

# LINUX

Quinta Parte

Abril 2023

# Tipo de Comando: Sesión de trabajo

Comando: `passwd`

Propósito: cambia contraseña

Sintaxis: `passwd [-opciones]`

La introducción de `passwd` sin opciones, pide la clave actual y posteriormente la nueva clave dos veces



## Tipo de Comando: Sesión de trabajo

Comando: ps

Propósito: Lista y maneja los procesos que estén activos al momento de introducir este comando

Sintaxis: ps [-opciones]

ps sin opciones lista los procesos del usuario con los estemos logueados.

Posibilidades de uso:

- a Lista los procesos de todos los usuarios.
- x Lista procesos de todas las terminales y usuarios.
- u Lista información del proceso: usuario que lo corre, y parámetros de la arquitectura: cpu, memoria



# Tipo de Comando: Almacenamiento de información

Comando: tar

Propósito: Permite empaquetar varios archivos en uno solo, sin comprimirlo y recuperarlos. También puede descomprimir. Permite hacer copias de seguridad

Sintaxis para comprimir es:

`tar -cvf archivo.tar`

-c indica a tar que cree el archivo

-v indica a tar que muestre lo que va empaquetando

-f indica a tar que el siguiente argumento es el nombre del fichero.tar



# Tipo de Comando: Sesión de trabajo

Comando: tar

(continuación)

Para desempaquetar los ficheros.tar, hacemos:

```
tar -zxvf archivo.tar
```

Extraerá componentes de un archivo comprimido



# Tipo de Comando: Ordenamiento de contenido

Comando: sort

Propósito: Permite ordenar líneas de archivos de texto.

Por defecto escribe en la salida estándar

Sintaxis: sort nombreArchivo

Posibilidades de uso:

- r ordenamiento inverso
- g ordenamiento numérico
- f ignora diferencia entre mayúsculas y minúsculas



# Tipo de Comando: Diferencia de contenidos

Comando: diff

Propósito: se utiliza para encontrar diferencias entre dos archivos

Sintaxis: diff [opciones] de-archivo a-archivo

Posibilidades de uso:

- a trata todos los archivos como texto y los compara línea a línea
- b ignora cambios en la cantidad de espacios
- i considera equivalentes mayúsculas y minúsculas



## Tipo de Comando: ayuda

Comando: man

Propósito: man es la abreviatura de manual, proporciona ayuda sobre el comando solicitado o permite a los usuarios buscar comandos relacionados con una determinada palabra clave.

Sintaxis: man nombre-comando [opciones]

Posibilidades de uso:

- a muestra un mensaje de ayuda de una línea y sale
- k busca las palabras claves en manuales disponibles





# Tipo de Comando: ayuda

Comando: `whatis`

Propósito: sirve para visualizar descripciones rápidas de palabras claves relacionadas con el Shell, comandos, etc

Sintaxis: `whatis cp`



## Tipo de Comando: ayuda

Comando: whereis

Propósito: permite conocer la ubicación de un comando y su página de manual

Sintaxis: whereis comando

por ejemplo:

```
$whereis passwd
```

```
passwd /usr/bin/passwd/etc/passwd /usr/man/  
passwd.5.gz
```

En el ejemplo se piden todos los archivos que tengan que ver con el comando passwd



# Ejecución diferida de comandos

Linux permite realizar ejecución de tareas en tiempo diferido. Mediante 3 posibilidades:

crontab

batch

at



# Ejecución diferida de comandos

- ✓ crontab ejecuta actividades periódicas en días y horarios determinados.
- ✓ cron es el nombre del programa que soporta esta modalidad de funcionamiento.
- ✓ crontab es el comando para utilizarlo.



# Ejecución diferida de comandos

## Comando: crontab

- ✓ Crontab es un archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado por el usuario.
- ✓ Verifica fecha y hora en que se debe ejecutar el script o el comando, los permisos de ejecución, se ejecuta en background.
- ✓ Cada usuario puede tener su propio archivo crontab. El */etc/crontab* se asume que es el archivo crontab del usuario root, cuando los usuarios normales (e incluso root) desean generar su propio archivo de crontab, entonces utilizamos el comando crontab.
- ✓ Crontab es la manera sencilla de administrar tareas de cron en sistemas multiusuario, ya sea como usuario de sistema o usuario root.



# Ejecución diferida de comandos

cron

cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab.



# Ejecución diferida de comandos

Continuación: uso del comando crontab

Ejemplo simple:

“automatizar la actualización de un sistema”.

- 1) Primero hacer un script, será llamado por cron y contendrá todas las instrucciones que queremos que haga
- 2) Probarlo en varios casos y de varias formas antes de incluirlo a cron. ( el error se repetirá con cada actualización)
- 3) Guardar el script, por ejemplo como actualiz.sh
- 4) Cambiar los permisos de ejecución del script. Por ejemplo:  
`chmod a+x ~/scripts/actualiz.sh`
- 5) Agregar la tarea al crontab





# Ejecución diferida de comandos

Continuación: uso del comando crontab

Sintaxis del comando:

**crontab** archcron

crontab -e

crontab -r

crontab -l

El archivo de crontab de cada usuario está en el directorio: /var/spool/cron/crontab con el nombre del usuario.

crontab archcron

La ejecución del comando copia archcron en el directorio que contiene los crontab de todos los usuarios y se ejecuta





# Ejecución diferida de comandos

Continuación: uso del comando crontab

## **crontab -e**

Edita el archivo de crontab del usuario que invoca el comando con el editor por defecto.

## **crontab -r**

Remueve del directorio de crontabs el archivo crontab del usuario que invoca el comando.

## **crontab -l**

Lista el contenido del archivo correspondiente al usuario que invoca el comando en el directorio de crontabs.



# Ejecución diferida de comandos

Comando: at

Propósito: Difiere la ejecución de un comando o conjunto de comandos para un día y horario determinado.

Sintaxis: at hora  
          primer comando  
          .....  
          último comando  
          ctrol -d

hora será reemplazada por la hora real que se desea que corra el comando. También de forma opcional se puede indicar día y mes,



# Ejecución diferida de comandos

Ejemplo:

( Continuación)

tecleamos

\$ at 16 :30

Salte por pantalla

warning : commands will be execute using /bin/ sh  
at >

escribimos:

killall console

para salir

ctrol + d

at < EOT>

job 3 at Tue Oct 16: 50: 00 2018

El comando at responde con el número de trabajo ( job number) y que será ejecutado usando el /bin/sh

EOT end of text



# Ejecución diferida de comandos

## Continuación at

La diferencia entre el comando at y cron, es que at no es persistente; lo cual significa que si se reinicia la PC perderemos la tarea que se le encomendó al comando at.



# El shell

- ❖ Nos permite acceder al kernel del SO
- ❖ Intérpreta la línea de comandos y ejecuta los programa que ésta indica.
- ❖ Constituye la interfaz entre el usuario y el SO.
- ❖ Nos permite administrar el sistema sin entorno gráfico.
- ❖ También puede ejecutar los comandos que se encuentran en un fichero.
- ❖ Hay varios Shell, el usuario puede elegir.
- ❖ Los Shell cumplen la función de lenguajes de programación.
- ❖ En Unix hay dos clases de ejecutables: shell scripts (archivos con comandos) y los binarios.



# El shell

- ❖ Un Shell scripts es un fichero donde introducimos cualquier comando que se pueda ejecutar en la shell.



# El editor sed

- ❖ Sed es un editor de flujo y ficheros de forma no interactiva.
- ❖ Permite modificar el contenido de las diferentes líneas de un fichero en base a una serie de comandos o fichero de comandos.



# El editor sed

Formato de sed:

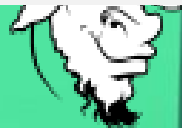
```
# sed [-ns] '[dirección] instrucción argumentos '
```

Usos:

```
# sed [-an] comando [archivo]
```

Opciones:

- a Por defecto los archivo de salida se crean cuando empieza a trabajar sed, con esta opción se abren sólo si hay salida generada.
- n por defecto todo lo que se lee a la entrada aparece a la salida, esta opción anula este comportamiento.





# El editor sed

Formato de sed (continuación):

```
# sed [-an] comando [archivo]
```

Formato de los comandos es.

```
[dirección[, dirección]] función [argumento]
```

Donde:

direcciones	especifican el inicio y el fin de las acciones
funciones	indica la operación a realizar
argumentos	son parámetros pasados a las funciones



# BIBLIOGRAFÍA

Título: Advanced Linux programming

Autor: mark L. Mitchell, Alex Samuel, Jeffrey Oldham

Editorial: New Riders Publishing

ISBN: 0735710430

Disponible gratis en [http: /www.advancedlinuxprogramming.com/](http://www.advancedlinuxprogramming.com/)

Título: Programación en Linux

Autor: Kurt Wall

Editorial: Prentice Hall

ISBN: 0-7897-2215-1



# GRACIAS

