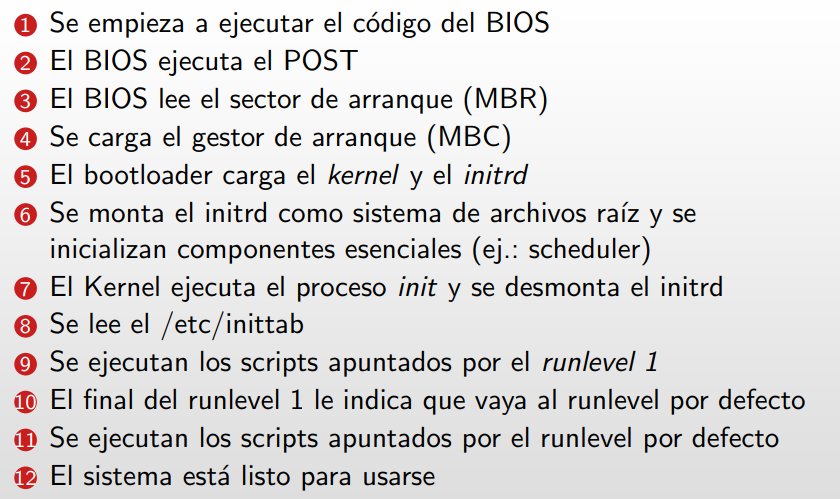
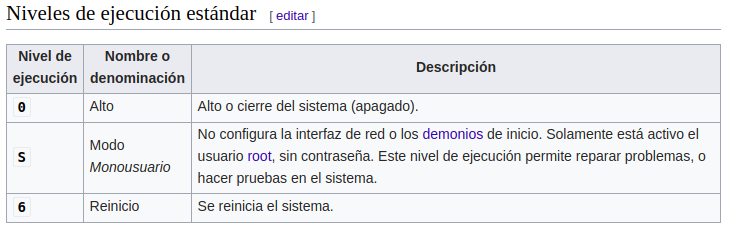
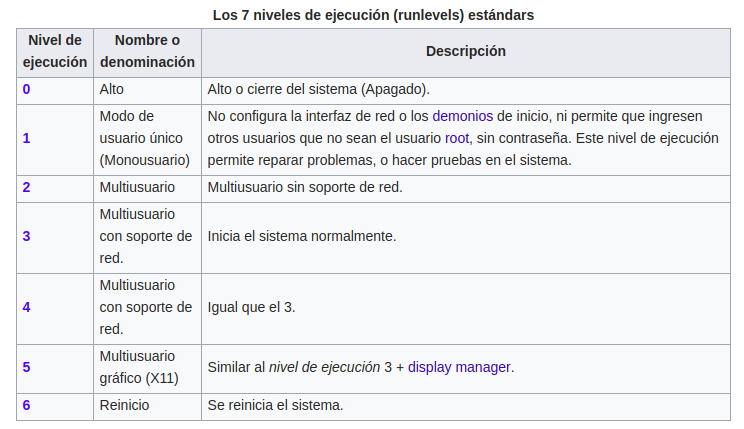
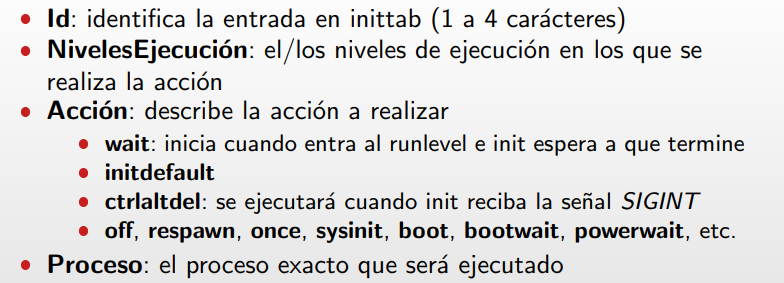
1. Editor de textos:
   1. Vim, subl, gedit
   2. En que en los editores puedo editar el texto. Cat more y less sólo lo muestran. Vim posee los siguientes modos:
      * **Modo comandos (Esc)**: Vim empieza en modo comando, también conocido como modo normal. En este modo se pueden emplear combinaciones de teclas para, por ejemplo, copiar líneas y trabajar en el formato del texto.
      * **Modo inserción (Ins ó i):** En modo inserción cuando se pulsan las teclas se edita el texto como en otros editores. Se puede cambiar del modo comandos al modo inserción pulsando la tecla i.
      * **Modo línea de órdenes:** A este modo se accede pulsando la tecla ***dos puntos :***. Tras los dos puntos se pueden introducir órdenes complejas, como por ejemplo buscar y reemplazar con expresiones regulares.
      * **Modo visual (v)**
   3. **:x** Sale del editor, salvando los cambios. Es decir, cualquier cambio que se haya efectuado en el documento, será guardado en el disco antes de salir.  
      **:q** Sale del editor, solo si no hubieron cambios en el mismo. En caso de haber cambios, se negará a salir, dando la oportunidad de guardar los cambios realizados.  
      **:q!** Igual que :q pero, en este caso, no guarda los cambios, ni alerta de los mismos.  
      **:wq** Guarda el archivo y sale del editor, **a diferencia de :x**, :wq, graba el archivo aún cuando no se hubieran realizado cambios al mismo.  
      **:w** Escribe los cambios  
      **dd** Cortar  
      **p** Pegar desde el portapapeles

**u** Deshacer

1. Proceso de arranque SystemV:
   1. .
   2. Es el primer proceso en ejecución tras la carga del kernel y el que a su vez genera todos los demás procesos. Lo ejecuta el kernel. Se encuentra en /sbin/init
   3. Muestra la locación el árbol de los procesos que están corriendo en el sistema.
   4. Se refiere al modo de operación en los sistemas operativos que implementan el estilo de sistema de arranque de iniciación.
   5. El programa init lee el archivo /etc/inittab para determinar el comportamiento para cada nivel de ejecución. A no ser que el usuario especifique otro valor como un parámetro de autoarranque del núcleo, el sistema intentará entrar (iniciar) al nivel de ejecución por defecto.  
        
        
      La mayoría de distribuciones respetan este estándar.
   6. Finalidad:
      * Que el proceso init se reiniciará
      * Qué procesos se deben iniciar, supervisar e reiniciar si se terminan
      * Qué acciones se deben realizar cuando el sistema ingresa a un nuevo nivel de ejecución

Se almacena un script de configuración en donde se puede elegir por ejemplo el runlevel por default.

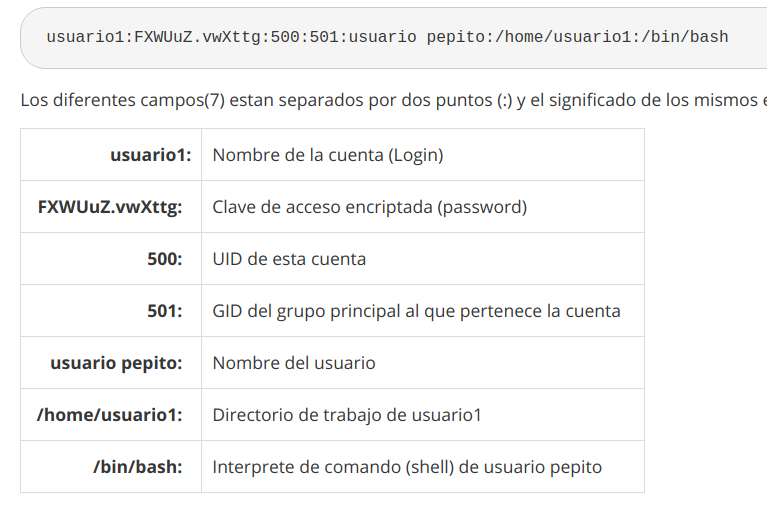
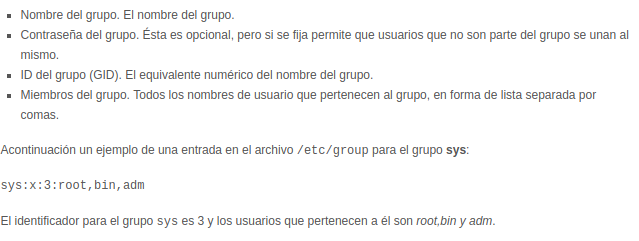
Se encuentra estructurado por:

id:nivelesEjecución:acción:proceso

* 1. Ejecutaría el comando telinit <Y> .. No, no es permanente solo dura lo que dura la sesión.
  2. - Son scripts para controlar el arranque del sistema y los niveles de ejecución. - - Se almacenan en etc/rc**N**.d

- Cuando GNU/Linux inicia primero se ejecuta en el runlevel 1 y luego cambia al runlevel que está por defecto. Se buscan en el orden.  
En el cambio de runlevels se ejecuta el script superior /etc/init.d/rc, el cual se encarga de que los scripts correspondientes sean ejecutados en el orden correcto

* 1. Se utiliza para administrar el orden de los enlaces simbólicos del /etc/rcX.d, resolviendo las dependencias de forma automática. Utiliza cabeceras en los scripts del /etc/init.d que permiten especificar la relación con otros scripts rc
  2. Permite la ejecución de jobs en forma asincrónica a través de eventos.  
     \* Los **jobs** son scripts en texto plano que contienen tareas o servicios que requiere el init y se ejecutan en forma asincrónica a través de eventos. Estan ubicados en /etc/init (.conf). Al inicio el job a ejecutar es el startup.
  3. .
     + Upstart permite la ejecución en forma asincrónica, en cambio en SystemV es sincrónica.
     + Upstart tiene jobs para el inicio del sistema
     + Upstart posee initctl para administrar los jobs, en cambio SystemV posee /etc/inittab para controlar los runlevels
  4. Los reemplaza los jobs .conf, se encuentran en /etc/init. Pueden ser task o service.
  5. # MySQL Service  
     description “MySQL Server” Descripción del job  
     author “info autor” Autor del job  
     start on (net-device-up  
      and local-filesystems  
      and runlevel [2345]) Se ejecuta ante el evento de alta de internet, cuando se monta filesystem y en los runlevels 2,3,4,5  
     stop on runlevel [016] Se termina la ejecución cuando se cambia a los runlevels 0,1 o 6  
       
     [...]  
     exec /usr/sbin/mysqld Path en donde está el script/tarea a ejecutar  
     [...]
  6. Es un sistema de arranque que se centraliza en la administración de demonios y de las librerías del sistema. Tiene capacidades de paralelización de procesos (manejar 2 o más procesos en el arranque al mismo tiempo),
  7. Es un mecanismo de iniciación bajo demanda, podemos ofrecer una variedad de servicios sin que realmente estén iniciados.  
     Cuando el socket recibe una conexión spawnea el servicio y le pasa el socket.  
     No hay necesidad de definir dependencias entre servicios, ya que se inician todos los sockets en primer medida
  8. Agrupa procesos y los usa como uno.

1. Usuarios:
   * + /etc/passwd: Se almacena la configuración de cada usuario
     + /etc/group: Lista todos los grupos existentes  
       
     + /etc/shadow: Contiene las contraseñas de los usuarios y información relacionada con la gestión de las contraseñas.
   1. UID: User ID, GID: Group ID. Puede haber UID repetidos, si lo autorizas como root ya que es sólo un archivo de texto, pero esto no se recomienda
   2. El usuario root tiene acceso total a todas las configuraciones del sistema, no puede haber otro, su cuenta es única. La UID del root es 0.
   3. sudo groupadd catedras **#creo el grupo cátedras**

sudo useradd iso2018 -d /home/iso2018/ -m -g catedras -p pepe **#creo el usuario iso 2018 con su home, le asigno el grupo cátedras y le pongo pepe como contraseña**

sudo touch /home/iso2018/hola **#creo el archivo**

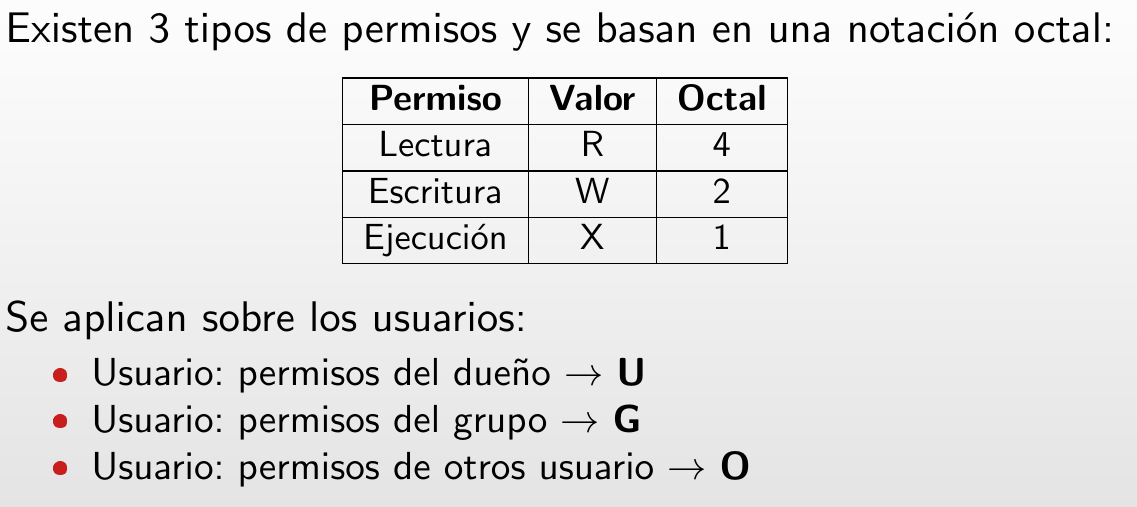
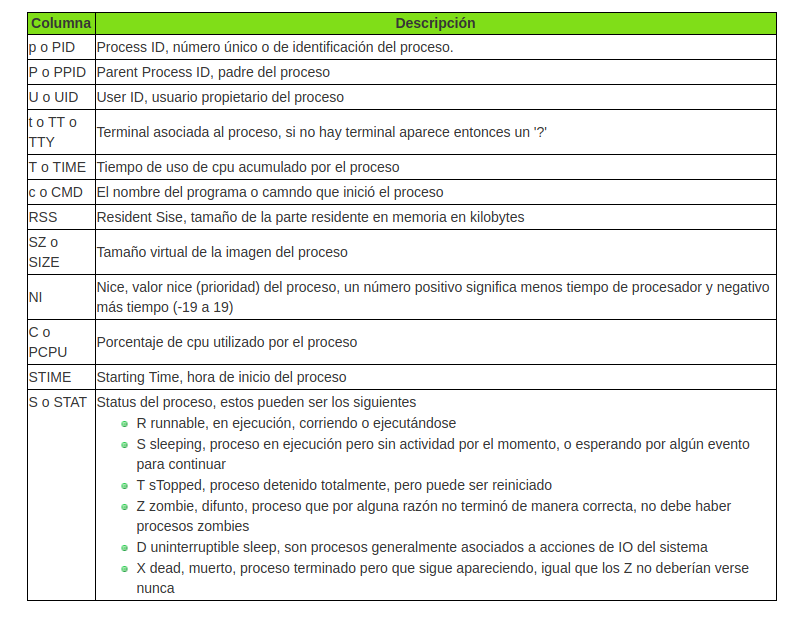
sudo chown iso2018 hola **#asigno la pertenencia al usuario iso2018**

sudo userdel -r iso2018 **#elimino al usuario y su home**

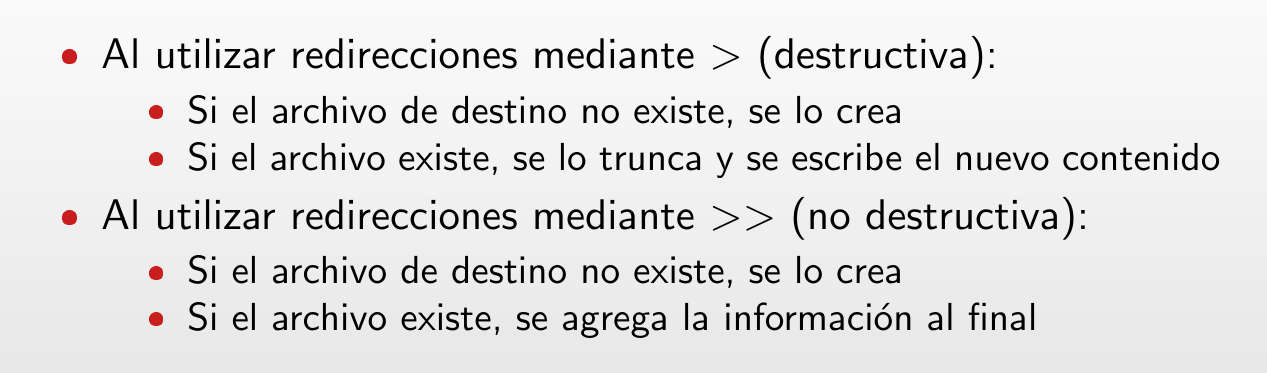
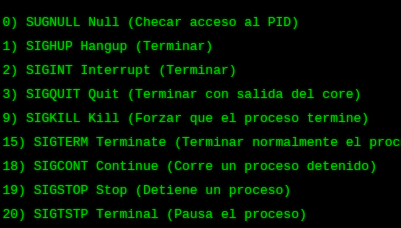
sudo groupdel catedras **#elimino el grupo**

cat /etc/passwd ---- cat /etc/group ----- ls /home/ **#para chequear**

* 1. .
     + **useradd:** Agrega un usuario al sistema. Modifica el archivo /etc/passwd
       1. -d <path> : Pone como home el path
       2. -p <pass>: Pone como contraseña la pass
       3. -m : Fuerza a crear una carpeta home si no está creada
       4. -s <path>: Elige la shell para el usuario
       5. -g <group ó gid>: Asigna grupos al usuario
     + **usermod:** Modifica las configuraciones del usuario
       1. -d <path>: Cambia la /home del usuario (con -m se mueve todo el contenido en la vieja home)
       2. -g <group ó gid>: Cambia el grupo inicial
       3. -p <pass>: Cambia la contraseña
       4. -s <shell>: Cambia la shell
       5. -u : Cambia el UID
     + **userdel:** Elimina a un usuario. Lo borra del /etc/passwd
       1. -r : Elimina el home
       2. -f: Fuerza a eliminarlo incluso si ese usuario está usando el sistema
     + **su:** Asigna a un usuario como superusuario ó cambia su UID por el de otro usuario
     + **groupadd:** Agrega un nuevo grupo. Modifica el archivo /etc/group
     + **who:** Muestra que usuario está logeado
       1. -q: Cantidad de conectados
       2. -r: Runlevel en ejecución
       3. -a: Información más completa
       4. -b: Ultimo arranque del sistema
     + **groupdel:** Elimina el grupo. Lo borra del /etc/group
     + **passwd:** Cambia la contraseña de un usuario. Modifica el archivo /etc/shadow

1. FileSystem:
   1. 
   2. .
      * **chmod:** Cambia los permisos de R,W,X de un archivo. Ejemplo: **chmod U,G,O archivo**. U, G, O reciben la sumatoria de los permisos que queremos. Si queremos que el Usuario sólo lectura, al Grupo lectura y escritura, Otros sin permisos. Entonces » **chmod 460 archivo**
        1. **-w**: Saca los permisos de escritura
        2. **+x**: Pone los permisos de ejecución
        3. **-u-w**: Le saca la escritura sólo al usuario
        4. **=r**: Lectura para todos
      * **chown:** Permite cambiar el propietario de un directorio o archivo
        1. -R: Recursivo
        2. -c: Informa los cambios
      * **chrpg:** Cambia el grupo al que pertenece un archivo o directorio
        1. -R: Recursivo
   3. Cada valor hacer referencia a User, Group o Others. Debido al resultado octal se decide si es r,w,x o alguna combinación entre ambas.
   4. Si, se puede mediante el **sudo**.
   5. **Path absoluto** es aquel que describe la ubicación de un archivo desde el directorio root. Por ejemplo /home/andres/iso2018/practica2  
      **Path relativo** es aquel que describe la ubicación de un archivo desde el directorio actual. Por ejemplo, si estamos en /home/andres, se puede acceder ./iso2018/practica2
   6. Con **pwd.** Se puede acceder a mi home con ~ o con $HOME. Se puede crear una variable de entorno con export $DOCS = /home/andres/Documentos
   7. .
      * **umount <path>:** Desmonta un sistema de archivos
        1. -a: Desmonta todas las unidades
        2. -t <type>: Desmonta todas las unidades de ese fs
      * **du:** Muestra el peso de los archivos / directorios (recursivo)
        1. **-h:** Muestra el formato humano del peso
        2. **-a:** Devuelve todos los archivos
      * **df:** Permite ver el espacio usado y disponible de cada file system del usuario que lo invoca que tenga permisos de lectura
        1. -T: Tipo de sistema de archivos
        2. -h: Formato humano
        3. <path>: Muestra los de ese path
      * **ln:** Crea enlaces entre archivos. **Simbólico no contiene los datos, físico sí.**
        1. -s <origen> <destino>: crea un enlace simbolico desde el origen al destino
      * **cp:** Copia un archivo o un directorio. <archivos> <destino>
        1. **-i:** Informa antes de sobrescribir un archivo en el destino indicado.
        2. **-r:** Recursivo
        3. **-f:** Fuerza el borrado de los archivos sin avisar
      * **Mv:** Mueve un archivo o un directorio a un directorio. <archivos> <destino>
        1. <archivo> <archivo> : Renombramiento
        2. **-i:** Pregunta antes de sobreescribir un archivo
2. Procesos:
   1. Es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados". PID: Process ID, PPID: Parent Process ID. Otros atributos:
   2. Top ó htop
   3. Un programa en foreground lanzado desde un terminal monopoliza dicho terminal, por lo que en principio, no podremos ejecutar ningún otro programa a la vez.

Por el contrario un programa en background una vez iniciado, deja de monopolizar el terminal desde el que se lanzó, y este nos vuelve a mostrar el prompt.

* 1. Para ejecutar un programa en background es: **gimp &**, y para alternar entre foreground y background, utilizando fg y bg. Si hay más de un programa hay que hacerlo con el ID del **jobs**. **Cuando esté en fg hay que apretar ctrl-z para pausar.**
  2. El pipe conecta la salida entrada del primer programa con la entrada estándar del primero. Permite comunicar 2 procesos
     + Ls -a | grep hola **#muestra todos los archivos/directorios que se llamen hola**
     + Ls | more **#muestra la salida del ls como entrada en el more**
  3. .
  4. **Kill <PID>:** Mata al proceso con ese PID. Se le manda por defecto la señal 9, pero puede mandar diferentes señales:
     + **Ps <PID>:** Permite ver la información **instantánea** de un proceso en específico
       1. **-e ó -a:** Muestra de todos
       2. **-u:** Filtra por usuario
       3. **aux:** Vemos todos los atributos, por usuario y en ejecución
     + **Kill:** **en g)**
     + **Pstree:** Muestra el árbol de procesos
       1. **-n:** Ordenamos por PID
       2. **-u:** Vemos los procesos de un usuario en específico
       3. **-H:** Ver el árbol de un proceso específico
     + **Killall:** Mata por nombre de proceso en vez de por PID como el kill.
     + **Top:** Visualiza el estado de todos los procesos en **tiempo real.**
     + **Nice:** Le asigna prioridad de a un proceso. **Nice -n (-19,19) proceso**

1. **Otros comandos:**
   1. Empaquetar es juntar varios archivos en sólo uno. La diferencia entre empaquetar un archivo y/o directorio, y comprimir. Pues bien Empaquetar es agrupar en un solo archivo varios archivos y/o directorios, en cambio, comprimir significa aplicar un algoritmo (que no es otra cosa que un conjunto de pasos) que harán que el archivo ocupe menos espacio en el disco.
   2. Tiene el mismo tamaño. Comandos utilizados:  
      **tar empaquetado.tar arch1 arch2 arch3 arch4  
      du -hc arch1 arch2 arch3 arch4  
      du -hc empaquetado.tar**
   3. **tar empaquetado.tar arch1 arch2 arch3 arch4**
   4. **.**
      * **tar:** Empaqueta muchos archivos en uno, sin comprimirlos
        1. **-c:** crea un archivo tar
        2. **-f:** indica que se le dará un nombre
      * **grep:** Imprime las líneas que cumplan con un patrón
        1. **-c:** Muestra el número de ocurrencias
        2. **-r:** Recursivo
        3. **-v:** Muestra lo que no coincide
        4. **-i:** Ignorar mayúsculas y minúsculas
      * **gzip:** Comprime (también descomprime) ficheros. Por cada fichero que se le pasa como argumento, crea otro comprimido con el mismo nombre y la extensión .gz. El fichero original se borra.
        1. **-d:** Descomprimir
      * **zgrep:** Similar al comando grep pero visualiza archivos comprimidos y no comprimidos.
      * **wc:** Sirve simplemente para contar líneas, palabras y caracteres que contiene un archivo
        1. **-c:** Muestra los caracteres
        2. **-w:** Muestra las palabras
        3. **-l:** Muestra las líneas
2. Ejercicio Práctico
   1. ls -l > prueba **#escribe el archivo prueba con el listado detallado del /home**
   2. ps > PRUEBA **#escribe un nuevo archivo PRUEBA con la información del proceso ps y del shell**
   3. chmod 710 prueba **#rwx para Users, x para Groups y ningún permiso para Others**
   4. chown root:root PRUEBA **#Falla porque necesitas superusuario para cambiar el propietario a root**
   5. chmod 777 PRUEBA **#rwx para users, groups y others en el archivo PRUEBA**
   6. chmod 700 /etc/passwd **#Falla, es un archivo que necesita acceso de superusuario para modificar sus permisos**
   7. passwd root **#No hay permisos para cambiar la contraseña del root, necesita sudo**
   8. rm PRUEBA **#Elimina el archivo PRUEBA**
   9. Man /etc/shadow **#Falla porque no hay permisos para ver el archivo /etc/shadow**
   10. find / -name \*.config **#Va a listar todos los archivos con extension config a partir de root, pero va a tirar permiso denegados en algunos**
   11. usermod root -d /home/newroot -L **#No se tiene permiso para cambiar el home del root**
   12. cd /root **#No se tiene permiso para ver los archivos de esa carpeta**
   13. rm \* **#Borra todos los archivos que confirmamos menos aquellos que no tengamos permisos o sea un directorio**
   14. cd /etc **#Entra al directorio de configuraciones de los usuarios**
   15. cp \* /home -R **#Falla porque no hay permisos para escribir en home**
   16. shutdown **#Programa el apagado en 1 minuto**
3. Ejercicios prácticos
   1. kill 23
   2. No lo permite
   3. find -user andres -name \*.conf
   4. ps -la > /home/andres/archivo
   5. sudo chmod 751 xxxx
   6. sudo chmod 650 yyyy
   7. rm -r /tmp
   8. sudo chown iso2010 /opt/isodata
   9. pwd >> /home/andres/donde
4. Ejercicios Prácticos
   1. su
   2. sudo useradd -n amilla  
      sudo passwd amilla  
      -> contraseña
   3. Fueron modificados los archivos /etc/passwd y /etc/shadow. No se creó ningún directorio ya que no especifiqué que se creara el /home
   4. mkdir /tmp/cursada2010
   5. sudo cp -r /var/log /\* tmp/cursada2010
   6. sudo chown -R amilla:users /tmp/cursada2010
   7. sudo chmod -R 723 /tmp/cursada2010
   8. su amilla
   9. echo $SHELL
   10. ps -aux | wc -l
   11. who | wc -l
   12. write user tty <<< “hola”
   13. shutdown -now
5. Ejercicio Práctico
   1. mkdir 149346  
      cd 149346
   2. vi LEAME
   3. chmod 017 LEAME
   4. su  
      cd /etc  
      ls  
      sudo ls > leame
   5. Utilizaría find y sus parámetros para ver características de los archivos
   6. sudo find / -name \*.so > ejercicio\_f
6. Ejercicio Práctico
   1. mkdir iso **#crea un nuevo directorio**
   2. cd ./iso; ps > f0 **#se mueve al directorio iso y escribe un archivo en ese directorio llamado f0 con el proceso del bash y ps**
   3. ls > f1  **#crea un nuevo archivo f1 con el listado de los archivos en el directorio (en este caso solo f0)**
   4. cd / **#se mueve al directorio root**
   5. echo $HOME **#imprime el directorio en donde está el home**
   6. ls -l > $HOME/iso/ls **#crea un nuevo archivo llamado ls con el contenido detallado del directorio raíz en /home/usuario/iso/**
   7. cd $HOME; mkdir f2 **#va al directorio /home/usuario y crea un directorio llamado f2**
   8. ls -ld f2 **#muestra el detalle detallado del directorio f2**
   9. chmod 341 f2 **#cambia los permisos del directorio f2: ejecución y escritura para usuario, sólo lectura para el grupo y sólo ejecución para otros usuarios**
   10. touch dir **#crea un nuevo archivo en la carpeta actual**
   11. cd f2 **#se mueve al directorio f2**
   12. cd ~/iso **#se mueve al directorio /home/usuario/iso**
   13. pwd > f3 **#escribe el path actual en un nuevo archivo llamado f3**
   14. ps | grep ‘ps’ | wc -l >> ../f2/f3 **#crea un archivo f3 en el directorio f2**
   15. chmod 700 ../f2; cd .. **#cambia los permisos de f2 a r,w,x para usuario, y le anula los permisos a los grupos y a los otros, luego se mueve al home**
   16. find . -name etc/passwd **#nunca va a encontrar nada ya que los archivos no pueden contener una barra**
   17. Lo mismo para este
   18. mkdir ejercicio5 **#crea un nuevo directorio llamado ejercicio5 en el home**
   19. cp -R ~/iso ~/149346
7. Ejercicio Práctico
   1. **ESTRUCTURA:**  
       mkdir dir1 dir2 dir1/dir11  
       touch f1 f2 dir1/f3 dir2/f4
      * mv dir1/f3 .
      * cp dir2/f4 dir1/dir11
      * cp dir/f4 dir1/dirr11/f7
      * mkdir copia  
        cp -R dir1 copia
      * mv f1 archivo
      * chmod 617 archivo
      * mv f3 f3.exe; mv dir2/f4 dir2/f4.exe
      * chmod 023 f3.exe dir2/f4.exe
8. Ejercicio Práctico
   1. mkdir /tmp/logs
   2. sudo cp -R /var/log /tmp/logs
   3. sudo tar -cf /tmp/misLogs.tar /tmp/logs
   4. sudo gzip -k /tmp/misLogs.tar -n misLogs.tar.gz
   5. cp /tmp/misLogs.tar misLogs.tar.gz ~
   6. sudo rm -R /tmp/logs
   7. mkdir misLogs1  
      tar -xzf misLogs.tar.gz -C misLogs1  
      mkdir misLogs2  
      tar -xf misLogs.tar -C misLogs2