

Resumen Linux Intermedio

1. Programación de Tareas en Linux

1.1 Programación de tareas con at (en momentos específicos)

El comando **at** permite programar tareas para ejecutarse una sola vez en un momento determinado.

- **Comandos principales:**
 - **at [opciones] TIME** - Programa una tarea para ejecutarse a una hora específica
 - **at TIME MMDDAA** - Programa una tarea para una fecha específica
 - **atq** - Muestra las tareas programadas pendientes
 - **at -c n°_trabajo** - Muestra el contenido de una tarea programada
 - **atrm n°_trabajo** - Elimina una tarea programada
- **Formatos de tiempo aceptados:**
 - **at 20:15** - Hora específica
 - **at 10:00 06/08/10** - Fecha y hora específicas
 - **at now +3 minutes** - Dentro de 3 minutos
 - **at now +4 days** - Dentro de 4 días
 - **at 4pm +2 days** - A las 4 PM dentro de 2 días
 - **at 10am tomorrow** - A las 10 AM mañana
- **Configuración:**
 - El archivo **/etc/at.deny** contiene una lista de usuarios que no pueden programar tareas

1.2 Programación de tareas periódicas con Cron

El demonio cron comprueba continuamente si hay tareas pendientes para ejecutarlas según lo programado.

- **Comandos principales:**
 - **crontab -e** - Edita las tareas programadas
 - **crontab archivo** - Carga tareas desde un archivo
 - **crontab -l** - Lista las tareas programadas
 - **crontab -r** - Elimina todas las tareas programadas
 - **crontab -l -u usuario** - Ve las tareas de un usuario específico (solo root)
 - **crontab -r -u usuario** - Borra las tareas de un usuario específico (solo root)
- **Formato de archivo crontab:**
 - minuto hora dia_mes mes dia_semana comando
- **Valores especiales:**
 - **@reboot** - Al iniciar el sistema
 - **@yearly** - Anualmente (0 0 1 1 *)
 - **@monthly** - Mensualmente (0 0 1 * *)
 - **@weekly** - Semanalmente (0 0 * * 0)
 - **@daily** - Diariamente (0 0 * * *)
 - **@hourly** - Cada hora (0 * * * *)

- **Símbolos especiales:**
 - * → Cualquier valor
 - , → Lista de valores (1,3,5)
 - - → Rango de valores (1-5)
 - / → Incrementos (* / 5 = cada 5 unidades)
- El archivo que se modifica es **/etc/crontab**.
- Los archivos crontab se guardan en el directorio **/var/spool/cron/crontabs**.

1.3 Tareas Periódicas como Root

El administrador del sistema puede programar tareas en /etc/crontab con formato especial que incluye el usuario:

- **minuto hora día_mes mes día_semana usuario comando**
- También puede usar directorios especiales:
 - **/etc/cron.hourly** - Scripts ejecutados cada hora
 - **/etc/cron.daily** - Scripts ejecutados diariamente
 - **/etc/cron.weekly** - Scripts ejecutados semanalmente
 - **/etc/cron.monthly** - Scripts ejecutados mensualmente

1.4 Anacron para sistemas que no están siempre encendidos

Anacron permite ejecutar tareas periódicas cuando el sistema se reinicia, si no se pudieron ejecutar porque estaba apagado.

- **Características:**
 - No puede programar tareas en intervalos menores a un día
 - Se ejecuta al iniciarse el sistema y diariamente a las 7:30
 - Configuración en /etc/anacrontab
- **Formato del archivo anacrontab:**
 - periodo retraso identificador comando
 - **periodo:** días entre ejecuciones
 - **retraso:** minutos a esperar tras el inicio del sistema
 - **identificador:** nombre único de la tarea
 - **comando:** comando a ejecutar
- El archivo que se modifica es **/etc/anacrontab**.
- Anacron guarda información sobre las ejecuciones en **/var/spool/anacron**.

CRON	ANACRON
Es un demonio (daemon)	No es un demonio (daemon)
Nos permite ejecutar tareas cada minuto	Solo permite ejecutar tareas diarias
Ideal para servidores 24/7	Apropiado para escritorios o portátiles
Usuarios Root o usuarios normales	Solo usuarios Root
No ejecuta al apagarse el equipo	Se lanza cuando se enciende el equipo

2. Gestión de Usuarios y Cuentas

2.1 Información de cuentas

- Los usuarios se caracterizan por username, UID (identificador de usuario) y GIDs (grupos)
- Archivos principales
 - **/etc/passwd**: Contiene información de usuarios → (nombre:password:uid:gid:gecos:home)
 - **/etc/shadow**: Contraseñas encriptadas y caducidad → (nombre:password:changed:min:max:warn:inactive)

2.2 Comandos para ver información

- **getent passwd usuario**: Muestra info del usuario
- **getent shadow usuario**: Muestra info de contraseña y caducidad
- **chage -l usuario**: Muestra información de caducidad
- **w**: Muestra usuarios conectados con tiempo de inactividad
- **who**: Usuarios conectados (menos detalle)
- **users**: Listado simple de usuarios conectados
- **lastlog**: Última vez que se conectaron los usuarios
- **lastb**: Intentos fallidos de login
- **pkill -9 -u usuario**: Desconecta a un usuario

2.3 Creación y Modificación de Usuarios

- **Creación de usuarios**
 - **adduser nombre**: Forma interactiva de crear usuarios
 - **useradd [opciones] nombre**: No interactiva, ideal para creación masiva
- **Ficheros importantes**
 - **/etc/default/useradd**: Valores por defecto
 - **/etc/login.defs**: Configuraciones predeterminadas
 - **/etc/skel**: Archivos que se copian al home del nuevo usuario
- **Modificación de usuarios**
 - **passwd usuario**: Cambiar contraseña y ver estado de cuenta
 - **passwd -S usuario**: Ver estado de la cuenta
 - **usermod [opciones] usuario**: Modificar cuenta
 - **-L**: Bloquear cuenta
 - **-U**: Desbloquear cuenta
 - **-l nuevo_nombre viejo_nombre**: Cambiar nombre de usuario
 - **chage [opciones] usuario**: Modificar caducidad
 - **-d 0**: Obliga a cambiar contraseña al loguear
 - **-E fecha**: Establece expiración (formato YYYY-MM-DD)
 - **-E -1**: Desactiva la expiración
- **Eliminación de usuarios**
 - **userdel [opciones] usuario**: Elimina cuenta
 - **-r**: Elimina también el directorio home

2.4 Gestión de Grupos

2.4.1 Información de grupos

- **/etc/group**: Contiene grupos del sistema (grupo:x:gid:usuarios)
- **/etc/gshadow**: Contraseñas de grupos (grupo:contraseña:administrador:usuarios)
- **groups usuario**: Muestra grupos a los que pertenece un usuario

2.4.2 Comandos para gestión de grupos

- **addgroup grupo o groupadd grupo**: Crear grupo
- **groupmod grupo**: Modificar grupo
- **groupdel grupo**: Eliminar grupo
- **adduser usuario grupo**: Añadir usuario a grupo
- **deluser usuario grupo**: Quitar usuario de grupo
- **newgrp grupo**: Cambiar grupo principal del usuario actual
- **chown usuario:grupo archivo**: Cambiar dueño y grupo de un archivo
 - **-R**: Aplicar recursivamente a un directorio

3. Gestión con Sudo

3.1 Configuración de sudo

- **/etc/sudoers**: Archivo de configuración (editar con visudo -f /etc/sudoers)
- **visudo -c**: Comprobar sintaxis
- Formato de reglas: **usuario equipo = (usuario:grupo) comando.**

3.2 Opciones de sudo

- **sudo -l**: Listar permisos de sudo del usuario actual
- Definir alias de comandos: **Cmnd_Alias NOMBRE=/ruta/comando1, /ruta/comando2.**
- Opciones especiales:
 - **NOPASSWD**: No pedir contraseña
 - **Defaults passwd_tries=N**: Limitar intentos de contraseña
 - **!comando**: Denegar un comando específico

4. Creación y Eliminación Masiva

4.1 Creación masiva

- Usar **useradd** con parámetros:
 - **-g grupo**: Establecer grupo principal
 - **-d /home/ruta**: Establecer directorio home
 - **-m**: Crear directorio home
 - **-s /bin/bash**: Establecer shell
 - **-p \$(mkpasswd -m sha-512 contraseña)**: Establecer contraseña encriptada
- **Automatización**
 - Usar scripts y archivos CSV para crear/eliminar usuarios masivamente
 - Usar **mkpasswd** para encriptar contraseñas (requiere paquete **whois**)
 - **mkpasswd -m sha-512 contraseña**

5. Seguridad y Caducidad de Cuentas

- **Contraseñas y expiración**
 - **chage -d 0 usuario:** Forzar cambio de contraseña en próximo inicio.
 - **chage -E fecha usuario:** Establecer caducidad (formato YYYY-MM-DD).
 - **chage -M días usuario:** Establecer máximo días para cambiar la contraseña.
 - **chage -m días usuario:** Establecer mínimo días para poder cambiar contraseña.
 - **chage -W días usuario:** Establecer días de aviso antes de expiración.
 - **chage -I días usuario:** Establecer días de inactividad antes de bloqueo.
- **Bloqueo y desbloqueo**
 - **usermod -L usuario:** Bloquear cuenta (añade "!" al principio de la contraseña en /etc/shadow)
 - **usermod -U usuario:** Desbloquear cuenta.
 - **passwd -l usuario:** Alternativa para bloquear cuenta.
 - **passwd -u usuario:** Alternativa para desbloquear cuenta.

6. Configuración de sudo Avanzada

6.1 Alias en sudoers

- **User_Alias:** Alias para usuarios.
- **Host_Alias:** Alias para equipos.
- **Cmnd_Alias:** Alias para comandos.
- **Runas_Alias:** Alias para ejecutar como otro usuario.

6.2 Ejemplos prácticos

- **Permitir a grupo ejecutar comandos de red:**
 - Cmnd_Alias RED=/bin/ip, /bin/systemctl networking
%tecnicos ALL=RED
- **Permitir editar archivos específicos:**
 - Cmnd_Alias RED=/bin/ip, /bin/systemctl, /bin/nano /etc/network/interfaces
- **Sin contraseña:**
 - %tecnicos ALL=NOPASSWD:RED
- **Limitar intentos:**
 - Defaults:ALL passwd_tries=2

7. Scripts para Administración de Usuarios

7.1 Script de creación masiva

```
#!/bin/bash
#parametros
nombreUsuario=$1
numeroUsuario=$2
#Inicio
for i in $(seq 1 $numeroUsuario)
do
    usuario="${nombreUsuario}${i}"
    contrasenya="${nombreUsuario}${i}"
    encriptada=$(mkpasswd -m sha-512 "$contrasenya")
    useradd -m -s /bin/bash -p "$encriptada" "$usuario"
    echo "Se ha creado el usuario $usuario con exito"
done
```

7.2 Script de eliminación masiva

```
#!/bin/bash
#Parametros
nombreUsuario=$1
numeroUsuario=$2
#Inicio
for i in $(seq 1 $numeroUsuario)
do
    usuario="${nombreUsuario}${i}"
    userdel -r "$usuario"
    echo "Usuario $usuario eliminado con exito"
done
```

7.3 Script para detectar usuarios bloqueados

```
#!/bin/bash
# UsuariosBloqueados.sh
# Buscar usuarios con UID entre 1000 y 2000
for usuario in $(awk -F: '$3 >= 1000 && $3 <= 2000 {print $1}' /etc/passwd)
do
    # Verificar si la cuenta está bloqueada
    estado=$(passwd -S "$usuario" | awk '{print $2}')
    if [ "$estado" = "L" ]; then
        echo "Usuario bloqueado: $usuario"
    fi
done
```

8. Buenas Prácticas en Gestión de Usuarios

8.1 Seguridad

- Establecer políticas de contraseñas robustas en **/etc/login.defs**.
- Configurar caducidad de contraseñas para todos los usuarios.
- Usar **TMOUT=1800** para cerrar sesiones inactivas automáticamente.
- Revisar regularmente los registros de autenticación con **lastlog** y **lastb**.

8.2 Automatización

- Crear plantillas en **/etc/skel** para usuarios nuevos
- Utilizar scripts para tareas repetitivas
- Usar grupos para gestionar permisos en vez de usuarios individuales
- Documentar los alias de sudo y grupos especiales

8.3 Monitoreo

- Comprobar regularmente cuentas y permisos:
 - # Encontrar usuarios con UID 0 (root)
 - `awk -F: '$3 == 0 {print $1}' /etc/passwd`
 - # Encontrar usuarios sin contraseña
 - `awk -F: '$2 == "" {print $1}' /etc/shadow`
- Implementar sistemas de auditoría como `auditd` para monitorear cambios en usuarios y grupos

9. Configuraciones Útiles para Entornos Multi-usuario

9.1 Directorios compartidos

- # Crear directorio compartido con sticky bit
- `mkdir -p /mnt/compartido`
- `chmod 1777 /mnt/compartido`

9.2 Script para la expiración de usuarios

```
#!/bin/bash
# Modificar CreacionUsuarios.sh para fecha de expiración
while IFS=, read -r username password
do
    # Crear usuario con contraseña encriptada
    useradd -m -s /bin/bash -p $(mkpasswd -m sha-512 "$password") "$username"
    # Establecer fecha de expiración
    chage -E "2022-06-30" "$username"
    echo "Usuario $username creado correctamente con expiración 30/06/2025"
done < /root/usuarios.csv
```