IES Puerto de la Cruz
Telesforo Bravo

Sistemas Informáticos

Alfonso García Jorge

1º DAM

1.Genera un fichero de 1GiB llamado "datos". Por ejemplo, puedes usar: dd if=/dev/zero of=datos bs=1MiB count=1024

```
alfonsogj@alfonso:~/Escritorio$ dd if=/dev/zero of=datos bs=1MiB count=1024
1024+0 registros leídos
1024+0 registros escritos
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 2,00173 s, 536 MB/s
alfonsogj@alfonso:~/Escritorio$
```

2.Separar por tamaño:

1.Divide el fichero orignal "datos" en trozos de 240 MiB con formato datosA0001, datosA0002, datosA0003, …

split -b 240M datos datos

```
-rw-rw-r-- 1 alfonsogj alfonsogj 1073741824 feb 11 22:31  datos
-rw-rw-r-- 1 alfonsogj alfonsogj  251658240 feb 11 22:45  datosAaa
-rw-rw-r-- 1 alfonsogj alfonsogj  251658240 feb 11 22:45  datosAab
-rw-rw-r-- 1 alfonsogj alfonsogj  251658240 feb 11 22:45  datosAac
-rw-rw-r-- 1 alfonsogj alfonsogj  251658240 feb 11 22:45  datosAad
-rw-rw-r-- 1 alfonsogj alfonsogj   67108864 feb 11 22:45  datosAae
```

2.Une todos los trozos del apartado anterior en un fichero llamado datosA

cat datos* > unido

```
-rw-rw-r-- 1 alfonsogj alfonsogj 1073741824 feb 11 22:42  unido
-rw-rw-r-- 1 alfonsogj alfonsogj  251658240 feb 11 22:38  xaa
```

3.Comprueba que datos datos y datosA son exactamente el mismo fichero

Si son lo mismo.

3.Separar por número de trozos:

1.Divide el fichero original "datos" en 10 trozos llamados datosB0001, datosB0002, datosB0003

2.Une todos los trozos del apartado anterior en un fichero llamado datosB

3.Comprueba que datos datos y datosB son exactamente el mismo fichero

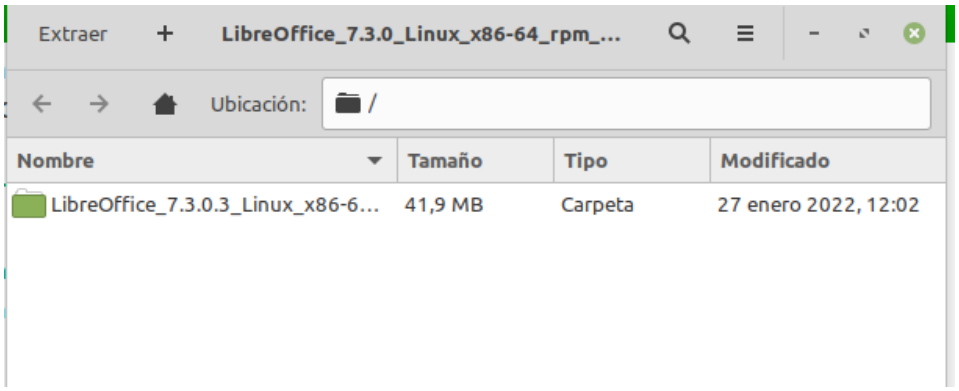
4.Separar por número de líneas:

1.Genera ahora un fichero de TEXTO con 15000 líneas llamado libro (¿Cómo lo harías? Ya hemos visto en clase algunas formas de hacerlo):

2.Divide ahora el fichero "libro" en otros que tengan 2000 líneas cada uno

3.Une los trozos anteriores y comprueba que es igual que el fichero original

1.Descarga el SDK (comprueba que estás en la versión GNU/Linux) y el código fuente de LibreOffice: https://es.libreoffice.org/descarga/libreoffice/



2.Descomprime el SDK y el código fuente, indica los comandos utilizados.

tar xvf LibreOffice_7.3.0_Linux_x86-64_rpm_sdk.tar.gz

3.Crea un directorio llamado LibreOffice y mueve dentro de ese directorio los dos directorios anteriormente descomprimidos (SDK y código fuente). ¿Cuánto espacio ocupa el directorio LibreOffice?

du -csh * | grep L

4.64K GNU-Linux09

5.41M LibreOffice_7.3.0.3_Linux_x86-64_rpm_sdk

6.37M LibreOffice_7.3.0_Linux_x86-64_rpm_sdk.tar.gz

7.Vamos a probar a comprimir/descomprimir usando diferentes aplicaciones y algoritmos. Debes rellenar la tabla que se indica al final.

Archivos .tar.gz:
Comprimir: tar -czvf empaquetado.tar.gz /carpeta/a/empaquetar/
Descomprimir: tar -xzvf archivo.tar.gz

Archivos .tar:
Empaquetar: tar -cvf paquete.tar /dir/a/comprimir/
Desempaquetar: tar -xvf paquete.tar

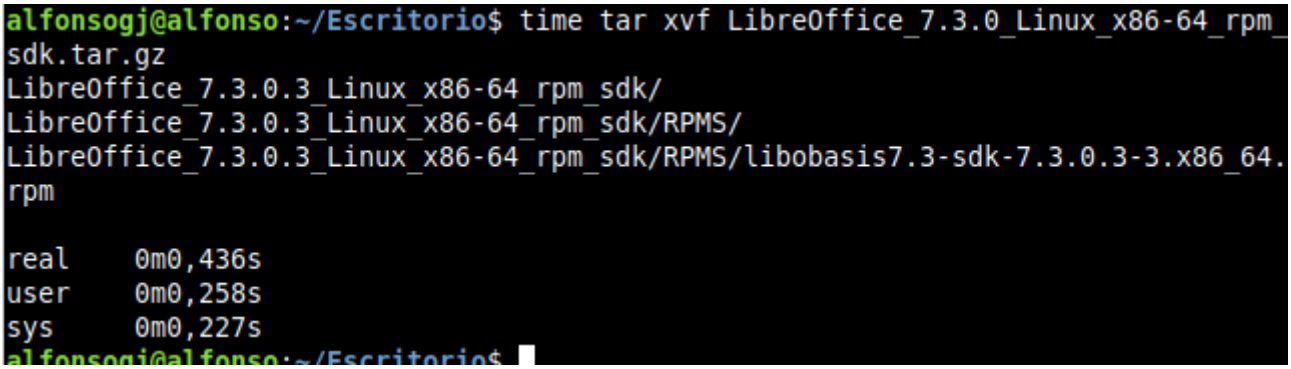
Archivos .gz:
Comprimir: gzip -9 index.php
Descomprimir: gzip -d index.php.gz

Archivos .zip:
Comprimir: zip archivo.zip carpeta
Descomprimir: unzip archivo.zip

8.Una vez hayas rellenado la tabla anterior, indica tus conclusiones.

9.¿Por qué crees que hay varias aplicaciones y algoritmos de compresión?

Para saber el tiempo con poner time antes del comando te aparece.



Aplicación/algoritmo	**Comando(s) comprimir**	**Tiempo comprimir**	**% espacio comp.**	**Comandos descomp.**	**Tiempo descomp.**
gzip	tar czvf archivo.tar.gz /archivo/mayo/*			gzip -d archivo.gz	real 0m0,436s user 0m0,258s sys 0m0,227s
xz	tar -c archivos | bzip2 > archivo.tar.bz2			bzip2 -dc archivo.tar.bz2 | tar –xv	
bzip2	bunzip2 archivo	real 0m0,025s user 0m0,002s sys 0m0,000s		bunzip2 archivo.bz2	
lha	lha archivo.lha /mayo/archivos			lha -x archivo.lha	
zip	zip archivo.zip /mayo/archivos			unzip archivo.zip	

Aplicación/algoritmo	Comando(s) comprimir	Tiempo comprimir	% espacio comp.	Comandos descomp.	Tiempo descomp.
rar	rar -a archivo.rar /mayo/archivos			rar -x archivo.rar	

1) Sobre el comando que se utiliza para modificar los permisos, indica lo siguiente

1.¿Qué comando es?

chmod

2.¿Qué opción debo usar para que me vaya mostrando todos los ficheros y directorios afectados?

ls l o chmod -c

3.¿Qué opción debo usar para que me vaya mostrando sólo los ficheros y directorios que han cambiado el permiso?

chmod -v

2) El comando para cambiar los permisos tiene una opción para no sólo cambiar los permisos de un directorio, sino también de TODOS su contenido, incluyendo subdirectorios.

1.¿Qué opción permite eso?

chmod -R

2.¿Por qué puede ser MUY peligroso asignar el mismo permiso a todo el contenido de un directorio?

Este comando cambiará el permiso de todos los archivos que estén contenido dentro de este directorio. Pudiendo dar permiso a otros usuarios de algo que no debería de tener permisos. Para ello se utiliza lo que se comenta en la siguiente pregunta.

3.Investiga cómo se podría evitar este problema que has comentado en el apartado anterior.

Una forma para limitar los permiso de todos los archivos de un directorio es utilizando en modo simbólico, asignando al usuario con la u, al grupo con la g, a otros usuarios con la o, cuando nos referimos a todos con la a, cuando ponemos el signo + para establecer el permiso y cuando utilizamos – para quitar el permiso. Con esto podremos limitar mucho mas a la hora de poner o quitar permisos. Un ejemplo es chmod 0+w archivo , chmod u+x.

3) Indica cómo se ven los permisos de ficheros y directorios en GNU/Linux y qué significa cada uno de ellos. ¿Cómo veo quién es el propietario del fichero y a qué grupo pertenecen?

Podemos ver los permisos cuando listamos un directorio con ls -l:
Un ejemplo: -rwxrwxr-- 1 sergio ventas 9090 sep 9 14:10 presentacion

rwx	rwx	rwx
usuario	grupo	otros

El primer grupo al usuario, el segundo grupo a grupo y el tercer a otros.

4) Sobre los permisos:

1.¿Cuáles son los permisos más comunes para ficheros y directorios y qué implican?

644 para ficheros y 755 para directorios

-rw-r--r-- 1 alfonsogj alfonsogj 0 feb 11 15:41 nose → 644

drwxr-xr-x 2 alfonsogj alfonsogj 4096 feb 11 15:42 nose2 → 755

2.Cuando creas un fichero cualquiera en tu equipo, ¿qué permisos se establecen por defecto y qué permiten y qué no?

-rw-rw-r-- 1 alfonsogj alfonsogj 0 feb 11 15:41 nose

El usuario y el grupo no tiene permiso para ejecutarlo y el otros solo tiene permiso para leer.

3.Y si en vez de crear un fichero creas un directorio, ¿cuáles son los permisos por defecto?

drwxrwxr-x 2 alfonsogj alfonsogj 4096 feb 11 15:42 nose2

El usuario y el grupo tiene los permiso de lectura, escritura y ejecución, pero otros no tiene los permiso de modificarlo.

4.Investiga cómo se podrían cambiar estos permisos por defecto

Lo único que necesitaremos será tener acceso al usuario root para poder ejecutar el comando chmod que nos permita modificar estos permisos rwx. Con estos comandos da igual los permisos que traiga nuestra distro Linux por defecto, podremos especificar rápidamente los que nosotros queramos.

5) Crea 10 archivos en tu equipo, desde fich01 a fich10. Elige los permisos más adecuados para los siguientes casos e indica el comando y opciones para asignarle los permisos:

1.fich01: Es un documento confidencial en el que estás trabajando tú con tu equipo

rwx------

2.fich02: Son unas instrucciones que deberían poder ser consultadas por todos, pero sólo tú puedes modificarlas

rwxr--r--

3.fich03: Es un documento muy confidencial, sólo tú deberías tener acceso

rwx------

4.fich04: Es un documento compartido que todos deberían poder leer y modificar

rw-rw-rw-

5.fich05: Es un documento que estás modificando y que, además de ti, sólo podrá ser consultado por tu grupo, sin que pueda modificarlo

rwxr-x---

6.fich06: A este documento falta por añadirle el permiso para que otros puedan leerlo y escribirlo

rwxrw-rw-

7.fich07: Este documento podía ser modificado por tu grupo y otras personas, pero ahora has decidido que sólo tú puedas modificarlo

rwxr-xr-x

8.fich08: Es un script que cualquiera puede ejecutar, pero sólo tú puedes modificar

rwxr-xr-x

9.fich09: Es un script que tanto tú como tu equipo pueden modificar y ejecutar

rwxrwx---

10.fich10: Es un documento que quieres proteger para que sólo tú puedas leerlo, pero no modificarlo

r--------

6) Crea 5 directorios en tu equipo, desde dir1 a dir5. Elige los permisos más adecuados para los siguientes casos e indica el comando y opciones para asignarle los permisos:

1.dir1: Tú y solo tú tienes el control total de este directorio, mientras que el resto no puede hacer nada

d rwx------

2.dir2: Tú tienes el control total, pero tu grupo puede acceder y ver el contenido

d rwxr-----

3.dir3: Tú y tu equipo pueden ver el contenido del directorio y acceder a él, pero no modificarlo. Los demás sólo acceder sin ver el contenido ni modificarlo

d r-xr-xr-x

4.dir4: Todos tienen el acceso total

d rwx rwx rwx

5.dir5: Tú tienes control total, tu grupo puede acceder y modificar sin ver contenido, los demás sólo ver contenido

d rwx-wxr--

7) Indica de qué tipo se trata (fichero, directorio, ...) y qué se puede hacer y que no con los siguientes permisos:

1.drwxr-xr-x

Directorio donde el usuario puede realizar todo, el grupo y otro no puede modificarlo

2.-rwxr-xr-x

Fichero donde el usuario puede realizar todo y el grupo y otro no pueden modificarlo.

3.Lrwxrwxrwx

Especial donde todos puedes leer, modificar y ejecutar

4.drwxr-x---

Directorio donde el usuario puede realizar de todo el grupo no puede modificarlo y otros no puede realizar nada

5.-rw-r—r--

Fichero donde el usuario puede leer y modificar pero no ejecutar , el grupo y otro solo pueden leer

6.-rw-rw----

Fichero donde el usuario y el grupo puede leer y modificar peor no ejecutar y otros no puede realizar nada.

7.

8.Fichero con permisos 644

-rw-r--r--

El usuario solo puede leer y modificar, el grupo y otros solo leer

9.Directorio con permisos 755

drwxr-xr-x

El usuario puede leer, modificar y ejecutar. El grupo y otros no pueden modificar.

10.Fichero con permisos 600

-rw-------

El usuario solo puede leer y modificar, el grupo y otros no pueden realizar nada

11.Directorio con permisos 740

drwxr-----

El usuario puede realizar todo, el grupo solo leer, y otros no pueden realizar nada.

8) En mi servidor web estoy teniendo un problema y sospecho que es por permisos, ¿es buena idea asignar el permiso 777 a los directorios para ver si así se soluciona el problema? Razona la respuesta.

Nunca debemos asignar permisos 777. Aunque a veces pueda parecer que una aplicación web no funciona si no le asignamos esos permisos a una carpeta donde se desea escribir información, el nivel 777 es muy peligroso ya que estamos asignando permisos totales a esos archivos o carpetas. Generalmente, hay que buscar soluciones de permisos más restrictivas y, en cambio, modificar los propietarios o el grupo de los archivos o carpetas.

Generalmente, trabajamos con 644 para ficheros y 755 para directorios. Esos son los permisos más estándar para los archivos y carpetas del servidor. Esta opción genérica no tiene por qué ser la necesaria para todos los proyectos, pero siempre es una solución bastante apropiada. Nuevamente, si nuestra aplicación no funciona bien con este nivel de permisos, conviene estudiar quién se encuentra asignado como propietario y grupo de los archivos y carpetas. Un problema

habitual es que el propietario sea root. Esto podría suponer que, cuando desde un lenguaje de programación se intenta acceder al archivo para su escritura, nos arroje un error de acceso. La solución adecuada nunca sería asignar 777 para que la aplicación funcione, sino encontrar el usuario y grupo adecuados para el contenido de la carpeta.

Los script se suben en un ZIP

1) Crear un script llamado **ej01-resta.sh** al que le pasemos dos argumentos y nos devuelva la resta.

Ejemplo: ./resta.sh 12 8 --> El resultado de la operación 12 - 8 = 4

2) Crear un script llamado **ej02-crea_usuario.sh** al que se le pasan tres parámetros (1: nombre, 2: apellidos, 3: usuario). El script imprimirá el siguiente mensaje en pantalla:

Bienvenido, {nombre}
Tus datos son: {nombre} {apellidos}
Vamos a crear tu usuario: {usuario}
Tu nueva ID es {aleatorio}

3) Crear un script llamado **ej03-dia_semana.sh** al que se le pasen tres parámetros (1: día, 2: mes, 3: año). El script deberá devolver un texto diciendo: "El día de la semana de la fecha indicada (día/mes/año) fue: xxxx".

Ejemplo: ./dia_semana.sh 17 11 2016 --> El día de la semana de la fecha indicada (17/11/2016) fue jueves.

4) Crear un script llamado **ej04-calcula_segundos.sh** al que se le pasen 4 parámetros (1: días, 2: horas, 3: minutos, 4: segundos). El script devolverá el total de segundos del tiempo indicado.

Ejemplo: ./calcula_segundos.sh 4 3 29 54 --> 4 días, 3 horas, 29 minutos y 54 segundos son 358.194 segundos.

5) Crear un script llamado **ej05-calcula_tiempos.sh** al que se le pase un único parámetro, el número de segundos. El script devolverá a cuántos días, horas, minutos y segundos corresponden los segundos indicados.

Ejemplo: ./calcula_segundos.sh 358194 --> 358.194 segundos son 4 días, 3 horas, 29 minutos y 54 segundos.

6) Crera un script llamado **ej06-calcula_cambio.sh** al que se le pase como parámetro el precio de un artículo. Luego el script debe solicitar el dinero entregado por teclado, y calcular el cambio que debe darle en billetes de 50, 20, 10 o 5 euros, y monedas de 2 o 1 euro.

Ejemplo: ./calcula_cambio.sh 467
Indique el dinero pagado: 500
El cambio son 33 euros, debe entregar 0 billetes de 50 euros, 1 billete de 20 euros, 1 billete de 10 euros, 0 billetes de 5 euros, 1 moneda de 2 euros y 1 moneda de 1 euro.

CONDICIONALES

Añade comprobación de argumentos en los ejercicios anteriores:

- En los ejercicios impares (1, 3 y 5): si el usuario no indica alguno de los argumentos, se le pedirá por teclado
- En los ejercicios pares (2, 4 y 6): todos los argumentos son obligatorios, si el usuario no especifica exactamente el número de argumentos correcto, el script debe mostrar un mensaje de error y detener la ejecución.
- En el ejercicio 3, permitir que el usuario pueda indicar el mes de forma numérica (1, 2, 3...) o bien en texto: Jan o January, Feb o February, etc. (puede que algunos sistemas esté en español).

7) Crea un script llamado **ej07-compara.sh** al que se le pasen dos números como argumentos (si no se le indica alguno, lo pedirá luego por teclado). El script nos debe decir cuál de los dos números es mayor, o si son iguales.

8) Haz un script con nombre **ej08-imc.sh** que calcule el IMC (Índice de Masa Corporal), teniendo en cuenta que:

- Se deben indicar dos parámetros (números enteros) OBLIGATORIOS: la altura (en centímetros) y el peso (en Kg).
- El IMC viene dado por la fórmula (peso en Kg y altura en cm) -> IMC = 10000*peso / altura2
- Además de mostrar el valor del IMC, el script también indicará la clasificación (delgadez severa, delgadez moderada, delgadez leve, normal, preobesidad, etc.), según la tabla de la OMS.

9) Crear un script llamado **ej09-info_ruta.sh** para mostrar información de los ficheros, directorios, etc. Recibe un argumento y se debe indicar la siguiente información:

1. Si se ha indicado un argumento o no (si no se ha indicado, se muestra mensaje de error y se aborta).
2. Si el argumento indicado existe o no en el disco (si no existe, se muestra mensaje de error y se aborta).
3. Si existe, si es un fichero, directorio, enlace simbólico o "tipo especial".
4. Si tiene o no cada uno de los permisos (lectura, escritura y/o ejecución).
5. Si está vació o no.

10) Haz un script con nombre **ej10-menu_sistema.sh** que muestre un menú que acepte las siguientes opciones:

- *porlibre* o 1: Indica el espacio libre de la partición raíz (en porcentaje).
- *Tamlibre* o 2: Indica el espacio libre de la partición raíz (en tamaño).
- *Usuario* o 3: Indica el usuario actual.
- *Maquina* o 4: Indica el nombre de la máquina.
- *Usuarios* o 5: Indica el número de usuarios del sistema.
- *Espacio* o 6: Total de espacio usado en todos mis directorios personales (en formato "humano": M, G, ...)
- Para cualquier otro valor, mostrar mensaje de error diciendo que no es una opción válida

11) Crea un script llamado **ej11-calculadora.sh** que acepte dos números como argumento (si no se introduce alguno, se pedirá por teclado) y que luego muestre un menú (usando **select**) para realizar 6 operaciones (suma, resta, multiplicación, división y dos más que añadas) entre estos dos números, mostrando el resultado al final.

12) Haz un script con nombre **ej12-cuadrados.sh** que reciba dos parámetros (si no los recibe, los pedirá por teclado hasta que el usuario los introduzca). El script devolverá los valores PARES situados entre esos dos números, y su cuadrado (por ello, deberá comprobar que el segundo número es mayor que el primero, y dar un error y abortar si no es así).

13) Crear un script llamado **ej13-conjuntos.sh** que almacenará conjuntos (varios números) de las tres siguientes formas:

- Conjunto A, por argumentos: se almacenarán los valores que el usuario haya indicado por argumentos, pero en ORDEN INVERSO.
- Conjunto B, por teclado: se le pedirá al usuario que vaya indicando valores por teclado y se parará cuando el usuario indique un 0.
- Conjunto C, generados: se le pedirá al usuario que indique un número de valores a generar por teclado, que por defecto será 30 (si el usuario no indica ningún número). Luego se irán generando números, se cogerá el valor de la iteración si es impar, o un número aleatorio si es par. Cada nuevo número generado se almacenará EN MEDIO de los demás.
- Una vez generados todos los conjuntos, para cada uno de ellos se mostrarán los valores mínimo, máximo y valor medio.
- Al final, mostrar valor mínimo, máximo y medio de un super conjunto formado por los tres conjuntos iniciales.

MODIFICAR LOS SIGUIENTES SCRIPTS PARA AÑADIR ARRAYS, BUCLES Y/O FUNCIONES:

6) El cálculo del cambio se hará mediante bucles y un array *dinero* donde estarán los valores de billetes y monedas (no hace falta distinguir entre billetes y monedas): 500, 200, 100, 50, 20, 10, 5, 2, 1. Solo mostrar las cantidades que hay que devolver, si da 0 no se muestra (es decir, NO mostrar mensajes como 0 de 200e).

9) Si el parámetro indicado es un directorio, mostrará la información de cada elemento que esté dentro de ese directorio (sólo el primer nivel).

11) El nombre de las operaciones deben almacenarse en un array (el menú se construirá directamente a partir de este array con el select). Cada operación se implementará como una función.

EXTRA (OPCIONAL): Tomar como base el script 8 (cálculo del IMC) y crear un script 8B que en vez de números enteros, calcule el IMC con valores decimales.

==========

REPETIR LOS SIGUIENTES SCRIPTS EN BATCH (MS WINDOWS)

1)

2)

3)

7)

8)

12)