



Utilidades de uso frecuente



UTILIDADES DE USO FRECUENTE - GREP

1. Descarga el fichero *Partida.java* de la URL siguiente:

<https://github.com/deividgr/othello/blob/master/Partida.java>

```
alberto@usuario-VirtualBox:~/Desktop$ nano Partida.java
alberto@usuario-VirtualBox:~/Desktop$ cat Partida.java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package othello;

import java.io.Serializable;

/**
 * Clase que contiene un dato miembro de la clase Tablero
 * que almacena la dificultad elegida por el jugador. Esta
 * guardar en un archivo de forma conjunta un tablero y el
 * nivel que se pueda retomar mas adelante.
 */
```

y utiliza grep para obtener la siguiente información:

1.a) [1] Las líneas del fichero que contienen la palabra “tablero”

```
alberto@usuario-VirtualBox:~/Desktop$ grep "tablero" Partida.java
 * guardar en un archivo de forma conjunta un tablero y el nivel que habia elegido el usuario, d
e
 private TableroOtelo tablero;
 * @param tablero Tablero logico.
 public Partida(TableroOtelo tablero, int dificultad){
     this.tablero = tablero;
     * Metodo que devuelve el objeto asociado al dato miembro tablero.
     return this.tablero;
alberto@usuario-VirtualBox:~/Desktop$
```

1.b) [1] Las líneas del fichero que no contienen la palabra “tablero”

```
alberto@usuario-VirtualBox:~/Desktop$ grep -v "tablero" Partida.java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package othello;

import java.io.Serializable;

/**
```

- 1.c) [1] Cuenta las ocurrencias del símbolo “;” utilizando la combinación *cat | grep*

```
alberto@usuario-VirtualBox:~/Desktop$ cat Partida.java | grep -c ";"
8
alberto@usuario-VirtualBox:~/Desktop$
```

- 1.d) [1] Cuenta las ocurrencias de la palabra “private” utilizando la combinación *cat | grep*

```
alberto@usuario-VirtualBox:~/Desktop$ cat Partida.java | grep -c "private"
2
alberto@usuario-VirtualBox:~/Desktop$
```

- 1.e) [1] Las líneas del fichero con su número de línea que contienen el símbolo “=”

```
alberto@usuario-VirtualBox:~/Desktop$ grep -n "=" Partida.java
38:     this.tablero = tablero;
39:     this.dificultad = dificultad;
alberto@usuario-VirtualBox:~/Desktop$
```

- 1.f) [3] Las líneas del fichero con su número de línea que contienen el símbolo “=” Y que, además, contengan la palabra “this”. Para conseguirlo, necesitas encadenar dos grep seguidos utilizando tuberías.

```
alberto@usuario-VirtualBox:~/Desktop$ grep -n "=" Partida.java | grep "this"
38:      this.tablero = tablero;
39:      this.dificultad = dificultad;
alberto@usuario-VirtualBox:~/Desktop$
```

- 1.g) [5] Encadena tres grep seguidos para ver las líneas del fichero que contengan los siguientes elementos: ";" y "(" y "=".

```
alberto@usuario-VirtualBox:~/Desktop$ grep ";" Partida.java | grep "(" | grep "="
alberto@usuario-VirtualBox:~/Desktop$
```

2. [2] Uno de los usos de grep consiste en sacar rápidamente información de un fichero de log en los que se guardan los eventos de cierta parte del sistema. Tomando el fichero /var/log/boot.log que nos muestra la información relativa al último arranque del sistema, busca las líneas en las que algún proceso ha fallado ("FAILED") imprimiendo también los números de líneas.

```
alberto@usuario-VirtualBox:~/Desktop$ sudo grep -ni "FAILED" /var/log/boot.log
6:          Starting GRUB failed boot detection...
8:[ OK ] Finished GRUB failed boot detection.
136:          Starting GRUB failed boot detection...
137:[ OK ] Finished GRUB failed boot detection.
261:          Starting GRUB failed boot detection...
262:[ OK ] Finished GRUB failed boot detection.
alberto@usuario-VirtualBox:~/Desktop$
```

3. [5] Visualiza el nombre de todos los ficheros de la carpeta /etc en los que aparezca tu nombre de usuario.

```
alberto@usuario-VirtualBox:~/Desktop$ ls -l /etc/ | grep -ri "alberto"
final.txt: 3922 pts/0    R+    0:00 ps ax -u alberto
final.txt:alberto    1259  0.0  0.2 19256 10576 ?        Ss   10:07
d --user
final.txt:alberto    1260  0.0  0.0 103480  3488 ?        S    10:07
final.txt:alberto    1266  0.0  0.4 1679580 19708 ?        S<sl 10:07
--daemonize=no --log-target=journal
final.txt:alberto    1269  0.0  0.6 520160  24832 ?        SNsl 10:07
r-miner-fs
final.txt:alberto    1271  0.0  0.1 248812  7580 ?        Sl   10:07
```

4. Tomando el fichero *Partida.java* que descargaste en el primer ejercicio y utiliza grep para obtener la siguiente información:

- 4.a) [1] Las líneas que contengan el texto "in" seguido de cualquier carácter.

```
alberto@usuario-VirtualBox:~/Desktop$ grep "in." Partida.java
* and open the template in the editor.
 * Variable de tipo entera con informacion sobre la dificultad elegida.
private int dificultad;
public Partida(TableroOtelo tablero, int dificultad){
public int getDificultad(){
```

- 4.b) [1] Las líneas que contengan cualquier carácter seguido por el texto “ate” y por otro carácter más.

```
alberto@usuario-VirtualBox:~/Desktop$ grep ".ate." Partida.java
* To change this template, choose Tools | Templates
* and open the template in the editor.
private TableroOtelo tablero;
private int dificultad;
alberto@usuario-VirtualBox:~/Desktop$
```

- 4.c) [1] Las líneas que contengan una “c” seguida por cualquier carácter.

```
alberto@usuario-VirtualBox:~/Desktop$ grep "c." Partida.java
* To change this template, choose Tools | Templates
package othello;
* Clase que contiene un dato miembro de la clase TableroOtelo, y otro de tipo primitivo
* que almacena la dificultad elegida por el jugador. Esta clase se utilizara para poder
* guardar en un archivo de forma conjunta un tablero y el nivel que habia elegido el usuario, d
e
public class Partida implements Serializable{
    * Dato miembro de la clase TableroOtelo.
    * Variable de tipo entera con informacion sobre la dificultad elegida.
    private int dificultad;
    * Constructor de la clase que recibe como argumentos un objeto de la clase TableroOtelo y
```

- 4.d) [2] Las líneas que contengan una “c” seguida por cualquier carácter, pero al principio de la línea.

```
alberto@usuario-VirtualBox:~/Desktop$ grep "^c." Partida.java
alberto@usuario-VirtualBox:~/Desktop$
```

4.e) [2] Las líneas que terminen en “();”

```
alberto@usuario-VirtualBox:~/Desktop$ grep "();$" Partida.java
alberto@usuario-VirtualBox:~/Desktop$
```

5. Con **ls -l** obtenemos una información detallada de los ficheros de una carpeta. Toma la salida de este comando, aplicado sobre tu carpeta de usuario, para encadenarlo con un **grep** que nos busque la siguiente información:

```
alberto@usuario-VirtualBox:~$ pwd
/home/alberto
alberto@usuario-VirtualBox:~$ ls -l
total 52
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Documents
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Downloads
-rw-rw-r-- 1 alberto alberto    29 ene 24 12:17 fecha_hoy
-rwxrw-r-- 1 alberto alberto     0 ene 13 10:56 Libro.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Music
-rw-r--r-- 1 john    john      0 ene 13 10:56 NoMeLeas.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Pictures
-rwxrw-r-- 1 alberto alberto     0 ene 13 10:56 Post-it.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Public
drwxrwxrwx 2 alberto alberto 4096 ene 13 10:57 Recetas
drwxrwxr-x 2 alberto alberto 4096 ene 18 16:06 redireccionamiento
drwx----- 3 alberto alberto 4096 ene 13 18:54 snap
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Templates
-rw-rw-r-- 1 alberto alberto 164 ene 24 12:06 usuarios.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Videos
alberto@usuario-VirtualBox:~$
```

5.a) [1] Las líneas cuya hora esté comprendida entre las 11:XX y las 15:XX.

```
alberto@usuario-VirtualBox:~$ ls -l | grep "[11,12,13,14,15]:[0-59]"
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
-rw-rw-r-- 1 alberto alberto    29 ene 24 12:17 fecha_hoy
-rw-rw-r-- 1 alberto alberto   164 ene 24 12:06 usuarios.txt
```

5.b) [1] Las líneas cuya hora contenga un número par de minutos, es decir, que los minutos acaban en 0 ó 2 ó 4 ó 6 ó 8.

```
alberto@usuario-VirtualBox:~$ ls -l | grep "[0-2][0-9]:[0-5][0,2,4,6,8]"
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Documents
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Downloads
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Libro.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Music
-rw-r--r-- 1 john    john     0 ene 13 10:56 NoMeLeas.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Pictures
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Post-it.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Public
drwxrwxr-x 2 alberto alberto 4096 ene 18 16:06 redireccionamiento
drwx----- 3 alberto alberto 4096 ene 13 18:54 snap
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Templates
-rw-rw-r-- 1 alberto alberto 164 ene 24 12:06 usuarios.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Videos
alberto@usuario-VirtualBox:~$
```

5.c) [1] Las líneas que se corresponden con un directorio (fíjate en el comienzo de la línea)

```
alberto@usuario-VirtualBox:~$ ls -l | grep "^\d"
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Documents
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Downloads
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Music
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Pictures
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Public
drwxrwxrwx 2 alberto alberto 4096 ene 13 10:57 Recetas
drwxrwxr-x 2 alberto alberto 4096 ene 18 16:06 redireccionamiento
drwx----- 3 alberto alberto 4096 ene 13 18:54 snap
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Templates
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Videos
alberto@usuario-VirtualBox:~$
```

5.d) [3] Las líneas que tengan algún permiso del archivo/carpeta que se permita tanto leer como escribir como ejecutar.

```
alberto@usuario-VirtualBox:~$ ls -l | grep "rwx"
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Documents
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Downloads
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Libro.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Music
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Pictures
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Post-it.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Public
drwxrwxrwx 2 alberto alberto 4096 ene 13 10:57 Recetas
drwxrwxr-x 2 alberto alberto 4096 ene 18 16:06 redireccionamiento
drwx----- 3 alberto alberto 4096 ene 13 18:54 snap
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Templates
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Videos
alberto@usuario-VirtualBox:~$ █
```

5.e) [5] Las líneas que tengan algún permiso del archivo/carpeta que se permita tanto leer como escribir como ejecutar y que además el mes de creación del fichero sea enero “ene”

```
alberto@usuario-VirtualBox:~$ ls -l | grep "rwx" | grep "ene"
drwxr-xr-x 2 alberto alberto 4096 ene 24 15:52 Desktop
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Documents
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Downloads
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Libro.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Music
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Pictures
-rwxrw-r-- 1 alberto alberto 0 ene 13 10:56 Post-it.txt
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Public
drwxrwxrwx 2 alberto alberto 4096 ene 13 10:57 Recetas
drwxrwxr-x 2 alberto alberto 4096 ene 18 16:06 redireccionamiento
drwx----- 3 alberto alberto 4096 ene 13 18:54 snap
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Templates
drwxr-xr-x 2 alberto alberto 4096 ene 12 10:40 Videos
```

UTILIDADES DE USO FRECUENTE - SORT

1. Utiliza **sort** en combinación con otros comandos para obtener la siguiente información:

1.a) [0,5] Muestra el contenido del fichero /etc/protocols ordenado alfabéticamente

```
alberto@usuario-VirtualBox:~$ sort /etc/protocols

#
#      99                      # any private encryption scheme
ah      51      IPSEC-AH        # Authentication Header [RFC2402]
# assigned by IANA and are not historical.
ax.25   93      AX.25          # AX.25 frames
dccp   33      DCCP            # Datagram Congestion Control Prot. [RFC4340]
ddp     37      DDP             # Datagram Delivery Protocol
egp     8       EGP             # exterior gateway protocol
```

- 1.b) [1] Vuelca a un fichero llamado /home/tu-usuario/protocolos.txt el contenido del fichero /etc/protocols ordenado alfabéticamente.

```
alberto@usuario-VirtualBox:~$ sort /etc/protocols > protocols.txt
alberto@usuario-VirtualBox:~$
```

```
alberto@usuario-VirtualBox:~$ cat protocols.txt

#
#      99                      # any private encryption scheme
ah      51          IPSEC-AH      # Authentication Header [RFC2402]
# assigned by IANA and are not historical.
ax.25   93          AX.25        # AX.25 frames
```

- 1.c) [1] Vuelca a un fichero llamado /home/tu-usuario/grupos.txt el contenido del fichero /etc/group ordenado alfabéticamente.

```
alberto@usuario-VirtualBox:~$ sort /etc/group > grupos.txt
alberto@usuario-VirtualBox:~$ cat grupos.txt
adm:x:4:syslog,usuario
alberto:x:1004:
audio:x:29:pulse
```

- 1.d) [1] Lista por orden alfabético los archivos contenidos en la carpeta /etc

```
alberto@usuario-VirtualBox:~$ ls -la /etc | sort
drwxr-s---  2 root dip    4096 ago 19 12:40 chatscripts
drwxr-xr-x  11 root root  4096 ago 19 12:42 X11
drwxr-xr-x 131 root root 12288 ene 24 12:35 .
```

- 1.e) [1] Lista por orden alfabético los dispositivos conectados al sistema.

```
alberto@usuario-VirtualBox:~$ ls /dev | sort
autofs
block
bsg
btrfs-control
```

- 1.f)[1] Lista por orden alfabético los dispositivos montados en el sistema mostrando su espacio libre en formato humano.

```
alberto@usuario-VirtualBox:~$ df -h | sort
/dev/loop0      128K  128K      0 100% /snap/bare/5
/dev/loop10     55M   55M      0 100% /snap/snap-store/558
```

1.g) [1] Muestra el contenido del fichero /etc/protocols ordenado alfabéticamente

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | sort

#
#      99          # any private encryption scheme
ah      51          IPSEC-AH      # Authentication Header [RFC2402]
```

1.h) [2] Lista en orden alfabético inverso los archivos contenidos en la carpeta /bin

```
alberto@usuario-VirtualBox:~$ ls /bin/ | sort -r
znew
zmore
zless
zjsdecode
```

2. Utiliza **sort** en combinación con otros comandos para obtener la siguiente información:

2.a) [0,5] Muestra el contenido del fichero /etc/protocols ordenado alfabéticamente por la segunda columna.

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | sort -k2

#
hopopt  0        HOPOPT          # IPv6 Hop-by-Hop Option [RFC1883]
ip       0        IP              # internet protocol, pseudo protocol number
pim      103      PIM             # Protocol Independent Multicast
ipcomp   108      IPCOMP          # IP Payload Compression Protocol
vrrp     112      VRRP            # Virtual Router Redundancy Protocol [RFC5798]
l2tp     115      L2TP             # Layer Two Tunneling Protocol [RFC2661]
```

2.b) [1] Muestra el contenido del fichero /etc/protocols en orden alfabético inverso por la quinta columna.

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | sort -rk5

encap    98      ENCAP           # Yet Another IP encapsulation [RFC1241]
xtp      36      XTP             # Xpress Transfer Protocol
xns-idp  22      XNS-IDP         # Xerox NS IDP
wesp    141      WESP            # Wrapped Encapsulating Security Payload
```

2.c) [1] Muestra el contenido del fichero /etc/protocols ordenado numéricamente por la segunda columna.

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | sort -nk2

#
# assigned by IANA and are not historical.
hopopt  0        HOPOPT          # IPv6 Hop-by-Hop Option [RFC1883]
```

2.d) [1] Muestra el contenido del fichero /etc/protocols en orden numérico inverso por la segunda columna.

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | sort -nrk2
rohc    142      ROHC          # Robust Header Compression
wesp    141      WESP          # Wrapped Encapsulating Security Payload
shim6   140      Shim6         # Shim6 Protocol [RFC5533]
```

2.e) [1] Filtra el contenido del fichero /etc/protocols para que no salgan las líneas que comienzan por #. La salida de este filtrado debe ordenarse alfabéticamente.

```
alberto@usuario-VirtualBox:~$ cat /etc/protocols | grep -v "^#" | sort
ah      51       IPSEC-AH      # Authentication Header [RFC2402]
ax.25   93       AX.25        # AX.25 frames
dccp   33       DCCP          # Datagram Congestion Control Prot. [RFC4340]
```

3. Utiliza **sort** en combinación con otros comandos para obtener la siguiente información:

3.a) [1] Muestra los procesos del sistema con todos sus detalles ordenándolos por el nombre del usuario propietario del proceso.

```
alberto@usuario-VirtualBox:~$ ps -aux | sort -k1
alberto      1286  0.0  0.3  21464 12868 ?
alberto      1287  0.0  0.0  103400  3488 ?
alberto      1293  0.0  0.4  1417436 19404 ?
```

3.b) [1] Filtra lo obtenido en el apartado anterior para que solo se muestren las líneas asociadas a tu nombre de usuario.

```
alberto@usuario-VirtualBox:~$ ps -aux | sort -k1 | grep "alberto"
alberto      1250  0.1  0.2  20744 11948 ?          Ss 14:08  0:00
alberto      1252  0.0  0.0  169008  3476 ?          S 14:08  0:00
alberto      1258  0.2  0.4  1491464 19792 ?          S<sl 14:08  0:00
ze=no --log-target=journal
```

3.c) [1] Muestra los procesos del sistema con todos sus detalles ordenándolos numéricamente por la columna PPID.

```
alberto@usuario-VirtualBox:~$ ps -ejf | sort -nk3
root        1      0      1      1  0 14:08 ?          00:00:01 /sbin/init
root        2      0      0      0  0 14:08 ?          00:00:00 [kthrea
UID        PID      PPID     PGID     SID  C STIME TTY          TIME CMD
```

- 3.d) [1] Filtra lo obtenido en el apartado anterior para que solo se muestren las líneas asociadas a tu nombre de usuario y vuélcalo a un fichero llamado *mis procesos.txt*.

```
alberto@usuario-VirtualBox:~$ ps -ejf | sort -nk3 | grep "alberto" >> misprocesos.txt
alberto@usuario-VirtualBox:~$
```

```
alberto@usuario-VirtualBox:~$ cat misprocesos.txt
alberto    1250      1    1250    1250  0 14:08 ?
alberto    1262      1    1261    1261  0 14:08 ?
n --daemonize --login
alberto    1426      1    1424    1424  0 14:08 ?
ard
alberto    1438      1    1436    1436  0 14:08 ?
```

- 3.e) [2] Muestra el contenido del fichero /etc/group, ordenándolo por la tercera columna y tomando como separador el carácter :

```
alberto@usuario-VirtualBox:~$ cat /etc/group | sort -k3 -t:
root:x:0:
daemon:x:1:
uucp:x:10:
users:x:100:
```

- 3.f)[1] Filtra el contenido del fichero /etc/group para obtener solo las líneas que contengan tu nombre de usuario. El resultado debes ordenarlo numéricamente por la tercera columna tomando como separador el carácter :

```
alberto@usuario-VirtualBox:~$ cat /etc/group | grep "alberto" | sort -nk3 -t:
sudo:x:27:alberto
alberto:x:1004:
```

- 3.g) [2] Muestra el contenido del fichero /etc/services, filtrando para quedarnos con las líneas que contengan “tcp” y ordénalas.

```
alberto@usuario-VirtualBox:~$ cat /etc/services | grep "tcp" | sort
acr-nema      104/tcp          dicom      # Digital Imag. & Comm. 300
afbackup      2988/tcp          # Afbackup system
afmbackup     2989/tcp          # Afmbackup system
```

3.h) [2] Muestra el contenido del fichero /etc/services, filtrando para quedarnos con las líneas que contengan “tcp” y ordénalas.

```
alberto@usuario-VirtualBox:~$ cat /etc/services | grep "tcp" | sort
acr-nema      104/tcp        dicom      # Digital Imag. & Comm. 300
afbackup      2988/tcp       # Afbackup system
afmbackup     2989/tcp       # Afmbackup system
```

3.i) [3] Muestra el contenido del fichero /etc/services, filtrando para quedarnos con las líneas que contengan “udp” y ordénalas numéricamente por la segunda columna.

```
alberto@usuario-VirtualBox:~$ cat /etc/services | grep "udp" | sort -nk2
echo          7/udp
discard      9/udp        sink null
```

3.j)[5] Muestra el contenido del fichero /etc/services, filtrando para quedarnos con las líneas que contengan “udp”, ordénalas numéricamente por la segunda columna y el resultado obtenido debes volcarlo a un fichero llamado *servicios-udp.txt*.

```
alberto@usuario-VirtualBox:~$ cat /etc/services | grep "udp" | sort -nk2 >> servicios-udp.txt
alberto@usuario-VirtualBox:~$
```

```
alberto@usuario-VirtualBox:~$ cat servicios-udp.txt
echo          7/udp
discard      9/udp        sink null
daytime      13/udp
chargen     19/udp        ttyst source
```

UTILIDADES DE USO FRECUENTE – TAR + GZIP

1. Utiliza **gzip** para realizar las siguientes acciones:

1.a) [0,5] Crea un fichero de texto llamado mitexto.txt y editalo. A continuación, pega al menos 4 párrafos del siguiente generador de textos de muestra: <https://getlorem.com/es/>. Por último, graba y cierra el fichero.

```
alberto@usuario-VirtualBox:~$ touch mitexto.txt
alberto@usuario-VirtualBox:~$
```

```
GNU nano 4.8                                     mitexto.txt
Lorem ipsum dolor sit amet consectetur adipiscing elit sapien rhon
Luctus aliquam justo mollis velit arcu sollicitudin natoque, enim
Magna ac netus iaculis aptent suscipit donec duis habitasse hendre
ctus gravida senectus ultrices lobortis vulputate dui potenti.■
```

```
alberto@usuario-VirtualBox:~$ cat mitexto.txt
Lorem ipsum dolor sit amet consectetur adipiscing e
i netus elementum maecenas sed himenaeos mattis int
asse quam primis erat. Eleifend curabitur ligula du
vivamus leo quis suscipit facilisi lobortis, suspe
```

1.b) [1] Comprime el fichero anterior, pero conservando la copia original.

```
alberto@usuario-VirtualBox:~$ gzip -k mitexto.txt
alberto@usuario-VirtualBox:~$ ls
Desktop    grupos.txt      mitexto.txt      Pictures      Public      snap
Documents  Libro.txt       mitexto.txt.gz   Post-it.txt   Recetas     Templates
Downloads  misprocesos.txt Music        protocolos.txt redireccionamiento usuarios.txt
fecha_hoy   mitexto.ts     NoMeLeas.txt   protocols.txt servicios-udp.txt Videos
alberto@usuario-VirtualBox:~$
```

1.c) [3] Utiliza la fórmula siguiente para calcular en qué porcentaje se ha reducido el tamaño del fichero al ser comprimido. Adjunta capturas de pantalla para atestiguar el tamaño del fichero.

$$\text{Porcentaje Compresión} = \left(100 - \frac{\text{tamañoComprimido} \cdot 100}{\text{tamañoSinComprimir}} \right)$$

```
alberto@usuario-VirtualBox:~$ ls -lh | grep "mitexto."
-rw-rw-r-- 1 alberto alberto 2,8K ene 29 11:04 mitexto.txt
-rw-rw-r-- 1 alberto alberto 1,2K ene 29 11:04 mitexto.txt.gz
alberto@usuario-VirtualBox:~$
```

```
alberto@usuario-VirtualBox:~$ ls -l | grep "mitexto."
-rw-rw-r-- 1 alberto alberto 2867 ene 29 11:04 mitexto.txt
-rw-rw-r-- 1 alberto alberto 1200 ene 29 11:04 mitexto.txt.gz
alberto@usuario-VirtualBox:~$
```

(En una captura se ve el tamaño exacto, y en la otra la unidad de medida: kilobytes)

$$\text{Porcentaje Compresión} = 100 - (1200 * 100) / 2867 = 58,14\%$$

1.d) [1] A continuación, descarga una imagen en formato .jpg de internet y comprímela conservando el original.

```
alberto@usuario-VirtualBox:~/Downloads$ gzip -k goodmorning.jpg
alberto@usuario-VirtualBox:~/Downloads$ ls -l | grep "goodmorning."
-rw-rw-r-- 1 alberto alberto 109206 ene 29 12:16 goodmorning.jpg
-rw-rw-r-- 1 alberto alberto 108922 ene 29 12:16 goodmorning.jpg.gz
alberto@usuario-VirtualBox:~/Downloads$ █
```

- 1.e) [5] Aplica la fórmula anterior y compara los porcentajes de compresión obtenidos en cada caso. ¿Por qué uno es mucho mayor que el otro?

$$\text{Porcentaje Compresión} = 100 - (108922 * 100) / 109206 = 0.26\%$$

La imagen con extensión .jpg obtiene una tasa de compresión mucho más baja que el fichero de texto. Esto ocurre porque las imágenes con esta extensión ya están de por sí comprimidas, al incluir sus propios esquemas de compresión.

2. Utiliza **tar** y/o **gzip** para realizar las siguientes acciones:

- 2.a) [1] Cámbiate a la carpeta Descargas y empaqueta todos los ficheros que haya.

```
alberto@usuario-VirtualBox:~$ cd Downloads
alberto@usuario-VirtualBox:~/Downloads$ tar cvf descargas.tar *
clonetroopers.jpeg
clonetroopers.jpeg.gz
goodmorning.jpg
goodmorning.jpg.gz
alberto@usuario-VirtualBox:~/Downloads$ █
```

- 2.b) [1] A continuación, lista los ficheros que están contenidos en el .tar resultante.

```
alberto@usuario-VirtualBox:~/Downloads$ tar tf descargas.tar
clonetroopers.jpeg
clonetroopers.jpeg.gz
goodmorning.jpg
goodmorning.jpg.gz
alberto@usuario-VirtualBox:~/Downloads$ █
```

- 2.c)[1] Copia el .tar a la carpeta Documentos y “desempaquetá” su contenido.

```
alberto@usuario-VirtualBox:~/Downloads$ cp descargas.tar ../Documents
alberto@usuario-VirtualBox:~/Downloads$ cd ../Documents
alberto@usuario-VirtualBox:~/Documents$ tar xf descargas.tar
alberto@usuario-VirtualBox:~/Documents$ ls
clonetroopers.jpeg  clonetroopers.jpeg.gz  descargas.tar  goodmorning.jpg
alberto@usuario-VirtualBox:~/Documents$ █
```

- 2.d) [1] Vuelve a la carpeta Descargas y elimina un archivo de los que están contenidos en el fichero .tar.

```
alberto@usuario-VirtualBox:~/Documents$ cd .../Downloads
alberto@usuario-VirtualBox:~/Downloads$ tar f descargas.tar --delete clonetroopers.jpeg
alberto@usuario-VirtualBox:~/Downloads$ █
```

- 2.e) [2] Vuelve a añadir el fichero que has borrado al fichero .tar ya existente, sin borrar el contenido previo existente.

```
alberto@usuario-VirtualBox:~/Downloads$ tar f descargas.tar --append clonetroopers.jpeg
alberto@usuario-VirtualBox:~/Downloads$ tar tf descargas.tar
clonetroopers.jpeg.gz
goodmorning.jpg
goodmorning.jpg.gz
clonetroopers.jpeg
```

- 2.f)[2] Sitúate en tu carpeta de usuario y ahora empaqueta y comprime la carpeta /etc (necesitarás permisos de superusuario). Para ellos debes generar en tu carpeta de usuario los ficheros etc.tar y etc.tar.gz.

```
alberto@usuario-VirtualBox:~$ sudo tar cf etc.tar /etc
tar: Removing leading `/' from member names
alberto@usuario-VirtualBox:~$ gzip -vk etc.tar
etc.tar:          81.6% -- created etc.tar.gz
alberto@usuario-VirtualBox:~$
```

```
alberto@usuario-VirtualBox:~$ ls -l etc*
-rw-r--r-- 1 root      root    6901760 ene 29 17:32 etc.tar
-rw-r--r-- 1 alberto   alberto 1271038 ene 29 17:32 etc.tar.gz
```

- 2.g) [3] Crea una carpeta etc dentro de tu carpeta Documentos. Copia el fichero etc.tar.gz a la carpeta Documentos/etc que has creado. Por último, descomprime y desempaquetá el fichero.

```
alberto@usuario-VirtualBox:~/Documents$ mkdir etc
alberto@usuario-VirtualBox:~/Documents$ cp ../etc.tar
etc.tar      etc.tar.gz
alberto@usuario-VirtualBox:~/Documents$ cp ../etc.tar.gz etc
alberto@usuario-VirtualBox:~/Documents$ █
```

```
alberto@usuario-VirtualBox:~/Documents/etc$ gzip -d etc.tar.gz
alberto@usuario-VirtualBox:~/Documents/etc$ tar -xf etc.tar.gz
alberto@usuario-VirtualBox:~/Documents/etc$ ls
etc  etc.tar
alberto@usuario-VirtualBox:~/Documents/etc$
```

3. [3] Investiga qué acción realiza el siguiente comando y explica el resultado obtenido: tar cvfz resultado.tar.gz fichero1 fichero2 fichero3...

El comando sirve para empaquetar (tar) y comprimir (z) directamente los ficheros “fichero1”, “fichero2” y “fichero3”, creando un fichero resultante llamado “resultado.tar.gz” (f), sin modificar o eliminar los archivos originales (c) y mostrando el proceso en la consola (v).