

Actividad | #3 | Comandos

para el hardware

Sistemas Operativos 1

Ingeniería en Desarrollo
de Software



TUTOR: Urbano Francisco Ortega Rivera

ALUMNO: Jenifer Ortiz Vazquez

FECHA: 03/01/2024

INDICE

1. Introducción.....	3
2. Descripción.....	4
3. Justificación.....	5
4. Desarrollo.....	6
4.1. Instalación de VirtualBox y Ubuntu.....	6
4.2. Ejecución de comandos básicos	11
4.3. Ejecución de comandos para el hardware.....	17
5. Conclusión.....	23
6. Referencias.....	24

INTRODUCCION

A lo largo de este trabajo estaremos conociendo un poco de la gestión de procesos principales y secundarios, así como algunas diferencias. Podremos conocer acerca de la memoria en Linux, tanto física como virtual, para comprender un poco más su funcionamiento.

Estaremos realizando una actividad con los comandos que sirven para la administración de la memoria dentro de Linux, para ello estarán ejecutando en nuestro Shell los comandos, /proc/meminfo, top, free y vmstat, con algunas variantes, para mostrar el procedimiento de un total de diez comandos, donde posteriormente se estarán anexando un nuestro documento con captura de pantalla de nuestro Shell, donde con una breve descripción estaremos relatando el procedimiento que se realizó con el resultado y la función de dicho comando ejecutado en nuestro sistema.

También estaremos agregando los procedimientos realizados de las actividades pasadas, que son la instalación de VirtualBox y Ubuntu, con los comandos básicos utilizados en la actividad 2

DESCRIPCION

En nuestro sistema Linux se pueden ejecutar varios procesos al mismo tiempo de los cuales existen dos tipos, procesos en primer y segundo plano. Los procesos en primer plano son aquellos en los que el usuario se encuentra interactuando con ellos, mientras que los procesos en segundo plano se ejecutan independientemente y algunos de estos se les conoce como Daemons.

Dentro de estos procesos al estarse ejecutando su estado puede cambiar por algunas variantes y pueden ser *en ejecución*, *espera*, *detenido*, y *zombie*. Existen dos formas para crear un nuevo proceso en Linux con la función *system()*, *fork()* y *exec()*, donde este último es más rápido y seguro que el primero. En System Monitor también podemos ver de manera grafica los procesos en los cuales es mucho más sencillo administrarlos ya que solo damos clic derecho en el nombre del proceso para realizarlo.

Si hablamos de memoria física en Linux nos referimos a la parte real de la maquina en la cual solo el kernel tiene acceso directo. La memoria virtual suele ser mayor que la física ya que se utiliza un espacio en el disco como extensión y en esta los programas reservan memoria.

A nuestra memoria física se le llama RAM y la parte central de nuestra computadora es la CPU, como nuestra RAM es el portal frontal de la CPU, todo lo que valla a la CPU pasara primero por la RAM, por tanto, en nuestro Linux la RAM es esencial para un buen funcionamiento del sistema operativo.

JUSTIFICACION

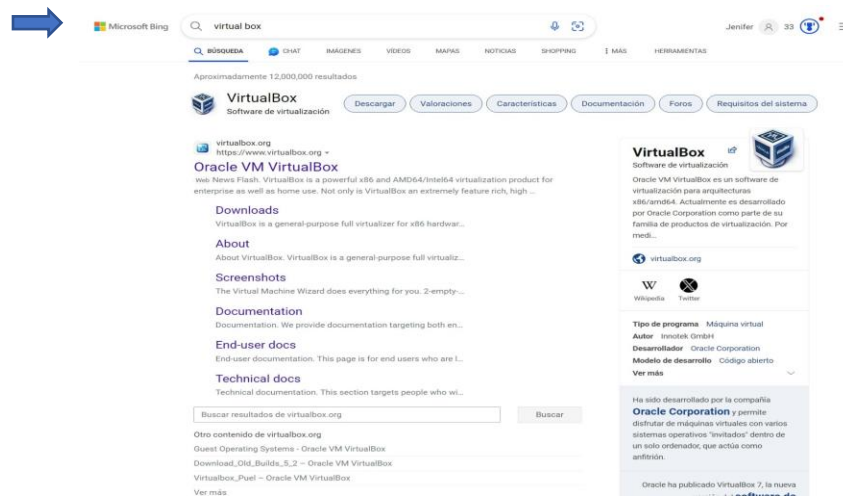
Conocer la parte física de nuestro equipo también es una parte esencial para nuestro conocimiento ya que complementa nuestra información para poder conocer el cómo trabaja o como trabajar nosotros con nuestro sistema operativo.

Con los comandos que veremos en esta actividad podremos conocer la información de nuestro hardware, que es importante para la realización de algunos de nuestros procesos ejecutados en nuestro equipo por toda la información que nos brindan cada uno de ellos ya que nos especifican muchos datos relevantes de nuestro hardware como el procesador, memoria, disco, etc.

Así con todo lo aprendido, conociendo mejor los comandos de hardware podremos utilizar mejor nuestro sistema ya que tendremos acceso a mucha de su información contenida de una forma más interna para la procesión de datos, de esta manera es que nuestro sistema opera de una buena forma y nos permite tener el acceso a su información, esa que no es posible ver a simple vista, por lo que podremos navegar por todo ese contenido de una forma externa con el simple uso de unos parámetros sin la necesidad de exponer nuestro equipo de cómputo .

DESARROLLO

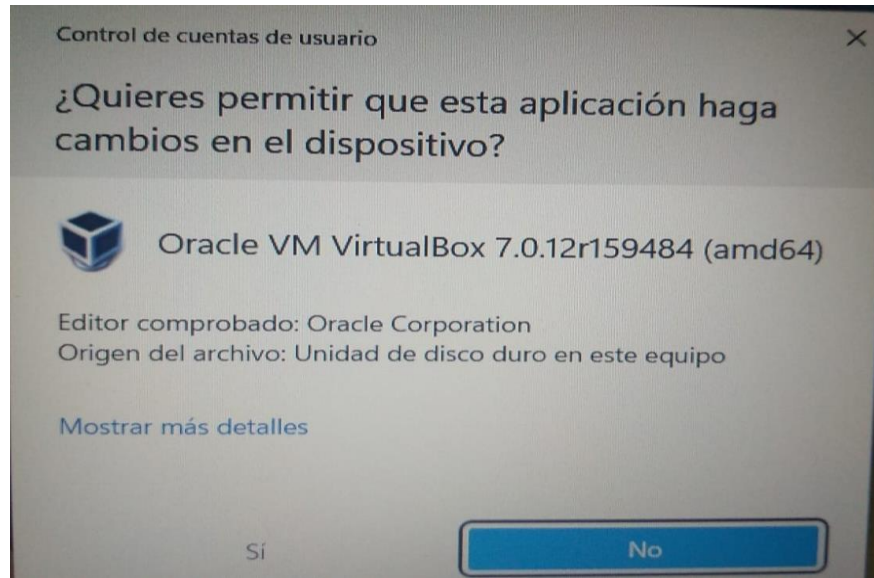
4.1 Para realizar la instalación de VirtualBox entraremos a nuestro navegador, buscamos la página oficial de VirtualBox y le damos clic



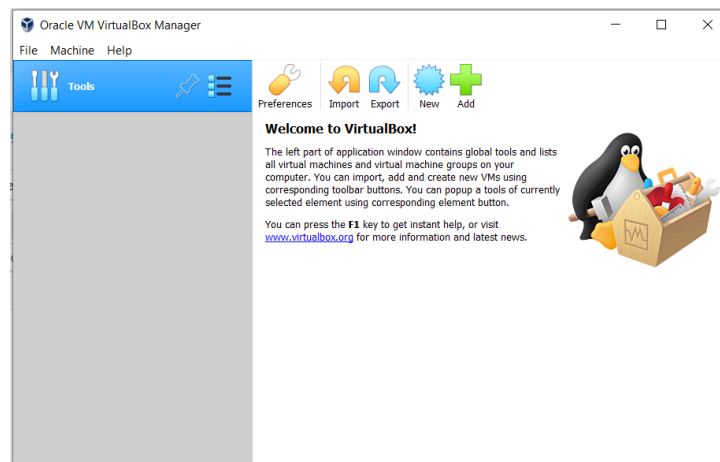
Ya estando dentro la página buscaremos la opción de descargas y procedemos a descargar la de Windows o dependiendo de tu sistema operativo nativo es el que se descarga



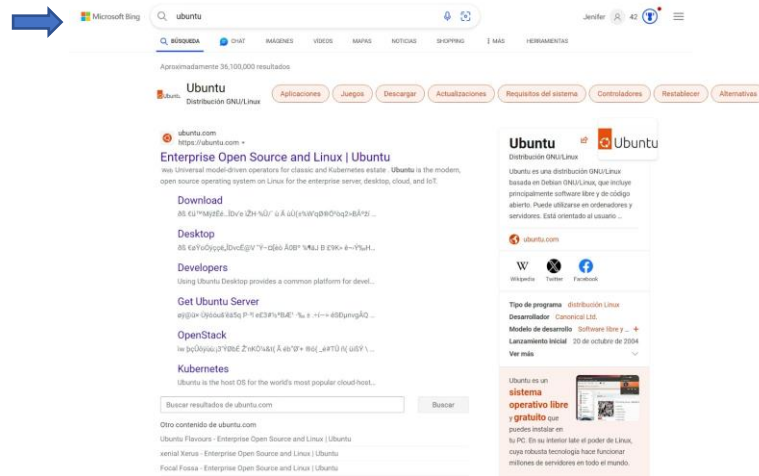
Al abrir el archivo descargado dejaremos la configuración que nos da por default y al emitir el siguiente mensaje que podemos observar en la imagen de daremos que si



Una vez descargado nos aparecerá esta pantalla apreciada en la imagen y procederemos a descargar la imagen de Ubuntu



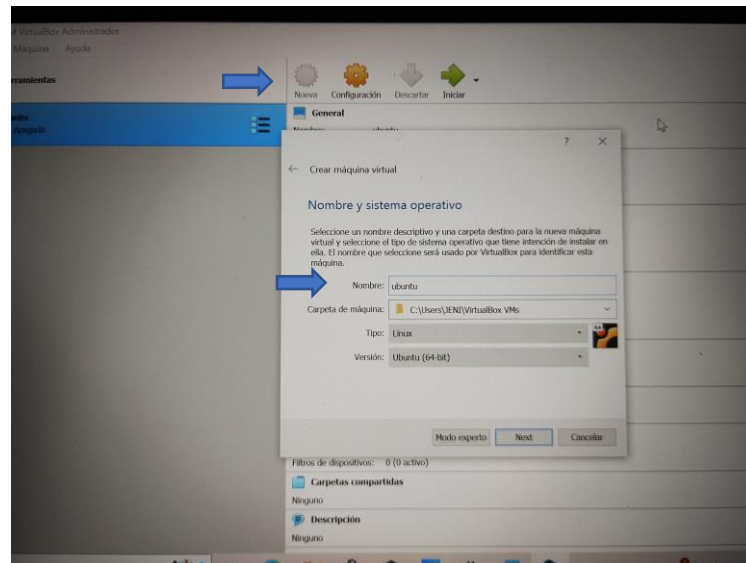
Nuevamente en nuestro navegador, entramos a la página de Ubuntu, en descargas



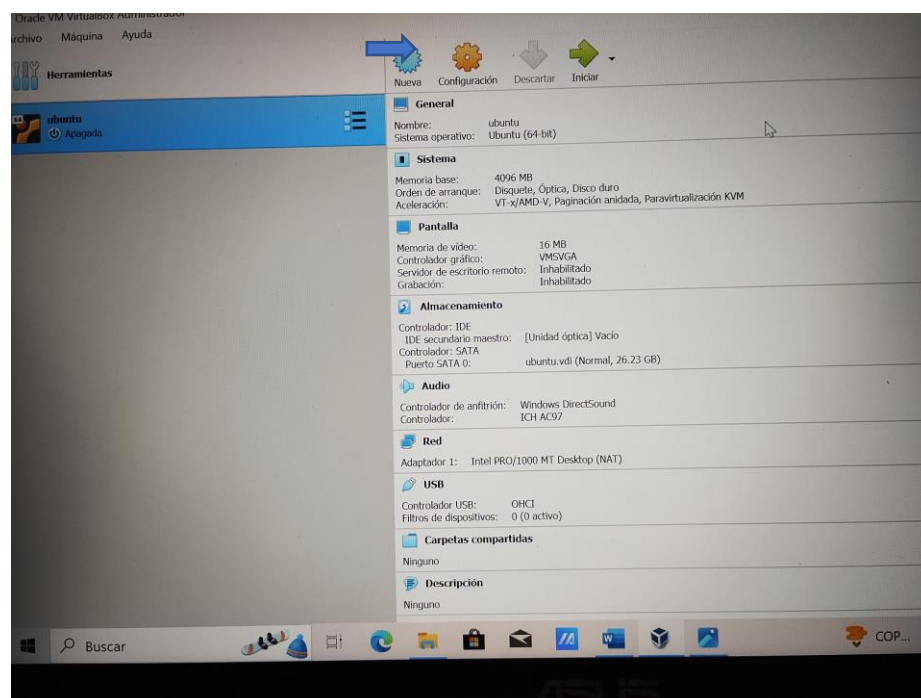
De igual forma procedemos a descarga el formato, este dura algunos minutos para su descarga



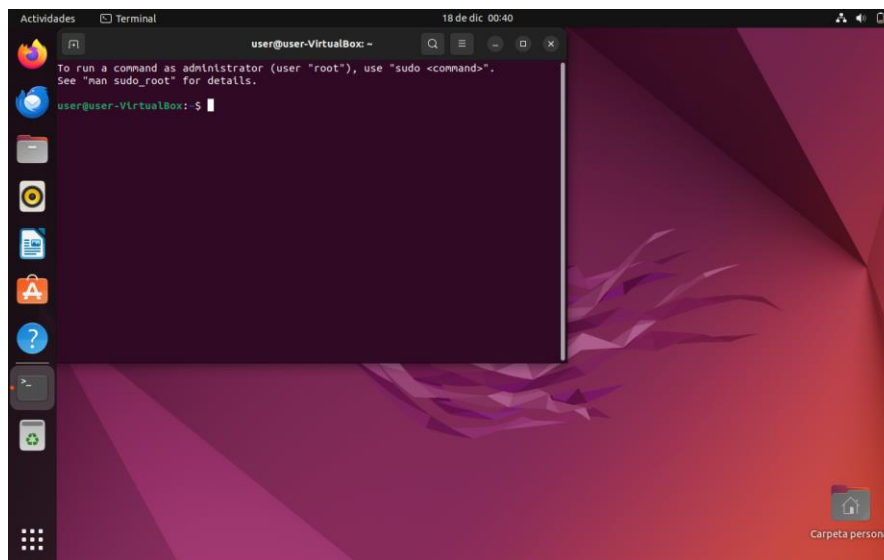
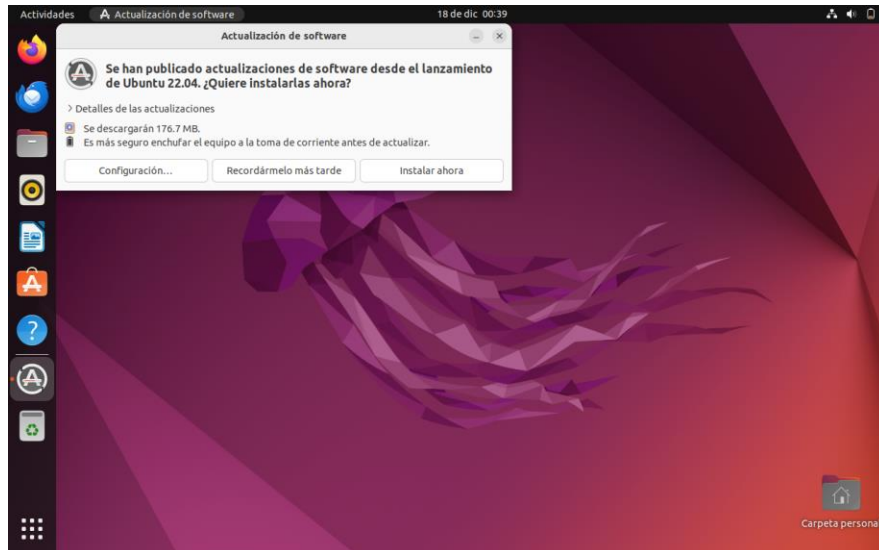
Una vez descargado abrimos VirtualBox y nos vamos en la parte donde nos dice “Nueva” en nombre le pondremos Ubuntu y dejaremos las siguientes opciones por default hasta finalizar



Una vez finalizado le daremos en iniciar

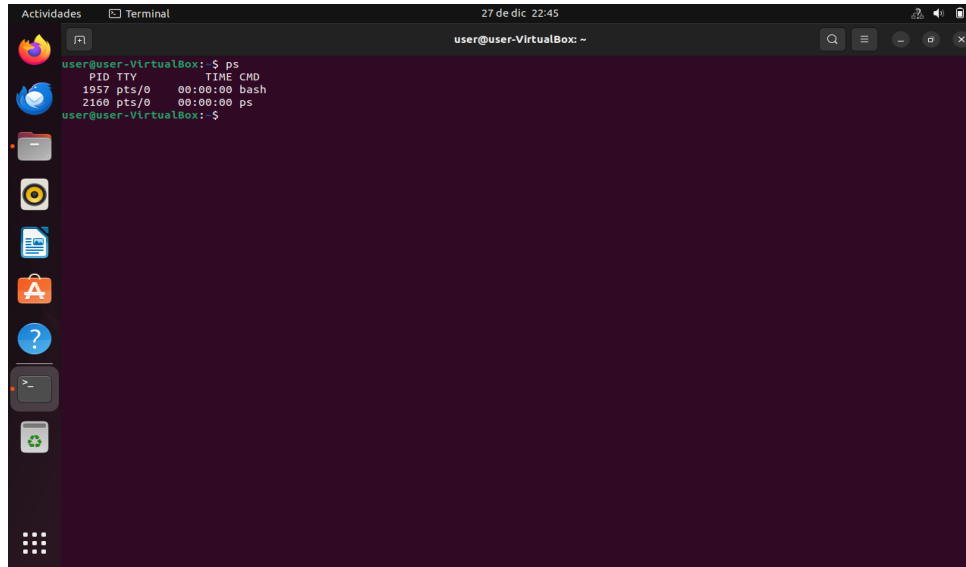


Habiendo realizado ya todo nuestro proceso y configuración nos estará abriendo esta imagen como se muestra a continuación donde ya podremos hacer uso de nuestro sistema operativo



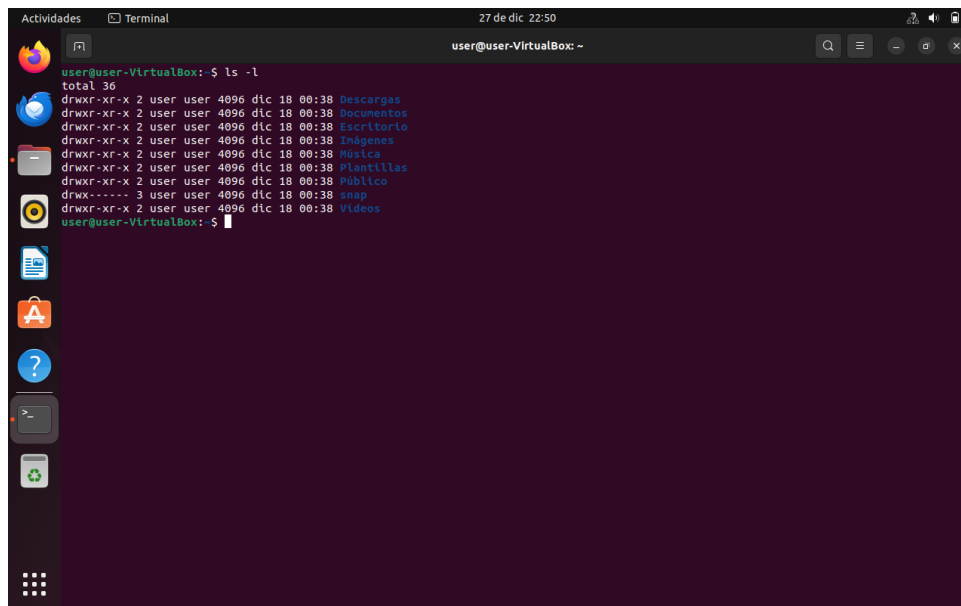
4.2 Una vez instalado nuestro sistema conoceremos algunos de los comandos básicos que podemos ejecutar en nuestro Shell.

1.El comando *ps* (estado de procesos) nos muestra todos los procesos en ejecución en el sistema. Nos muestra el identificador único del proceso (PID), tipo de terminal (TTY), el tiempo de funcionamiento (TIME), y el comando que lanza el proceso (CMD).



```
user@user-VirtualBox:~$ ps
  PID TTY          TIME CMD
 1957 pts/0    00:00:00 bash
 2160 pts/0    00:00:00 ps
user@user-VirtualBox:~$
```

2.Con el comando *ls* podemos ver el contenido de un directorio, este comando tiene muchas opciones adicionales un ejemplo de ellas es al agregar *ls -l* donde nos enlistara de una forma larga y detallada los permisos del archivo, propietario, grupo, tamaño, y fecha hora de creación.

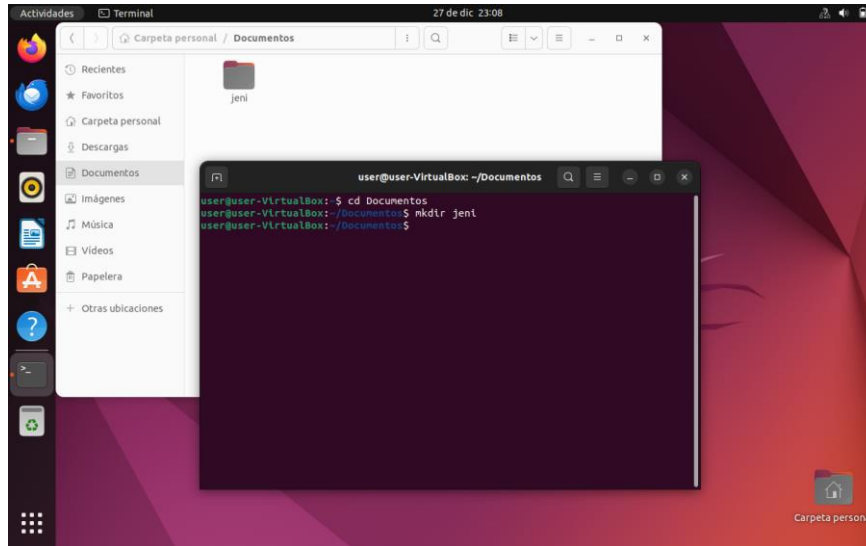


```
user@user-VirtualBox:~$ ls -l
total 36
drwxr-xr-x 2 user user 4096 dic 18 00:38 Descargas
drwxr-xr-x 2 user user 4096 dic 18 00:38 Documentos
drwxr-xr-x 2 user user 4096 dic 18 00:38 Escritorio
drwxr-xr-x 2 user user 4096 dic 18 00:38 Imágenes
drwxr-xr-x 2 user user 4096 dic 18 00:38 Música
drwxr-xr-x 2 user user 4096 dic 18 00:38 Plantillas
drwxr-xr-x 2 user user 4096 dic 18 00:38 Público
drwx----- 3 user user 4096 dic 18 00:38 snap
drwxr-xr-x 2 user user 4096 dic 18 00:38 Videos
user@user-VirtualBox:~$
```

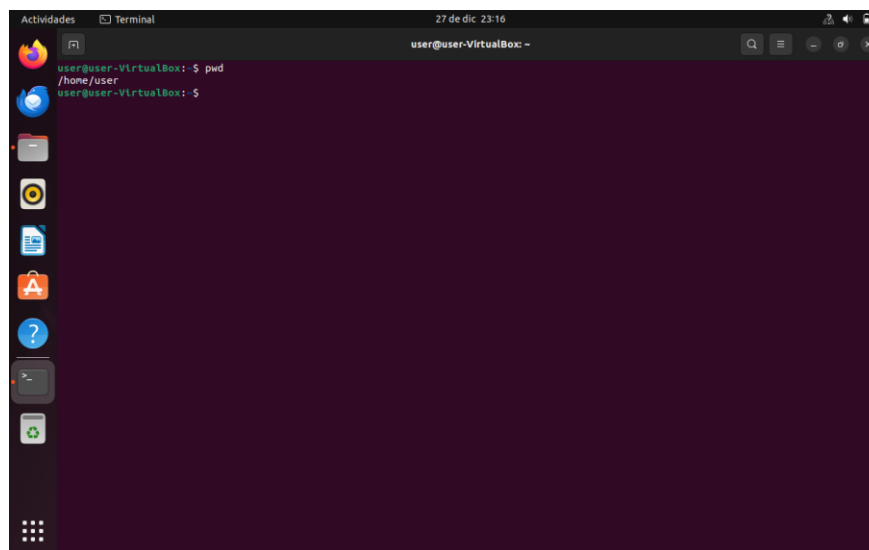
3. Para navegar por los archivos y directorios de Linux usamos *cd* donde se acompaña del nombre o ruta donde queremos ir.

4. El comando *mkdir* crea directorios uno o varios y establece permisos.

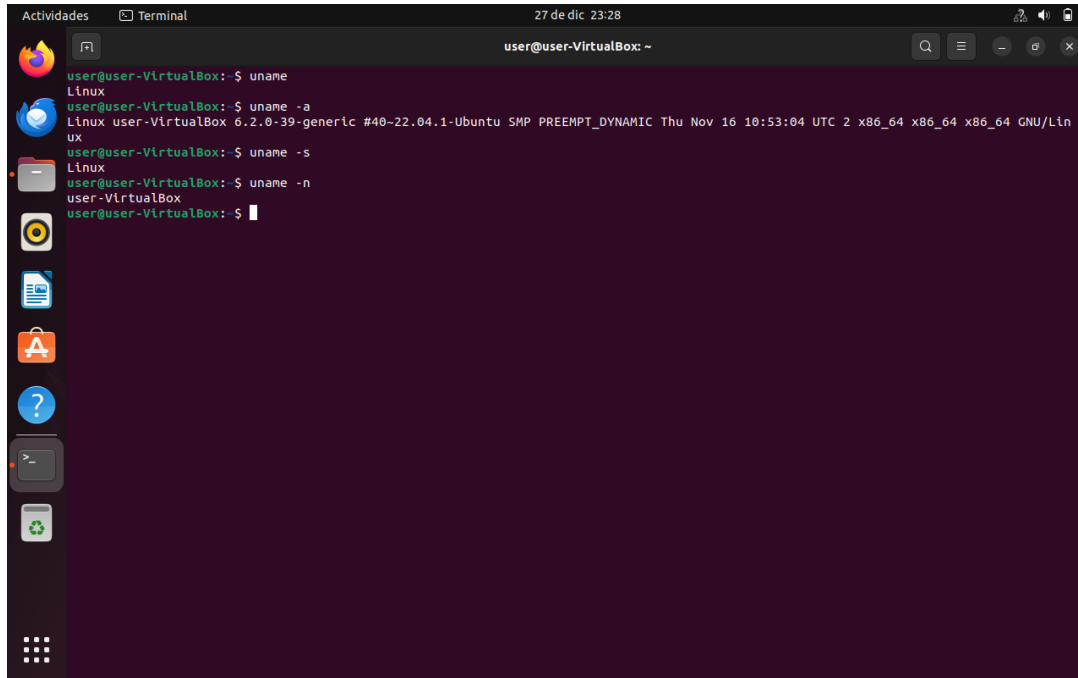
En nuestra pantalla podemos observar que una vez dentro de nuestra carpeta se creó otra utilizando los comandos anteriormente mencionados y de fondo se puede apreciar la validación de la misma.



5. Utilizamos este comando *pwd* para encontrar la ruta del directorio de trabajo actual

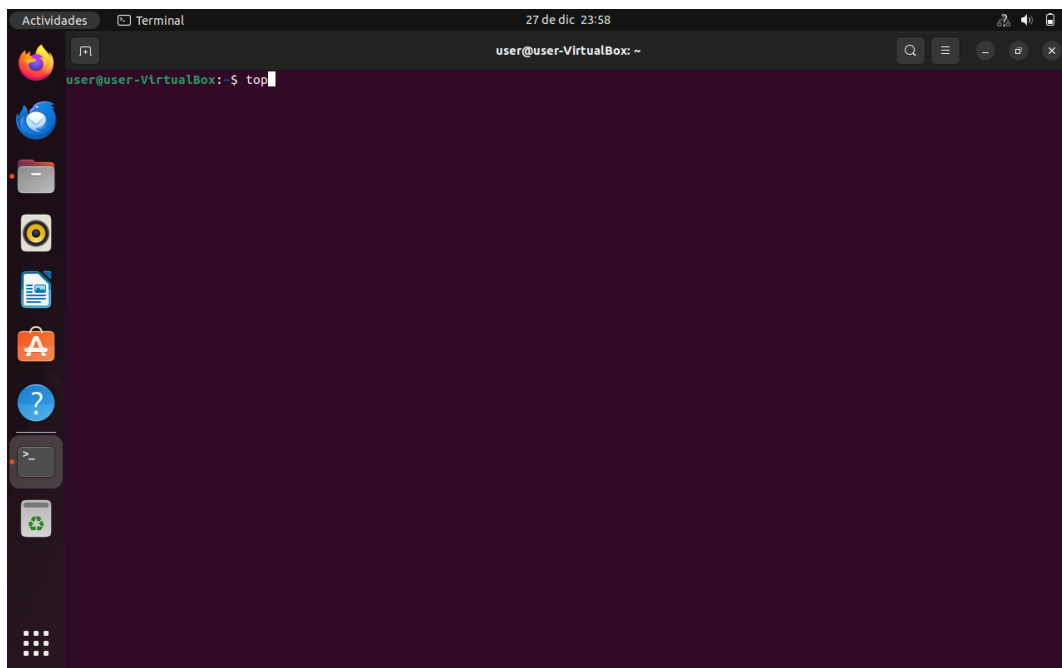


6.El comando *uname* imprime información detallada sobre el sistema Linux y el hardware. Se puede complementar con *-a* (imprime toda la información del sistema), *-s* (imprime el nombre del núcleo) y *-n* (imprime el nombre del host del nodo del sistema).



```
user@user-VirtualBox: ~  
user@user-VirtualBox:~$ uname  
Linux  
user@user-VirtualBox:~$ uname -a  
Linux user-VirtualBox 6.2.0-39-generic #40~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 16 10:53:04 UTC 2 x86_64 x86_64 x86_64 GNU/Linux  
user@user-VirtualBox:~$ uname -s  
Linux  
user@user-VirtualBox:~$ uname -n  
user-VirtualBox  
user@user-VirtualBox:~$
```

7.Comando *top* este nos mostrara todos los procesos en ejecución y una vista dinámica en tiempo real del sistema actual, nos resume la utilización de recursos desde la CPU hasta el uso de memoria.



```
user@user-VirtualBox: ~  
user@user-VirtualBox:~$ top
```

comando *top*

```
Activades [x] Terminal 27 de dic 2018:58 user@user-VirtualBox: ~  
top - 23:58:13 up 1:24, 1 user, load average: 0.37, 0.13, 0.03  
Tasks: 166 total, 2 ejecutar, 164 hibernar, 0 detener, 0 zombie  
kcpu(s): 1.7 us, 0.7 sy, 0.0 id, 97.3 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 3907.9 total, 2151.8 libre, 717.3 usado, 1038.9 búfer/cache  
MiB Intercambio: 2810.0 total, 2810.0 libre, 0.0 usado, 2933.4 dispon Mem  
  
PID USUARIO PR NI VIRT RES SHR S %CPU %MEM HORA+ ORDEN  
1443 user 20 0 3755208 347548 138276 S 1.0 8.7 1:58.95 gnome-shell  
1662 user 20 0 318024 12190 7296 S 0.3 0.3 0:03.14 dbus-daemon  
2505 user 20 0 367668 55340 42264 S 0.3 1.4 0:00.39 gnome-terminal-  
2530 user 20 0 15824 4224 3456 R 0.3 0.1 0:00.01 top  
1 root 20 0 166676 11732 8276 S 0.0 0.3 0:01.14 systemd  
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd  
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp  
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp  
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slab_flushhq  
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns  
8 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd  
10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq  
11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_kthreadd  
12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthreadd  
13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthreadd  
14 root 20 0 0 0 0 S 0.0 0.0 0:00.24 ksoftirqd/0  
15 root 20 0 0 0 0 R 0.0 0.0 0:01.26 rcu_preempt  
16 root rt 0 0 0 S 0.0 0.0 0:00.05 migration/0  
17 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0  
19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuphp/0  
20 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs  
21 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 tnet_frag_wq  
22 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd  
23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd  
25 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper  
26 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback  
28 root 20 0 0 0 0 S 0.0 0.0 0:00.51 kcompactd0  
29 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd  
30 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged  
31 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd  
32 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd  
33 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg_punt_bio  
34 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 tpm_dev_wq
```

8.El comando *kill* es utilizado para terminar manualmente un programa que no responde, pero para ello debemos de conocer su número de identificación de proceso (PID), donde para conocer el PID se ejecuta el comando *ps*

The screenshot shows a Linux desktop with a dark theme. The top panel displays the date and time as '27 de dic 23:40'. The left sidebar contains icons for the Dash, Home, and various applications including Firefox, LibreOffice, and the Dash icon. The main window is a terminal titled 'user@user-VirtualBox: ~'. The terminal output shows the command 'ps u' and its results:

```
user@user-VirtualBox:~$ ps u
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
user         1293  0.0  0.1 165184 6272 tty2    Ssl+  22:34   0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu
user         1305  0.0  0.3 225836 15872 tty2    Sl+   22:34   0:00 /usr/libexec/gnome-session-binary --session=ubuntu
user         2295  0.0  0.1  13744  5120 pts/0    Ss   23:09   0:00 bash
user         2400  0.5  0.1  15824  4224 pts/0    S+   23:29   0:02 top
user         2414  0.0  0.1  13616  5120 pts/1    Ss   23:32   0:00 bash
user         2436  0.0  0.0  15364  3456 pts/1    R+   23:37   0:00 ps u
```

Below the process list, the command 'kill -9 2400' is entered and partially executed, with the cursor visible at the end of the command.

```
Activades [x] Terminal 27 de dic 23:59
```

user@user-VirtualBox: ~

Navegador web Firefox

user@user-VirtualBox: ~

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
143	user	20	0	3755208	347548	138276	S	1.3	8.7	2:00.05	gnome-shell
2530	user	20	0	15824	4224	3456	R	0.3	0.1	0:00.10	top
1	root	20	0	166676	11732	8276	S	0.0	0.3	0:01.15	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthreadd
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthreadd
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthreadd
14	root	20	0	0	0	0	S	0.0	0.0	0:00.25	ksoftirqd/0
15	root	20	0	0	0	0	R	0.0	0.0	0:01.27	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.05	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	inet_frag_wq
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
26	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writework
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	lscmpackd
29	root	25	5	0	0	0	S	0.0	0.0	0:00.00	kmsd
30	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged
31	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	integritd
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
33	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
34	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
35	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ata_sff
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	nd

Terminado (killed)
user@user-VirtualBox: ~

9. Otro comando utilizado es *man* que proporciona un manual de usuario de cualquier comando o utilidad que pueda ejecutar en terminal, incluyendo el nombre, la descripción y las opciones.

The image shows a Linux desktop environment with a dark theme. A terminal window is open, displaying the prompt 'user@user-VirtualBox: ~' and the command 'man man' being entered. The terminal output is not visible. On the left side of the screen, there is a dock with several application icons: a web browser, a file manager, a music player (labeled 'Rhythmbox'), a document viewer, a shopping bag icon, a question mark icon, a terminal icon, and a recycling bin icon. The top of the screen shows the system menu with the date '28 de dic 00:04' and various system icons like volume and network.



```

user@user-VirtualBox: ~
MAN(1)                               Utilidades de paginador del manual                               MAN(1)

NOMBRE
man - interfaz de los manuales de referencia del sistema

SINOPSIS
man [opciones de man] [[sección] página ...] ...
man -k [opciones de apropos] regexp ...
man -k [opciones de man] [sección] term ...
man -f [whatis opciones] página ...
man -l [opciones de man] archivo ...
man -w|-W [opciones de man] página ...

DESCRIPCIÓN
man es el paginador de manuales del sistema. Cada argumento de página dado a man normalmente es el nombre de un
programa, utilidad o función. La página de manual asociada con cada uno de estos argumentos es, pues, encontrada y
mostrada. Si se proporciona una sección, man mirará solo en esa sección del manual. La acción predeterminada es buscar
en todas las secciones disponibles siguiendo un orden predefinido (véase DEFAULTS), y mostrar solo la primera página
encontrada, incluso si la página existe en varias secciones.

La tabla de abajo muestra los números de sección del manual seguidos por los tipos de página que contienen.

1  Programas ejecutables u órdenes de la shell
2  Llamadas al sistema (funciones proporcionadas por el núcleo)
3  Llamadas a biblioteca (funciones dentro de bibliotecas de programa)
4  Archivos especiales (normalmente se encuentran en /dev)
5  Formatos de archivo y convenios, p.e. /etc/passwd
6  Juegos
7  Miscelánea (incluidos paquetes de macros y convenios), p.e. man(7), groff(7), man-pages(7)
8  Órdenes de administración del sistema (normalmente solo para root)
9  Rutinas del núcleo [No estándar]

Una página de manual contiene varias secciones.

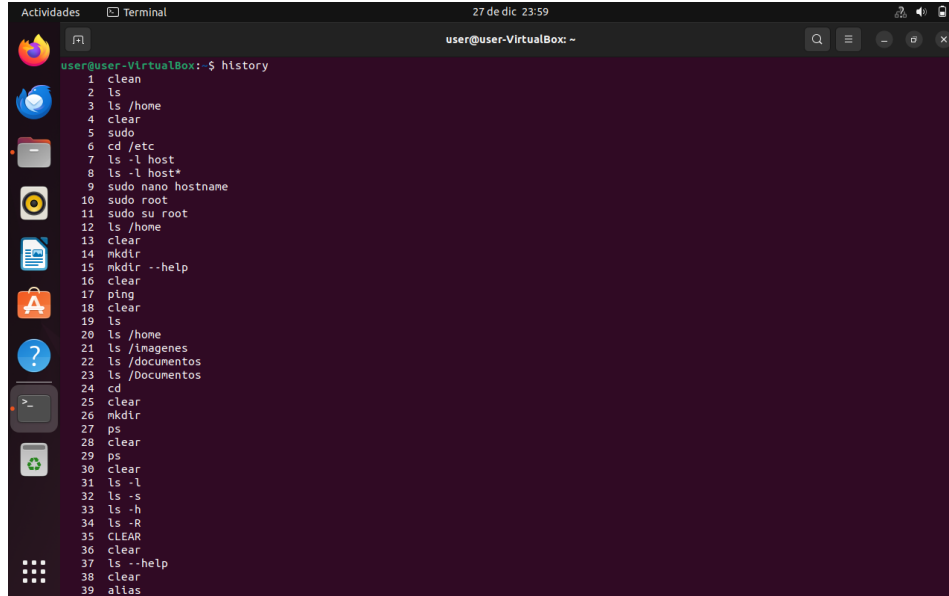
Nombres de sección convencionales: NOMBRE, SINOPSIS, CONFIGURACIÓN, DESCRIPCIÓN, OPCIONES, ESTADO DE SALIDA,
VALOR DEVUELTO, ERRORES, ENTORNO, ARCHIVOS, VERSIONES, CONFORME A, NOTAS, DEFECTOS, EJEMPLO, AUTORES, y VÉASE TAMBIÉN.

Los siguientes convenios se aplican a la sección SINOPSIS y pueden utilizarse como guía en otras secciones.

Manual page man(1) line 1 (press h for help or q to quit)

```

10.El comando *history*, este nos enlistara hasta 500 comandos ejecutados previamente, reuniéndolos sin necesidad de volver a entrar.



```

user@user-VirtualBox: ~
user@user-VirtualBox: $ history
1 clean
2 ls
3 ls /home
4 clear
5 sudo
6 cd /etc
7 ls -l host
8 ls -l host*
9 sudo nano hostname
10 sudo root
11 sudo su root
12 ls /home
13 clear
14 mkdir
15 mkdir --help
16 clear
17 ping
18 clear
19 ls
20 ls /home
21 ls /imagenes
22 ls /documentos
23 ls /Documentos
24 cd
25 clear
26 mkdir
27 ps
28 clear
29 ps
30 clear
31 ls -l
32 ls -s
33 ls -h
34 ls -R
35 CLEAR
36 clear
37 ls --help
38 clear
39 alias

```

4.3 Ahora trabajaremos específicamente con los comandos encargados del hardware de nuestro sistema.

Nuestro primer comando es `/proc/meminfo`, donde obtendremos toda la información relacionada con la memoria, para ello agregaremos el comando `cat`:

The image displays two screenshots of a Linux terminal window, showing the output of the `cat /proc/meminfo` command. The terminal window has a dark purple background and a light gray title bar.

Top Screenshot: The terminal shows the output of `cat /proc/meminfo` for a user named `user` on a system named `user-VirtualBox`. The output lists various memory statistics:

```

user@user-VirtualBox:~$ cat /proc/meminfo
MemTotal:        4601692 kB
MemFree:         2256800 kB
MemAvailable:    2925252 kB
Buffers:         33932 kB
Cached:          845864 kB
SwapCached:      0 kB
Active:          319796 kB
Inactive:        1142836 kB
Active(anon):    1788 kB
Inactive(anon):  613088 kB
Active(file):    318008 kB
Inactive(file):  529748 kB
Unevictable:     0 kB
Mlocked:         0 kB
SwapTotal:       2877436 kB
SwapFree:        2877436 kB
Zswap:           0 kB
Zswapped:        0 kB
Dirty:           0 kB
Writeback:       0 kB
AnonPages:       582888 kB
Mapped:          289756 kB
Shmem:           37140 kB
KReclaimable:    46496 kB
Slab:            131396 kB
SReclaimable:    46496 kB
SUnreclaim:     84900 kB
KernelStack:    6352 kB
PageTables:      13136 kB
SecPageTables:   0 kB
NFS_Unstable:    0 kB
Bounce:         0 kB
WritebackTmp:    0 kB
CommitLimit:     4878280 kB
Committed_AS:    3513328 kB
VmallocTotal:    34359738367 kB
VmallocUsed:     26316 kB
VmallocChunk:    0 kB
Percpu:          732 kB

```

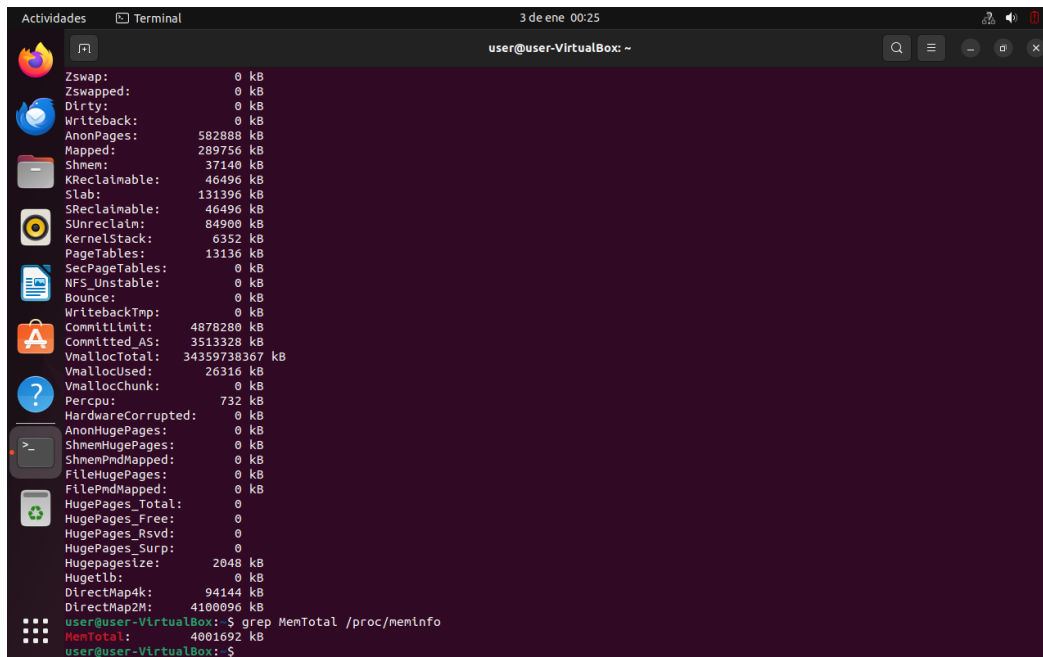
Bottom Screenshot: The terminal shows the output of `cat /proc/meminfo` again, but with additional statistics and a list of memory-mapped files:

```

user@user-VirtualBox:~$ cat /proc/meminfo
MemTotal:        2877436 kB
MemFree:         2877436 kB
MemAvailable:    2877436 kB
Buffers:         0 kB
Cached:          0 kB
SwapCached:      0 kB
Active:          0 kB
Inactive:        0 kB
Active(anon):    0 kB
Inactive(anon):  0 kB
Active(file):    0 kB
Inactive(file):  0 kB
Unevictable:     0 kB
Mlocked:         0 kB
SwapTotal:       0 kB
SwapFree:        0 kB
Zswap:           0 kB
Zswapped:        0 kB
Dirty:           0 kB
Writeback:       0 kB
AnonPages:       582888 kB
Mapped:          289756 kB
Shmem:           37140 kB
KReclaimable:    46496 kB
Slab:            131396 kB
SReclaimable:    46496 kB
SUnreclaim:     84900 kB
KernelStack:    6352 kB
PageTables:      13136 kB
SecPageTables:   0 kB
NFS_Unstable:    0 kB
Bounce:         0 kB
WritebackTmp:    0 kB
CommitLimit:     4878280 kB
Committed_AS:    3513328 kB
VmallocTotal:    34359738367 kB
VmallocUsed:     26316 kB
VmallocChunk:    0 kB
Percpu:          732 kB
HardwareCorrupted: 0 kB
AnonHugePages:   0 kB
ShmemHugePages:  0 kB
ShmemPmdMapped:  0 kB
FileHugePages:   0 kB
FilePmdMapped:   0 kB
HugePages_Total: 0
HugePages_Free:  0
HugePages_Rsvd:  0
HugePages_Surp:  0
Hugepagesize:    2048 kB
Hugetlb:         0 kB
DirectMap4k:    94144 kB
DirectMap2M:    410096 kB

```

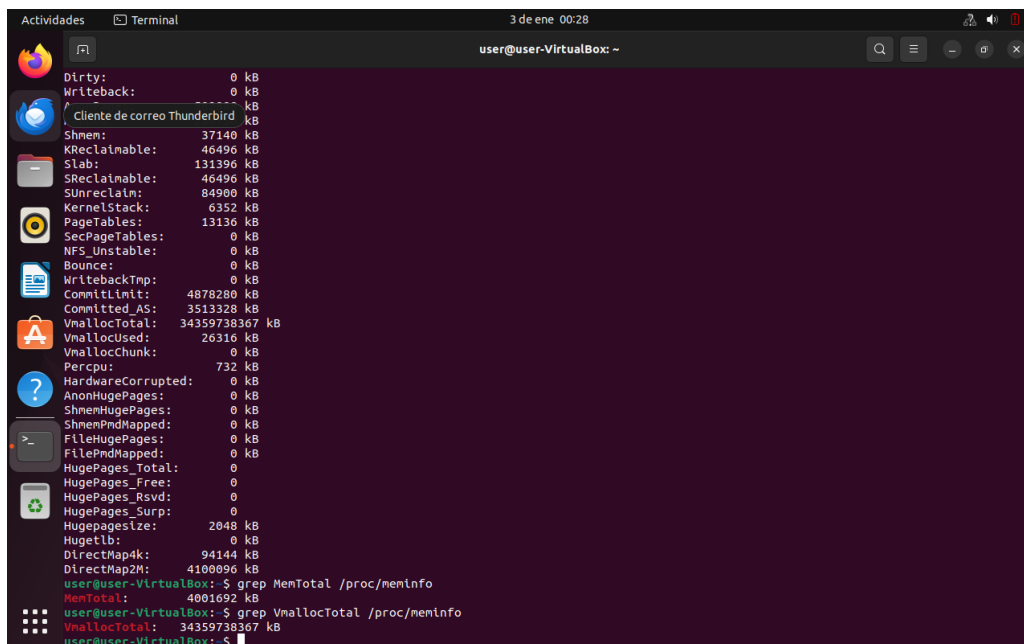
Para ver la memoria física que usa el archivo utilizaremos *grep MemTotal /proc/meminfo*



```
Actividades Terminal 3 de ene 00:25
user@user-VirtualBox: ~

Zswap: 0 kB
Zswapped: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 582888 kB
Mapped: 289756 kB
Shmem: 37140 kB
KReclaimable: 46496 kB
Slab: 131396 kB
SReclaimable: 46496 kB
SUnreclaim: 84900 kB
KernelStack: 6352 kB
PageTables: 13136 kB
SecPageTables: 0 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 4878280 kB
Committed_AS: 3513328 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 26316 kB
VmallocChunk: 0 kB
Percpu: 732 kB
HardwareCorrupted: 0 kB
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
FileHugePages: 0 kB
FilePmdMapped: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
Hugetlb: 0 kB
DirectMap4k: 94144 kB
DirectMap2M: 4100096 kB
user@user-VirtualBox: $ grep MemTotal /proc/meminfo
MemTotal: 4001692 kB
user@user-VirtualBox: $
```

Y para ver la memoria virtual utilizamos *grep VmallocTotal /proc/meminfo*



```
Actividades Terminal 3 de ene 00:28
user@user-VirtualBox: ~

Dirty: 0 kB
Writeback: 0 kB
Shmem: 37140 kB
KReclaimable: 46496 kB
Slab: 131396 kB
SReclaimable: 46496 kB
SUnreclaim: 84900 kB
KernelStack: 6352 kB
PageTables: 13136 kB
SecPageTables: 0 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 4878280 kB
Committed_AS: 3513328 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 26316 kB
VmallocChunk: 0 kB
Percpu: 732 kB
HardwareCorrupted: 0 kB
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
FileHugePages: 0 kB
FilePmdMapped: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
Hugetlb: 0 kB
DirectMap4k: 94144 kB
DirectMap2M: 4100096 kB
user@user-VirtualBox: $ grep MemTotal /proc/meminfo
MemTotal: 4001692 kB
user@user-VirtualBox: $ grep VmallocTotal /proc/meminfo
VmallocTotal: 34359738367 kB
user@user-VirtualBox: $
```

El segundo comando es *top* con el cual podemos monitorear los procesos y el uso de recursos del sistema, con una vista real:

```

top - 00:29:36 up 17 min, 1 user, load average: 0.08, 0.05, 0.07
Tareas: 165 total, 1 ejecutar, 164 hibernar, 0 detener, 0 zombie
           0.0 ni, 98.3 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem: 2203.2 total, 794.1 libre, 910.6 búfer/cache
MiB Intercambio: 2810.0 total, 2810.0 libre, 0.0 usado, 2861.9 dispon Men

  PID USUARIO PR NI VIRT RES SHR S %CPU %MEM    HORA+ ORDEN
1457 user    20  0 3754268 347488 135976 S   1.0  8.7   0:17.45 gnome-shell
  1 root      20  0 166676 11884 8428 S   0.0  0.3   0:01.07 systemd
  2 root      20  0 0 0 0 S   0.0  0.0   0:00.00 kthreadd
  3 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 rcu_gp
  4 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 rcu_par_gp
  5 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 slub_flushwq
  6 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 netns
  8 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kworker/0:0H-kblockd
 10 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 mm_percpu_wq
 11 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_kthread
 12 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_rude_kthread
 13 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_trace_kthread
 14 root      20  0 0 0 0 S   0.0  0.0   0:00.17 ksoftirqd/0
 15 root      20  0 0 0 0 I   0.0  0.0   0:00.55 rcu_preempt
 16 root      rt  0 0 0 0 S   0.0  0.0   0:00.00 migration/0
 17 root     -51  0 0 0 0 S   0.0  0.0   0:00.00 idle_inject/0
 19 root      20  0 0 0 0 S   0.0  0.0   0:00.00 cpuhp/0
 20 root      20  0 0 0 0 S   0.0  0.0   0:00.00 kdevtmpfs
 21 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 inet_frag_wq
 22 root      20  0 0 0 0 S   0.0  0.0   0:00.00 kauditd
 23 root      20  0 0 0 0 S   0.0  0.0   0:00.00 khungtaskd
 25 root      20  0 0 0 0 S   0.0  0.0   0:00.00 oom_reaper
 26 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 writeback
 28 root      20  0 0 0 0 S   0.0  0.0   0:00.11 kcompactd0
 29 root      25  5 0 0 0 S   0.0  0.0   0:00.00 ksm
 30 root      39 19 0 0 0 S   0.0  0.0   0:00.00 khugepaged
 31 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kintegrityd
 32 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kblockd
 33 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 blkcg_punt_bio
 34 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 tpm_dev_wq
 35 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 ata_sff
 36 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 md
 37 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 edac-poller

```

Para ver específicamente los procesos de algún usuario podemos utilizar la variante *U* de esta forma, *top -U username*

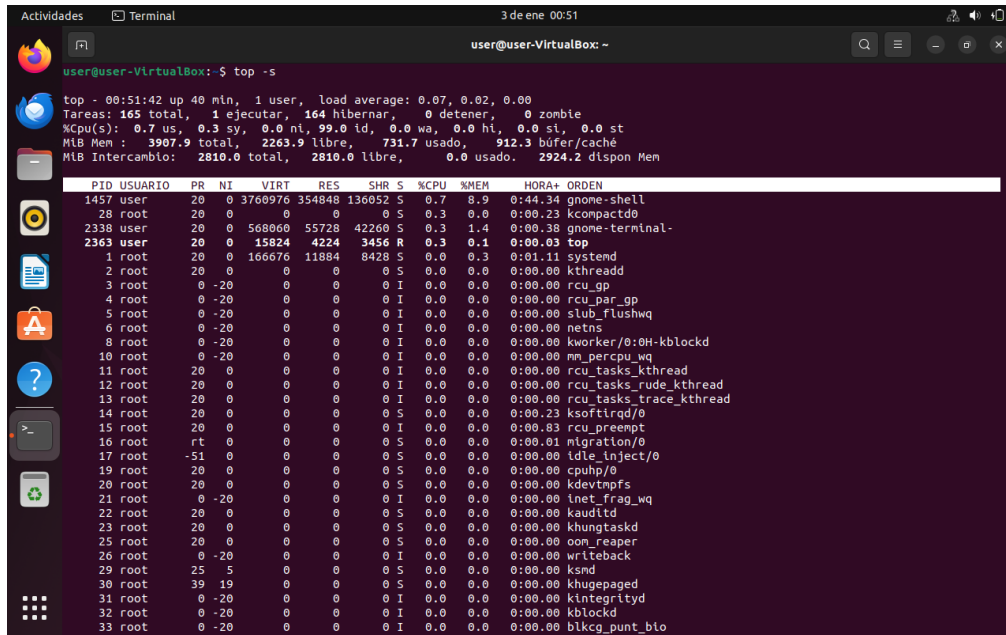
```

top - 00:38:04 up 26 min, 1 user, load average: 0.07, 0.06, 0.06
Tareas: 167 total, 1 ejecutar, 166 hibernar, 0 detener, 0 zombie
           5.0 us, 1.0 sy, 0.0 nt, 93.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem: 3907.9 total, 2249.4 libre, 747.4 usado, 911.1 búfer/cache
MiB Intercambio: 2810.0 total, 2810.0 libre, 0.0 usado, 2908.6 dispon Men

  PID USUARIO PR NI VIRT RES SHR S %CPU %MEM    HORA+ ORDEN
  3 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 rcu_gp
  4 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 rcu_par_gp
  5 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 slub_flushwq
  6 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 netns
  8 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kworker/0:0H-kblockd
 10 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 mm_percpu_wq
 11 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_kthread
 12 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_rude_kthread
 13 root      20  0 0 0 0 I   0.0  0.0   0:00.00 rcu_tasks_trace_kthread
 14 root      20  0 0 0 0 S   0.0  0.0   0:00.19 ksoftirqd/0
 15 root      20  0 0 0 0 I   0.0  0.0   0:00.66 rcu_preempt
 16 root      rt  0 0 0 0 S   0.0  0.0   0:00.01 migration/0
 17 root     -51  0 0 0 0 S   0.0  0.0   0:00.00 idle_inject/0
 19 root      20  0 0 0 0 S   0.0  0.0   0:00.00 cpuhp/0
 20 root      20  0 0 0 0 S   0.0  0.0   0:00.00 kdevtmpfs
 21 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 inet_frag_wq
 22 root      20  0 0 0 0 S   0.0  0.0   0:00.00 kauditd
 23 root      20  0 0 0 0 S   0.0  0.0   0:00.00 khungtaskd
 25 root      20  0 0 0 0 S   0.0  0.0   0:00.00 oom_reaper
 26 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 writeback
 28 root      20  0 0 0 0 S   0.0  0.0   0:00.16 kcompactd0
 29 root      25  5 0 0 0 S   0.0  0.0   0:00.00 ksm
 30 root      39 19 0 0 0 S   0.0  0.0   0:00.00 khugepaged
 31 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kintegrityd
 32 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 kblockd
 33 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 blkcg_punt_bio
 34 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 tpm_dev_wq
 35 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 ata_sff
 36 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 md
 37 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 edac-poller
 38 root      0 -20 0 0 0 I   0.0  0.0   0:00.00 devfreq_wq

```

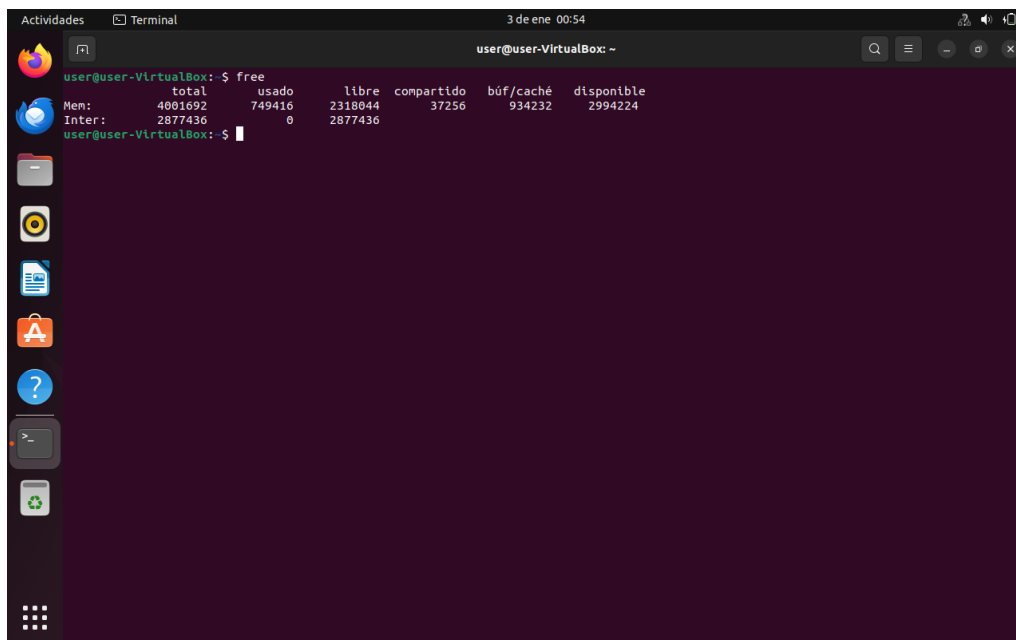
También dentro del comando *top* podemos evitar realizar modificaciones en nuestro equipo y para ella podemos abrirlo en modo seguro utilizando *top -s*



```
user@user-VirtualBox:~$ top -s
top - 00:51:42 up 40 min, 1 user, load average: 0.07, 0.02, 0.00
Tasks: 165 total, 1 ejecutar, 164 hibernar, 0 detener, 0 zombie
%Cpu(s): 0.7 us, 0.3 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3907.9 total, 2263.9 libre, 731.7 usado, 912.3 búfer/cache
MiB Intercambio: 2810.0 total, 2810.0 libre, 0.0 usado, 2924.2 dispon Mem

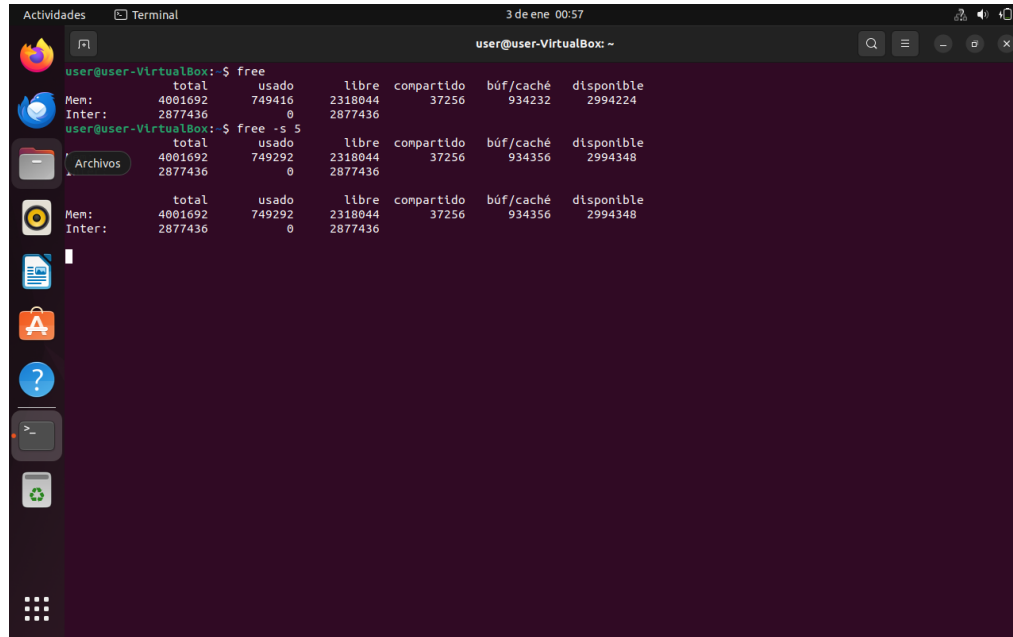
  PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  HORA+ ORDEN
    1457 user      20   0 3760976 354848 136052 S   0.7   8.9  0:44.34 1  gnome-shell
    28 root       20   0 0 0 0 S   0.3   0.0  0:00.23 2  kcompactd0
   2338 user      20   0 5680608 55728 42260 S   0.3   1.4  0:00.38 3  gnome-terminal-
  2363 user      20   0 15924 4224 3456 R   0.3   0.1  0:00.03 4  top
    1 root       20   0 166676 11884 8428 S   0.0   0.3  0:01.11 5  systemd
    2 root       20   0 0 0 0 S   0.0   0.0  0:00.00 6  kthreadd
    3 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 7  rcu_gp
    4 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 8  rcu_par_gp
    5 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 9  slub_flushwq
    6 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 10 netns
    8 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 11 kworker/0:0H-kblockd
   10 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 12 mm_percpu_wq
   11 root       20   0 0 0 0 I   0.0   0.0  0:00.00 13 rcu_tasks_kthread
   12 root       20   0 0 0 0 I   0.0   0.0  0:00.00 14 rcu_tasks_rude_kthread
   13 root       20   0 0 0 0 I   0.0   0.0  0:00.00 15 rcu_tasks_trace_kthread
   14 root       20   0 0 0 0 S   0.0   0.0  0:00.23 16 ksoftirqd/0
   15 root       20   0 0 0 0 I   0.0   0.0  0:00.83 17 rcu_preempt
   16 root       rt   0 0 0 0 S   0.0   0.0  0:00.01 18 migration/0
   17 root       -51  0 0 0 0 S   0.0   0.0  0:00.00 19 idle_inject/0
   19 root       20   0 0 0 0 S   0.0   0.0  0:00.00 20 cpuhp/0
   20 root       20   0 0 0 0 S   0.0   0.0  0:00.00 21 kdevtmpfs
   21 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 22 inet_frag_wq
   22 root       20   0 0 0 0 S   0.0   0.0  0:00.00 23 kauditd
   23 root       20   0 0 0 0 S   0.0   0.0  0:00.00 24 khungtaskd
   25 root       20   0 0 0 0 S   0.0   0.0  0:00.00 25 oom_reaper
   26 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 26 writeback
   29 root       25   5 0 0 0 S   0.0   0.0  0:00.00 27 ksm
   30 root       39  19 0 0 0 S   0.0   0.0  0:00.00 28 khugepaged
   31 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 29 kintegrityd
   32 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 30 kblockd
   33 root       0 -20 0 0 0 I   0.0   0.0  0:00.00 31 blkcg_punt_bto
```

Ahora conoceremos el comando *free* que nos sirve para conocer la memoria libre como la utilizada que hay en el sistema, así como la cantidad total de memoria física y de intercambio:



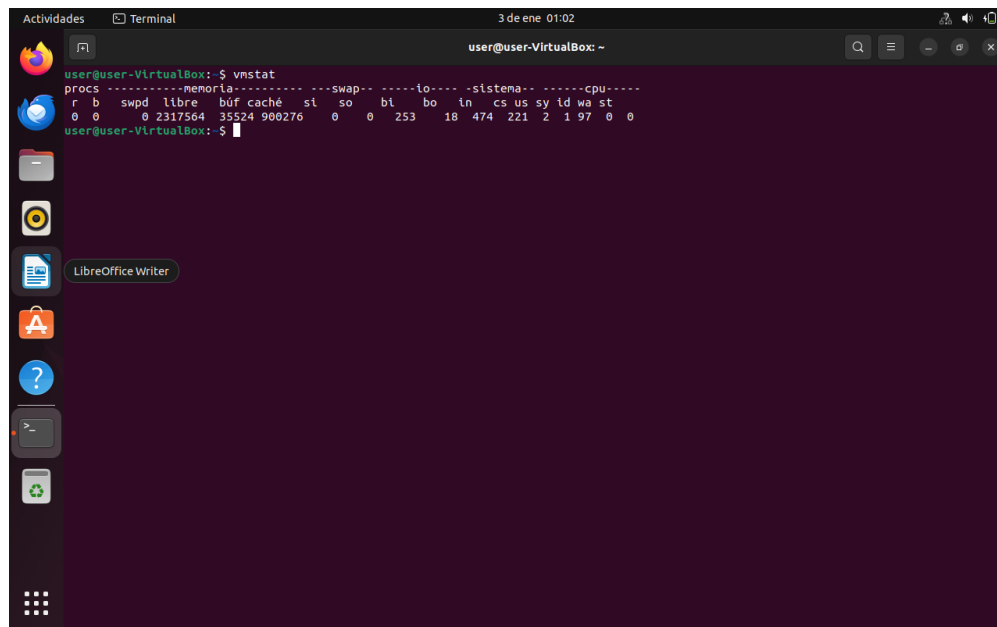
```
user@user-VirtualBox:~$ free
              total        usado         libre compartido búf/cache disponible
Mem:          4001692       749416       2318044         37256       934232       2994224
Inter:         2877436           0       2877436
user@user-VirtualBox:~$
```

Si utilizamos el comando `free` durante un intervalo de tiempo designado en segundos seria de la siguiente manera `free -s 5`



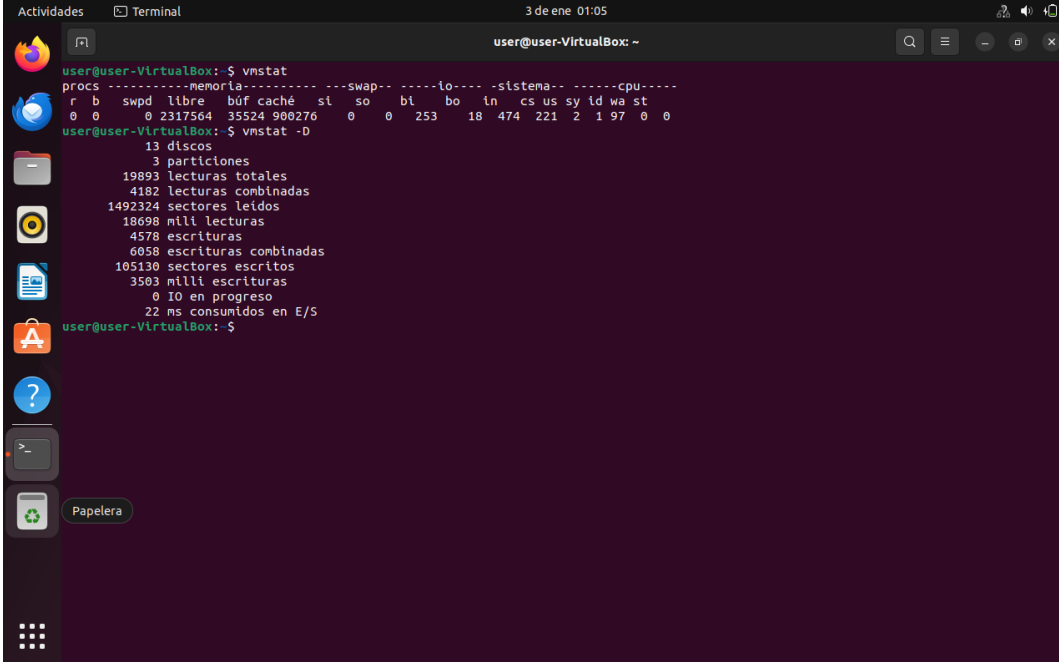
```
user@user-VirtualBox:~$ free
total        usado        libre compartido    búf/caché    disponible
Mem: 4001692  749416      2318044      37256      934232      2994224
Inter: 2877436           0      2877436
user@user-VirtualBox:~$ free -s 5
total        usado        libre compartido    búf/caché    disponible
Mem: 4001692  749292      2318044      37256      934356      2994348
Inter: 2877436           0      2877436
user@user-VirtualBox:~$
```

Por último, tenemos el comando `vmstat` el cual monitorea el rendimiento en Linux, nos brinda información sobre procesos, memoria, paginación, disco E/S de bloqueo y programación de CPU. Informa también estadísticas de memoria virtual:



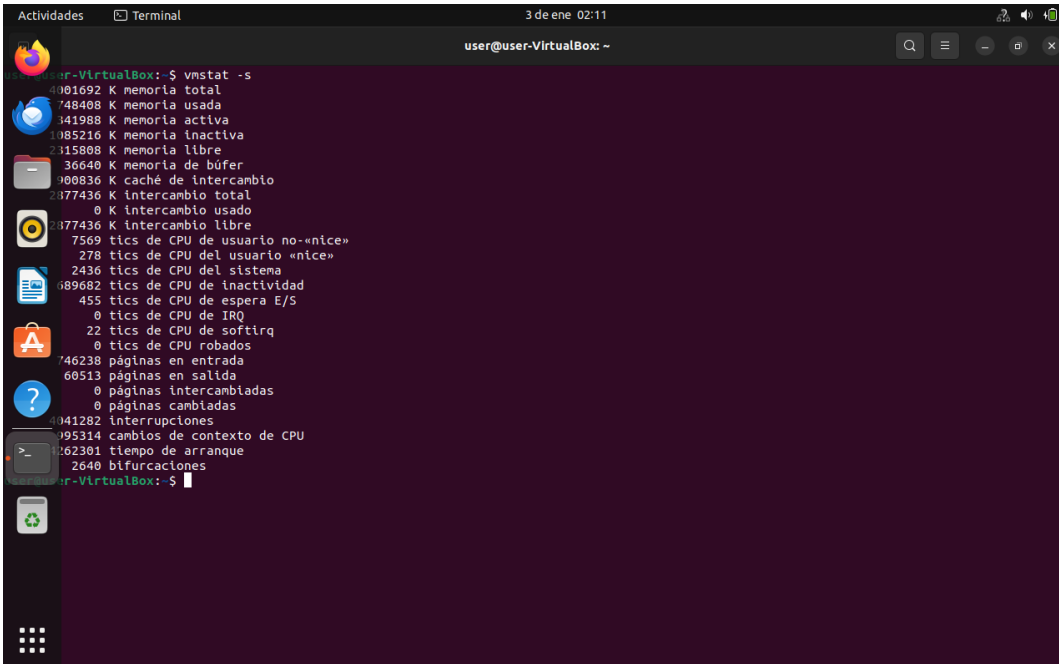
```
user@user-VirtualBox:~$ vmstat
procs -----memoria----- --swap-- -----io----- -sistema-- -----cpu-----
r b swpd libre búf caché si so bi bo in cs us sy id wa st
0 0 0 2317564 35524 900276 0 0 253 18 474 221 2 1 97 0 0
user@user-VirtualBox:~$
```

Obtenemos un reporte detallado de la actividad del disco duro de esta forma, *vmstat -D*

A terminal window titled 'Terminal' with a dark background. The prompt is 'user@user-VirtualBox: ~'. The command 'vmstat' has been executed, showing a table of system statistics. Below it, the command 'vmstat -D' has been executed, showing detailed disk activity statistics.

```
user@user-VirtualBox:~$ vmstat
procs -----memoria-----swap-- ----io---- -sistema-- -----cpu-----
r  b  swpd  libre  búf  caché  si   so   bi   bo   in   cs  us  sy  id  wa  st
0  0    0 2317564 35524 900276  0   0  253  18 474 221  2  1 97  0  0
user@user-VirtualBox:~$ vmstat -D
13 discos
3 particiones
19893 lecturas totales
4182 lecturas combinadas
1492324 sectores leídos
18698 milli lecturas
4578 escrituras
6058 escrituras combinadas
105130 sectores escritos
3503 milli escrituras
0 IO en progreso
22 ms consumidos en E/S
user@user-VirtualBox:~$
```

Si deseamos ver los eventos del sistema vamos a ejecutar *vmstat -s*

A terminal window titled 'Terminal' with a dark background. The prompt is 'user@user-VirtualBox: ~'. The command 'vmstat -s' has been executed, showing a list of system events and their counts.

```
user@user-VirtualBox:~$ vmstat -s
4001692 K memoria total
748408 K memoria usada
141988 K memoria activa
1085216 K memoria inactiva
2115808 K memoria libre
36640 K memoria de búfer
900836 K caché de intercambio
2877436 K intercambio total
0 K intercambio usado
2877436 K intercambio libre
7569 tics de CPU de usuario no-«nice»
278 tics de CPU del usuario «nice»
2436 tics de CPU del sistema
89682 tics de CPU de inactividad
455 tics de CPU de espera E/S
0 tics de CPU de IRQ
22 tics de CPU de softirq
0 tics de CPU robados
746238 páginas en entrada
60513 páginas en salida
0 páginas intercambiadas
0 páginas cambiadas
4041282 interrupciones
195314 cambios de contexto de CPU
1262301 tiempo de arranque
2640 bifurcaciones
user@user-VirtualBox:~$
```

CONCLUSION

En esta actividad pudimos aprender a usar algunos comandos que nos sirven para saber información relacionada con nuestro hardware, utilizamos algunos comandos para saber su información y así lograr utilizar nuestro equipo de una manera rápida y fluida, logrando evitar que se sature y evitemos alguna descompostura, alargando más la vida útil de nuestro equipo y también logramos ampliar nuestro conocimiento en este rubro y logrando aprender mas comandos para agilizar nuestros trabajos que se vuelven complejos, es bueno saber utilizar comandos ya que nos puede ayudar a realizar tareas complejas de una forma más fácil y eficiente, en conclusión ampliamos mas nuestro conocimiento.

REFERENCIAS

Mario Pérez Esteso. 05 AGO. 2014. Funcionamiento de comandos en Linux.

[Funcionamiento del comando TOP en Linux \(geekytheory.com\)](https://www.geekytheory.com/funcionamiento-del-comando-top-en-linux/)

Julio 25, 2023. Free: el comando para saber todo sobre nuestra RAM.

[free: el comando para saber todo sobre nuestra RAM \(voidnull.es\)](https://voidnull.es/free-el-comando-para-saber-todo-sobre-nuestra-ram/)