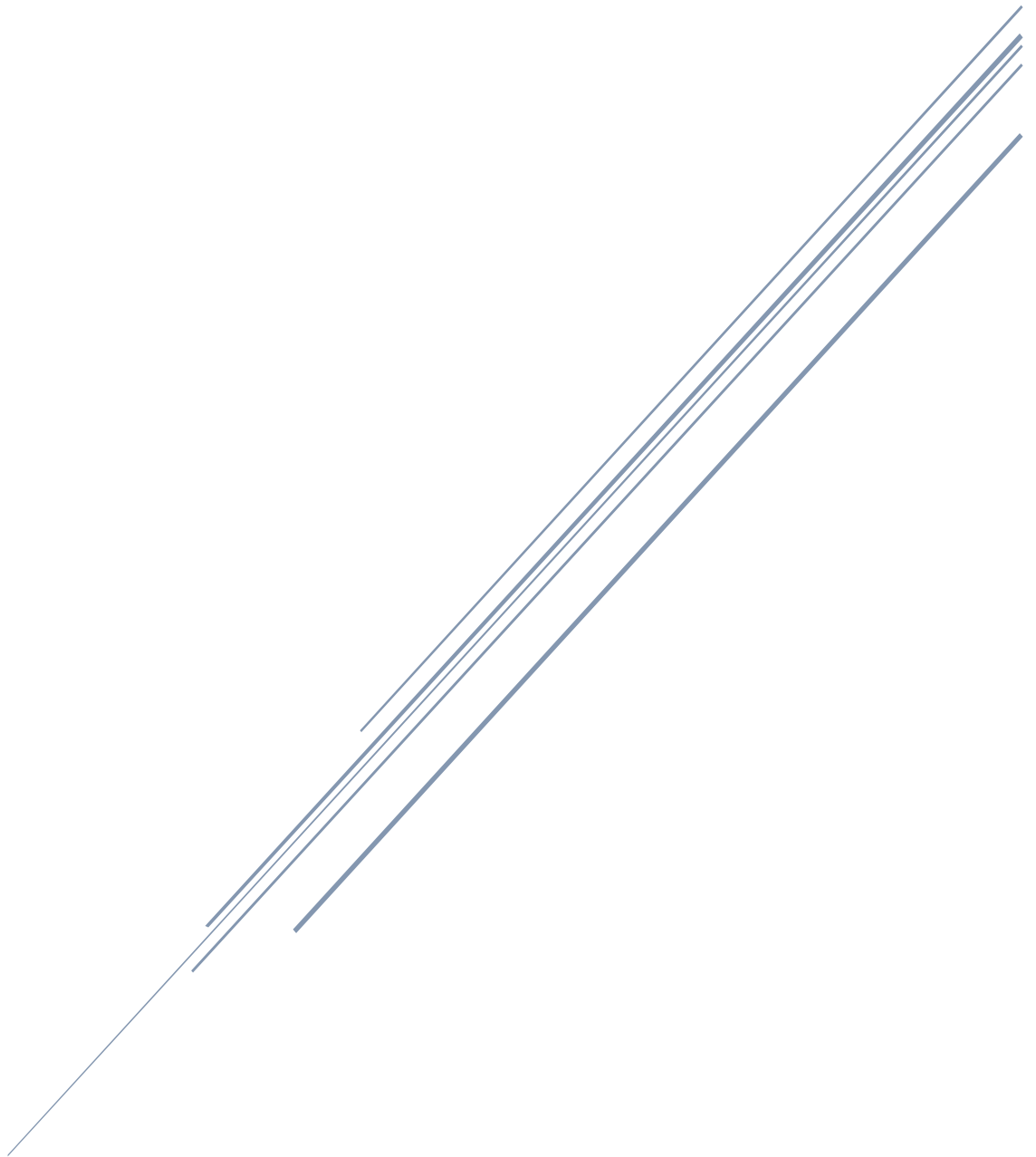


U3 ADMINISTRACIÓN DE SOFTWARE BASE

ACTIVIDAD PRÁCTICA 7: Gestión de Procesos 2



Sergio Cobo García
SISTEMAS INFORMÁTICOS

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

Contenido

Comunicación entre procesos y utilización de filtros.....	2
---	---

Comunicación entre procesos y utilización de filtros

01	Respecto al paginador less , indique cómo se pueden llevar a cabo las acciones indicadas en la mesa de abajo:
----	--

PAGINADOR LESS	
Avanzar página a página	ESPACIO o z
Retroceder una página	b o w
Avanzar línea a línea	ENTER, Flecha hacia abajo, j o e
Retroceder línea a línea	y, Flecha hacia arriba o k
Buscar una palabra, hacia adelante	/palabra (ejemplo: /termino)
Buscar una palabra, hacia atrás	?palabra (ejemplo: ?termino)
Ir a una línea específica	gNumeroLinea (ejemplo: g100)
Salir	q, Q o ZZ

02	En Linux, escriba un comando que muestre ÚNICAMENTE los nombres de los usuarios que tienen un nombre que comienza por 'm'. <i>Pista: Debe trabajar con el archivo /etc/passwd.</i>
----	---

El comando sería **grep "^m" /etc/passwd | cut -d: -f1.**

```
alumno@alumno-VirtualBox:~$ grep "^m" /etc/passwd | cut -d: -f1
man
mail
messagebus
```

03	En Linux, ¿Qué efecto tiene el comando ls -l tee primero.txt sort -r > segundo.txt ? ¿Qué diferencia existe con el comando ls -l tee primero.txt sort -R > segundo.txt ? Es necesario que explique, paso a paso, qué hace cada uno de los comandos.
----	---

El primer comando lo que hace es copiar el resultado de **ls -l** en el archivo **primero.txt**, y luego ordena el resultado a la inversa y lo almacena en otro archivo llamado **segundo.txt**.

```
alumno@alumno-VirtualBox:~/Escritorio/Prueba2$ ls -l | tee
primero.txt | sort -r > segundo.txt
```

Abrir ▾	primero.txt ~/Escritorio/Prueba2
<pre>archivo1.txt archivo2.txt primero.txt segundo.txt</pre>	
Abrir ▾	segundo.txt ~/Escritorio/Prueba2
<pre>segundo.txt primero.txt archivo2.txt archivo1.txt</pre>	

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

La diferencia que hay entre el primer comando y el segundo es que en el segundo sale el comando sort -R, en mayúscula, y eso quiere decir que ordena los elementos de manera aleatorio.

El comando ls -l lista los archivos y directorios en una sola columna en vez de varias.

El comando tee copia la salida del comando anterior en un archivo nuevo o existente llamado primero.txt.

El comando sort -r lo que hace es ordenar los elementos de manera invertida y lo almacena en el archivo segundo.txt.

04	<p>Linux: Escriba un comando que muestre el nombre del segundo archivo más grande del directorio /etc.</p> <p>PowerShell: Escriba un comando que muestre el nombre del segundo archivo más grande del directorio C:\WINDOWS\system32\drivers\etc.</p>
----	---

En Linux:

Comando para mostrar el segundo archivo más grande del directorio /etc.

```
alumno@alumno-VirtualBox:~$ ls -Sr /etc/ | tail -n 2 | head -n 1
ld.so.cache
```

En PowerShell:

```
PS C:\Users\Sergio> Get-ChildItem -Path "C:\WINDOWS\system32\drivers\etc" |
Sort-Object -Property Length -Descending | Select-Object -Skip 1 -First 1
-ExpandProperty Name
lmhosts.sam
```

05	En Linux , escriba un comando que una dos archivos de texto, uno a continuación del otro, en un tercero.
----	---

El comando para unir el contenido de archivos en otro es **cat archivo1.txt archivo2.txt > archivo3.txt**.

```
alumno@alumno-VirtualBox:~/Escritorio/Prueba2$ cat archivo1.txt archivo2.txt >
archivo3.txt
```

06	<p>¿Cómo podríamos copiar un archivo sin utilizar el comando cp, utilizando las redirecciones de la entrada y salida estándar?</p> <p>Copie el archivo C:\WINDOWS\system32\drivers\etc\hosts en su directorio de usuario.</p>
----	---

El comando similar al uso del comando cp sería **cat hosts > rutaCompletaArchivo**.

```
PS C:\WINDOWS\system32\drivers\etc> cat hosts > C:\Users\alumno\Desktop\hosts2.txt
PS C:\WINDOWS\system32\drivers\etc>
```

07	<p>En Linux, ¿Cómo podríamos construir una pipe o tubería () con los redirectores de la entrada y salida estándar (< >)? Hay que utilizar un archivo temporal que debemos borrar después de su uso.</p> <p>Escriba el equivalente a ls -l wc -l.</p>
----	---

La forma sería la siguiente:

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

```
alumno@alumno-VirtualBox:~/Escritorio/prueba$ ls -l > /home/alumno/Escritorio/tmp.txt; wc -l tmp.txt; rm /home/alumno/Escritorio/tmp.txt
wc: tmp.txt: No existe el archivo o el directorio
```

08	En Linux , escriba un comando basado en cat que tenga el mismo comportamiento que nl .
----	---

El comando pertinente es **cat -n archivo1.txt**.

```
alumno@alumno-VirtualBox:~/Escritorio/prueba$ cat -n archivo1.txt
 1  a
 2  b
 3  c
 4  d
```

09	<p>Linux: Escriba un comando que muestre un listado de todos los usuarios del sistema (ÚNICAMENTE el nombre), ordenados por orden alfabético. <i>Pista: Debe trabajar con el archivo /etc/passwd.</i></p> <p>PowerShell: Escriba un comando que muestre un listado de todos los procesos del sistema (ÚNICAMENTE el nombre), ordenados por número de PID.</p>
----	---

En Linux:

El comando **cut -d: -f1 /etc/passwd** muestra un listado de los usuarios del sistema.

```
alumno@alumno-VirtualBox:~/Escritorio/prueba$ cut -d: -f1 /etc/passwd | sort -d
alumno
_apt
avahi
avahi-autoipd
backup
bin
colord
cups-pk-helper
daemon
dnsmasq
fwupd-refresh
```

En PowerShell:

El comando para mostrar el listado de todos los procesos del sistema es **Get-Process | Sort-Object -Property Id | Select-Object -Property ProcessName**.

```
PS C:\Users\Sergio> Get-Process | Sort-Object -Property Id | Select-Object -Property ProcessName

ProcessName
-----
Idle
System
Secure System
Registry
smss
svchost
svchost
```

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

10	<p>Linux: Intente crear un directorio en la raíz de su sistema (/) y envíe la salida de error de este comando a un archivo de su directorio de usuario. ¿Qué ocurre? Pruebe el mismo con la salida estándar, ¿qué ocurre? ¿Cómo podríamos hacer que esta salida de error NO se mostrara por pantalla ni se guardas en ningún archivo?</p> <p>PowerShell: Intente listar un archivo que no exista y almacene la salida de error a un archivo 'error.txt'. ¿Cómo podríamos hacer que esta salida de error NO se mostrara por pantalla ni se guardas en ningún archivo?</p>
----	---

En Linux:

```
alumno@alumno-VirtualBox:/$ mkdir directorioPrueba 2> /home/alumno/Escritorio/prueba/d.txt
```



No tengo permisos para crear la carpeta ya que se trata de la raíz del sistema y no está permitido.

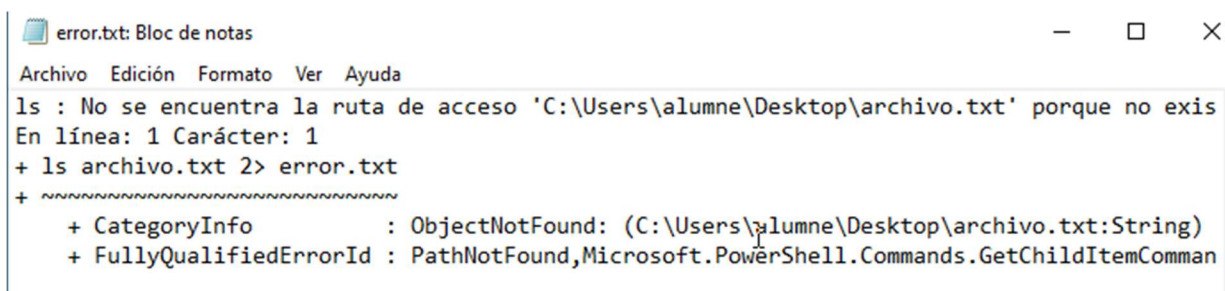
Con la salida estándar me sigue diciendo que me ha denegado la operación, pero tampoco me añade el error al archivo que le he indicado, ya que la redirección usada no lo hace. Crea el archivo, pero vacío.

```
alumno@alumno-VirtualBox:/$ mkdir directorioPrueba > tee /home/alumno/Escritorio/prueba/d.txt
bash: tee: Permiso denegado
```

En PowerShell:

Listo un archivo del escritorio y mando la salida de error al archivo error.txt.

```
PS C:\Users\alumno\Desktop> ls archivo.txt 2> error.txt
```



Con **\$null** evitamos que se muestre el error en pantalla y que este se guarde en un archivo.

```
PS C:\Users\alumno\Desktop> ls archivo.txt 2> $null
PS C:\Users\alumno\Desktop>
```

11	<p>Linux: Ejecute un comando que le muestre ÚNICAMENTE las líneas repetidas, de forma consecutiva, del archivo give_it_away.txt.</p>
----	---

	<p>Ejecute un comando que le muestre el número de repeticiones de cada uno de las líneas del archivo give_it_away.txt.</p> <p>PowerShell:</p> <p>Ejecute un comando que muestre todos los tipos de comandos (debe fijarse con la columna 'Source') que encontramos en PowerShell (utilice el comando Get-Command).</p>
--	--

En Linux:

El comando para mostrar las líneas repetidas de forma consecutiva es **sort give_it_a_way.txt | uniq -D**.

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ sort give_it_away.txt | uniq -D
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away give it away give it away now
Give it away now
Give it away now
Give it away now
Give it away now
Give it away now
```

Con la terminación **uniq -c** nos cuenta las veces que están repetidas las líneas.

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ sort give_it_away.txt | uniq -c
15
  1 Blessed with a bucket of lucky mobility
  1 Bob Marley poet and a prophet
  1 Bob Marley taught me how to off it
  1 Bob Marley walkin' like he talk it
  1 Come and drink it up from my fertility
  1 Come on everybody time to deliver
  1 Confide wisely you'll be the wiser
  1 Confide with sly you'll be the wiser
  1 Dancin' down on life with agility
  1 Drinkin' my juice young love chug-a-lug me
  1 Feelin' good my brother gonna hug me
  1 Get smart get down with the pow wow
16 Give it away give it away give it away now
  1 Give it away No-ooo-wway
20 Give it away now
  1 Goodness me can't you see I'm gonna cough it
```

En PowerShell:

El comando **Get-Command** muestra todos los comandos.


```
PS C:\Users\alumne> Get-Command | more
```

CommandType	Name	Version	Source
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-ProvisionedAppPackage	3.0	Dism
Alias	Add-ProvisionedAppxPackage	3.0	Dism
Alias	Add-ProvisioningPackage	3.0	Provisioning
Alias	Add-TrustedProvisioningCertificate	3.0	Provisioning
Alias	Apply-WindowsUnattend	3.0	Dism
Alias	Disable-PhysicalDiskIndication	2.0.0.0	Storage
Alias	Disable-StorageDiagnosticLog	2.0.0.0	Storage
Alias	Dismount-AppPackageVolume	2.0.1.0	Appx

12	En Linux , utilice el comando paste para unir tres archivos. Haga que todos los tabuladores devueltos se conviertan en espacios. Realice toda esta interacción en una única ejecución.
----	---

Primero muestro el contenido que tiene cada uno de los tres archivos.

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ cat archivo1.txt
hola 1
```

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ cat archivo2.txt
que      ????
```

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ cat archivo3.txt
tal?  !
```

Al usar el siguiente comando me junta el contenido de los archivos y me cambia las tabulaciones por un espacio.

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ paste archivo1.txt archivo2.txt ar
chivo3.txt | tr '\t' ' '
hola 1 que      ???? tal? !
```

13	En Linux , ¿Qué opción del comando tr permite comprimir caracteres repetidos consecutivamente en uno solo? Pruebe este comando con las "o" de la siguiente cadena "boomboompooooowoooooot". ¿Cómo lo haríamos si en vez de querer comprimir caracteres repetidos lo hiciéramos para comprimir espacios consecutivos (comprimir pues los espacios de una salida del comando 'ls -l')?
----	---

El comando para comprimir en uno los caracteres repetidos consecutivamente es el siguiente.

```
alumne@alumne-VirtualBox:~/Escritorio/prueba$ echo "boomboompooooowoooooot" | tr
-s "o"
bombpowot
```

Para en vez de hacerlo con caracteres hacerlo con espacios sería de la siguiente manera.

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

```

alumne@alumne-VirtualBox:~$ ls -l | tr -s " "
total 48
-rw-rw-r-- 1 alumne alumne 17 dic 17 00:05 archivo1.txt
-rw-rw-r-- 1 alumne alumne 20 dic 17 00:05 archivo2.txt
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Descargas
drwxrwxr-x 2 alumne alumne 4096 dic 19 22:57 DirectorioPrueba
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Documentos
drwxr-xr-x 7 alumne alumne 4096 ene 20 18:29 Escritorio
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Imágenes
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Música
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Plantillas
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Público
drwx----- 4 alumne alumne 4096 ene 7 01:09 snap
drwxr-xr-x 2 alumne alumne 4096 dic 10 13:21 Vídeos

```

14	En Linux , escriba un comando que, para cada uno de los directorios de la raíz del sistema , cree un archivo de texto con su nombre en el interior (en formato texto). Los archivos de texto generados tendrán que empezar con el prefijo ' carpeta_ '.
----	--

15	<p>En Linux, escriba una línea, para cada apartado, que le permitan:</p> <p>a) Crear un directorio y, si se ha podido crear correctamente, mostrar el mensaje "directorio creado correctamente".</p> <p>b) Crear un directorio y, si no se ha podido crear correctamente, mostrar el mensaje "no se ha podido crear el directorio".</p> <p>Para comprobar el funcionamiento de estos apartados intente crear las carpetas en su directorio de usuario y/o en la raíz del sistema.</p>
----	--

Para crear un directorio y lanzar un mensaje de que se ha realizado correctamente se usa lo siguiente:

```

alumne@alumne-VirtualBox:~/Escritorio$ mkdir prueba2 && echo "directorio creado correctamente"
directorio creado correctamente

```

Para crear un directorio y lanzar un mensaje de que NO se ha realizado correctamente se usa lo siguiente:

```

alumne@alumne-VirtualBox:~/Escritorio$ mkdir prueba3 && echo "no se ha podido crear el directorio"
no se ha podido crear el directorio

```

16	En Linux , ¿Qué efecto tiene el comando echo "-----" cat archivo1 - archivo2 ? ¿Qué efecto tiene el comando cat archivo1 - archivo2 < archivo3 ?
----	--

El comando **echo "-----" | cat archivo1 - archivo2** primero muestra el contenido de archivo1 seguido de una línea y finalmente el contenido del archivo2.

```
a
b
c
d
-----
m
n
b
v
```

El comando **cat archivo1 – archivo2 < archivo3** concatena el contenido de los tres archivos, primero muestra el contenido del archivo1, luego el del archivo3 y finalmente el del archivo2.

```
alumno@alumno-VirtualBox:~/Escritorio/prueba$ cat archivo1.txt - archivo2.txt <
archivo3.txt
a
b
c
d
r
t
y
u
m
n
b
v
```

17	<p>Linux: Los archivos de configuración suelen indicar los comentarios con líneas que comienzan con #. Escriba un comando que muestre todas las líneas de comentarios del archivo grub.cfg (indique la ruta completa). Escriba un comando que muestre el contenido del archivo grub.cfg eliminando los comentarios y también las líneas vacías. Pista: <i>una línea vacía es aquella que no contiene ningún carácter.</i></p> <p>PowerShell: Escriba un comando que muestre el contenido del archivo C:\WINDOWS\system32\drivers\etc\networks eliminando los comentarios y también las líneas vacías.</p>
----	--

En Linux:

El siguiente comando muestra los comentarios del archivo grub.cfg.

```
alumno@alumno-VirtualBox:~$ grep "^#" /boot/grub/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
### BEGIN /etc/grub.d/00_header ###
### END /etc/grub.d/00_header ###
### BEGIN /etc/grub.d/05_debian_theme ###
### END /etc/grub.d/05_debian_theme ###
### BEGIN /etc/grub.d/10_linux ###
### END /etc/grub.d/10_linux ###
### BEGIN /etc/grub.d/10_linux_zfs ###
### END /etc/grub.d/10_linux_zfs ###
### BEGIN /etc/grub.d/20_linux_xen ###
```

El siguiente es el comando para mostrar el contenido del grub.cfg sin los comentarios y sin líneas vacías.

```
alumno@alumno-VirtualBox:~$ grep -v "^#\|^$" /boot/grub/grub.cfg
if [ -s $prefix/grubenv ]; then
  set have_grubenv=true
  load_env
fi
if [ "${initrdfail}" = 2 ]; then
  set initrdfail=
elif [ "${initrdfail}" = 1 ]; then
  set next_entry="${prev_entry}"
  set prev_entry=
  save_env prev_entry
  if [ "${next_entry}" ]; then
```

En PowerShell:

El comando correcto es el siguiente.

```
PS C:\Users\alumno> Get-Content C:\Windows\System32\drivers\etc\networks
| Where-Object {$_. -ne "" -and $_. -notmatch "^#"}
loopback 127
```

18

Tome el archivo de texto **give_it_away.txt** y realice las siguientes acciones (en el mismo orden, comunicadas con tuberías o pipas):

Linux:

1. Elimine las líneas repetidas consecutivas.
2. Cambie todas las letras mayúsculas que haya por minúsculas.
3. Elimine todas las líneas vacías existentes.
4. Ordene las líneas resultantes en orden alfabético inverso.
5. Numere todas las líneas.
6. Quédese sólo con las líneas que van de la 10 a la 40.
7. Elimine la numeración de las líneas que había añadido al paso 5.
8. Reordenar las líneas en orden alfabético por la segunda palabra de cada línea.

PowerShell:

1. Elimine las líneas repetidas consecutivas.
2. Elimine todas las líneas vacías existentes.
3. Numere todas las líneas.
4. Quédese sólo con las líneas que van de la 10 a la 40.
5. Elimine la numeración de las líneas que había añadido al paso 5.

6. Ordene las líneas resultantes en orden alfabético inverso.
Explique cada uno de los pasos que haya utilizado en este comando.

En Linux:

1. **Cat give_it_away.txt**: muestra el contenido del archivo.
2. **Uniq**: elimina las líneas repetidas consecutivamente.
3. **Tr A-Z a-z**: convierte las letras mayúsculas en minúsculas.
4. **Grep -v "^\$"**: elimina las líneas vacías del documento.
5. **Sort -r**: invierte alfabéticamente el orden de las líneas.
6. **Nl**: numera las líneas del documento.
7. **Sed -n '10,40p'**: muestra únicamente las líneas de la 10 a la 40.
8. **Sed 's/^[0-9]*//'**: elimina la numeración de las líneas.
9. **Sort -k2**: reordena las líneas en orden alfabético por la segunda palabra de cada línea.

```
alumno@alumno-VirtualBox: ~/Escritorio/prueba$ cat give_it_away.txt | uniq | tr A-Z a-z | gr
ep -v "^$" | sort -r | nl | sed -n '10,40p' | sed 's/^[0-9]*// ' | sort -k2
39 give it away give it away give it away now
40 give it away give it away give it away now
38 give it away no-ooo-wway
37 give it away now
36 goodness me can't you see i'm gonna cough it
35 greedy little people in a sea of distress
34 his heart is never gonna wither
32 how come everybody wanna keep it like the kaiser
33 how come everybody wanna keep it like the kaiser
30 i can't tell if i'm a king pin or a pauper
31 i can't tell if i'm a king pin or a pauper
29 i can't tell iff i'm a king pin or a pauper
28 i'm a low brow but i rock a little know how
27 keep you warm won't let you shiver
26 keep your more to receive your less
25 long gone are the times when she scrub me
24 love is free love me say hell yes
23 lucky me swimmin' in my ability
22 my mom i love her 'cause she love me
21 never been a better time than right now
20 no time for the piggies or the hoosegow
19 ooh... ooh yeah
17 realize i don't want to be a miser
18 realize i don't want to be a miser
15 reeling with the feeling don't stop continue
16 reeling with the feeling don't stop continue
14 there's a river born to be a giver
13 unimpressed by material excess
11 what i've got you've got to get it put it in you
12 what i've got you've got to get it put it in you
10 what i've got you've got to give it to your daughter
```

En PowerShell:

1. **Get-Content give_it_away.txt**: muestra el contenido del archivo.
2. **Select-Object -Unique**: elimina las líneas consecutivas repetidas
3. **Where-Object {\$_. -ne ""}**: elimina las líneas vacías existentes.
4. **Add-Content -PassThru**: añade numeración a cada línea.
5. **Select-Object -Skip 9 y Select-Object -Skip 31**: se queda sólo con las líneas que van de la 10 a la 40.
6. **Nl**: numera las líneas del documento.
7. **%{\$_. -replace "^[0-9]*"}**: elimina la numeración de las líneas.

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

8. **Sort-Object -Descending**: ordena las líneas resultantes en orden alfabético inverso.

```
PS C:\Users\alumne\Desktop> Get-Content give_it_away.txt | Select-Object -Unique | Where-Object {$_.ne ""} | Add-Content -PassThru | Select-Object -Skip 9 | Select-Object -First 31 | %{$_.replace "[0-9]*"} | Sort-Object -Descending
Unimpressed by material excess
There's a river born to be a giver
Ooh... Ooh yeah
No time for the piggies or the hoosegow
Never been a better time than right now
My mom I love her 'cause she love me
Lucky me swimmin' in my ability
Love is free love me say hell yes
Long gone are the times when she scrub me
Keep your more to receive your less
Keep you warm won't let you shiver
I'm a low brow but I rock a little know how
I can't tell iff I'm a king pin or a pauper
I can't tell if I'm a king pin or a pauper
How come everybody wanna keep it like the kaiser
His heart is never gonna wither
Greedy little people in a sea of distress
Goodness me can't you see I'm gonna cough it
Give it away now
Give it away give it away give it away now
Get smart get down with the pow wow
Feelin' good my brother gonna hug me
Drinkin' my juice young love chug-a-lug me
Dancin' down on life with agility
Confide wisely you'll be the wiser
Come on everybody time to deliver
Come and drink it up from my fertility
Bob Marley walkin' like he talk it
Bob Marley taught me how to off it
Bob Marley poet and a prophet
Blessed with a bucket of lucky mobility
```

19	<p>Tanto en Linux como en Windows, escriba un comando que devuelva la MAC de la nuestra tarjeta de red. Para ello, será necesario que:</p> <ol style="list-style-type: none"> 1. Ejecute el comando ifconfig ('ipconfig /all' o 'Get-NetAdapter' en PowerShell). 2. Seleccione de alguna manera la línea que contiene la MAC de su tarjeta de red. <p>Nota: En Linux seleccione únicamente el valor de la MAC, en Windows podemos obtener toda la línea entera.</p> <p>Explique cada uno de los pasos que haya utilizado en este comando.</p>
----	--

En Linux:

```
alumne@alumne-VirtualBox:~$ ifconfig | grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}'
08:00:27:be:9e:24
```

En PowerShell:

```
PS C:\Users\alumne> Get-NetAdapter | Select-Object -Property MACAddress

MACAddress
-----
08-00-27-BE-9E-24
```

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

20	<p>Tanto en Linux como en Windows, escriba un comando que muestre ÚNICAMENTE los PID de los 5 procesos que consumen mayor memoria del sistema.</p> <p>Nota: en el caso de Windows nos referimos a la memoria paginada.</p>
----	--

En Linux:

```
alumine@alumne-VirtualBox:~/Escritorio/prueba$ ps -e -o pid --sort=-rss | head
-n 6 | tail -n 5
1607
2820
2016
1957
1889
```

En Windows:

```
PS C:\Users\alumne> Get-Process | Sort-Object -Property WS -Descending
| Select-Object -First 5 -Property ID

Id
--
5400
2692
6724
4872
6796
```

21	<p>Escriba un comando que permita mostrar por pantalla el siguiente mensaje: ***MICRO\$OFT RULE\$***.</p>
----	--

Con el comando **echo** podemos imprimir el mensaje que queremos por pantalla.

```
alumine@alumne-VirtualBox:~/Escritorio/prueba$ echo '***MICRO$OFT RULE$***'
***MICRO$OFT RULE$***
```

22	<p>En Linux, ¿Para qué sirve cada una de las expresiones regulares de grep que se indican a continuación?</p>
----	---

COMANDO	RESULTADO
^	Se usa para buscar líneas que empiecen con uno o más caracteres indicados después del símbolo.
\$	Utilizado para buscar líneas que terminen con uno o más caracteres indicados antes del símbolo.
[...]	Se usa para buscar líneas que contengan al menos uno de los caracteres que haya dentro de los corchetes.
[^...]	Utilizado para buscar líneas que no contengan ninguno de los caracteres que haya dentro de los corchetes.
[m-r]	Se utiliza para buscar líneas que contengan al menos uno de los caracteres que se encuentren dentro del rango de m y r.
.	Sirve para buscar una línea que contenga cualquier carácter. El "." es un comodín que representa cualquier carácter.
c*	Se usa para buscar una línea que contenga cero o más instancias del carácter "c".

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 7 Gestión de procesos 2
---	---

.	Busca cualquier línea que contenga cualquier carácter, ya sea una o varias veces.
\{n\}	Sirve para buscar una línea que contenga una cantidad específica de un carácter o conjunto de caracteres consecutivamente.
\{n,\}	Se usa para especificar una cantidad mínima de veces que un carácter o conjunto de caracteres debe aparecer.
\{n,m\}	Se utiliza para especificar un rango mínimo de apariciones y un número máximo de apariciones (por ejemplo: grep "a{3,5}", donde 3 es el mínimo y 5 es el máximo.)
\	Utilizado en las expresiones regulares para escapar caracteres especiales (por ejemplo: grep "a*b, busca una cadena que tenga una a, seguido de un * y de una b).

23	En Linux , explique qué hace cada uno de los siguientes comandos:
----	--

COMANDO	RESULTADO
grep give *	Muestra las líneas que tienen la contengan give.
grep -l give *	Muestra el nombre de los archivos los cuales en su contenido hay alguna cadena "give".
grep -i give *.txt	Busca en todas las líneas de todos los archivos .txt la palabra "give", indistintamente de si están en mayúscula o minúscula, y las imprime.
grep -i -h give *.txt	Hace lo mismo que el anterior, pero elimina el nombre del archivo del principio de la línea y el delimitador.
grep -n give give_it_away.txt	Busca una cadena de texto específica en un archivo y muestra las líneas en las que se encuentra esa cadena, junto con el número de línea.
grep -cv give give_it_away.txt	Busca una cadena de texto en un archivo y muestra la cantidad de veces que no se encuentra la cadena en el archivo.
grep give\$ give_it_away.txt	Muestra las líneas en las que las que "give" sea el ultimo carácter de la línea.
grep ^give\$ *	Muestra solo las líneas que contengan solo la palabra "give" y no otras palabras o caracteres adicionales.
grep [A-Z] give_it_away.txt	Busca cualquier carácter en mayúsculas en el archivo "give_it_away.txt".
grep 'Stal\{1,2\}man' *	
grep '^error' *.log > problemas.txt grep '^dump' *.log >> problemas.txt	El comando "grep '^error' *.log > problemas.txt" busca la cadena "error" al principio de cualquier línea en todos los archivos con extensión .log del directorio actual y redirige el resultado a un archivo llamado "problemes.txt" en el directorio actual. El comando "grep '^dump' *.log >> problemas.txt" busca la cadena "dump" al principio de cualquier línea en todos los archivos con extensión .log del directorio actual y agrega el resultado a "problemes.txt" existente en el directorio actual.
grep -c /bin/bash /etc/passwd	Busca la cadena "/bin/bash" en el archivo "/etc/passwd" y cuenta el número de veces que se encuentra esa cadena.