UNIVERSIDAD MÉXICO INTERNACIONAL

Ingeniería en Desarrollo de Software

Actividad | 2 | Solución de Problemas

Nombre del curso

Ingeniería en Desarrollo de Software

TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 12 de junio del 2025.

Tabla de Contenido

Introducción

Esta es la segunda actividad en donde nos centraremos en los comandos de Linux, comandos que aportan características al sistema como a la red, esto es indispensable para entender como algunas funciones ayudan a conocer más sobre el sistema en el que estamos trabajando y también observar los cambios que ocurren en tiempo real o por lapsos determinados. Linux es un sistema operativo que permite a los usuarios tener mayor libertad en cuanto a las acciones que se pueden realizar y también en la seguridad del sistema ya como sabemos la gran comunidad se ha encargado de mantener vivo este sistema. Los comandos son instrucciones que se le dan al sistema, normalmente se utiliza la terminal para ejecutar dichas instrucciones y que estas tengan un impacto, la ventaja es que estos comandos si se quisieran descargar no robaría mucho espacio ya que estos no tienen una interfaz gráfica, solo son líneas de texto que se ejecutan y muestran información.

Descripción

Los comandos de Linux fueron creados para que los usuarios pudieran interactuar con el sistema operativo de manera específica, otorgando acciones y ejecutándolas según el usuario necesitará, comandos tan simples como ver una parte del hardware instalado hasta crear servidores que se comunican entre sí y mientras los dispositivos tengan una conexión a internet y su sistema se los permita estos pueden mostrar la información que el servidor posee dentro de sí mismo. En la actividad pasada se utilizaron ciertos comandos que permitían crear un servidor y con la dirección IP de la “VirtualBox” se visualizaba la página, se lograba apreciar en los dispositivos como PC, celulares (Android e IOS), la consola, la SMARTV y por supuesto también funcionaba con la PSP 3000, todos los dispositivos permitían observar el pequeño servidor que se había creado en la “VirtualBox”, esto nos dice que Linux y sus comandos nos abre puertas hacia grandes posibilidades puesto que hay que recordar que este sistema se encuentra en supercomputadoras como en celulares.

Justificación

De acuerdo a lo que hemos revisado en la actividad anterior y lo que se va a investigar y ejecutar en esta actividad nos queda una pregunta, ¿Por qué es importante utilizar los comandos de Linux?, esta pregunta nos abre la puerta a diferentes respuestas que pueden ser universales o que van de la mano de cada uno de los usuarios y la manera en la que estos gestionan su ordenador. Como ya se vio anteriormente, los comandos ayudan a ejecutar tareas en el sistema, una serie de pasos a completar con la sintaxis correcta que

permite al usuario observar cómo se comporta el sistema hasta, lograr utilizar diferentes opciones a su favor, nos da un control total sobre el sistema, nosotros somos los que tenemos la decisión de crear, eliminar o suspender procesos o tareas que asignan los comandos específicos, la eficiencia con la que trabajan los comandos permiten ejecutar las tareas de manera rápida y fluida, esto a su vez es porque estos solo son líneas de código y no tienen una interfaz gráfica, por ello cuando se trata de descargar una nueva herramienta no se necesita un gran espacio de memoria, los diagnósticos que realizan los comandos en tiempo real son útiles y necesarios para mantener el ordenador o dispositivo optimo y en buenas condiciones.

Investigación

Comandos para monitorear el rendimiento del sistema

- **top:** muestra de manera dinámica los procesos que está llevando a cabo el sistema en tiempo real (uso de CPU, uso de memoria, el promedio de carga., entre otros).
- **htop:** este comando de Linux nos abre una interfaz amigable y fácil de utilizar para observar los procesos que están en ejecución (uso de memoria, CPU, el promedio de carga, entre otros) es más interactiva que el comando **top**.
- **vmstat:** recopila y muestra estadísticas sobre el rendimiento del sistema, tiene variables que muestran determinadas acciones y tareas:
    - **vmsat 2:** muestra las estadísticas cada dos segundos.
    - **vmsat –s:** muestra un resumen de las estadísticas desde el inicio del sistema.
    - **vmstat –d:** muestra estadísticas de uso del disco.
    - **vmstat –p sda1:** muestra estadísticas de la partición sda1.
- **free:** muestra información de la memoria RAM y la memoria de intercambio (swap) en un sistema.
- **tlp:** optimizar el sobrecalentamiento con la gestión adecuada de tareas y procesos.
- **cat/proc/meminfo:** muestra en tiempo real el uso de la memoria del sistema en tiempo real.
- **lshw:** muestra información detallada sobre el hardware del sistema.

Nota: todos los comandos anteriores se mostrarán en la sección “monitoreo del sistema” junto con la presentación de capturas de pantalla, así como una descripción de lo que se aprecia en ella.

Comandos para monitorear el rendimiento de la red

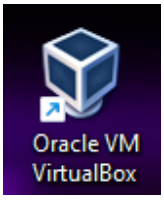
- **tcpdump:** analiza el tráfico de una red en específico, también ayuda a depurar problemas de la misma.
- **netstat:** muestra la conexión de redes que tengan una conexión activa, y los puertos de escucha.
- **iftop:** monitorea el tráfico de red en tiempo real, como también ayuda a detectar cuellos de botella que existan en nuestra conexión.
- **ss:** es una herramienta más moderna y rápida que **netstat** que ayuda a comprobar las conexiones de la red, así como los puertos abiertos.
- **ifconfig:** muestra la información de las interfaces de red.
- **traceroute:** muestra la ruta que sigue un paquete de datos desde su origen hasta su destino.
- **ping:** verifica la conectividad de varios dispositivos dentro de una red.

Nota: todos los comandos anteriores se mostrarán en la sección “monitoreo de la red” junto con la presentación de capturas de pantalla, así como una descripción de lo que se aprecia en ella.

Desarrollo

Monitoreo del sistema

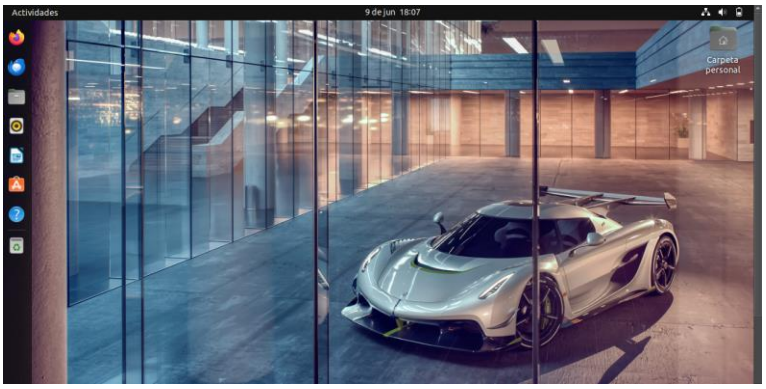
Iniciaremos primero por dirigirnos a nuestro escritorio donde tenemos instalado nuestra VirtualBox:



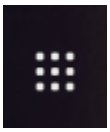
Lo seleccionamos y nos abrirá la siguiente ventana, donde existen diferentes características:

Procedemos a dar inicio y marcha de nuestra VirtualBox para que empecemos con la creación del servidor:

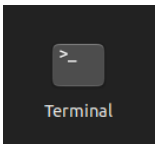
Y tenemos el escritorio de nuestro sistema operativo de Linux:



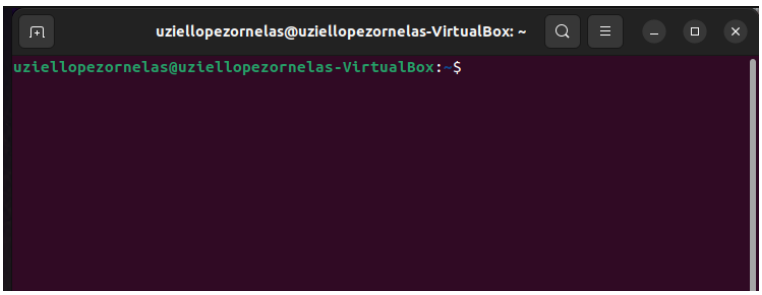
Nos deslizamos en la parte de abajo para que se nos muestren las diferentes aplicaciones que el SO maneja y buscaremos una en específico:



Lo seleccionamos para después ubicar la aplicación de “Terminal”:



Al seleccionarlo nos abrirá nuestra ventana de comandos:



Ahora empezaremos con el monitoreo del sistema en donde colocaremos el primer comando que nos mostrará varias características del sistema, este comando serio "**top**":

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

```
top - 13:14:56 up 4 min,  1 user,  load average: 0.16, 0.42, 0.23
Tareas: 180 total,   2 ejecutar,  178 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 :   1959.7 total,    209.4 libre,    974.6 usado,    775.7 búfer/caché
MiB Intercambio:   2680.0 total,   2643.0 libre,     37.0 usado.    810.8 dispon

    PID USUARIO   PR  NI    VIRT    RES    SHR S  %CPU  %MEM     HORA+ ORDEN
   1736 uziello+  20   0 3453980 283544 105460 S   2.3  14.1   0:06.52 gnome-s+
    665 root      20   0  242892   8124   7228 S   0.3   0.4   0:00.09 account+
   2842 uziello+  20   0  564968  47896  35084 S   0.3   2.4   0:00.67 gnome-t+
   3640 uziello+  20   0   15828   4224   3456 R   0.3   0.2   0:00.22 top
      1 root      20   0  167908  12896   7904 S   0.0   0.6   0:01.20 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+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root      20   0       0      0      0 I   0.0   0.0   0:00.57 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
```

Lo que muestra esta captura de pantalla es la hora en la que se realizó el comando:

```
top - 13:14:56
```

El sistema ha estado encendido por cuatro minutos aproximadamente:

```
up 4 min,
```

Hay un usuario activo en el sistema:

```
1 user
```

La carga promedio del sistema en los últimos minutos e indica también el numero promedio de procesos que se están ejecutando o están en espera:

```
load average: 0.16, 0.42, 0.23
```

Uso de la CPU por categorías:

```
%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
```

Información de la memoria RAM en el sistema:

```
MiB Mem :   1959.7 total,    209.4 libre,    974.6 usado,    775.7 búfer/caché
```

Memoria de intercambio en el sistema:

```
MiB Intercambio:   2680.0 total,   2643.0 libre,     37.0 usado.    810.8 dispon
```

En total tenemos “180” procesos, “2” en ejecución, “178” procesos en suspensión, “0” que están detenidos y “0” procesos “zombies”:

```
Tareas: 180 total,   2 ejecutar,  178 hibernar,    0 detener,    0 zombie
```

El “PDI” es el identificador único del usuario, “USUARIO” es el usuario propietario del proceso:

```
   PID USUARIO
  1736 uziello+
   665 root
  2842 uziello+
  3640 uziello+
     1 root
     2 root
     3 root
     4 root
     5 root
     6 root
     7 root
     8 root
     9 root
    10 root
    11 root
    12 root
    13 root
```

El “PR” es la prioridad del proceso:

```
 PR
 20
 20
 20
 20
 20
 20
  0
  0
  0
  0
 20
  0
 20
  0
 20
 20
 20
```

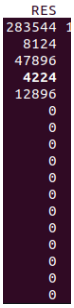
El “NI” es el valor de “nice” que afecta su propiedad:

```
 NI
  0
  0
  0
  0
  0
  0
-20
-20
-20
-20
  0
-20
  0
-20
  0
  0
  0
```

El “VIRT” es la memoria virtual total utilizada en el proceso:

```
  VIRT
3453980
 242892
 564968
  15828
 167908
      0
      0
      0
      0
      0
      0
      0
      0
      0
      0
      0
```

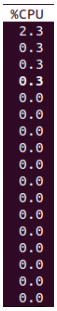
El “RES” es la memoria física no intercambiada en el proceso:



El “SHR” es la memoria compartida en el proceso:

El “S” es el estatus:

El “%CPU” es el porcentaje de uso de la CPU por el proceso:



El “%MEM” es el porcentaje de uso de la memoria física por el proceso:

```
%MEM
14.1
 0.4
 2.4
 0.2
 0.6
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
 0.0
```

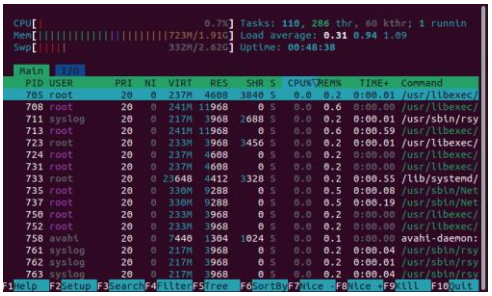
La “HORA+” es el tiempo que la CPU ha consumido el proceso:

```
  HORA+
 0:06.52
 0:00.09
 0:00.67
 0:00.22
 0:01.20
 0:00.00
 0:00.00
 0:00.00
 0:00.00
 0:00.00
 0:00.00
 0:00.00
 0:00.00
 0:00.57
 0:00.00
 0:00.00
 0:00.00
 0:00.00
```

El “ORDEN” es el numero de comando o proceso:

```
ORDEN
gnome-s+
account+
gnome-t+
top
systemd
kthreadd
rcu_gp
rcu_par+
slub_fl+
netns
kworker+
kworker+
kworker+
mm_perc+
rcu_tas+
rcu_tas+
rcu_tas+
```

El siguiente comando que vamos a utilizar es “**htop**”, este nos muestra una interfaz mas amigable que el comando anterior:

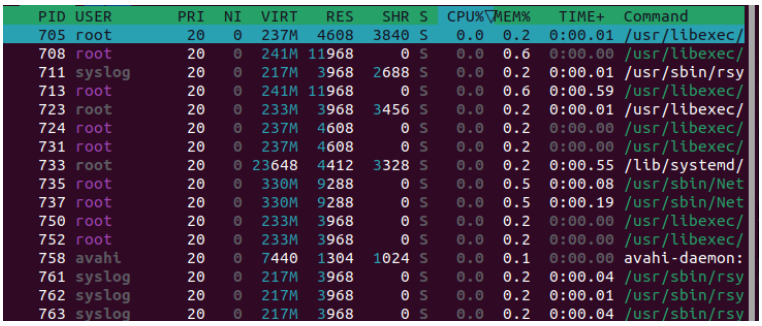


Tenemos información sobre la “CPU”, la “MEM” y el “SWP”:

La siguiente imagen muestra el número de procesos o tareas, la carga promedio del sistema y el tiempo en el que el sistema ha estado encendido:

“MAIN” es la vista principal que se está mostrando, e “I/O” es una pestaña que ve la entrada y salida de los procesos:

Si nos dirigimos a la ventana de “MAIN” logramos apreciar una similitud con el comando anterior:

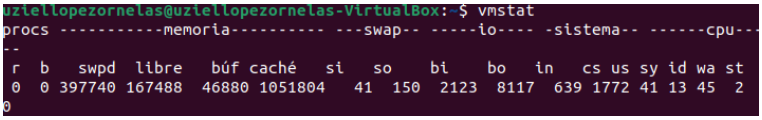


En la parte inferior tenemos una serie de acciones que podemos ejecutar dependiendo de la tecla o función que se seleccione:

- **F1**: Ayuda.
- **F2**: Configuración del comando.
- **F3**: Buscar un proceso.
- **F4**: Filtrar la lista de procesos.
- **F5**: Ver los procesos en un formato de árbol.
- **F6**: Ordenar los procesos por una columna especifica.
- **F7**: Disminuir el valor “nice” de un proceso seleccionado.
- **F8**: Aumentar el valor “nice” de un proceso seleccionado.
- **F9**: Terminar un proceso seleccionado.

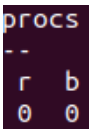
- **F10**: Salir del comando.

A continuación, se escribirá el comando de "**vmstat**":



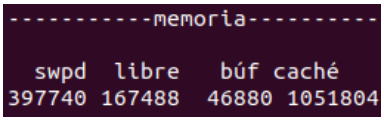
En este apartado tenemos la sección de "PROCS" procesos, aquí están los siguientes apartados:

- **r**: Numero de procesos en ejecución.
- **b**: Numero de procesos en estado de espera interrumpible.



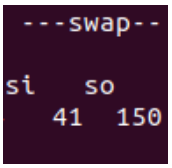
En el siguiente tenemos "MEMORIA" como su nombre lo indica es el uso de la memoria RAM:

- **swpd**: cantidad de memoria virtual utilizada.
- **libre**: Cantidad de memoria RAM libre en KB.
- **búf**: cantidad de memoria RAM utilizada como búfer en KB.
- **caché**: Cantidad de memoria RAM utilizada como caché en KB.



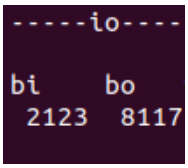
El "SWAP" es la información sobre la actividad de la memoria de intercambio:

- **si**: KB de memoria intercambiados desde el disco por segundo.
- **so**: KB de memoria intercambiados hacia el disco por segundo.



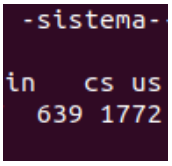
El “IO” es la información de la entrada y salida de bloques:

- **bi**: Bloques recibidos de un dispositivo de bloque por segundo.
- **bo**: Bloques enviados a un dispositivo de bloque por segundo.



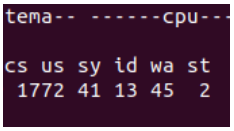
El “SISTEMA” es la información sobre la actividad del sistema:

- **in**: número de interrupciones por segundo, incluyendo el reloj.
- **cs**: Numero de cambios de contexto por segundo.

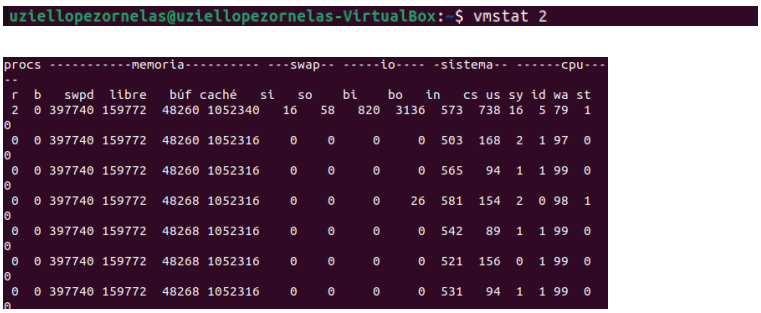


La “CPU” es el porcentaje de tiempo que la CPU está utilizando por diferentes tipos de operaciones:

- **us**: Tiempo de usuario.
- **sy**: Tiempo de sistema.
- **id**: Tiempo inactivo.
- **wa**: Tiempo en espera.
- **st**: Tiempo “robado”.



El comando que también podemos utilizar es el de “**vmstat 2**” que nos mostrara estadísticas cada dos segundos hasta que nosotros deseemos detener este proceso:



Otro comando es “**vmstat –s**” que muestra un resumen de las estadísticas desde el inicio del sistema:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ vmstat -s
```

```
     2006700 K memoria total
      746284 K memoria usada
      516912 K memoria activa
      964580 K memoria inactiva
      159772 K memoria libre
       48344 K memoria de búfer
     1052300 K caché de intercambio
     2744316 K intercambio total
      397740 K intercambio usado
     2346576 K intercambio libre
       49880 tics de CPU de usuario no-«nice»
       87261 tics de CPU del usuario «nice»
       41858 tics de CPU del sistema
      705683 tics de CPU de inactividad
        7190 tics de CPU de espera E/S
           0 tics de CPU de IRQ
        1009 tics de CPU de softirq
           0 tics de CPU robados
     7143968 páginas en entrada
    27322401 páginas en salida
       34403 páginas intercambiadas
       34403 páginas intercambiadas
      125852 páginas cambiadas
     5114416 interrupciones
     6448846 cambios de contexto de CPU
1750360212 tiempo de arranque
      132050 bifurcaciones
```

El siguiente comando que se utilizara es “**vmstat –d**” que muestra las estadísticas de uso de disco:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ vmstat -d
```

```
disco- ------------lecturas------------ ------------escrituras----------- -----I
O------
       total combinado sectores      ms  total combinado sectores      ms    cur
    seg
loop0     15        0      42       1      0        0       0       0      0      0
loop1     85        0    1396      30      0        0       0       0      0      0
loop2    280        0    5754      25      0        0       0       0      0      0
loop3    158        0    6360      38      0        0       0       0      0      0
loop4    656        0   26800     248      0        0       0       0      0      1
loop5    110        0    4510      38      0        0       0       0      0      0
loop6    110        0    4264      49      0        0       0       0      0      0
loop7    104        0    4372      50      0        0       0       0      0      0
sr0        0        0       0       0      0        0       0       0      0      0
sda   167100    49708 13806546  106455 240042 395297 54646234  661054      0    34
6
loop8   3317        0   28636     105      0        0       0       0      0      0
loop9   3689        0  135438    1130      0        0       0       0      0      3
loop10    85        0    1414      23      0        0       0       0      0      0
loop11   100        0    1446      27      0        0       0       0      0      0
loop12  1348        0  101670     825      0        0       0       0      0      4
```

El siguiente es “**vmstat –p sda1**” que nos muestra las estadísticas de la partición “sda1”.

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ vmstat -p sda1
```

```
sda1          lecturas  sectores leídos  escrituras  escrituras solicitadas
                   213             15160           0                       0
```

Pasaremos al comando “**free**” que nos muestra la información de la memoria RAM y la memoria de intercambio (swap) en un sistema:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ free
```

```
               total        usado       libre  compartido   búf/caché   disponible
Mem:         2006700       660740       238164       10180     1107796     1148104
Inter:       2744316       358572      2385744
```

En este apartado se nos muestra el total que se ha utilizado:

```
   total
 2006700
 2744316
```

El “usado”, cantidad de memoria actualmente en uso:

```
  usado
 660740
 358572
```

La memoria “Libre”, es la cantidad de memoria que está completamente libre.

```
  libre
 238164
2385744
```

La memoria “compartido” es la memoria que está siendo compartida por varios procesos:

```
compartido
     10180
```

La memoria “búf/caché” es la cantidad de memoria utilizada en los búferes del kernel y caché de las páginas:

```
 búf/caché
   1107796
```

La memoria “disponible”, es la memoria que está realmente disponible para nuevas aplicaciones sin que estas estén comprometidas al “swapping”:

```
 disponible
   1148104
```

El siguiente comando que probaremos será el de “**tlp**” que optimiza la gestión adecuada de tareas y procesos con el fin de brindar el correcto funcionamiento del sistema sin sobrecalentarse:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ tlp

Usage: tlp start|true|bat|false|ac|usb|bayoff|chargeonce|discharge|setcharge|fullch
arge|recalibrate|diskid
```

La leyenda que tenemos en este momento es la siguiente:

- **tpl start**: inicia “tpl” y aplica su configuración de ahorro de energía.

```
Usage: tlp start
```

- **true/false**: se refieren a comandos booleanos que se usarían con otras opciones, que tal vez funcionaria con activar o desactivar una característica especifica.

```
|true|bat|false|
```

- **bat**: aplica el perfil de ahorro de energía diseñado para cuando el portátil está funcionando con batería.

```
|bat|
```

- **ac**: aplica el perfil de rendimiento/ahorro de energía diseñado para cuando el portátil esté conectado a una corriente alterna.

```
|ac|
```

- **bayoff**: apaga la bahía de unidad (por ejemplo, una bahía de DVD o un segundo disco duro).

```
|bayoff|
```

- **chargeonce**: carga la batería una sola vez, posiblemente hasta un cierto umbral para después detener la carga.

```
|chargeonce|
```

- **discharge**: descarga la batería.

```
|discharge|
```

- **setcharge**: establece los umbrales de carga de la batería.

```
|setcharge|
```

- **fullcharge**: carga la batería hasta su capacidad máxima.

```
|fullch
```

- **recalibrate**: inicia el proceso de calibración de la batería.

```
|recalibrate|
```

- **disksid**: gestión de energía de los HDD o SSD.

```
|diskid
```

El comando “**lshw”** muestra la información detallada del hardware del sistema:

```
port:376 ioport:d000(size=16)
          *-cdrom
               descripción: DVD reader
               producto: CD-ROM
               fabricante: VBOX
               id físico: 0.0.0
               información del bus: scsi@1:0.0.0
               nombre lógico: /dev/cdrom
               nombre lógico: /dev/sr0
               versión: 1.0
               capacidades: removable audio dvd
               configuración: ansiversion=5 status=nodisc
       *-display
            descripción: VGA compatible controller
            producto: SVGA II Adapter
            fabricante: VMware
            id físico: 2
            información del bus: pci@0000:00:02.0
            nombre lógico: /dev/fb0
            versión: 00
            anchura: 32 bits
            reloj: 33MHz
            capacidades: vga_controller bus_master rom fb
            configuración: depth=32 driver=vmwgfx latency=64 resolution=1280,80
```

```
0
            recursos: irq:18 ioport:d010(size=16) memoria:e0000000-e0ffffff mem
oria:f0000000-f01fffff memoria:c0000-dffff
       *-network
            descripción: Ethernet interface
            producto: 82540EM Gigabit Ethernet Controller
            fabricante: Intel Corporation
            id físico: 3
            información del bus: pci@0000:00:03.0
            nombre lógico: enp0s3
            versión: 02
            serie: 08:00:27:f6:32:12
            tamaño: 1Gbit/s
            capacidad: 1Gbit/s
            anchura: 32 bits
            reloj: 66MHz
            capacidades: bus_master cap_list ethernet physical tp 10bt 10bt-fd
100bt 100bt-fd 1000bt-fd autonegotiation
            configuración: autonegotiation=on broadcast=yes driver=e1000 driver
version=6.8.0-60-generic duplex=full ip=192.168.1.23 latency=64 link=yes mingnt=
255 multicast=yes port=twisted pair speed=1Gbit/s
            recursos: irq:19 memoria:f0200000-f021ffff ioport:d020(size=8)
       *-generic
            descripción: System peripheral
```

```
            producto: VirtualBox mouse integration
            fabricante: InnoTek Systemberatung GmbH
            id físico: 4
            información del bus: pci@0000:00:04.0
            nombre lógico: input7
            nombre lógico: /dev/input/event6
            nombre lógico: /dev/input/js1
            nombre lógico: /dev/input/mouse2
            versión: 00
            anchura: 32 bits
            reloj: 33MHz
            capacidades: pci
            configuración: driver=vboxguest latency=0
            recursos: irq:20 ioport:d040(size=32) memoria:f0400000-f07fffff mem
oria:f0800000-f0803fff
       *-multimedia
            descripción: Multimedia audio controller
            producto: 82801AA AC'97 Audio Controller
            fabricante: Intel Corporation
            id físico: 5
            información del bus: pci@0000:00:05.0
            nombre lógico: card0
            nombre lógico: /dev/snd/controlC0
            nombre lógico: /dev/snd/pcmC0D0c
```

```
           nombre lógico: /dev/snd/pcmC0D0p
           nombre lógico: /dev/snd/pcmC0D1c
           versión: 01
           anchura: 32 bits
           reloj: 33MHz
           capacidades: bus_master
           configuración: driver=snd_intel8x0 latency=64
           recursos: irq:21 ioport:d100(size=256) ioport:d200(size=64)
      *-usb:0
           descripción: USB controller
           producto: KeyLargo/Intrepid USB
           fabricante: Apple Inc.
           id físico: 6
           información del bus: pci@0000:00:06.0
           versión: 00
           anchura: 32 bits
           reloj: 33MHz
           capacidades: ohci bus_master cap_list
           configuración: driver=ohci-pci latency=64
           recursos: irq:22 memoria:f0804000-f0804fff
      *-bridge
           descripción: Bridge
           producto: 82371AB/EB/MB PIIX4 ACPI
           fabricante: Intel Corporation
```

```
           descripción: Bridge
           producto: 82371AB/EB/MB PIIX4 ACPI
           fabricante: Intel Corporation
           id físico: 7
           información del bus: pci@0000:00:07.0
           versión: 08
           anchura: 32 bits
           reloj: 33MHz
           capacidades: bridge
           configuración: driver=piix4_smbus latency=0
           recursos: irq:9
      *-usb:1
           descripción: USB controller
           producto: 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
           fabricante: Intel Corporation
           id físico: b
           información del bus: pci@0000:00:0b.0
           versión: 00
           anchura: 32 bits
           reloj: 33MHz
           capacidades: ehci bus_master cap_list
           configuración: driver=ehci-pci latency=64
           recursos: irq:19 memoria:f0805000-f0805fff
      *-sata
```

```
           descripción: SATA controller
           producto: 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode]
           fabricante: Intel Corporation
           id físico: d
           información del bus: pci@0000:00:0d.0
           versión: 02
           anchura: 32 bits
           reloj: 33MHz
           capacidades: sata ahci_1.0 bus_master cap_list
           configuración: driver=ahci latency=64
           recursos: irq:21 ioport:d240(size=8) ioport:d248(size=4) ioport:d25
0(size=8) ioport:d258(size=4) ioport:d260(size=16) memoria:f0806000-f0807fff
  *-input:0
       producto: Power Button
       id físico: 1
       nombre lógico: input0
       nombre lógico: /dev/input/event0
       capacidades: platform
  *-input:1
       producto: Sleep Button
       id físico: 2
       nombre lógico: input1
       nombre lógico: /dev/input/event1
       capacidades: platform
```

```
 *-input:1
      producto: Sleep Button
      id físico: 2
      nombre lógico: input1
      nombre lógico: /dev/input/event1
      capacidades: platform
 *-input:2
      producto: AT Translated Set 2 keyboard
      id físico: 3
      nombre lógico: input2
      nombre lógico: /dev/input/event2
      nombre lógico: input2::capslock
      nombre lógico: input2::numlock
      nombre lógico: input2::scrolllock
      capacidades: i8042
 *-input:3
      producto: Video Bus
      id físico: 4
      nombre lógico: input4
      nombre lógico: /dev/input/event3
      capacidades: platform
 *-input:4
      producto: ImExPS/2 Generic Explorer Mouse
      id físico: 5
```

```
 *-input:3
      producto: Video Bus
      id físico: 4
      nombre lógico: input4
      nombre lógico: /dev/input/event3
      capacidades: platform
 *-input:4
      producto: ImExPS/2 Generic Explorer Mouse
      id físico: 5
      nombre lógico: input5
      nombre lógico: /dev/input/event4
      nombre lógico: /dev/input/mouse0
      capacidades: i8042
 *-input:5
      producto: VirtualBox USB Tablet
      id físico: 6
      nombre lógico: input6
      nombre lógico: /dev/input/event5
      nombre lógico: /dev/input/js0
      nombre lógico: /dev/input/mouse1
      capacidades: usb
AVISO: la salida puede ser incompleta o imprecisa, debería ejecutar este program
a como superusuario.
```

Monitoreo de la red

El comando “sudo **tcpdump**” permite capturar el tráfico de una interfaz especifica:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo tcpdump
```

En la imagen muestra lo que son los diferentes tipos de red que se detectaron a la hora de realizar dicho proceso:

```
[sudo] contraseña para uziellopezornelas:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:02:29.899991 IP 192.168.1.17.57621 > 192.168.1.255.57621: UDP, length 40
23:02:29.967818 IP 192.168.1.23.43176 > 192.168.1.1.domain: 39697+ PTR? 255.1.16
8.192.in-addr.arpa. (44)
23:02:29.969781 IP 192.168.1.1.domain > 192.168.1.23.43176: 39697* 1/0/0 PTR 192
.168.1.255. (71)
23:02:29.970246 IP 192.168.1.23.40854 > 192.168.1.1.domain: 39612+ PTR? 17.1.168
.192.in-addr.arpa. (43)
23:02:29.972134 IP 192.168.1.1.domain > 192.168.1.23.40854: 39612* 1/0/0 PTR 192
.168.1.17. (69)
23:02:30.070091 IP 192.168.1.23.55753 > 192.168.1.1.domain: 42812+ PTR? 1.1.168.
192.in-addr.arpa. (42)
23:02:30.071804 IP 192.168.1.1.domain > 192.168.1.23.55753: 42812* 1/0/0 PTR 192
.168.1.1. (67)
23:02:30.072394 IP 192.168.1.23.41637 > 192.168.1.1.domain: 20792+ PTR? 23.1.168
.192.in-addr.arpa. (43)
23:02:30.074138 IP 192.168.1.1.domain > 192.168.1.23.41637: 20792* 1/0/0 PTR 192
.168.1.23. (69)
23:02:34.978469 ARP, Request who-has 192.168.1.23 tell 192.168.1.1, length 46
23:02:34.978488 ARP, Reply 192.168.1.23 is-at 08:00:27:f6:32:12 (oui Unknown), l
ength 28
```

Seguiremos con el siguiente comando "**netstat**" que nos muestra la muestra la conexión de redes que tengan una conexión activa:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ netstat
```

Muestra las conexiones activas al igual que los sockets activos en el dominio de UNIX:

```
Conexiones activas de Internet (servidores w/o)
Proto  Recib Enviad Dirección local         Dirección remota       Estado
udp        0      0 192.168.1.23:bootpc     192.168.1.1:bootps      ESTABLECIDO
Sockets activos de dominio UNIX (servidores w/o)
Proto RefCnt Flags       Type       State         I-Node   Ruta
unix  3      [ ]         FLUJO      CONECTADO     13146
unix  3      [ ]         FLUJO      CONECTADO     11366
unix  3      [ ]         FLUJO      CONECTADO     11270    /run/user/1000/bus
unix  3      [ ]         FLUJO      CONECTADO     12362    @/tmp/.ICE-unix/1679
unix  3      [ ]         FLUJO      CONECTADO     8029     /run/dbus/system_bus_
socket
unix  3      [ ]         FLUJO      CONECTADO     12075
unix  3      [ ]         FLUJO      CONECTADO     6781     /run/dbus/system_bus_
socket
unix  3      [ ]         FLUJO      CONECTADO     12835
unix  3      [ ]         DGRAM      CONECTADO     4435
unix  3      [ ]         FLUJO      CONECTADO     26140
unix  3      [ ]         FLUJO      CONECTADO     13679    /run/user/1000/pipewi
re-0
unix  3      [ ]         FLUJO      CONECTADO     13060    /run/user/1000/pulse/
native
unix  3      [ ]         FLUJO      CONECTADO     11429    /run/user/1000/bus
unix  3      [ ]         FLUJO      CONECTADO     12483    /run/user/1000/bus
```

El siguiente en la lista es "**iftop**", con este comando nos permite monitorear al uso de ancho de banda de una red especifica:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ iftop
```

Muestra la información IPv4 asignada a la interfaz, la dirección IPv6, la dirección MAC así como el error de permisos:

```
interface: enp0s3
IP address is: 192.168.1.23
IPv6 address is: 2806:261:49a:cc3:34b7:7aef:6567:691
MAC address is: 08:00:27:f6:32:12
pcap_open_live(enp0s3): enp0s3: You don't have permission to capture on that dev
ice (socket: Operación no permitida)
```

El comando "**ss**" es ideal para enlistar todos los sockets abiertos:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ ss
```

```
Netid State  Recv-Q Send-Q                                    Local Address:
Port                        Peer Address:Port            Process
u_str ESTAB  0      0                                                     *
13146                               * 13147
u_str ESTAB  0      0                                                     *
11366                               * 11383
u_str ESTAB  0      0                                        /run/user/1000/bus
11270                               * 11269
u_str ESTAB  0      0                                       @/tmp/.ICE-unix/1679
12362                               * 12361
u_str ESTAB  0      0                                 /run/dbus/system_bus_socket
8029                                * 8028
u_str ESTAB  0      0                                                     *
12075                               * 12080
u_str ESTAB  0      0                                 /run/dbus/system_bus_socket
6781                                * 6558
u_str ESTAB  0      0                                                     *
12835                               * 12836
u_dgr ESTAB  0      0                                                     *
4435                                * 4434
u_str ESTAB  0      0                                                     *
26140                               * 26142
u_str ESTAB  0      0                                   /run/user/1000/pipewire-0
```

El comando “**ifconfig**” es ideal para mostrar y configurar la conexión de la interfaz de red:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ ifconfig
```

En esta acción podemos observar la dirección IP de nuestra “VirtualBox” y demás información que puede ser relevante para la gestión y configuración de la red:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.23  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::d17a:967a:2fd8:d7e6  prefixlen 64  scopeid 0x20<link>
        inet6 2806:261:49a:cc3:34b7:7aef:6567:691  prefixlen 64  scopeid 0x0<glo
bal>
        inet6 2806:261:49a:cc3:15c3:e5df:b257:6126  prefixlen 64  scopeid 0x0<gl
obal>
        ether 08:00:27:f6:32:12  txqueuelen 1000  (Ethernet)
        RX packets 11239  bytes 14508677 (14.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2323  bytes 236991 (236.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Bucle local)
        RX packets 213  bytes 20896 (20.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 213  bytes 20896 (20.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

“**traceroute**” es un comando que nos ayuda a rastrear la ruta que siguen los paquetes de datos entre un dispositivo local y un destino en la red:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ traceroute
```

```
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [
 -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,
HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --fwmark=num ] hos
t [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
                              Specify a network interface to operate with
```

Conclusión

Aprender los comandos de Linux sin duda es una gran ayuda para entender cómo funciona el sistema operativo, conocer las ventajas y desventajas que nos pueden ofrecer, así como la manera en que lo podemos utilizar a nuestro favor para lograr gestionar de manera óptima todos los procesos, naturalmente los sistemas que más se ven en los dispositivos son “Windows” y “MAC” ya que son los más comercializados, en una tienda departamental estos serán los reyes de las ventas gracias a que casi todo el mundo lo conoce pero no por ello Linux deja de ser importante puesto que su libertad y seguridad hacen de este sistema algo atractivo para aquellos que quieren experimentar en algo distinto y que también desean que la gran

comunidad realice los aportes para actualizaciones o algún código importante en el que ellos logren adentrarse mucho más en este mundo de la informática, donde las cosas que en ocasiones son más interesantes son aquellas que no son tan conocidas por el público en general

Link de GitHub

https://github.com/UZLOP984/Sistemas-Operativos-II.git

Referencias

Stamp, M. (2025, 16 enero). *Top 100 comandos Linux que debes conocer*. Guías Para Sitios Web, Tips & Conocimiento. https://www.dreamhost.com/blog/es/comandos-linux-que-debes-conocer/

Los 7 principales comandos de rendimiento de Linux para administradores de sistemas: Site24x7. (s. f.). Site24x7. https://www.site24x7.com/es/learn/linux/top-commands-for-sysadmins.html

Cómo monitorear el tráfico de red en Linux: Site24x7. (s. f.). Site24x7. https://www.site24x7.com/es/learn/linux/traffic-monitor.html