**PRÁCTICA 2 DE SISTEMAS OPERATIVOS**

**TEMA: Administración de Procesos**

**Nombre: Fernando Huilca**

**Carrera: Ingeniería de Software**

**Grupo: GR1SW**

**Fecha: 17 / 08 / 2024**

**Índice de Contenidos**

[1. OBJETIVOS 2](#_Toc167837453)

[2. INFORME 2](#_Toc167837454)

[3. CONCLUSIONES Y RECOMENDACIONES 23](#_Toc167837455)

[4. BIBLIOGRAFÍA 23](#_Toc167837456)

**Índice de Imágenes**

[Ilustración 1 fecha y hora de la computadora 2](#_Toc169646622)

[Ilustración 2 calendario septiembre 1983 3](#_Toc169646623)

[Ilustración 3 Usuarios del sistema 3](#_Toc169646624)

[Ilustración 4 Comando para limpiar el terminal 4](#_Toc169646625)

[Ilustración 5 Contenido de las variables de entorno 4](#_Toc169646626)

[Ilustración 6 Mensaje de texto en la terminal comando write 5](#_Toc169646627)

[Ilustración 7 Historial de comandos 5](#_Toc169646628)

[Ilustración 8 Fichero bash\_history con uso de cat 6](#_Toc169646629)

[Ilustración 9 Comando !-1 6](#_Toc169646630)

[Ilustración 10 Comando !! 7](#_Toc169646631)

[Ilustración 11 Comando !7 7](#_Toc169646632)

[Ilustración 12 Figura realizada con el comando Vi 8](#_Toc169646633)

[Ilustración 13 Ejecución del comando :1,$s /DE/SA/g 9](#_Toc169646634)

[Ilustración 14 Comando Pstree -p 10](#_Toc169646635)

[Ilustración 15 Comando pstree -h 11](#_Toc169646636)

[Ilustración 16 Ejecucion del comando yes 13](#_Toc169646637)

[Ilustración 17 Uso de grep yes 13](#_Toc169646638)

[Ilustración 18 Directorio para observar PCB 17](#_Toc169646639)

[Ilustración 19 Ver procesos con /proc/PID/status 18](#_Toc169646640)

[Ilustración 20 Creación del scritp para imprimir numeros 19](#_Toc169646641)

[Ilustración 21 Ejcucion del Script en segundo plano 20](#_Toc169646642)

[Ilustración 22 Confirmación de permiso para ejecución 20](#_Toc169646643)

[Ilustración 23 Lo que sucede en segundo plano desde el PIC 21](#_Toc169646644)

[Ilustración 24 Ejecución del script en primer plano 22](#_Toc169646645)

# OBJETIVOS

1.1 Familiarizarse con el sistema operativo Linux, la consola y sus instrucciones.

1.2 Ser capaces de entender la diferencia entre los comandos en Linux y Windows.

# INFORME

## 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*.

Texto

Descripción generada automáticamente

Ilustración 1 fecha y hora de la computadora

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

Descripción generada automáticamente

Ilustración 2 calendario septiembre 1983

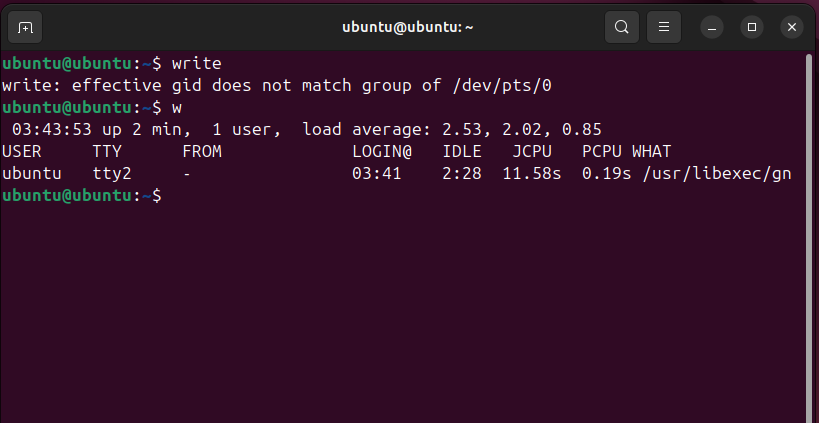
3. Verifique qué usuarios se encuentran dentro del sistema con el comando w.   


Ilustración 3 Usuarios del sistema

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



Ilustración 4 Comando para limpiar el terminal

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

Texto

Descripción generada automáticamente

Ilustración 5 Contenido de las variables de entorno

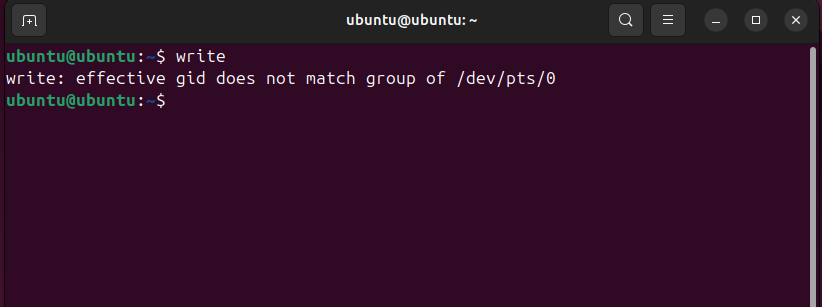
6. Muestre un mensaje de texto en la terminal de un usuario con el comando *write*.   


Ilustración 6 Mensaje de texto en la terminal comando write

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

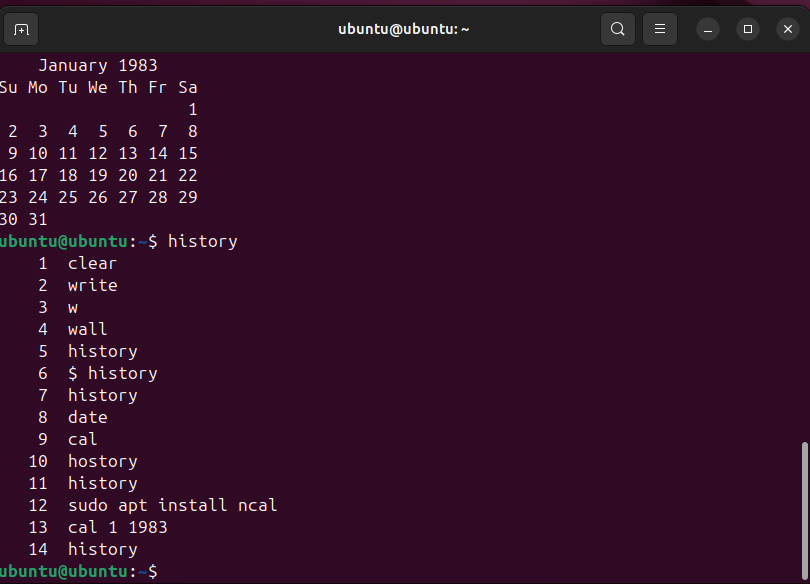


Ilustración 7 Historial de comandos

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).   
Texto

Descripción generada automáticamente

Ilustración 8 Fichero bash\_history con uso de cat

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

$ !-1

$ !! 10

$ !7

El comando !-1 ejecuta el último comando que se ejecutó en la terminal.

Texto

Descripción generada automáticamente

Ilustración 9 Comando !-1

El comando !! también ejecuta el último comando que se ejecutó en la terminal.  
Texto

Descripción generada automáticamente

Ilustración 10 Comando !!

El comando !7 ejecuta el séptimo comando en el historial de la sesión actual.  
Texto

Descripción generada automáticamente

Ilustración 11 Comando !7

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

****

Ilustración 12 Figura realizada con el comando Vi

**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***

El comando :1,$s /DE/SA/g en vi/Vim realiza una sustitución global en todo el archivo desde la primera hasta la última línea, reemplazando todas las instancias de "DE" por "SA".  
 **Comando y Modo Línea (:)**: El comando comienza con : para indicar que se está ejecutando en el Modo Línea de vi/Vim. Este modo permite ejecutar comandos específicos.

 **Rango (1,$)**: 1,$ especifica que el comando se aplicará desde la primera línea hasta la última línea del archivo.

 **Sustitución (s /DE/SA/g)**:

* s: Indica que se va a realizar una sustitución.
* /DE/SA/: Es la sintaxis para la sustitución. En este caso, se buscará la cadena "DE" y se reemplazará por "SA".
* g: Es una bandera opcional que significa "global", lo que indica que todas las apariciones de "DE" en cada línea serán reemplazadas, no solo la primera.

Diagrama, Texto

Descripción generada automáticamente

Ilustración 13 Ejecución del comando :1,$s /DE/SA/g

# 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. Utilice también *pstree –p* y *pstree -h*. ¿Cuál es la diferencia?

Pstree -p

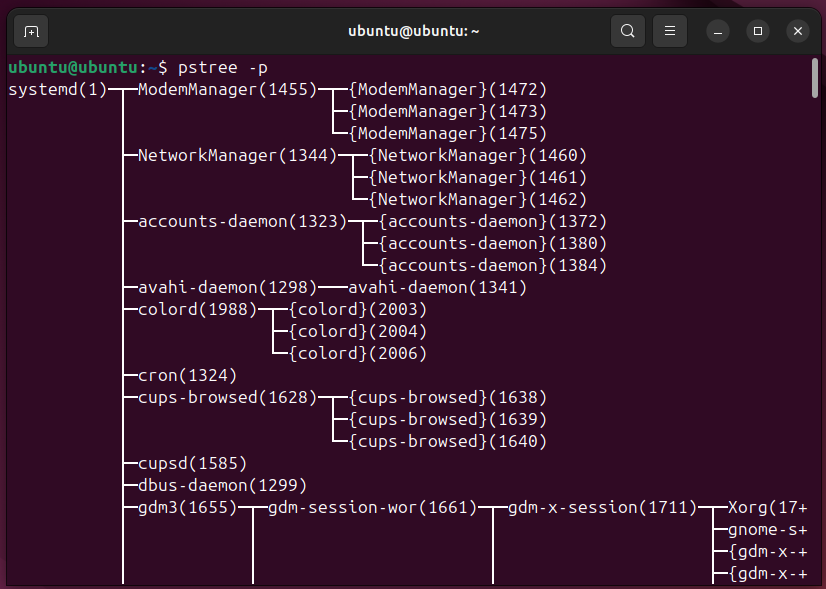


Ilustración 14 Comando Pstree -p

Pstree -h

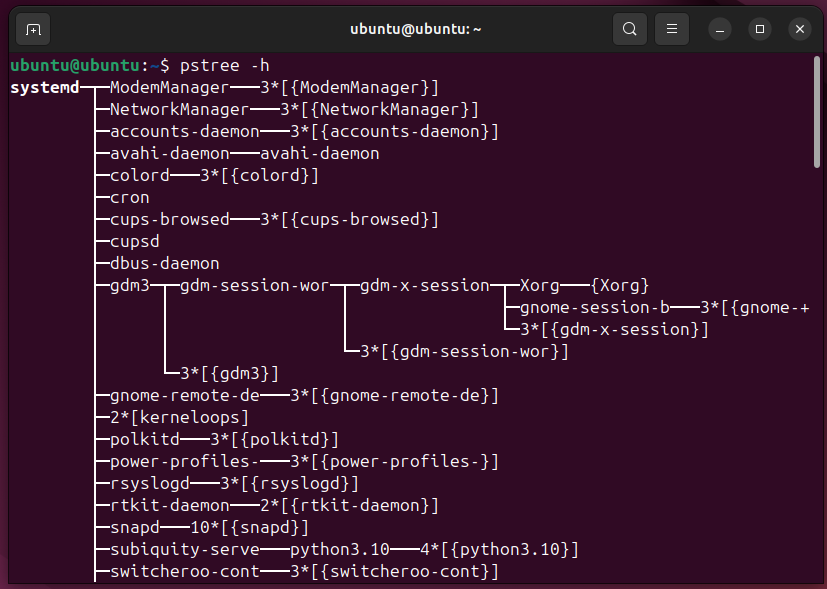
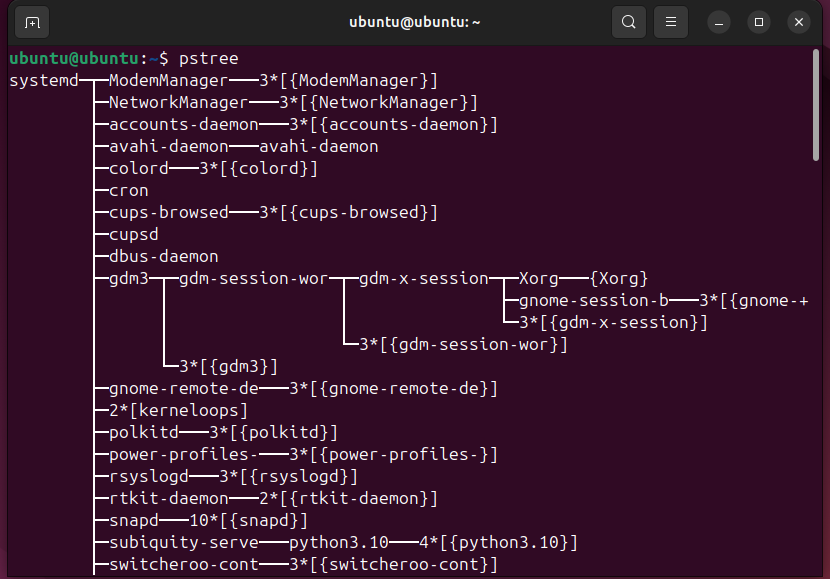


Ilustración 15 Comando pstree -h

Diferencia entre pstree -p y pstree -h

* pstree -p: Añade información sobre los PIDs de los procesos, lo que puede ser útil para identificar procesos específicos y sus relaciones.
* pstree -h: Resalta el proceso actual y sus ancestros, facilitando la identificación del proceso que estás ejecutando y cómo se relaciona con otros procesos en el sistema

Pstree



# 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.

# Hay varias opciones útiles en ps:

# ps aux: Muestra información de todos los procesos en el sistema.

# ps -ef: Otra forma de mostrar todos los procesos con una estructura de árbol.

# ps -o pid,ppid,cmd -C yes: Muestra los PID, PPID y el comando de los procesos específicos con el nombre yes.

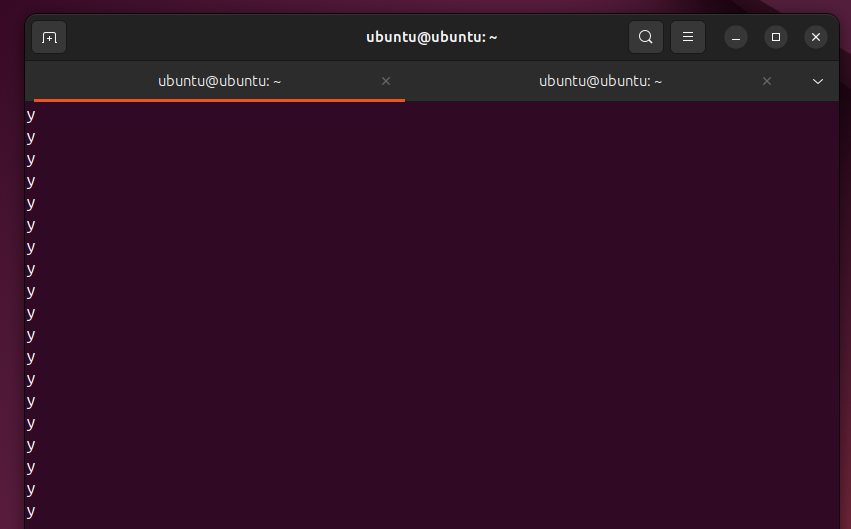


Ilustración 16 Ejecucion del comando yes

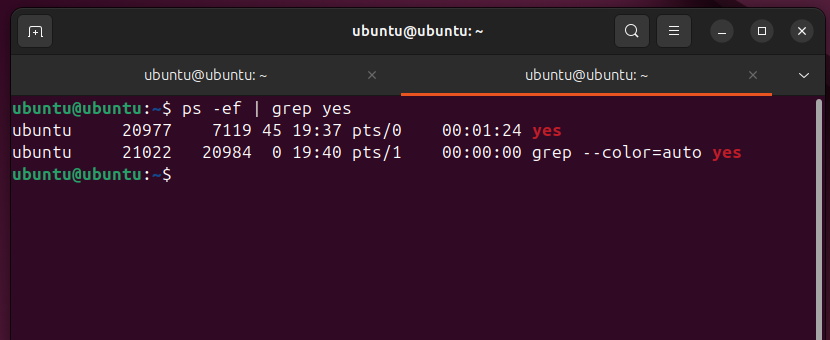


Ilustración 17 Uso de grep yes

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

* • F – Indicadores asociados con el proceso.
* • S – Estado del proceso.
* • UID – Identificación del usuario (ID) propietario del proceso.
* • CPU – Utilización del proceso.
* • PRI – Prioridad del proceso.
* • WCHAN – El suceso por el cual el proceso está esperando.
* • TTY – El Terminal que controla el proceso
* • TIME – Tiempo de ejecución acumulativa del proceso.

**Procesos Linux ps**

El comando ps es utilizado para mostrar de forma enumerada los procesos que se están ejecutando.

**Modificadores (Opciones):**

* A - muestra todos los procesos del sistema.
* u <nombre usuario> - El comando muestra los procesos del usuario, además puede mostrar los procesos de un usuario determinado.
* F - Indicadores asociados al proceso.
* s - Estado del proceso.
* ef - Los procesos con el ID de proceso y predecesor.
* I - permite contar los procesos.

**Elementos de consola:**

* UID – Identificación del usuario (ID) propietario del proceso.
* CPU – Utilización del proceso.
* PRI – Prioridad del proceso.
* WCHAN – El suceso por el cual el proceso está esperando.
* TTY – El Terminal que controla el proceso
* TIME – Tiempo de ejecución acumulativa del proceso.



#### Figura. - Ejecución comando yes.

Texto

Descripción generada automáticamente con confianza media

#### Figura. - Repetición del carácter “y”, asociado al comando yes.

Texto

Descripción generada automáticamente

#### Figura. – Ejecución comando para impresión de todos los procesos del sistema.

**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>.**

**Texto

Descripción generada automáticamente**

#### Figura. – Ejecución comando top

La utilización del comando *top*, presenta información detallada acerca de los procesos que realiza cada uno de los usuarios en tiempo real por lo que se va actualizando constantemente hasta que el usuario con ayuda de un teclado ingrese *Ctrl + C*, además esto se puede filtrar por usuario.

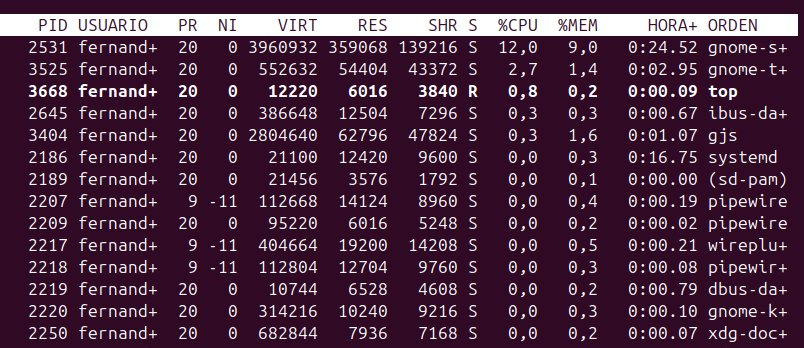


#### Figura. – Comando ingresado en consola.

Texto

Descripción generada automáticamente

#### Figura. – Resumen general de los procesos en ejecución y uso del hardware.



#### Figura. – Procesos en ejecución asociados al usuario Fernando-Huilca.

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

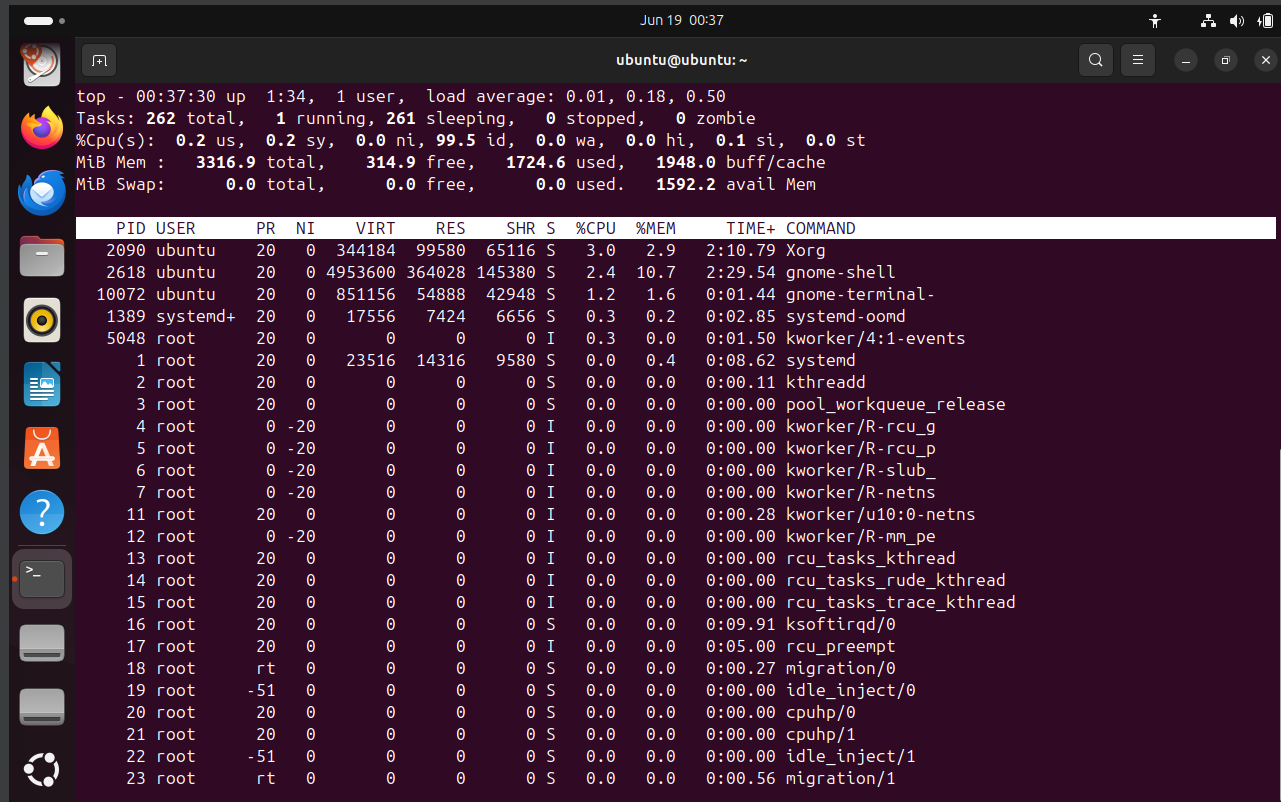
En sistemas Linux, el PCB (Process Control Block) de un proceso se puede observar en el directorio /proc, que es un sistema de archivos especial que presenta información sobre el sistema y los procesos. Cada proceso tiene un subdirectorio identificado por su PID, como /proc/<PID>. Dentro de este subdirectorio, se pueden encontrar archivos como stat y status que contienen los detalles del proceso. Por ejemplo, en /proc/<PID>/status, se puede encontrar información detallada sobre el proceso.  
Para saber cuáles son esos procesos podemos usar el comando top:   
****

Ilustración 18 Directorio para observar PCB

Luego, podemos veremos el estado de uno de esos procesos con /proc/PID/status:

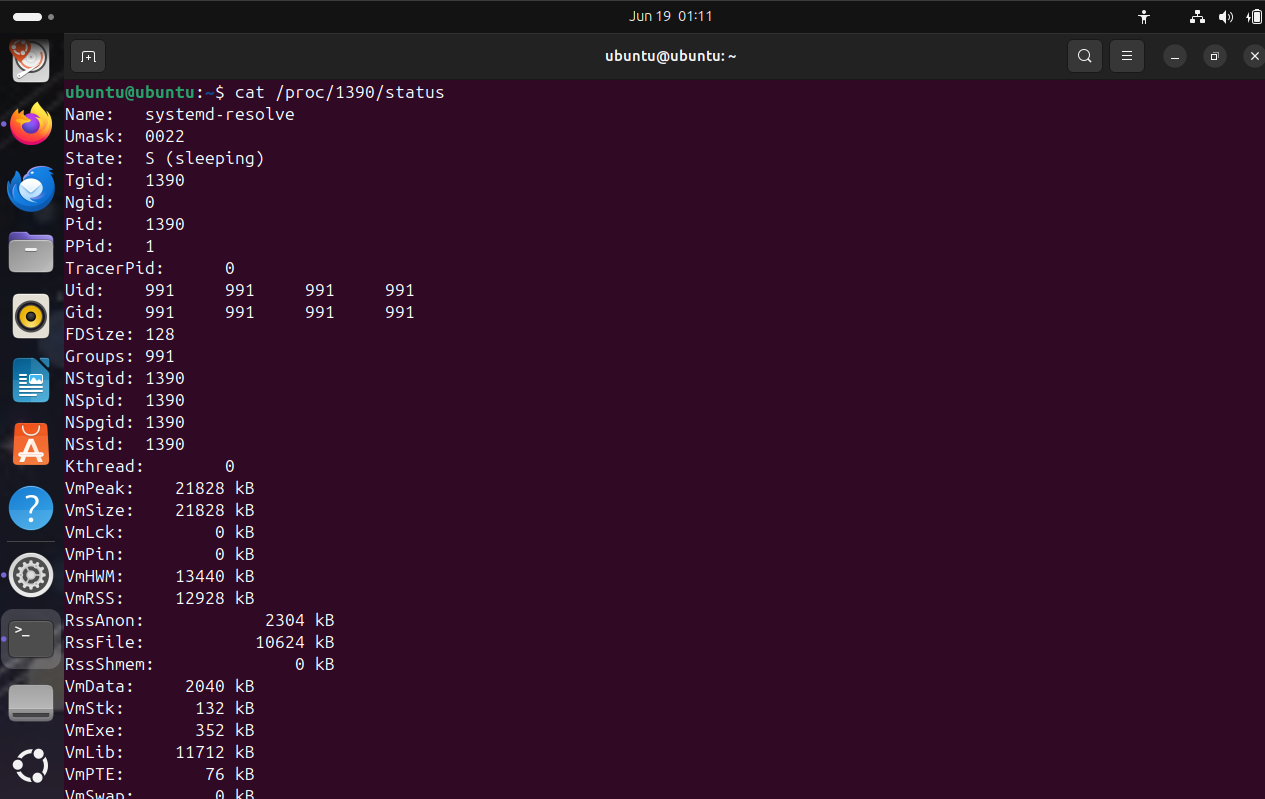


Ilustración 19 Ver procesos con /proc/PID/status

**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.**

Para esto vamos a:

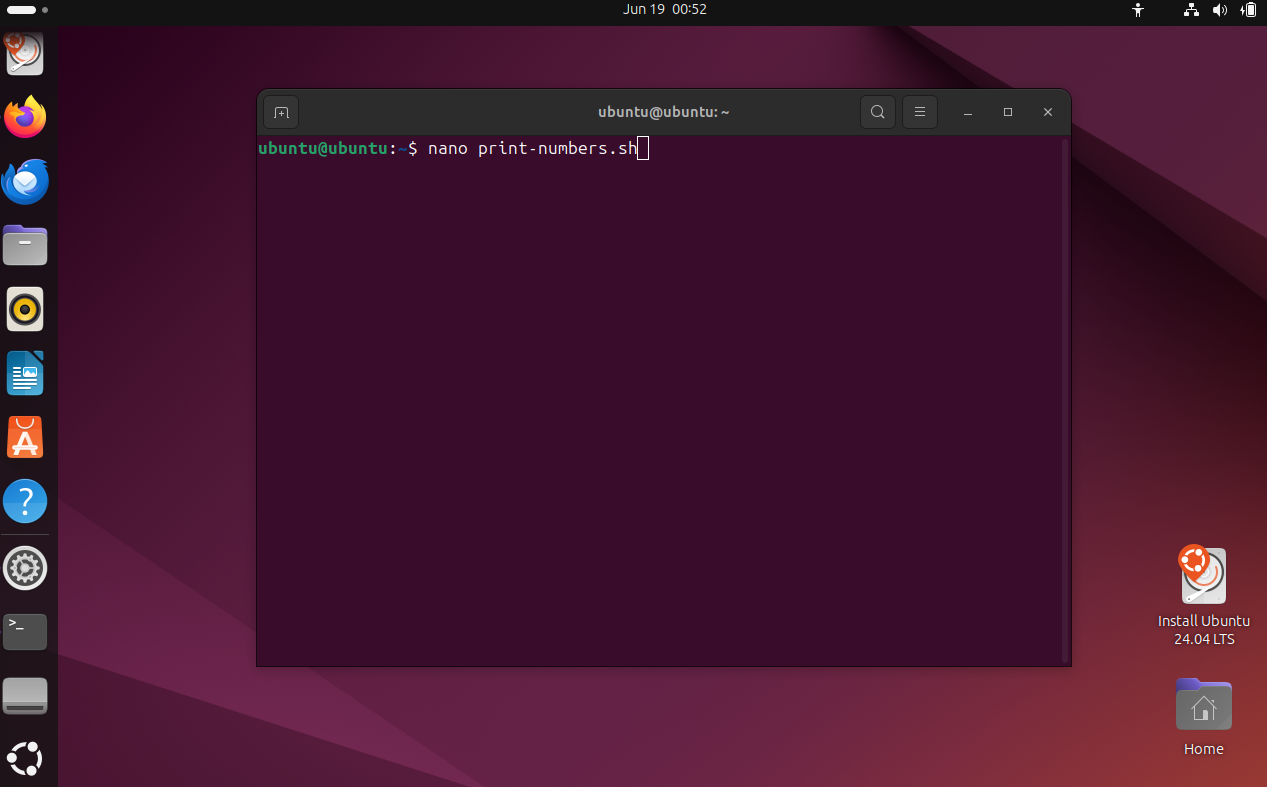
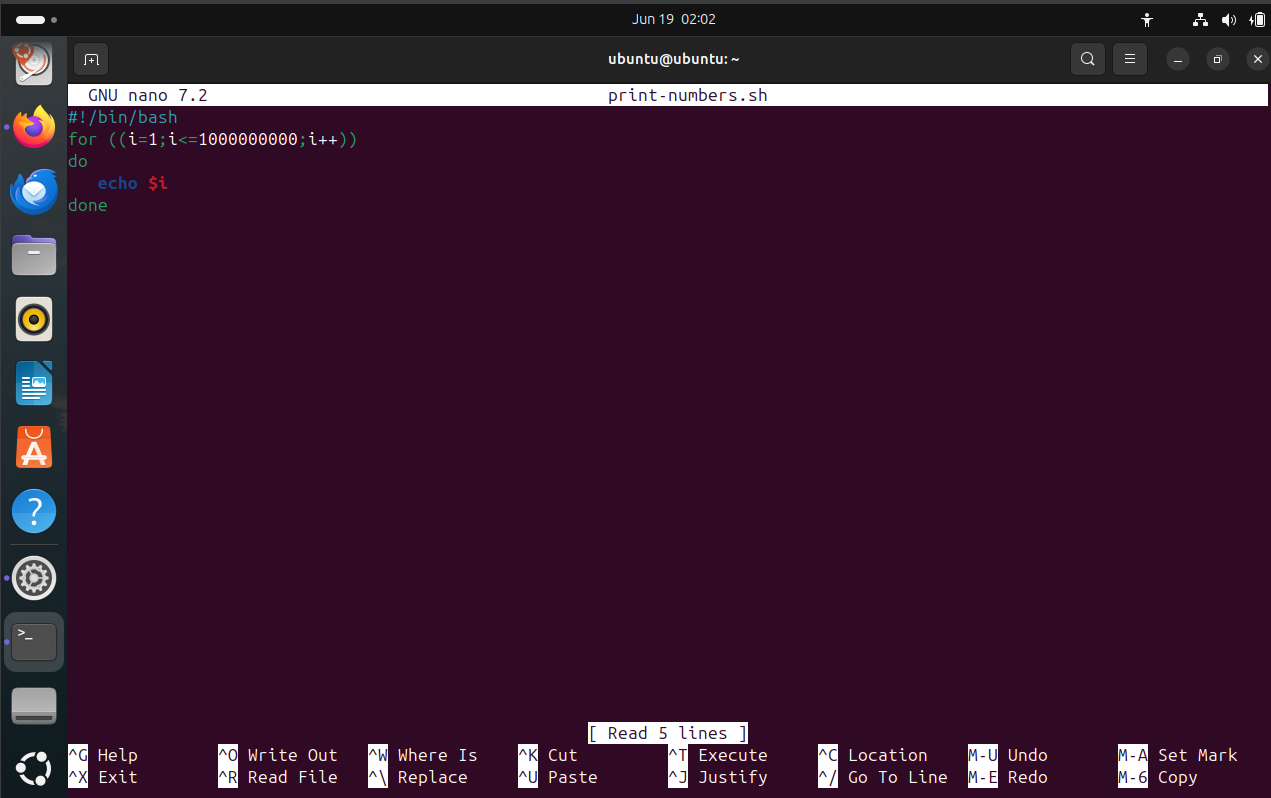
1. Crear un script en la terminal desde la máquina virtual de Ubuntu:  
   

Ilustración 20 Creación del scritp para imprimir numeros

1. 
2. Ejecutar el Script en segundo plano

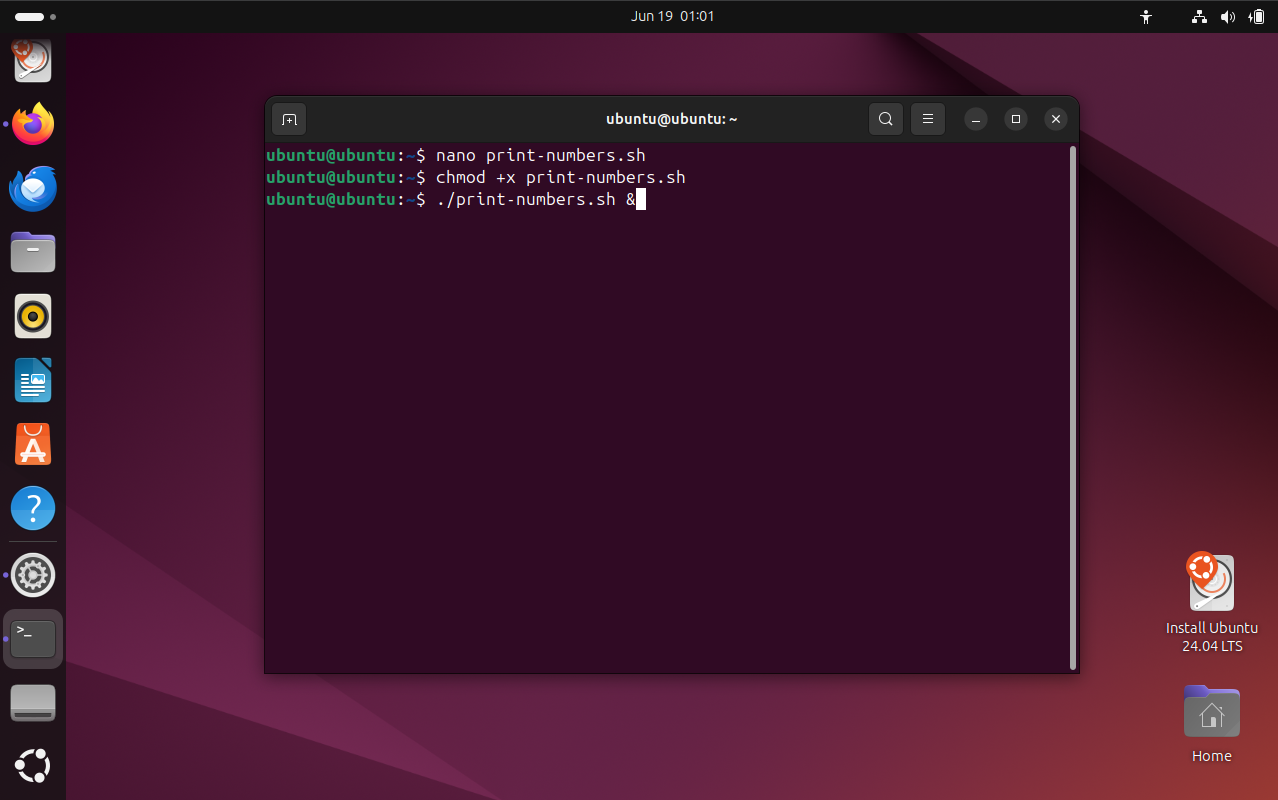


Ilustración 21 Ejcucion del Script en segundo plano

1. Veremos si podemos ejecutar el script, si tenemos permiso o no. De no tenerlo ejecutaremos el comando bash ./print-numbers.sh

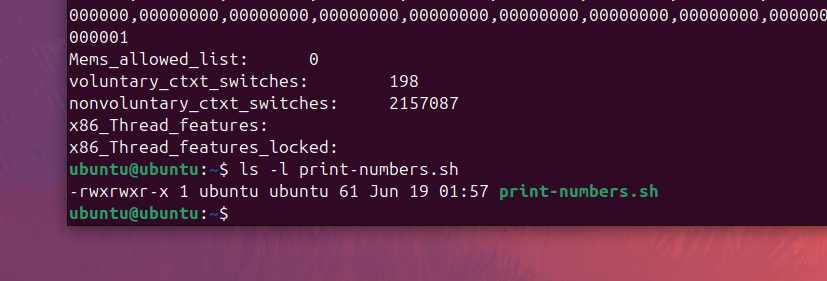


Ilustración 22 Confirmación de permiso para ejecución

1. Revisamos con cat /proc/PID/status cómo va la ejecución del programa

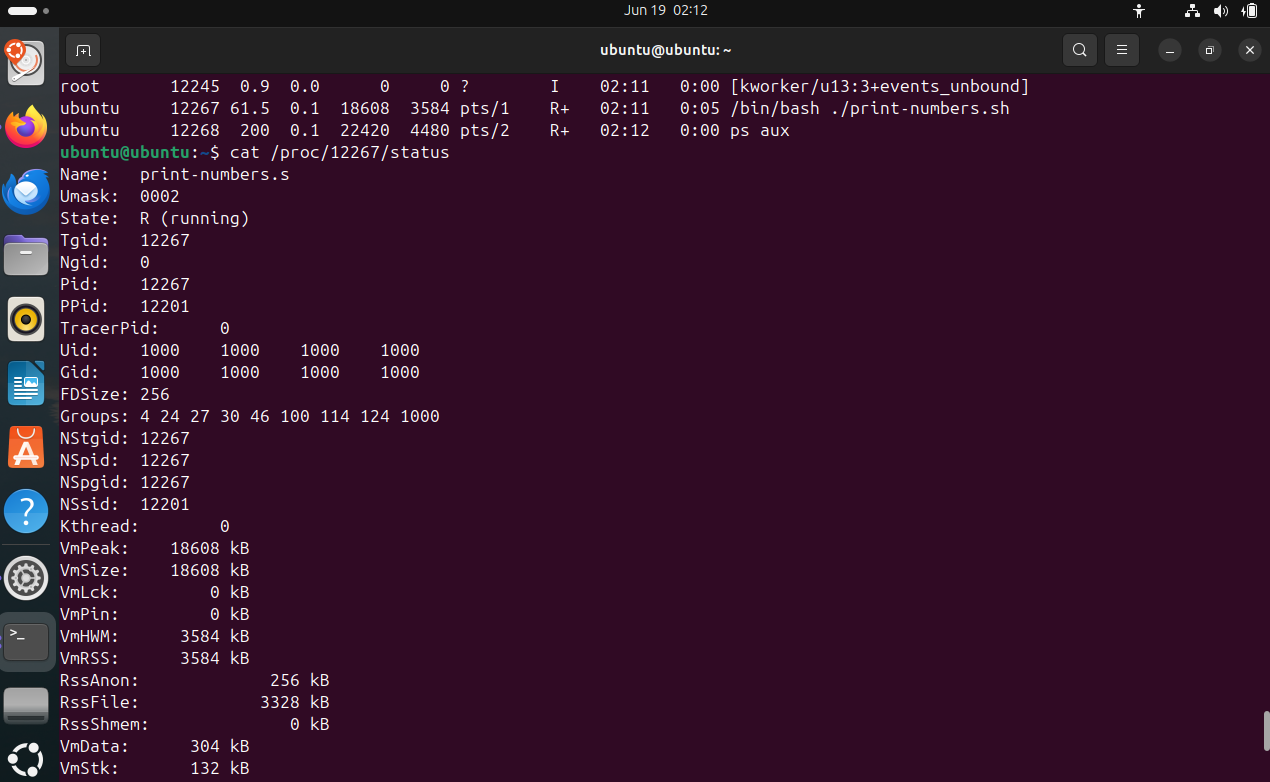
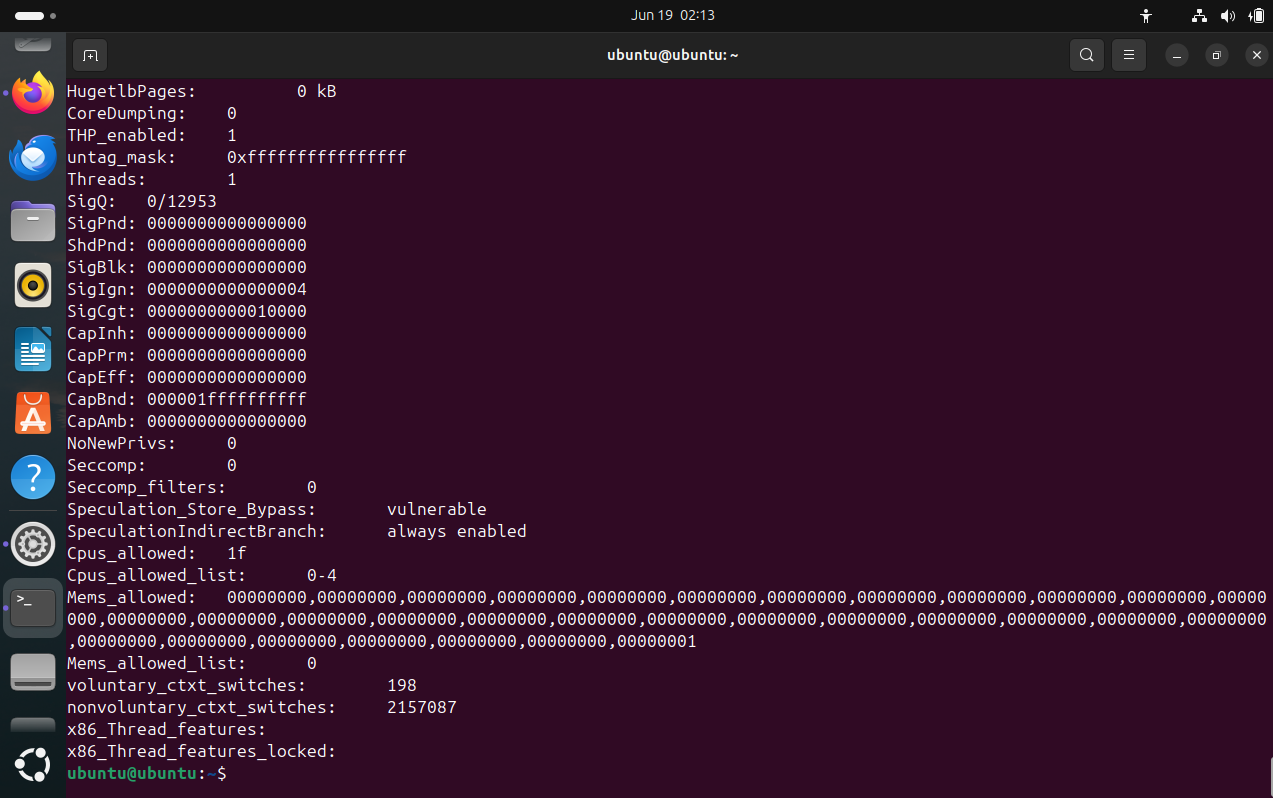


Ilustración 23 Lo que sucede en segundo plano desde el PIC



Podemos ver la ejecución en primer plano solo escribiendo ./print-numbers sin el & porque éste es el que lo hace ejecutarse en segundo plano.

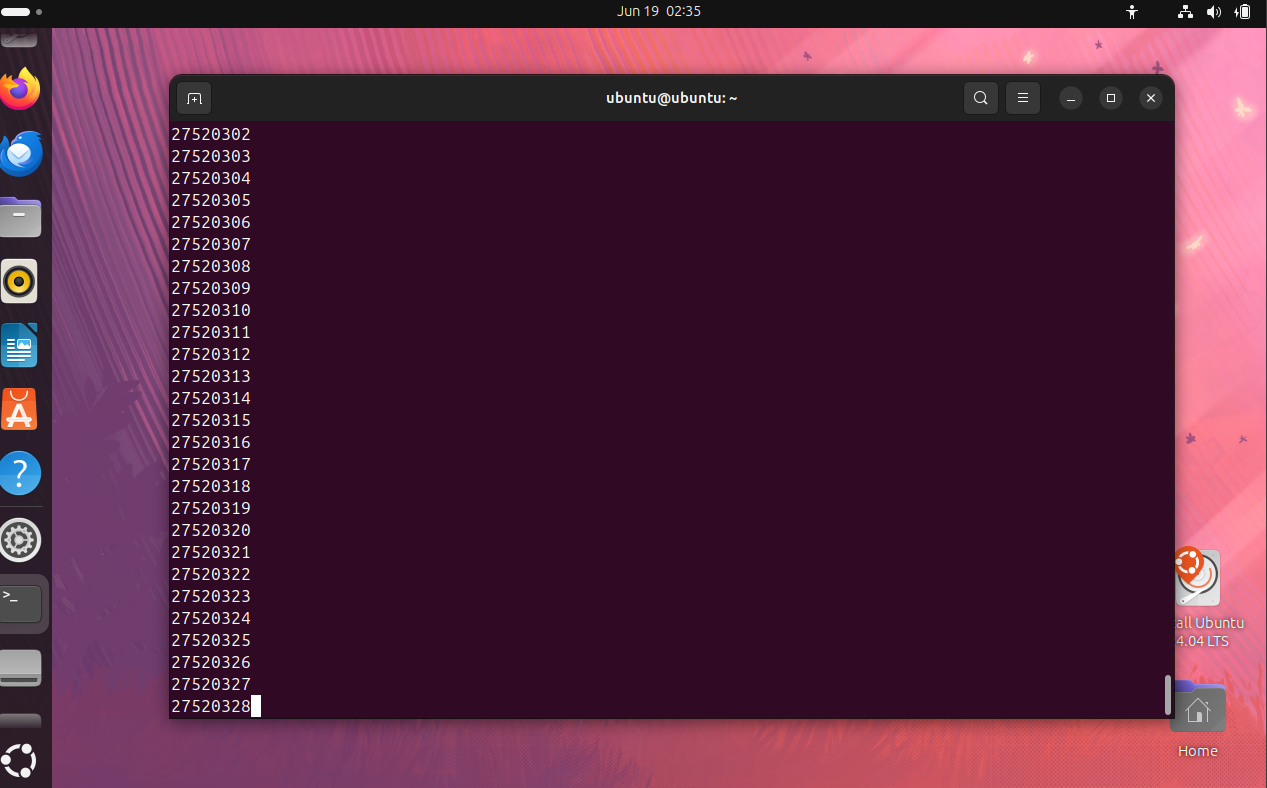


Ilustración 24 Ejecución del script en primer plano

# CONCLUSIONES Y RECOMENDACIONES

### **Conclusiones de la Práctica**

1. **Dominio de Comandos Básicos**: La práctica permitió adquirir un dominio sólido de los comandos básicos de la terminal de Ubuntu, esenciales para la administración eficiente del sistema operativo.
2. **Eficiencia Operativa**: Se observó una mejora significativa en la eficiencia al ejecutar tareas como la gestión de archivos, la navegación por directorios y la administración de procesos mediante comandos específicos.
3. **Aplicación Práctica de Conocimientos**: La experiencia práctica facilitó la aplicación de conceptos teóricos aprendidos en clase sobre sistemas operativos, reforzando la comprensión de su funcionamiento en entornos reales.

### **Recomendaciones**

1. **Práctica Continua**: Es recomendable continuar practicando regularmente los comandos aprendidos para mantener y mejorar las habilidades en la terminal de Ubuntu.
2. **Exploración de Funcionalidades Avanzadas**: Se recomienda explorar comandos más avanzados y sus aplicaciones para ampliar el conjunto de habilidades en la administración de sistemas.
3. **Uso de Recursos Adicionales**: Utilizar recursos adicionales como documentación oficial, tutoriales en línea y comunidades de usuarios para seguir aprendiendo y resolver dudas específicas.
4. **Experimentación Segura**: Realizar pruebas en entornos de desarrollo o máquinas virtuales para evitar riesgos en sistemas en producción mientras se experimenta con nuevos comandos y técnicas.

# BIBLIOGRAFÍA

[1] Hostinger. "Linux comandos." [En línea]. Disponible en: <https://www.hostinger.es/tutoriales/linux-comandos>. [Consulta: 18-jun-2024].

[2] Dell. "Introducción a los comandos básicos de solución de problemas en Ubuntu Linux." [En línea]. Disponible en: <https://www.dell.com/support/kbdoc/es-cl/000123974/introduccion-a-los-comandos-basicos-de-solucion-de-problemas-en-ubuntu-linux>. [Consulta: 18-jun-2024].