1. Sistema de ficheros y gestión de aplicaciones I

1.1. Sabe crear carpetas y ficheros y gestionarlos usando los comandos básicos:

- Crea la carpeta “prueba” en tu carpeta personal.
  mkdir prueba

- Dentro de la carpeta “prueba” crea otra carpeta “trabajo”.
  mkdir prueba/trabajo

- Crea los siguientes ficheros vacíos dentro de la carpeta “trabajo”:
  gato.txt
  gata.txt
  gatita.txt
  gatito.txt
  perro.txt
  perra.txt
  gallo.txt
  gallina.txt
  cd prueba/trabajo
  touch gato.txt gata.txt gatita.txt gatito.txt perro.txt perra.txt gallo.txt gallina.txt

- Crea una carpeta “alumnos” en tu carpeta personal. Dentro de la carpeta “alumnos” crea 2 ficheros con la siguiente información de cada alumno. Por ejemplo “nano bego” que incluya la siguiente información:
  Nombre: Bego
  Calle: Nieves Cano
  Número: 3
  Teléfono: 945945945
  mkdir alumnos
  nano alumnos/bego
  Nombre: Bego
  Calle: Nieves Cano
  Número: 3
  Teléfono: 945945945
  Ctrl O + Enter + Ctrl X
  nano alumnos/iker
  Nombre: Iker
  Calle: Alameda Urquijo
  Número: 5
  Teléfono: 678678678
  Ctrl O + Enter + Ctrl X

- Cambia el nombre de los ficheros a mayúsculas.
  cd alumnos
  mv bego BEGO
  mv iker IKER

- Crea un enlace simbólico para un alumno y un enlace real para el otro. ¿Qué diferencias hay entre los diferentes tipos de enlaces?
  ln -s ~/alumnos/BEGO ~
  ln ~/alumnos/IKER ~/IKERTXU
  El enlace simbólico, es decir, el de BEGO se comporta como un "Acceso directo" de Windows, es decir, apunta al fichero con su nombre, si se borra el fichero apuntado no se borrara el enlace, pero no encontrará el archivo (tienen diferentes inodos). El enlace real, en cambio, es un fichero con otro nombre pero que apunta al mismo contenido de datos (tienen el mismo inodo), por lo que, un cambio uno afecta en todos.

- Lista el contenido de este directorio(también los ficheros ocultos), con información de inodo, tamaño e información larga que contenga permisos ,propietario, grupo, fecha de actualización y nombre.
  ls -lias ~

- Listar todos los ficheros de la carpeta personal (los ocultos también)
  ls -a ~

- Busca los ficheros modificados en menos de 5 minutos de la carpeta personal.
  find /home/$USER -mmin -5

- Busca los ficheros cuyo tamaño es superior a 100 Kb de la carpeta personal.
  find /home/$USER -size +100k

1.2. Sabe manejarse usando rutas absolutas y relativas:

- Copia un fichero de uno de los alumnos a la carpeta padre usando la ruta absoluta.
  cp home/$USER/alumnos/BEGO home/$USER

- Copia un fichero de uno de los alumnos a la carpeta padre usando la ruta relativa sabiendo que estamos en el directorio personal.
  cp alumnos/BEGO .

- Cambia al directorio "prueba", usando la ruta absoluta.
  cd home/$USER/prueba

- Cambia al directorio “prueba”, usando la ruta relativa sabiendo que estamos en el directorio personal.
  cd prueba

- Cambia al directorio “prueba”, usando la ruta relativa sabiendo que estamos en el directorio “trabajo”.
  cd ..

- Cambia al directorio "prueba", usando la ruta relativa sabiendo que estamos en el directorio “/tmp”.
  cc ../home/$USER/prueba

1.3. Sabe utilizar las expresiones regulares utilizando la sintaxis globbing utilizando el comando ls:

- Cambia a la carpeta "trabajo" donde tienes creados los siguientes ficheros vacíos: gato.txt, gata.txt, gatita.txt, gatito.txt, perro.txt, perra.txt, gallo.txt y gallina.txt para realiza los siguientes ejercicios utilizando la sintaxis globbing:
  cd prueba/trabajo

- Lista los archivos cuyo nombre coincida con las palabras:gato.txt, gata.txt, gatito.txt y gatita.txt
  ls gat*.txt

- Lista los archivos cuyo nombre coincida con las palabras:gato.txt y gata.txt
  ls gat?.txt

- Lista los archivos cuyo quinto carácter sea “o”: gallo.txt y perro.txt
  ls ????o.txt

- Lista los archivos cuyo segundo y cuarto carácter sea una vocal: gato.txt, gata.txt,gatito.txt y gatita.txt
  ls ?[aeiouAEIOU]?[aeiouAEIOU]?*.txt

1.4. Sabe utilizar las expresiones regulares utilizando la sintaxis regex utilizando el comando grep:

- Buscar las líneas en las que aparece la palabra bash en el archivo /etc/passwd.
  grep ‘bash’ /etc/passwd

- Buscar en el archivo /etc/group todas las líneas que empiezan por r.
  grep ‘^r’ /etc/group

- Buscar las líneas en las que aparece la palabra bash al final de la línea en el archivo /etc/passwd.
  grep 'bash$' /etc/passwd

- Buscar las líneas de /etc/passwd de los usuarios cuyo identificador de usuario es menor que 1000.
  grep 'x:[0-9][0-9][0-9]:' /etc/passwd

- Buscar las líneas de /etc/passwd de los usuarios cuyo identificador de usuario es mayor o igual a 1000.
  grep 'x:[0-9][0-9][0-9][0-9]*' /etc/passwd

- Buscar las líneas que contienen 2 o más apariciones de "o" seguidas en el fichero /etc/passwd.
  grep '?*ooo*?*' /etc/passwd

1.5. Sabe gestionar usuarios

- Crea un nuevo usuario "lsi1" con contraseña "lsi1" utilizando el comando useradd. (man useradd)
  sudo useradd -m lsi1
  sudo passwd lsi1

- Crea un nuevo usuario "lsi2" con contraseña "lsi2" utilizando el comando adduser. (man adduser)
  sudo adduser lsi2
  Se añade una contraseña para el usuario

- Crea el grupo "desarrolladores" y añade a "lsi1" y "lsi2" a dicho grupo.
  sudo addgroup developers
  sudo usermod -a -G developers lsi1
  sudo usermod -a -G developers lsi2

- Dale el poder de administrar el grupo "desarrolladores" al jefe de proyecto "lsi2"
  sudo gpasswd -A lsi2 developers

- Dale a "lsi2" los mismos permisos que el administrador del sistema añadiendo los mismos grupos que el administrador del sistema.
  id
  sudo usermod -a -G adm,cdrom,sudo,dip,plugdev,users,lpadmin lsi2

- Con el comando id observa los grupos a los que pertenece "lsi1" y "lsi2". Explica para qué sirve cada grupo.
  id lsi1
  uid=1001(lsi1) gid=1001(lsi1) grupos=1001(lsi1),100(users),1002(developers)
  id lsi2

uid=1001(lsi2) gid=1001(lsi2)
grupos=1001(lsi2),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),100(users),114(lpadmin),1002(developers)
adm, administración del sistema, pueden leer logs del sistema.
cdrom, permite acceder a CD/DVD.
sudo, permisos de superusuario.
dip, administración de redes antiguas.
plugdev, acceso a dispositivos como USBs.
users, grupo de usuarios.
lpadmin, se utiliza administrar las impresoras.

- Finalmente elimina las cuentas creadas (lsi1 y lsi2) y el grupo "desarrolladores"
  sudo deluser lsi1
  sudo deluser lsi2
  sudo delgroup developers

1.6. Sabe manejarse con los permisos de usuario:

- Crear un fichero llamado "reloj.sh" añadele como contenido el comando "date".
  nano reloj.sh
  date
  Ctrl O + Enter + Ctrl X

- Darle permisos de ejecución a "reloj.sh"
  chmod 764 reloj.sh
  chmod u=rwx,g=rw,o=r reloj.sh

- Ejecuta "reloj.sh"
  ./reloj.sh

- Crea un fichero "SoyNuevo" para asignarle usando el modo octal y usando el modo literal los siguientes permisos:
  - El propietario tenga permisos de lectura, escritura y ejecución
  - El grupo solo lectura
  - El resto u otros solo lectura

  touch SoyNuevo
  chmod 744 SoyNuevo
  chmod u=rwx,g=r,o=r SoyNuevo

- Comprueba la máscara actual con el comando (man umask).
  umask

- Asigna una nueva máscara "umask 077", crea ficheros y carpetas para comprobar que la máscara funciona correctamente.
  umask 077
  touch prueba
  ls -lias prueba
  11708987 0 -rw------- 1 iker iker 0 feb 5 17:47 prueba

- Finalmente deja la máscara como estaba al principio por defecto.
  umask 0002
- ¿Cuál es el comando para que las carpetas que se creen en dicha sesión tengan los siguientes permisos?:
  - El propietario tenga permisos de lectura, escritura y ejecución
  - El grupo solo lectura y escritura
  - El resto u otros solo lectura

  umask 013
  nano .bashrc
  umask 013
  Ctrl O + Enter + Ctrl X

- Finalmente deja la máscara como estaba al principio por defecto.
  nano .bashr
  umask 0002
  Ctrl O + Enter + Ctrl X

- Para siempre (futuras sesiones) modifica el fichero ".bashrc" con la siguiente máscara "umask 0002".
  nano .bashr
  umask 0002
  Ctrl O + Enter + Ctrl X

- Añadele el bit s al "reloj.sh"
  chmod 4764 reloj.sh

- Detecta el bit sticky en la carpeta tmp, crea un archivo con un usuario e intenta borrar con otro usuario. ¿Puedes?
  ls -dlias /tmp
  10485761 4 drwxrwxrwt 22 root root 4096 feb 6 09:20 /tmp
  touch /tmp/prueba
  rm /tmp/prueba
  rm: ¿borrar el fichero regular '/tmp/prueba' protegido contra escritura? (s/n) s
  rm: no se puede borrar '/tmp/prueba': Operación no permitida

Sabe utilizar redireccionamiento y tuberias:

- Averigua cuántos ficheros hay en tu directorio de trabajo (*man wc*).
  ls . | wc -l
- Guarda la fecha en el fichero "conectados.txt" (*man date*)
  date > conectados.txt
- Añade al fichero "conectados.txt" los usuarios conectados a la máquina (*man who*)
  who » conectados.txt

- Busca el fichero “fstab” en toda la partición raíz y redirecciona los errores al fichero “errores.txt” (man find)

  find / -name fstab 2> errores.txt
- Crea un fichero “animales.txt” que contenga:

  perro
  gato
  vaca
  toro
  cerdo

  cat > animales.txt

  perro
  gato
  vaca
  toro
  cerdo
  Ctrl D
- Y crea un fichero que se llama “animalesordenados.txt” que contendrá todos los animales de “animales.txt” ordenados alfabéticamente usando el comando *sort* (*man sort*).

  sort animales.txt > animalesordenados.txt
- Redirecciona el error estándar a un archivo “err.txt” al ejecutar el siguiente comando: “cat /ceq11”

  cat /ceq11 2> err.txt
- Redirecciona tanto la salida estándar como error estándar del siguiente comando “cat /etc/fstab” al fichero “salidaerror.txt”.

  cat /etc/fstab &> salidaerror.txt
- Muestra la cantidad de archivos y directorios que hay en el directorio /, usando el comando *wc* y *ls*.

  ls / | wc -l
- Dado un archivo con nombres de personas “pers.txt” (con un nombre en cada línea), se pide que se muestre los 10 primeros nombres por orden alfabético (usa *sort* y *head*).

  sort pers.txt | head pers.txt
- Dada una lista de palabras en un archivo “palabras.txt” (con una palabra en cada línea), se pide que se muestre la cantidad de palabras no repetidas. (usa *sort* y *wc*)

  sort -u persona.txt | wc -w

- Cuenta el número de líneas que tiene el fichero “animalesordenados.txt” utilizando redireccionamiento de entrada.

  wc -l < animalesordenados.txt