



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**

**FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA PROGRAMA DE  
INGENIERO EN SOFTWARE Y TECNOLOGÍAS EMERGENTES**

**Administración de Sistemas Operativos (351)**

# **Taller 6. Procesos**

**16 de Octubre 2023**

---

**Docente:**

Alma Leticia Palacios Guerrero

**Participante(es):**

Luis Eduardo Galindo Amaya (1274895)

# Índice

<b>1. Introduccion</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>3</b>
2.1. Genere un listado completo de todos los procesos que están en el sistema y muestre la información completa de todos los que se empezaron a ejecutar el 7 de septiembre en una sola línea. . . . .	3
2.2. Qué están haciendo los procesos que actualmente esta ejecutando maestro. (Comando) . . . . .	4
2.3. Genere un listado con el número de proceso, número del proceso padre, comando en ejecución y prioridad de tres de sus compañeros. . . . .	4
2.4. Explique la diferencia entre las opciones de ps e,f,l y j . . . . .	5
2.5. Explique la diferencia entre las opciones de ps a y u . . . . .	5
2.6. Explique qué es lo que hace la opción de ps t y u Si tiene dos sesiones de ssh abiertas con el mismo user name . . . . .	6
2.7. ¿qué procesos muestra al ejecutar ps? . . . . .	7
2.8. ¿Qué opción de ps debería de usar par . . . . .	7
2.9. ¿Cómo identifico a los procesos que el usuario está ejecutando en cada terminal?	8
2.10. ¿Cuál es significado de TODAS las columnas de formato que maneja ps -o? (Sólo las que no están explicadas en este material). . . . .	8
2.11. Ejecute dos comandos en background (los que quiera). . . . .	10
2.12. Ejecute el comando cat >lista, ¿Qué prioridad tiene asignada? . . . . .	10
2.13. Mate el proceso anterior. . . . .	11
2.14. Vuelva a ejecutar cat>lista pero con menor prioridad. . . . .	11
2.15. Una vez más ejecute cat>lista, pero ahora en el background . . . . .	12
2.16. ¿Cuál es su prioridad ahora? . . . . .	12
2.17. Verifique que el comando en background este en la lista de procesos. . . . .	12
2.18. Verifique que el comando en background este en la lista de tareas (jobs). . . . .	13
2.19. Pase una de las tareas al foreground (use el número de tarea) . . . . .	13
2.20. Pase la otra tarea al foreground, pero ahora use el número de PID. . . . .	13
2.21. Envíe otro comando al background. . . . .	13
2.22. Finalice este proceso. . . . .	14
<b>3. Conclusión</b>	<b>14</b>
<b>4. Fuentes</b>	<b>14</b>

**Universidad Autónoma de Baja California**  
**Facultad de ciencias químicas e ingeniería**

Ingeniero en software y tecnologías emergentes

**Información de la materia**

**Nombre de la materia y clave:** Administración de Sistemas Operativos

**Grupo y periodo:** 351 (2023-2)

**Profesor:** Alma Leticia Palacios Guerrero.

**Información de la actividad**

**Nombre de la actividad:** Taller 6. Procesos

**Lugar y fecha:** 16 de Octubre 2023

**Carácter de la actividad:** Individual.

## **1. Introduccion**

A lo largo de esta practica aprenderemos que son los procesos dentro del los sistemas operativos, como funcionan para que sirven y como podemos utilizarlos de manera efectiva para lograr diversos tipos de tareas.

## 2. Desarrollo

2.1. Genere un listado completo de todos los procesos que están en el sistema y muestre la información completa de todos los que se empezaron a ejecutar el 7 de septiembre en una sola línea.

```
1 ps -ef | grep "Sep 7"
```



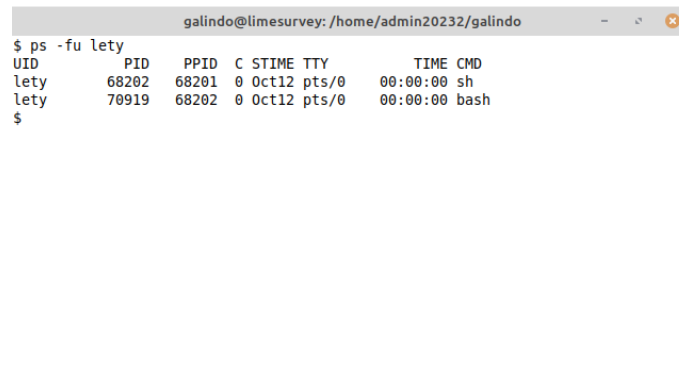
```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps -ef | grep "Sep 7"
galindo  92455  92434  0 00:53 pts/1    00:00:00 grep --color=auto Sep 7
galindo@limesurvey ~$
```

Figura 1: Procesos iniciados el 7 de Septiembre

- **ps** Para mostrar los procesos del usuario:
  - **-e** Lista información sobre cada proceso del sistema
  - **-f** muestra los detalle de los procesos
- **grep** para buscar el string de la fecha

## 2.2. Qué están haciendo los procesos que actualmente esta ejecutando maestro. (Comando)

```
1 ps -fu lety
```



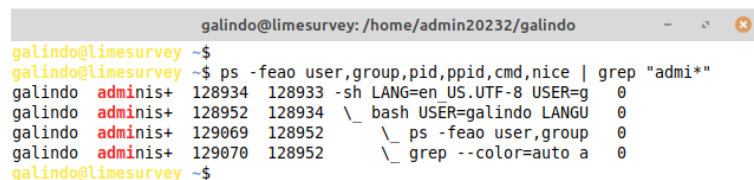
```
galindo@limesurvey: /home/admin20232/galindo
$ ps -fu lety
UID          PID    PPID  C  STIME TTY          TIME CMD
lety         68202   68201  0  Oct12 pts/0    00:00:00 sh
lety         70919   68202  0  Oct12 pts/0    00:00:00 bash
$
```

Figura 2: Nada solo aparecen que tiene sh abierto

## 2.3. Genere un listado con el número de proceso, número del proceso padre, comando en ejecución y prioridad de tres de sus compañeros.

```
1 ps -feao user,group,pid,ppid,cmd,nice | grep "admi*"
```

- f Muestra una lista completa de procesos con detalles y usuario.
- e Lista información sobre cada proceso en ejecución ahora.
- a Muestra los procesos de otros usuarios.
- o Muestra la información de acuerdo a un formato especificado.



```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$
galindo@limesurvey ~$ ps -feao user,group,pid,ppid,cmd,nice | grep "admi*"
galindo adminis+ 128934 128933 -sh LANG=en_US.UTF-8 USER=g 0
galindo adminis+ 128952 128934 \_ bash USER=galindo LANGU 0
galindo adminis+ 129069 128952 \_ ps -feao user,group 0
galindo adminis+ 129070 128952 \_ grep --color=auto a 0
galindo@limesurvey ~$
```

Figura 3: Solo estoy yo en el servidor

**2.4. Explique la diferencia entre las opciones de ps e,f,l y j**

- e** Lista información sobre cada proceso en ejecución ahora.
- f** Muestra una lista completa de procesos con detalles y usuario.
- l** Genera una lista con información detallada de los procesos.
- j** La información se presenta empezando por el PID

**2.5. Explique la diferencia entre las opciones de ps a y u**

- a** muestra los procesos de todos los usuarios.
- u** muestra procesos de un usuario específico.

## 2.6. Explique qué es lo que hace la opción de ps t y u Si tiene dos sesiones de ssh abiertas con el mismo user name

### Flag -t

Lista los procesos asociados con la terminal. Por ejemplo, term/a, o pts/0, si tenemos mas de una terminal y solo usamos **-t** mostrara los procesos de esa terminal.

```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps -t
  PID TTY          STAT TIME COMMAND
 128934 pts/2    Ss      0:00 -sh
 128952 pts/2    S       0:00 bash
 130200 pts/2    R+      0:00 ps -t
galindo@limesurvey ~$
```

Figura 4: salida de ps -t

### Flag -u

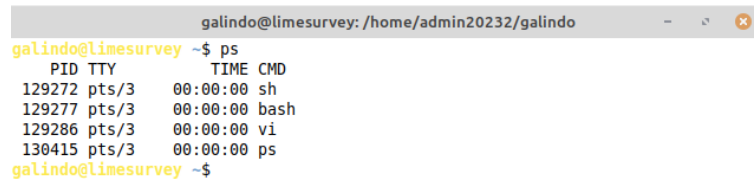
el usuario visualiza los procesos del usuario independientemente de la terminal que use.

```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
galindo  128934  0.0  0.0  2480  1632 pts/2    Ss   19:48   0:00 -sh
galindo  128952  0.0  0.0  5176  4500 pts/2    S    19:50   0:00 bash
galindo  129272  0.0  0.0  2480  1664 pts/3    Ss   20:10   0:00 -sh
galindo  129277  0.0  0.0  5308  4564 pts/3    S+   20:10   0:00 bash
galindo  129286  0.0  0.0  4936  3360 pts/3    T    20:10   0:00 vi
galindo  130202  0.0  0.0  6756  2992 pts/2    R+   21:37   0:00 ps -u
galindo@limesurvey ~$
```

Figura 5: Salida de ps -u

## 2.7. ¿qué procesos muestra al ejecutar ps?

El comando ps reporta el estado de los procesos activos



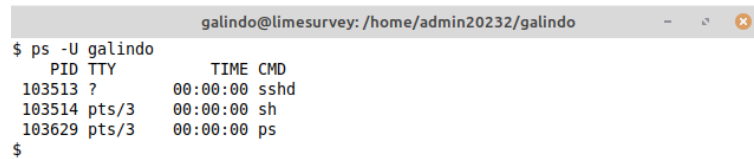
```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps
  PID TTY          TIME CMD
 129272 pts/3    00:00:00 sh
 129277 pts/3    00:00:00 bash
 129286 pts/3    00:00:00 vi
 130415 pts/3    00:00:00 ps
galindo@limesurvey ~$
```

Figura 6: Procesos activos en mi usuario

## 2.8. ¿Qué opción de ps debería de usar para

a ver todos los procesos de un usuario?

```
1 ps -U <nombre del usuario>
```



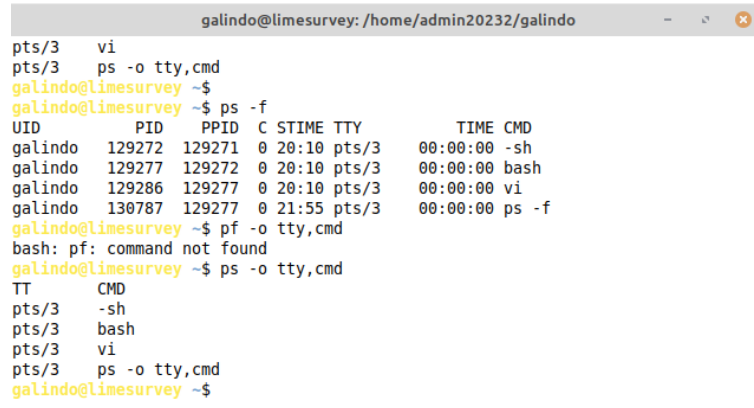
```
galindo@limesurvey: /home/admin20232/galindo
$ ps -U galindo
  PID TTY          TIME CMD
 103513 ?      00:00:00 sshd
 103514 pts/3    00:00:00 sh
 103629 pts/3    00:00:00 ps
$
```

Figura 7: Procesos activos en mi usuario



## 2.9. ¿Cómo identifico a los procesos que el usuario está ejecutando en cada terminal?

- Utilizando la columna TTY en `ps -f` o también utilizando `ps -o tty,cmd`



```
galindo@limesurvey: /home/admin20232/galindo
pts/3 vi
pts/3 ps -o tty,cmd
galindo@limesurvey ~$
galindo@limesurvey ~$ ps -f
UID      PID     PPID    C  STIME TTY      TIME CMD
galindo  129272  129271  0  20:10 pts/3    00:00:00 -sh
galindo  129277  129272  0  20:10 pts/3    00:00:00 bash
galindo  129286  129277  0  20:10 pts/3    00:00:00 vi
galindo  130787  129277  0  21:55 pts/3    00:00:00 ps -f
galindo@limesurvey ~$ pf -o tty,cmd
bash: pf: command not found
galindo@limesurvey ~$ ps -o tty,cmd
TT      CMD
pts/3   -sh
pts/3   bash
pts/3   vi
pts/3   ps -o tty,cmd
galindo@limesurvey ~$
```

Figura 8:

## 2.10. ¿Cuál es significado de TODAS las columnas de formato que maneja `ps -o?` (Sólo las que no están explicadas en este material).

(«ps Command», s.f.)

**ruser** Indica el ID de usuario real del proceso. Se muestra el ID de usuario en formato de texto. Si no se puede obtener el ID de usuario en formato de texto, se utiliza una representación decimal. El encabezado predeterminado para este campo es RUSER.

**rgroup** Indica el ID de grupo real del proceso. Se muestra el ID de grupo en formato de texto. Si no se puede obtener el ID de grupo en formato de texto, se utiliza una representación decimal. El encabezado predeterminado para este campo es RGROUP.

**ruid** Indica el número de ID de usuario real del proceso en formato decimal. El encabezado predeterminado para este campo es RUID.

**gid** Indica el número de ID de grupo efectivo del proceso en formato decimal. El nombre de inicio de sesión se imprime bajo la opción -f.

**rgid** Indica el número de ID de grupo real del proceso en formato decimal. El encabezado predeterminado para este campo es RGID.

**pid** Indica el valor decimal del ID de proceso. El encabezado predeterminado para este campo es PID.

**ppid** Indica el valor decimal del ID de proceso principal (padre). El encabezado predeterminado para este campo es PPID.

**pgid** Indica el valor decimal del ID de grupo de procesos. El encabezado predeterminado para este campo es PGID.

**sid** Indica el ID de proceso del líder de sesión. El encabezado predeterminado para este campo es SID.

**pcpu** Indica la proporción de tiempo de CPU utilizado en relación al tiempo de CPU disponible, expresada como porcentaje. El encabezado predeterminado para este campo es %CPU.

**pmem** Indica el porcentaje de memoria real utilizada por este proceso. El encabezado predeterminado para este campo es %MEM.

**vsz** Indica, en formato decimal, el tamaño en kilobytes de la imagen base del núcleo del proceso. El encabezado predeterminado para este campo es VSZ.

**rss** (bandera v) El tamaño de memoria real (conjunto residente) del proceso (en unidades de 1 KB).

**nice** Indica el valor decimal del valor "nice" del proceso. El encabezado predeterminado para este campo es NI.

**class** Indica la política de programación para un hilo de kernel. Las políticas son sched\_other, sched\_fifo y sched\_rr. El encabezado predeterminado para este campo es CLS.

**time** Indica el tiempo acumulado de CPU desde que el proceso se inició. El tiempo se muestra en el mismo formato que en etime. El encabezado predeterminado para este campo es TIME.

**etime** Indica el tiempo transcurrido desde que el proceso se inició.

**stime** La hora de inicio del proceso. Las variables de entorno LANG controlan la apariencia de este campo.

**lwp** El tiempo de ejecución de un hilo de ejecución ligero individual.

**nlwp** Indica el número de hilos de kernel propiedad del proceso. El encabezado predeterminado para este campo es THCNT.

**psr** El número de procesador lógico al que está vinculado el hilo de kernel (si lo tiene). Para un proceso, este campo se muestra si todos sus hilos están vinculados al mismo procesador.

**tty** La terminal de control para el proceso.

**addr** Contiene el número de segmento de la pila del proceso, si es normal; si es un proceso de kernel, la dirección del área de datos previos al procesamiento.

**wchan** Indica el evento por el que el proceso o hilo de kernel está esperando o durmiendo. Para un hilo de kernel, este campo está en blanco si el hilo de kernel está en ejecución.

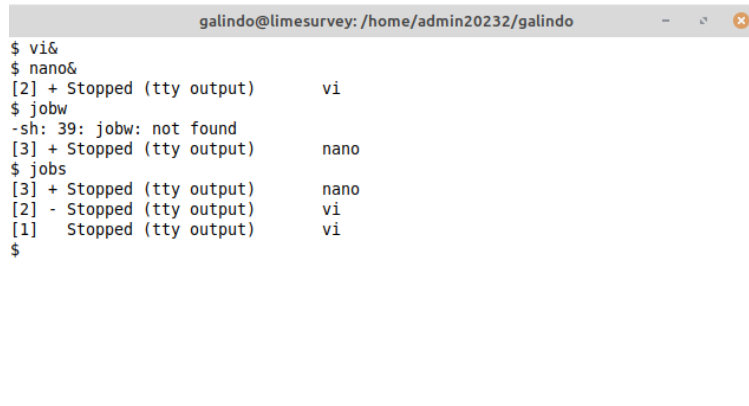
**fname** Indica los primeros 8 bytes del nombre base del archivo ejecutable del proceso. El encabezado predeterminado para este campo es COMMAND.

**args** Indica el nombre completo del comando que se está ejecutando. Se incluyen todos los argumentos de línea de comandos, aunque puede producirse truncamiento. El encabezado predeterminado para este campo es COMMAND.

**project** Nombre del proyecto asignado al proceso. En el entorno operativo actual, los campos PROJECT y USER no se traducen a nombres para los procesos que se ejecutan dentro de una partición de trabajo.

## 2.11. Ejecute dos comandos en background (los que quiera).

```
1 nano&
2 vi&
```

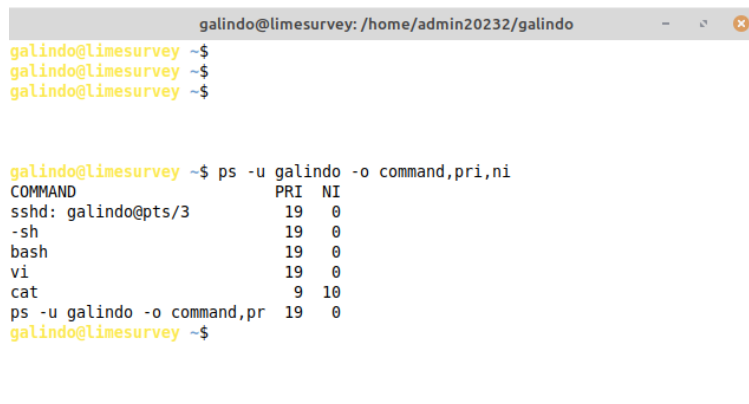


```
galindo@limesurvey: /home/admin20232/galindo
$ vi&
$ nano&
[2] + Stopped (tty output)      vi
$ jobw
-sh: 39: jobw: not found
[3] + Stopped (tty output)      nano
$ jobs
[3] + Stopped (tty output)      nano
[2] - Stopped (tty output)      vi
[1]  Stopped (tty output)      vi
$
```

Figura 9: Nano y Vi corriendo en el fondo

## 2.12. Ejecute el comando cat >lista, ¿Qué prioridad tiene asignada?

- Tiene la prioridad 19



```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$
galindo@limesurvey ~$
galindo@limesurvey ~$

galindo@limesurvey ~$ ps -u galindo -o command,pri,ni
COMMAND          PRI  NI
sshd: galindo@pts/3  19   0
-sh               19   0
bash              19   0
vi                19   0
cat               9   10
ps -u galindo -o command,pr 19   0
galindo@limesurvey ~$
```

Figura 10: use ps desde otra terminal para ve la prioridad

### 2.13. Matar el proceso anterior.

```
1 kill -9 130985
```



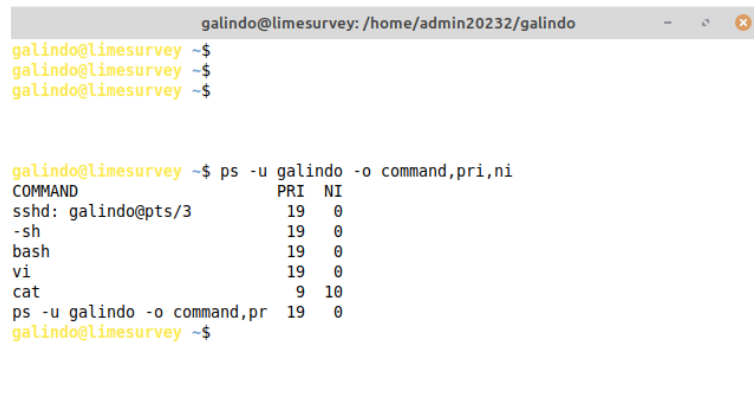
A terminal window titled 'galindo@limesurvey: /home/admin20232/galindo' showing the following commands and output:

```
galindo@limesurvey ~$ ps -u galindo -o pid,command
PID COMMAND
129271 sshd: galindo@pts/3
129272 -sh
129277 bash
129286 vi
130985 cat
131748 ps -u galindo -o pid,command
galindo@limesurvey ~$ kill -9 130985
galindo@limesurvey ~$ ps -u galindo -o pid,command
PID COMMAND
129271 sshd: galindo@pts/3
129272 -sh
129277 bash
129286 vi
131751 ps -u galindo -o pid,command
[2]+ Killed cat > lista
galindo@limesurvey ~$
```

Figura 11: Matar el proceso

### 2.14. Vuelva a ejecutar cat>lista pero con menor prioridad.

```
1 nice -n 10 cat>lista
```



A terminal window titled 'galindo@limesurvey: /home/admin20232/galindo' showing the following commands and output:

```
galindo@limesurvey ~$
galindo@limesurvey ~$
galindo@limesurvey ~$

galindo@limesurvey ~$ ps -u galindo -o command,pri,ni
COMMAND      PRI  NI
sshd: galindo@pts/3  19   0
-sh           19   0
bash          19   0
vi            19   0
cat           9   10
ps -u galindo -o command,pr  19   0
galindo@limesurvey ~$
```

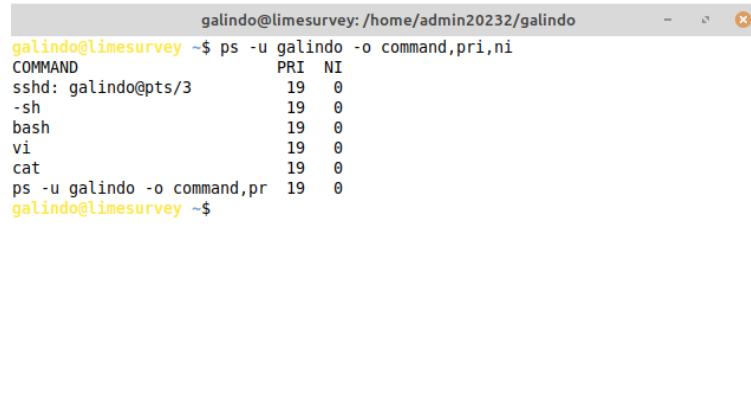
Figura 12: Nice ahora es mayor y la prioridad es mas baja

### ¿Qué prioridad le fue asignada?

- Prioridad marca 9

## 2.15. Una vez más ejecute cat>lista, pero ahora en el background .

```
1 cat > lista &
```



```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps -u galindo -o command,pri,ni
COMMAND      PRI  NI
sshd: galindo@pts/3  19   0
-sh           19   0
bash          19   0
vi            19   0
cat           19   0
ps -u galindo -o command,pr  19   0
galindo@limesurvey ~$
```

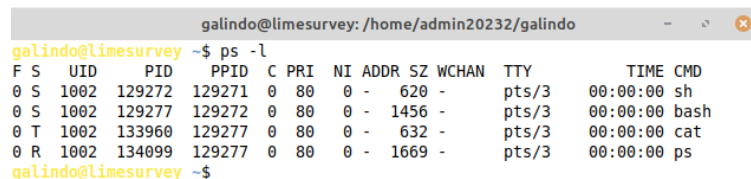
Figura 13: salida de ps -u

## 2.16. ¿Cuál es su prioridad ahora?

- El valor de prioridad es '19' y el valor de nice es '0'

## 2.17. Verifique que el comando en background este en la lista de procesos.

```
1 ps -l
```

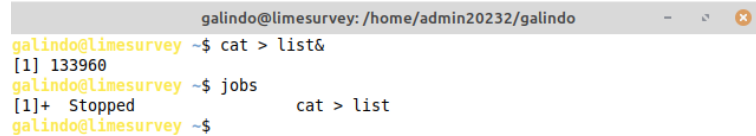


```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ ps -l
F S  UID      PID      PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY      TIME  CMD
0 S  1002    129272  129271  0   80   0  -    620  -      pts/3    00:00:00 sh
0 S  1002    129277  129272  0   80   0  -   1456  -      pts/3    00:00:00 bash
0 T  1002    133960  129277  0   80   0  -    632  -      pts/3    00:00:00 cat
0 R  1002    134099  129277  0   80   0  -   1669  -      pts/3    00:00:00 ps
galindo@limesurvey ~$
```

Figura 14: lista de procesos

## 2.18. Verifique que el comando en background este en la lista de tareas (jobs).

```
1 jobs
```

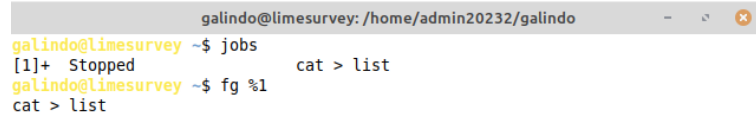
A terminal window titled 'galindo@limesurvey: /home/admin20232/galindo' showing the execution of 'cat > list&' followed by '[1] 133960'. Then, the 'jobs' command is entered, resulting in '[1]+ Stopped cat > list'.

```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ cat > list&
[1] 133960
galindo@limesurvey ~$ jobs
[1]+  Stopped cat > list
galindo@limesurvey ~$
```

Figura 15: Lista de tareas

## 2.19. Pase una de las tareas al foreground (use el número de tarea)

```
1 fg %1
```

A terminal window titled 'galindo@limesurvey: /home/admin20232/galindo' showing the 'jobs' command output '[1]+ Stopped cat > list'. Then, the 'fg %1' command is entered, bringing the process back to the foreground.

```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ jobs
[1]+  Stopped cat > list
galindo@limesurvey ~$ fg %1
cat > list
```

Figura 16: traer el proceso '1' al foreground

## 2.20. Pase la otra tarea al foreground, pero ahora use el número de PID.

- en este comando no se puede ejecutar en esta distribución

## 2.21. Envíe otro comando al background.

```
1 vi &
```

## 2.22. Finalice este proceso.

```
1 kill -9 134242
```



```
galindo@limesurvey: /home/admin20232/galindo
galindo@limesurvey ~$ jobs
[1]+  Stopped                  cat > list
galindo@limesurvey ~$ vi&
[2] 134242
galindo@limesurvey ~$ jobs
[1]-  Stopped                  cat > list
[2]+  Stopped                  vi
galindo@limesurvey ~$ jobs -l
[1]- 134239 Stopped (tty input) cat > list
[2]+ 134242 Stopped (tty output) vi
galindo@limesurvey ~$ kill -9 134242
galindo@limesurvey ~$ jobs
[1]-  Stopped                  cat > list
[2]+  Killed                   vi
galindo@limesurvey ~$
```

Figura 17: Proceso vi creado en el fondo y detenido

## 3. Conclusión

Durante esta practica aprendi como crear procesos, como leer procesos y como usar comandos para mandar al fondo y trer de vuelta, siento que estos comandos me seran muy utiles para poder trabajar y no tener que estar haciendo scripts para todo.

## 4. Fuentes

ps Command. (s.f.). [https://www.ibm.com/docs/en/aix/7.3?topic=p-ps-command#ps\\_\\_row-d3e111723](https://www.ibm.com/docs/en/aix/7.3?topic=p-ps-command#ps__row-d3e111723)