

# Universidad Autónoma de Baja California

## Facultad de Ciencias Químicas e Ingeniería

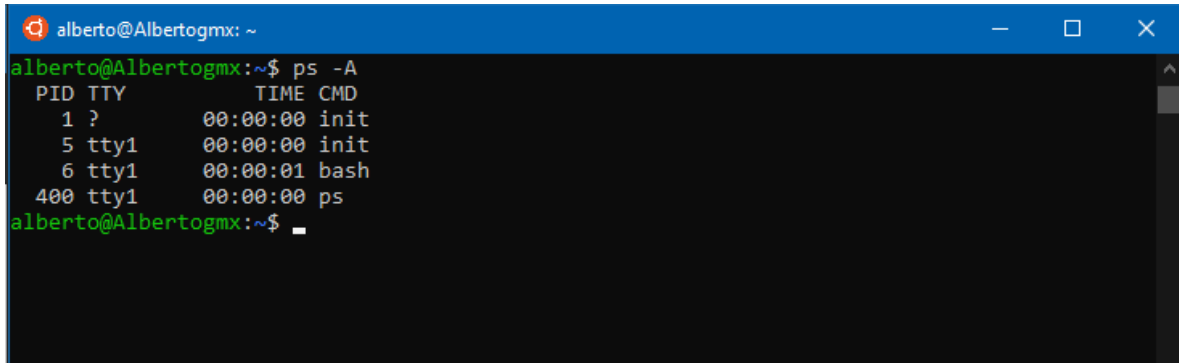


## **Taller de Sistema Operativo UNIX**

### **PROCESOS**

Docente: Dra. Pérez Ornelas Felicitas  
Alumno: Gómez Cárdenas Emmanuel Alberto  
Matrícula: 1261509

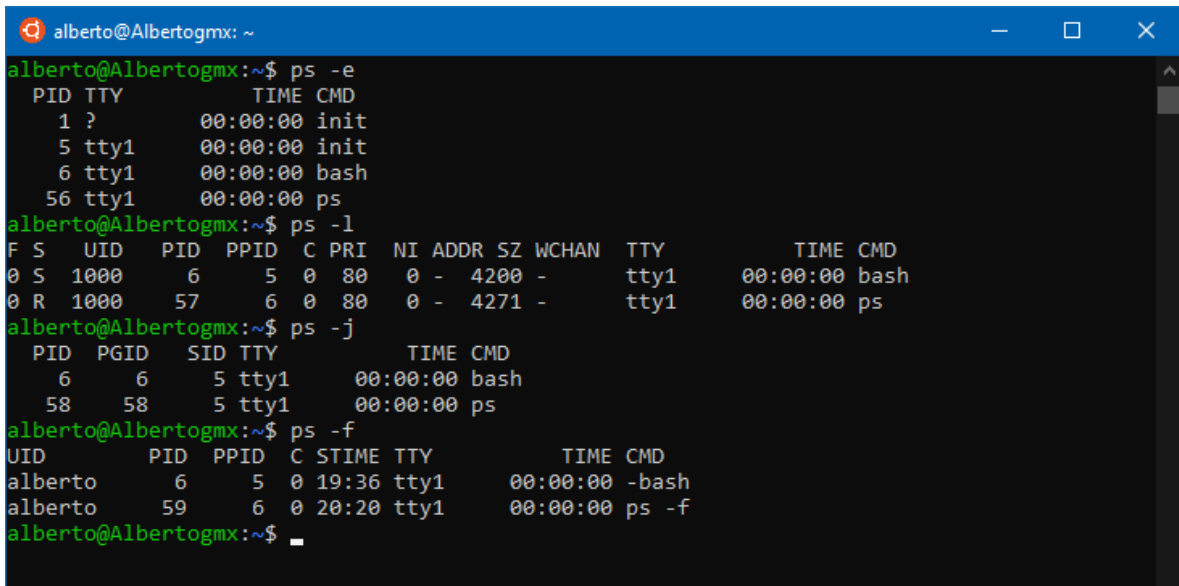
## 1.- Genere un listado completo de todos los procesos que están en el sistema.



```
alberto@Albertogmx: ~  
alberto@Albertogmx:~$ ps -A  
  PID TTY          TIME CMD  
    1 ?            00:00:00 init  
    5 tty1        00:00:00 init  
    6 tty1        00:00:01 bash  
   400 tty1        00:00:00 ps  
alberto@Albertogmx:~$
```

## 2. Explique la diferencia entre las opciones de ps e, f, l y j.

La cantidad de datos de los procesos que se imprimen y lo que muestra cada dato.



```
alberto@Albertogmx: ~  
alberto@Albertogmx:~$ ps -e  
  PID TTY          TIME CMD  
    1 ?            00:00:00 init  
    5 tty1        00:00:00 init  
    6 tty1        00:00:00 bash  
   56 tty1        00:00:00 ps  
alberto@Albertogmx:~$ ps -l  
F S   UID     PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S   1000      6     5   0  80   0 -  4200 -          tty1        00:00:00 bash  
0 R   1000     57     6   0  80   0 -  4271 -          tty1        00:00:00 ps  
alberto@Albertogmx:~$ ps -j  
  PID PGID   SID TTY          TIME CMD  
    6   6     5 tty1        00:00:00 bash  
   58  58     5 tty1        00:00:00 ps  
alberto@Albertogmx:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
alberto    6    5   0  19:36 tty1        00:00:00 -bash  
alberto   59    6   0  20:20 tty1        00:00:00 ps -f  
alberto@Albertogmx:~$
```

## 3. Explique la diferencia entre las opciones de ps a y u.

“a” muestra información de otros usuarios, mientras que “u” solo muestra la información del usuario que ha solicitado la acción.

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

Procesos actuales.

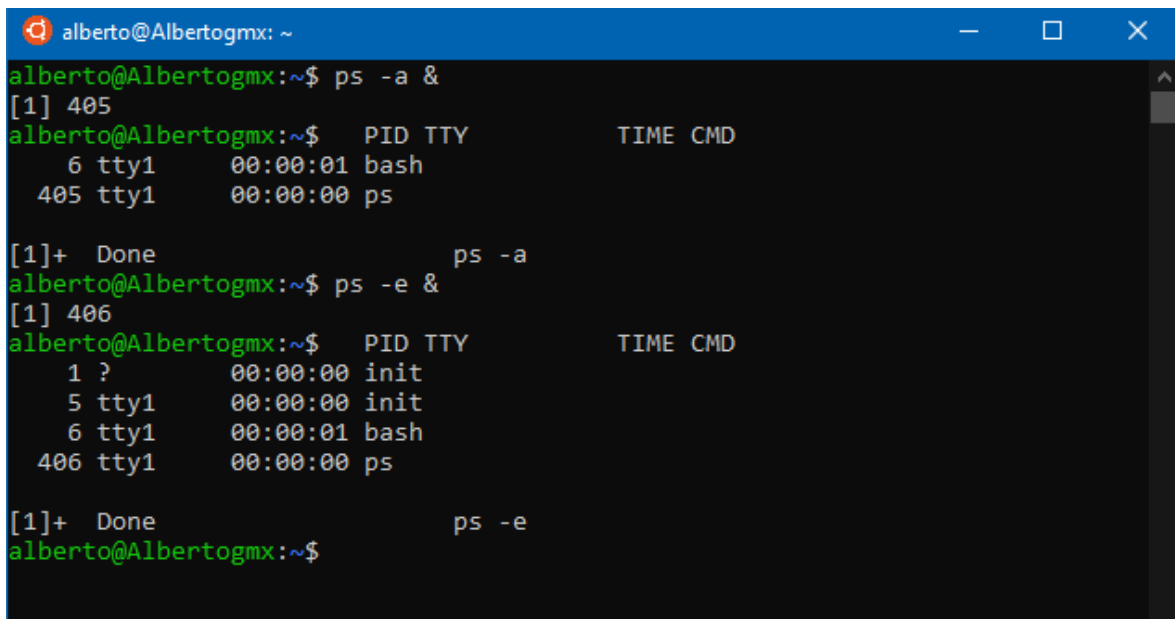
#### 5. ¿Qué opción de ps debería de usar para ver todos los procesos de un usuario?

El comando `ps -u *usuario*`, sustituyendo lo que está entre los asteriscos por el nombre del usuario.

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

El comando `ps` muestra un apartado llamado USER

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



```
alberto@Albertogmx: ~  
alberto@Albertogmx:~$ ps -a &  
[1] 405  
alberto@Albertogmx:~$ PID TTY          TIME CMD  
   6 tty1      00:00:01 bash  
  405 tty1      00:00:00 ps  
  
[1]+  Done                  ps -a  
alberto@Albertogmx:~$ ps -e &  
[1] 406  
alberto@Albertogmx:~$ PID TTY          TIME CMD  
    1 ?          00:00:00 init  
    5 tty1      00:00:00 init  
    6 tty1      00:00:01 bash  
  406 tty1      00:00:00 ps  
  
[1]+  Done                  ps -e  
alberto@Albertogmx:~$
```

**8. Ejecute el comando `cat >lista`, ¿Qué prioridad tiene asignada?**

```
alberto@Albertogmx: ~  
412 tty1 00:00:00 init  
413 tty1 00:00:00 bash  
460 tty1 00:00:00 ps  
ps -e &  
[2] 461  
[1] Done ps -A  
alberto@Albertogmx:~$ PID TTY TIME CMD  
1 ? 00:00:00 init  
412 tty1 00:00:00 init  
413 tty1 00:00:00 bash  
461 tty1 00:00:00 ps  
$cat>lista  
[2]+ Done ps -e  
alberto@Albertogmx:~$
```

**9. Mate el proceso anterior.**

```
alberto@Albertogmx:~$ kill 470  
[1] Terminated nice -1 cat > lista  
alberto@Albertogmx:~$ _
```

**10. Vuelva a ejecutar `cat>lista`, pero con menor prioridad.**

```
alberto@Albertogmx:~$ nice -1 cat>lista  
_
```

**11. ¿Qué prioridad le fue asignada?**

**12. Una vez más ejecute `cat>lista`, pero ahora en el background.**

```
alberto@Albertogmx:~$ cat>lista &  
[3] 475  
alberto@Albertogmx:~$
```

**13. ¿Cuál es su prioridad ahora?**

```
alberto@Albertogmx:~$ ps f
  PID TTY          STAT       TIME COMMAND
  413 tty1      S           0:00 -bash
  474 tty1      TN          0:00 \_ cat
  475 tty1      T           0:00 \_ cat
  481 tty1      TN          0:00 \_ cat
  482 tty1      TN          0:00 \_ cat
  486 tty1      R           0:00 \_ ps f
```

## Conclusiones

El comando ps es muy importante ya que, gracias a este, podemos hacer una lista de los procesos que están siendo ejecutados e identificar algunos procesos que estén ya sea, comportándose indebidamente o gastando más recursos de lo que deberían.

## Referencias

- francisconi.org. (2019). *ps*. [online] Available at: <https://francisconi.org/linux/comandos/ps> [Accessed 19 Sep. 2019].
- Hidalgo, U. (2019). *Monitorización de procesos en Linux*. [online] Uaeh.edu.mx. Available at: <https://www.uaeh.edu.mx/scige/boletin/huejutla/n11/p2.html> [Accessed 19 Sep. 2019].
- nixCraft. (2019). *Linux / UNIX: Kill Command Examples - nixCraft*. [online] Available at: <https://www.cyberciti.biz/faq/unix-kill-command-examples/> [Accessed 19 Sep. 2019].