

# Laboratorio 1: Administrador de Procesos — Parte A: Procesos y Comunicación Entre Procesos

**Integrantes:** Facundo Nicolás Farias Lozano, Juan Cruz Paez, Tomás Agustín Muñoz.

**Docentes:** Mg. Cs. Molina Silvia, Dra. Natalia Miranda, Mg. Palacio Gabriela.

Sistemas Operativos  
Universidad Nacional de San Luis  
2025

## Ejercicio 1: Acerca de Linux

---

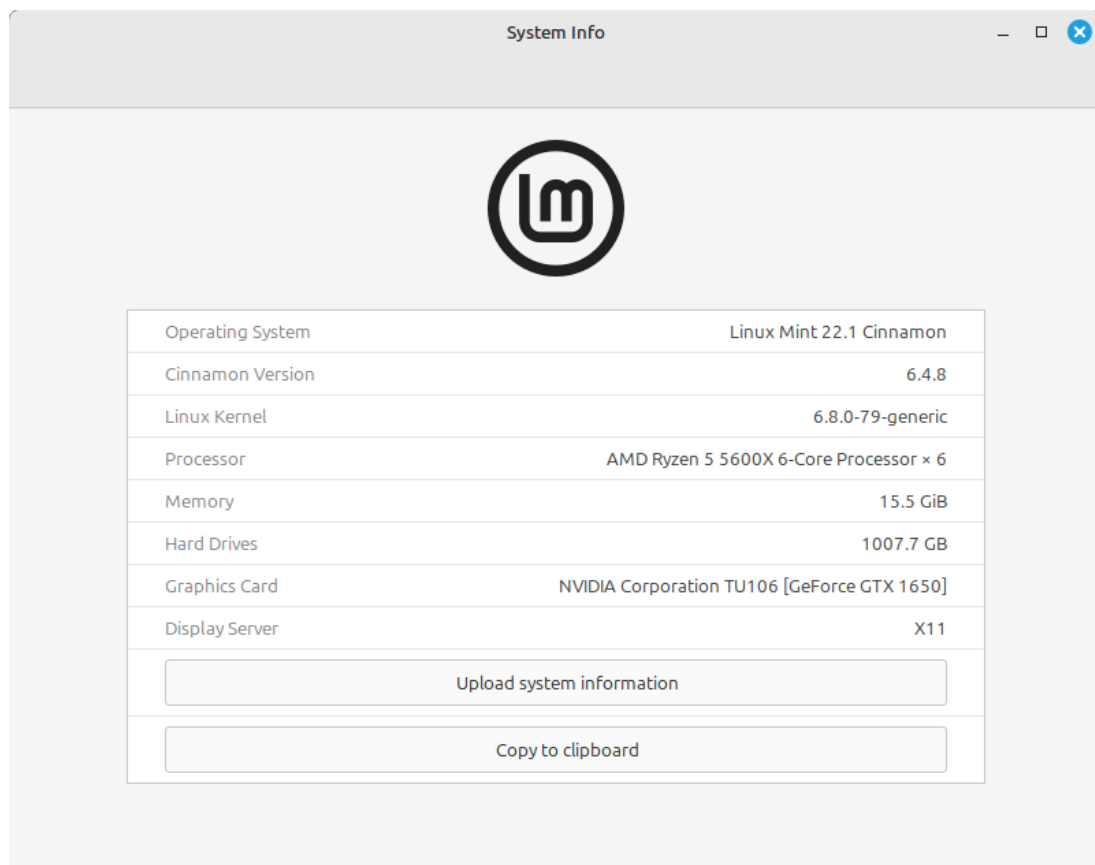


Figura 1: System info

**a- ¿Qué Sistema Operativo tiene instalado? ¿Qué distribución? ¿Qué versión?**

El sistema operativo que se encuentra instalado es *Linux Mint 22.1 Cinnamon* y es la version 6.4.8

**b- ¿Cuántos procesadores posee su sistema de computadora? ¿Cuáles son sus características?**

El sistema cuenta con un procesador *AMD Ryzen 5 5600X*, el cual posee 6 núcleos físicos y 12 hilos lógicos, arquitectura *Zen 3* y frecuencia base de 3.7 GHz hasta 4.6 Ghz

**c- ¿Cuál es la capacidad de memoria disponible?**

La memoria disponible es de *15.5 GiB*.

**d- ¿Qué placa de video o gráfica posee?**

La placa de vídeo es la *GeForce GTX 1650 NVIDIA*.

**e- ¿Cuál es la capacidad de disco que posee?**

La capacidad de disco con la que se cuenta es de *1007.7 GB*.

## Ejercicio 2: Monitor de sistema

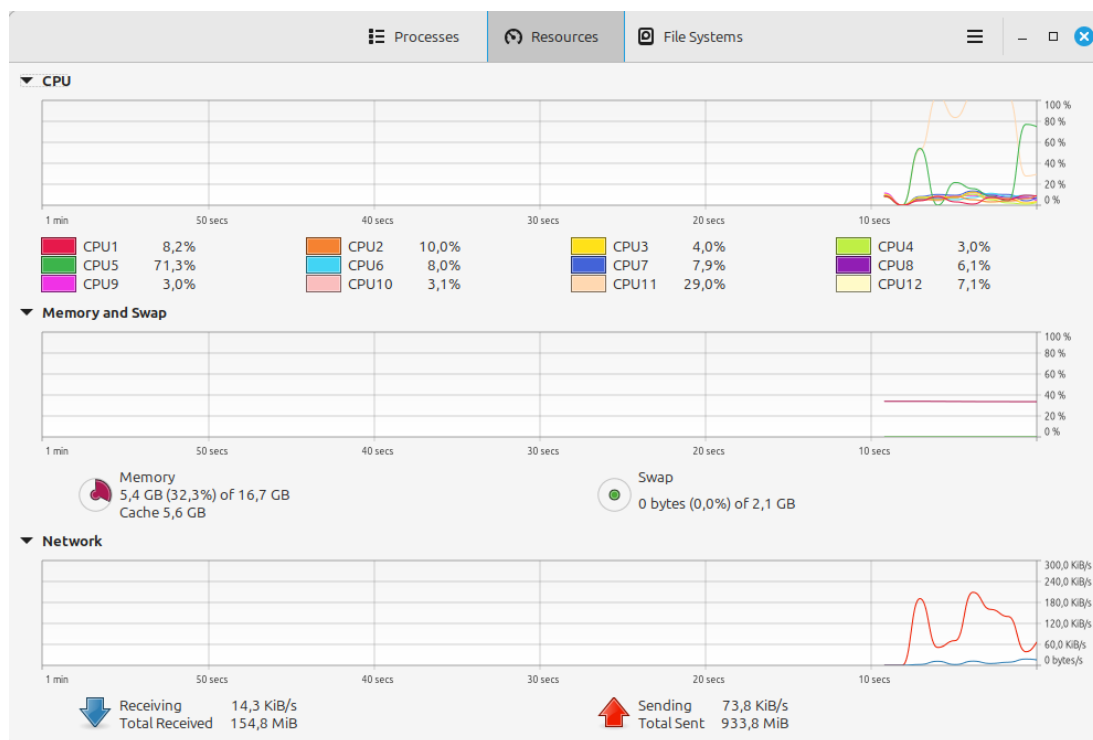


Figura 2: System monitor

**a- ¿Qué información del sistema muestra?**

Al abrir el *monitor de sistema* podemos observar:

- **Uso de la CPU:** Muestra un histograma del uso de los procesador en tiempo real.
- **Uso de la memoria y espacio intercambiado (swap)**<sup>1</sup>: Muestra de forma gráfica el uso de la memoria RAM y el espacio de intercambio.
- **Actividad de Red:** Muestra de manera gráfica la transmisión de datos de red.

**b- Mencione y describa qué información relevante sobre “Procesos” se puede mostrar (pestaña de “Procesos”).**

En dicha pestaña se muestra información en tiempo real de todos los procesos activos en el sistema. De ellos podemos ver:

- **Nombre de Proceso:** Indica el programa que está ejecutando

<sup>1</sup>Herramienta de gestión de memoria que actúa cuando la memoria RAM física se llena

Processes									
Process Name	User	% CPU	ID	Memory	Disk read total	Disk write total	Disk read	Disk write	Priority
xorg	root	0,78	1314	30,0 MB	N/A	N/A	N/A	N/A	Normal
kworker/8:2-cgroup_destroy	root	0,00	12967	N/A	N/A	N/A	N/A	N/A	Normal
systemd-journald	root	0,00	460	917,5 kB	N/A	N/A	N/A	N/A	Normal
pipewire	sirius	0,06	1630	10,9 MB	1,6 MB	N/A	N/A	N/A	Very High
pipewire-pulse	sirius	0,17	1636	38,2 MB	880,6 kB	N/A	N/A	N/A	Very High
wireplumber	sirius	0,00	1634	8,8 MB	24,8 MB	155,6 kB	N/A	N/A	Very High
kworker/0:0H-events_highpri	root	0,00	10	N/A	N/A	N/A	N/A	N/A	Very High
kworker/0:1H-kblockd	root	0,00	137	N/A	N/A	N/A	N/A	N/A	Very High
kworker/10:0H-events_highpri	root	0,00	80	N/A	N/A	N/A	N/A	N/A	Very High
kworker/10:1H-kblockd	root	0,00	197	N/A	N/A	N/A	N/A	N/A	Very High
kworker/1:0H-events_highpri	root	0,00	26	N/A	N/A	N/A	N/A	N/A	Very High
kworker/11:0H-events_highpri	root	0,00	86	N/A	N/A	N/A	N/A	N/A	Very High
kworker/11:1H-kblockd	root	0,00	234	N/A	N/A	N/A	N/A	N/A	Very High
kworker/1:1H-kblockd	root	0,00	209	N/A	N/A	N/A	N/A	N/A	Very High
kworker/2:0H-events_highpri	root	0,00	32	N/A	N/A	N/A	N/A	N/A	Very High
kworker/2:1H-kblockd	root	0,00	233	N/A	N/A	N/A	N/A	N/A	Very High
kworker/3:0H-events_highpri	root	0,00	38	N/A	N/A	N/A	N/A	N/A	Very High
kworker/3:1H-kblockd	root	0,00	242	N/A	N/A	N/A	N/A	N/A	Very High
kworker/4:0H-events_highpri	root	0,00	44	N/A	N/A	N/A	N/A	N/A	Very High

Figura 3: System monitor: Process

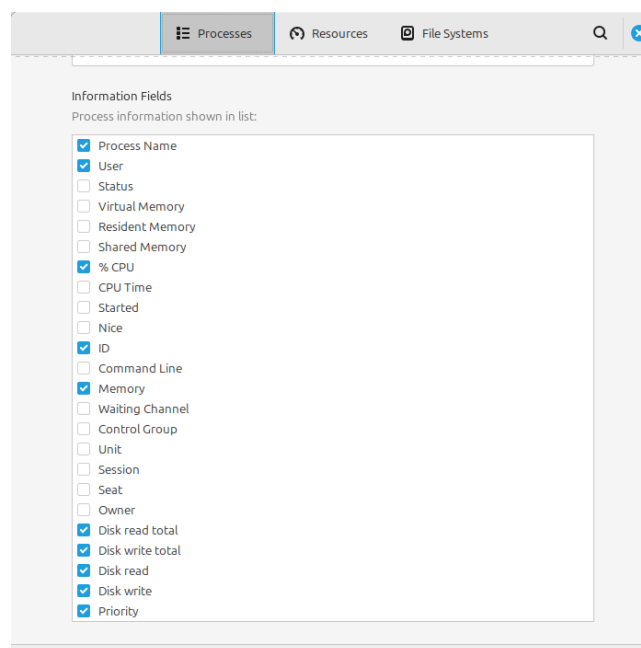


Figura 4: System monitor: informacion de procesos

- **Usuario:** Indica que usuario inicio el proceso.
- **Uso de CPU:** Porcentaje del procesador que esta consumiendo el proceso
- **Id:** Numero unico que identifica a cada proceso
- **Memoria:** Cantidad de memoria RAM que esta consumiendo el proceso
- **Lectura de disco:** Cantidad de informacion leida desde el disco por el proceso
- **Escritura de disco:** Cantidad de informacion escrita en el disco por el proceso
- **Prioridad:** Indica que prioridad le da el sistema al proceso frente a otros.
- **Estado:** Indica si el proceso esta durmiendo, ejecutandose,etc

Además, desde la pestaña procesos se agrega la posibilidad de mostrar información útil como **CPU Time** y **Waiting Channel**.

### c- ¿Qué operaciones se permite hacer respecto a los procesos?

Las operaciones que se permiten hacer sobre un proceso son:

- Open Files
- Change Priority
- Set Affinity
- Stop
- continue
- Terminate
- Memory Maps
- Kill
- End Procesos
- Show Process Properties

#### d- ¿Cómo se muestran las prioridades de un proceso?

Las prioridades se muestran en una columna dedicada, donde cada proceso (fila) puede tomar los valores de: *Very High, High, Normal, Low, Very Low, Custom*

**e- Inicializar las siguientes aplicaciones: un navegador Web, un procesador de texto y una terminal/console, luego responda observando la pestaña “Procesos” del Monitor de sistema:**

Nombre Proceso	PID	Estado	Ejecutando	Tipo Cola
bash	8039	Sleeping	No	do_select
brave	10361	Running	Si	futex_wait_queue
xed	15517	Sleeping	No	do_poll.constprop.0



Detener un proceso; este quedará en memoria, pero no se puede ejecutar hasta que se reanude (se cuelga). Finalizarlo hará que el proceso finalice, aunque existen casos en que el proceso puede ignorar la señal. Matarlo lo finaliza sin excepciones.

**f- Desde la pestaña “Procesos”, ordene los procesos por número de proceso, donde se muestren los siguientes datos (identificador del procesos, nombre del proceso, usuario, propietario, estado, prioridad, memoria real):**

Process Name	User	Status	ID	Memory	Owner	Priority
agetty	root	Sleeping	1431	131,1 kB		Normal
lightdm	root	Sleeping	1540	1,2 MB	sirius	Normal
systemd	sirius	Sleeping	1547	2,8 MB	sirius	Normal
(sd-pam)	sirius	Sleeping	1548	2,0 MB	sirius	Normal
pipewire	sirius	Sleeping	1564	786,4 kB	sirius	Normal
cinnamon-session-binary	sirius	Sleeping	1567	2,2 MB	sirius	Normal
dbus-daemon	sirius	Sleeping	1578	2,4 MB	sirius	Normal
gnome-keyring-daemon	sirius	Sleeping	1804	1,0 MB	sirius	Normal
csd-background	sirius	Sleeping	1809	13,5 MB	sirius	Normal
csd-print-notifications	sirius	Sleeping	1810	1,3 MB	sirius	Normal

Figura 5: System monitor: informacion de procesos

Proceso	PID	Nombre Proceso	Usuario	Propietario	Estado	Prioridad	Memoria
1	1431	agetty	root	—	Sleeping	Normal	131,1 kB
3	1647	systemd	root	siruis	Sleeping	Normal	2,8 MB
5	1564	pipewire	sirius	siruis	Sleeping	Normal	786,4 kB
7	1678	dbus-daemon	sirius	siruis	Sleeping	Normal	2,4 MB
9	1809	csd-background	sirius	siruis	Sleeping	Normal	13,5 MB

### Ejercicio 3: Simulador de Planificador de procesos

**a- ¿Qué partes del diagrama del ciclo de vida de un proceso se pueden visualizar en la ventana principal?**

En el diagrama de ejecución podemos observar los estados del ciclo de vida

- New
- Ready
- Running
- Bloqued
- Exit

**b- ¿Qué información de los procesos puede ser visualizada? Ejemplifique.**

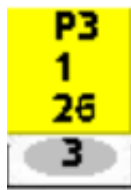


Figura 6: Información de un proceso en el simulador

Analizando la imagen, podemos obtener distintos datos sobre cada proceso, por ejemplo:

- **P3:** indica el nombre o identificador del proceso.
- **1:** representa el número de ráfagas restantes que el proceso debe ejecutar.
- **26:** muestra la cantidad de unidades de tiempo que le faltan para completar la ráfaga actual.
- **3:** indica la prioridad asignada al proceso dentro del sistema.

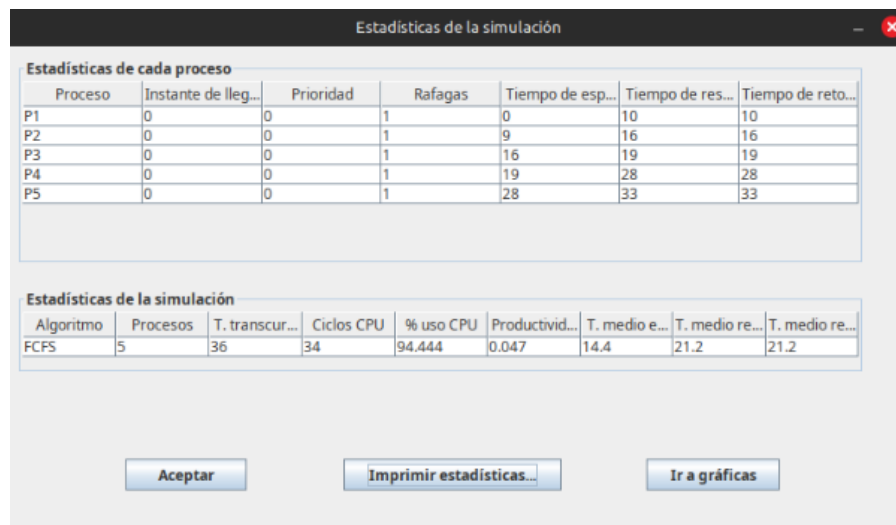
**c- Respecto a la configuración de la simulación. ¿Qué información de los procesos se puede configurar?**

La información que se puede configurar para los procesos incluye:

- **Algoritmo de planificación** a utilizar y, si corresponde, la especificación del quantum.
- **Llegadas de los procesos**, que pueden simularse según distintas distribuciones de probabilidad: uniforme, exponencial, normal o Poisson.
- **Instante de llegada** de cada proceso.
- **Prioridad** asignada a cada proceso.
- **Número de ráfagas** que debe ejecutar el proceso.
- **Tipo y duración** de cada ráfaga (CPU o E/S).
- **Limitaciones de las ráfagas**, pudiendo restringirlas específicamente para E/S o CPU.

#### d- ¿Qué información importante se puede observar una vez ejecutada la simulación?

Al acceder a la pestaña Resultados, encontramos tres secciones distintas que proporcionan información valiosa: *Estadísticas* permite consultar información detallada de la simulación. Incluye datos específicos de cada proceso, como instante de llegada, prioridad, número de ráfagas, tiempo de espera, entre otros. Además, muestra estadísticas globales propias de toda la simulación.



The screenshot shows a window titled 'Estadísticas de la simulación'. It contains two tables. The first table, 'Estadísticas de cada proceso', lists data for processes P1 through P5. The second table, 'Estadísticas de la simulación', provides overall simulation metrics. At the bottom are three buttons: 'Aceptar', 'Imprimir estadísticas...', and 'Ir a gráficas'.

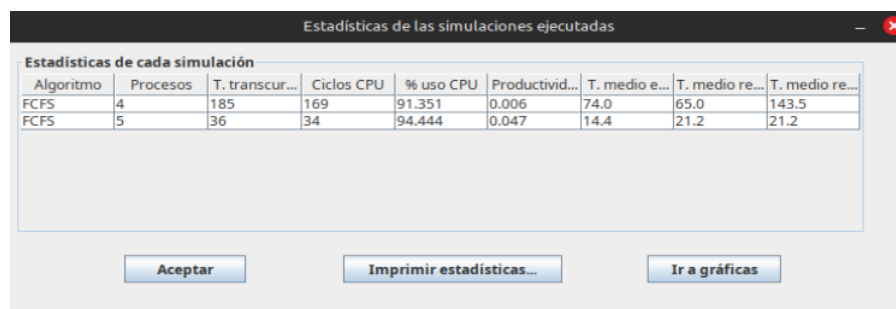
Proceso	Instante de lleg...	Prioridad	Ráfagas	Tiempo de esp...	Tiempo de res...	Tiempo de reto...
P1	0	0	1	0	10	10
P2	0	0	1	9	16	16
P3	0	0	1	16	19	19
P4	0	0	1	19	28	28
P5	0	0	1	28	33	33

Algoritmo	Procesos	T. transcur...	Ciclos CPU	% uso CPU	Productivid...	T. medio e...	T. medio re...	T. medio re...
FCFS	5	36	34	94.444	0.047	14.4	21.2	21.2

Figura 7: Estadísticas específicas y globales

**Comparación** En esta sección se presentan los datos globales de cada simulación realizada, facilitando el análisis comparativo entre diferentes ejecuciones.



The screenshot shows a window titled 'Estadísticas de las simulaciones ejecutadas'. It contains a table comparing two simulation runs. At the bottom are three buttons: 'Aceptar', 'Imprimir estadísticas...', and 'Ir a gráficas'.

Algoritmo	Procesos	T. transcur...	Ciclos CPU	% uso CPU	Productivid...	T. medio e...	T. medio re...	T. medio re...
FCFS	4	185	169	91.351	0.006	74.0	65.0	143.5
FCFS	5	36	34	94.444	0.047	14.4	21.2	21.2

Figura 8: Comparación de simulaciones

**Gráficos** Permite visualizar de manera gráfica los datos anteriormente mencionados, tanto los específicos de cada proceso como las estadísticas globales de cada simulación.<sup>2</sup>

<sup>2</sup>**Nota:** Todas las imágenes presentadas corresponden a los resultados obtenidos a partir de la simulación utilizando los datos de la Tabla 1.



Figura 9: Izquierda: Gráfico de tiempo de espera en cola | Derecha: Tiempos medios de espera

## Ejercicio 4: Inicialice la aplicación Terminal de Linux

a- Utilizando el comando ps listar los procesos del sistema.

```
Command Line
tomas@tomas-HP-245-G7-Notebook-PC:~$ ps
  PID TTY          TIME CMD
 10820 pts/0        00:00:00 bash
 10828 pts/0        00:00:00 ps
```

b- Utilizando el comando ps con el parámetro -u listar los procesos del usuario actual únicamente.

```
Command Line
tomas@tomas-HP-245-G7-Notebook-PC:~$ ps -u
  USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
  tomas    10820  0.0  0.0  14060  5248 pts/0    Ss   11:00   0:00 bash
  tomas    10861 100  0.0  16492  4608 pts/0    R+   11:02   0:00 ps -u
```

c- Utilizando el comando top listar los procesos.

## Command Line

```
top - 11:05:42 up 2:04, 1 user, load average: 1,82, 1,58, 1,72
Tareas: 249 total, 1 ejecutar, 248 hibernar, 0 detener, 0 zombie
%Cpu s: 15,2 us, 3,9 sy, 0,0 ni, 80,3 id, 0,1 wa, 0,0 hi, 0,5 si, 0,0 st
MiB Mem: 5835,5 total, 898,8 libre, 2940,5 usado, 2320,9 búf/caché
MiB Intercambio: 2048,0 total, 1618,2 libre, 429,8 usado. 2895,0 dispon Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
5365	tomas	20	0	1398,9g	538740	179392	S	52,5	9,0	84:04.65	Discord
5277	tomas	20	0	32,6g	166108	129620	S	13,3	2,8	14:16.99	Discord
2220	root	20	0	1303608	160792	95956	S	3,3	2,7	10:06.64	Xorg
2694	tomas	9	-11	162348	34900	11204	S	2,7	0,6	4:23.31	pipewire-pulse
3122	tomas	20	0	4942804	275096	120592	S	2,7	4,6	11:39.49	cinnamon
631	root	-51	0	0	0	0	S	2,3	0,0	2:58.75	irq/45-rtw88_pci
2690	tomas	9	-11	126460	16084	9264	S	1,7	0,3	3:19.48	pipewire
1	root	20	0	22988	13312	9216	S	0,7	0,2	0:02.93	systemd
1162	root	35	15	9816	6664	2304	S	0,3	0,1	0:03.52	preload
2251	mysql	20	0	2375904	254476	10496	S	0,3	4,3	0:56.81	mysqld
2788	tomas	20	0	312236	6144	5888	S	0,3	0,1	0:00.33	xdg-permission-
3817	tomas	20	0	32,9g	367464	253916	S	0,3	6,1	3:51.85	brave
3953	tomas	20	0	1394,0g	439296	145652	S	0,3	7,4	7:00.65	brave
5149	tomas	20	0	1392,1g	176052	136860	S	0,3	2,9	1:35.70	Discord
10288	root	20	0	0	0	0	I	0,3	0,0	0:11.34	kworker/u32:1-gfx_low
10659	root	20	0	0	0	0	I	0,3	0,0	0:00.09	kworker/u33:4-rtw_tx_wq
10895	tomas	20	0	545576	43008	34288	S	0,3	0,7	0:00.38	gnome-terminal-
10940	tomas	20	0	17348	6016	3840	R	0,3	0,1	0:00.07	top
2	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-netns
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
12	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-mm_pe
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_kthread
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_rude-kthread
15	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_trace-kthread
16	root	20	0	0	0	0	S	0,0	0,0	0:00.49	ksoftirqd/0
17	root	20	0	0	0	0	I	0,0	0,0	0:18.40	rcu_preempt
18	root	rt	0	0	0	0	S	0,0	0,0	0:00.03	migration/0

### d- ¿Cuál es la diferencia de utilizar el comando *top* respecto de utilizar el comando *ps*?

El comando *ps* muestra el estado de los procesos de un momento determinado, es estático, es decir que no se actualiza automáticamente.

Por otra parte, el comando *top* muestra de manera interactiva y en tiempo real como cambia el estado de los procesos.

### e- Observe la estructura jerárquica de los procesos en Linux utilizando el comando *pstree*.



## Command Line

```
systemd+-ModemManager---3*[{ModemManager}]
|-NetworkManager---3*[{NetworkManager}]
|-accounts-daemon---3*[{accounts-daemon}]
|-agetty
|-at-spi2-registr---3*[{at-spi2-registr}]
|-avahi-daemon---avahi-daemon
|-bluetoothd
|-2*[chrome_crashpad---2*[chrome_crashpad]]
|-chrome_crashpad---{chrome_crashpad}
|-colord---3*[{colord}]
|-cron
|-csd-printer---3*[{csd-printer}]
|-cups-browsed---3*[{cups-browsed}]
|-cupsd
|-dbus-daemon
|-fwupd---5*[{fwupd}]
|-irqbalance---{irqbalance}
|-2*[kerneloops]
'-lightdm+-Xorg--12*[{Xorg}]
    '-lightdm+-cinnamon-session---agent---3*[{agent}]
        |-at-spi-bus-laun---dbus-daemon---4*[{at-spi-bu+}]
        |-caribou---3*[{caribou}]
        |-cinnamon-killer---4*[{cinnamon-}]
        |-cinnamon-launch---cinnamon---27+[{cinnamon}]
        '-csd-ally-settin---6*[{cinnamon+}]
            '-4*[{csd-ally-+}]
```

Este comando muestra los procesos de manera que, indica cuales son *padres* y cuales *hijos*. *System* es el primer proceso que ejecuta linux y este va ser el padre de casi todos los demas. Podemos observar que algunos de sus hijos son *modem manager*, *network manager*, *accounts-daemon*, etc

**f- Ejecute el comando `htop`, luego visualice los estados de los procesos a través de la columna S (state).**

## Command Line

Here's the transcription of the information displayed in the htop interface:

Header Section:

CPUs: Multiple CPU usage meters (graphical bars with percentages, e.g., |||||20.0%, SHR S, etc.)

Mem: [|||||2.75G/5.700] (Memory usage: 2.75GB used out of 5.700GB total)

Swp: [|||50M/2.00G] (Swap usage: 50MB used out of 2.00GB total)

Tasks: 147, 759 thr, 122 kthr; 3 running

Load average: 1.37 2.10 1.93

Uptime: 02:27:03

Process List (with column headers):

PID	USER	PRI	NI	VRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
13617	tomas	20	0	7228M	5564M	3072	R	4.5	8.1	0:09.63	/snap/htop/5092/usr/local/bin/htop
1	root	20	0	22988	1696	9216	S	0.0	0.2	0:04.32	/sbin/init splash
291	root	19	-1	83736	33436	32156	S	0.0	8.6	0:02.45	/usr/lib/systemd/systemd-journald
362	root	20	0	30632	6204	4748	S	0.0	8.1	0:04.87	/usr/lib/systemd/systemd-udev
781	systemd-re	20	0	22480	12544	10624	S	0.0	0.2	0:00.37	/usr/lib/systemd/systemd-resolved
799	systemd-ti	20	0	91212	7296	912	S	0.0	0.1	0:00.18	/usr/lib/systemd/systemd-timesyncd
845	systemd-ti	20	0	91212	7296	0	S	0.0	0.1	0:00.00	/usr/lib/systemd/systemd-timesyncd
999	root	20	0	309M	7388	124	S	0.0	0.1	0:00.00	/usr/libexec/accounts-daemon
1000	root	20	0	309M	7388	0	S	0.0	0.1	0:00.00	/usr/libexec/accounts-daemon
1001	root	20	0	309M	7388	0	S	0.0	0.1	0:00.52	/usr/libexec/accounts-daemon
1003	avahi	20	0	6616	3968	3840	S	0.0	0.1	0:00.20	avahi-daemon: running[tomas-HP-245-G7-Notebook-PC.local]
1005	root	20	0	13544	5632	5248	S	0.0	0.1	0:00.12	/usr/libexec/bluetooth/bluetoothd
1006	root	20	0	12048	688	560	S	0.0	0.0	0:00.02	/usr/sbin/cron -f -P
1007	messagebus	20	0	6100S6	6656	352	S	0.0	0.1	0:02.43	@dbus-daemon --system --address=systemd: --nofork
--nospidfile --systemd-activation --syslog-only											
1015	root	20	0	82920	3712	584	S	0.0	0.1	0:00.76	/usr/sbin/irqbalance
1035	polkitd	20	0	155M	11988	524	S	0.0	0.2	0:00.57	/usr/lib/polkit-1/polkitd --no-debug
1038	root	20	0	82920	3712	0	S	0.0	0.1	0:00.00	/usr/sbin/irqbalance
1039	root	20	0	309M	7552	0	S	0.0	0.1	0:00.11	/usr/libexec/power-profiles-daemon
1046	root	20	0	19508	30444	10512	S	6.9	0.5	0:02.19	/usr/lib/snapd/snapd
1047	root	20	0	305M	6400	272	S	0.0	0.1	0:00.06	/usr/libexec/switcheroo-control
1048	root	20	0	309M	7388	0	S	0.0	0.1	0:00.01	/usr/libexec/accounts-daemon
1050	root	20	0	10148	6212	808	S	0.0	0.1	0:00.43	/usr/lib/systemd/systemd-logind
1062	root	20	0	253M	14208	12416	S	0.0	0.2	0:16.34	/usr/bin/touchegg-daemon
1064	root	20	0	308M	7552	0	S	0.0	0.1	0:00.00	/usr/libexec/power-profiles-daemon
1065	root	20	0	308M	7552	0	S	0.0	0.1	0:00.00	/usr/libexec/power-profiles-daemon
1067	root	20	0	308M	7552	0	S	0.0	0.1	0:00.00	/usr/libexec/power-profiles-daemon
1083	root	20	0	458M	11568	9776	S	0.0	0.2	0:00.22	/usr/libexec/udisks2/udisksd
1089	root	20	0	305M	6400	0	S	0.0	0.1	0:00.00	/usr/libexec/switcheroo-control
1090	root	20	0	305M	6400	0	S	0.0	0.1	0:00.00	/usr/libexec/switcheroo-control

Footer Menu:

F1Help F2Setup F3Search F4Filter F5Tree F6Sortby F7Nice- F8Nice+ F9Kill F10Quit

La columna *S* que resulta del comando *htop* muestra el estado de un proceso.

**g- Utilizando el comando kill listar las opciones de parámetros posibles utilizando el parámetro -l, luego investigue qué parámetros son necesarios para matar un proceso, muestre un ejemplo donde elige un proceso para matar y luego lo mata aplicando el comando con los parámetros correspondiente.**

Listamos todas las señales posibles con el comando *kill-l*.

## Command Line

tomas@tomas-HP-245-G7-Notebook-PC:~\$ kill -l

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL	5) SIGTRAP
6) SIGABRT	7) SIGBUS	8) SIGFPE	9) SIGKILL	10) SIGUSR1
11) SIGSEGV	12) SIGUSR2	13) SIGPIPE	14) SIGALRM	15) SIGTERM
16) SIGSTKFLT	17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
21) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO	30) SIGPWR
31) SIGSYS	34) SIGRTMIN	35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3
38) SIGRTMIN+4	39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7	42) SIGRTMIN+8
43) SIGRTMIN+9	44) SIGRTMIN+10	45) SIGRTMIN+11	46) SIGRTMIN+12	47) SIGRTMIN+13
48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14	51) SIGRTMAX-13	52) SIGRTMAX-12
53) SIGRTMAX-11	54) SIGRTMAX-10	55) SIGRTMAX-9	56) SIGRTMAX-8	57) SIGRTMAX-7
58) SIGRTMAX-6	59) SIGRTMAX-5	60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX			

Cada número o nombre representa una señal que podés enviar a un proceso.

Las más usadas para terminar procesos son:

- **SIGTERM** (15) → Finalización ordenada: consiste en solicitar al proceso que termine su ejecución, permitiéndole cerrar archivos, guardar datos, liberar memoria y limpiar recursos antes de finalizar. Sin embargo, el proceso puede optar por ignorar la señal y no terminar con su ejecución.
- **SIGKILL** (9) → Finalización forzada: se solicita al proceso que termine de manera inmediata y obligatoria, sin darle la posibilidad de cerrar archivos abiertos, liberar recursos o guardar datos. En este caso, el proceso no puede ignorar la señal y debe finalizar.

Veamos un ejemplo de uso:

Listamos los procesos usando el comando *top*

Command Line											
PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
3122	tomas	20	0	4945876	240424	122640	S	6,6	4,0	14:00.76	cinnamon
11227	tomas	20	0	3050700	332584	194800	S	4,6	5,6	1:12.07	spotify
5365	tomas	20	0	1398,9g	394004	121524	S	3,0	6,6	93:20.76	Discord
2694	tomas	9	-11	162324	35600	11264	S	2,0	0,6	5:14.95	pipewire-pulse

Mataremos el proceso Discord el cual tiene como PID 5365:

Para finalizar amigablemente, se debe usar *-SIGTERM* ó *-15* equivalentemente.

Command Line											
tomas@tomas-HP-245-G7-Notebook-PC:~\$ kill -15 5365											

En caso de que el programa no finalice procedemos a usar *-9* o *-SIGKILL*.

Command Line											
tomas@tomas-HP-245-G7-Notebook-PC:~\$ kill -9 5365											
bash: killK (5365) - No existe el proceso											

En este caso, el mensaje se refiere a que el proceso no existe porque fue eliminado anteriormente con *SIGTERM*.