Conceptos de Sistemas Operativos
Practica 2

Ramiro Cabral

September 3, 2023

1 SystemV

1.1 Proceso de Inicio en GNU/Linux

1. Se comienza a ejecutar el codigo del Bios.
2. El BIOS ejecuta el POST.
3. El bios lee el sector de arranque (MBR).
4. Se carga el gestor de arranque (MBC).
5. El bootloader carga el kernel y el initrd.
6. Se monta el initrd como sistema de archivos raiz y se inicializan componentes esenciales.
7. El kernel ejecuta el proceso init y se desmonta el initrd.
8. Se lee el /etc/inittab.
9. Se ejecutan los scripts apuntados pr el runlevel 1.
10. El final del runlevel 1 le indica que vaya al runlevel por defecto.
11. Se ejecutan los scipts apuntados por el runlevel por defetco.
12. El sistema esta listo para usarse.

1.2 Init

initrd :Es cargado como parte del procedimiento de booteo del kernel. Contiene un set minimio de ejecutables y archivos necesarios para la inicializacion del kernel del sistema.

- Carga los subprocesos necesarios para el correcto funcionamiento del SO.
- Posee el PID 1 y se encuentra usualmente en /sbin/init.
- En SystemV se lo configura a traves del archivo /etc/inittab.
- No tiene padre y es el padre de todos los procesos (pstree).
- Es el encargado de montar los filesystems y de hacer disponible los demas dipositivos.

1.3 RunLevels

- Hacen referencia al modo en que arranca Linux.
- Define que servicios del sistema estan operando.
- El proceso de arranque es dividido en diferentes niveles.
- Se encuentran definidos en */etc/inittab*.
- Syntaxis: ***id:nivelesEjecucion:accion:proceso***.
  - **ID:** identifica la entrada en inittab.
  - **nivelesEjecucion:** el/los niveles de ejecucion en los que se realiza la accion.
  - **Accion:** describe la accion a realizar.
  - **Proceso:** proceso exacto que sera ejecutado.
- Existen 7 distintos RunLevels, y cada uno puede iniciar un conjunto de procesos al arranque o apagado del sistema.
- Los scripts que se ejecutan se encuentran en */etc/init.d*.
- En */etc/rcX.d* (con X entre 0 y 6) hay links a los archivos del */etc/init.d*.
- Formato de los links: `[S-K]<orden><nombreScript>`.
- Para administrar el orden de los enlaces simbolicos, y resolver dependencias entre ellos se utiliza **insserv**.
- insserv usa cabeceras (De tipo LSB) en los scripts del init.d para especificar la relacion con otros scripts rc.

2 Upstart

- Reemplazo propuesto para SystemV.
- Ejecucion de trabajos en forma asincronica a traves de eventos, a diferencia de SystemV que es estrictamente sincronico (*dependency-based*).
- Estos trabajos se denominan ***jobs***.
- Los jobs definen servicios o tareas a ser ejecutadas por init.
- Son definidos en */etc/init*.

- Los jobs son ejecutados ante eventos, y es posible crear nuestros propios eventos.
- Cada job posee un objetivo (goal start/stop) y un estado (state).

3 SystemD

- Sistema creado por RedHat (the devil himself) que centraliza la administracionde demons y librerias del sistema.
- Mejora el paralelismo de booteo.
- Puede ser controlado por el comando ***systemctl***.
- Compatible con SystemV.
- Los runlevels son reemplazados por ***targets***.
- Las unidades de trabajo son denominadas ***units*** de tipo:
  - ***Service:*** controla un servicio particular.
  - ***Socket:*** encapsula, un IPC, un socket del sistema o filesystem.
  - ***Target:*** agrupa units o establece puntos de sincronizacion.
  - ***Snapshot:*** almacena el estado de un conjunto de unidades que puede ser restablecido mas tarde.

3.1 Activacion por Socket

- Es un mecanismo de iniciacion bajo demanda.
- Cuando el socket recibe una conexion spawnea el servicio y le pasa el socket.
- No hay necesidad de definir dependencias entre servicios, ya que se inician todos los sockets en primier medida.

3.2 cgroups

- Permite organizar un grupo de procesos en forma jerarquica.
- Agrupa conjuntos de procesos realcionados.

3.3 fstab

- Define que particiones se montan al arranque.
- Su configuracion se encuentra en ***/etc/fstab***.

4 Usuarios en GNU/Linux

- Cada usuario debe poseer credenciales para acceder al sistema.
  - *root:* administrador del sistema (superuser).
  - *otros:* usuarios del sistema estandar (*/etc/sudoers*).
- Archivos de configuracion:
  - ***/etc/passwd***
    - Cada linea representa a un usuario.
    - Posee informacion general del usuario (grupo,nombre,descripcion,homedir,shell,etc).
  - ***/etc/group***
    - Muestra cada grupo del sistema.
    - Cada usuario puede tener grupos secundarios.
  - ***/etc/shadow***
    - Posee nombres y contrasenas encriptadas (solo accesible por el root).

4.1 UID y GID

- Un **UID** (User identifier) es un numero asignado por Linux a cada usuario del sistema. Este numero es usado para identificar el usuario y para determinar a que recursos puede acceder.
- Son almacenados en el archivo */etc/passwd*.
- El Root posee el UID 0.
- Las UID por debajo de 1000,por convencion, son reservadas para usuarios creados por el sistema, servicios y otras cuentas especiales.
- Al crear un nuevo usuario, se le asignara un UID mayor a 1000.
- Un **GID** (group identifier) es un numero asignado a cada grupo del sistema.

- Son almacenados en el archivo */etc/groups*.
- Los primeros 100 GIDs son reservados para el uso del sistema.
- El GID 0 corresponde al grupoo root.
- El GID 100 correponde al gruop 'users'.

5 Procesos

- Cuando un programa es ejecutado, el sistema le provee una instancia especial al proces. La instancia consiste en todos los servicios/recursos que podria utilizar el proceso en la ejecucion.
- Un numero de ID de 5 digitos es asignado al proceso, este numero es denominado **PID**. Cada proceso posee un unico PID.
- Un processo que se conecta a la terminal es llamado un **foreground job**. Es llamado asi ya que puede comunicarse con el usuario mediante la pantalla y el teclado.
- Un proceso que se desconecta de la terminal y no puede comunicarse con el usuario es llamado un **background job**.
- Si un proceso en background requiere inhteraccion con el usuario, el mismo se detiene y espera hasta establecer una conexion con la temrinal.