

PRACTICA 09

GESTIÓN DE

PROCESOS

Integrantes:

Marina Rosa Carvalho Santos

Juan Diego Sánchez

Ross Hillary Cáceres

Diana Carolina Pelaez

ÍNDICE

1.- Introducción.....	3
2.- Parte 1 - Identificación de procesos.....	3
3.- Parte 2 - Gestión de procesos.....	5
4.- Parte 3 - Monitorización avanzada.....	6

Introducción

En esta práctica, el grupo irá desarrollar habilidades claves en la gestión y monitorización de procesos dentro de sistemas operativos basados en Linux. A lo largo del ejercicio, se profundizará en el uso de diversas herramientas y comandos, tanto básicos como avanzados, que permiten identificar y gestionar los procesos en ejecución. A través de una secuencia estructurada, que incluye el uso de comandos sencillos como **ps** y **kill**, hasta técnicas más avanzadas como la modificación de la prioridad de los procesos utilizando los comandos **nice** y **renice**, además del uso de comandos de monitoreo en tiempo real con herramientas como **top** y **htop**.

Aprender a manejar estos comandos es esencial para cualquier administrador de sistemas, ya que garantiza un manejo eficiente de los recursos del sistema, optimiza el rendimiento y asegura que los procesos más importantes reciban la atención necesaria en entornos de trabajo reales.

Parte 1 - Identificación de procesos

Para empezar escribimos estos comandos dentro del archivo “prueba.sh”. Este generará un archivo que en el nombre incluirá la fecha y la hora a la que fue creado.

```
#!/bin/bash

OUTPUT_FILE=$(date +%Y-%m-%d_%H:%M:%S_PARTE1.txt)

echo "=== Listado completo de procesos ===" >> $OUTPUT_FILE
ps aux >> $OUTPUT_FILE

echo "\n=== Procesos del usuario $(whoami) ===" >> $OUTPUT_FILE
ps u -U $(whoami) >> $OUTPUT_FILE

echo "\n=== Procesos ordenados por uso de CPU ===" >> $OUTPUT_FILE
ps aux --sort=%cpu | head -n 10 >> $OUTPUT_FILE

echo "\n=== Procesos ordenados por uso de Memoria ===" >> $OUTPUT_FILE
ps aux --sort=-%mem | head -n 10 >> $OUTPUT_FILE
```

Aquí está la prueba de la existencia de este archivo.

```
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ ./prueba.sh
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ ls
2025-04-03_19:33:03_PARTE1.txt  2025-04-07_17:23:56_PARTE1.txt  prueba.sh
```

Al ser un archivo que contiene todos los procesos ordenados y sin ordenar de varias formas distintas enseñamos a continuación una pequeña prueba del contenido.

```

Abrir  ▾  [?]  2025-04-07_17:23:56_PARTE1.txt  ~/Documents/Practica_9
prueba.sh  2025-04-07_17:23:56_PARTE1.txt

=== Listado completo de procesos ===

USER      PID  %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1    0.0  0.2  23368 11776 ?        Ss   abr06    0:42 /usr/lib/systemd/systemd --system --deserialize=33
splash
root      2    0.0  0.0      0      0 ?        S    abr06    0:00 [kthreadd]
root      3    0.0  0.0      0      0 ?        S    abr06    0:00 [pool_workqueue_release]
root      4    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/R-rcu_g]
root      5    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/R-rcu_p]
root      6    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/R-slub_]
root      7    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/R-netns]
root     10    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/0:0H-events_highpri]
root     12    0.0  0.0      0      0 ?        I<   abr06    0:00 [kworker/R-mm_pe]
root     13    0.0  0.0      0      0 ?        I    abr06    0:00 [rcu_tasks_kthread]
root     14    0.0  0.0      0      0 ?        I    abr06    0:00 [rcu_tasks_rude_kthread]
root     15    0.0  0.0      0      0 ?        I    abr06    0:00 [rcu_tasks_trace_kthread]
root     16    0.0  0.0      0      0 ?        S    abr06    0:02 [ksoftirqd/0]
root     17    0.0  0.0      0      0 ?        I    abr06    0:27 [rcu_preempt]
root     18    0.0  0.0      0      0 ?        S    abr06    0:00 [migration/0]
root     19    0.0  0.0      0      0 ?        S    abr06    0:00 [idle_inject/0]
root     20    0.0  0.0      0      0 ?        S    abr06    0:00 [cpuhp/0]
root     21    0.0  0.0      0      0 ?        S    abr06    0:00 [cpuhp/1]
root     22    0.0  0.0      0      0 ?        S    abr06    0:00 [idle_inject/1]
root     23    0.0  0.0      0      0 ?        S    abr06    0:00 [migration/1]
root     24    0.0  0.0      0      0 ?        S    abr06    0:03 [ksoftirqd/1]
root     27    0.0  0.0      0      0 ?        S    abr06    0:00 [kdevtmpfs]

```

Y aquí hacemos uso de la herramienta htop.

[illegible]

Parte 2 - Gestión de procesos

Usaremos comandos como `<< sudo kill -15 PID >>` y `<< sudo kill -9 PID >>` del proceso para finalizar su ejecución en el sistema.

```
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps aux | grep apache2
root      112490  0.0  0.1  6804 4852 ?        Ss   19:52   0:00 /usr/sbin/apache2 -k start
www-data  112491  0.0  0.1 1212944 5376 ?        Sl   19:52   0:00 /usr/sbin/apache2 -k start
www-data  112493  0.0  0.1 1212944 5504 ?        Sl   19:52   0:00 /usr/sbin/apache2 -k start
www-data  112697  0.0  0.0   3616 1544 ?        Ss   19:53   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod
d_cache_disk -l 300M -n
alumno    113183  0.0  0.0   6648 2432 pts/0    S+   20:09   0:00 grep --color=auto apache2
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ kill -15 112490
bash: kill: (112490) - Operación no permitida
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ sudo kill -15 112490
[sudo] contraseña para alumno:
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps aux | grep apache2
www-data  112697  0.0  0.0   3616 1544 ?        Ss   19:53   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod
d_cache_disk -l 300M -n
alumno    113223  0.0  0.0   6648 2176 pts/0    S+   20:10   0:00 grep --color=auto apache2
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ sudo systemctl restart apache2
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps aux | grep apache2
www-data  112697  0.0  0.0   3616 1544 ?        Ss   19:53   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod
d_cache_disk -l 300M -n
root      113241  0.0  0.1  6804 4836 ?        Ss   20:13   0:00 /usr/sbin/apache2 -k start
www-data  113243  0.0  0.1 1212944 5616 ?        Sl   20:13   0:00 /usr/sbin/apache2 -k start
www-data  113244  0.0  0.1 1212944 5616 ?        Sl   20:13   0:00 /usr/sbin/apache2 -k start
alumno    113300  0.0  0.0   6648 2432 pts/0    S+   20:13   0:00 grep --color=auto apache2
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ kill -9 113241
bash: kill: (113241) - Operación no permitida
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ sudo kill -9 113241
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps aux | grep apache2
www-data  112697  0.0  0.0   3616 1544 ?        Ss   19:53   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod
d_cache_disk -l 300M -n
root      113313  0.0  0.1  6804 4852 ?        Ss   20:13   0:00 /usr/sbin/apache2 -k start
www-data  113314  0.0  0.1 1212944 5504 ?        Sl   20:13   0:00 /usr/sbin/apache2 -k start
www-data  113316  0.0  0.1 1212944 5504 ?        Sl   20:13   0:00 /usr/sbin/apache2 -k start
alumno    113372  0.0  0.0   6648 2432 pts/0    S+   20:14   0:00 grep --color=auto apache2
```

En esta siguiente captura usaremos `<< sudo kill -SIGSTOP PID >>` y `<< sudo kill -SIGCONT PID >>`. En el lado derecho usaremos los comandos mientras que en lado izquierdo vamos haciendo la confirmación de que se han llevado a cabo los cambios.

```
abr 03 20:24:03 alumno-VirtualBox apachectl[113552]: AH00558: SIGCONT 113553
abr 03 20:24:03 alumno-VirtualBox systemd[1]: Started apache2.service
alumno@alumno-VirtualBox: ~$ ps -o pid,state,command -p 1135
53
PID S COMMAND
113553 T /usr/sbin/apache2 -k start
alumno@alumno-VirtualBox: ~$ ps -o pid,state,command -p 1135
53
PID S COMMAND
113553 S /usr/sbin/apache2 -k start
alumno@alumno-VirtualBox: ~$
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps -o pid
,state,command -p 113553
PID S COMMAND
113553 S /usr/sbin/apache2 -k start
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ sudo kill
-SIGSTOP 113553
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ sudo kill
-SIGCONT 113553
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ ps aux |
grep apache2
www-data  112697  0.0  0.0   3616 1544 ?        Ss   19:53
```

Y a continuación usamos el comando `<< nice -n 10 yes > /dev/null >>` para ejecutar un proceso con baja prioridad.

```
alumno@alumno-VirtualBox: ~/Documentos/Practica_9$ nice -n 1
0 yes > /dev/null
[+]
0[|] 4.0% Tasks: 122, 439 thr, 74 kth
1[|] 100.0% Load average: 0.95 0.64 0.4
Mem[|] 1.13G/3.94G Uptime: 1 day, 01:48:34
Swp[|] 329M/3.94G

Main 1/0
PID USER PRI NI VIRT RES SHR S CPU% MEM%
118031 alumno 30 10 6020 2304 2176 R 103.6 0.1
118034 alumno 20 0 8772 4992 3712 R 4.0 0.1
2715 alumno 20 0 4022M 331M 125M S 0.7 8.2 2
2744 alumno 20 0 4022M 331M 125M S 0.7 8.2 1
2745 alumno 20 0 4022M 331M 125M S 0.7 8.2 1
115537 alumno 20 0 691M 67260 48380 S 0.7 1.6
```

Parte 3 - Monitorización avanzada

Usaremos la herramienta top para ajustar el intervalo de actualización de los procesos activos en el sistema. En este caso usaremos << **top -d 5** >> para poder mostrar los procesos con una actualización constante cada 5 segundos.

```
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ top -d 5

top - 17:59:58 up 1 day,  2:02,  1 user,  load average: 0,18, 0,21, 0,31
Tareas: 194 total,  1 ejecutar, 193 hibernar,  0 detener,  0 zombie
%Cpu(s):  1,0 us,  1,0 sy,  0,0 ni, 98,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  4033,1 total,  1343,5 libre,  1388,6 usado,  1645,1 búf/caché
MiB Intercambio:  4033,0 total,  3687,2 libre,  345,8 usado.  2644,5 dispon

  PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  HORA+  ORDEN
    2715 alumno    20   0 4121656 337740 126148 S   8,3   8,2 59:26.74 gnome-s+
118118 alumno    20   0 12248   5632  3584 R   8,3   0,1 0:00.02 top
     1 root      20   0  23368  11776  7680 S   0,0   0,3 0:42.90 systemd
     2 root      20   0      0      0      0 S   0,0   0,0 0:00.06 kthreadd
     3 root      20   0      0      0      0 S   0,0   0,0 0:00.00 pool_wor+
     4 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
     5 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
     6 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
     7 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
    10 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
    12 root      0 -20      0      0      0 I   0,0   0,0 0:00.00 kworker+
    13 root      20   0      0      0      0 I   0,0   0,0 0:00.00 rcu_tas+
    14 root      20   0      0      0      0 I   0,0   0,0 0:00.00 rcu_tas+
    15 root      20   0      0      0      0 I   0,0   0,0 0:00.00 rcu_tas+
```

Al ejecutar el comando << **top -o %MEM** >>ordenaremos los procesos según su uso de memoria. También es posible ordenar por uso de CPU, tal como se muestra en la imagen.

```
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ top -o %MEM

top - 17:59:23 up 1 day,  2:01,  1 user,  load average: 0,25, 0,22, 0,32
Tareas: 194 total,  1 ejecutar, 193 hibernar,  0 detener,  0 zombie
%Cpu(s):  1,4 us,  1,9 sy,  0,0 ni, 96,6 id,  0,0 wa,  0,0 hi,  0,1 si,  0,0 st
MiB Mem :  4033,1 total,  1343,5 libre,  1388,6 usado,  1645,1 búf/caché
MiB Intercambio:  4033,0 total,  3687,2 libre,  345,8 usado.  2644,5 dispon

  PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  HORA+  ORDEN
    3404 alumno    20   0 1781864 385944 116492 S   0,0   9,3 2:19.35 xdg-des+
    2715 alumno    20   0 4121656 337740 126148 S   5,1   8,2 59:25.61 gnome-s+
    5863 alumno    20   0 1358784 166428  42608 S   0,0   4,0 0:13.27 snapd-d+
   115537 alumno    20   0  706616  65768  46248 S   1,6   1,6 0:35.25 gnome-t+
   111387 alumno    20   0 2947720  64964  49756 S   0,0   1,6 0:02.07 gjs
    58048 root      20   0  584008  35888  31276 S   0,0   0,9 0:15.67 fwupd
    3538 alumno    20   0 1098832  31720  30552 S   0,0   0,8 0:01.21 mutter-+
    2866 alumno    20   0  824256  29572  27524 S   0,0   0,7 0:00.65 evoluti+
   44366 root      20   0 1395812  26172  11776 S   0,0   0,6 0:25.65 snapd
    3514 alumno    20   0  623236  25704  23832 S   0,0   0,6 0:00.45 gsd-xse+
    3389 alumno    39  19  957084  25456  15204 S   0,0   0,6 0:07.18 tracker+
    3677 alumno    20   0  578780  25256  20776 S   0,0   0,6 0:04.29 update-+
    2815 alumno    20   0 2593780  24132  21248 S   0,0   0,6 0:00.17 gjs
```

Al ejecutar el comando << **top -b -n 1 > procesos_top.txt** >> exportamos la información detallada de los procesos a un archivo de texto. A continuación, mostramos el script con el comando << **cat procesos_top.txt** >> con el cual mostraremos el contenido del archivo.

```
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ top -b -n 1 > procesos_top.txt
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ ls
2025-04-03_19:33:03_PARTE1.txt  procesos_top.txt
2025-04-07_17:23:56_PARTE1.txt  prueba.sh
alumno@alumno-VirtualBox:~/Documentos/Practica_9$ cat procesos_top.txt
top - 18:00:51 up 1 day,  2:03,  1 user,  load average: 0,25, 0,22, 0,31
Tareas: 195 total,   1 ejecutar, 194 hibernar,   0 detener,   0 zombie
%Cpu(s):  0,4 us,   0,8 sy,   0,0 ni, 98,3 id,   0,4 wa,   0,0 hi,   0,0 si,   0,0 st
MiB Mem :   4033,1 total,   1345,4 libre,   1386,6 usado,   1645,1 búf/caché
MiB Intercambio:   4033,0 total,   3687,2 libre,   345,8 usado.   2646,4 dispon

  PID  USUARIO  PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    HORA+  ORDEN
 2715  alumno    20   0 4121640 337740 126148 S   4,7   8,2 59:28.62  gnome-s+
115537 alumno    20   0 706616  65768  46248 S   2,3   1,6  0:35.76  gnome-t+
118186 alumno    20   0   12092   5888   3712 R   2,3   0,1  0:00.02  top
    1  root      20   0   23368  11776   7680 S   0,0   0,3  0:42.92  systemd
    2  root      20   0         0         0         0 S   0,0   0,0  0:00.06  kthreadd
    3  root      20   0         0         0         0 S   0,0   0,0  0:00.00  pool_wo+
    4  root       0 -20         0         0         0 I   0,0   0,0  0:00.00  kworker+
    5  root       0 -20         0         0         0 I   0,0   0,0  0:00.00  kworker+
    6  root       0 -20         0         0         0 I   0,0   0,0  0:00.00  kworker+
    7  root       0 -20         0         0         0 I   0,0   0,0  0:00.00  kworker+
```