

ESCUELA POLITÉCNICA NACIONAL  
ESCUELA DE FORMACIÓN DE TECNÓLOGOS



GUÍA PRÁCTICA DE LABORATORIO  
No. Práctica: 7

CARRERA: ASI

ASIGNATURA: Sistemas Operativos I CÓDIGO: TSI 213

FECHA: 02-01-2017

**1. PROPÓSITO DE LA PRÁCTICA: (exponga los Resultados de Aprendizaje esperados)**

Adquirir los conocimientos suficientes para desenvolverse con soltura en la línea de comandos del Manejo de procesos

**2. OBJETIVO GENERAL:**

Adquirir destreza en el manejo de procesos mediante la línea de comandos de verificación de espacio en disco.

**3. OBJETIVOS ESPECÍFICOS:**

Adquirir destreza en el manejo de procesos mediante la línea de comandos (top, htop, kill), Comandos de verificación de espacio en disco (df, du)

**4. INSTRUCCIONES:**

- 4.1 Lea y comprenda la introducción a la línea de comandos, información del sistema y manuales
- 4.2 Responda las preguntas y realice las actividades propuestas en el documento.



**ESCUELA POLITÉCNICA NACIONAL  
ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

PÁG -2-

**5. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA:**

Siga las instrucciones en el documento adjunto.

**6. BIBLIOGRAFÍA REFERENCIAL (estilo IEEE):**

[http://www.linuxtotal.com.mx/?cont=info\\_admon\\_002](http://www.linuxtotal.com.mx/?cont=info_admon_002)

<http://asofranruiz.blogspot.com/2014/01/comando-top-y-htop.html>

[http://www.linuxtotal.com.mx/?cont=info\\_admon\\_002](http://www.linuxtotal.com.mx/?cont=info_admon_002)

\_\_\_\_\_  
**FIRMA DEL DOCENTE:**

## ANEXO

En UNIX los procesos se identifican con un número entero denominado ID de proceso o PID. Al proceso que ejecuta la solicitud para la creación de un proceso se le suele llamar procesos padre, y al proceso creado se le suele llamar proceso hijo.

### Ps

El comando ps es el que permite informar sobre el estado de los procesos. ps esta basado en el sistema de archivos /proc, es decir, lee directamente la información de los archivos que se encuentran en este directorio. Tiene una gran cantidad de opciones, incluso estas opciones varían dependiendo del estilo en que se use el comando. Estas variaciones sobre el uso de ps son las siguientes:

- Estilo UNIX, donde las opciones van precedidas por un guión -
- Estilo BSD, donde las opciones no llevan guión
- Estilo GNU, donde se utilizan nombres de opciones largas y van precedidas por doble guión --

Sea cual sea el estilo utilizado, dependiendo de las opciones indicadas, varias columnas se mostrarán en el listado de procesos que resulte, estas columnas pueden ser entre muchas otras, las siguientes (y principales):

p o PID	Process ID, número único o de identificación del proceso.
P o PPID	Parent Process ID, padre del proceso
U o UID	User ID, usuario propietario del proceso
t o TT o TTY	Terminal asociada al proceso, si no hay terminal aparece entonces un '?'
T o TIME	Tiempo de uso de cpu acumulado por el proceso
c o CMD	El nombre del programa o camndo que inició el proceso
RSS	Resident Sise, tamaño de la parte residente en memoria en kilobytes
SZ o SIZE	Tamaño virtual de la imagen del proceso
NI	Nice, valor nice (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19)
C o PCPU	Porcentaje de cpu utilizado por el proceso
STIME	Starting Time, hora de inicio del proceso
S o STAT	Status del proceso, estos pueden ser los siguientes <ul style="list-style-type: none"><li>• R runnable, en ejecución, corriendo o ejecutándose</li><li>• S sleeping, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar</li></ul>

- T sTopped, proceso detenido totalmente, pero puede ser reiniciado
- Z zombie, difunto, proceso que por alguna razón no terminó de manera correcta, no debe haber procesos zombies
- D uninterruptible sleep, son procesos generalmente asociados a acciones de IO del sistema
- X dead, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca

Las opciones completas de ps las encuentras en las páginas del manual (man ps), o escribiendo en la terminal ps L, y para ver un resumen de sus opciones más comunes usa ps --help:

```
#> ps --help
***** simple selection ***** ***** selection by list *****
-A all processes          -C by command name
-N negate selection       -G by real group ID (supports names)
-a all w/ tty except session leaders -U by real user ID (supports names)
-d all except session leaders -g by session OR by effective group name
-e all processes          -p by process ID
T all processes on this terminal -s processes in the sessions given
a all w/ tty, including other users -t by tty
g OBSOLETE -- DO NOT USE      -u by effective user ID (supports names)
r only running processes    U processes for specified users
x processes w/o controlling ttys t by tty
***** output format ***** ***** long options *****
-o,o user-defined -f full      --Group --User --pid --cols --ppid
-j,j job control s signal     --group --user --sid --rows --info
-O,O preloaded -o v virtual memory --cumulative --format --deselect
-l,l long          u user-oriented --sort --tty --forest --version
-F extra full X registers --heading --no-heading --context
***** misc options *****
-V,V show version  L list format codes f ASCII art forest
-m,m,-L,-T,H threads S children in sum -y change -l format
-M,Z security data c true command name -c scheduling class
-w,w wide output  n numeric WCHAN,UID -H process hierarchy
```

A continuación algunos cuantos ejemplos de ps con la salida recortada.

```
># ps -e (-e muestra todos los procesos)
PID TTY      TIME CMD
 1 ?        00:00:01 init
 2 ?        00:00:00 kthreadd
 3 ?        00:00:00 migration/0
 4 ?        00:00:00 ksoftirqd/0

#> ps -ef (-f muestra opciones completas)
UID      PID PPID C STIME TTY      TIME CMD
root      1  0  0 10:12 ?        00:00:01 init [5]
root      2  0  0 10:12 ?        00:00:00 [kthreadd]
...
root    6130 5662 0 10:24 pts/0    00:00:00 su -
root    6134 6130 0 10:24 pts/0    00:00:00 -bash
```

```

sergon 6343 5604 0 10:28 ? 00:00:00 kio_file [kdeinit] file /home/sergon/tmp/ksocket-sergon/kl
root 6475 6134 0 10:38 pts/0 00:00:00 ps -ef

#> ps -eF (-F muestra opciones completas extra)
UID    PID PPID C  SZ  RSS PSR STIME TTY    TIME CMD
root    1   0  0 412 556 1 16:59 ?    00:00:01 init [5]
root    2   0  0 0   0 1 16:59 ?    00:00:00 [kthreadd]
sergon 8326 8321 0 902 1272 0 17:07 ?    00:00:00 /bin/sh /usr/lib/firefox-2.0.0.8/run-mozilla.sh /usr/lib/f
sergon 8331 8326 4 53856 62604 0 17:07 ?    00:00:50 /usr/lib/firefox-2.0.0.8/mozilla-firefox-bin
sergon 8570 7726 2 15211 37948 0 17:17 ?    00:00:10 quanta

#> ps ax (formato BSD sin gui3n, a muestra todos, x sin mostrar tty)
PID TTY  STAT TIME COMMAND
1 ?    Ss   0:01 init [5]
2 ?    S<   0:00 [kthreadd]
3 ?    S<   0:00 [migration/0]
4 ?    S<   0:00 [ksoftirqd/0]

#> ps aux (formato BSD sin gui3n, u muestra usuarios y dem4s columnas)
USER    PID %CPU %MEM  VSZ  RSS TTY  STAT START  TIME COMMAND
root    1 0.0 0.0 1648 556 ?    Ss  16:59  0:01 init [5]
root    2 0.0 0.0 0   0 ?    S<  16:59  0:00 [kthreadd]
root    3 0.0 0.0 0   0 ?    S<  16:59  0:00 [migration/0]
root    4 0.0 0.0 0   0 ?    S<  16:59  0:00 [ksoftirqd/0]
root    5 0.0 0.0 0   0 ?    S<  16:59  0:00 [migration/1]

#> ps -eo user,pid,tty (-o output personalizado, se indican los campos separados por coma, ver ps --help o ps L)
USER    PID TT
root    1 ?
root    2 ?
sergon 8570 tty 1
root 8876 pts/1

#> ps -eH (muestra 4rbol de procesos)
#> ps axf (lo mismo en formato BSD)
#> ps -ec (el comando que se esta ejecutando, sin la ruta, solo el nombre real)
#> ps -el (muestra formato largo de varias columnas, muy pr4ctico)
#> ps L (No muestra procesos, lista todos los c3digos de formatos)

```

## kill

El comando kill, que literalmente quiere decir matar, sirve no solo para matar o terminar procesos sino principalmente para enviar se4ales (signals) a los procesos. La se4al por default (cuando no se indica ninguna es terminar o matar el proceso), y la sintaxis es kill PID, siendo PID el n4mero de ID del proceso. As4 por ejemplo, es posible enviar una se4al de STOP al proceso y se detendr4 su ejecuci3n, despu4s cuando se quiera mandar una se4al de Continuar y el proceso continuara desde donde se qued3.

```

#> kill -l (lista todas las posibles se4ales que pueden enviarse a un proceso)
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL

```

```

5) SIGTRAP  6) SIGABRT  7) SIGBUS  8) SIGFPE
9) SIGKILL 10) SIGUSR1 11) SIGSEGV 12) SIGUSR2
13) SIGPIPE 14) SIGALRM 15) SIGTERM 16) SIGSTKFLT
17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU
25) SIGXFSZ 26) SIGVTALRM 27) SIGPROF 28) SIGWINCH
29) SIGIO 30) SIGPWR 31) SIGSYS 34) SIGRTMIN
35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4
39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7 58) SIGRTMAX-6
59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX

```

La lista previa presenta una lista de todas las posibles señales que pueden mandarse a un proceso y estas pueden ser invocadas a través del número de la señal o de su código, por ejemplo:

```

#> kill -9 11428      (termina, mata un proceso completamente)
#> kill -SIGKILL 11428 (Lo mismo que lo anterior)

```

Las señales más comunes son la 19 y 20 que detienen momentáneamente la ejecución de un proceso o programa, 18 la continua, 1 que es la señal de hang up que obliga al proceso a releer sus archivos de configuración estando en ejecución y 9 que termina rotundamente un proceso.

## top

Una utilidad muy usada y muy útil para el monitoreo en tiempo real del estado de los procesos y de otras variantes del sistema es el programa llamado top, se ejecuta desde la línea de comandos, es interactivo y por defecto se actualiza cada 3 segundos.

```

$> top
top - 13:07:30 up 8 days, 6:44, 4 users, load average: 0.11, 0.08, 0.08
Tasks: 133 total, 1 running, 131 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.0%us, 0.2%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 497356k total, 472352k used, 25004k free, 21500k buffers
Swap: 1156640k total, 257088k used, 899552k free, 60420k cached

  PID USER   PR NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
26156 sergon 15  0 2160 1016 784 R   1  0.2   0:00.93 top
   1 root    15  0 2012  616  584 S   0  0.1   0:00.98 init
   2 root    RT  0   0    0   0 S   0  0.0   0:00.29 migration/0
   3 root    34 19   0    0   0 S   0  0.0   0:00.00 ksoftirqd/0
   4 root    RT  0   0    0   0 S   0  0.0   0:00.00 watchdog/0
   5 root    RT  0   0    0   0 S   0  0.0   0:00.38 migration/1
...

```

Estando adentro de la aplicación, presionando 'h' muestra una ayuda de los posibles comandos que permiten configurar top, por ejemplo, al presionar 's' pregunta por el tiempo en segundos de actualización, etc.

Estas son algunas de las herramientas, las más importantes y usadas, para administrar procesos, hay varios programas en ambientes gráficos que en una sola pantalla permiten todo lo anterior y más.

## HTOP

Una de las mejoras que tiene HTOP respecto a TOP es que permite la navegación entre procesos, tanto vertical como horizontalmente mediante las flechas en el teclado. Esto da mucha comodidad al usuario.

Con HTOP es posible visualizar 2 o 4 medidores de CPU (una limitación importante que tenía esta herramienta), de modo de poder diferenciar bien claramente el uso que cada uno de los núcleos está recibiendo.

Otra mejora interesante es que para matar un proceso no necesitas introducir el número de éste, tan sólo situarte sobre él y matarlo, lo que da una comodidad mucho mayor con respecto a TOP.

Además HTOP tiene soporte para colores y para ratón algo facilita mucho las cosas al usuario.

Por último cabe destacar que HTOP carga a una velocidad mayor, además de todas las mejoras mencionadas anteriormente.

Esta herramienta no viene instalada en el sistema por defecto y por tanto hay que instalarla con el comando `apt-get install htop`:

```
root@debian:/home/usuario# apt-get install htop
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  strace ltrace
Se instalarán los siguientes paquetes NUEVOS:
  htop
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 71,6 kB de archivos.
Se utilizarán 200 kB de espacio de disco adicional después de esta operación.
Des:1 http://ftp.fr.debian.org/debian/ wheezy/main htop 1386 1.0.1-1 [71,6 kB]
Descargados 71,6 kB en 0seg. (154 kB/s)
Seleccionando el paquete htop previamente no seleccionado.
Leyendo la base de datos ... 50%
```

Ahora veremos cómo es el monitor de HTOP, mucho más intuitivo que el anterior:

The screenshot shows the htop interface on a Debian system. At the top, it displays system statistics: CPU usage at 25.3%, memory usage at 133/502MB, and swap usage at 4/382MB. It also shows system tasks (88), threads (133), and uptime (02:37:56). Below this is a table of running processes with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The processes listed include htop, gnome-terminal, gnome-panel, init, udevd, and NetworkManager. At the bottom, there are function key shortcuts like F1 for help, F2 for setup, and F10 for quit.

También hay otros tipos de funcionamientos que no nos lo muestra a simple vista HTOP, pero que con el comando man podemos ver. Algunos son los siguientes:

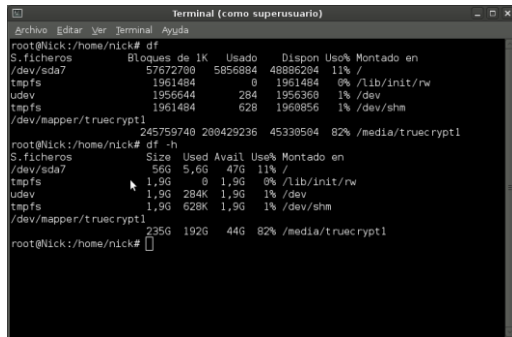
- 'T': ordena por el tiempo que llevan funcionando.
- 'u': muestra los usuarios para seleccionar el que deseemos y ver sus procesos.
- 'M': Ordena por el rendimiento de la memoria usada.
- 'P': Ordena por el rendimiento del procesador.
- 'C': Muestra un menú de opciones en el que podemos configurar el monitor como más nos interese.

## Df

Este es un pequeño comando para los administradores de sistemas que nos es de mucha utilidad para poder ver el espacio disponible y el espacio utilizado de nuestros discos rígidos de nuestro equipo con linux. Se trata del comando df abrimos la consola y ejecutamos el comando para ver una breve descripción de la utilización de los discos, si le agregamos el parámetro -h nos dará un detalle en gigas para que sea más fácil de leer.

# df -h





```
Terminal (como superusuario)
root@Nick:/home/nick# df
S.ficheros Bloques de 1K Usado Dispon Use% Montado en
/dev/sda7 57672700 5856804 48886204 11% /
tmpfs 1961484 0 1961484 0% /lib/init/rw
udev 1956644 284 1956360 1% /dev
tmpfs 1961484 628 1960856 1% /dev/shm
/dev/mapper/truecrypt1 245759740 200429236 45330504 82% /media/truecrypt1
root@Nick:/home/nick# df -h
S.ficheros Size Used Avail Use% Montado en
/dev/sda7 56G 5,6G 47G 11% /
tmpfs 1,9G 0 1,9G 0% /lib/init/rw
udev 1,9G 284K 1,9G 1% /dev
tmpfs 1,9G 628K 1,9G 1% /dev/shm
/dev/mapper/truecrypt1 235G 192G 44G 82% /media/truecrypt1
root@Nick:/home/nick#
```

También podemos saber qué tamaño tienen nuestras carpetas haciendo uso del comando **du**



```
mc [redacted] :/ (como superusuario)
Archivo Editar Ver Terminal Ayuda

@Nic :/# du -s /*
5784    /bin
15196   /boot
288     /dev
10760   /etc
1501824 /home
0       /initrd.img
4       /iron64.deb
135364  /lib
4648    /lib32
0       /lib64
16      /lost+found
339591270 /media
4       /mnt
```

## EJERCICIOS

1. Listar los procesos existentes en el root
2. Eliminar un proceso
3. Monitorear en estado real un proceso escogido
4. Verificar el tamaño de las carpetas del directorio curso
5. Verificar el espacio disponible de los discos.

## REFERENCIAS

- [http://www.linuxtotal.com.mx/?cont=info\\_admon\\_002](http://www.linuxtotal.com.mx/?cont=info_admon_002)
- <http://asofranruiz.blogspot.com/2014/01/comando-top-y-htop.html>
- [http://www.linuxtotal.com.mx/?cont=info\\_admon\\_002](http://www.linuxtotal.com.mx/?cont=info_admon_002)