Supervisión de procesos y optimización de rendimiento en linux

1. Análisis en tiempo real del sistema.

Ejecutar y analizar las siguientes herramientas:

- Comandos top y htop: Para ver procesos activos, carga y uso de CPU/RAM.
- Comando uptime y free -m.

Identificar:

- Proceso con mayor consumo de CPU.
- Proceso con mayor uso de memoria.
- Tiempo que lleva encendido el sistema y carga promedio.

2. Gestión activa de procesos y prioridades.

- Finalizar un proceso inactivo o no esencial: kill. killall o pkill.
- Cambiar la prioridad de un proceso en ejecución: Con comando renice.
- Lanzar un proceso en segundo plano (&) y enviarlo al primer plano con fq.
- Usar nice para iniciar un proceso con prioridad baja: Por ejemplo una copia pesada con cp.

3. Monitorización y registro del uso de recursos.

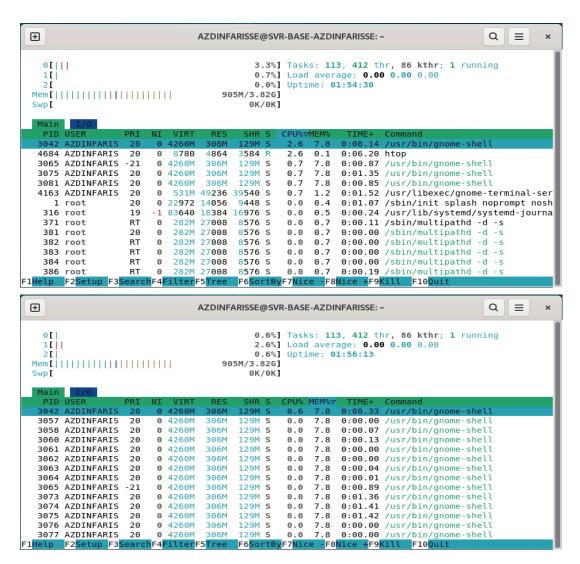
- Usar el comando vmstat y guardar su salida en un archivo /srv/logs/vmstat.log.
- Configurar una tarea en crontab que guarde el uso de recursos (top -b -n 1) cada 5 minutos en /srv/logs/top.log.
- Explorar iotop: Para monitorizar I/O de disco.

4. Simulación de sobrecarga controlada.

- Instalar el paquete stress o stress-ng.
- Ejecutar una prueba con carga simulada de: CPU, memoria o disco durante 1 minuto.
- Observar el comportamiento del sistema con htop y anotar el resultado.

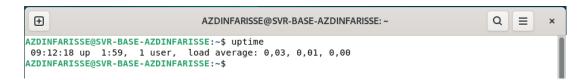
1. Análisis en tiempo real del sistema.

Utilizamos los comandos **top** y **htop** para identificar los procesos con mayor consumo de CPU y con mayor consumo de memoria.



Como se puede observar en las imágenes, el proceso que está consumiendo mayor CPU y mayor memoria es: **gnome-shell.**

Ahora, para identificar cuánto tiempo lleva encendido el sistema y su carga promedio lo haremos con el comando **uptime.**



En la imagen podemos observar que el sistema lleva encendido 1 hora y 59 minutos, mientras que el nivel de carga está prácticamente en reposo.

2. Gestión activa de procesos y prioridades.

Para finalizar un proceso inactivo o que no sea esencial lo haremos con el comando **pkill.** El proceso que utilizaremos en este caso será **firefox**.

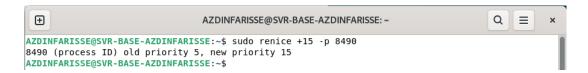
```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~

AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ pkill firefox
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

Ahora vamos a cambiar la prioridad de **firefox**. Para cambiar la prioridad tenemos que saber primero el **PID** de firefox utilizando el comando **pgrep** -**f firefox**.



Ahora que ya sabemos su PID (8490) le cambiamos la prioridad con el comando **renice**. Para ello introducimos **sudo renice +15 -p 8490**.



Lo siguiente es lanzar sleep en segundo plano con & y enviarlo al primer plano con fg.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~

AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sleep 30 &

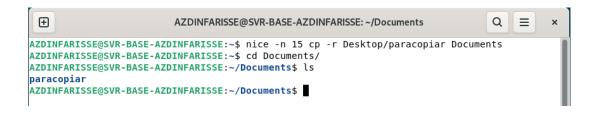
[1] 10819

AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ fg %1

sleep 30

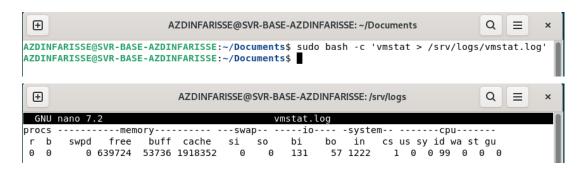
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

Ahora vamos a usar nice para iniciar el proceso **cp** con prioridad baja. Para ello he creado una carpeta llamada paracopiar, la cual será la que se utilice para lanzar el comando cp. El comando quedaría: **nice -n 15 cp -r Desktop/paracopiar documents**



3. Monitorización y registro del uso de recursos.

Para usar el comando **vmstat** y guardar su salida en un archivo /svr/logs/vmstat.log lo haremos con el comando **sudo bash -c 'vmstat > /srv/logs/vmstat.log'**



Ahora para configurar la tarea en crontab editaremos el cron con el comando crontab -e y le añadiremos al final del archivo la siguiente línea: */5 * * * * top -b -n 1 >> /srv/logs/top.log



Lo siguiente es explorar **iotop.** Lo haremos con los siguientes comandos: **sudo apt install iotop sudo iotop**

```
Q
   \oplus
                                                                                AZDINFARISSE@SVR-BASE-AZDINFARISSE: /srv/logs
                                                                                                                                                                                                                                                              =
Total DISK READ:
Current DISK READ:
TID PRIO USER
                                                                         0.00 B/s | Total DISK WRITE:
0.00 B/s | Current DISK WRIT
                                                                      0.00 B/s | Current DI
DISK READ DISK WRITE>
                                                                                                                                                COMMAND
                                                                                                            0.00 B/s init splash noprompt noshell
0.00 B/s [kthreadd]
0.00 B/s [pool_workqueue_release]
0.00 B/s [kworker/R-rcu_g]
0.00 B/s [kworker/R-rcu_p]
0.00 B/s [kworker/R-slub_]
0.00 B/s [kworker/R-netns]
0.00 B/s [kworker/R-netns]
0.00 B/s [kworker/R-mm_pe]
0.00 B/s [kworker/R-mm_pe]
0.00 B/s [rcu_tasks_kthread]
0.00 B/s [rcu_tasks_rude_kthread]
0.00 B/s [ksoftirqd/0]
0.00 B/s [ksoftirqd/0]
0.00 B/s [rcu_trasks_trace_kthread]
                                                                                                             0.00 B/s init splash noprompt noshell automatic-ubiquity
                                                                         0.00 B/s
                                                                         0.00 B/s
0.00 B/s
0.00 B/s
0.00 B/s
                     be/4 root
be/4 root
                     be/4 root
be/0 root
be/0 root
be/0 root
be/0 root
                                                                        0.00 B/S
              12 be/0
                                     root
                     be/4 root
be/4 root
be/4 root
be/4 root
              16
              17 be/4
18 rt/4
19 rt/4
20 be/4
                                                                                                                                        [rcu_preempt]
[migration/0]
[idle_inject/0]
[cpuhp/0]
                                    root
                                                                                                             0.00 B/s
0.00 B/s
                                                                         0.00 B/s
0.00 B/s
                                                                                                             0.00
                                                                                                                            B/s
                                      root
                                                                                                                           B/s
                                      root
                                                                         0.00 B/s
                                                                                                             0.00 B/s
                          any: refresh <u>q</u>: quit <u>i</u>: ionice <u>o</u>: active <u>p</u>: procs <u>r</u>: asc <u>left</u>: DISK READ <u>right</u>: COMMAND <u>home</u>: TID <u>end</u>
SOFIG TASK_DELAY_ACCT and kernel.task_delayacct sysctl not enabled in kernel, cannot determi
```

4. Simulación de sobrecarga controlada.

Lo primero es instalar el paquete stress con el comando sudo apt install stress.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv/logs$ sudo apt install stress
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
    stress
    0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 10 no actualizados.
Se necesita descargar 18,1 kB de archivos.
Se ustilizarán 52,2 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble/universe amd64 stress amd64 1.0.7-1 [18,1 kB]
Descargados 18,1 kB en 0s (46,4 kB/s)
Seleccionando el paquete stress previamente no seleccionado.
((Leyendo la base de datos ... 210846 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../stress_1.0.7-1_amd64.deb ...
Desempaquetando stress (1.0.7-1) ...
Configurando stress (1.0.7-1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Scanning processes...
Scanning kernel seems to be up-to-date.

No services need to be restarted.
```

Lo siguiente es ejecutar una prueba con carga simulada de CPU con 2 núcleos por un minuto, con el comando **stress --cpu 2 --timeout 60.**

Y mientras se realiza la prueba, en otra ventana del terminal ejecutamos el comando **htop** y podemos observar que la CPU tiene los 2 primeros núcleos al 100% de utilización.

