

**UNIVERSIDAD DE LAS FUERZAS
ARMADAS-ESPE SEDE SANTO DOMINGO**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

PERIODO : 202450

ASIGNATURA : Sistemas Operativos

TEMA : Laboratorio 1_U2

NOMBRES : Juan David Jiménez Romero

NIVEL-PARALELO : 15310

DOCENTE : Ing. Javier Cevallos. MSc.

FECHA DE ENTREGA : 04/07/2024

SANTO DOMINGO - ECUADOR

2023

Contenido

1.Introducción.....	3
Sistemas de Objetivos	3
Objetivo general del trabajo:	3
2. Desarrollo.....	4
3. Conclusiones	16
4. Recomendaciones	16
5. Bibliografía/ Referencias	17
6.Bibliografía	17
7. Anexos	18

Introducción

En el presente informe de laboratorio se revisará los pasos que se realizaron para la elaboración del laboratorio que se realizó el día 04/07/2024, el laboratorio consiste en: La comparación de entre las líneas de comando para revisar los procesos en Windows y Linux (Debian) que nos permitirá revisar de mejor manera cuantos recursos ocupan cada uno, además de realizar programación en archivos .sh en el terminal de Linux.

Un terminal es un dispositivo de hardware, ya sea de naturaleza electromecánica o electrónica, que se puede usar tanto para ingresar como para transcribir información. Esas tareas se pueden llevar a cabo bien desde un ordenador o echando mano de un sistema informático. (Desarrollo, 2022)

Sistemas de Objetivos

Objetivo General:

Objetivo general del trabajo:

La revisión de los comandos en windows y linux para revisar los procesos que se están ejecutando y como detenerlos o iniciarlos además de ver que tanto recurso estos usan.

Objetivos Específicos:

1.1.1. Objetivo específico 1

Realizar funciones dentro del terminal de Debian que realice actividades con el solo hecho de ejecutarlos.

1.1.2. Objetivo específico 2

Realizar la comparación entre los comandos para revisar los procesos dentro de Windows PowerShell y Terminal.

2. Desarrollo

Ejercicios:

- a. Contar el número de procesos que se están ejecutando: (Get-Process).count:



```
Administrador: Windows PowerShell
PS C:\Windows\system32> (ps).count
15
PS C:\Windows\system32>
```

Este comando “(ps).count” muestra el valor numérico de los procesos que se están ejecutando en el momento que se envíe el comando.

- b. Ver los programas que se arrancan cuando un usuario inicia sesión: Get-WmiObject -Class win32_startupCommand:



```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-WmiObject -Class win32_startupCommand

Command                                                                 User
-----
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  NT AUTHORITY\...
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  NT AUTHORITY\...
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  LAB04-PC25\po...
"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start LAB-DCCO-SS\ESPE
"C:\Users\ESPE.LAB-DCCO-SS\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background LAB-DCCO-SS\ESPE
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  LAB-DCCO-SS\ad
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  NT SERVICE\SQ...
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  NT SERVICE\ks...
C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup                  NT SERVICE\MS...
%windir%\system32\SecurityHealthSystray.exe                          Public
"C:\Program Files\Realtek\Audio\HDA\RtkNGUI64.exe" -s                 Public

PS C:\Windows\system32>
```

Con el comando “Get-WmiObject -Class win32_startupCommand” este comando sirve para mostrar todos los procesos que se ejecutan cuando inicia un usuario además de darnos también cual usuario está usando el proceso.

- c. Mostrar el tiempo transcurrido en la ejecución de un proceso: Get-Process | Select-Object Id,TotalProcessorTime:

```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-Process | select-Object Id,TotalProcessorTime

Id TotalProcessorTime
--
9160 00:00:06.2343750
16580 00:00:00.2343750
17212 00:00:00.0312500
12612 00:00:02.9687500
7880 09:35:54.3281250
11072 00:00:00.5156250
12484 00:00:23.4062500
12132 00:00:01.7187500
6276 00:00:00.1562500
712 00:00:36.8750000
1200 00:01:05.3593750
3116 00:00:00.1406250
3604 00:00:38.2343750
5960 00:00:00.3125000
8828 00:00:10
10928 00:00:01.3750000
11404 00:00:00.0468750
12768 00:00:02.7968750
14908 00:00:27.7031250
16836 00:00:00.1093750
6072 00:00:00.0625000
6080 00:00:00.0468750
7256 00:00:00.0156250
7264 00:00:00.0156250
3300 00:00:00.2500000
5908 00:00:00.1562500
6088 00:00:00.2187500
6096 00:00:00.2812500
17824 00:00:13.7187500
5180 00:00:00.2187500
```

```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-Process | select-Object Id,TotalProcessorTime,Name

Id TotalProcessorTime Name
--
9160 00:00:06.2343750 AggregatorHost
16580 00:00:00.2343750 ApplicationFrameHost
17212 00:00:00.0312500 armsvc
12612 00:00:03.2656250 audiodg
7880 09:35:55.8437500 avp
11072 00:00:00.5156250 avp
12484 00:00:23.4062500 avpsus
12132 00:00:01.7343750 avpui
6276 00:00:00.1718750 backgroundTaskHost
16788 00:00:00.0937500 backgroundTaskHost
712 00:00:36.9218750 chrome
1200 00:01:06.1093750 chrome
3116 00:00:00.1406250 chrome
3604 00:00:38.2656250 chrome
5960 00:00:00.3125000 chrome
8828 00:00:10.0156250 chrome
10928 00:00:01.3750000 chrome
11404 00:00:00.0468750 chrome
12768 00:00:02.7968750 chrome
14908 00:00:27.7031250 chrome
16836 00:00:00.1093750 chrome
6072 00:00:00.0625000 cmd
6080 00:00:00.0468750 cmd
7256 00:00:00.0156250 cmd
7264 00:00:00.0156250 cmd
3300 00:00:00.2500000 conhost
5908 00:00:00.1562500 conhost
6088 00:00:00.2187500 conhost
6096 00:00:00.2812500 conhost
```

Este comando “Get-Process | select-Object Id,TotalProcessorTime” nos muestra todo el tiempo que se ha estado ejecutando dichoso proceso que se muestra solamente su id, para saber su nombre solo basta añadir después del “select-Object” Name.

- d. Mostrar los procesos cuya zona de memoria para trabajar es mayor a 50MB: @(Get-Process).where{\$_ .WorkingSet -gt 50MB}:

```
Administrador: Windows PowerShell
PS C:\Windows\system32> @(Get-Process).where{$_.WorkingSet -gt 50MB}

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
4471 263 337476 108908 34.556,36 7880 0 avp
699 26 102356 94044 37,13 712 14 chrome
1674 66 69408 158088 67,31 1200 14 chrome
693 26 105240 109400 38,27 3604 14 chrome
353 25 35276 72956 2,84 12768 14 chrome
325 25 56472 83068 28,16 14908 14 chrome
1244 45 58960 87364 193,45 13144 14 dwm
2377 86 66244 139236 47,22 3308 14 explorer
443 27 2276904 80348 5,39 7308 0 java
953 139 42248 55128 343,89 5624 0 klnagent
0 0 1120 241176 237,48 2904 0 Memory Compression
1145 41 40824 85828 2,70 8808 14 msedge
857 50 54280 94496 4,00 17780 14 OneDrive
777 42 81136 97932 20,92 17240 14 powershell
0 20 10896 80004 6,88 132 0 Registry
2088 135 210984 115020 27,30 10952 14 SearchApp
965 79 51336 54088 135,88 8132 0 SearchIndexer
736 32 21956 58096 3,66 15244 14 ShellExperienceHost
753 78 428624 89312 4.660,47 8188 0 sqlservr
583 30 26608 66752 3,05 14720 14 StartMenuExperienceHost
1252 44 3317972 2069488 417,56 3472 14 VirtualBoxVM
1183 69 170808 197412 318,39 17700 14 WINWORD
```

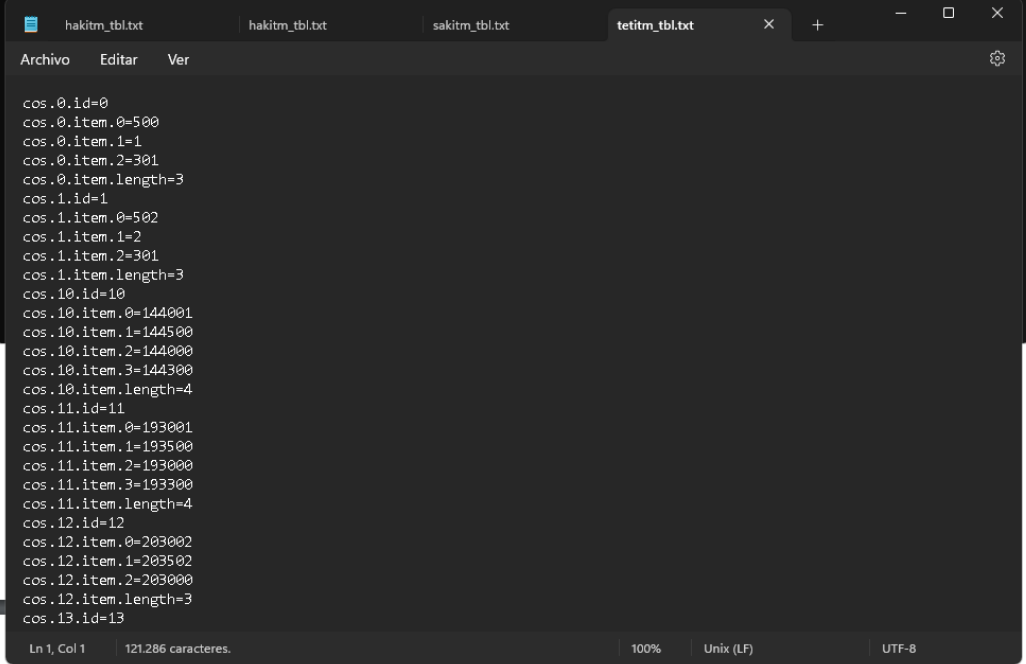
El comando “`@(Get-Process).where{$_.WorkingSet -gt 50MB}`” muestra únicamente los procesos que ocupen una cantidad mayor a 50MB memoria útil para ver cuales ocupan mucho recursos de RAM.

e. Iniciar el programa notepad: `Start-Process notepad`:

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

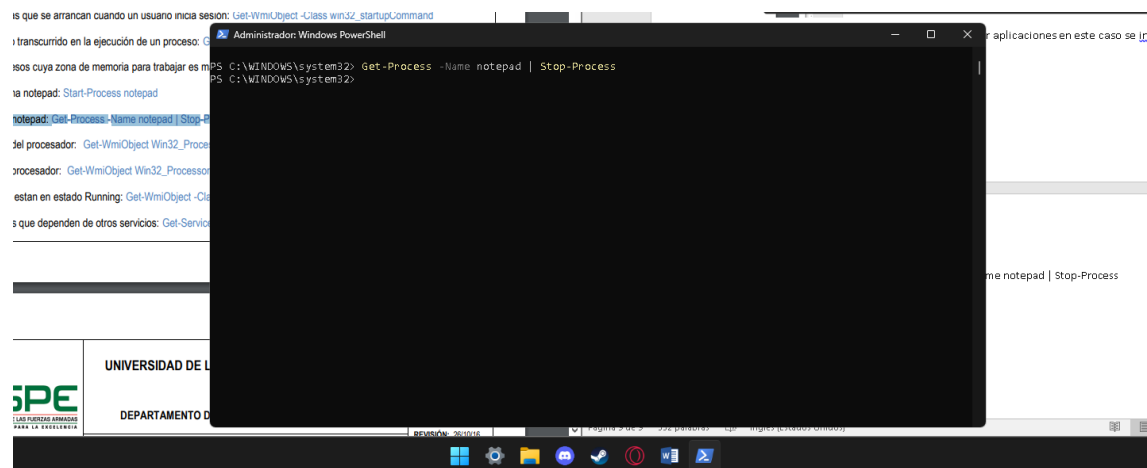
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Start-Process notepad
PS C:\WINDOWS\system32>
```



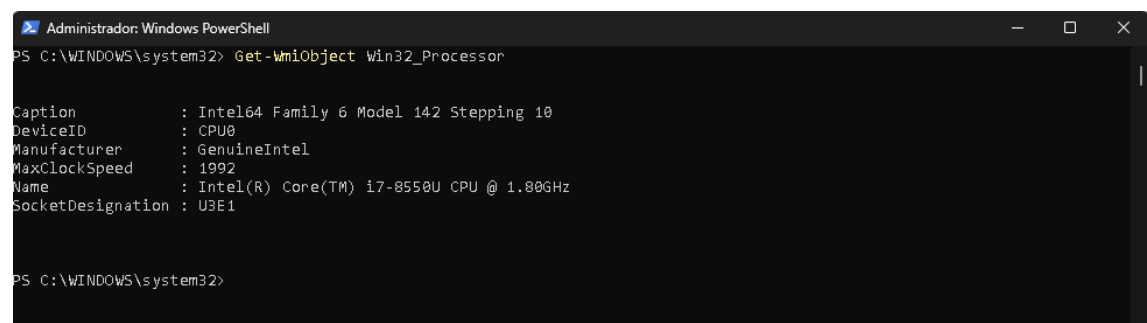
El comando “Start-Process” sirve para iniciar aplicaciones en este caso se inicio la aplicación de notepad o blog de notas.

f. Parar el proceso notepad: `Get-Process -Name notepad | Stop-Process`



El commando “Get-Process –Name notepad | Stop-Process” realiza la function de cerrar procesos en este caso al anteriormente iniciado notepad.

g. Ver información del procesador: `Get-WmiObject Win32_Processor`:



El Comando “Get-WmiObject Win32_Processor” este nos muestra la información del procesador del equipo.

h. Ver la carga del procesador: `Get-WmiObject Win32_Processor | Select-Object LoadPercentage`:

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> Get-WmiObject Win32_Processor | Select-Object LoadPercentage

LoadPercentage
-----
2

PS C:\WINDOWS\system32>
```

El comando “Get-WmiObject Win32_Processor | Select-Object LoadPercentage” se encarga de mostrar en consola el porcentaje de carga en el procesador.

- i. Ver servicios que estan en estado Running: Get-WmiObject -Class Win32_Service | Where-Object State -EQ 'Running':

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> Get-WmiObject -Class Win32_Service | Where-Object State -EQ Running

ExitCode : 0
Name      : Appinfo
ProcessId : 1220
StartMode : Manual
State     : Running
Status    : OK

ExitCode : 0
Name      : AppXSvc
ProcessId : 13656
StartMode : Manual
State     : Running
Status    : OK

ExitCode : 0
Name      : AudioEndpointBuilder
ProcessId : 3476
StartMode : Auto
State     : Running
Status    : OK

ExitCode : 0
Name      : Audiosrv
ProcessId : 3628
StartMode : Auto
State     : Running
Status    : OK
```

El commando “Get-WmiObject -Class Win32_Service | Where-Object State -EQ Running” sirve para mostrar en consola todos los servicios que estén corriendo en ese momento.

- j. Listar los servicios que dependen de otros servicios: Get-Service | select Name,ServicesDependedOn


```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> Get-Service | select Name,ServicesDependedOn

Name                                     ServicesDependedOn
----
AarSvc_3870f1f                          {}
AJRouter                               {}
ALG                                    {}
AppIDSvc                               {RpcSs, CryptSvc, AppID}
Appinfo                                {RpcSs, ProfSvc}
AppMgmt                                {}
AppReadiness                           {}
AppVClient                             {AppvVfs, RpcSS, AppvStrm, netprofm}
AppXSvc                                {rpcss, staterepository}
AssignedAccessManagerSvc               {}
AudioEndpointBuilder                   {}
Audiosrv                               {AudioEndpointBuilder, RpcSs}
autotimesvc                            {rpcss}
AxInstSV                               {rpcss}
BcastDVRUserService_3870f1f            {}
BDESVC                                 {}
BEService                              {}
BFE                                    {RpcSs}
BITS                                   {RpcSs}
BluetoothUserService_3870f1f           {}
BrokerInfrastructure                   {DcomLaunch, RpcSs, RpcEptMapper}
BTAGService                            {rpcss}
BthAvctpSvc                            {rpcss}
bthserv                                {}
camsvc                                 {}
CaptureService_3870f1f                 {}
```

El comando “Get-Service | select Name,ServicesDependedOn” sirve para mostrar en consola todos los servicios que dependen de otros servicios.

Ejercicios en Linux (debian):

14. Crear un script con nombre fecha.sh, que imprima:

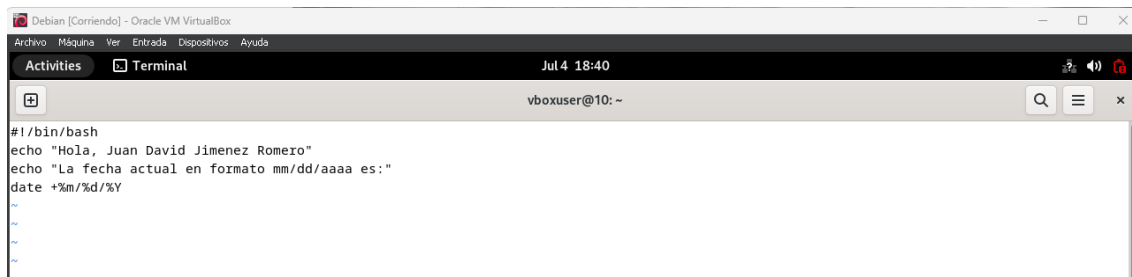
```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 18:33
vboxuser@10: ~
#!/bin/bash
echo "Hola, Juan David Jimenez Romero"
echo "la fecha actual es: "
date
```

El código dentro del archivo fecha.sh

```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 18:35
vboxuser@10: ~
root@10:~/scripts# vi fecha.sh
root@10:~/scripts# bash fecha.sh
Hola, Juan David Jimenez Romero
la fecha actual es:
Thu Jul 4 06:34:59 PM -05 2024
root@10:~/scripts#
```

Y la ejecución del archivo.

15. Crear un script con nombre fecha1.sh, que imprima:



```
#!/bin/bash
echo "Hola, Juan David Jimenez Romero"
echo "La fecha actual en formato mm/dd/aaaa es:"
date +%m/%d/%Y
~
~
~
```

Líneas de código dentro del archivo fecha1.sh



```
root@10:~/scripts# vi fecha1.sh
root@10:~/scripts# bash fecha1.sh
Hola, Juan David Jimenez Romero
La fecha actual en formato mm/dd/aaaa es:
07/04/2024
root@10:~/scripts#
```

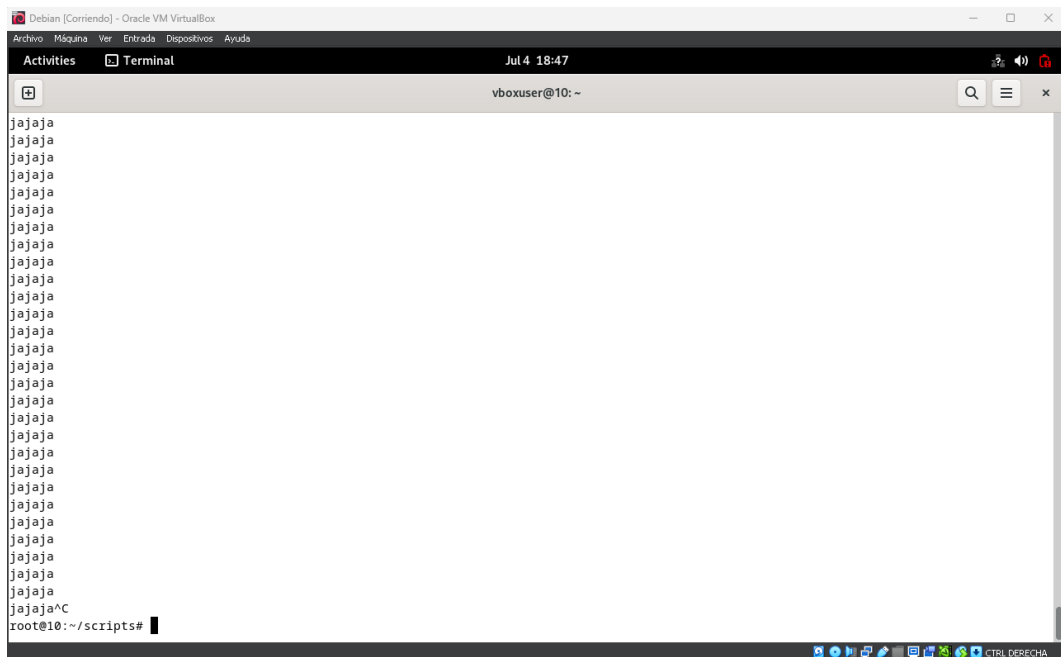
Ejecutando el archivo fecha1.sh

16. Crear un script con fecha viruschiste.sh que imprima:



```
#!/bin/bash
echo "Hola, Juan David Jimenez Romero"
echo "esto me parece:"
while true
do
echo "jajaja"
done
~
```

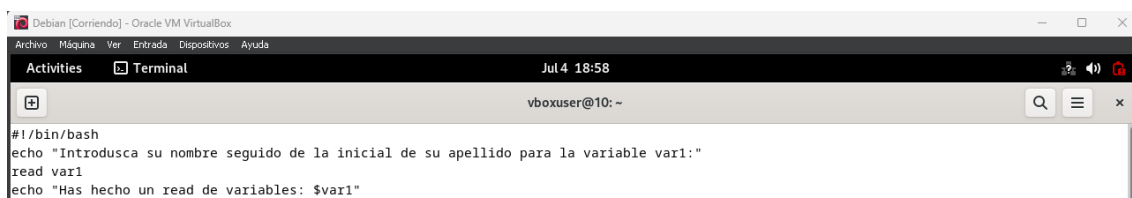
Código para realizar la función de escribir en pantalla jajaja infinitamente



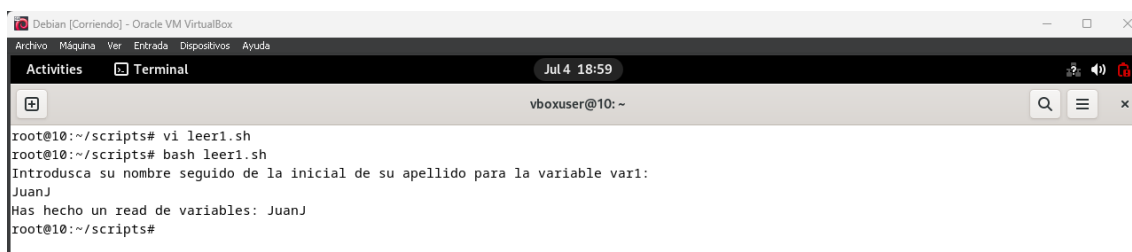
Ejecución

del archivo “viruschiste.sh”

17. Muestre el valor que el usuario ingrese por teclado (ingrese su nombre seguido de la inicial de su apellido), nombre al archivo como leer.sh



Código para escribir nombre en archivo leer1.sh

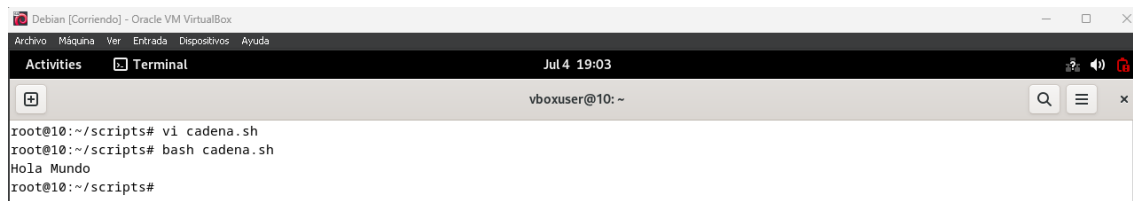


Función realizada por el comando leer1.sh

18. Unir dos cadenas de texto y mostrar en pantalla



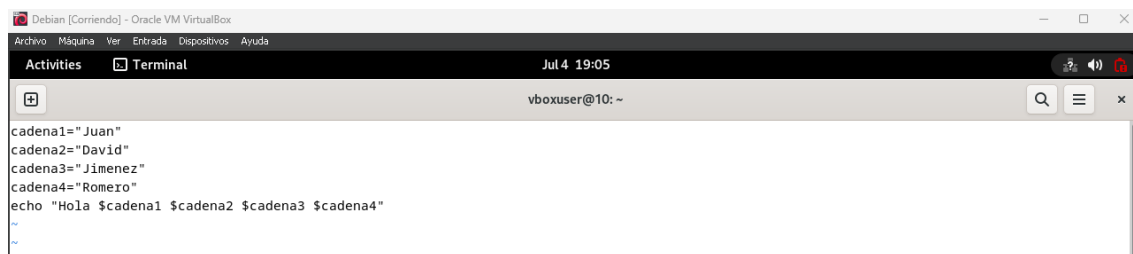
Código para realizar la actividad 18



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 19:03
vboxuser@10: ~
root@10:~/scripts# vi cadena.sh
root@10:~/scripts# bash cadena.sh
Hola Mundo
root@10:~/scripts#
```

Función al ejecutar el programa `cadena.sh`

19. Unir 4 cadenas de texto, cadena 1= su 1er nombre, cadena 2=su 2do nombre, cadena 3= su 1er apellido, cadena 4= su 2do apellido, llamar al archivo como `cadenanombre.sh`



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 19:05
vboxuser@10: ~
cadena1="Juan"
cadena2="David"
cadena3="Jimenez"
cadena4="Romero"
echo "Hola $cadena1 $cadena2 $cadena3 $cadena4"
~
~
```

Código dentro del archivo `cadenanombre.sh`



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 19:07
vboxuser@10: ~
root@10:~/scripts# vi cadenanombre.sh
root@10:~/scripts# bash cadenanombre.sh
Hola Juan David Jimenez Romero
root@10:~/scripts#
```

Ejecución del archivo `cadenanombre.sh`

20. Operaciones matemáticas: Para sumar 2 números procedemos como sigue, guardar un nuevo script con el nombre `suma.sh`



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 19:09
vboxuser@10: ~
#!/bin/bash
var1=$(( 5 + 5 ))
echo "Hola Juan David Jimenez Romero"
echo "la suma de 5+5 es:" $var1
~
~
```

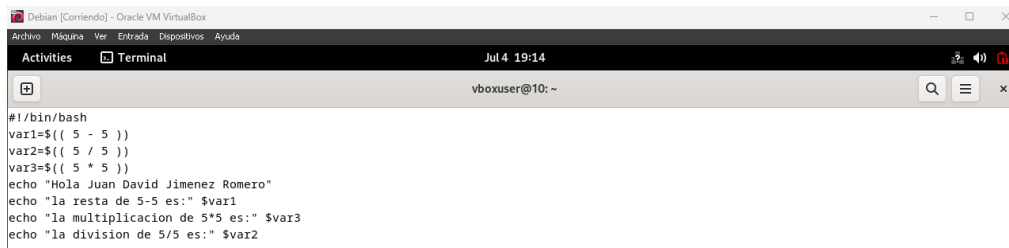
Código dentro del archivo `suma.sh`



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal Jul 4 19:10
vboxuser@10: ~
root@10:~/scripts# vi suma.sh
root@10:~/scripts# bash suma.sh
Hola Juan David Jimenez Romero
la suma de 5+5 es: 10
root@10:~/scripts#
```

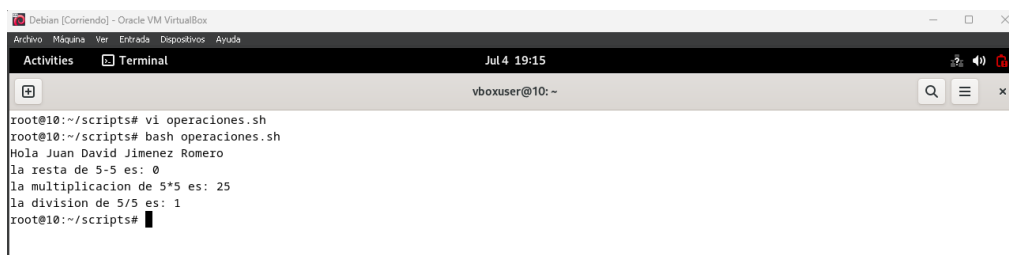
Ejecución del archivo suma.sh

21. Repita el código del paso 20 pero para resta, multiplicación y división



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal Jul 4 19:14
vboxuser@10: ~
#!/bin/bash
var1=$(( 5 - 5 ))
var2=$(( 5 / 5 ))
var3=$(( 5 * 5 ))
echo "Hola Juan David Jimenez Romero"
echo "la resta de 5-5 es:" $var1
echo "la multiplicacion de 5*5 es:" $var3
echo "la division de 5/5 es:" $var2
```

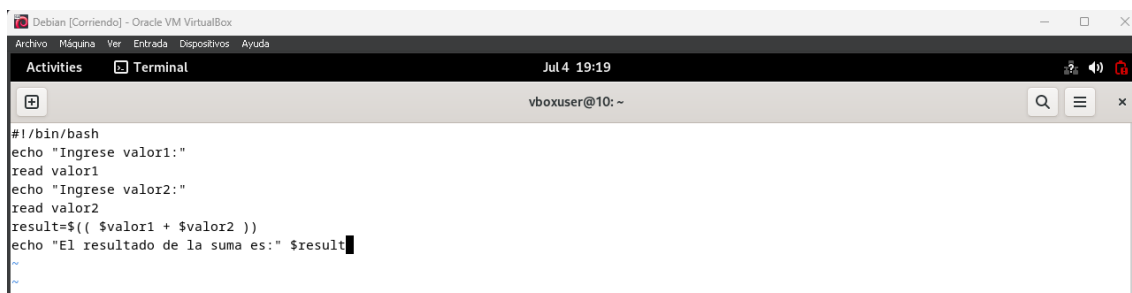
Código del archivo operaciones.sh



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal Jul 4 19:15
vboxuser@10: ~
root@10:~/scripts# vi operaciones.sh
root@10:~/scripts# bash operaciones.sh
Hola Juan David Jimenez Romero
la resta de 5-5 es: 0
la multiplicacion de 5*5 es: 25
la division de 5/5 es: 1
root@10:~/scripts#
```

Función del archivo operaciones.sh

22. Programar un script para hacer las 4 operaciones con valores que ingrese el usuario. Un script para cada operación.



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal Jul 4 19:19
vboxuser@10: ~
#!/bin/bash
echo "Ingrese valor1:"
read valor1
echo "Ingrese valor2:"
read valor2
result=$(( $valor1 + $valor2 ))
echo "El resultado de la suma es:" $result
```

Operación suma.sh



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  Jul 4 19:20
vboxuser@10: ~
root@10:~/scripts# vi suma
root@10:~/scripts# bash suma
Ingrese valor1:
3
Ingrese valor2:
8
El resultado de la suma es: 11
root@10:~/scripts#
```

Resultado suma.sh



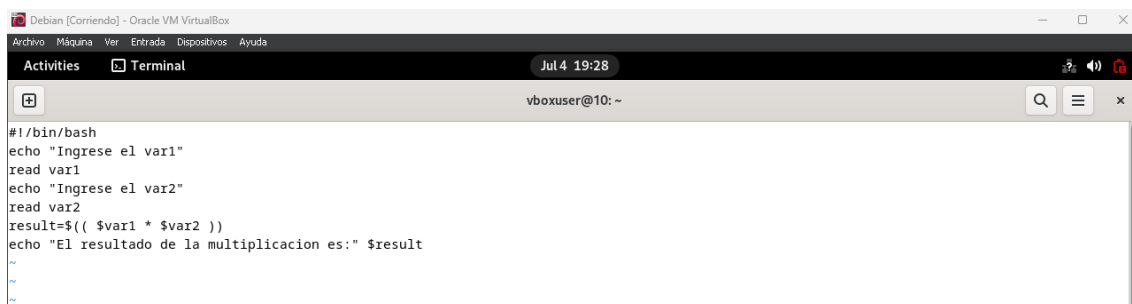
```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  Jul 4 19:23
vboxuser@10: ~
#!/bin/bash
echo "ingrese el var1:"
read var1
echo "ingrese el var2:"
read var2
result=$(( $var1 + $var2 ))
echo "La suma del var1 y var2 es:" $result
```

Código de resta.sh



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  Jul 4 19:25
vboxuser@10: ~
root@10:~/scripts# vi resta.sh
root@10:~/scripts# bash resta.sh
ingrese el var1:
69
ingrese el var2:
33
La resta del var1 y var2 es: 36
root@10:~/scripts#
```

Resultado al momento al ejecutar resta.sh



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  Jul 4 19:28
vboxuser@10: ~
#!/bin/bash
echo "Ingrese el var1"
read var1
echo "Ingrese el var2"
read var2
result=$(( $var1 * $var2 ))
echo "El resultado de la multiplicacion es:" $result
```

Código de multiplicacion.sh



The screenshot shows a terminal window titled "Debian [Corriendo] - Oracle VM VirtualBox". The menu bar includes "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The window has tabs for "Activities" and "Terminal". The terminal shows the following commands and output:

```
root@10:~/scripts# vi multiplicacion.sh
root@10:~/scripts# bash multiplicacion.sh
Ingrese el var1
23
Ingrese el var2
3
El resultado de la multiplicacion es: 69
root@10:~/scripts#
```

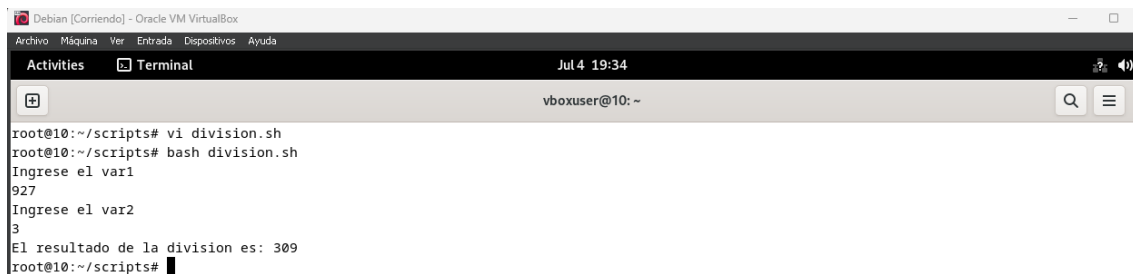
Resultado de la ejecución del archivo multiplicación.sh



The screenshot shows a terminal window titled "Debian [Corriendo] - Oracle VM VirtualBox". The menu bar includes "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The window has tabs for "Activities" and "Terminal". The terminal shows the following code for a script:

```
#!/bin/bash
echo "Ingrese el var1"
read var1
echo "Ingrese el var2"
read var2
result=$(( $var1 / $var2 ))
echo "El resultado de la division es:" $result
```

El código del programa divison.sh



The screenshot shows a terminal window titled "Debian [Corriendo] - Oracle VM VirtualBox". The menu bar includes "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The window has tabs for "Activities" and "Terminal". The terminal shows the following commands and output:

```
root@10:~/scripts# vi division.sh
root@10:~/scripts# bash division.sh
Ingrese el var1
927
Ingrese el var2
3
El resultado de la division es: 309
root@10:~/scripts#
```

Resultado de la ejecución del programa división.sh

3. Conclusiones

- El conocer los comandos para visualizar los procesos en el PowerShell y en el Terminal nos sirve para saber cuáles programas están realizando un gran uso de recursos del equipo además de permitirnos ver cuál es el origen del proceso y verificar si este es de origen seguro, entre otros.
- El desarrollo de programación mediante “vi” en el terminal de Debian es una forma de ejecutar funciones dentro de la terminal mediante archivos .sh que nos facilitaría mucho al tener que realizar funciones que necesiten varios códigos.
- Se concluye que ambos terminales son ideales para su SO, pero se destaca que debían tener más accesibilidad en los comandos y en la cantidad de funciones que se pueden hacer en este debido a que se le pueden instalar complementos que ayuden al momento de realizar codificación.

4. Recomendaciones

- Se recomienda conocer el significado de los comandos y aprender si existen variaciones más cortas para comandos que podrían ser bastantes largos y difíciles de aprenderse.
- Al momento de realizar codificación con el comando “vi” se recomienda revisar o ver cómo funciona el mismo debido a que este no se controla de manera tradicional, sino que se necesita saber ciertas combinaciones de botones para realizar varias funciones.
- *Se recomienda aprender más sobre programación y comandos en estas aplicaciones* como lo serían el terminal o Windows powershell debido a que son bastantes útiles aprenderse en caso de que se necesite revisar el equipo con posibles fallos.

5. Bibliografía/ Referencias

Bibliografía

Desarrollo, P. (2022, febrero 16). *5 tipos de terminales informáticas*. GSC MADRID GRUPO DE SOFTWARE Y CONSULTING. <https://gscmadrid.com/tipos-de-terminales-informaticas/>

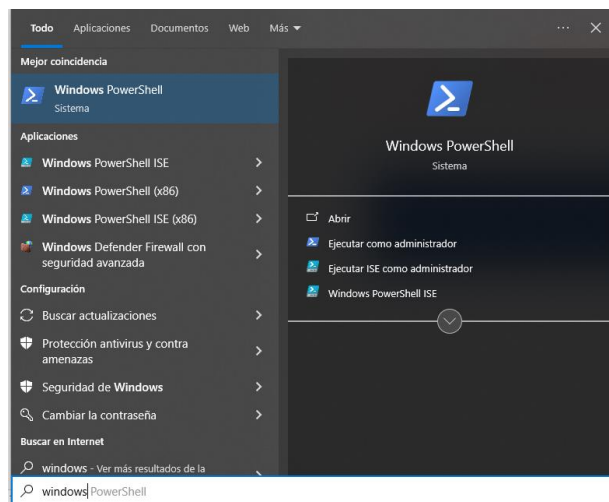
Euroinnova Business School. (2021, julio 28). *Conoce cuáles son las carreras relacionadas con computación y sus salidas laborales*. Euroinnova Business School. <https://www.euroinnova.ec/sistema-operativo>

Informática Básica: Qué es un sistema operativo. (s/f). Gcfglobal.org. Recuperado el 25 de mayo de 2024, de <https://edu.gcfglobal.org/es/informatica-basica/que-es-un-sistema-operativo/1/>

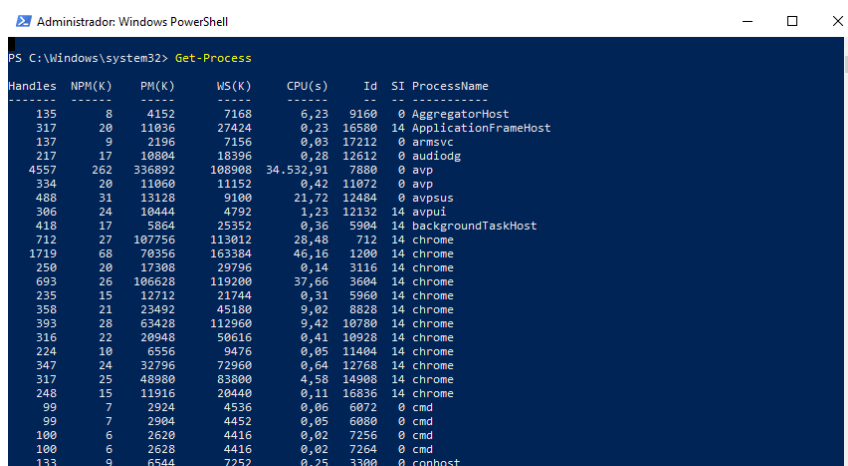
Tipos de sistemas operativos y sus características. (2023, junio 14). UCMA. <https://www.universitatcarlemany.com/actualidad/blog/tipos-de-sistemas-operativos/>

6. Anexos

Iniciamos Windows powershell mediante el buscador de Windows:



Se nos mostrara la siguiente ventana que nos dice que ya se inició el Windows PowerShell:



El comando “Get-Process” sirve para mostrar todos los procesos que se están ejecutando ya sea en primer o segundo plano con su id, consumo de CPU y otros datos importantes.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> ps
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
135	8	4152	7168	6,23	9160	0	AggregatorHost
317	19	11008	27412	0,23	16580	14	ApplicationFrameHost
137	9	2196	7156	0,03	17212	0	armsvc
232	17	10500	18732	0,52	12612	0	audiodg
4554	261	337708	116188	34,533,28	7880	0	avp
336	20	11072	11168	0,42	11072	0	avp
487	31	13060	9088	21,72	12484	0	avpsus
306	24	10444	4792	1,30	12132	14	avpui
482	22	5648	23136	0,16	8320	14	backgroundTaskHost
712	27	112816	118228	28,94	712	14	chrome
1717	68	70340	163404	46,91	1200	14	chrome
250	20	17308	29792	0,14	3116	14	chrome
693	26	106628	119032	37,66	3604	14	chrome
235	15	12712	21744	0,31	5960	14	chrome
355	21	23436	45156	9,06	8828	14	chrome
393	28	63428	112976	9,42	10780	14	chrome
316	22	20944	50360	0,41	10928	14	chrome
224	10	6556	9476	0,05	11404	14	chrome
347	24	32936	72972	0,73	12768	14	chrome
317	25	49264	84484	5,19	14908	14	chrome
248	15	11916	20440	0,11	16836	14	chrome
99	7	2924	4536	0,06	6072	0	cmd
99	7	2904	4452	0,05	6080	0	cmd
100	6	2620	4416	0,02	7256	0	cmd
100	6	2628	4416	0,02	7264	0	cmd
133	9	6544	7252	0,25	3300	0	conhost
139	9	6548	7228	0,16	5908	0	conhost
141	9	6564	7244	0,22	6088	0	conhost
141	9	6540	7236	0,28	6096	0	conhost
258	15	4404	18376	1,09	17824	14	conhost

Con el comando “ps” realiza exactamente lo mismo que el comando “Get-Process” solo que lo resume a 2 letras siendo P la letra inicial de Process y la S de la letra final.

```
Administrador: Windows PowerShell
```

```
ExecutablePath      : C:\Windows\system32\ApplicationFrameHost.exe
ExecutionState       :
Handle               : 16580
HandleCount          : 317
InstallDate          :
KernelModeTime       : 1875000
MaximumWorkingSetSize : 1380
MinimumWorkingSetSize : 200
Name                 : ApplicationFrameHost.exe
OSCreationClassName  : Win32_OperatingSystem
OSName               : Microsoft Windows 10 Pro[C:\Windows\Device\Harddisk0\Partition3
OtherOperationCount   : 229
OtherTransferCount    : 1132
PageFaults           : 8949
PageFileUsage         : 11008
ParentProcessId       : 656
PeakPageFileUsage     : 11488
PeakVirtualSize       : 2203570745344
PeakWorkingSetSize    : 29020
Priority              : 8
PrivatePageCount      : 11272192
ProcessId             : 16580
QuotaNonPagedPoolUsage : 20
QuotaPagedPoolUsage   : 324
QuotaPeakNonPagedPoolUsage : 22
QuotaPeakPagedPoolUsage : 342
ReadOperationCount    : 9
ReadTransferCount     : 14977
SessionId             : 14
Status                :
TerminationDate       :
ThreadCount           : 2
UserModeTime          : 468750
VirtualSize           : 2203556528128
```

El comando “Get-WmiObject -Class win32_process” lo que realiza este comando es mostrar todos los procesos que se encuentran en la dirección de Win32 y este nos dice gran parte de los datos y detalles que lo componen como el uso de RAM, la prioridad, entre otros.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-Process|select cpu,id,name|sort cpu -Descending

CPU      Id  Name
----  --  ---
34538,21875  7880  avp
4544,671875  8188  sqlservr
1296,5625    4     System
340,9375    5624  klntagent
255,21875   3472  VirtualBoxVM
230,78125   2904  Memory Compression
203,78125   17700  WINWORD
172,90625   4552  mongod
137,234375  1064  svchost
132,921875  8132  SearchIndexer
116,46875   13144  dwm
106,359375  948   lsass
92,625      2760  svchost
90,640625   656   svchost
63,34375    1672  svchost
54,484375   2284  svchost
52          3176  svchost
50,734375   1200  chrome
47,5625     1900  svchost
46,84375    2224  svchost
46,828125   5664  SecurityHealthService
40,65625    4256  svchost
37,671875   3604  chrome
31,859375   3308  explorer
30,390625   4148  svchost
30,125      712   chrome
27,09375    10952  SearchApp
27,046875   2212  svchost
24,84375    904   services
23,796875   1744  svchost
```

Con el comando “Get-Process|select cpu,id,name|sort cpu -Descending” sirve para mostrar los procesos que tienen un alto consumo de CPU este mostrado de forma descendente con y mostrado con la forma de CPU, id, Name.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> ps|select Name

Name
----
AggregatorHost
ApplicationFrameHost
armsvc
audiodg
avp
avp
avpsus
avpui
backgroundTaskHost
chrome
chrome
chrome
chrome
chrome
chrome
chrome
chrome
chrome
chrome
cmd
cmd
cmd
cmd
```

El comando “ps|select Name” estas nomas muestra el nombre de la lista de procesos en orden de abecedario.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> ps|select Name,Company

Name                Company
----                -
AggregatorHost
ApplicationFrameHost Microsoft Corporation
armsvc              Adobe Inc.
audiodg             Microsoft Corporation
avp
avp
avpsus
avpui
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
chrome              Google LLC
cmd                 Microsoft Corporation
cmd                 Microsoft Corporation
```

Con el comando “ps|select Name, Company” este realiza la función de mostrar en consola el nombre del proceso y la Compañía que tiene registrado a su nombre.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> (ps) | %{if($_.path){(Get-ChildItem $_.path | select VersionInfo)}}

VersionInfo
-----
File:      C:\Windows\System32\AggregatorHost.exe...
File:      C:\Windows\system32\ApplicationFrameHost.exe...
File:      C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe...
File:      C:\Windows\system32\AUDIODG.EXE...
File:      C:\Windows\system32\backgroundTaskHost.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Program Files\Google\Chrome\Application\chrome.exe...
File:      C:\Windows\SysWOW64\cmd.exe...
File:      C:\Windows\SysWOW64\cmd.exe...
File:      C:\Windows\SysWOW64\cmd.exe...
File:      C:\Windows\SysWOW64\cmd.exe...
File:      C:\Windows\system32\conhost.exe...
File:      C:\Windows\system32\conhost.exe...
File:      C:\Windows\system32\conhost.exe...
File:      C:\Windows\system32\conhost.exe...
File:      C:\Windows\system32\conhost.exe...
File:      C:\Windows\SysWOW64\cscript.exe...
File:      C:\Windows\SysWOW64\cscript.exe...
File:      C:\Windows\system32\ctfmon.exe...
File:      C:\Windows\system32\DllHost.exe...
```

El comando “(ps) | %{if(\$_.path){(Get-ChildItem \$_.path | select VersionInfo)}}” nos sirve para mostrarnos la dirección de los procesos donde se encuentra y además nos muestra la información del mismo.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-Process | select Name,Product,Description

Name                Product                Description
----                -
AggregatorHost      Microsoft® Windows® Operating System Application Frame Host
ApplicationFrameHost Acrobat Update Service Acrobat Update Service
armsvc              Sistema operativo Microsoft® Windows® Aislamiento de gráficos de dispositivo de audio de Windows
audiodg
avp
avp
avpsus
avpui
backgroundTaskHost  Microsoft® Windows® Operating System Background Task Host
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
chrome              Google Chrome          Google Chrome
cmd                 Microsoft® Windows® Operating System Windows Command Processor
cmd                 Microsoft® Windows® Operating System Windows Command Processor
cmd                 Microsoft® Windows® Operating System Windows Command Processor
cmd                 Microsoft® Windows® Operating System Windows Command Processor
conhost             Sistema operativo Microsoft® Windows® Host de ventana de consola
conhost             Sistema operativo Microsoft® Windows® Host de ventana de consola
conhost             Sistema operativo Microsoft® Windows® Host de ventana de consola
conhost             Sistema operativo Microsoft® Windows® Host de ventana de consola
conhost             Sistema operativo Microsoft® Windows® Host de ventana de consola
cscrip              Microsoft® Windows Script Host Microsoft® Console Based Script Host
```

Con el comando “Get-Process | select Name,Product,Description” este comando sirve para listar los procesos solamente los Nombres, el producto y una descripción de los que tienen alguna.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> Get-WmiObject win32_process | Sort-Object Processid | Select-Object Processid,Name,CommandLine

Processid Name                CommandLine
-----
0 System Idle Process
4 System
72 Secure System
132 Registry
276 RuntimeBroker.exe    C:\Windows\System32\RuntimeBroker.exe -Embedding
496 smss.exe
656 svchost.exe          C:\Windows\system32\svchost.exe -k DcomLaunch -p
704 csrss.exe
708 fontdrvhost.exe      "fontdrvhost.exe"
712 chrome.exe           "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=gpu-process --g...
784 wininit.exe
876 sppsvc.exe
904 services.exe
940 lsaiso.exe
948 lsass.exe
1064 svchost.exe          C:\Windows\system32\svchost.exe -k RPCSS -p
1112 svchost.exe          C:\Windows\system32\svchost.exe -k DcomLaunch -p -s LSM
1200 chrome.exe           "C:\Program Files\Google\Chrome\Application\chrome.exe"
1344 svchost.exe          C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s NcbService
1412 svchost.exe          C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p -s DevQuery...
1420 svchost.exe          C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p -s HvHost
1460 svchost.exe          C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p -s hidserv
1468 IntelCpHDGPSvc.exe   C:\Windows\System32\DriverStore\FileRepository\igdlh64.inf_amd64_9671054d3bbcb...
1556 svchost.exe          C:\Windows\system32\svchost.exe -k netsvcs -p -s ProfSvc
1656 svchost.exe          C:\Windows\system32\svchost.exe -k LocalService -p -s nsi
1672 svchost.exe          C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p -s EventLog
1676 svchost.exe          C:\Windows\system32\svchost.exe -k LocalService -s W32Time
1688 svchost.exe          C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s TimeBro...
1744 svchost.exe          C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule
```

Con el comando “Get-WmiObject win32_process | Sort-Object Processid | Select-Object Processid,Name,CommandLine” nos muestra el id del proceso, el nombre, y la ubicación de este de los procesos que se ejecutan al iniciar el equipo o el SO.

```
PS C:\Windows\system32> (ps).Where{-not $_.Responding}

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
529 37 19448 2644 0,30 4108 14 WinStore.App

PS C:\Windows\system32>
```

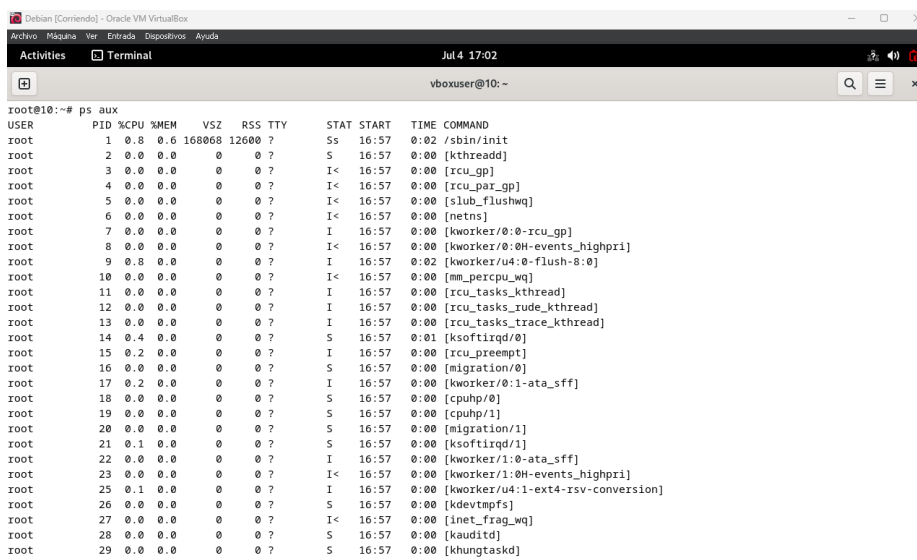
El comando “(ps).Where{-not \$_.Responding}” este muestra los procesos que no están respondiendo este caso muestra únicamente 1 que es WinStore.App que sería la tienda de Windows.

Actividad en Linux (debían):



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 17:01
vboxuser@10: ~
root@10:~# ps
  PID TTY          TIME CMD
  3000 pts/0    00:00:00 su
  3001 pts/0    00:00:00 bash
  3005 pts/0    00:00:00 ps
root@10:~#
```

El comando “ps” nos muestra la cantidad de procesos que hay ejecutándose y el tiempo que este ha estado activo.



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 17:02
vboxuser@10: ~
root@10:~# ps -aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.6 168068 12600 ?        Ss   16:57   0:02 /sbin/init
root           2  0.0  0.0      0     0 ?        S    16:57   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   16:57   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   16:57   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   16:57   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   16:57   0:00 [netns]
root           7  0.0  0.0      0     0 ?        I<   16:57   0:00 [kworker/0:0-rcu_gp]
root           8  0.0  0.0      0     0 ?        I<   16:57   0:00 [kworker/0:0H-events_highpri]
root           9  0.8  0.0      0     0 ?        I    16:57   0:02 [kworker/u4:0-flush-8:0]
root          10  0.0  0.0      0     0 ?        I<   16:57   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    16:57   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    16:57   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    16:57   0:00 [rcu_tasks_trace_kthread]
root          14  0.4  0.0      0     0 ?        S    16:57   0:01 [ksoftirqd/0]
root          15  0.2  0.0      0     0 ?        I    16:57   0:00 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    16:57   0:00 [migration/0]
root          17  0.2  0.0      0     0 ?        I    16:57   0:00 [kworker/0:1-ata_sff]
root          18  0.0  0.0      0     0 ?        S    16:57   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    16:57   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    16:57   0:00 [migration/1]
root          21  0.1  0.0      0     0 ?        S    16:57   0:00 [ksoftirqd/1]
root          22  0.0  0.0      0     0 ?        I    16:57   0:00 [kworker/1:0-ata_sff]
root          23  0.0  0.0      0     0 ?        I<   16:57   0:00 [kworker/1:0H-events_highpri]
root          25  0.1  0.0      0     0 ?        I    16:57   0:00 [kworker/u4:1-ext4-rsv-conversion]
root          26  0.0  0.0      0     0 ?        S    16:57   0:00 [kdevtmpfs]
root          27  0.0  0.0      0     0 ?        I<   16:57   0:00 [inet_frag_wq]
root          28  0.0  0.0      0     0 ?        S    16:57   0:00 [kauditd]
root          29  0.0  0.0      0     0 ?        S    16:57   0:00 [khungtaskd]
```

El comando “ps -aux” nos muestra todos los procesos que existen en el sistema en este caso se ven que hay más de los que muestra normalmente.

```
root@10:~# ps axjf
PPID    PID    PGID    SID    TTY        TPGID  STAT   UID    TIME  COMMAND
0        2      0       0      ?          -1 S    0       0:00  [kthreadd]
2        3      0       0      ?          -1 I<    0       0:00  \_ [rcu_gp]
2        4      0       0      ?          -1 I<    0       0:00  \_ [rcu_par_gp]
2        5      0       0      ?          -1 I<    0       0:00  \_ [slub_flushwq]
2        6      0       0      ?          -1 I<    0       0:00  \_ [netns]
2        8      0       0      ?          -1 I<    0       0:00  \_ [kworker/0:0H-events_highpri]
2        9      0       0      ?          -1 I     0       0:02  \_ [kworker/u4:0-flush-8:0]
2       10      0       0      ?          -1 I<    0       0:00  \_ [mm_percpu_wq]
2       11      0       0      ?          -1 I     0       0:00  \_ [rcu_tasks_kthread]
2       12      0       0      ?          -1 I     0       0:00  \_ [rcu_tasks_rude_kthread]
2       13      0       0      ?          -1 I     0       0:00  \_ [rcu_tasks_trace_kthread]
2       14      0       0      ?          -1 S     0       0:01  \_ [ksoftirqd/0]
2       15      0       0      ?          -1 I     0       0:00  \_ [rcu_preempt]
2       16      0       0      ?          -1 S     0       0:00  \_ [migration/0]
2       17      0       0      ?          -1 I     0       0:00  \_ [kworker/0:1-ata_sff]
2       18      0       0      ?          -1 S     0       0:00  \_ [cpuhp/0]
2       19      0       0      ?          -1 S     0       0:00  \_ [cpuhp/1]
2       20      0       0      ?          -1 S     0       0:00  \_ [migration/1]
2       21      0       0      ?          -1 S     0       0:00  \_ [ksoftirqd/1]
2       23      0       0      ?          -1 I<    0       0:00  \_ [kworker/1:0H-events_highpri]
2       25      0       0      ?          -1 I     0       0:00  \_ [kworker/u4:1-ext4-rsv-conversion]
2       26      0       0      ?          -1 S     0       0:00  \_ [kdevtmpfs]
2       27      0       0      ?          -1 I<    0       0:00  \_ [inet_frag_wq]
2       28      0       0      ?          -1 S     0       0:00  \_ [kauditd]
2       29      0       0      ?          -1 S     0       0:00  \_ [khungtaskd]
2       30      0       0      ?          -1 I     0       0:00  \_ [kworker/u4:2-ext4-rsv-conversion]
2       31      0       0      ?          -1 S     0       0:00  \_ [oom_reaper]
2       32      0       0      ?          -1 I<    0       0:00  \_ [writeback]
```

Con el comando “ps axjf” nos muestra un árbol jerarquico con la ruta donde se encuentra cada uno de los procesos.

```
root@10:~# ps aux | grep bash
vboxuser 2414  0.0  0.1  8004 3548 pts/0  Ss   16:59   0:00 bash
root     3001  0.0  0.2  7988 4760 pts/0   S    17:01   0:00 -bash
root     3028  0.0  0.1  6332 2152 pts/0   S+   17:04   0:00 grep bash
root@10:~#
```

comando “ps aux | grep bash” nos muestra los procesos que son exclusivamente de bash.


```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 17:05
vboxuser@10: ~

top - 17:05:33 up 7 min, 1 user, load average: 0.14, 0.64, 0.48
Tasks: 168 total, 1 running, 167 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.8 us, 1.0 sy, 0.0 ni, 97.8 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
MiB Mem : 1967.1 total, 469.4 free, 1107.6 used, 553.6 buff/cache
MiB Swap: 975.0 total, 968.2 free, 6.8 used, 859.5 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1418 vboxuser  20   0 3905208 389292 127964 S   6.6  19.3   0:39.65 gnome-shell
 2224 vboxuser  20   0 559340  51380 38884 S   1.7   2.6   0:01.91 gnome-terminal-
 246 root      20   0      0      0      0 I   0.3   0.0   0:00.79 kworker/0:3-events
3035 root      20   0 11612   4960  3068 R   0.3   0.2   0:00.01 top
   1 root      20   0 168068 12600  9156 S   0.0   0.6   0:02.12 systemd
   2 root      20   0      0      0      0 S   0.0   0.0   0:00.02 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-events_highpri
  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   0:01.24 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   0:00.02 migration/0
  17 root      20   0      0      0      0 I   0.0   0.0   0:00.77 kworker/0:1-ata_sff
  18 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
  19 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
  20 root      rt   0      0      0      0 S   0.0   0.0   0:00.24 migration/1
  21 root      20   0      0      0      0 S   0.0   0.0   0:00.49 ksoftirqd/1
```

Con el comando “top” visualizamos el panel de tareas por así decirlo de Linux en el terminal donde nos muestran los procesos ejecutándose y cuanto ocupan cada uno en memoria, entre otros.

Con el comando “top -d 5” solamente se mostrará esa interfaz durante 5 segundos eso dependiendo del número que se coloque después de la d.

```
top - 17:08:07 up 10 min, 1 user, load average: 0.01, 0.37, 0.40
Tasks: 168 total, 1 running, 167 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.5 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1967.1 total, 429.2 free, 1147.8 used, 553.7 buff/cache
MiB Swap: 975.0 total, 968.2 free, 6.8 used, 819.3 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1418 vboxuser  20   0 3905208 389320 127964 S   1.3 19.3   0:44.43 gnome-shell
   746 root       20   0   6728   4368 3404 S   0.3  0.2   0:00.06 apache2
     1 root       20   0 168068 12600  9156 S   0.0  0.6   0:02.12 systemd
     2 root       20   0     0     0     0 S   0.0  0.0   0:00.02 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-events_highpri
    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   0:01.25 ksoftirqd/0
    15 root       20   0     0     0     0 I   0.0  0.0   0:00.69 rcu_preempt
    16 root       rt   0     0     0     0 S   0.0  0.0   0:00.02 migration/0
    17 root       20   0     0     0     0 I   0.0  0.0   0:00.77 kworker/0:1-ata_sff
    18 root       20   0     0     0     0 S   0.0  0.0   0:00.00 cpuhp/0
    19 root       20   0     0     0     0 S   0.0  0.0   0:00.00 cpuhp/1
    20 root       rt   0     0     0     0 S   0.0  0.0   0:00.24 migration/1
    21 root       20   0     0     0     0 S   0.0  0.0   0:00.50 ksoftirqd/1
    23 root       0 -20     0     0     0 I   0.0  0.0   0:00.00 kworker/1:0H-events_highpri
    26 root       20   0     0     0     0 S   0.0  0.0   0:00.02 kdevtmpfs
```

Con

el comando “top -o %CPU” mostrara de forma ordenada dependiendo del consumo de la CPU en forma descendente.

```
Tasks: 168 total, 1 running, 167 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 50.0 sy, 0.0 ni, 50.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1967.1 total, 409.8 free, 1167.1 used, 553.8 buff/cache
MiB Swap: 975.0 total, 968.2 free, 6.8 used, 799.9 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1418 vboxuser  20   0 3905208 390224 127964 S   6.2 19.4   0:47.34 gnome-shell
 1229 vboxuser  20   0  19612  11428  9032 S   0.0  0.6   0:00.51 systemd
 1232 vboxuser  20   0  103644   3028     0 S   0.0  0.2   0:00.00 (sd-pam)
 1253 vboxuser  9 -11  53668  11780  7716 S   0.0  0.6   0:00.34 pipewire
 1257 vboxuser  9 -11 257204  14280  9104 S   0.0  0.7   0:00.26 wireplumber
 1259 vboxuser  9 -11  32196  9780  7324 S   0.0  0.5   0:00.06 pipewire-pulse
 1260 vboxuser  20   0  10404   5824  4016 S   0.0  0.3   0:00.89 dbus-daemon
 1265 vboxuser  20   0 239908  11064  8012 S   0.0  0.5   0:00.09 gnome-keyring-d
 1273 vboxuser  20   0 237512  11368  6336 S   0.0  0.6   0:00.09 gvfsd
 1286 vboxuser  20   0 159280  7804  5276 S   0.0  0.4   0:00.01 gdm-wayland-ses
 1292 vboxuser  20   0 297956 20112 14148 S   0.0  1.0   0:00.09 gnome-session-b
 1295 vboxuser  20   0 380372  7948  5284 S   0.0  0.4   0:00.01 gvfsd-fuse
 1377 vboxuser 39 19 631216 59764 15180 S   0.0  3.0   0:01.33 tracker-miner-f
 1379 vboxuser  20   0  88380  5112  4476 S   0.0  0.3   0:00.00 gcr-ssh-agent
 1381 vboxuser  20   0  88828  6904  4556 S   0.0  0.3   0:00.03 gnome-session-c
 1382 vboxuser  20   0   7800   5144  4496 S   0.0  0.3   0:00.03 ssh-agent
 1392 vboxuser  20   0 593688 19776 15020 S   0.0  1.0   0:00.33 gnome-session-b
 1410 vboxuser  20   0 311200  7608  6896 S   0.0  0.4   0:00.04 at-spi-bus-laun
 1421 vboxuser  20   0   9120  4540  4024 S   0.0  0.2   0:00.07 dbus-daemon
 1422 vboxuser  20   0 351532 12648  8928 S   0.0  0.6   0:00.06 gvfs-udisks2-vo
 1430 vboxuser  20   0 233352  8152  5584 S   0.0  0.4   0:00.04 gvfs-mtp-volume
 1441 vboxuser  20   0 312408  9612  6480 S   0.0  0.5   0:00.13 gvfs-afc-volume
 1446 vboxuser  20   0 233520  7948  5432 S   0.0  0.4   0:00.05 gvfs-goa-volume
```

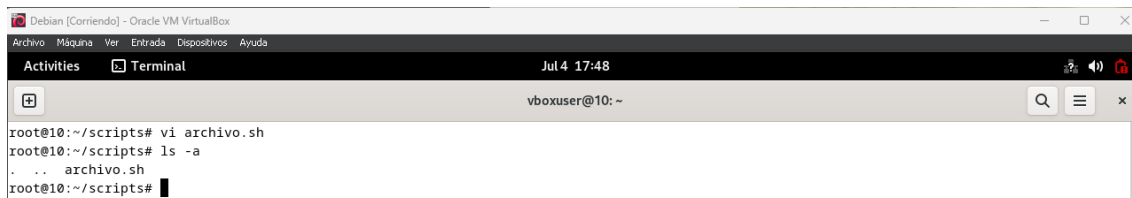
Con

el comando “top -u (NombreDelUsuario)” mostrara los procesos ejecutados únicamente por ese usuario.



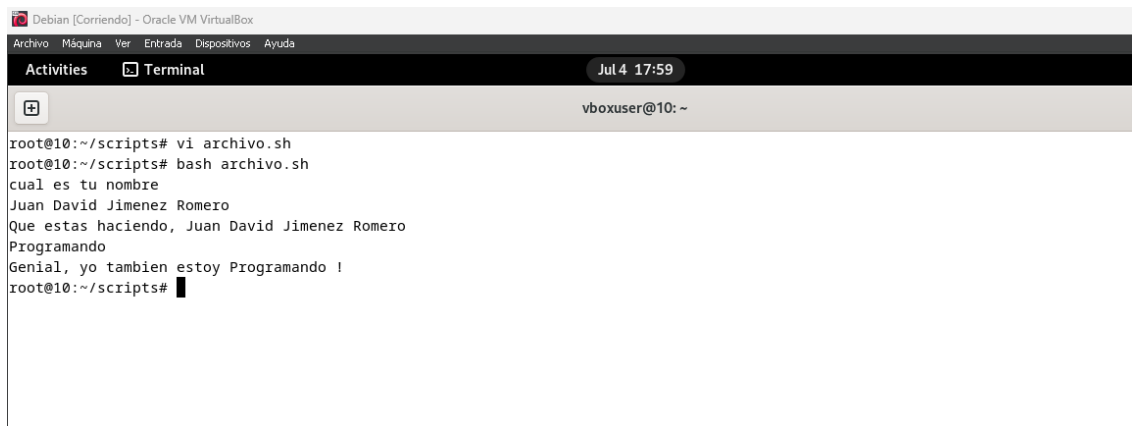
```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 17:16
vboxuser@10: ~
root@10:~/scripts# cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
root@10:~/scripts#
```

Revisamos los shells de bash con el comando “cat /etc/shells” generalmente vienen ya incluidos en la mayoría de distribuciones de Linux



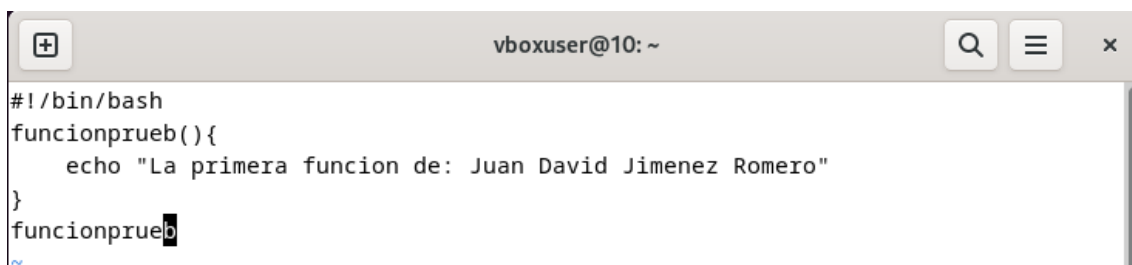
```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal
Jul 4 17:48
vboxuser@10: ~
root@10:~/scripts# vi archivo.sh
root@10:~/scripts# ls -a
.  ..  archivo.sh
root@10:~/scripts#
```

Con el comando vi “archivo.sh” dentro de la carpeta creada con el nombre de scripts con el comando “mkdir”, entramos al Shell de Linux donde ponemos dentro el comando “#!/bin/sh” que es como la librería que usa y abajo un comando en este caso coloque el de “top” y luego pulsando el botón “esc + shift + (alguna letra)” colocamos la siguiente línea “wq!” que es para guardar los archivos en “vi” y con el comando “ls -a” visualizamos que esté hay guardado.



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 17:59
vboxuser@10: ~
root@10:~/scripts# vi archivo.sh
root@10:~/scripts# bash archivo.sh
cual es tu nombre
Juan David Jimenez Romero
Que estas haciendo, Juan David Jimenez Romero
Programando
Genial, yo tambien estoy Programando !
root@10:~/scripts#
```

Resultado al ejecutar el archivo.sh con el comando bash

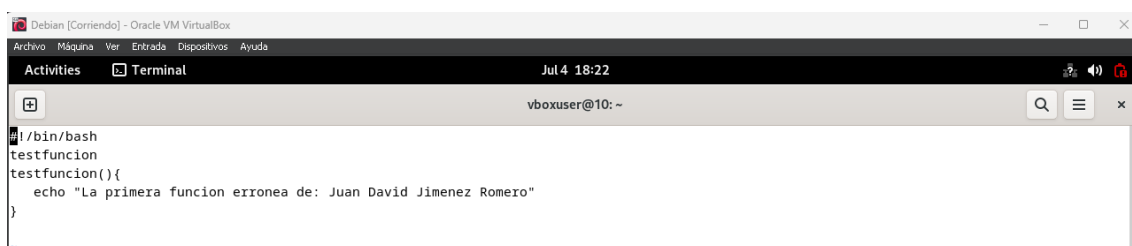


```
vboxuser@10: ~
#!/bin/bash
funcionprueb(){
    echo "La primera funcion de: Juan David Jimenez Romero"
}
funcionprueb
```



```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 18:18
vboxuser@10: ~
vboxuser@10:~$ su -
Password:
root@10:~# cd scripts/
root@10:~/scripts# vi funcion
root@10:~/scripts# vi funcion.sh
root@10:~/scripts# bash funcion.sh
La primera funcion de: Juan David Jimenez Romero
root@10:~/scripts#
```

Creamos una función dentro de otro archivo .sh nombrado como función.sh y dentro realizamos una función que muestre en pantalla un mensaje



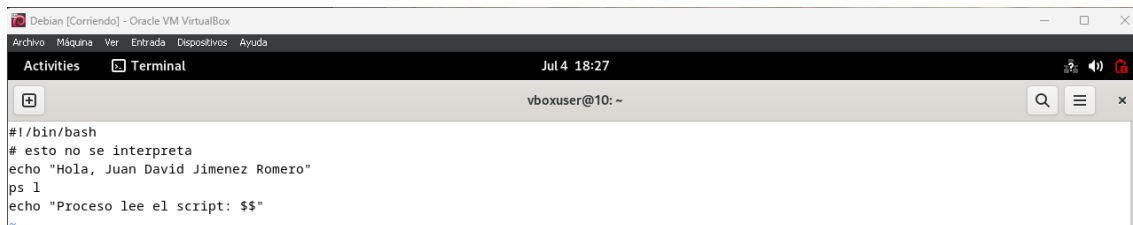
```
Debian [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Jul 4 18:22
vboxuser@10: ~
#!/bin/bash
testfuncion
testfuncion(){
    echo "La primera funcion erronea de: Juan David Jimenez Romero"
}
```

Creamos otro archivo .sh llamado funcionerror.sh y le realizamos lo mismo que el anterior solo que la llamada a la función esta primero que la función para ver que sale error.

A terminal window titled 'Debian [Corriendo] - Oracle VM VirtualBox' with a menu bar (Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda) and a status bar (Jul 4 18:23). The terminal shows a user 'vboxuser@10: ~' running commands: 'vi funcionerror.sh', 'bash funcionerror.sh', and 'testfuncion: command not found'. The error message 'funcionerror.sh: line 2: testfuncion: command not found' is displayed.

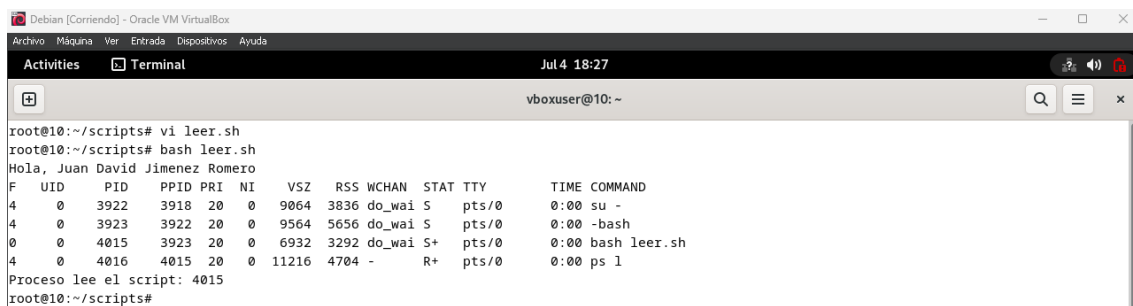
```
root@10:~/scripts# vi funcionerror.sh
root@10:~/scripts# vi funcionerror.sh
root@10:~/scripts# bash funcionerror.sh
funcionerror.sh: line 2: testfuncion: command not found
root@10:~/scripts#
```

Nos sale un error y nos dice en que línea se encuentra el error en este caso la 2 debido a que está haciendo la llamada a la función inexistente.

A terminal window titled 'Debian [Corriendo] - Oracle VM VirtualBox' with a menu bar (Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda) and a status bar (Jul 4 18:27). The terminal shows a user 'vboxuser@10: ~' running commands: '#!/bin/bash', '# esto no se interpreta', 'echo "Hola, Juan David Jimenez Romero"', 'ps 1', and 'echo "Proceso lee el script: \$\$"'. The output shows the command being executed and the process ID.

```
#!/bin/bash
# esto no se interpreta
echo "Hola, Juan David Jimenez Romero"
ps 1
echo "Proceso lee el script: $$"
```

Creamos un archivo .sh con el nombre de leer.sh y lo llenamos con el siguiente código

A terminal window titled 'Debian [Corriendo] - Oracle VM VirtualBox' with a menu bar (Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda) and a status bar (Jul 4 18:27). The terminal shows a user 'vboxuser@10: ~' running commands: 'vi leer.sh', 'bash leer.sh', and 'ps 1'. The output shows the command being executed and the process ID. Below the command output, a table of processes is displayed.

```
root@10:~/scripts# vi leer.sh
root@10:~/scripts# bash leer.sh
Hola, Juan David Jimenez Romero
ps 1
echo "Proceso lee el script: 4015"
root@10:~/scripts#
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
4	0	3922	3918	20	0	9064	3836	do_wai	S	pts/0	0:00	su -
4	0	3923	3922	20	0	9564	5656	do_wai	S	pts/0	0:00	-bash
0	0	4015	3923	20	0	6932	3292	do_wai	S+	pts/0	0:00	bash leer.sh
4	0	4016	4015	20	0	11216	4704	-	R+	pts/0	0:00	ps 1

Usando el comando “bash” con el archivo leer.sh y lo ejecutamos nos muestra lo de la imagen.