

UNIVERSIDAD RAFAEL LANDÍVAR

FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS

SECCIÓN 1 VESPERTINA

ING. JULIO REQUENA

PRÁCTICA 1

Julio Anthony Engels Ruiz Coto 1284719

GUATEMALA DE LA ASUNCIÓN, FEBRERO 14 DE 2023

CAMPUS CENTRAL

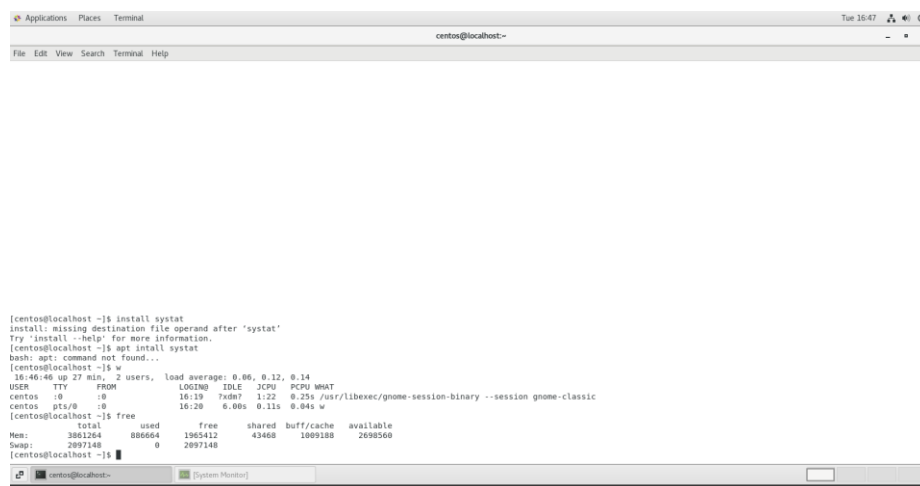
1) Que es el PID en GNU/LINUX

Es un identificador de proceso, cada proceso que ese ejecuta en un sistema operativo contiene un identificador único llamado PID [process id], este es el encargado para identificar de manera única un proceso en el sistema, este es un numero entero asignado por el SO a cada proceso en el momento en el que se crea, comúnmente se utilizan para monitorizar y controlar los procesos en ejecución en el sistema, se puede usar el PID para detener, continuar o matar un proceso.

2) Qué diferencia hay entre la opción -a y la opción -x de la orden ps

El comando ps -a muestra información sobre todos los procesos en ejecución del sistema incluyendo aquellos que están siendo ejecutados por otros usuarios, en cambio el comando ps -x muestra información sobre todos los procesos en el sistema incluyendo solo procesos que no tienen un terminal asociado en otras palabras muestra solo procesos en segundo plano.

3) Ejecutar la instrucción FREE



```
centos@localhost:~$ free
total        used        free      shared  buff/cache   available
Mem:  3861264  886664      1965412    43468    1009188    2698560
Swap:  2897148      0      2897148
```

The screenshot shows a terminal window with the 'free' command output. The output is a table with columns: total, used, free, shared, buff/cache, and available. The rows represent memory (Mem) and swap space. The 'Mem' row shows 3861264 total, 886664 used, 1965412 free, 43468 shared, 1009188 buff/cache, and 2698560 available. The 'Swap' row shows 2897148 total, 0 used, and 2897148 free.

4) Para que sirve el comando PS y cuáles son sus opciones

El comando PS lo que hace es mostrar información sobre los procesos en ejecución en el sistema, una vez escrito el comando en la terminal de CENTOS muestra en columnas información detallada sobre los procesos, como el PID, el usuario que ejecuta el proceso, el nombre del proceso, el tiempo de CPU utilizado, la memoria utilizada. El comando busca un proceso específico por su nombre y muestra información detallada solo sobre ese proceso. Las opciones que muestra son las siguientes en la presente tabla:

a :	Muestra todos los procesos asociados a un TTY.
e / -A :	Muestra todos los procesos.
x :	Muestra los no asociados.
-f :	Muestra el formato largo: UID : Usuario que lo ejecutó. PPID : Id del proceso padre. C : Uso del procesador. STIME : Inicio de ejecución.
u :	Orientado al usuario: USER % CPU : uso de procesador. % MEM : uso de memoria. VSZ : Memoria virtual. RSS : Memoria física. STAT : Estado. START : Iniciado.

5) Verificar en tiempo real como se están consumiendo recursos los procesos que se están ejecutando en el SO. (top)

```
[centos@localhost ~]$ top
top - 16:29:02 up 9 min, 2 users, load average: 0.34, 0.33, 0.31
tasks: 197 total, 1 running, 196 sleeping, 0 stopped, 0 zombie
%cpu(s): 12.9 us, 1.4 sy, 0.0 ni, 85.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 3861264 total, 2001756 free, 855284 used, 1004224 buff/cache
Mem Swap: 2097148 total, 2097148 free, 0 used, 2734080 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR %  CPU  MEM     TIME+ COMMAND
 4477 centos    20   0 3027136 209492 75928  9.3  5.4   0:15.98 gnome-shell
1714 root        20   0 3446336 60288 39944  5.3  1.8   0:04.02 X
 9247 centos    20   0 671756 27908 16744  5.1  0.7   0:01.25 gnome-terminal-
 598 root      20   0 295376 5256 4832  0.3  0.1   0:01.42 vmtolisd
5169 centos    20   0 614400 31452 18988  0.3  0.0   0:01.54 vmtolisd
5188 centos    20   0 530284 16848 10920  0.3  0.4   0:00.16 abrt-applet
  1 root       20   0 128284 6988 4196  0.0  0.2   0:02.36 systemd
  2 root       20   0 0 0 0  0.0  0.0   0:00.00 kthreadd
  4 root       20   0 0 0 0  0.0  0.0   0:00.00 kworker/0:0H
  6 root       20   0 0 0 0  0.0  0.0   0:00.62 ksoftirqd/0
  7 root       20   0 0 0 0  0.0  0.0   0:00.02 migration/0
  8 root       20   0 0 0 0  0.0  0.0   0:00.00 rcu_bh
  9 root       20   0 0 0 0  0.0  0.0   0:00.82 rcu_sched
10 root       0 -20  0 0 0  0.0  0.0   0:00.00 lru-add-drain
11 root       rt   0 0 0  0.0  0.0   0:00.01 watchdog/0
13 root       20   0 0 0 0  0.0  0.0   0:00.00 kdevtmpfs
14 root       0 -20  0 0 0  0.0  0.0   0:00.00 netns
15 root       20   0 0 0 0  0.0  0.0   0:00.00 khungtaskd
16 root       0 -20  0 0 0  0.0  0.0   0:00.00 writeback
17 root       0 -20  0 0 0  0.0  0.0   0:00.00 kintegrityd
18 root       0 -20  0 0 0  0.0  0.0   0:00.00 bioset
19 root       0 -20  0 0 0  0.0  0.0   0:00.00 bioset
20 root       0 -20  0 0 0  0.0  0.0   0:00.00 bioset
21 root       0 -20  0 0 0  0.0  0.0   0:00.00 kblockd
22 root       0 -20  0 0 0  0.0  0.0   0:00.00 md
23 root       0 -20  0 0 0  0.0  0.0   0:00.00 edac-poller
24 root       0 -20  0 0 0  0.0  0.0   0:00.00 watchdogd
25 root       20   0 0 0 0  0.0  0.0   0:00.21 kworker/0:1
30 root       20   0 0 0 0  0.0  0.0   0:00.00 kswapd0
```

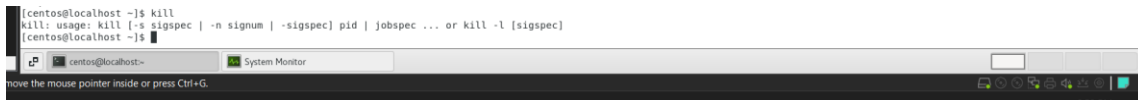
6) Averiguar la hora actual, cuánto tiempo lleva en marcha el sistema, el número de terminales abiertas y la carga media del sistema operativo. (uptime)

```
centos@localhost ~$ uptime
16:21:12 up 2 min, 2 users, load average: 1.04, 1.18, 0.46
centos@localhost ~$ ps
  PID TTY          TIME CMD
 9261 pts/0    00:00:00 bash
 9359 pts/0    00:00:00 ps
centos@localhost ~$ pstree
systemd--NodeManager--2*[{NodeManager}]
      NetworkManager--dnclient
      VGAuthService
      abrt-dbus--2*[{abrt-dbus}]
      abrt-watch-log
      abrt-d
      accounts-daemon--2*[{accounts-daemon}]
      alsaactl
      at-spi-bus-launcher--dbus-daemon--[{dbus-daemon}]
      at-spi2-registryd--3*[{at-spi-bus-launcher}]
      atd
      auditd--auditd--auditd
      avahi-daemon--avahi-daemon
      bolt--2*[{bolt}]
      chronyd
      colord--2*[{colord}]
      crond
      cupsd
      2*[{dbus-daemon}--[{dbus-daemon}]]
      dbus-launch
      dconf-service--2*[{dconf-service}]
      dnsmasq--dnsmasq
      evolution-address-book-plugin--5*[{evolution-address-book-plugin}]
      evolution-calendar-integration-plugin--8*[{evolution-calendar-integration-plugin}]
      evolution-source-plugin--1*[{evolution-source-plugin}]
```

7) Muestre los procesos del sistema en una estructura de árbol e indique que comandos utilizar (pstree)

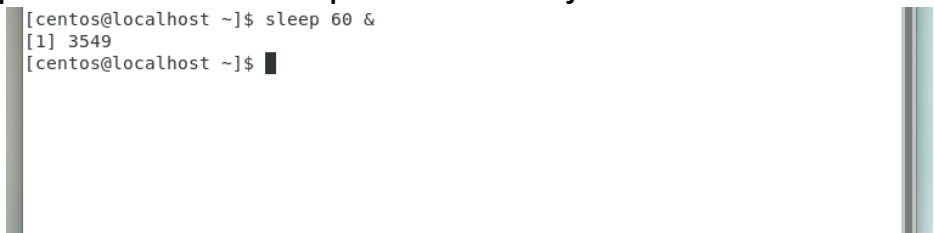


8) ¿Cuál es la instrucción que se utiliza para matar un proceso?
El comando KILL



9) Ejecutar el comando siguiente
sleep 60 &
crear un segundo proceso
sleep 30 &

comprobar que los procesos se estén ejecutando
colocar pantallazo de los 2 procesos en ejecución.



```
[centos@localhost ~]$ nice
0
[centos@localhost ~]$ sleep 60 &
[1] 3549
[centos@localhost ~]$ sleep 30 &
[2] 3647
[centos@localhost ~]$

File Edit View Search Terminal Help
[centos@localhost ~]$ nice
0
[centos@localhost ~]$ sleep 60 &
[1] 3549
[centos@localhost ~]$ sleep 30 &
[2] 3647
[centos@localhost ~]$ ps
  PID TTY          TIME CMD
  2806 pts/0    00:00:00 bash
  3832 pts/0    00:00:00 ps
[1]-  Done                  sleep 60
[2]+  Done                  sleep 30
[centos@localhost ~]$ █
```

10) Averiguar para que sirve la instrucción NICE y RENICE

La instrucción NICE se utiliza para ajustar la prioridad de un proceso en la cola de prioridades del sistema. Permite a un usuario ejecutar un proceso con una prioridad relativamente baja en comparación con el resto de los procesos en ejecución en el sistema. La instrucción RENICE permite ajustar la prioridad de un proceso en ejecución es similar a la instrucción NICE pero se utiliza más que todo para ajustar la prioridad de un proceso ya en ejecución en lugar de iniciar un nuevo proceso con prioridad específica.