

Práctica 1: Linux Básico

Administración de Servidores
Grado en Ingeniería Informática



Álvaro Santos Romero

Preparación del entorno

Ejercicio 1.

Para la realización de este ejercicio, se ha utilizado la práctica 0 como modelo, por lo que se ha creado una máquina virtual de Ubuntu.

```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"
  config.vm.network :forwarded_port, guest: 80, host: 4567
  config.vm.provision :shell, path: "bootstrap.sh"
end
```

El **vagrantfile** contiene:

- Configuración de red solicitada por el enunciado, **puerto 80** de la máquina invitada redireccionado al **puerto 4567** del equipo.
- Ejecución del script **bootstrap.sh** mediante la palabra reservada *'provision'*, que permite que el sistema utilice el provisioner shell para configurar la máquina en el arranque.

bootstrap.sh:

```
#!/usr/bin/env bash
|
apt-get update
apt-get install -y apache2
```

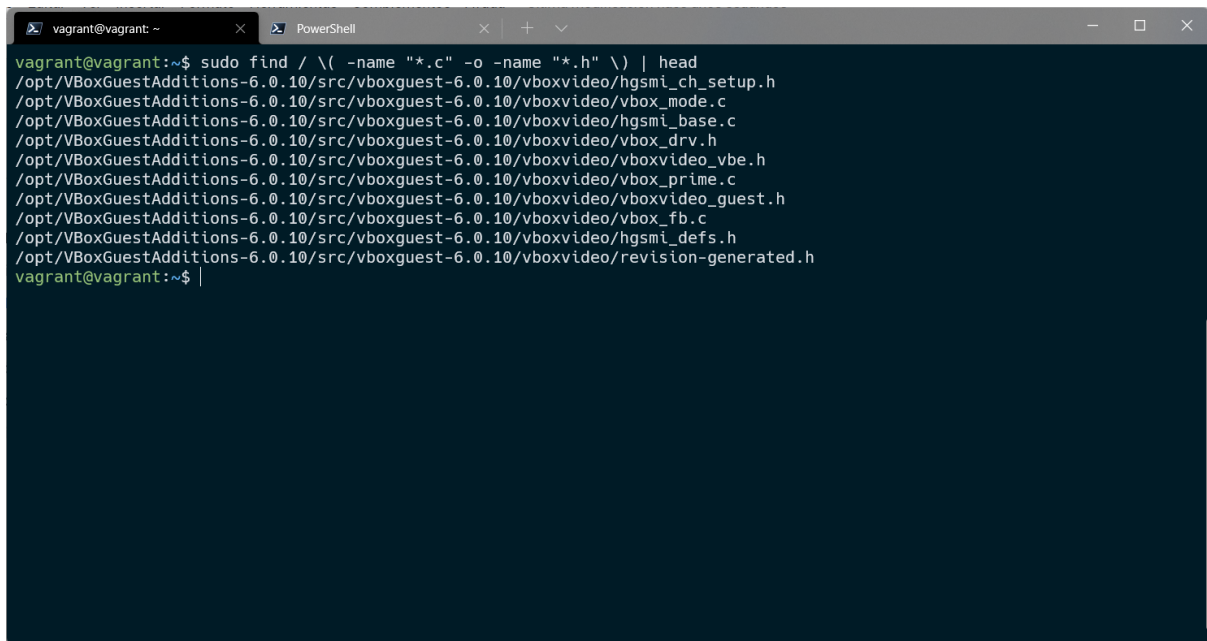
Como se observa en el script, este ejecuta dos comandos:

- **apt-get update**: descarga las actualizaciones de los paquetes del sistema.
- **apt-get install -y apache2**: descarga Apache automáticamente en la máquina.

Órdenes Básicas

Ejercicio 2.

`sudo find / -type f \(-name "*.c" -o -name "*.h" \) | head`
(Se ha utilizado sudo para evitar mostrar avisos de permiso denegado a fichero)



```
vagrant@vagrant: ~$ sudo find / -type f \( -name "*.c" -o -name "*.h" \) | head
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/hgsmi_ch_setup.h
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vbox_mode.c
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/hgsmi_base.c
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vbox_drv.h
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vboxvideo_vbe.h
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vbox_prime.c
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vboxvideo_guest.h
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/vbox_fb.c
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/hgsmi_defs.h
/opt/VBoxGuestAdditions-6.0.10/src/vboxguest-6.0.10/vboxvideo/revision-generated.h
vagrant@vagrant:~$ |
```

Información adicional:

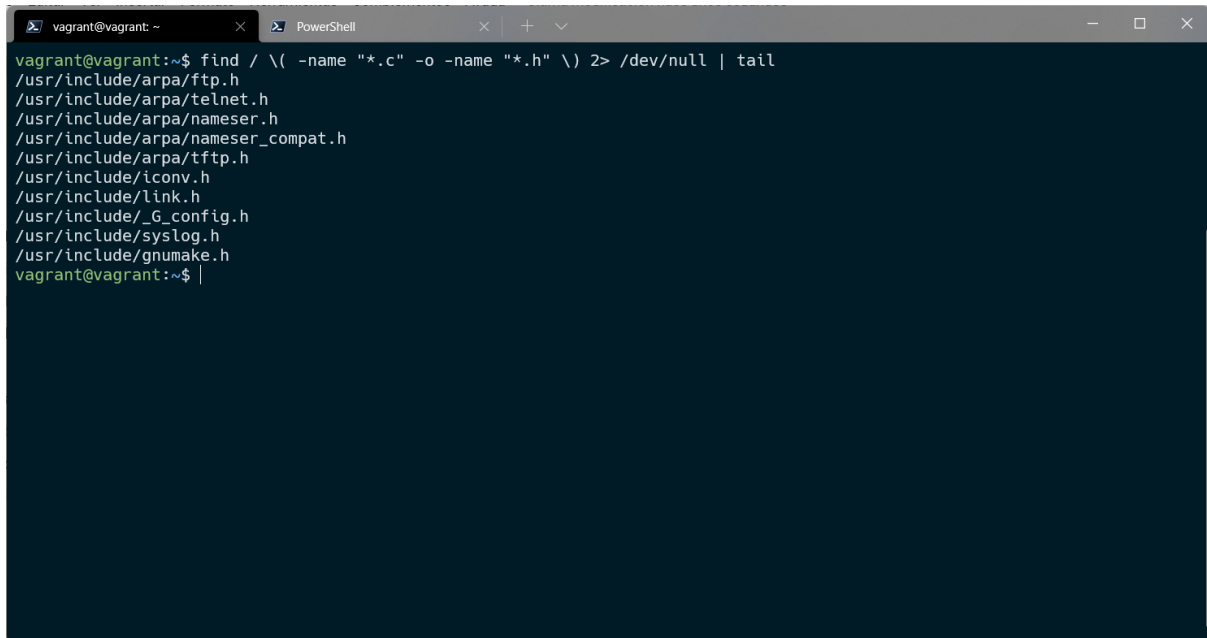
- **find[1]**: Permite buscar ficheros en el directorio indicado.

Explicación detallada del comando:

- **\(-name "*.c" -o name "*.h" \)**: Expresión regular para seleccionar ficheros con nombre terminado en .c o en .h
- **head**: permite mostrar las 10 primeras líneas del fichero (irrelevante para el funcionamiento del comando, utilizado para realizar captura más cómoda de la salida del comando).

Ejercicio 3.

`find / \(-name "*.c" -o -name "*.h" \) 2> /dev/null | tail`



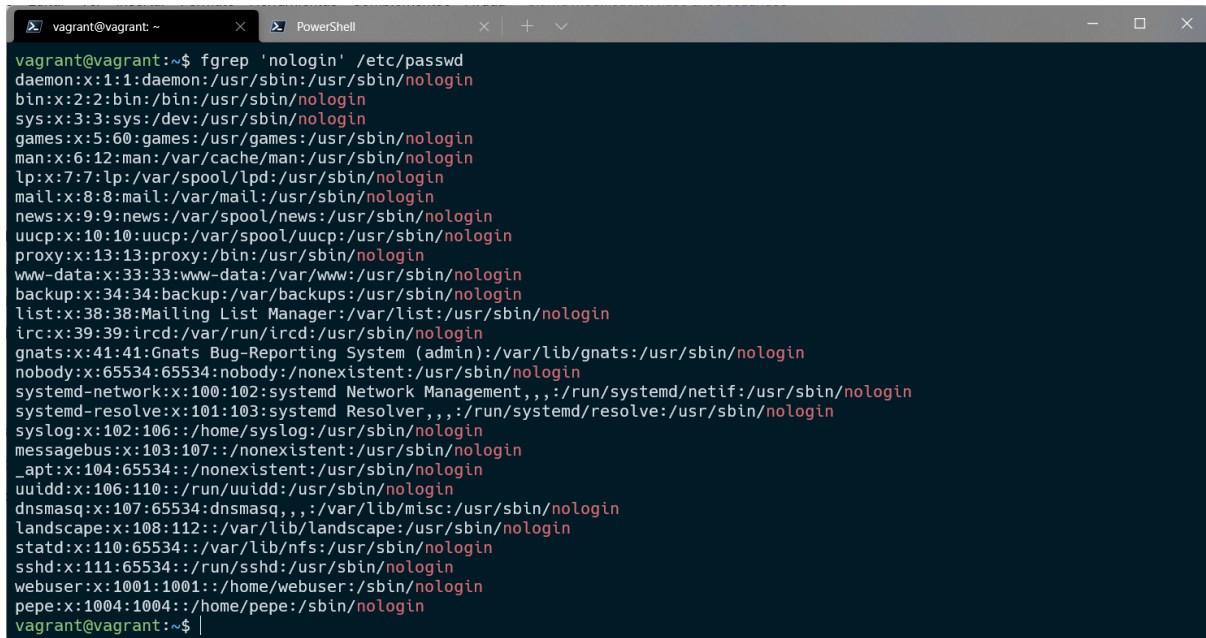
The screenshot shows a terminal window with the command `find / \(-name "*.c" -o -name "*.h" \) 2> /dev/null | tail` being executed. The output lists several header files in the `/usr/include/arpa/` directory, including `ftp.h`, `telnet.h`, `nameser.h`, `nameser_compat.h`, `tftp.h`, `iconv.h`, `link.h`, `_G_config.h`, `syslog.h`, and `gnumake.h`. The terminal window has a dark background and a title bar with tabs for 'vagrant@vagrant: ~' and 'PowerShell'.

Explicación detallada del comando:

- **find**: Permite buscar ficheros en el directorio indicado, en este caso, en el directorio raíz.
- **\(-name "*.c" -o -name "*.h" \)**: Expresión regular para seleccionar ficheros con nombre terminado en `.c` o en `.h`
- **2> /dev/null**: Redirecciona la salida de errores (2) al fichero **null**, que desecha toda información que recibe.
- **tail**: permite mostrar las 10 últimas líneas del fichero (irrelevante para el funcionamiento del comando, utilizado para realizar captura más cómoda de la salida del comando.)

Ejercicio 4.

fgrep 'nologin' /etc/passwd



```
vagrant@vagrant: ~$ fgrep 'nologin' /etc/passwd
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
uuidd:x:106:110:./run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
statd:x:110:65534:./var/lib/nfs:/usr/sbin/nologin
sshd:x:111:65534:./run/sshd:/usr/sbin/nologin
webuser:x:1001:1001:./home/webuser:/sbin/nologin
pepe:x:1004:1004:./home/pepe:/sbin/nologin
vagrant@vagrant: ~$
```

Información adicional:

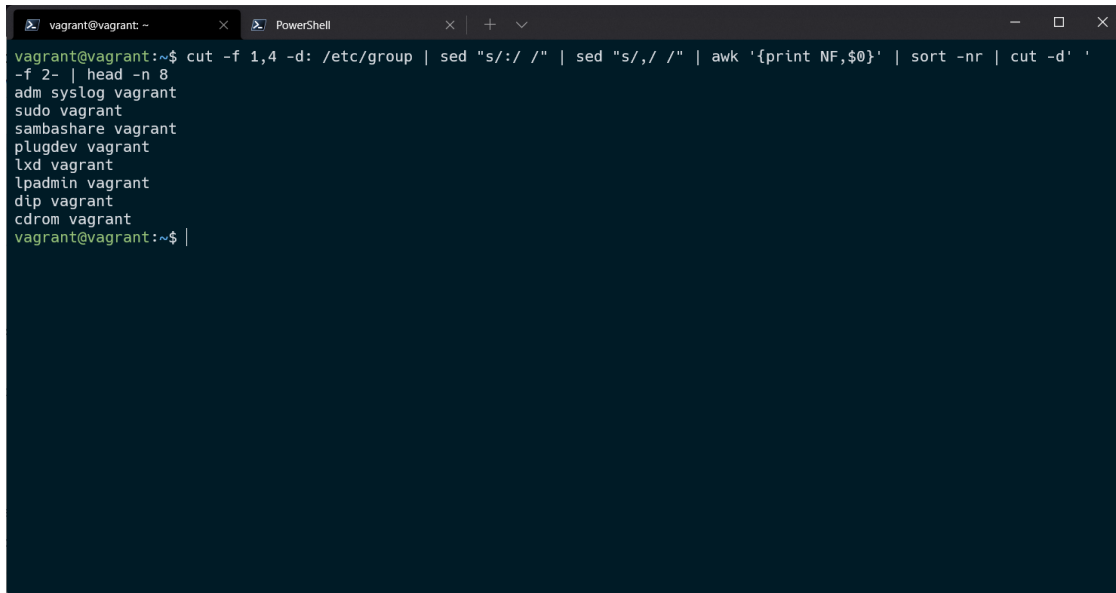
- **fgrep[2]**: Nos permite imprimir las líneas que coincidan con un patrón.
- **fgrep** es idéntico a **grep -f**.
- **nologin[3]**: 'Fichero' especial que rechaza un inicio de sesión, es decir, que no permite al usuario entrar en la cuenta.
- Este puede aparecer por dos motivos: cuando se intenta iniciar sesión en una cuenta inexistente o cuando un usuario sin privilegios intenta acceder al sistema.
- En el fichero `passwd`, **/sbin/nologin** se encuentra en el campo perteneciente al shell del usuario.

Explicación detallada del comando:

- **'nologin'**: Patrón a buscar en las líneas.
- **/etc/passwd**: Fichero en el que se va a ejecutar el comando.

Ejercicio 5.

```
cut -f 1,4 -d: /etc/group | sed "s:/ /" | sed "s/, /" | awk '{print NF,$0}' | sort -nr | cut -d' ' -f 2-
```



```
vagrant@vagrant:~$ cut -f 1,4 -d: /etc/group | sed "s:/ /" | sed "s/, /" | awk '{print NF,$0}' | sort -nr | cut -d' ' -f 2-  
-f 2- | head -n 8  
adm syslog vagrant  
sudo vagrant  
sambashare vagrant  
plugdev vagrant  
lxd vagrant  
lpadmin vagrant  
dip vagrant  
cdrom vagrant  
vagrant@vagrant:~$ |
```

Información adicional:

- **cut[4]**: Nos permite eliminar secciones de cada línea de un fichero e imprimir las seleccionadas.
- **sed[5]**: Permite filtrar y transformar texto.
- **awk[6]**: Es un lenguaje de programación diseñado para leer y procesar archivos de texto.
- **sort[7]**: Ordena las líneas de un fichero.

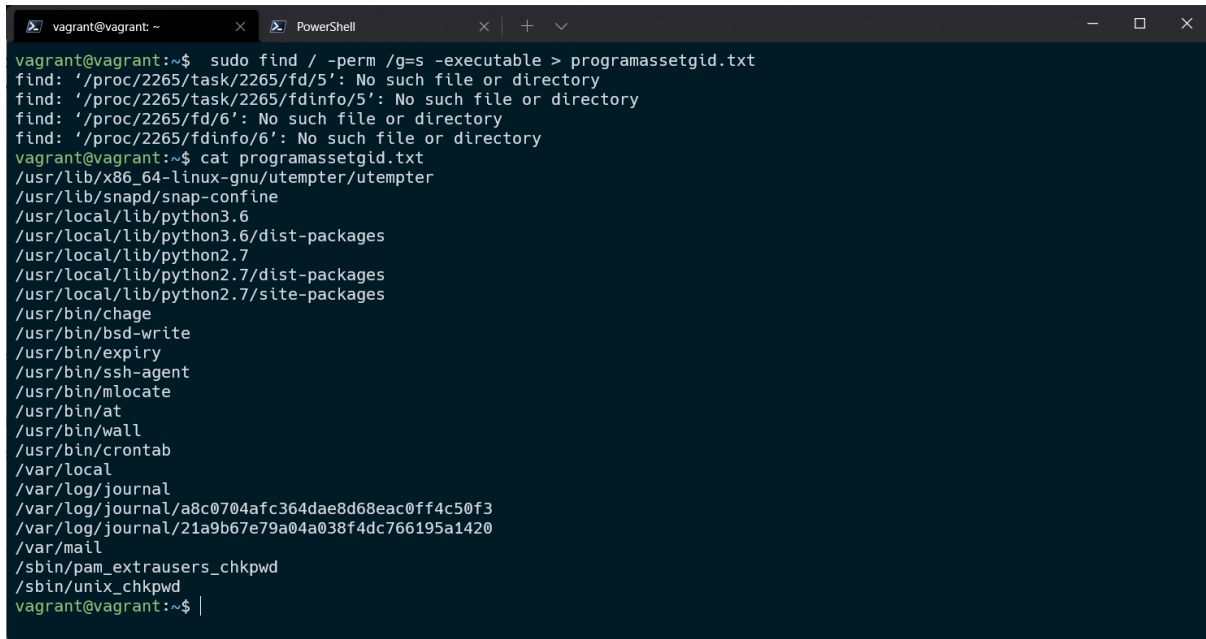
Explicación detallada del comando:

- **cut -f 1,4 -d: /etc/group**: con **-f** indicamos que líneas queremos imprimir (1,4) y con **-d** indicamos el delimitador de esas líneas (por defecto es TAB).
/etc/group es el fichero seleccionado.
- **sed "s:/ /" | sed "s/, /"**: reemplaza toda aparición de : y de , por espacios (necesario para la utilización de awk).
- **awk '{print NF,\$0}'**:
 - print: imprime líneas del archivo.
 - NF: número de campos en la línea.
 - \$0: variable que contiene toda la línea leída.
- **sort -nr**: nos permite ordenar numéricamente (**-n**) y orden descendente (**-r**).
- **cut -d ' ' -f 2-**: muestra desde la segunda columna (evitamos imprimir el contador de palabras de awk).

Procesos

Ejercicio 6.

sudo find / -perm /g=s -executable > programassetgid.txt



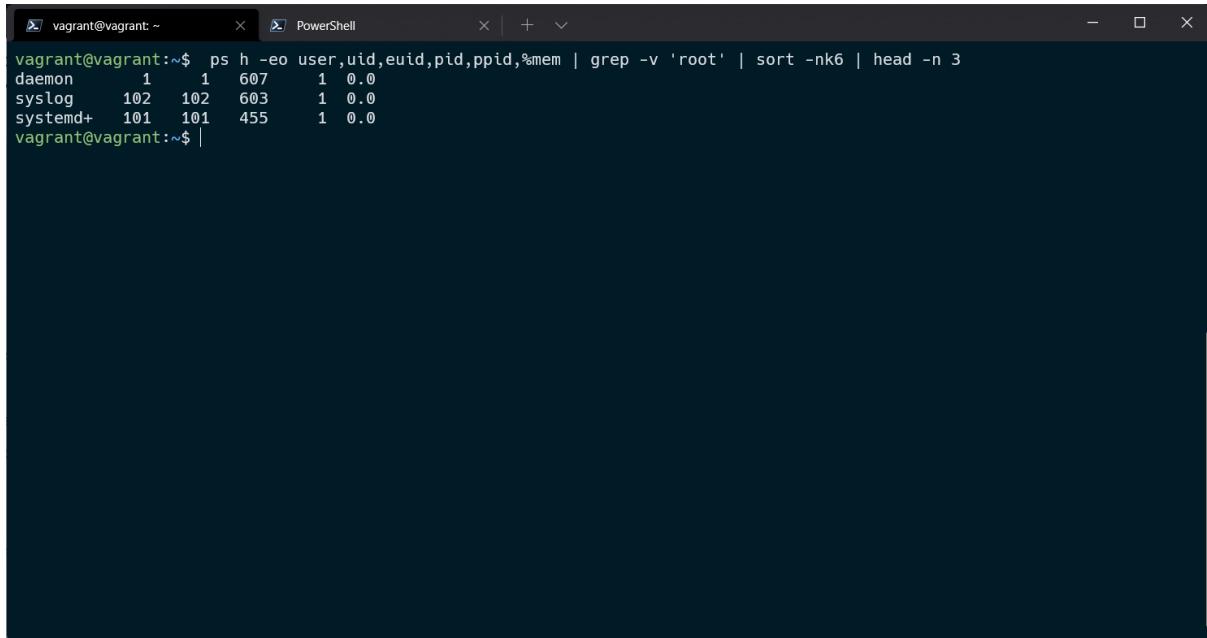
```
vagrant@vagrant: ~$ sudo find / -perm /g=s -executable > programassetgid.txt
find: '/proc/2265/task/2265/fd/5': No such file or directory
find: '/proc/2265/task/2265/fdinfo/5': No such file or directory
find: '/proc/2265/fd/6': No such file or directory
find: '/proc/2265/fdinfo/6': No such file or directory
vagrant@vagrant: ~$ cat programassetgid.txt
/usr/lib/x86_64-linux-gnu/utempter/utempter
/usr/lib/snapd/snap-confine
/usr/local/lib/python3.6
/usr/local/lib/python3.6/dist-packages
/usr/local/lib/python2.7
/usr/local/lib/python2.7/dist-packages
/usr/local/lib/python2.7/site-packages
/usr/bin/chage
/usr/bin/bsd-write
/usr/bin/expiry
/usr/bin/ssh-agent
/usr/bin/mlocate
/usr/bin/at
/usr/bin/wall
/usr/bin/crontab
/var/local
/var/log/journal
/var/log/journal/a8c0704afc364dae8d68eac0ff4c50f3
/var/log/journal/21a9b67e79a04a038f4dc766195a1420
/var/mail
/sbin/pam_extrausers_chkpwd
/sbin/unix_chkpwd
vagrant@vagrant: ~$ |
```

Explicación detallada del comando:

- **-perm /g=s**: permite buscar por permisos, en este caso, todo fichero que tenga el **setgid** activado.
- **-executable**: sólo busca archivos ejecutables (que tengan al menos el permiso de ejecución en uno de los 3 campos de permisos).
- **>**: redirección de la salida del comando a un fichero.

Ejercicio 7.

```
ps h -eo user,uid,euid,pid,ppid,%mem | grep -v 'root' | sort -nk6 | head -n 3
```



```
vagrant@vagrant: ~  
vagrant@vagrant:~$ ps h -eo user,uid,euid,pid,ppid,%mem | grep -v 'root' | sort -nk6 | head -n 3  
daemon      1      1    607      1  0.0  
syslog     102    102    603      1  0.0  
systemd+   101    101    455      1  0.0  
vagrant@vagrant:~$ |
```

Información adicional:

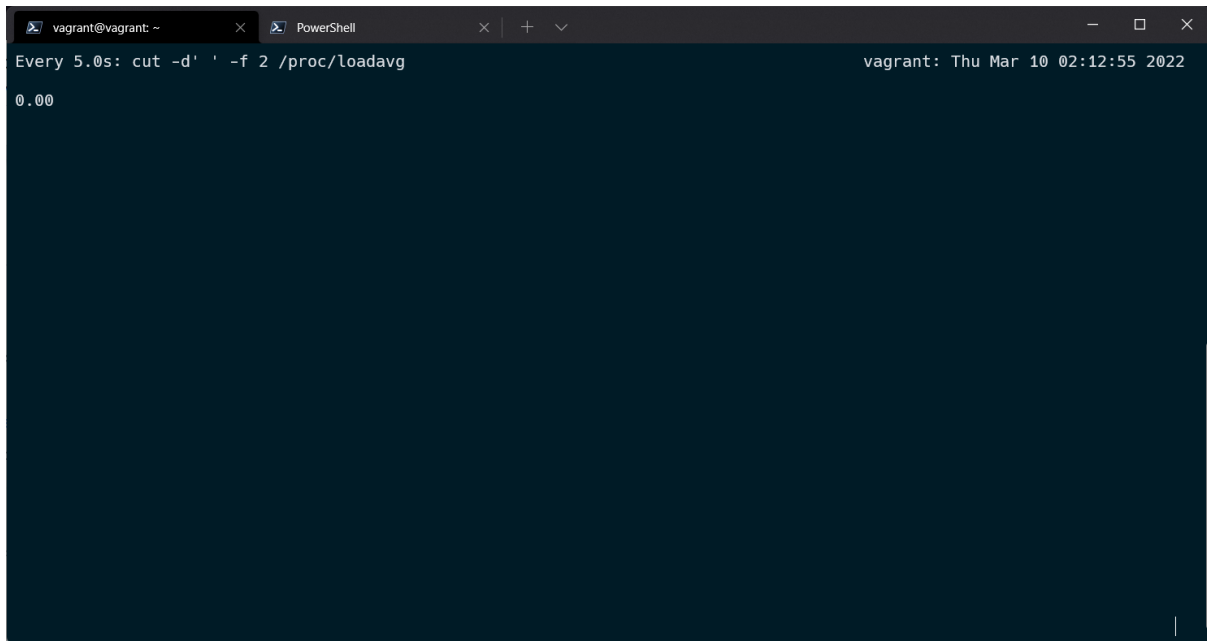
- **ps[8]**: Proporciona una lista de los procesos del sistema.

Explicación detallada del comando:

- **ps h -eo user,uid,euid,pid,ppid,%mem**: muestra todos los procesos (-e) del siguiente formato (-o user...) sin mostrar la cabecera del formato (h)(para evitar que sort interprete esa línea).
- **grep -v 'root'**: mostrar toda línea que no contenga 'root'.
- **sort -nk6**: ordenar por la sexta columna (-k 6) numéricamente(-n) y ascendente (por defecto).
- **head -n 3**: mostrar las 3 primeras líneas.

Ejercicio 8.

`watch -n 5 "cut -d' ' -f 2 /proc/loadavg"`



The screenshot shows a PowerShell terminal window with a dark blue background. The title bar indicates the window is titled 'PowerShell'. The terminal content shows the command 'watch -n 5 "cut -d' ' -f 2 /proc/loadavg"' being executed. The output of the command is '0.00'. The terminal window also shows the prompt 'vagrant@vagrant: ~' and the date 'vagrant: Thu Mar 10 02:12:55 2022'.

Información adicional:

- **watch[9]**: permite ejecutar un comando periódicamente.
- **proc[10]**: pseudo sistema de ficheros de información de procesos.
 - **loadavg**: contiene información de la carga promedio del sistema, cuyos tres primeros campos indican la carga promedio en los últimos 1, 5, 15 minutos.

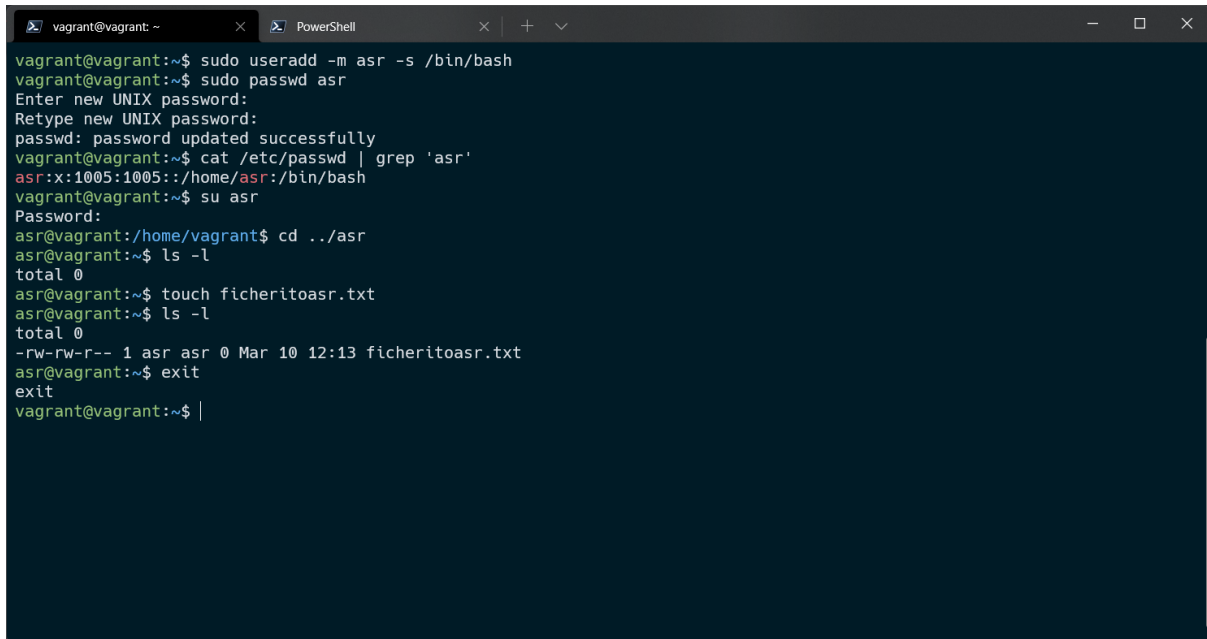
Explicación detallada del comando:

- **watch -n 5**: ejecuta un comando cada 5 segundos (**-n 5**).
- **cut -d' ' -f 2 /proc/loadavg**: selecciona el segundo campo del fichero **/proc/loadavg**.

Gestión de usuarios

Ejercicio 9.

sudo useradd -m asr -s /bin/bash
sudo passwd asr



```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo useradd -m asr -s /bin/bash  
vagrant@vagrant:~$ sudo passwd asr  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
vagrant@vagrant:~$ cat /etc/passwd | grep 'asr'  
asr:x:1005:1005::/home/asr:/bin/bash  
vagrant@vagrant:~$ su asr  
Password:  
asr@vagrant:/home/vagrant$ cd ../asr  
asr@vagrant:~$ ls -l  
total 0  
asr@vagrant:~$ touch ficheritoasr.txt  
asr@vagrant:~$ ls -l  
total 0  
-rw-rw-r-- 1 asr asr 0 Mar 10 12:13 ficheritoasr.txt  
asr@vagrant:~$ exit  
exit  
vagrant@vagrant:~$ |
```

Información adicional:

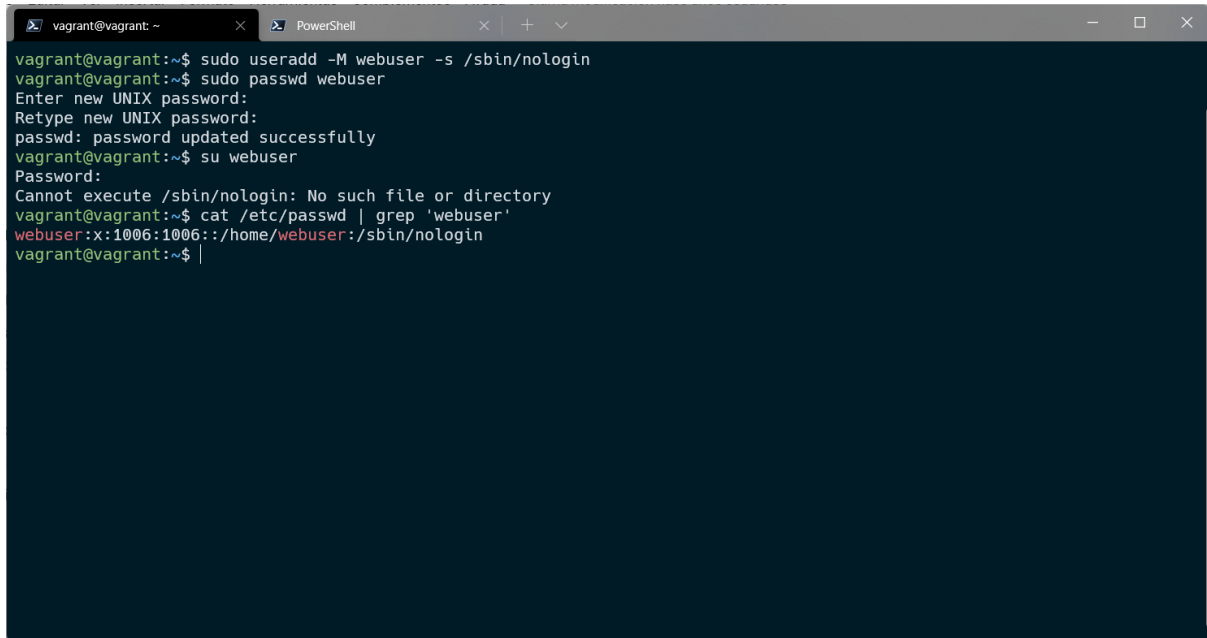
- **useradd[11]**: Permite crear un nuevo usuario en el sistema.
- **useradd -D**: Da información detallada acerca de los valores por defecto a la hora de la creación del usuario.

Explicación detallada del comando:

- **useradd -m asr**: crea al nuevo usuario un directorio casa con ese nombre (por defecto en **/home**).
- **-s /bin/bash**: asigna al usuario una shell (por defecto **/bin/sh**).
- **sudo passwd asr**: proporciona una contraseña al usuario para poder iniciar sesión.

Ejercicio 10.

sudo useradd -M webuser -s /sbin/nologin
sudo passwd webuser



```
vagrant@vagrant: ~  
vagrant@vagrant:~$ sudo useradd -M webuser -s /sbin/nologin  
vagrant@vagrant:~$ sudo passwd webuser  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
vagrant@vagrant:~$ su webuser  
Password:  
Cannot execute /sbin/nologin: No such file or directory  
vagrant@vagrant:~$ cat /etc/passwd | grep 'webuser'  
webuser:x:1006:1006:./home/webuser:/sbin/nologin  
vagrant@vagrant:~$ |
```

Explicación detallada del comando:

- ***useradd -M webuser***: crea al nuevo usuario pero sin directorio casa, ignorando al fichero ***/etc/login.defs*** (contiene una configuración predeterminada a la hora de crear usuarios.)
- ***-s /sbin/nologin***: Le asignamos el fichero ***nologin*** que no permite al usuario iniciar sesión.
- ***sudo passwd webuser***: proporcionamos una contraseña al usuario para comprobar que no podemos iniciar sesión.

Ejercicio 11.

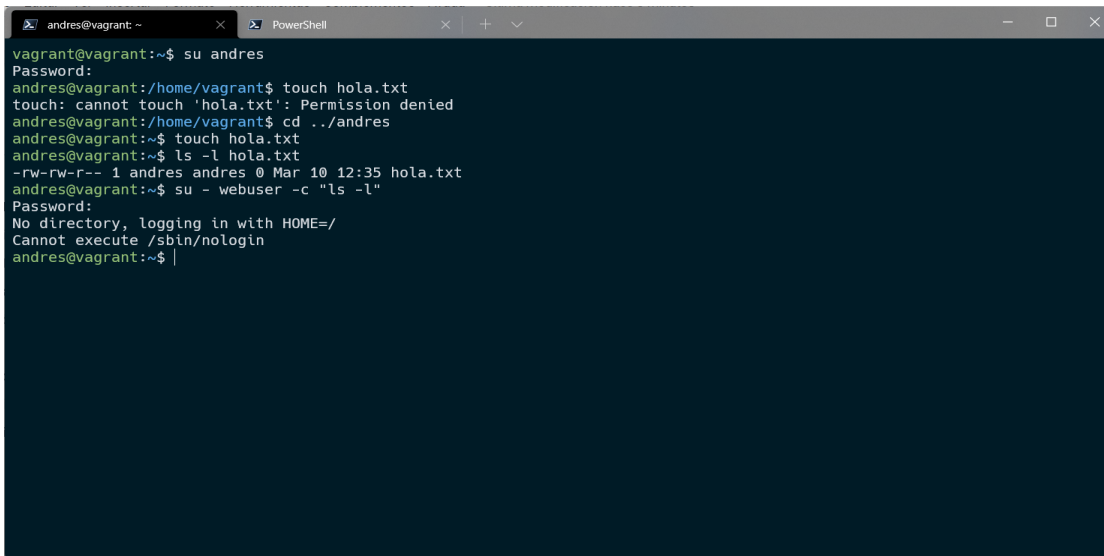
```
sudo useradd -m andres -s /bin/bash
```

```
sudo passwd andres
```

```
sudo visudo
```

```
vagrant@vagrant:~$ sudo useradd -m andres -s /bin/bash
vagrant@vagrant:~$ sudo passwd andres
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

```
# User alias specification
andres ALL=(webuser) ALL
```



```
andres@vagrant: ~
PowerShell
vagrant@vagrant:~$ su andres
Password:
andres@vagrant:/home/vagrant$ touch hola.txt
touch: cannot touch 'hola.txt': Permission denied
andres@vagrant:/home/vagrant$ cd ../andres
andres@vagrant:~$ touch hola.txt
andres@vagrant:~$ ls -l hola.txt
-rw-rw-r-- 1 andres andres 0 Mar 10 12:35 hola.txt
andres@vagrant:~$ su - webuser -c "ls -l"
Password:
No directory, logging in with HOME=/
Cannot execute /sbin/nologin
andres@vagrant:~$
```

Información adicional:

- **visudo[12]**: Permite editar el fichero **/etc/sudoers**.
- **sudoers[13]**: Fichero que contiene una lista de usuarios autorizados a usar sudo (también permite que otros usuarios puedan ejecutar ciertos comandos como otro usuario, entre otras funcionalidades).

Explicación detallada:

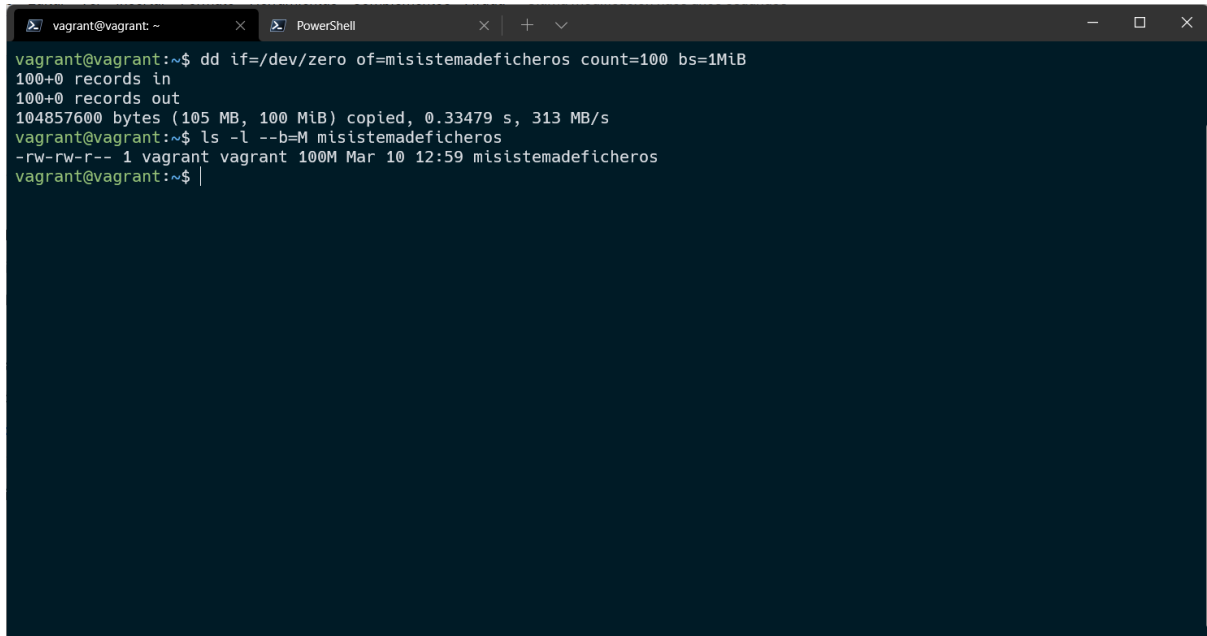
- **andres ALL=(webuser) ALL**: Sudoers utiliza EBNF, que es una forma de notación recursiva complicada de entender, pero muy potente.
En este caso, esa línea de comando se traduciría como:
El usuario **andres** puede ejecutar **cualquier comando** únicamente como el usuario **webuser**.

Como se observa en la última captura, el usuario **andres** no puede ejecutar ningún comando como **webuser** aunque tenga permisos, ya que **webuser** no tiene ni **directorio casa** ni **shell** asociada (no puede hacer **sudo** puesto no pertenece al grupo **root**).

Sistema de ficheros

Ejercicio 12.

dd if=/dev/zero of=misistemadeficheros count=100 bs=1MiB



```
vagrant@vagrant: ~  
vagrant@vagrant:~$ dd if=/dev/zero of=misistemadeficheros count=100 bs=1MiB  
100+0 records in  
100+0 records out  
104857600 bytes (105 MB, 100 MiB) copied, 0.33479 s, 313 MB/s  
vagrant@vagrant:~$ ls -l --b=M misistemadeficheros  
-rw-rw-r-- 1 vagrant vagrant 100M Mar 10 12:59 misistemadeficheros  
vagrant@vagrant:~$
```

Información adicional:

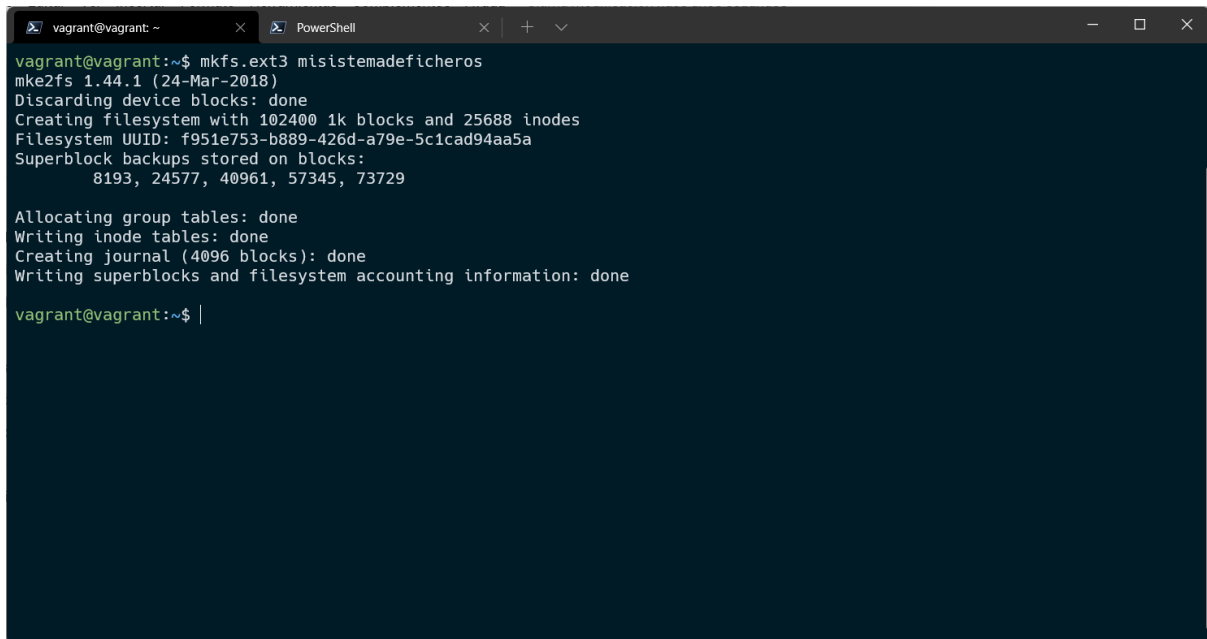
- **dd[14]:** Permite convertir o copiar un fichero.

Explicación detallada del comando:

- **dd:** copiar
- **if=/dev/zero:** desde **/dev/zero**
- **of=misistemadeficheros:** a **misistemadeficheros**
- **count=100:** Número de bloques a copiar.
- **bs=1MiB:** Tamaño de los bloques a copiar.

Ejercicio 13.

mkfs.ext3 misistemadeficheros

A terminal window with a dark blue background and white text. The window has two tabs: 'vagrant@vagrant: ~' and 'PowerShell'. The terminal shows the command 'mkfs.ext3 misistemadeficheros' being executed. The output includes version information for mke2fs (1.44.1, 24-Mar-2018), progress reports for discarding device blocks, creating the filesystem (102400 1k blocks and 25688 inodes), displaying the filesystem UUID (f951e753-b889-426d-a79e-5c1cad94aa5a), listing superblock backup locations (8193, 24577, 40961, 57345, 73729), allocating group tables, writing inode tables, creating the journal (4096 blocks), and finally writing superblocks and filesystem accounting information. The prompt returns to 'vagrant@vagrant:~\$ |'.

Información adicional:

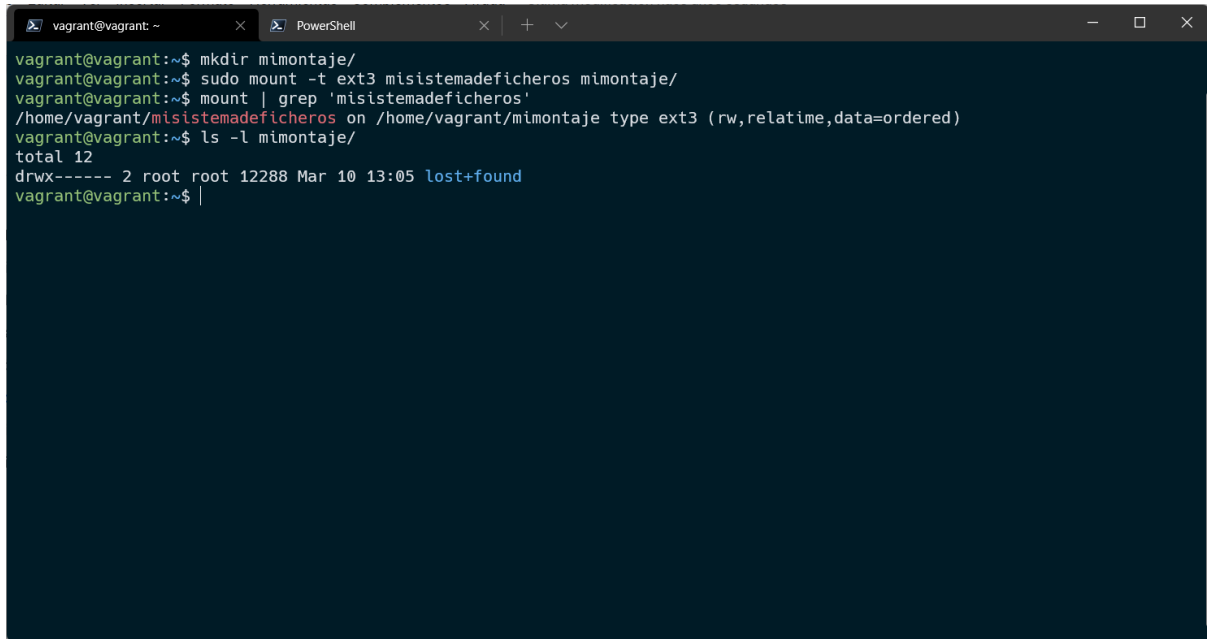
- **mkfs[15]:** Permite crear un sistema de ficheros con un formato determinado.

Explicación detallada del comando:

- **.ext3:** Formato utilizado en el sistema de ficheros. Cabe destacar que cada formato tiene un propósito, en este caso, se utiliza ext3 porque es rápido y permite reducción/expansión.

Ejercicio 14.

sudo mount -t ext3 misistemadeficheros mimontaje/



```
vagrant@vagrant: ~  
vagrant@vagrant:~$ mkdir mimontaje/  
vagrant@vagrant:~$ sudo mount -t ext3 misistemadeficheros mimontaje/  
vagrant@vagrant:~$ mount | grep 'misistemadeficheros'  
/home/vagrant/misistemadeficheros on /home/vagrant/mimontaje type ext3 (rw,relatime,data=ordered)  
vagrant@vagrant:~$ ls -l mimontaje/  
total 12  
drwx----- 2 root root 12288 Mar 10 13:05 lost+found  
vagrant@vagrant:~$
```

Información adicional:

- **mount[16]:** Monta un sistema de ficheros, es decir, inserta el sistema de ficheros en el árbol de nuestro sistema.
- **mount:** El comando mount permite ver qué sistemas están montados en nuestro sistema, por lo que con **grep** podemos ver que nuestro sistema se montó correctamente.

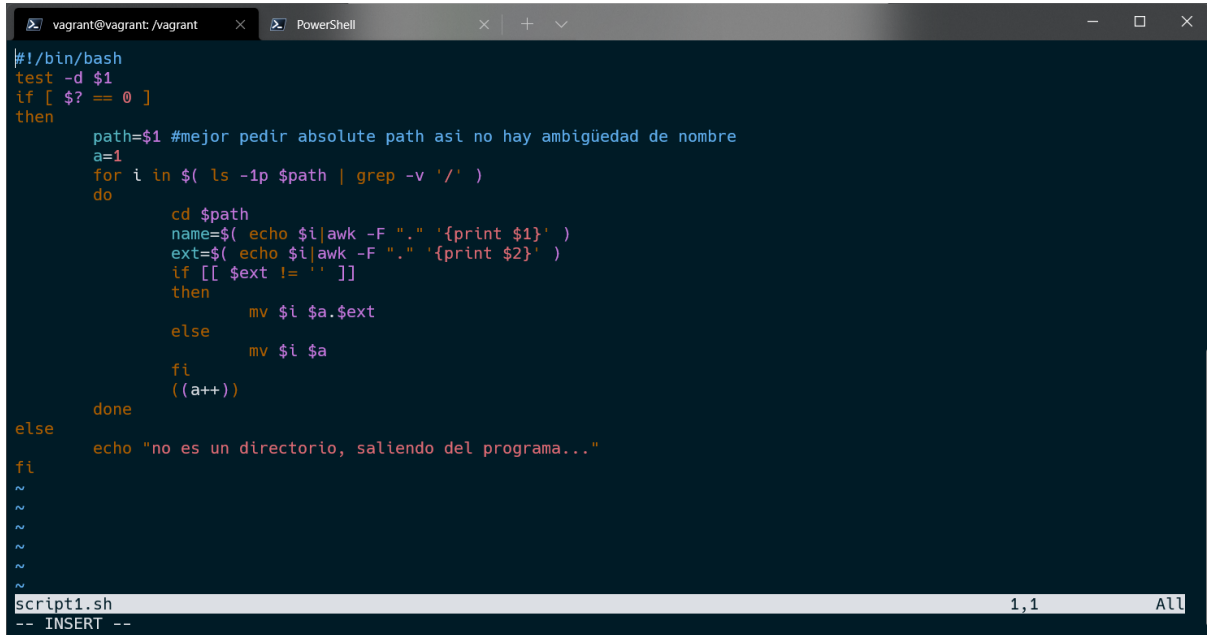
Explicación detallada del comando:

- **-t ext3:** No es necesario indicar el formato, pero en este caso se ha considerado indicarlo para que sea más visual.
- **misistemadeficheros:** Sistema de ficheros a montar.
- **mimontaje/:** Directorio donde se va a montar nuestro sistema de ficheros.

Nota: el directorio debe estar creado previamente!

Bash scripting

Ejercicio 15.



```
#!/bin/bash
test -d $1
if [ $? == 0 ]
then
    path=$1 #mejor pedir absolute path asi no hay ambigüedad de nombre
    a=1
    for i in $( ls -lp $path | grep -v '/' )
    do
        cd $path
        name=$( echo $i|awk -F "." '{print $1}' )
        ext=$( echo $i|awk -F "." '{print $2}' )
        if [[ $ext != '' ]]
        then
            mv $i $a.$ext
        else
            mv $i $a
        fi
        ((a++))
    done
else
    echo "no es un directorio, saliendo del programa..."
fi
~
~
~
~
~
script1.sh 1,1 All
-- INSERT --
```

Información adicional:

- **mv[17]**: Permite mover,renombrar ficheros o ambas.
- Se ha supuesto que el usuario introduce la **ruta absoluta** de este (por enunciado, directorio dado como argumento...)

Explicación detallada del script:

Previamente a la ejecución de este, mediante un test se ha comprobado que el directorio introducido por argumento existe, y es un directorio (si no lo es, no ejecutamos el script).

Se ha implementado un bucle for en el cual cada iteración contiene el nombre de un fichero (previamente se ha restringido la salida del comando **ls** con **grep**).

Se inicializa una variable numérica para asignarla a los ficheros.

Dentro del bucle:

- Cambiamos de directorio para poder cambiar los nombres cómodamente.
- mediante **awk**, obtenemos las dos partes del fichero, el nombre y la extensión.
- Si el fichero tiene extensión (variable ext no esté vacía), entonces el nuevo nombre será un número secuencial + extensión.
- Si no tiene extensión, entonces sólo cambiamos el nombre del fichero.
- Aumentamos la variable.

Explicación detallada de los comandos:

- **echo \$i|awk -F "." '{print \$1}':** Mediante **awk**, obtenemos la primera parte (**print \$1**) de un fichero (**echo \$i**), dividido por un "."
- Idem con **ext**, pero en este caso, obtenemos la segunda parte del nombre (**print \$2**).

Ejercicio 16.

[illegible]

Información adicional:

- **tar[18]**: Permite crear y manipular archivos **tar** que son colecciones de archivos. Básicamente nos permite comprimir archivos en uno solo.
- El comando **sed** permite cambiar las barras (/) por otros caracteres, por ejemplo (|).

Explicación detallada del script:

Se ha implementado un bucle for en el cual cada iteración contiene el nombre de un directorio (previamente se ha restringido la salida del comando **ls** con **grep** y posteriormente se ha eliminado la barra del nombre del directorio utilizando **sed**).

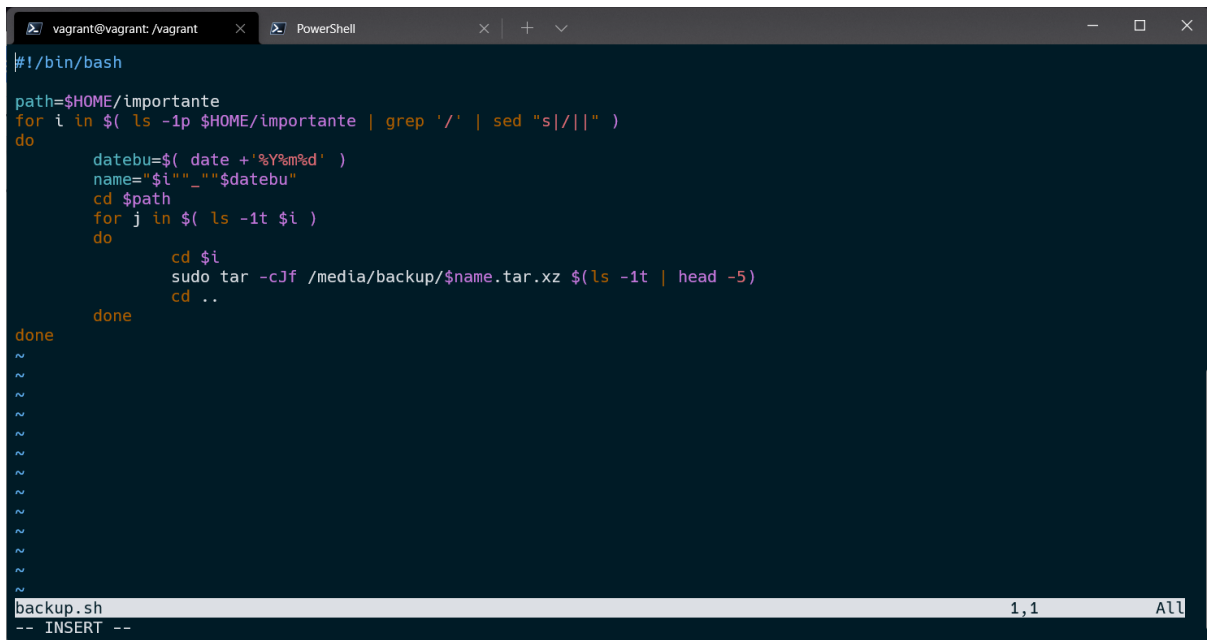
Dentro del bucle:

- se obtiene la fecha actual
- se formatea el nombre del directorio para que sea **directorio_yyyymmdd**
- dentro del directorio, se comprimen los archivos en **tar.xz**
- finalmente, se mueven al directorio **backup** junto con un cambio de nombre final

Explicación detallada de los comandos:

- **sudo tar -cJf \$i.tar.xz \$i**: Comprime el directorio **\$i** en el archivo **\$i.tar.xz** (J para xz).

Ejercicio 17.



```
#!/bin/bash

path=$HOME/importante
for i in $( ls -lp $HOME/importante | grep '/' | sed "s|/||" )
do
    datebu=$( date +%Y%m%d' )
    name="$i"_"$datebu"
    cd $path
    for j in $( ls -lt $i )
    do
        cd $i
        sudo tar -cJf /media/backup/$name.tar.xz $(ls -lt | head -5)
        cd ..
    done
done
```

backup.sh 1,1 All

Información adicional:

Explicación detallada del script:

Se ha implementado un bucle for en el cual cada iteración contiene el nombre de un directorio (previamente se ha restringido la salida del comando **ls** con **grep** y posteriormente se ha eliminado la barra del nombre del directorio utilizando **sed**).

Dentro del bucle:

- se obtiene la fecha actual
- se formatea el nombre del directorio para que sea **directorio_yyyymmdd**
- Se cambia de directorio al directorio deseado (en este caso ~/importante)

Implementamos otro bucle for en el que en cada iteración vamos a ir comprimiendo los 5 primeros archivos del directorio actual.

Dentro del bucle:

- Vamos a un directorio del ~/importante
- comprimimos los archivos de este usando el nombre formateado del directorio en **tar.xz**

- volvemos al directorio ~/importante para seleccionar otro directorio.

Explicación detallada de los comandos:

- **ls -lt \$i**: Listamos el directorio \$i, mostrando el contenido hacia abajo (-l) y ordenándolos por fecha de creación (-t)
- **sudo tar -cJf /media/backup/\$name.tar.xz \$(ls -lt | head -5)**: Comprime los primeros 5 archivos (**ls -lt | head -n 5**) en el archivo **\$name.tar.xz** (J para xz).

Ejercicio 18.

```
30 09 * * * sudo /vagrant/backup.sh
```

Información adicional:

- **cron[19]**: Es un demonio que ejecuta comandos de forma planificada.
- **crontab[20]**: Son ficheros (tablas) de configuración para ejecutar las órdenes cron.

Explicación detallada:

- **sudo crontab -u root -e**: Abrimos el fichero crontab del usuario root para modificarlo (-e).
- **30 09 * * * sudo sh /vagrant/backup.sh**:
 - **30**: Minutos.
 - **09**: Horas.
 - *****: Tanto día, mes y día de la semana están indicados como cualquiera, es decir, todos.
 - **sudo /vagrant/backup.sh**: ejecutar el comando backup.sh (ejercicio 17)

Referencias

[1] man find, fuente:

<https://linux.die.net/man/1/find>

[2] man grep, fuente:

<https://linux.die.net/man/1/grep>

[3] man nologin, fuente:

<https://linux.die.net/man/5/nologin>

[4] man cut, fuente:

<https://linux.die.net/man/1/cut>

[5] man sed, fuente:

<https://linux.die.net/man/1/sed>

[6] man awk, fuente:

<https://linux.die.net/man/1/awk>

[7] man sort, fuente:

<https://linux.die.net/man/1/sort>

[8] man ps, fuente:

<https://linux.die.net/man/1/ps>

[9] man watch, fuente:

<https://linux.die.net/man/1/watch>

[10] man proc, fuente:

<https://linux.die.net/man/5/proc>

[11] man useradd, fuente:

<https://linux.die.net/man/8/useradd>

[12] man visudo, fuente:

<https://linux.die.net/man/8/visudo>

[13] man sudoers, fuente:

<https://linux.die.net/man/5/sudoers>

[14] man dd, fuente:

<https://linux.die.net/man/1/dd>

[15] man mkfs, fuente:

<https://linux.die.net/man/8/mkfs>

[16] man mount, fuente:

<https://linux.die.net/man/8/mount>

[17] man mv, fuente:

<https://linux.die.net/man/1/mv>

[18] man tar, fuente:

<https://linux.die.net/man/1/tar>

[19] man cron, fuente:

<https://linux.die.net/man/8/cron>

[20] man crontab, fuente:

<https://linux.die.net/man/5/crontab>

