**Guía de Administración de Procesos en Linux**

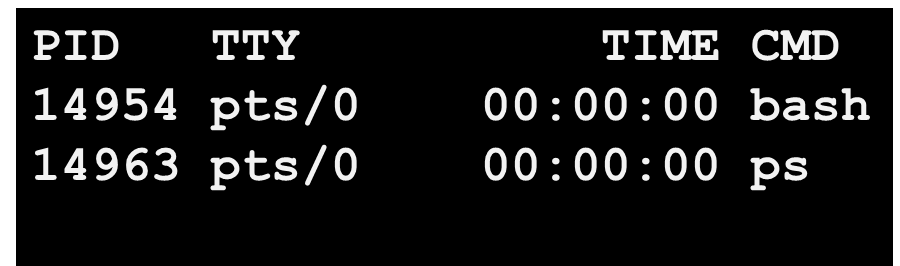
**Nombre: Benjamín Martínez Jeldres**

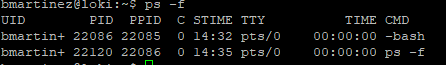
**Carrera= IECI**

**Instrucciones:**

* Ingrese con su cuenta al sistema Linux
* Ejecute las tareas indicadas
* Responda las preguntas

**Información sobre los procesos personales:**

* El comando para ver la información de los procesos es **ps**
* Ejecute **ps** y obtendrá algo como esto:  
  
* La información que se muestras es:
  + **PID**: Número que identifica al proceso
  + **TTY**: Terminal al cual el proceso está conectado
  + **TIME**: Tiempo acumulado de CPU en el formato hh:mm:ss
  + **CMD**: Nombre del programa ejecutable
* Ejecute el comando **ps** con la opción **–f**, es decir, **ps –f**  
  **¿Qué nuevas columnas aparecen?  
  Indique nombre y significado:**



**UID** = Representación a través de un identificador de usuario.

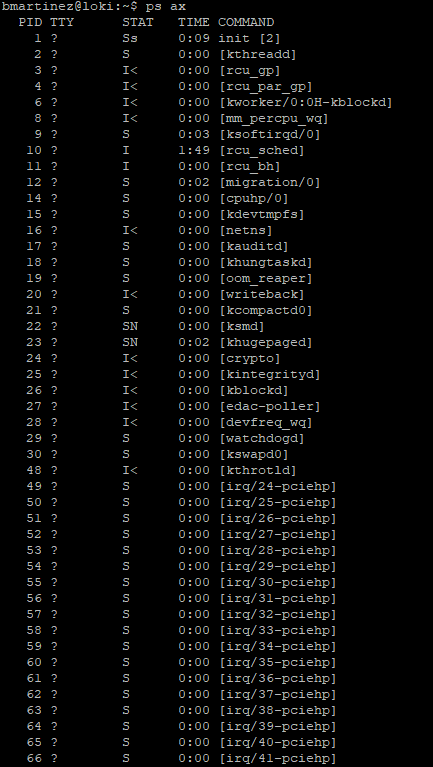
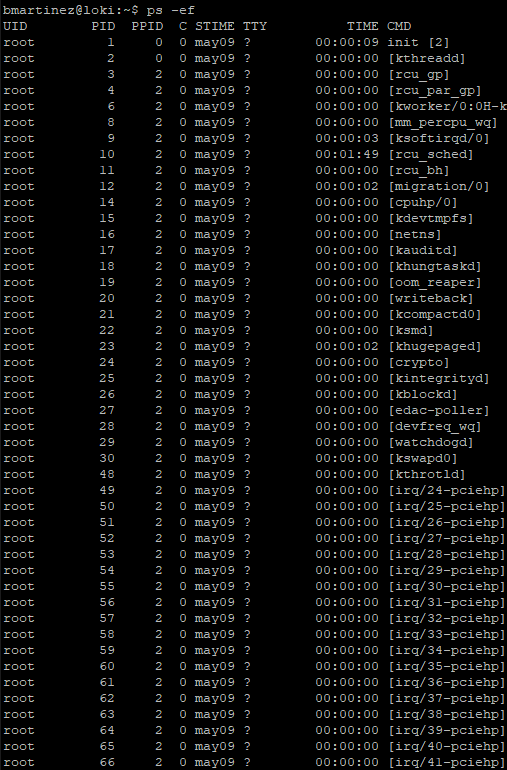
**PPID**= Identificador del proceso padre.

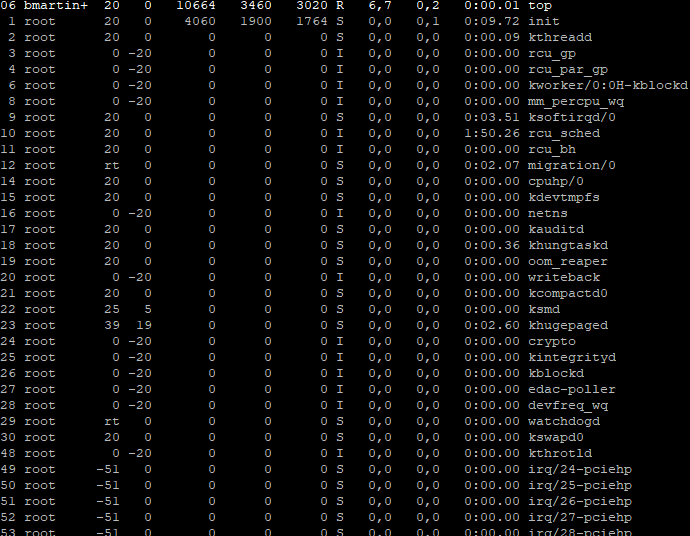
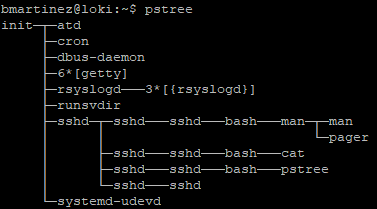
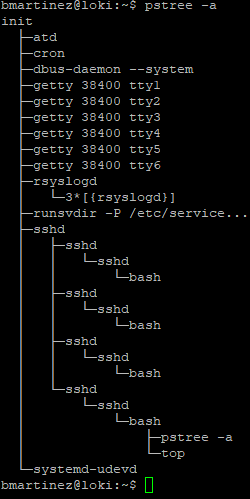
**C**=

**STIME**= Set Time establecerá la idea del sistema de la hora y la fecha.

**Información sobre los procesos del sistema:**

* Se pueden ver todos los procesos del sistema con alguna de estas dos opciones  
  **ps –ef  
  ps ax**  
  **¿Cuál opción encuentra más productiva? ¿Por qué?  
  Respuesta:**

****  
Encuentro mas productivo el proceso **ps -ef,** ya que, muestra más detalles respecto a todos los procesos.

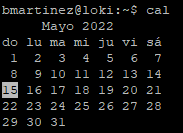
* También existen formas opcionales de cómo ver y monitorear procesos
  + Comando **top**
  + Comando **pstree o pstree –a   
      
    ¿Qué comando permite monitorear la carga de trabajo en tiempo real?  
    Respuesta:  
    top**
  +   
    **¿Qué comando permite ver las dependencias entre los procesos?  
    Respuesta:**  
    **pstree –a pstree**
  + 

**Procesos en primer plano o interactivos:**

* Son comandos que interactúan con el usuario o muestran información al usuario.
* Ejecute los siguientes comandos de ejemplos:
  + Ver mes actual: **cal**

**¿Qué muestra la ejecución del comando?** **cal 2022  
Respuesta:**

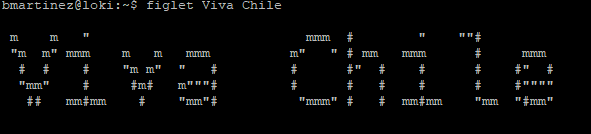
Muestra la fecha actual



* + Sumar 8 más 11 (es muy importante respetar los espacios): **expr 8 + 11**

# ¿Cómo se puede usar expr para obtener el resultado de multiplicar 5 por 45? Respuesta:

# expr 5 \\*45

* + Mostrar una vaca parlante: **cowsay HOLA**
  + ****
  + Mostrar un mensaje más grande: **figlet Viva Chile**
  + 

**Procesos en segundo plano:**

* Son procesos que se ejecutan sin la interacción de un usuario o sin hacer uso del terminal
* Comando para esperar un tiempo: **sleep**  
  Ejemplo de uso (esperar 10 segundos): **sleep 10**
* Comando para desplegar un texto: **echo**  
  Ejemplo: **echo "Hola a todos"**
* Encadenar comandos en secuencia con el uso del punto y coma.  
  Ejemplo: **sleep 10; echo "Tiempo"**
* Tarea en ejecución en segundo plano con uso de los metacaracteres () y &.  
  Ejemplo: **(sleep 10; echo "Tiempo")&**
* Comando para ver los trabajos en segundo plano: **jobs**  
  Ejemplo, ejecutar la siguiente secuencia de comandos:   
  **jobs**  
  no debiera mostrar nada, pues no hay tareas en segundo plano  
  Ejecutar un comando y dejarlo en segundo plano, como:  
  **cat &**  
  el resultado será algo como
* **[1] 23433**
* **¿Qué es el valor numérico 23433 (en el ejemplo el valor después de [1]?  
  Respuesta:**
* Es el **PID** del Proceso

Si se presiona la tecla **enter** en la shell debería aparecer:  
**[1]+ Detenido cat**  
Y se usa ejecuta ahora **jobs** se verá el comando en segundo plano, algo como:  
**[1]+ Detenido cat**

* Se puede traer a primer plano el comando en segundo plano usando el comando **fg**  
  Ejemplo, traer a primer plano el comando **cat** anterior  
  **fg**  
  el resultado es   
  **cat**  
  pues la shell espera algún parámetro.  
  Aborte la ejecución de **cat** usando la secuencia **control+c**Use **jobs** nuevamente y verá que no hay ninguna tarea en segundo plano
* Se puede abortar casi cualquier comando en ejecución con la secuencia **control+c**  
  Ejemplo: el comando interactivo **rev** invierte una palabra, por ejemplo ejecute  
  **rev**  
  quedará esperando palabras para invertir, escriba algo como **Hola mundo** y saldrá:  
  **odnum aloH**  
  termine la ejecución del comando con **control+c** ó **control+d  
  ¿Cuál es la diferencia entre control+c y control+d?  
  Respuesta:  
  control+c=** envia señal a todos los procesos en primer plano, a menudo para finalizar el programa en ejecución

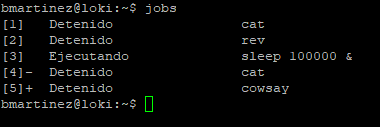
**control+d=** En vez de enviar una señal a los procesos, este representa un valor binario especial para terminar el proceso.

* Cualquier comando interactivo se puede pasar a segundo plano con la secuencia **control+z**  
  Ejemplo, ejecute **rev** nuevamente  
  **rev**  
  escriba algunas palabras para probar que funciona y luego en cualquier momento haga **control+z**aparecerá   
  **[1]+ Detenido rev**  
  Vea con **jobs** que el proceso está en segundo plano.  
  Vea con **ps** que el comando está en segundo plano y puede ver su **PID**  
  Traiga el comando a primer plano con **fg**  
  Aborte el comando con **control+c**

**Ejercicio:**

* Dejar varios comando detenidos en segundo plano, por ejemplo:  
  **rev&  
  sleep 10000&  
  cat&  
  cowsay&**
* Verificar que están en segundo plano con **ps** y **jobs**.

**¿Cuál de los comandos en segundo plano sigue en ejecución? ¿Cómo lo pudo saber?  
Respuesta:**  
**el comando sleep sigue en ejecución, lo pude saber usando el comando Jobs**

****

* Traer a primer plano el último de los comandos en segundo plano  
  **fg**  
  traerá **cowsay** a primer plano  
  Escriba un mensaje, después **enter** y termine con la combinación **control+d**  
  aparecerá la vaca parlante con su mensaje y se terminará el programa.
* Quedan aún tres comandos en segundo plano. Verificar con **jobs**.
* Puede traer el comando en segundo plano que desee usando el número de la primera columna de la salida con Jobs. Por ejemplo, **rev** tiene el número 1.
* Traer a primer plano al comando **rev**  
  **fg 1**  
  escriba unos mensajes para ver que **rev** funciona y cuando quiera terminar use **control+d** o **control+c**
* Elimine los dos comandos restantes que están en segundo plano con lo aprendido.
* Probar lo siguiente: Dejar en segundo plano varios procesos:  
  **sleep 2000&  
  cat&  
  cowsay&  
  rev&**  
  **¿Se puede traer a primer plano alguno de esos procesos, usando el comando fg y el nombre del programa?**  
  **Respuesta:**

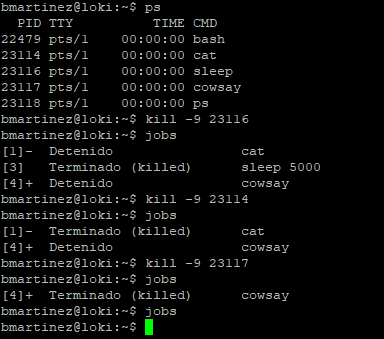
Si se puede, incluso solo con la primera letra del nombre del proceso al que queremos buscar mientras no haya otros procesos que empiecen igual.

**fg Inicial o nombre completo del proceso + %**



**Envío de señales:**

* Se pueden enviar señales a los procesos.
* El comando para eso es **kill**
* Probemos primero dejando varios comandos en segundo plano  
  **cat&  
  cownsay&  
  sleep 5000&**
* Si se ejecuta **ps** podrá ver los tres comandos y sus **PID**
* Para eliminar un comando puede usar el comando **kill** y la opción **-9** más el **PID** del proceso a eliminar  
  Por ejemplo:  
  **kill -9 34112**  
  eliminará el proceso con el **PID 34112**
* Elimine los tres procesos en segundo plano usando **kill -9**

****

* Después de cada **kill** podrá ver un mensaje como por ejemplo  
  **[1]+ Terminado (killed) cat**

**¿Cómo se puede eliminar más de un proceso con un sólo comando kill?. Ejemplifique  
Respuesta:**

Agregando el PID de varios procesos, ej.:

Kill -9 PID1 PID2 PID3

**Agendamiento sencillo de una tarea:**

* Agendar es programar la ejecución de una tarea para el futuro.
* Por ejemplo que me avise con un mensaje que es hora de almuerzo en una hora más.  
  **(sleep 3600; echo "Es hora de almorzar";echo "vamos terminando!!")&**
* Pruebe el comando anterior con 10 segundos aproximadamente para evitar esperar una hora.  
  Espere los 10 segundos en la Shell y verá los dos mensajes.
* 