multiusuario con funciones red e inicio de sesión gráfica.
7. Nivel 6: Sistema reiniciando.

Algunos subdirectorios son:

1. /sbin/runlevel: permite saber en qué nivel está el sistema.
2. /sbin/telinit: cambia de nivel de ejecución:

a) telinit 1 -> modo monousuario.
b) telinit 6 -> reinicia el sistema.
c) telinit 3 -> cambiar al nivel 3.

Al arrancar mediante GRUB, al núcleo se le puede pasar como parámetro un número indicando en el nivel en el que queremos arrancar.

- Ficheros de inicialización: se pueden personalizar los niveles de ejecución con las carpetas '/etc/rc?.d/ ', donde ? es el nivel de ejecución, todos ellos son ejecutados por Init durante el arranque.

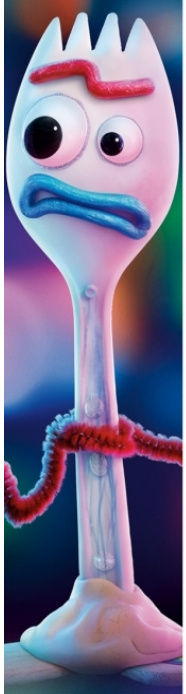
  Se ejecutan al arrancar o cambiar de nivel. El nombre del script empieza por s o k, seguido de dos dígitos y un nombre descriptivo:

  k35mb k15httpd s40atd …

  Lo ejecuta en orden alfabético, primero los k despues los s, y los dígitos establecen el orden entre todos los k y todos los s.

  k -> Detener los demonios o matar procesos.
  s -> lanzar demonios o ejecutar funciones de inicio.

  -. Los scripts reciben varios parámetros: start, stop, restart… Esto nos permite lanzar o relanzar demonios sin reiniciar el sistema. El comando 'rc' ejecuta los ficheros k con el parámetro stop y los s con el parámetro start. Estos scripts se mantienen con retrocompatibilidad, se tiende a usar upstart y el comando service: muchos de los scripts simplemente llaman a este (upstart).



- upstart: proceso de arranque/parada del sistema basado en eventos, reemplazo del clásico Init (aunque los ficheros siguen denominándose Init). Este convive con los sysv scripts. Realiza las siguientes tareas de manera asíncrona:

1. Dirige el inicio de las tareas y demonios.
2. Controla los demonios mientras el sistema está encendido.
3. Detiene los demonios durante el proceso de apagado.

En el directorio /etc/init hay una serie de ficheros de configuración de eventos (evento conf), que Init ejecuta según la orden de las dependencias establecidas en los mismos. Los eventos indican qué tarea ejecutar, cuándo y cómo, mediante su propio lenguaje.

-. Initctl: permite al administrador interactuar con init, para decirle que realice determinadas acciones:

start evento stop evento status evento

- Ficheros de configuración de eventos (.conf):

1. start on <event> -> describe en qué condiciones se lanzará este evento.
2. stop on <event> -> describe en qué condiciones se parará el evento.
3. respawn -> volver a lanzar ese proceso o demonio cuando se pare.
4. console -> hacía donde dirigir la salida del evento.
5. pre-start -> ejecuta la orden/guión shell antes de lanzar el proceso.
6. pre-stop -> ejecuta la orden/guión shell antes de parar al proceso.
7. post-start -> ejecuta la orden/guión shell después de lanzar ese proceso.
8. post-stop -> ejecuta la orden/guión shell después de parar ese proceso.

- Systemd: es un reemplazo del proceso Init. Amplía las funcionalidades, pudiendo gestionar cosas que Init no gestiona, como por ejemplo un sistema de logs. Hay cierta controversia debido a que…

1. Es un sistema muy complejo, que causa algunas dependencias innecesarias.
2. Se gestiona mediante unidades (servicios) y targets (algo similar a los niveles de ejecución).
3. Es compatible hacia atrás con los scripts Sysv.

- Parada del sistema: En ocasiones es necesario apagar o reiniciar el sistema: mantenimiento, diagnóstico, hardware nuevo, etc. Acciones durante el proceso de parada:

1. Se notifica a los usuarios.
2. Procesos en ejecución -> envían una señal de terminación (TERM).
3. Se paran los demonios.
4. A los usuarios que quedan conectados se les echa del sistema.
5. Los procesos que quedan en ejecución -> envían una señal de fin (KILL).
6. Actualizaciones del disco pendiente (integridad del sistema de ficheros) con sync.

-. Shutdown [opciones] tiempo [mensaje]

1. Sin opciones: modo monousuario (telinit 1).
2. -r: reiniciar (telinit 1).
3. -h: parar (telinit 0).
4. -c: cancelar.
5. -k: hacer una simulación de apagado.
6. tiempo: +minutos,now.

Al volver al modo monousuario, vuelve al nivel por defecto.

-. Posibles causas de las caídas del sistema:

1. Fallos hardware.
2. Errores de hardware irrecuperables.
3. Fallos de luz.
4. Problemas de Entrada/Salida.
5. Problemas de algún sistema de ficheros.

-. Problemas de arranque:

1. Fallos de Hardware o bien hardware incompatible.
2. No se puede leer el sistema de ficheros de los discos de trabajo.
3. Hay áreas dañadas en el disco que no pertenecen al sistema de ficheros.
4. Errores de la configuración del sistema.

-. Caídas del sistema y problemas de arranque:

1. Al re-arrancar es bueno mirar los mensajes que hay en el fichero en /var/log/messages.
2. La orden dmesg -> mensajes producidos durante el arranque.
3. En el arranque al núcleo se le pueden pasar otros parámetros.