ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

LABORATORIO DE SISTEMAS OPERATIVOS

NOMBRE: Andrés Felipe Merino Bravo

PERÍODO ACADÉMICO: 2025 – B **EQUIPO:**

PROFESOR: Msc. Marcela Saavedra

TIPO DE INSTRUMENTO: Guía de Laboratorio

TEMA: ADMNISTRACIÓN DE PROCESOS EN LINUX

Tópico 2: Comandos básicos, auditoría, de gestión de procesos, memoria y conectividad en Linux

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. OBJETIVOS

1.1. Familiarizar al estudiante con el uso de los comandos básicos del sistema operativo Linux.

2. MARCO TEÓRICO

- Linux es un S.O. multiusuario y multitarea:
- Muchos usuarios pueden ejecutar muchas tareas al mismo tiempo, independientes entre sí.
- Siempre es necesario hacer log in, antes de usar el sistema.
- Identificarse con nombre de usuario y contraseña.
- Múltiples formas de hacer login en el sistema:
  - Consola: teclado, mouse, monitor, conectados directamente.
  - Terminal serial, Shell.
  - Conexión vía red (telnet, ssh).

Comandos en Linux

Cualquier actividad o tarea puede hacerse en Linux mediante comandos, se incluye:

- Navegar en la web.
- Activar la GUI (Interfaz gráfica de usuario) (Sistema X Window) no es necesaria para correr el S.O. Linux.
- Para ejecutar comandos en X Window, se necesita un emulador de terminal.

Una de las tareas más importantes del sistema operativo (SO) es la gestión de los procesos que se están ejecutando en una máquina.

El hecho de que todos los procesos deban compartir los recursos hardware disponibles (memoria RAM, CPU, etc.) hace que el SO juegue un papel primordial en gestionar dichos recursos para que los procesos se ejecuten de forma simultánea (al menos de cara al usuario) y compatible.

Una posible definición de proceso es que es un programa que se encuentra en ejecución. Cada proceso, durante su ejecución, guarda información sobre su "contexto" que incluye, entre otras cosas, información sobre su proceso padre, los recursos del sistema que se están consumiendo (segmentos de memoria asignados), permisos de seguridad, etc.

3. PROCEDIMIENTO

3.1 Iniciando un emulador de terminal

Permite ejecutar un comando Linux dentro de un ambiente gráfico (X).

- Emula una CONSOLA DE TEXTO.
- El tipo de emulador de terminal depende de la distribución.

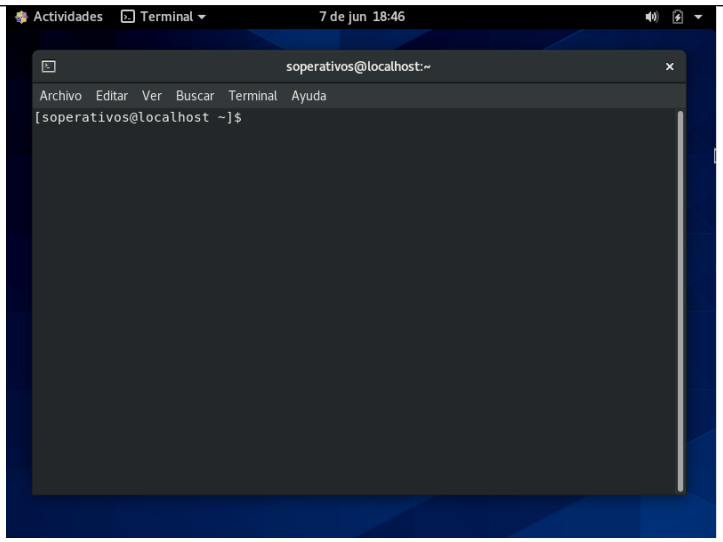


Figura 1. Terminal de comandos dentro del ambiente gráfico CentOS

3.2 Prompt de Comandos

El prompt de comandos indica que el sistema está listo para aceptar comandos.

- Puede ser personalizado.
- El prompt por defecto depende de la distribución.

Ejemplos:

Prompt para el usuario soperativos:

[soperativos@localhost ~]$

Prompt para el superusuario root:

[root@localhost ~] #

- El símbolo dólar ($) regularmente significa “logueado como usuario regular”.
- El símbolo numeral (#) usualmente significa “logueado como usuario root”.

3.3 Sintaxis de comandos Linux

Los comandos en Linux tienen el siguiente formato:

$ comando(s) opcion(es) argumento(s)

Ejemplos:

$ ls

$ ls -l

```
$ ls /dev
$ ls -l /dev
```

Ejemplos de formato de comandos

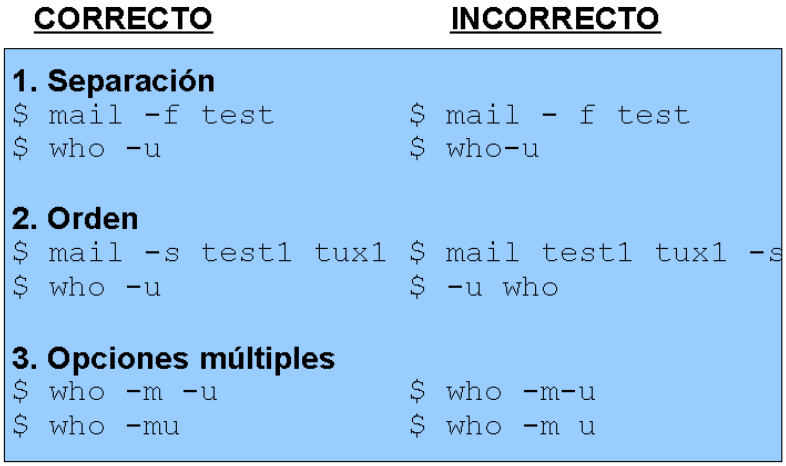


Figura 2. Formato de comandos para la terminal de CentOS

3.4 Algunos comandos básicos

Comando	Descripción	Ejemplos
at [-lr] hora [fecha]	Ejecuta un comando más tarde	at 6pm Friday miscript
cal [[mes] año]	Muestra un calendario del mes/año	cal 1 2025
date [mmddhhmm] [+form]	Muestra la hora y la fecha	date
echo string	Escribe mensaje en la salida estándar	echo "Hola mundo"
finger usuario	Muestra información general sobre un usuario en la red	finger druiz@pc11.lsi.us.es
id	Número id de un usuario	id usuario
kill [-señal] PID	Enviar una señal a un proceso (dependiendo de la señal, a veces lo finalizará)	kill 1234
man comando	Ayuda del comando especificado	man gcc
passwd	Cambia la contraseña.	passwd
ps [axiu]	Muestra información sobre los procesos que se están ejecutando en el sistema	ps -ux, ps -ef
who / rwho	Muestra información de los usuarios conectados al sistema	who

Tabla 1. Comandos básicos (parte 1)

Comando	Descripción	Ejemplos
cat f_1 [... f_n]	Concatena y muestra los archivos	cat /etc/passwd
cd [dir]	Cambia de directorio	cd /tmp
chmod permisos fich	Cambia los permisos de un archivo	chmod +x miscript
chown usuario:grupo fich	Cambia el dueño un archivo	chown nobody miscript
cp f_1... f_n dir	Copia archivos	cp foo foo.backup
diff [-e] f_1 f_2	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [-sabr] f_1 [... f_n]	Devuelve el tamaño del directorio	du -s /home/
file f	Muestra el tipo de un archivo	file a.out
find dir test acción	Encuentra archivos.	find . -name ".bak" -print
grep expr f_1 [... f_n]	Busca patrones en archivos	grep druiz /etc/passwd
head -n f	Muestra las n primeras líneas de un archivo	head prog1.c
mkdir dir	Crea un directorio.	mkdir temp
mv f_1 ... f_n dir	Mueve un archivo(s) a un directorio	mv a.out prog1
mv f_1 f_2	Renombra un archivo.	mv *.c prog_dir
less / more fich(s)	Visualiza página a página un archivo. (less acepta comandos vi)	more /less muy_largo.c
ln [-s] f acceso	Crea un acceso directo a un archivo	ln -s /users/mike/.profile .
ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	pwd
rm f	Borra un fichero.	rm foo.c
rm -r dir	Borra un todo un directorio	rm -rf prog_dir
rmdir dir	Borra un directorio vacío	rmdir prog_dir
tail -n fich	Muestra las n últimas líneas de un archivo	tail prog1.c
vi fich	Edita un archivo.	vi .profile

Tabla 2. Comandos básicos (parte 2)

3.5 Sistema de ayuda

Linux dispone de un completo sistema de ayuda. Podemos obtener ayuda sobre cualquier comando o sobre cualquier aspecto del sistema mediante el comando *man*, cuyo formato es:

man comando

Dónde:

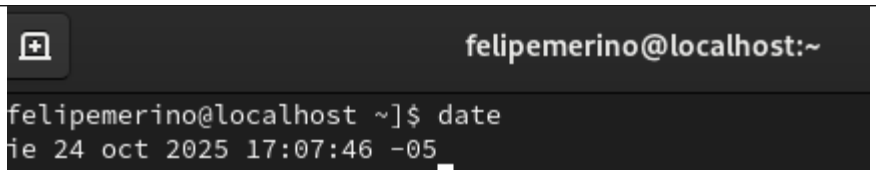
comando es el comando del cual se solicita información.

Ejemplo:

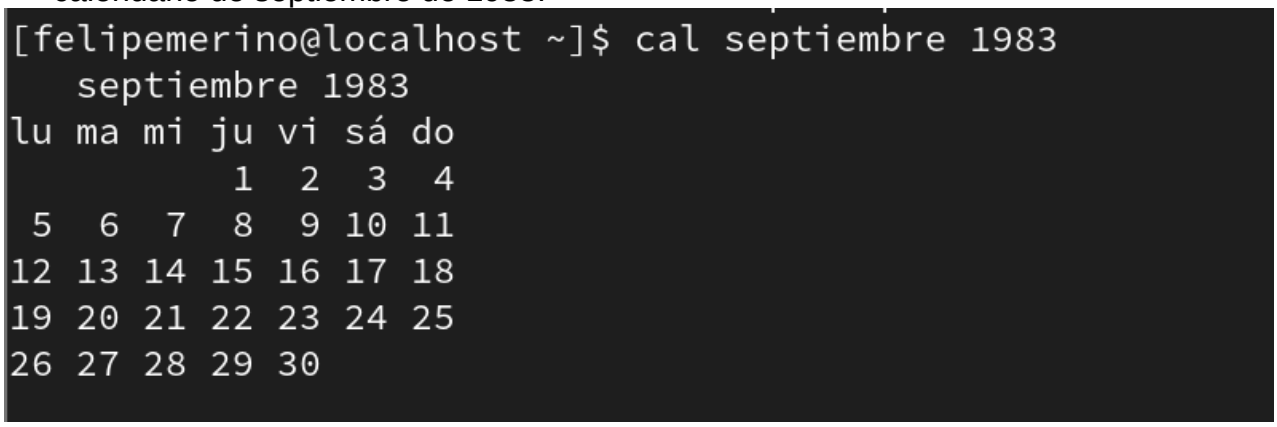
man ls

3.6. Mediante el uso de los comandos mencionados, realice las siguientes tareas (No mediante la interfaz gráfica).

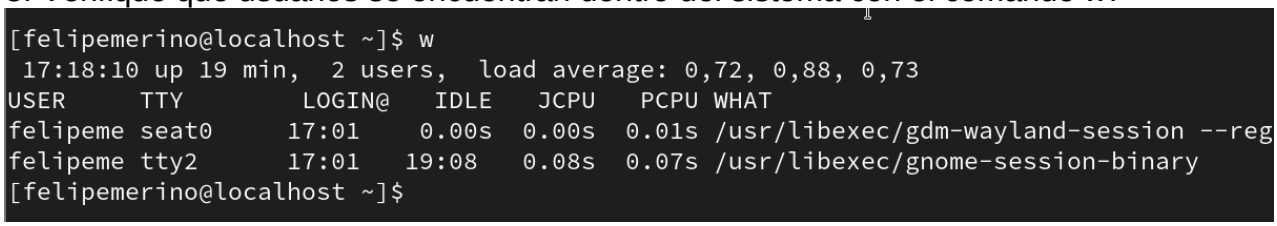
1. Verifique la fecha y hora de su computador con el comando *date*.



2. Muestre en pantalla el calendario con el comando *cal*. Luego muestre el calendario de septiembre de 1983.



3. Verifique qué usuarios se encuentran dentro del sistema con el comando *w*.



4. Limpie la pantalla de su terminal con el comando *clear*.

4. Muestre el contenido de las variables de entorno de la tabla debajo con el comando *echo* (respete el uso de mayúsculas y minúsculas).

```
[felipemerino@localhost ~]$ echo

[felipemerino@localhost ~]$ echo $DISPLAY
:0
[felipemerino@localhost ~]$ echo $HOME
/home/felipemerino
[felipemerino@localhost ~]$ echo $HOSTNAME
localhost
[felipemerino@localhost ~]$ echo $MAIL
/var/spool/mail/felipemerino
[felipemerino@localhost ~]$ echo $PATH
/home/felipemerino/.local/bin:/home/felipemerino/bin:/usr/local/bin:/usr/local/sbin:/us
in:/usr/sbin
[felipemerino@localhost ~]$ echo $PS1
[\u@\h \W]\$
[felipemerino@localhost ~]$ echo $SHELL
/bin/bash
[felipemerino@localhost ~]$ echo $TERM
xterm-256color
[felipemerino@localhost ~]$ echo $USER
felipemerino
[felipemerino@localhost ~]$ s
```

Ejemplo:

echo $DISPLAY

Variable	Descripción
DISPLAY	Dirección IP a donde se envían los gráficos de los clientes X.
HOME	Directorio personal.
HOSTNAME	Nombre de la máquina.
MAIL	Archivo de correo.
PATH	Lista de directorios donde buscar los programas.
PS1	Prompt.
SHELL	Intérprete de comandos por defecto.
TERM	Tipo de terminal.
USER	Nombre del usuario.

Tabla 3. Variables de entorno y su descripción

6. Muestre un mensaje de texto en la terminal de un usuario con el comando *write*.

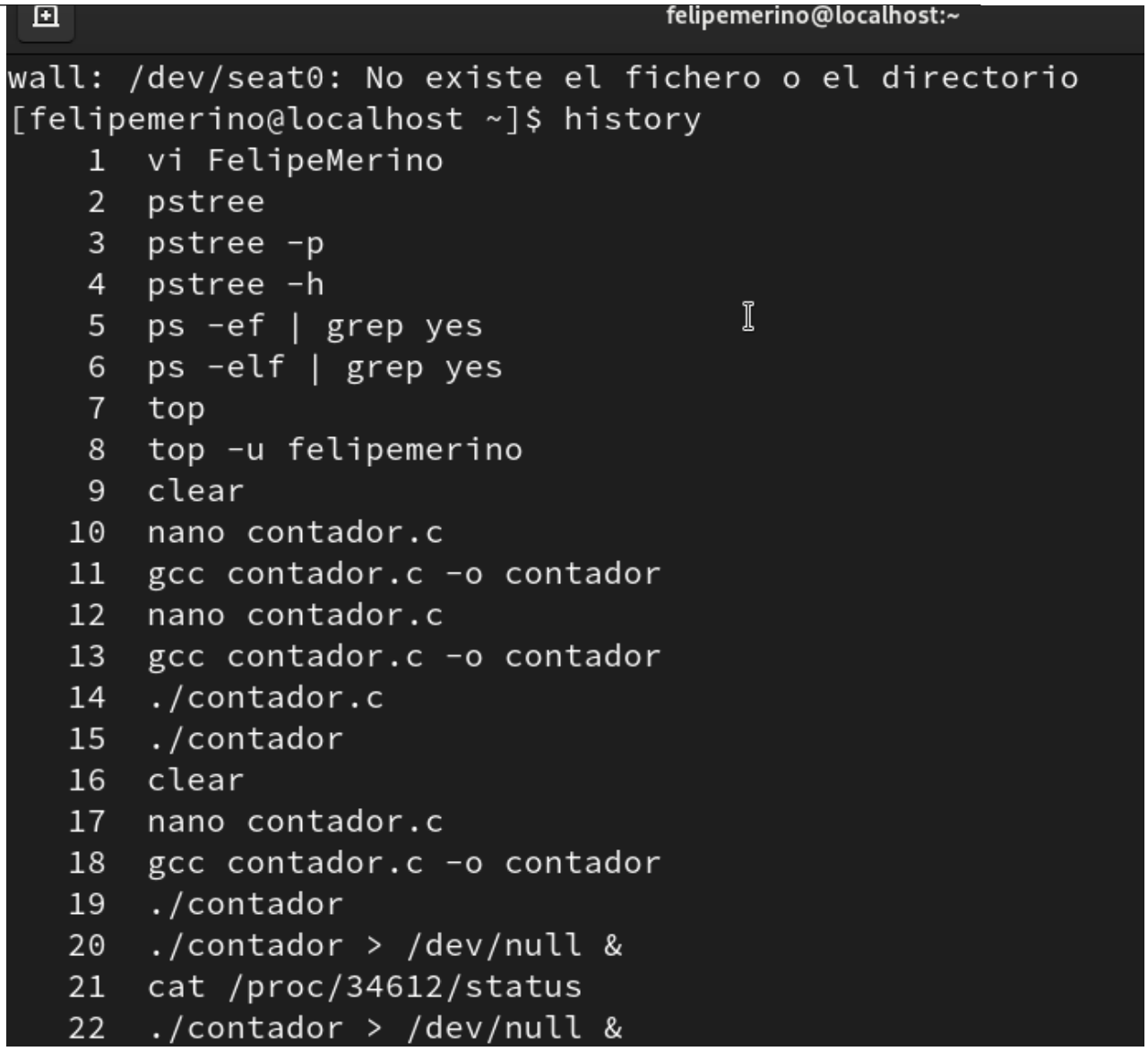
```
[felipemerino@localhost ~]$ write felipemerino tty3
Hola
Deber
SO
Gol
[felipemerino@localhost ~]$
```

```
[felipemerino@localhost ~]$
Mensaje de felipemerino@localhost.localdomain el pts/1 a las 00:49 ...
Hola
Deber
SO
Gol
EOF
```

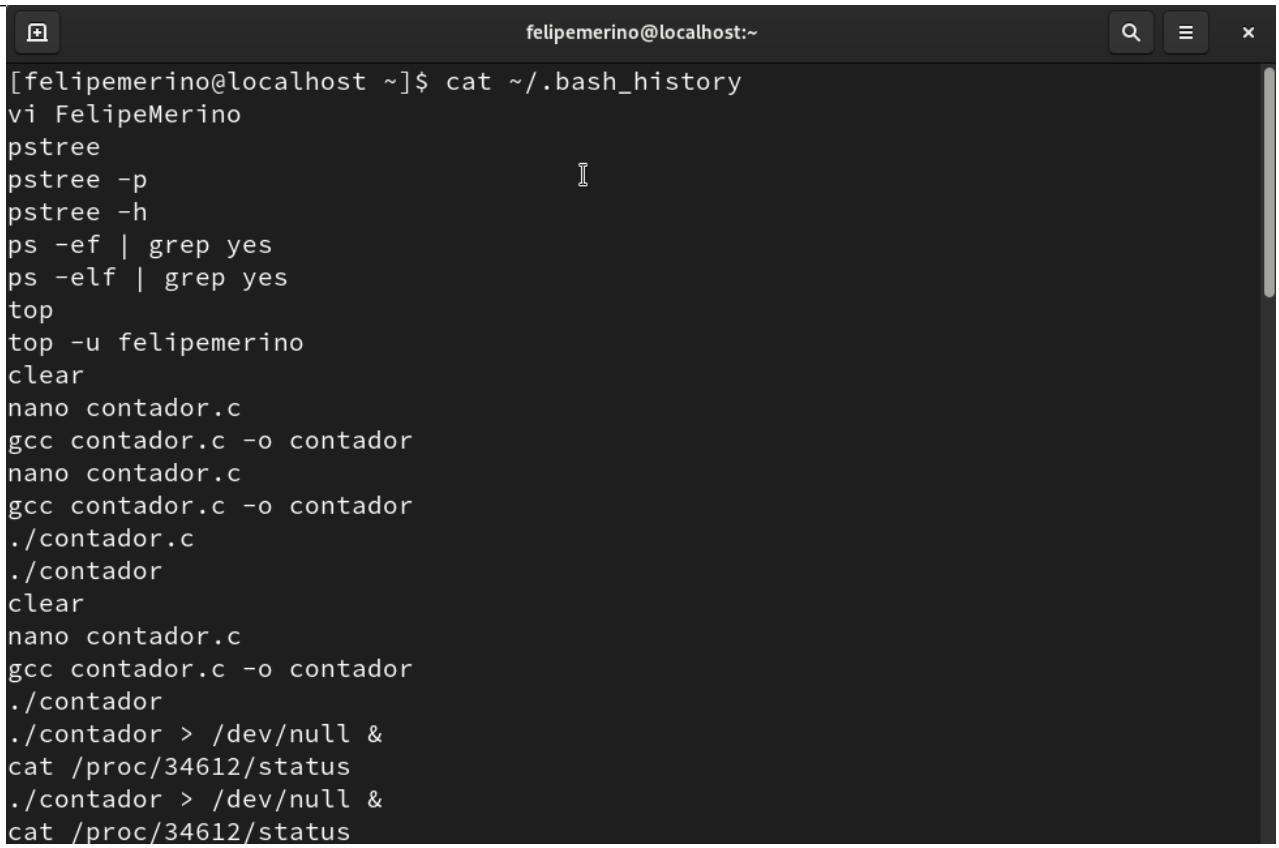
7. Con el comando *wall,* coloque un mensaje en todas las pantallas de los usuarios logueados. Abrir varias terminales virtuales con Ctrl + Alt + Fn para comprobar la presentación del mensaje. Para salir del comando usar “Ctrl +D”.

```
Mensaje de difusión general (broadcast) de felipemerino@localhost.localdomain

asdas
dsad
asd
asd
as
d
a

Mensaje de difusión general (broadcast) de felipemerino@localhost.localdomain

asdasd
asdasd
asdasd
fd
```

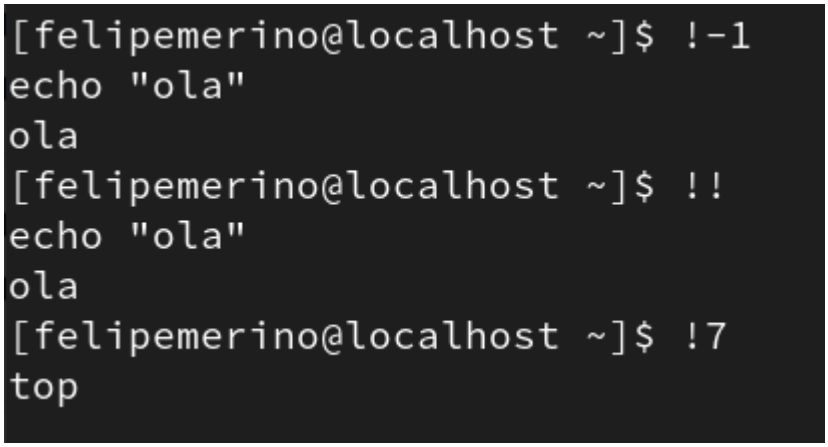
8. Verificar el historial de comandos con *history*.



9. Verificar el contenido del fichero *~/.bash_history* con el comando *cat*. Verificar que se encuentra logueado con el usuario root. (Para utilizar el símbolo ~ pruebe con Alt Gr + 4 o, si posee teclado numérico, también puede utilizar Alt Gr + 9 desactivando previamente el teclado numérico).



10. Verificar el resultado obtenido con la ejecución de los siguientes comandos. Explicar su uso.



```
top - 01:09:42 up 43 min,  3 users,  load average: 0,10, 0,29, 0,22
Tasks: 231 total,   1 running, 228 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0,8 us,  0,2 sy,  0,0 ni, 98,8 id,  0,0 wa,  0,2 hi,  0,0 si,  0,0 st
MiB Mem :   6952,6 total,   3942,7 free,   2002,5 used,   1292,8 buff/cache
MiB Swap:   3616,0 total,   3616,0 free,      0,0 used.   4950,1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3852 felipem+  20   0 2902692 218452 127512 S   1,3   3,1   0:10.85 Isolate+
   2017 felipem+  20   0 4667960 362972 143800 S   1,0   5,1   1:13.25 gnome-s+
   2056 felipem+  20   0  527128  11880   6980 S   0,3   0,2   0:02.07 ibus-da+
   3511 root      20   0       0      0      0 I   0,3   0,0   0:00.53 kworker+
   3570 felipem+  20   0  764828  51864  40348 S   0,3   0,7   0:02.22 gnome-t+
   3699 felipem+  20   0 3617004 466112 186204 S   0,3   6,5   0:46.06 firefox
   3848 felipem+  20   0 3161356 291500 118492 S   0,3   4,1   0:22.69 Isolate+
      1 root      20   0  174716  17332  11148 S   0,0   0,2   0:55.13 systemd
      2 root      20   0       0      0      0 S   0,0   0,0   0:00.02 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+
      9 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
     10 root      20   0       0      0      0 I   0,0   0,0   0:00.00 kworker+
     11 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
```

!-1: ejecuta el ultimo comando

¡!: Ejecuta el ultimo comando

¡7: Ejecuta el comando de la línea 7

3.7 Herramientas para visualización de ficheros

- ***cat*** → De *catenate*, permite leer datos de ficheros y mostrar sus contenidos. Es la forma más sencilla. Cat es uno de los comandos más utilizados en Linux.
- ***more*** → Permite leer el contenido de un fichero o comando y visualizarlo por páginas (para avanzar pulsar la barra espaciadora).
- ***less*** → Permite leer el contenido de un fichero línea a línea, no se puede manipular ni editar el texto. Con este comando se puede subir y/o bajar por el texto.
- ***file*** → Determina el tipo de un archivo e imprime en pantalla el resultado.

3.8 Herramientas para edición de ficheros

Para editar un fichero el usuario debe poseer permisos de escritura.

vi → Es el editor por defecto en todos los sistemas Linux. Existen tres modos de operación en vi:

- **Modo comando:** Este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- **Modo edición:** Este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- **Modo línea o ex:** Se escriben comandos en la última línea al final de la pantalla.

3.9 Otros editores

Una distribución Linux típica incluye un gran número de editores.

Editores en modo texto:
- pico
- vim
- Emacs

Editores en modo gráfico:
- kedit, kwrite
- gedit

Editores hexadecimales: permiten modificar archivos que no son de texto.
- Khexedite

3.10 INICIO DE VI:

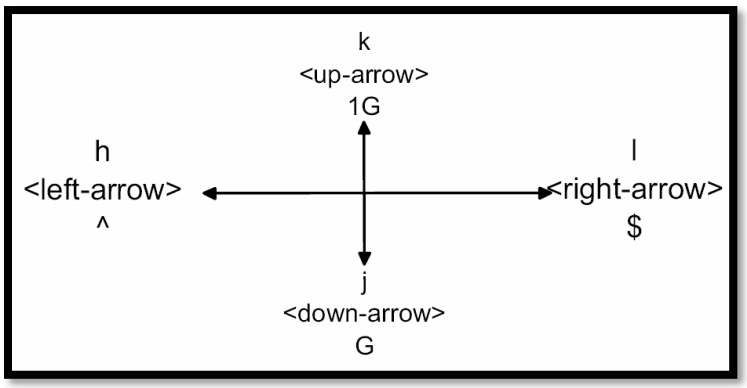
Abrir un archivo específico para su edición ($vi NombreArchivo):
$vi
$vi suNombre

Nota: Si el archivo no existe, lo crea. Presione la tecla ***i*** para poder introducir texto, y escriba algunas líneas. Utilice **ESC** para regresar al modo comando

MODO COMANDO:

El editor vi, al igual que UNIX, diferencia entre mayúsculas y minúsculas. Después de abrir un fichero con el comando vi, digite los siguientes comandos (digite las letras) y observe su funcionamiento:

h
j
k
l
^
$
1G
G



<up arrow>
<down arrow>
<left arrow>
<right arrow>
ZZ

Figura 3. Resumen de los comandos para navegar dentro de vi

Resumen:

<left arrow> o ***h***	un caracter a la izquierda
<right arrow> o ***l***	un caracter a la derecha
^	moverse al inicio de la línea
$	moverse al final de la línea
<up arrow> o ***k***	una línea arriba
<down arrow> o ***j***	una línea abajo
1G	ir a la primera línea
G	ir a la última línea

Nota: Para usar estos comandos asegurarse de estar en modo comando.

MODO EDICIÓN:

Para cambiar de modo comando a modo edición, dentro de algún fichero digite:

I (i mayúscula, o tecla INSERT):

Tomar en cuenta que se muestra ***-- INSERT --*** o ***-- INSERTAR --*** en modo Edición:

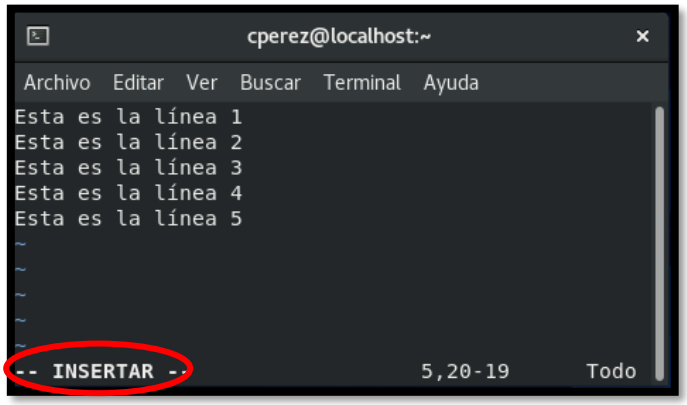


Figura 4. Texto de ayuda para saber si se está dentro del Modo Edición

Luego pruebe el funcionamiento de los siguientes comandos:

- ***i***

- ***A***
- ***a***
- ***O***
- ***o***

Para insertar texto al inicio de la línea	***I (letra i mayúscula)***
Para insertar texto antes del cursor	***i***
Para agregar texto después del cursor	***a***
Para agregar texto al final de la línea	***A***
Para regresar al modo de comando	***<ESC>***

Dentro del Modo Comando, compruebe los siguientes comandos.

- ***x***
- ***dd***
- ***2dd***
- ***ndd (siendo n un número entero cualquiera)***
- ***D***
- ***dw***

Es importante destacar que todo lo que se borra queda almacenado en un buffer (área temporal de memoria), de modo que, si se borró algo por error, puede volver a escribirse (si se hace antes de realizar otros cambios, es decir, inmediatamente luego de eliminar el texto por error). Esto se hace simplemente ejecutando el comando ***p***.

Cortar y pegar: Esto implica mover partes del archivo de un lugar a otro del mismo. Para esto se debe:

- Cortar el texto que se desea mover utilizando alguno de los comandos usados para borrar texto (***dd***, ***2dd***, etc).
- Mover el cursor (con alguno de los comandos utilizados para desplazar el cursor en el texto) hasta el lugar donde se desee pegar el texto.
- Pegar el texto con el comando ***p***.

Copiar y pegar: Esta operación difiere de la anterior. En este caso lo que se hace es repetir partes del texto en otro lugar del archivo. Para esto se debe:

- Utilizar el comando ***yy***, cuya función es copiar la línea donde se encuentra situado el cursor. ***8yy*** es una variante que copia múltiples líneas, en este caso 8.
- Mover el cursor (con alguno de los comandos utilizados para desplazar el cursor en el texto) hasta el lugar donde se desee pegar el texto.
- Pegar el texto con el comando ***p***.

Deshacer cambios: Se puede deshacer el último cambio realizado, utilizando el comando u.

Para cambiar de Modo Edición a Modo Comando digitar la tecla **Esc**.

MODO LÍNEA O EX:

Para ingresar al Modo Línea desde el Modo Comando, se debe utilizar alguna de las siguientes teclas:

- ***:***
- ***/***
- ***?***

Para volver al Modo Comando desde el Modo Línea, se debe digitar la tecla **ENTER** (al finalizar el comando) o la tecla **ESC** (que interrumpe el comando).

Dentro de algún fichero abierto con el comando vi, digite:

- ***/textoABuscar***
- ***?textoABuscar***
- ***n***
- ***:q***
- ***:q!***
- ***:w***
- ***:w!***
- ***:w archivo1***
- ***:wq***
- ***:x***

Resumen de Comandos:

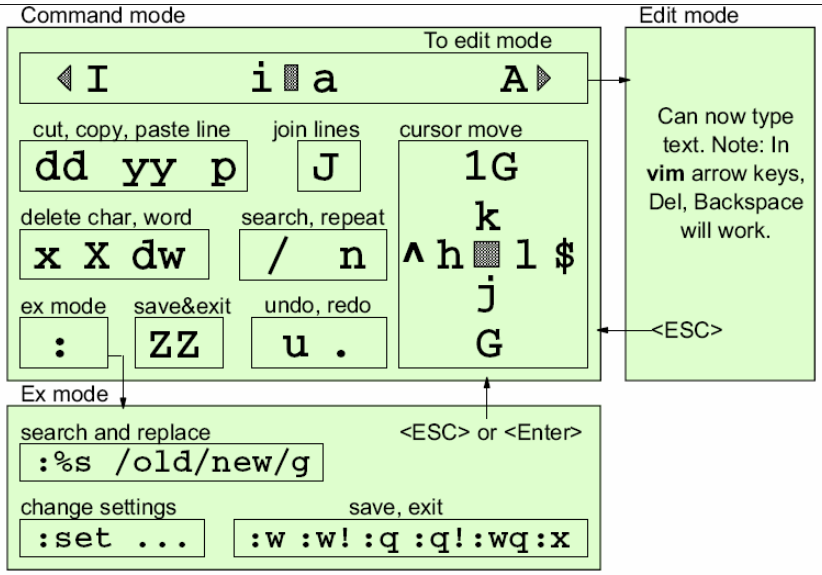


Figura 5. Resumen de comandos del editor vi

3.11. Realizar la siguiente figura con el comando vi dentro de un fichero que tendrá su nombre:

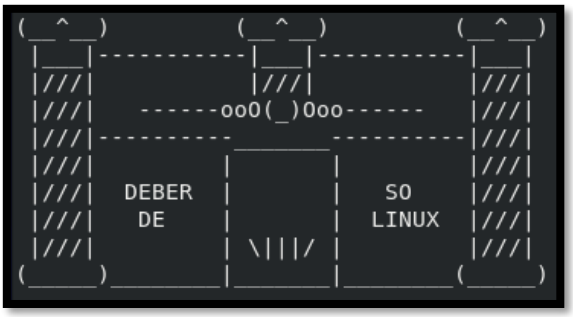
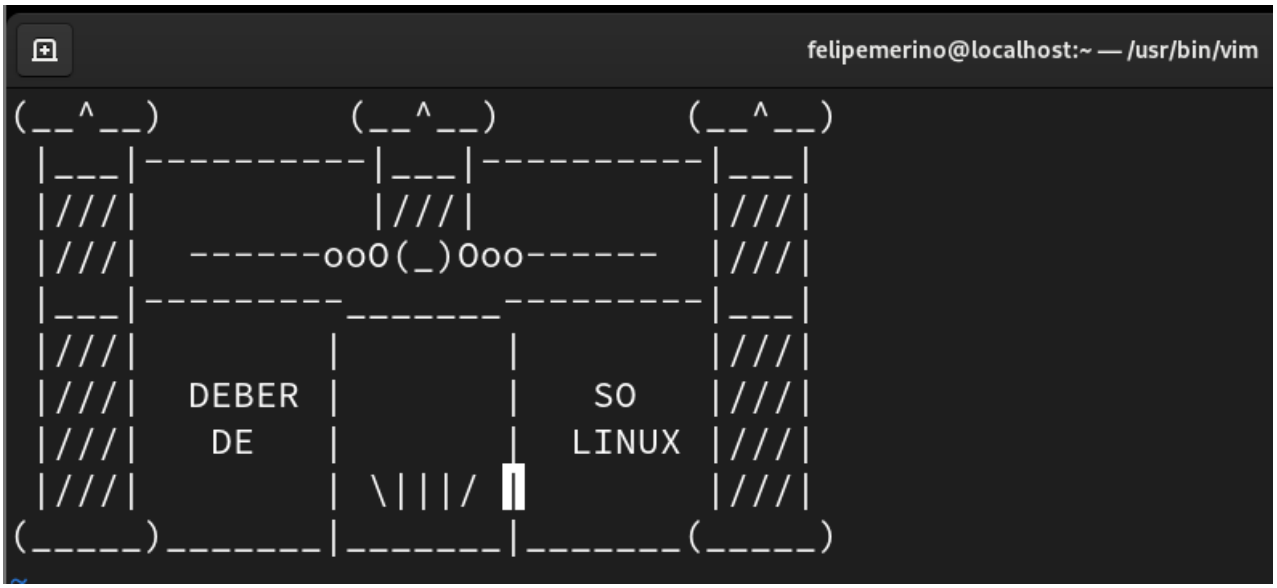


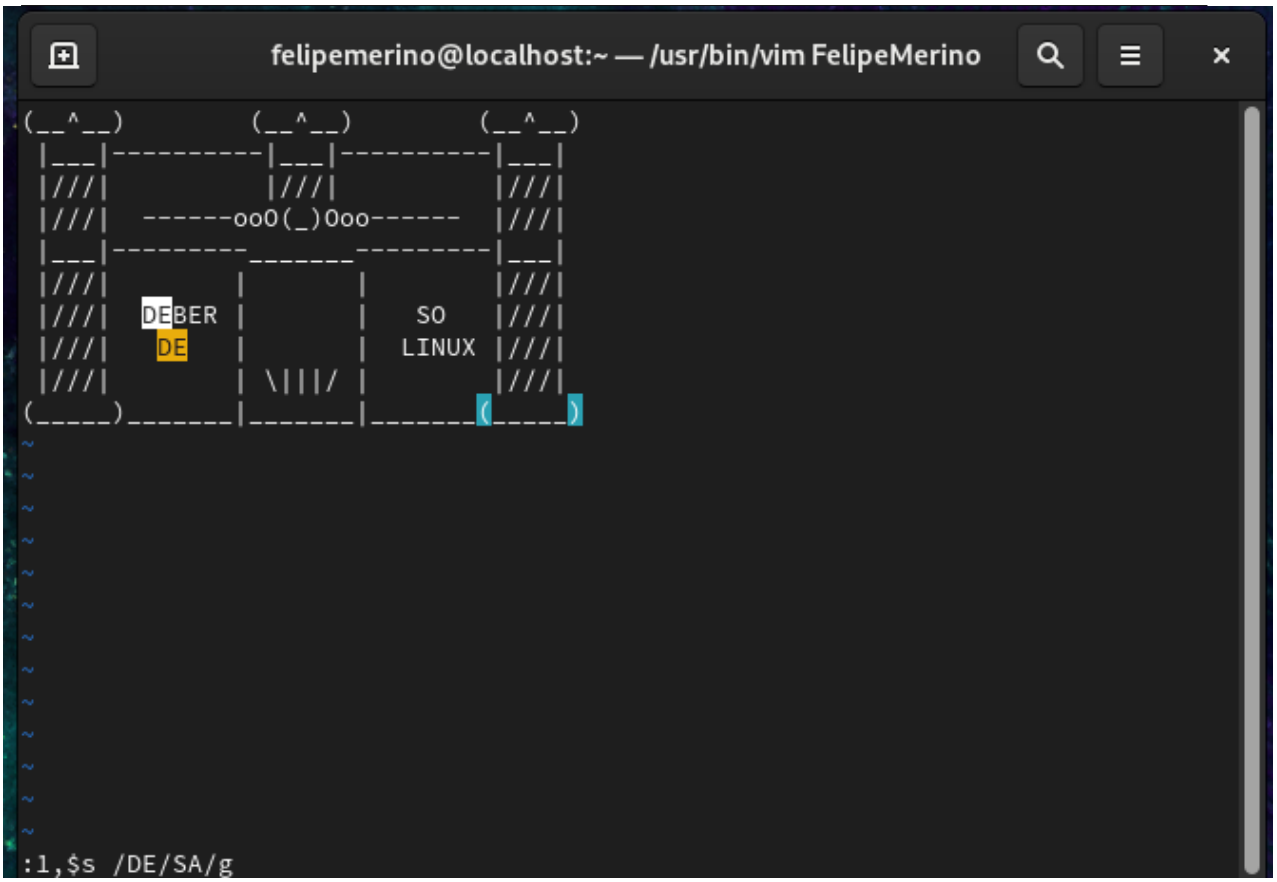
Figura 6. Figura creada utilizando símbolos y comandos en vi

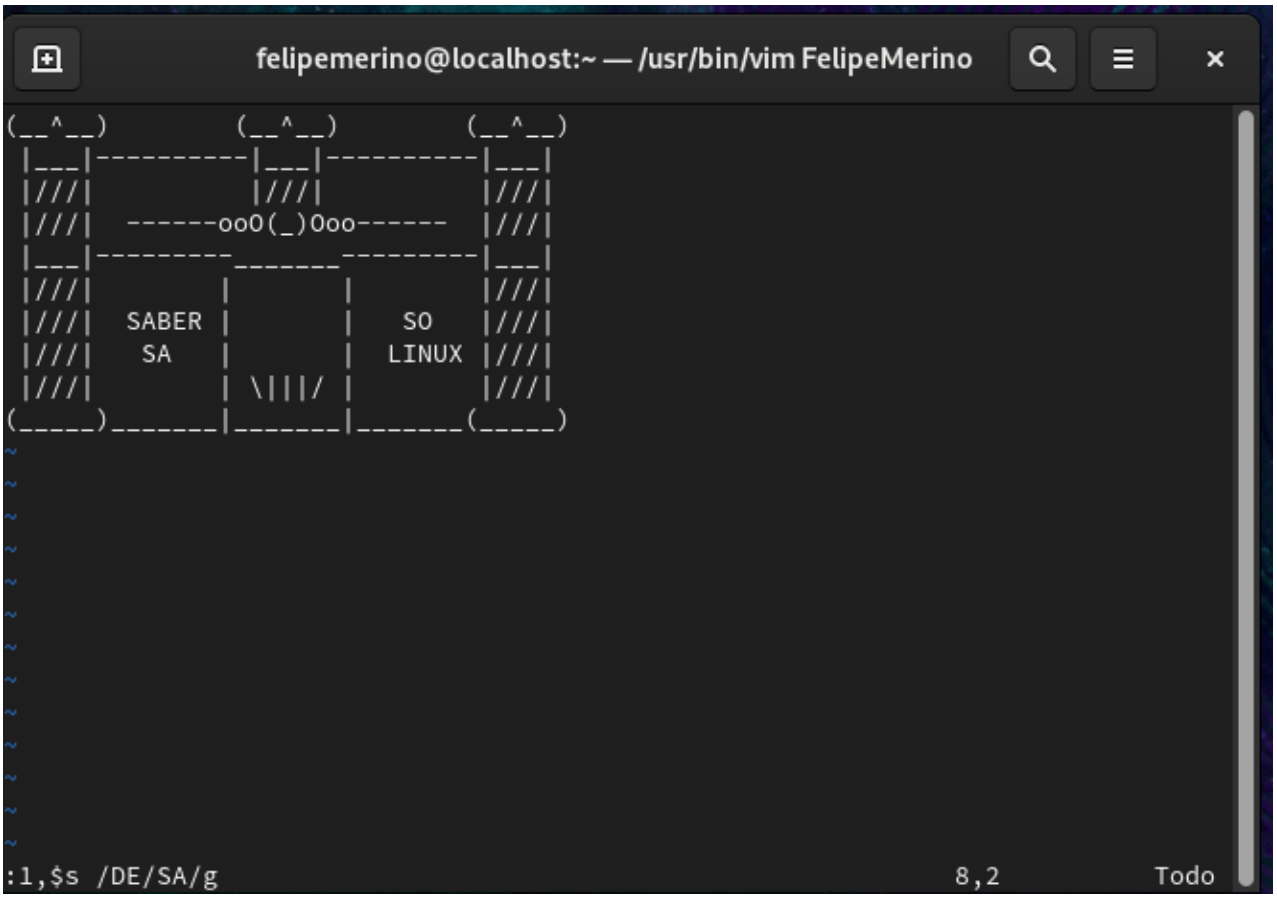
Imagen dibujada:



3.12 Ejecutar el siguiente comando, verificar qué sucede y explicar de manera detallada su uso. El comando inicia con el símbolo “dos puntos” (:) para acceder al Modo Línea. Utilizar todos los demás símbolos indicados dentro del comando como la “coma” (,) y el símbolo de “dólar” ($).

:1,$s /DE/SA/g


felipemerino@localhost:~ — /usr/bin/vim FelipeMerino
DEBER
DE
SO
LINUX
:1,$s /DE/SA/g



3.13 Verificar los procesos que se encuentran ejecutando:

$ ps

Figura 7. Ejecución del comando ps

$ ps aux

```
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.4  0.1  19356  1536 ?        Ss   06:40   0:04 /sbin/init
root          2  0.0  0.0      0     0 ?        S    06:40   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    06:40   0:00 [migration/0]
root          4  0.0  0.0      0     0 ?        S    06:40   0:00 [ksoftirqd/0]
root          5  0.0  0.0      0     0 ?        S    06:40   0:00 [stopper/0]
root          6  0.0  0.0      0     0 ?        S    06:40   0:00 [watchdog/0]
root          7  0.1  0.0      0     0 ?        S    06:40   0:01 [events/0]
root          8  0.0  0.0      0     0 ?        S    06:40   0:00 [events/0]
root          9  0.0  0.0      0     0 ?        S    06:40   0:00 [events_long/0]
root         10  0.0  0.0      0     0 ?        S    06:40   0:00 [events_power_ef]
root         11  0.0  0.0      0     0 ?        S    06:40   0:00 [cgroup]
root         12  0.0  0.0      0     0 ?        S    06:40   0:00 [khelper]
root         13  0.0  0.0      0     0 ?        S    06:40   0:00 [netns]
root         14  0.0  0.0      0     0 ?        S    06:40   0:00 [async/mgr]
root         15  0.0  0.0      0     0 ?        S    06:40   0:00 [pm]
root         16  0.0  0.0      0     0 ?        S    06:40   0:00 [sync_supers]
root         17  0.0  0.0      0     0 ?        S    06:40   0:00 [bdi-default]
root         18  0.0  0.0      0     0 ?        S    06:40   0:00 [kintegrityd/0]
root         19  0.0  0.0      0     0 ?        S    06:40   0:00 [kblockd/0]
root         20  0.0  0.0      0     0 ?        S    06:40   0:00 [kacpid]
root         21  0.0  0.0      0     0 ?        S    06:40   0:00 [kacpi_notify]
root         22  0.0  0.0      0     0 ?        S    06:40   0:00 [kacpi_hotplug]
root         23  0.0  0.0      0     0 ?        S    06:40   0:00 [ata_aux]
root         24  0.0  0.0      0     0 ?        S    06:40   0:00 [ata_sff/0]
root         25  0.0  0.0      0     0 ?        S    06:40   0:00 [ksuspend_usbd]
root         26  0.0  0.0      0     0 ?        S    06:40   0:00 [khubd]
root         27  0.0  0.0      0     0 ?        S    06:40   0:00 [kseriod]
root         28  0.0  0.0      0     0 ?        S    06:40   0:00 [md/0]
root         29  0.0  0.0      0     0 ?        S    06:40   0:00 [md_misc/0]
root         30  0.0  0.0      0     0 ?        S    06:40   0:00 [linkwatch]
```

Figura 8. Ejecución del comando ps aux

$ ps l

```
F   UID   PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
0   501  5427  5419  20   0 108336  1820 wait   Ss   pts/0      0:00 bash
0   501  5965  5427  20   0 108092  1568 signal T    pts/0      0:00 nano
0   501  5990  5419  20   0 108336  1808 n_tty_ Ss+  pts/1      0:00 bash
0   501  6524  5427  20   0 108128   996 -      R+   pts/0      0:00 ps l
```

Figura 9. Ejecución del comando ps l

Cada una de las columnas que aparecen al ejecutar el comando anterior tiene su propio significado:

F: Son los indicadores asociados al proceso
UID: Identificación del usuario propietario del proceso
***PID*:** ID de un proceso
PPID: ID proceso padre
PRI: Prioridad del proceso

NI: Valor de bondad. Mientras más elevado, menor es la prioridad.
VSZ: Tamaño de la memoria virtual del proceso en KB.
RSS: Tamaño de la memoria física usada en KB.
WCHAN: Para los procesos que esperan o están dormidos, enumera el evento que espera.
TTY: Terminal virtual
TIME: tiempo de ejecución
CMD/COMMAND: Comando
STAT: Estado del proceso. Un proceso puede tener cualquiera de los siguientes estados:

- R – Ready/Ejecutando.
- D – Interrumpido
- S – Suspendido
- s – Es el proceso líder de la sesión
- T – Detenido
- Z – Zombie
- N – Tiene una prioridad menor que lo normal.
- < – Tiene una prioridad mayor que lo normal.
- L – Tiene páginas bloqueadas en memoria
- l – Es multi-hilo.
- + – Está en el grupo de procesos en foreground

3.14 En GNU/Linux se puede crear una estructura jerárquica de procesos (relación padre-hijo).

$ ps -o ppid,pid,cmd | grep ps

```
[operativos@localhost ~]$ ps -o ppid,pid,cmd | grep ps
   4618    4841 ps -o ppid,pid,cmd
   4618    4842 grep --color=auto ps
```

Figura 10. Ejecución del comando ps -o ppid,pid,cmd | grep ps

```
[operativos@localhost ~]$ echo $$
2963
[operativos@localhost ~]$ bash
[operativos@localhost ~]$ echo $$
3507
[operativos@localhost ~]$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    2963    2958  0  80   0 -  6165 -      pts/0    00:00:00 bash
0 S  1000    3507    2963  1  80   0 -  6167 -      pts/0    00:00:00 bash
0 R  1000    3551    3507  0  80   0 - 11351 -      pts/0    00:00:00 ps
```

Figura 11. Verificación de los procesos activos mediante el comando ps -l

3.15 Procesos en segundo plano

```
[cloudera@quickstart ~]$ ps -fe|more
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 12:39 ?        00:00:02 /sbin/init
root         2     0  0 12:39 ?        00:00:00 [kthreadd]
root         3     2  0 12:39 ?        00:00:00 [migration/0]
root         4     2  0 12:39 ?        00:00:00 [ksoftirqd/0]
root         5     2  0 12:39 ?        00:00:00 [migration/0]
root         6     2  0 12:39 ?        00:00:00 [watchdog/0]
root         7     2  0 12:39 ?        00:00:01 [events/0]
root         8     2  0 12:39 ?        00:00:00 [cgroup]
root         9     2  0 12:39 ?        00:00:00 [khelper]
root        10     2  0 12:39 ?        00:00:00 [netns]
root        11     2  0 12:39 ?        00:00:00 [async/mgr]
root        12     2  0 12:39 ?        00:00:00 [pm]
```

Figura 12. Ejecución del comando ps -fe|more

3.16 Mediante el comando *pstree* determinar el orden jerárquico de los procesos que se están ejecutando actualmente en su terminal bash. Ventaja: El árbol muestra

los procesos en una relación padre-hijo.

pstree

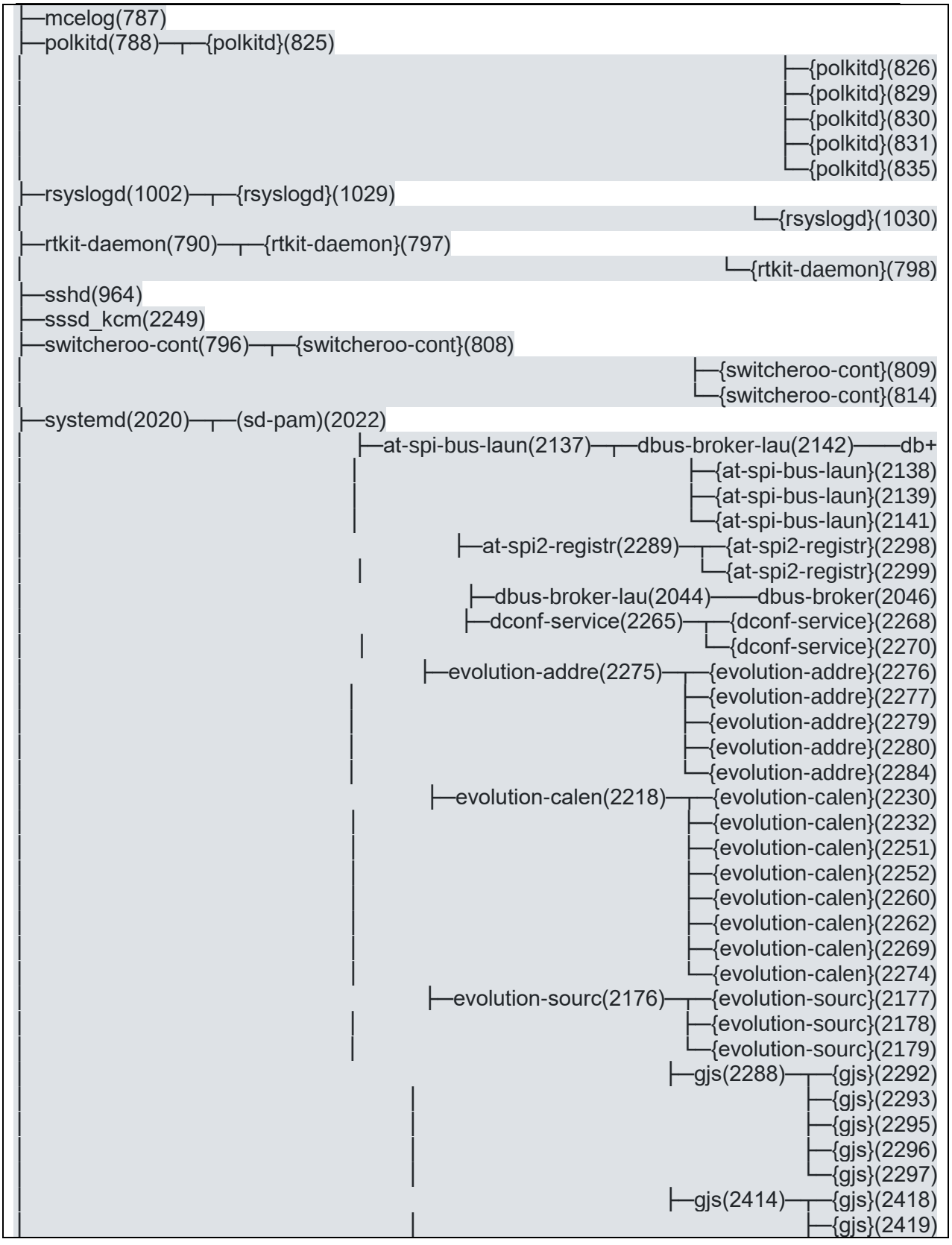
```
[felipemerino@localhost ~]$ pstree
systemd─┬─ModemManager───3*[{ModemManager}]
 ├─NetworkManager───2*[{NetworkManager}]
 ├─accounts-daemon───3*[{accounts-daemon}]
 ├─alsactl
 ├─atd
 ├─auditd─┬─sedispatch
 │        └─2*[{auditd}]
 ├─avahi-daemon───avahi-daemon
 ├─chronyd
 ├─colord───3*[{colord}]
 ├─crond
 ├─cupsd
 ├─dbus-broker-lau───dbus-broker
 ├─firewalld───{firewalld}
 ├─fwupd───4*[{fwupd}]
 ├─gdm─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b───3*[{gnome+
 │     │                 │                 └─2*[{gdm-wayland-ses}]
 │     │                 └─2*[{gdm-session-wor}]
 │     └─2*[{gdm}]
 ├─geoclue───3*[{geoclue}]
 ├─gnome-keyring-d───3*[{gnome-keyring-d}]
 ├─irqbalance───{irqbalance}
 ├─lsmd
 ├─mcelog
 ├─polkitd───6*[{polkitd}]
 ├─rsyslogd───2*[{rsyslogd}]
 ├─rtkit-daemon───2*[{rtkit-daemon}]
 ├─sshd
 ├─sssd_kcm
 ├─switcheroo-cont───3*[{switcheroo-cont}]
 ├─systemd─┬─(sd-pam)
 │         ├─at-spi-bus-laun─┬─dbus-broker-lau───dbus-broker
 │         │                 └─3*[{at-spi-bus-laun}]
 │         ├─at-spi2-registr───2*[{at-spi2-registr}]
 │         ├─dbus-broker-lau───dbus-broker
 │         ├─dconf-service───2*[{dconf-service}]
 │         ├─evolution-addre───5*[{evolution-addre}]
 │         ├─evolution-calen───8*[{evolution-calen}]
 │         ├─evolution-sourc───3*[{evolution-sourc}]
 │         ├─2*[gjs───5*[{gjs}]]
 │         ├─gnome-session-b─┬─evolution-alarm───5*[{evolution-alarm}]
 │         │                 ├─gnome-software───6*[{gnome-software}]
 │         │                 ├─gsd-disk-utilit───2*[{gsd-disk-utilit}]
 │         │                 └─3*[{gnome-session-b}]
 │         ├─gnome-session-c───{gnome-session-c}
```

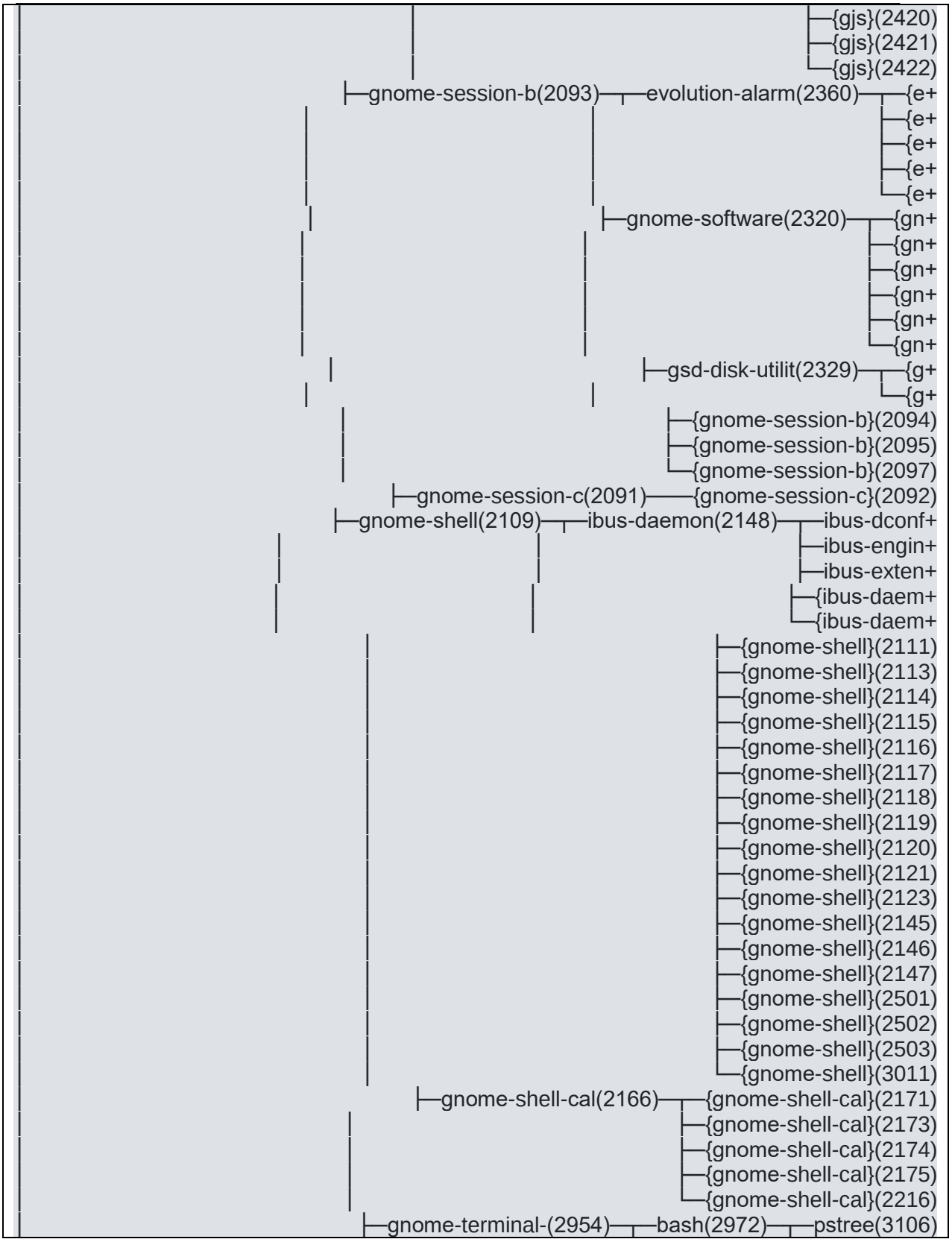
```
│                    ├─gnome-shell─┬─ibus-daemon─┬─ibus-dconf───3*[{ibus-dconf}]
│                    │             │             ├─ibus-engine-sim───2*[{ibus-engi+
│                    │             │             ├─ibus-extension-───3*[{ibus-exte+
│                    │             │             └─2*[{ibus-daemon}]
│                    │             └─18*[{gnome-shell}]
│                    ├─gnome-shell-cal───5*[{gnome-shell-cal}]
│                    ├─gnome-terminal-─┬─bash─┬─pstree
│                    │                 │      └─3*[vim]
│                    │                 └─3*[{gnome-terminal-}]
│                    ├─goa-daemon───3*[{goa-daemon}]
│                    ├─goa-identity-se───3*[{goa-identity-se}]
│                    ├─gsd-a11y-settin───3*[{gsd-a11y-settin}]
│                    ├─gsd-color───3*[{gsd-color}]
│                    ├─gsd-datetime───3*[{gsd-datetime}]
│                    ├─gsd-housekeepin───3*[{gsd-housekeepin}]
│                    ├─gsd-keyboard───3*[{gsd-keyboard}]
│                    ├─gsd-media-keys───3*[{gsd-media-keys}]
│                    ├─gsd-power───3*[{gsd-power}]
│                    ├─gsd-print-notif───2*[{gsd-print-notif}]
│                    ├─gsd-printer───2*[{gsd-printer}]
│                    ├─gsd-rfkill───2*[{gsd-rfkill}]
│                    ├─gsd-screensaver───2*[{gsd-screensaver}]
│                    ├─gsd-sharing───3*[{gsd-sharing}]
│                    ├─gsd-smartcard───5*[{gsd-smartcard}]
│                    ├─gsd-sound───3*[{gsd-sound}]
│                    ├─gsd-usb-protect───3*[{gsd-usb-protect}]
│                    ├─gsd-wacom───3*[{gsd-wacom}]
│                    ├─gvfs-goa-volume───2*[{gvfs-goa-volume}]
│                    ├─gvfs-gphoto2-vo───2*[{gvfs-gphoto2-vo}]
│                    ├─gvfs-mtp-volume───2*[{gvfs-mtp-volume}]
│                    ├─gvfs-udisks2-vo───3*[{gvfs-udisks2-vo}]
│                    ├─gvfsd─┬─gvfsd-trash───2*[{gvfsd-trash}]
│                    │       └─2*[{gvfsd}]
│                    ├─gvfsd-fuse───5*[{gvfsd-fuse}]
│                    ├─gvfsd-metadata───2*[{gvfsd-metadata}]
│                    ├─ibus-portal───2*[{ibus-portal}]
│                    ├─pipewire───2*[{pipewire}]
│                    ├─pipewire-pulse───2*[{pipewire-pulse}]
│                    ├─tracker-miner-f───5*[{tracker-miner-f}]
│                    ├─wireplumber───4*[{wireplumber}]
│                    ├─xdg-desktop-por───5*[{xdg-desktop-por}]
│                    ├─2*[xdg-desktop-por───3*[{xdg-desktop-por}]]
│                    ├─xdg-document-po─┬─fusermount
│                    │                 └─5*[{xdg-document-po}]
│                    └─xdg-permission-───2*[{xdg-permission-}]
├─systemd-journal
├─systemd-logind
├─systemd-udevd
├─tuned───3*[{tuned}]
```

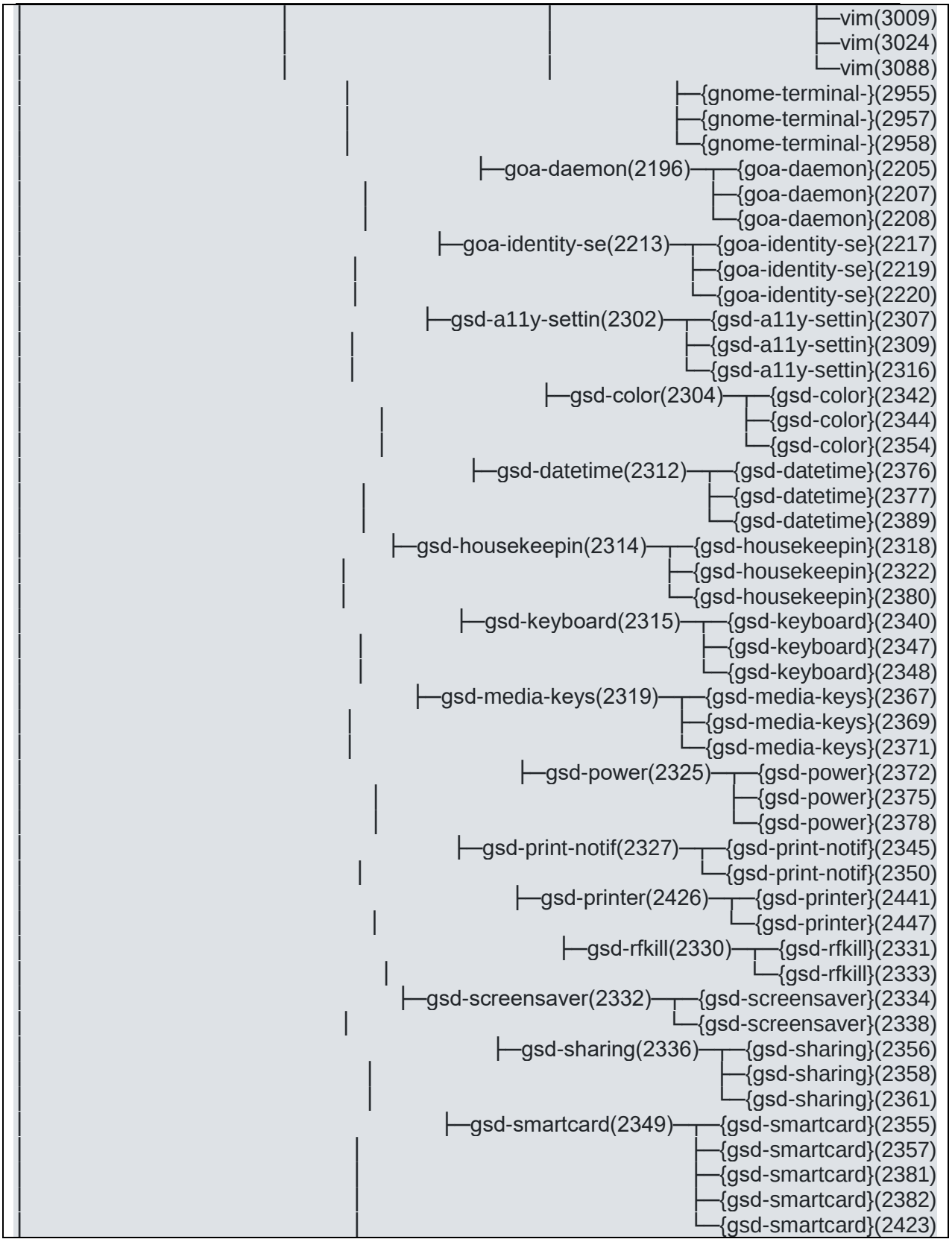
```
├─tuned-ppd───3*[{tuned-ppd}]
├─udisksd───4*[{udisksd}]
├─upowerd───2*[{upowerd}]
└─wpa_supplicant
```

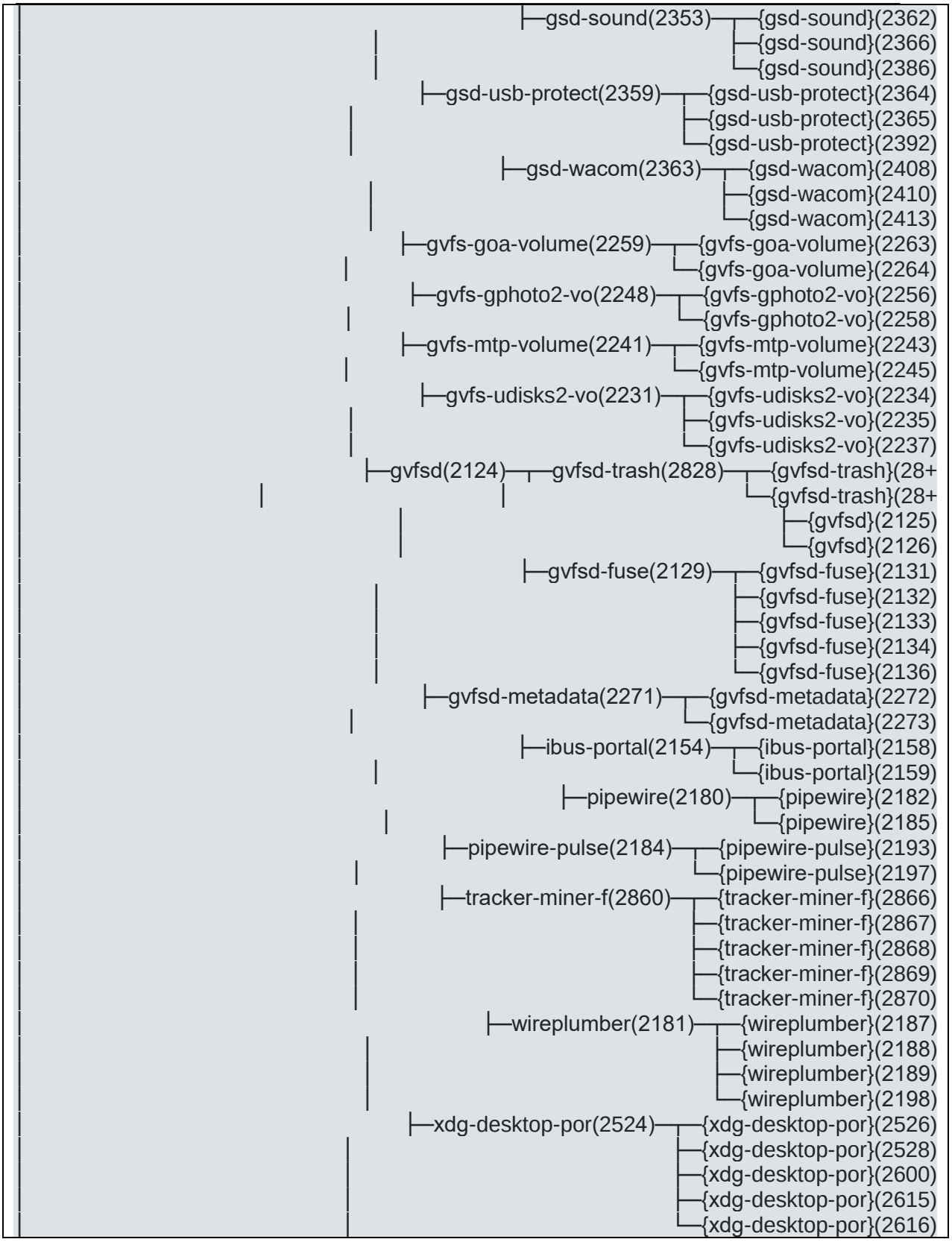
pstree –p

```
[felipemerino@localhost ~]$ pstree -p
systemd(1)─┬─ModemManager(837)─┬─{ModemManager}(848)
           │                   ├─{ModemManager}(849)
           │                   └─{ModemManager}(852)
           ├─NetworkManager(953)─┬─{NetworkManager}(954)
           │                     └─{NetworkManager}(955)
           ├─accounts-daemon(794)─┬─{accounts-daemon}(802)
           │                      ├─{accounts-daemon}(803)
           │                      └─{accounts-daemon}(805)
           ├─alsactl(821)
           ├─atd(977)
           ├─auditd(752)─┬─sedispatch(754)
           │             ├─{auditd}(753)
           │             └─{auditd}(755)
           ├─avahi-daemon(782)───avahi-daemon(815)
           ├─chronyd(813)
           ├─colord(1854)─┬─{colord}(1858)
           │              ├─{colord}(1859)
           │              └─{colord}(1861)
           ├─crond(978)
           ├─cupsd(963)
           ├─dbus-broker-lau(780)───dbus-broker(781)
           ├─firewalld(839)───{firewalld}(952)
           ├─fwupd(2621)─┬─{fwupd}(2628)
           │             ├─{fwupd}(2629)
           │             ├─{fwupd}(2630)
           │             └─{fwupd}(2631)
           ├─gdm(980)─┬─gdm-session-wor(2013)─┬─gdm-wayland-ses(2042)─┬─gnome-s+
           │          │                       │                       ├─{gdm-wa+
           │          │                       │                       └─{gdm-wa+
           │          │                       ├─{gdm-session-wor}(2014)
           │          │                       └─{gdm-session-wor}(2015)
           │          ├─{gdm}(992)
           │          └─{gdm}(993)
           ├─geoclue(2201)─┬─{geoclue}(2224)
           │               ├─{geoclue}(2225)
           │               └─{geoclue}(2228)
           ├─gnome-keyring-d(2038)─┬─{gnome-keyring-d}(2039)
           │                       ├─{gnome-keyring-d}(2040)
           │                       └─{gnome-keyring-d}(2104)
           ├─irqbalance(785)───{irqbalance}(807)
           ├─lsmd(786)
```


├─mcelog(787)
├─polkitd(788)─┬─{polkitd}(825)
├─{polkitd}(826)
├─{polkitd}(829)
├─{polkitd}(830)
├─{polkitd}(831)
└─{polkitd}(835)
├─rsyslogd(1002)─┬─{rsyslogd}(1029)
└─{rsyslogd}(1030)
├─rtkit-daemon(790)─┬─{rtkit-daemon}(797)
└─{rtkit-daemon}(798)
├─sshd(964)
├─sssd_kcm(2249)
├─switcheroo-cont(796)─┬─{switcheroo-cont}(808)
├─{switcheroo-cont}(809)
└─{switcheroo-cont}(814)
├─systemd(2020)─┬─(sd-pam)(2022)
├─at-spi-bus-laun(2137)─┬─dbus-broker-lau(2142)──db+
├─{at-spi-bus-laun}(2138)
├─{at-spi-bus-laun}(2139)
└─{at-spi-bus-laun}(2141)
├─at-spi2-registr(2289)─┬─{at-spi2-registr}(2298)
└─{at-spi2-registr}(2299)
├─dbus-broker-lau(2044)──dbus-broker(2046)
├─dconf-service(2265)─┬─{dconf-service}(2268)
└─{dconf-service}(2270)
├─evolution-addre(2275)─┬─{evolution-addre}(2276)
├─{evolution-addre}(2277)
├─{evolution-addre}(2279)
├─{evolution-addre}(2280)
└─{evolution-addre}(2284)
├─evolution-calen(2218)─┬─{evolution-calen}(2230)
├─{evolution-calen}(2232)
├─{evolution-calen}(2251)
├─{evolution-calen}(2252)
├─{evolution-calen}(2260)
├─{evolution-calen}(2262)
├─{evolution-calen}(2269)
└─{evolution-calen}(2274)
├─evolution-sourc(2176)─┬─{evolution-sourc}(2177)
├─{evolution-sourc}(2178)
└─{evolution-sourc}(2179)
├─gjs(2288)─┬─{gjs}(2292)
├─{gjs}(2293)
├─{gjs}(2295)
├─{gjs}(2296)
└─{gjs}(2297)
├─gjs(2414)─┬─{gjs}(2418)
├─{gjs}(2419)


├─{gjs}(2420)
├─{gjs}(2421)
└─{gjs}(2422)
├─gnome-session-b(2093)─┬─evolution-alarm(2360)─┬─{e+
├─{e+
├─{e+
├─{e+
└─{e+
├─gnome-software(2320)─┬─{gn+
├─{gn+
├─{gn+
├─{gn+
├─{gn+
└─{gn+
├─gsd-disk-utilit(2329)─┬─{g+
└─{g+
├─{gnome-session-b}(2094)
├─{gnome-session-b}(2095)
└─{gnome-session-b}(2097)
├─gnome-session-c(2091)───{gnome-session-c}(2092)
├─gnome-shell(2109)─┬─ibus-daemon(2148)─┬─ibus-dconf+
├─ibus-engin+
├─ibus-exten+
├─{ibus-daem+
└─{ibus-daem+
├─{gnome-shell}(2111)
├─{gnome-shell}(2113)
├─{gnome-shell}(2114)
├─{gnome-shell}(2115)
├─{gnome-shell}(2116)
├─{gnome-shell}(2117)
├─{gnome-shell}(2118)
├─{gnome-shell}(2119)
├─{gnome-shell}(2120)
├─{gnome-shell}(2121)
├─{gnome-shell}(2123)
├─{gnome-shell}(2145)
├─{gnome-shell}(2146)
├─{gnome-shell}(2147)
├─{gnome-shell}(2501)
├─{gnome-shell}(2502)
├─{gnome-shell}(2503)
└─{gnome-shell}(3011)
├─gnome-shell-cal(2166)─┬─{gnome-shell-cal}(2171)
├─{gnome-shell-cal}(2173)
├─{gnome-shell-cal}(2174)
├─{gnome-shell-cal}(2175)
└─{gnome-shell-cal}(2216)
├─gnome-terminal-(2954)─┬─bash(2972)─┬─pstree(3106)


vim(3009)
vim(3024)
vim(3088)
{gnome-terminal-}(2955)
{gnome-terminal-}(2957)
{gnome-terminal-}(2958)
goa-daemon(2196)
{goa-daemon}(2205)
{goa-daemon}(2207)
{goa-daemon}(2208)
goa-identity-se(2213)
{goa-identity-se}(2217)
{goa-identity-se}(2219)
{goa-identity-se}(2220)
gsd-a11y-settin(2302)
{gsd-a11y-settin}(2307)
{gsd-a11y-settin}(2309)
{gsd-a11y-settin}(2316)
gsd-color(2304)
{gsd-color}(2342)
{gsd-color}(2344)
{gsd-color}(2354)
gsd-datetime(2312)
{gsd-datetime}(2376)
{gsd-datetime}(2377)
{gsd-datetime}(2389)
gsd-housekeepin(2314)
{gsd-housekeepin}(2318)
{gsd-housekeepin}(2322)
{gsd-housekeepin}(2380)
gsd-keyboard(2315)
{gsd-keyboard}(2340)
{gsd-keyboard}(2347)
{gsd-keyboard}(2348)
gsd-media-keys(2319)
{gsd-media-keys}(2367)
{gsd-media-keys}(2369)
{gsd-media-keys}(2371)
gsd-power(2325)
{gsd-power}(2372)
{gsd-power}(2375)
{gsd-power}(2378)
gsd-print-notif(2327)
{gsd-print-notif}(2345)
{gsd-print-notif}(2350)
gsd-printer(2426)
{gsd-printer}(2441)
{gsd-printer}(2447)
gsd-rfkill(2330)
{gsd-rfkill}(2331)
{gsd-rfkill}(2333)
gsd-screensaver(2332)
{gsd-screensaver}(2334)
{gsd-screensaver}(2338)
gsd-sharing(2336)
{gsd-sharing}(2356)
{gsd-sharing}(2358)
{gsd-sharing}(2361)
gsd-smartcard(2349)
{gsd-smartcard}(2355)
{gsd-smartcard}(2357)
{gsd-smartcard}(2381)
{gsd-smartcard}(2382)
{gsd-smartcard}(2423)


├─gsd-sound(2353)─┬─{gsd-sound}(2362)
├─{gsd-sound}(2366)
└─{gsd-sound}(2386)
├─gsd-usb-protect(2359)─┬─{gsd-usb-protect}(2364)
├─{gsd-usb-protect}(2365)
└─{gsd-usb-protect}(2392)
├─gsd-wacom(2363)─┬─{gsd-wacom}(2408)
├─{gsd-wacom}(2410)
└─{gsd-wacom}(2413)
├─gvfs-goa-volume(2259)─┬─{gvfs-goa-volume}(2263)
└─{gvfs-goa-volume}(2264)
├─gvfs-gphoto2-vo(2248)─┬─{gvfs-gphoto2-vo}(2256)
└─{gvfs-gphoto2-vo}(2258)
├─gvfs-mtp-volume(2241)─┬─{gvfs-mtp-volume}(2243)
└─{gvfs-mtp-volume}(2245)
├─gvfs-udisks2-vo(2231)─┬─{gvfs-udisks2-vo}(2234)
├─{gvfs-udisks2-vo}(2235)
└─{gvfs-udisks2-vo}(2237)
├─gvfsd(2124)─┬─gvfsd-trash(2828)─┬─{gvfsd-trash}(28+
└─{gvfsd-trash}(28+
├─{gvfsd}(2125)
└─{gvfsd}(2126)
├─gvfsd-fuse(2129)─┬─{gvfsd-fuse}(2131)
├─{gvfsd-fuse}(2132)
├─{gvfsd-fuse}(2133)
├─{gvfsd-fuse}(2134)
└─{gvfsd-fuse}(2136)
├─gvfsd-metadata(2271)─┬─{gvfsd-metadata}(2272)
└─{gvfsd-metadata}(2273)
├─ibus-portal(2154)─┬─{ibus-portal}(2158)
└─{ibus-portal}(2159)
├─pipewire(2180)─┬─{pipewire}(2182)
└─{pipewire}(2185)
├─pipewire-pulse(2184)─┬─{pipewire-pulse}(2193)
└─{pipewire-pulse}(2197)
├─tracker-miner-f(2860)─┬─{tracker-miner-f}(2866)
├─{tracker-miner-f}(2867)
├─{tracker-miner-f}(2868)
├─{tracker-miner-f}(2869)
└─{tracker-miner-f}(2870)
├─wireplumber(2181)─┬─{wireplumber}(2187)
├─{wireplumber}(2188)
├─{wireplumber}(2189)
└─{wireplumber}(2198)
├─xdg-desktop-por(2524)─┬─{xdg-desktop-por}(2526)
├─{xdg-desktop-por}(2528)
├─{xdg-desktop-por}(2600)
├─{xdg-desktop-por}(2615)
└─{xdg-desktop-por}(2616)

```
│                                  ├─xdg-desktop-por(2544)─┬─{xdg-desktop-por}(2568)
│                                  │                       ├─{xdg-desktop-por}(2569)
│                                  │                       └─{xdg-desktop-por}(2572)
│                                  ├─xdg-desktop-por(2603)─┬─{xdg-desktop-por}(2605)
│                                  │                       ├─{xdg-desktop-por}(2607)
│                                  │                       └─{xdg-desktop-por}(2608)
│                                  ├─xdg-document-po(2531)─┬─fusermount(2537)
│                                  │                       ├─{xdg-document-po}(2534)
│                                  │                       ├─{xdg-document-po}(2535)
│                                  │                       ├─{xdg-document-po}(2540)
│                                  │                       ├─{xdg-document-po}(2541)
│                                  │                       └─{xdg-document-po}(2542)
│                                  └─xdg-permission-(2161)─┬─{xdg-permission-}(2163)
│                                                          └─{xdg-permission-}(2165)
├─systemd-journal(639)
├─systemd-logind(799)
├─systemd-udevd(653)
├─tuned(974)─┬─{tuned}(1168)
│                                                          ├─{tuned}(1990)
│                                                          └─{tuned}(1993)
├─tuned-ppd(1596)─┬─{tuned-ppd}(1646)
│                                                          ├─{tuned-ppd}(1647)
│                                                          └─{tuned-ppd}(1648)
├─udisksd(800)─┬─{udisksd}(817)
│                                                          ├─{udisksd}(819)
│                                                          ├─{udisksd}(838)
│                                                          └─{udisksd}(886)
├─upowerd(801)─┬─{upowerd}(827)
│                                                          └─{upowerd}(832)
└─wpa_supplicant(1629)
```

pstree -h

```
[felipemerino@localhost ~]$ pstree -h
systemd─┬─ModemManager───3*[{ModemManager}]
 ├─NetworkManager───2*[{NetworkManager}]
 ├─accounts-daemon───3*[{accounts-daemon}]
 ├─alsactl
 ├─atd
 ├─auditd─┬─sedispatch
 │                                                          └─2*[{auditd}]
 ├─avahi-daemon───avahi-daemon
 ├─chronyd
 ├─colord───3*[{colord}]
 ├─crond
 ├─cupsd
 ├─dbus-broker-lau───dbus-broker
 ├─firewalld───{firewalld}
```

```
├─fwupd───4*[{fwupd}]
├─gdm─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b───3*[{gnome+
│     │                 │                 └─2*[{gdm-wayland-ses}]
│     │                 └─2*[{gdm-session-wor}]
│     └─2*[{gdm}]
├─geoclue───3*[{geoclue}]
├─gnome-keyring-d───3*[{gnome-keyring-d}]
├─irqbalance───{irqbalance}
├─lsmd
├─mcelog
├─polkitd───6*[{polkitd}]
├─rsyslogd───2*[{rsyslogd}]
├─rtkit-daemon───2*[{rtkit-daemon}]
├─sshd
├─sssd_kcm
├─switcheroo-cont───3*[{switcheroo-cont}]
├─systemd─┬─(sd-pam)
│         ├─at-spi-bus-laun─┬─dbus-broker-lau───dbus-broker
│         │                 └─3*[{at-spi-bus-laun}]
│         ├─at-spi2-registr───2*[{at-spi2-registr}]
│         ├─dbus-broker-lau───dbus-broker
│         ├─dconf-service───2*[{dconf-service}]
│         ├─evolution-addre───5*[{evolution-addre}]
│         ├─evolution-calen───8*[{evolution-calen}]
│         ├─evolution-sourc───3*[{evolution-sourc}]
│         ├─2*[gjs───5*[{gjs}]]
│         ├─gnome-session-b─┬─evolution-alarm───5*[{evolution-alarm}]
│         │                 ├─gnome-software───6*[{gnome-software}]
│         │                 ├─gsd-disk-utilit───2*[{gsd-disk-utilit}]
│         │                 └─3*[{gnome-session-b}]
│         ├─gnome-session-c───{gnome-session-c}
│         ├─gnome-shell─┬─ibus-daemon─┬─ibus-dconf───3*[{ibus-dconf}]
│         │             │             ├─ibus-engine-sim───2*[{ibus-engi+
│         │             │             ├─ibus-extension-───3*[{ibus-exte+
│         │             │             └─2*[{ibus-daemon}]
│         │             └─18*[{gnome-shell}]
│         ├─gnome-shell-cal───5*[{gnome-shell-cal}]
│         ├─gnome-terminal-─┬─bash─┬─pstree
│         │                 │      └─3*[vim]
│         │                 └─3*[{gnome-terminal-}]
│         ├─goa-daemon───3*[{goa-daemon}]
│         ├─goa-identity-se───3*[{goa-identity-se}]
│         ├─gsd-a11y-settin───3*[{gsd-a11y-settin}]
│         ├─gsd-color───3*[{gsd-color}]
│         ├─gsd-datetime───3*[{gsd-datetime}]
│         ├─gsd-housekeepin───3*[{gsd-housekeepin}]
│         ├─gsd-keyboard───3*[{gsd-keyboard}]
│         ├─gsd-media-keys───3*[{gsd-media-keys}]
│         ├─gsd-power───3*[{gsd-power}]
```

```
│                                                  ├─gsd-print-notif───2*[{gsd-print-notif}]
│                                                  ├─gsd-printer───2*[{gsd-printer}]
│                                                  ├─gsd-rfkill───2*[{gsd-rfkill}]
│                                                ├─gsd-screensaver───2*[{gsd-screensaver}]
│                                                  ├─gsd-sharing───3*[{gsd-sharing}]
│                                                ├─gsd-smartcard───5*[{gsd-smartcard}]
│                                                  ├─gsd-sound───3*[{gsd-sound}]
│                                                ├─gsd-usb-protect───3*[{gsd-usb-protect}]
│                                                  ├─gsd-wacom───3*[{gsd-wacom}]
│                                                ├─gvfs-goa-volume───2*[{gvfs-goa-volume}]
│                                                ├─gvfs-gphoto2-vo───2*[{gvfs-gphoto2-vo}]
│                                                ├─gvfs-mtp-volume───2*[{gvfs-mtp-volume}]
│                                                ├─gvfs-udisks2-vo───3*[{gvfs-udisks2-vo}]
│                                                ├─gvfsd─┬─gvfsd-trash───2*[{gvfsd-trash}]
│                                                │       └─2*[{gvfsd}]
│                                                  ├─gvfsd-fuse───5*[{gvfsd-fuse}]
│                                                ├─gvfsd-metadata───2*[{gvfsd-metadata}]
│                                                  ├─ibus-portal───2*[{ibus-portal}]
│                                                  ├─pipewire───2*[{pipewire}]
│                                                ├─pipewire-pulse───2*[{pipewire-pulse}]
│                                                ├─tracker-miner-f───5*[{tracker-miner-f}]
│                                                  ├─wireplumber───4*[{wireplumber}]
│                                                ├─xdg-desktop-por───5*[{xdg-desktop-por}]
│                                              ├─2*[xdg-desktop-por───3*[{xdg-desktop-por}]]
│                                                  ├─xdg-document-po─┬─fusermount
│                                                  │                 └─5*[{xdg-document-po}]
│                                                └─xdg-permission-───2*[{xdg-permission-}]
├─systemd-journal
├─systemd-logind
├─systemd-udevd
├─tuned───3*[{tuned}]
├─tuned-ppd───3*[{tuned-ppd}]
├─udisksd───4*[{udisksd}]
├─upowerd───2*[{upowerd}]
└─wpa_supplicant
[felipemerino@localhost ~]$
```

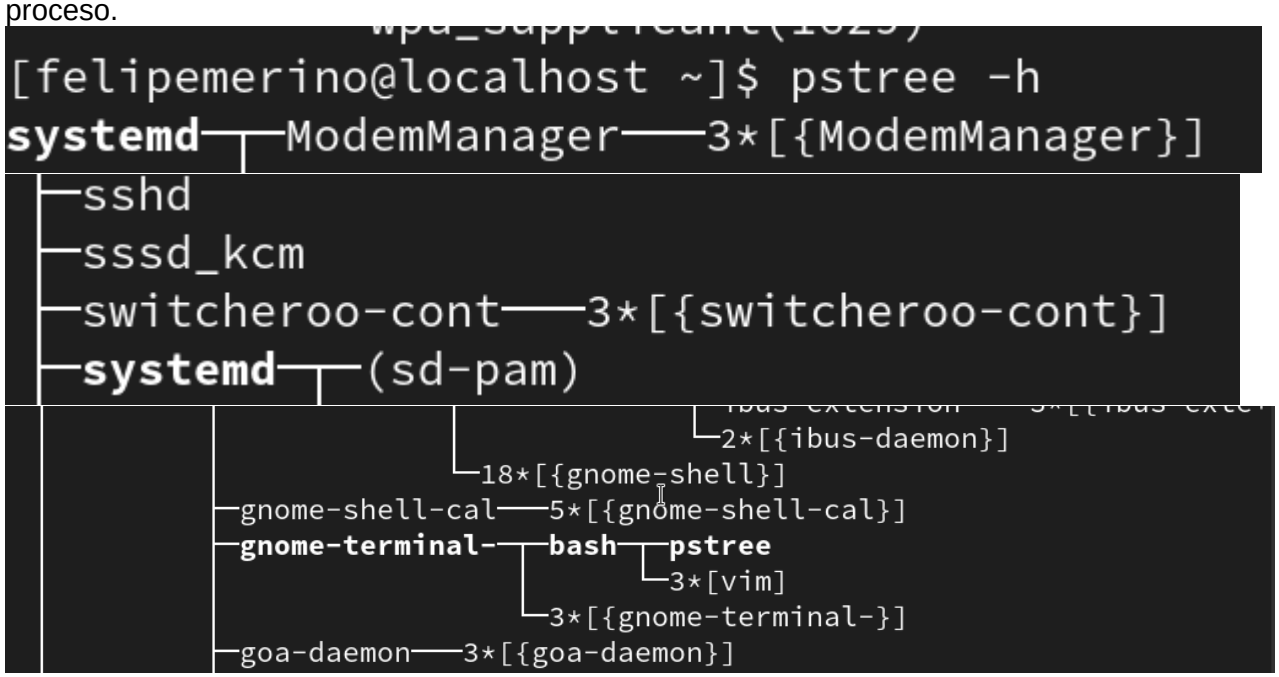
¿Cuál es la diferencia?

pstree: Muestra el árbol jerárquico de procesos usando únicamente sus nombres.

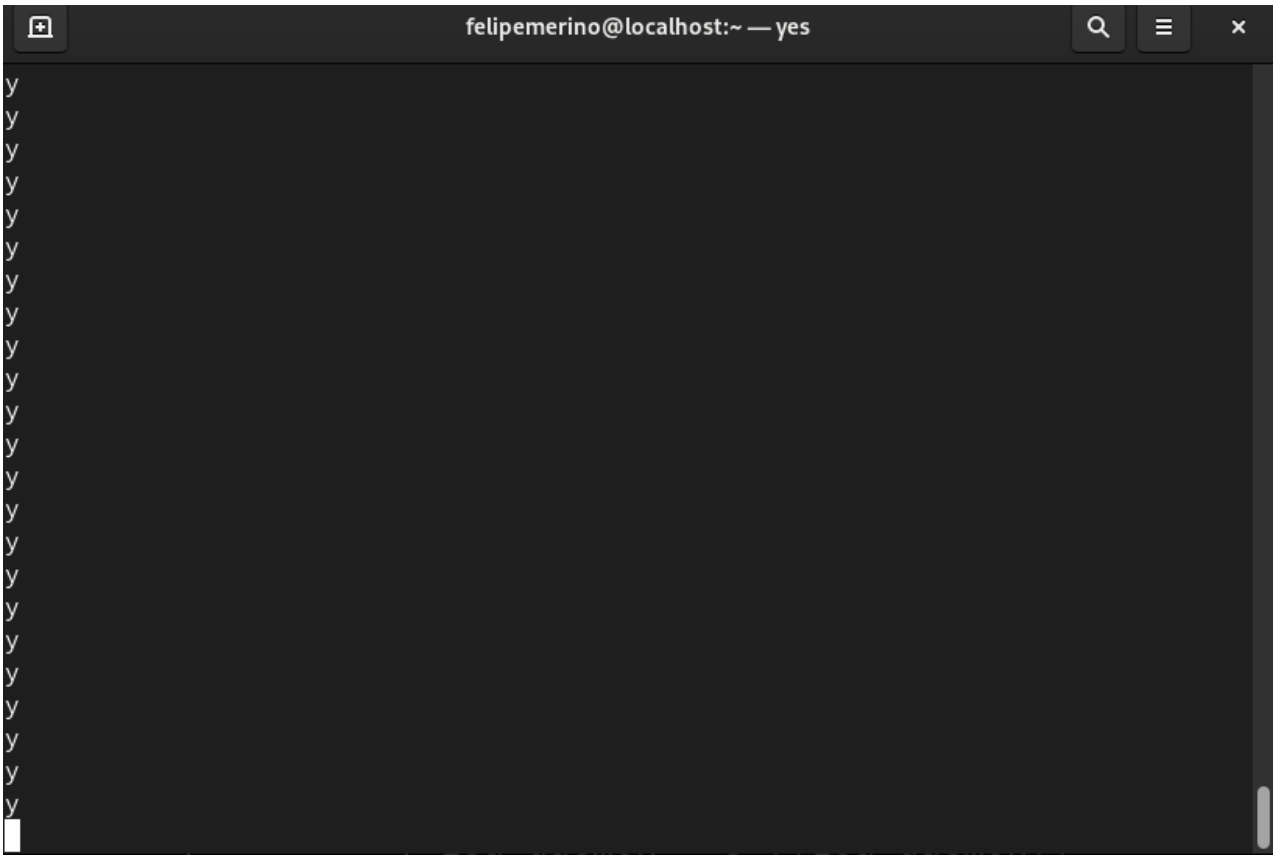
pstree -p: La opción -p significa "PIDs" (Process IDs). Este comando muestra el mismo árbol, pero añade el PID (Número de Identificación del Proceso) entre paréntesis al lado de cada nombre de proceso. Es útil para saber el número exacto de un proceso específico.

pstree -h: La opción -h significa "highlight" (resaltar). Este comando resalta la línea del proceso actual (en este caso, pstree mismo) y todas sus líneas "padre" o "ancestros" hasta llegar al inicio

del sistema. No muestra los PIDs, su función es puramente visual para trazar el linaje de un proceso.



3.17 Ejecutar el comando yes en una terminal y mediante el comando *ps* (investigar las opciones) determinar los PID y PPID del proceso asociado.




```
        └─wpa_supplicant
[felipemerino@localhost ~]$ ps -ef | grep yes
felipem+    4171    4141 43 17:54 pts/1    00:00:49 yes
felipem+    4208    2972  0 17:56 pts/0    00:00:00 grep --color=auto yes
[felipemerino@localhost ~]$
```

PID: 4171
PPID: 4141

3.18 Mediante el comando *ps* (investigar las opciones) determinar para el proceso asociado al comando yes:

```
[felipemerino@localhost ~]$ ps -elf | grep yes
0 S felipem+    4171    4141 44  80   0 - 55240 wait_w 17:54 pts/1    00:04:27 yes
0 S felipem+    4330    2972  0  80   0 - 55420 pipe_r 18:04 pts/0    00:00:00 grep --colo
r=auto yes
[felipemerino@localhost ~]$ S
```

- F – Indicadores asociados con el proceso: 0
- S – Estado del proceso: S (Sleeping)
- UID – Identificación del usuario (ID) propietario del proceso: felipem+

- CPU – Utilización del proceso: 44
- PRI – Prioridad del proceso: 80
- WCHAN – El suceso por el cual el proceso está esperando: wait_w
- TTY – El Terminal que controla el proceso: pts/1
- TIME – Tiempo de ejecución acumulativa del proceso: 00:04:27

3.19 Utilice el comando *top* para mostrar todos los procesos, usuarios a los que pertenecen los procesos, y la serie de recursos que ocupan en memoria los procesos. Usar también el comando top –u <Usuario>.

```
[felipemerino@localhost ~]$ top

top - 21:09:04 up  4:10,  2 users,  load average: 2,99, 2,74, 2,56
Tasks: 237 total,   4 running, 230 sleeping,   3 stopped,   0 zombie
%Cpu(s): 17,6 us, 36,6 sy,  0,0 ni, 41,9 id,  0,1 wa,  3,4 hi,  0,4 si,  0,0 st
MiB Mem :   6952,6 total,   3612,9 free,   2009,6 used,   1641,4 buff/cache
MiB Swap:   3616,0 total,   3616,0 free,      0,0 used.   4943,0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2954 felipem+  20   0  769976  57196  43644 R  51,7   0,8 175:15.99 gnome-terminal-
   4171 felipem+  20   0  220960   1888   1772 R  48,0   0,0  19:33.23 yes
   3349 felipem+  20   0 3390348 317192 119620 S  13,9   4,5   4:53.57 Isolated Web Co
   4475 root      20   0       0      0      0 I   9,9   0,0   1:03.29 kworker/u15:1-eve+
   3038 root      20   0       0      0      0 R   9,3   0,0   1:35.98 kworker/u15:2-eve+
   3083 root      20   0       0      0      0 I   7,9   0,0   1:28.97 kworker/u13:1-eve+
   4181 root      20   0       0      0      0 I   7,9   0,0   1:40.17 kworker/u14:2-eve+
   4512 root      20   0       0      0      0 I   7,6   0,0   0:48.96 kworker/u14:1-eve+
   4398 root      20   0       0      0      0 I   5,3   0,0 154:06.46 kworker/u13:0-eve+
   2109 felipem+  20   0 4704100 390880 148768 S   3,3   5,5   6:19.72 gnome-shell
   4624 felipem+  20   0  225936   4316   3444 R   0,7   0,1   0:00.12 top
   3120 felipem+  20   0   11,5g 546224 209508 S   0,3   7,7   5:19.08 firefox
      1 root      20   0  174968  17308  11108 S   0,0   0,2   0:05.49 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_workqueue
```

```
[felipemerino@localhost ~]$ top -u felipemerino

top - 21:09:47 up  4:10,  2 users,  load average: 2,92, 2,76, 2,58
Tasks: 238 total,   5 running, 229 sleeping,   4 stopped,   0 zombie
%Cpu(s): 14,5 us, 36,4 sy,  0,0 ni, 45,3 id,  0,0 wa,  3,5 hi,  0,2 si,  0,0 st
MiB Mem :   6952,6 total,   3606,5 free,   2016,0 used,   1641,4 buff/cache
MiB Swap:   3616,0 total,   3616,0 free,      0,0 used.   4936,6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2954 felipem+  20   0  769976  57196  43644 R  55,3   0,8 175:38.72 gnome-terminal-
   4171 felipem+  20   0  220960   1888   1772 R  51,0   0,0  19:53.76 yes
   2109 felipem+  20   0 4699940 390636 148768 S   3,6   5,5   6:22.24 gnome-shell
   4634 felipem+  20   0  225924   4396   3524 R   1,0   0,1   0:00.07 top
   3349 felipem+  20   0 3390364 330092 119636 S   0,3   4,6   4:54.29 Isolated Web Co
   2020 felipem+  20   0   25820  15652  11040 S   0,0   0,2   0:00.84 systemd
   2022 felipem+  20   0  176356   8360   2008 S   0,0   0,1   0:00.00 (sd-pam)
   2038 felipem+  20   0  453244   7640   6536 S   0,0   0,1   0:00.07 gnome-keyring-d
   2042 felipem+  20   0  374356   6168   5596 S   0,0   0,1   0:00.01 gdm-wayland-ses
   2044 felipem+  20   0   10584   4584   4108 S   0,0   0,1   0:00.03 dbus-broker-lau
   2046 felipem+  20   0    6888   4644   2404 S   0,0   0,1   0:00.50 dbus-broker
   2050 felipem+  20   0  513380  17720  15536 S   0,0   0,2   0:00.07 gnome-session-b
   2091 felipem+  20   0  303688   5608   5148 S   0,0   0,1   0:00.00 gnome-session-c
   2093 felipem+  20   0  808972  19036  16056 S   0,0   0,3   0:00.24 gnome-session-b
   2124 felipem+  20   0  452840   8416   7384 S   0.0   0.1   0:00.04 gvfsd
```

3.20 Los procesos pueden ser controlados en dos formas:

1. Desde el *shell* que lo inició usando su *job number*.
2. Desde cualquier lugar en el sistema, usando su PID.

Señales enviadas a procesos:

- *Foreground*: Un proceso que recibe la entrada del teclado que es escrita en una terminal.
- *Background*: Un proceso que no recibe ninguna entrada del teclado.

Procesos Foreground

Los procesos foreground son invocados simplemente ejecutando un comando en la línea de comandos.

 # find / -name README

Procesos de Background

Los procesos de background se invocan colocando un & al final de la línea de comandos.

 # find / -name README &

<Ctrl-z>	Suspende la tarea *foreground*
jobs	Lista los jobs en *background* o suspendidos
fg	devuelve la tarea suspendida al *foreground*
bg	devuelve las tarea suspendida al *background*

- Los comandos **fg, bg y kill** pueden ser usados con un número de job. Para matar al job 3: **kill %3**
- Si se especifica el parámetro **nohup** los procesos no mueren a pesar de que el usuario se desloguee.

Figura 13. Resumen de los comandos para el control de procesos

Ejecutar:

$ sleep 100&

$ nohup sleep 100&

Copie los PID y, en una nueva terminal, ejecute:

$ ps -fe |grep sleep

3.21 Señales Kill

Muchas señales pueden ser enviadas a un proceso:

- Usando interrupciones de teclado (procesos foreground).
- Usando el comando ***kill -signal PID***.
- Ejecutar ***kill -l***

Las señales más importantes:

Nombre de la señal	Número	Significado y uso
HUP	1	Hang up. Esta señal es enviada automáticamente cuando se desloguea o el módem es desconectado. Es utilizado por procesos "demonios" para hacer que el archivo de configuración vuelva a leerse.
INT	2	Interrupt. Dejar de correr. Esta señal es enviada cuando se presiona Ctrl+c.
KILL	9	Kill. Detener el proceso inmediatamente y sin ninguna condición. Esta es una señal enviada en casos drásticos, ya que el proceso no puede ignorarla. Es una señal de "detención de emergencia".
TERM	15	Terminate. Terminar el proceso lo más ameno posible. Esta señal se utiliza para preguntar al proceso por un cierre tranquilo.
TSTP	20	Stop executing. Termina de ejecutar y está listo para continuar. Esta señal es enviada cuando se presiona Ctrl+z.
CONT	18	Continue execution. Continuar la ejecución. Esta señal es enviada para empezar de nuevo un proceso que fue detenido por SIGTSP o SIGSTOP. (La *Shell* envía esta señal cuando se utiliza los comandos *fg* o *bg* después de

		detener un proceso con Ctrl+z).

Tabla 4. Señales más importantes enviadas a un proceso

Si un proceso se ejecuta de manera adecuada, basta terminarlo con la señal ***SIGTERM*** (15).

Luego de que detenga todo lo que está haciendo, ejecute:

$ kill -15 PID

Si un proceso se cuelga y no puede responder, debe usarse el comando más fuerte para detenerlo:

$ kill -9 PID

$ kill -KILL PID

En ocasiones la señal HUP puede ser utilizada para re-leer determinados archivos de configuración.

$ kill -1 `cat /var/run/rsyslog.pid`

Ejecute el comando: ***yes que tal > /dev/null***

Abra otra terminal y determine el PID del comando yes.
Termine el proceso yes.
Verifique que sucedió y el mensaje que apareció en la primera terminal.

Ejecute el comando: ***yes que tal > /dev/null***

Abra otra terminal y determine el PID del comando yes.
Mate el proceso yes.
Verifique que sucedió y el mensaje que apareció en la primera terminal.

Ejecute el comando: ***yes que tal > /dev/null &***

Ejecute el comando ***jobs.***
¿En qué estado se encuentra el proceso yes?
¿Qué porcentaje de CPU y memoria consume el proceso yes?

Ejecute ***fg job_ID***, y explicar qué sucede.
Termine el proceso yes con la señal SIGTERM.

Nota: el job_ID se refiere al número que aparece entre corchetes al ejecutar el comando jobs.

Ejecute el comando: ***yes que tal > /dev/null &***

Ejecute ***kill %job_ID.***
Ejecute el comando ***jobs.***

¿En qué estado se encuentra el proceso yes?

3.22 Cree el fichero bucle con el código:

```
#! /bin/bash
yes que tal > /dev/null
exec bucle
```

Nota: Puede crear el fichero en el "Editor de textos" de CentOS o utilizar uno de los editores presentados en clases anteriores.

Asigne al fichero bucle permisos de ejecución. (Si no funciona, pruebe a asignar todos los permisos al fichero)
Ejecutar con: ***./bucle***
En otra terminal ejecute ***top***
Verifique los datos del proceso yes, consumo CPU, memoria, prioridad.
Digite ***r***
Coloque el número 10 y presione Enter dos veces
Verifique nuevamente los datos del proceso yes, consumo CPU, memoria, prioridad.

3.23 Cambio de prioridades a los procesos

El comando ***nice*** es usado para iniciar un proceso con una prioridad definida por el usuario.
nice [-n <value>] <original command>
value: entre -20 y +19

```
$ nice -n 10 my_program &
[1] 4862
$ ps l
F  UID   PID   PRI    NI   VSZ  ...   COMMAND
0  500 4372     9     0   4860 ...    -bash
0  500 4862    15    10   3612 ...    my_program
0  500 4863    21     0   1556 ...    ps l
```

Figura 14. Ejemplo de uso del comando nice

El comando renice

El comando ***renice*** es usado para cambiar la prioridad de un proceso que está corriendo.

renice <new priority> <PID>

```
$ renice 15 4862
4862: old priority 10, new priority 15
$ ps l
F  UID   PID   PRI    NI   VSZ  ...   COMMAND
0  500 4372     9     0   4860 ...    -bash
0  500 4862    22    15   3612 ...    my_program
0  500 4868    21     0   1556 ...    ps l
```

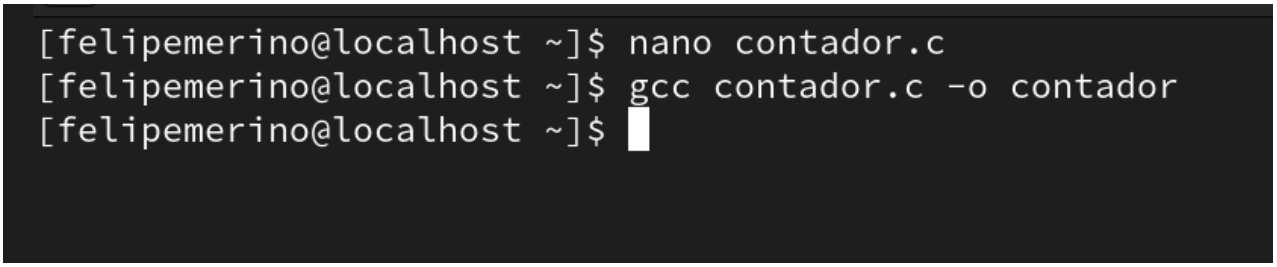
Figura 15. Ejemplo de uso del comando renice

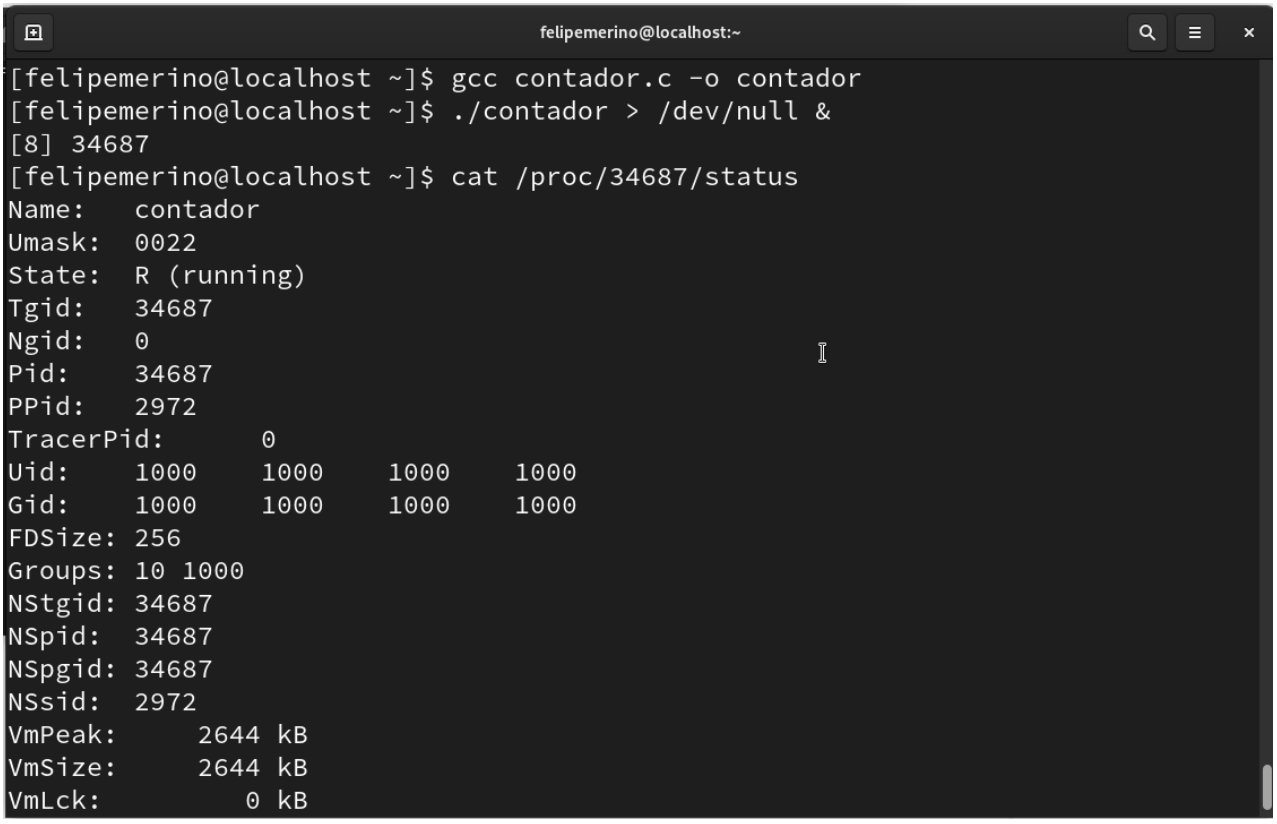
3.24 Investigue en qué directorio se puede observar el PCB de un proceso en Linux.

En Linux, la información conceptual del PCB (Process Control Block) no se almacena en un solo archivo, sino que se expone a través de un sistema de archivos virtual llamado /proc.

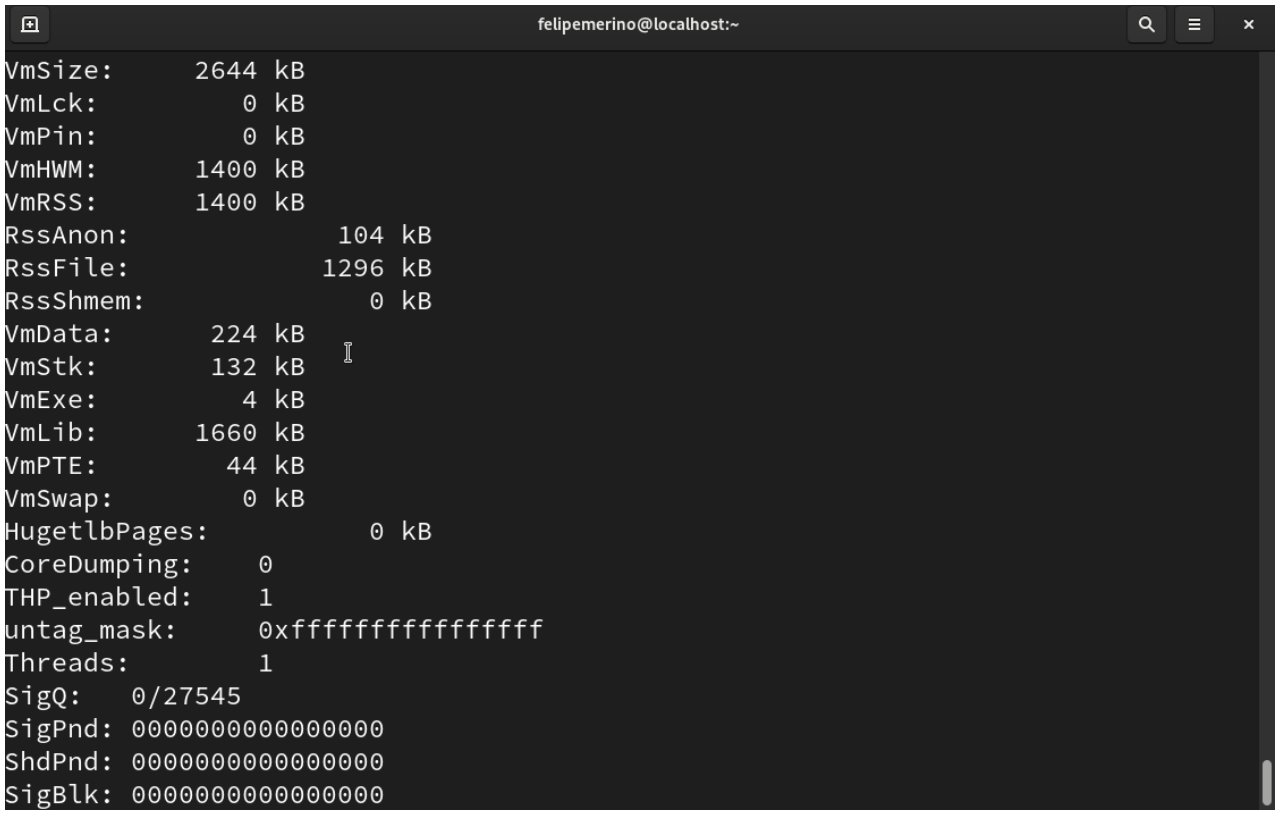
Dentro del directorio /proc, el sistema operativo crea un subdirectorio numérico por cada proceso que está en ejecución, usando su PID (Process ID) como nombre.

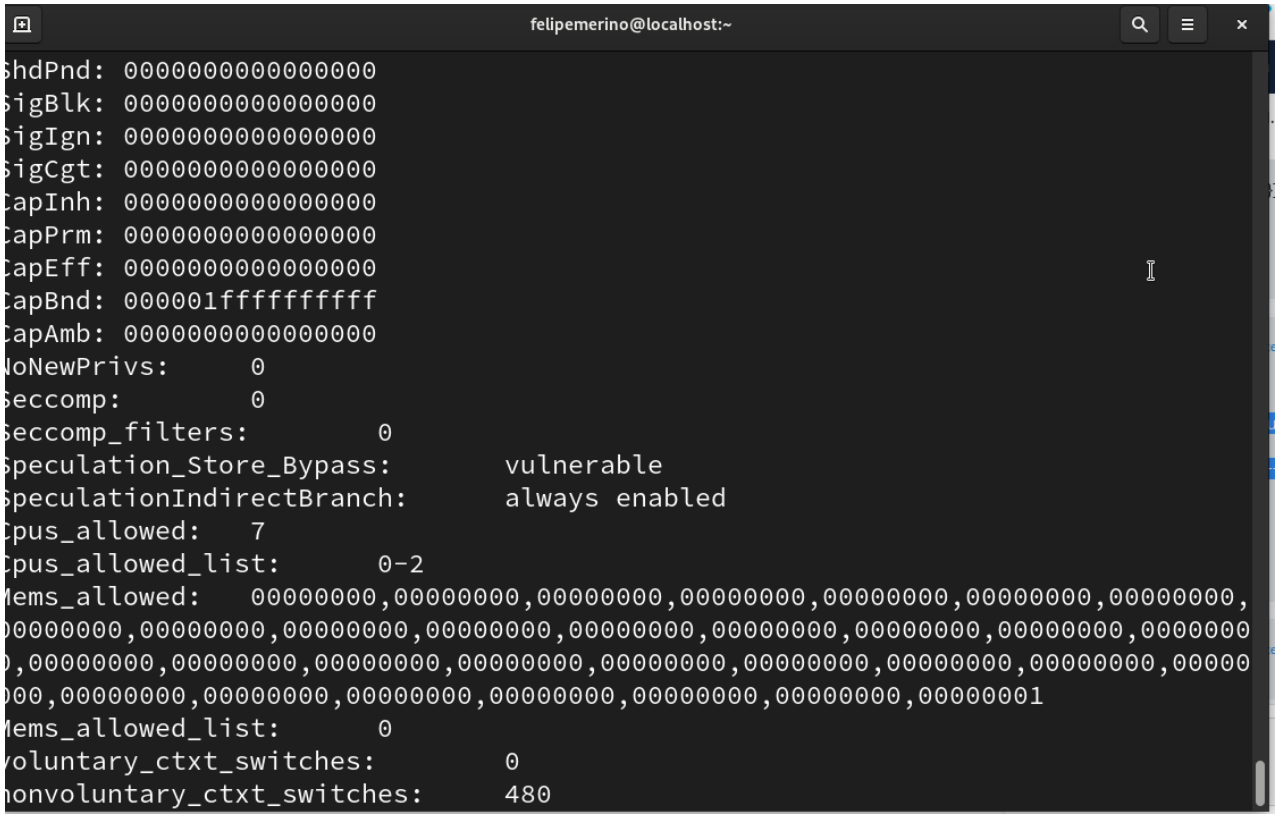
3.25 Cree un proceso que imprima en pantalla números del 1 al 100 000 000, ejecútelo en background y revise la información de su PCB.





El PID es 34687


felipemerino@localhost:~
VmSize: 2644 kB
VmLck: 0 kB
VmPin: 0 kB
VmHWM: 1400 kB
VmRSS: 1400 kB
RssAnon: 104 kB
RssFile: 1296 kB
RssShmem: 0 kB
VmData: 224 kB
VmStk: 132 kB
VmExe: 4 kB
VmLib: 1660 kB
VmPTE: 44 kB
VmSwap: 0 kB
HugetlbPages: 0 kB
CoreDumping: 0
THP_enabled: 1
untag_mask: 0xffffffffffffffff
Threads: 1
SigQ: 0/27545
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000000000


felipemerino@localhost:~
ShdPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000000000000
SigCgt: 0000000000000000
CapInh: 0000000000000000
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 000001ffffffffff
CapAmb: 0000000000000000
NoNewPrivs: 0
Seccomp: 0
Seccomp_filters: 0
Speculation_Store_Bypass: vulnerable
SpeculationIndirectBranch: always enabled
Cpus_allowed: 7
Cpus_allowed_list: 0-2
Mems_allowed: 00000000,00000000,00000000,00000000,00000000,00000000,00000000,
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,0000000
0,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000
000,00000000,00000000,00000000,00000000,00000000,00000000,00000001
Mems_allowed_list: 0
voluntary_ctxt_switches: 0
nonvoluntary_ctxt_switches: 480

4. INFORME

Ejercicio 3.6: Comandos Básicos de la Terminal

Análisis de Comandos:

Comando	Descripción de la Tarea
date	Muestra la fecha y hora actuales del sistema.
cal / cal septiembre 1983	Muestra un calendario. La segunda forma muestra el calendario específico para ese mes y año.
w	Muestra qué usuarios están conectados (logged in) al sistema, desde dónde y qué comandos están ejecutando.
clear	Limpia todo el texto de la pantalla de la terminal.
echo $HOME	Imprime el valor de una variable de entorno. En este caso, la ruta al directorio personal (ej: /home/felipemerino).
write felipemerino	Permite enviar un mensaje de texto en tiempo real a la terminal de otro usuario que esté conectado.
wall "mensaje"	(Write to All) Envía un mensaje a las terminales de *todos* los usuarios conectados en el sistema.
history	Muestra una lista numerada de los comandos más recientes que se han ejecutado en la terminal.
cat ~/.bash_history	Muestra el contenido del archivo de texto donde se almacena el historial de comandos de forma persistente.
!-1 o !!	Vuelve a ejecutar el último comando (el más reciente).
!7	Ejecuta el comando que tiene el número 7 en la lista de history.

Ejercicios 3.10 y 3.11: Uso del Editor vi

Ejercicio 3.10: Aprendizaje de Modos de vi

Ejercicio que cumplió la función de enseñarnos a usar el editor de texto vi, en el que se realizó una figura

Modo	**Tecla de Acceso (desde M. Comando)**	**Propósito Principal**
Modo Comando	Por defecto (o Esc)	Navegar (h,j,k,l), borrar (dd), copiar (yy), pegar (p), y cambiar de modo.
Modo Edición	i, a, O, etc.	Insertar y escribir texto.
Modo Línea (Ex)	:	Ejecutar comandos avanzados (guardar, salir, buscar, reemplazar).

Ejercicio 3.12: Comando de Sustitución en vi

Pregunta: Ejecutar y explicar el comando :1,$s /DE/SA/g.

Análisis:

Al ejecutar el comando sobre el arte ASCII, el texto "DEBER DE" se transformó en "SABER SA". Es un comando de búsqueda y reemplazo.

Componente	**Significado**
:	Activa el **Modo Línea** para insertar un comando.
1,$	Define el **rango** de operación. 1 es la primera línea y $ es la última. Significa "en todo el archivo".
s	Comando substitute (sustituir).
/DE/	El **patrón de búsqueda** (el texto a encontrar).
/SA/	El **texto de reemplazo** (el texto por el cual se sustituirá).
/g	Indicador **"global"**, que reemplaza *todas* las instancias en cada línea (no solo la primera).

Ejercicio 3.16: Jerarquía de Procesos con pstree

Comparativa de Comandos:

Comando	Función
pstree	Muestra el árbol jerárquico de procesos usando solo sus **nombres**.
pstree -p	Muestra el árbol e **incluye el PID (Process ID)** de cada proceso entre paréntesis.
pstree -h	Muestra el árbol y **resalta (ilumina)** el proceso actual y su linaje (todos sus "padres").

Ejercicio 3.17: Obtener PID y PPID

Metodología:

1. **Terminal 1:** Se ejecutó yes, bloqueando la terminal con una salida constante.
2. **Terminal 2:** Se ejecutó ps -ef | grep yes para encontrar el proceso.

Ejercicio 3.18: Análisis Detallado del Proceso yes

Metodología:

Se utilizó el comando ps -elf | grep yes para obtener un formato de salida largo y detallado.

Cabecera	Pregunta	Valor	Significado
F	Indicadores	0	Indicadores de estado del proceso.
S	Estado	S	*Sleeping* (Durmiendo).
UID	Identificación	felipem+	El nombre del usuario propietario del proceso.
C	Utilización de CPU	44	44% de uso de CPU en ese instante.
PRI	Prioridad	80	Prioridad de ejecución (asignada por el kernel).

WCHAN	Suceso de Espera	wait_w	Esperando una operación de escritura (*wait write*).
TTY	Terminal	pts/1	La terminal que controla el proceso.
TIME	Tiempo de Ejecución	00:04:27	Tiempo de CPU acumulado que ha usado el proceso.

Ejercicio 3.19: Monitoreo en Tiempo Real con top

Pregunta: Utilizar top y top -u <Usuario>.

Comparativa de Comandos:

Comando	Función
top	Inicia un monitor de sistema interactivo en tiempo real. Muestra una lista de procesos que se actualiza constantemente, ordenada por uso de CPU. Se sale con la tecla q.
top -u felipem+	Inicia top pero **filtra la lista** para mostrar únicamente los procesos que pertenecen al usuario especificado.

Ejercicio 3.24: Ubicación del PCB

La información del PCB (Process Control Block) en Linux se expone a través del sistema de archivos virtual /proc. Por cada proceso en ejecución, se crea un directorio con su PID (ej: /proc/4171).

El archivo más completo para la inspección humana es **/proc/[PID]/status**, que contiene un resumen legible del estado del proceso, su padre (PPid), su usuario (Uid), el uso de memoria (VmSize), etc.

Ejercicio 3.25: Creación y Revisión de Proceso

Pregunta: Crear un proceso que cuente hasta 100,000,000, ejecutarlo en *background* y revisar su PCB.

Metodología:

Se decidió crear un programa en C para tener un proceso claramente identificable.

1. **Creación:** Se usó vi contador.c para crear el código fuente.
2. **Compilación:** Se usó gcc contador.c -o contador para crear el programa ejecutable. (Un intento de ejecutar /.contador.c falló con "Permiso denegado" porque el código fuente no es un ejecutable).
3. **Ejecución:** Se ejecutó en *background* (&) con la salida redirigida:
4. ./contador > /dev/null &

   La terminal devolvió el PID (p.ej., 34612).

5. **Revisión del PCB:** Se intentó revisar el archivo status:
   cat /proc/34612/status

6. **Resultado Final:** Al volver a ejecutar cat /proc/PID_NUEVO/status, el comando tuvo éxito, mostrando la información del PCB del proceso contador mientras seguía en ejecución.