UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Curso Operador Linux

Módulo 6

CONTROL DE PROCESOS

Presentación

En esta unidad se introducirá a los participantes en el manejo y control de los procesos.

Objetivos

Los participantes al finalizar la Unidad:

- Entenderán como maneja los procesos el sistema
- Podrán obtener información de los procesos
- Sabrán como comunicarse con los procesos y controlarlos

Temario

6.1 Comprensión de los procesos y la multitarea
6.2 Procesamiento en segundo plano
6.3 Como iniciar procesos múltiples
6.4 Información y supervisión de procesos
6.5 Como controlar procesos múltiples

Actividad de aplicación del conocimiento

Los participantes encontraran la actividad de aplicación en un archivo por separado en caso de ser requerida.

Examen

Los participantes deberán rendir el examen online o presentar el material solicitado según corresponda.

6.1 Comprensión de los procesos y la multitarea

Procesos es un nombre que se le da a los programas que se están ejecutando, por la tanto un programa será un archivo cuando se encuentre en el disco rígido y será un proceso al encontrarse en memoria

Dado que el sistema es multiusuario y multitarea es muy probable que dos o mas usuarios utilicen el mismo programa simultáneamente. Por ejemplo un usuario que utiliza un programa de edición de textos. El usuario abre un archivo en el editor, los datos de ese archivo no forman parte del programa pero si del proceso en memoria. Si otro usuario ejecuta el mismo editor, al tiempo que el otro usuario lo esta utilizando, ambos estarán usando el mismo programa pero diferentes procesos. Los programas pueden ser únicos para todos los usuarios, pero los procesos pertenecen a cada uno de los usuarios.

El sistema lleva el control de los procesos en ejecución, lleva registro sobre en que terminal, a que usuario pertenece y demás datos en una tabla de procesos

Para ver la lista de procesos se utiliza el comando «ps» el cual imprime en pantalla los procesos que se están ejecutando.

 # ps

PID	TTY	TIME	CMD
230	tty1	00:00:01	login
256	tty2	00:00:01	mingetty
257	tty3	00:00:01	mingetty
258	tty4	00:00:01	mingetty
259	tty5	00:00:01	mingetty
260	tty1	00:00:04	bash
300	tty1	00:00:00	ps

Este comando permite listar todos los procesos del sistema, el identificador de proceso (PID), numero de terminal donde se esta ejecutando, estado, tamaño, nombre, propietario, tiempo de CPU, tiempo de reloj y demás. Sin argumentos solo mostrara el identificador de proceso (PID) de la terminal sobre la que se está ejecutando.

A un proceso en ejecución (corriendo) se lo denomina tarea

El mencionado comando acepta, entre otras, las siguientes opciones:

-a Muestra los procesos con consola controlada, menos los del usuario actual

-u Muestra los propietarios de los procesos

-x Muestra los procesos que no tienen una terminal controlada

-r Muestra solo los procesos en ejecución

-l Produce un listado en formato largo

-ww Muestra todos los parámetros de la línea de comandos de los procesos

 # ps aux

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	2100	692	?	Ss	Nov06	0:04	init [2]
root	2	0.0	0.0	0	0	?	S<	Nov06	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S<	Nov06	0:00	[migration/0]
root	4	0.0	0.0	0	0	?	S<	Nov06	0:04	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	Nov06	0:00	[watchdog/0]
root	6	0.0	0.0	0	0	?	S<	Nov06	0:00	[migration/1]
root	7	0.0	0.0	0	0	?	S<	Nov06	0:03	[ksoftirqd/1]
root	8	0.0	0.0	0	0	?	S<	Nov06	0:00	[watchdog/1]
root	9	0.0	0.0	0	0	?	S<	Nov06	0:10	[events/0]
root	10	0.0	0.0	0	0	?	S<	Nov06	0:18	[events/1]
root	3048	0.0	0.1	3772	1128	pts/0	Ss+	Nov06	0:00	/bin/su root
root	3057	0.0	1.5	38704	16196	pts/0	Sl+	Nov06	0:06	gnome-term

Significado de cada una de las columnas

Columna	Estado	Significado
USER		Indica el dueño de cada proceso
PID		Numero de identificación del proceso
%CPU		Porcentaje de CPU ocupado por el proceso
%MEM		Porcentaje de memoria ocupada por el proceso
VSZ		Memoria virtual ocupada por el proceso
RSS		Memoria residente ocupada por el proceso
TTY		La consola controlada por el proceso. Un signo ? significa que no está conectada a ninguna terminal
STAT		Estado del proceso
	D	Durmiente ininterrumpible
	R	Proceso en ejecución o encolado para correr
	S	Proceso dormido
	T	Proceso en estado de traceo por debugger o detenido
	Z	Proceso Zombie
	W	Sin paginas residentes (BSD)
	<	Proceso con alta prioridad (BSD)
	N	Proceso de baja prioridad (BSD)
	L	Con paginas bloqueadas en memoria (real time) (BSD)
	s	Proceso en la consola actual
	+	Proceso en foreground

Para ver una lista de procesos que se actualice debemos recurrir al comando «top»

 # top

```
top - 03:30:06 up 1 day, 10:48,  4 users,  load average: 0.05, 0.03, 0.00
Tasks: 109 total,   2 running, 107 sleeping,   0 stopped,   0 zombie
Cpu(s):  2.5%us,  2.9%sy,  0.0%ni, 94.6%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1036040k total,   473096k used,   562944k free,   170152k buffers
Swap:  2650684k total,        0k used,  2650684k free,   167012k cached

  PID  USER PR  NI  VIRT  RES    SHR S %CPU %MEM    TIME+  COMMAND
 2707  root  20   0 42808  14m   6648 R    5  1.4   2:27.57 Xorg
 3057  root  20   0 38704  15m     9m S    3  1.6   0:05.62 gnome-terminal
 2891  user  20   0 71512  19m    13m S    2  2.0   0:12.02 nautilus
20349  root  20   0  2392  1136   880 R    1  0.1   0:00.08 top
 2909  user  20   0 25484  14m   9408 S    0  1.4   2:20.60 nm-applet
    1  root  20   0  2100   692    592 S    0  0.1   0:04.62 init
    2  root  15  -5     0     0      0 S    0  0.0   0:00.00 kthreadd
```

6.2 Procesamiento en segundo plano

Un proceso puede estar en primer plano, como en segundo plano. Sólo puede haber un proceso en primer plano al mismo tiempo, este proceso es el que va a interactuar con el usuario, recibiendo señales de su entrada (el teclado o Mouse) y enviando señales a la salida (la pantalla salvo que se haya redirigido la salida estándar).

El proceso en segundo plano no recibe ninguna señal desde el teclado y se ejecuta en background sin necesidad de interacción.

Los procesos pueden ser suspendidos o parados. Un proceso parado es aquel que no se esta ejecutando actualmente, sino que esta temporalmente detenido. Después de suspender una tarea, puede indicarle que se continúe en primer plano o en segundo plano, según se necesite. Retomar una tarea suspendida no altera en nada el estado de la misma y continúa ejecutándose justo donde se la dejo.

Enviando a segundo plano

Para enviar una tarea a segundo plano deberá utilizar el símbolo “&”. Por ejemplo supongamos que quisiera copiar una imagen de un CD (netinst.iso) a la carpeta /usr/debian/, para realizar esto debería ejecutar:

```
# cp /mnt/cdrom/netinst.iso /usr/debian
```

Este proceso duraría el tiempo en que demore en copiar el CD, una vez que finalice el proceso devolverá el control al bash, permitiendo ejecutar nuevos comandos. Para no tener que esperar al que proceso termine lo podría enviar a segundo plano ejecutándolo de la siguiente manera:

```
# cp /mnt/cdrom/netinst.iso /usr/ ebian&
[1]  345
#
```

Como podrá observar el control es regresado al shell, es decir tenemos nuevamente prompt y el sistema queda listo para ejecutar otras tareas. El [1] representa el número de tarea del proceso cp. El shell le asigna un número a cada tarea que se esta ejecutando en segundo plano o background, como este es el único comando que se esta ejecutando se le asigno el número uno. El 345 es el número de identificación del proceso o PID. Este número lo asigna el sistema a cada proceso que se ejecuta. Ambos números se utilizan para referirse a la tarea.

Para verificar el estado del proceso que ejecutamos en segundo plano existe el comando jobs el cual listará los procesos enviados a segundo plano:

```
jobs
[1] + Running  cp /mnt/cdrom/netinst.iso /usr/debian &
```

Parando y relanzando tareas

Otra manera de enviar un proceso a segundo plano es deteniendo el proceso que actualmente estamos ejecutando y ordenarle que vuelva a ejecutarse en segundo plano.

Por ejemplo ejecutemos el comando cp sin el & y luego presionemos las teclas <Ctrl+Z> para suspender el proceso:

```
# cp /mnt/cdrom/netinst.iso /usr/ ebian
^Z
[1]+   Stopped  cp /mnt/cdrom/netinst.iso /usr/debian
#
```

El proceso puede ser detenido en cualquier momento y mientras esta en este estado, o sea suspendido, no utiliza tiempo de CPU. Para relanzar la tarea en primer plano, ejecutamos el comando fg (foreground)

```
# fg
cp /mnt/cdrom/netinst.iso /usr/debian
```

El shell muestra el nombre del comando para que el usuario tenga conocimiento de que tarea puso en ejecución.

Si deseará poner una tarea suspendida en segundo plano, deberá utilizar el comando bg (background), al igual que fg el comando bg pone en ejecución la tarea pero en segundo plano:

```
# bg
[1]+   cp /mnt/cdrom/netinst.iso /usr/debian &
#
```

Tanto el comando fg como el bg actúan sobre las ultimas tareas, identificadas con un símbolo +.

Debemos entender que el control lo realiza el shell y los llamados fg, bg y jobs son internos del shell y no del sistema.

Dado que cada usuario tiene su espacio de control de tareas y su shell, cada numero de trabajo tiene significado para ese usuario en ese shell

6.3 Como iniciar procesos múltiples

Para indicarle que afecten a una tarea determinada, debe usar el número de identificación del shell, no se puede usar el PID para referirse a una tarea.

Por ejemplo iniciaremos dos tareas en background

```
# cp /var/log/* /mnt/monitor &
[1] 776

# cp /usr/bin/* /mnt/monitor &
[2] 777
```

Como se puede observar se lanzaron dos procesos en segundo plano, con el comando «jobs» verificamos que sea así:

```
# jobs
[1] Running    cp /var/log/* /mnt/monitor

[2] + Running   cp /usr/bin/* /mnt/monitor
```

Vemos que el proceso 2, al ser el último ejecutado tiene un signo +. Pero, para poner en primer plano el proceso 1, debe utilizar el siguiente comando:

```
# fg %1

cp /var/log/* /mnt/monitor
```

Para referirse al proceso se usa un signo % y el número de proceso

6.4 Información y supervisión de procesos

Para obtener información y/o supervisar los procesos mediante el shell utilizaremos los siguientes comandos del shell

jobs

Lista información de los trabajos que están en ejecución o suspendidos en ese momento. Suele también enumerar procesos que acaban de terminar.

fg % numerotrabajo

Devuelve el trabajo referido al primer plano. Si deseamos devolver al foreground el trabajo por defecto debemos ejecutar jobs y seleccionar el que esta indicado con el signo +

&

Este indicador como último parámetro en la línea de comandos le indica al shell que el comando se ejecute en segundo plano automáticamente.

bg % numerotrabajo

Manda un trabajo suspendido a ejecución en segundo plano. Si deseamos enviar al background el trabajo por defecto debemos ejecutar jobs y seleccionar el que esta indicado con el signo +

<Ctrl+C>

Combinación de teclas para lograr la interrupción de un programa o comando. Si presionamos Control-C mientras se ejecuta un programa en primer plano se matara la tarea.

<Ctrl+Z>

Combinación de teclas para lograr la suspensión de un programa o comando. Luego de suspendido el programa pude matarse o ser relanzado en primer o segundo plano.

kill PID
kill % numerotrabajo

Sirve para terminar, o enviar una señal como se vera a continuación, a un trabajo ya sea suspendido o en ejecución

6.5 Como controlar procesos múltiples

Enviando señales a los procesos

Cada proceso recibe y envía señales desde y hacia el sistema. Estas señales son las que permiten que el proceso funcione, salga del sistema si esta realizando alguna tarea incorrecta, etc. Pero hay veces que esto falla y el programa se cuelga o funciona mal. Para manejar esto existe un programa que envía señales al proceso para que haga ciertas cosas, por ejemplo que salga del sistema o deje de ejecutarse. Estos comandos son kill y killall. Los dos hacen exactamente lo mismo, pero el primero envía la señal al identificador del proceso, ya sea del sistema (PID) o del shell, cuando lo paramos o lo enviamos a segundo plano. Y el otro, las envía al nombre del proceso, la columna CMD del comando ps.

La sintaxis de los dos comandos es:

kill <señal> <PID 1> <PID 2> .... <PID n>
killall <señal> <CMD 1> <CMD 2> .... <CMD n>

Las señales pueden ser llamadas por su nombre o su número y las más utilizadas son las siguientes:

-1 o –HUP	Tira abajo un proceso, no lo mata, relee configuración
-2 o –INT	Interrumpe el proceso, es igual que apretar <Ctrl+C>
-3 o –QUIT	Sale del proceso
-9 o –KILL	Mata el proceso inmediatamente
-15 o –TERM	Obliga al proceso a terminar, es la opción por default
-19 o –STOP	Para o suspende el proceso, símil a apretar <Ctrl+Z>

La lista completa de señales se puede obtener por medio del comando con el argumento –l

```
# kill -l
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     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      32) SIGRTMIN    33)SIGRTMIN+1
34) SIGRTMIN+2  35) SIGRTMIN+3  36) SIGRTMIN+4  37)SIGRTMIN+5
38) SIGRTMIN+6  39) SIGRTMIN+7  40) SIGRTMIN+8  41)SIGRTMIN+9
42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45)SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMAX-15 49)SIGRTMAX-14
50) SIGRTMAX-13 51) SIGRTMAX-12 52) SIGRTMAX-11 53)SIGRTMAX-10
54) SIGRTMAX-9  55) SIGRTMAX-8  56) SIGRTMAX-7  57) SIGRTMAX-6
58) SIGRTMAX-5  59) SIGRTMAX-4  60) SIGRTMAX-3  61) SIGRTMAX-2
62) SIGRTMAX-1  63) SIGRTMAX
```

Ejemplo: para matar el proceso yes con identificador (PID) 345, deberá hacer lo siguiente:

```
# kill -9 345
```

ó

```
# killall -9 yes
```

O si es que se detuvo y su número de identificador es 3, deberá hacer lo siguiente:

```
# kill -9 %3
```