

LINUX

Procesos en Linux.

Implantación de Sistemas Operativos.





- 1. Procesos en linux**
- 2. Comandos para procesos**



1. Procesos en linux.

Un proceso es una instancia de un **programa en ejecución** (corriendo).

A los procesos frecuentemente se les refiere como **tareas**.

El **contexto de un programa** que esta en ejecución es lo que se llama un proceso. Este contexto puede ser mas procesos hijos que se hayan generado del principal (proceso padre), los recursos del sistema que este consumiendo, sus atributos de seguridad (tales como su propietario y permisos de archivos, así como roles y demás de Linux), etc.

Linux es un sistema operativo multitarea y multiusuario. Esto quiere decir que múltiples procesos pueden operar simultáneamente sin interferirse unos con los otros. Cada proceso tiene la "ilusión" que es el único proceso en el sistema y que tiene acceso exclusivo a todos los servicios del sistema operativo.

Programas y procesos son entidades distintas. En un sistema operativo multitarea, múltiples instancias de un programa pueden ejecutarse sumultáneamente. Cada instancia es un proceso separado.

Por ejemplo, si cinco usuarios desde equipos diferentes, ejecutan el mismo programa al mismo tiempo, habría cinco instancias del mismo programa, es decir, cinco procesos distintos.



1. Procesos en linux.

Cada proceso que se inicia es referenciado con un número de identificación único conocido como Process ID **PID**, que es siempre un entero positivo.

Prácticamente todo lo que se está ejecutando en el sistema en cualquier momento es un proceso, incluyendo el shell, el ambiente gráfico que puede tener múltiples procesos, etc.

La excepción a lo anterior es el kernel en si, el cual es un conjunto de rutinas que residen en memoria y a los cuales los procesos a través de llamadas al sistema pueden tener acceso.

Los **estados de un proceso** pueden ser:

- **R runnable**, en ejecución, corriendo o ejecutándose
- **S sleeping**, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar
- **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



2. Comandos...

ps

- El comando que da información sobre todos los procesos del sistema es el comando ps. Nos va a dar también información sobre todos los procesos de un usuario concreto.

ps [-aux]

- aux: muestra todos los procesos de los usuarios en ese momento
- au: muestra los procesos de los terminales de texto
- A modo general, la información que devuelve es:
 - Número de indentificación de proceso (PID). Éste comienza en 0 cuando el sistema arranca y va creciendo hasta un máximo, generalmente 32767.
 - Terminal desde el que se inició el proceso (TT)
 - Tiempo que el proceso se lleva ejecutando (TIME)
 - Línea de orden que se escribió para iniciar, el proceso (COMMAND). Es decir, nombre del proceso.

\$ps

PID	TTY	TIME	COMMAND
209	ttys0a	6:05	-sh
217	ttys0a	0:02	ps



2. Comandos...

ps -l

- Formato largo, devuelve más información que el anterior

<i>F</i>	<i>S</i>	<i>UID</i>	<i>PID</i>	<i>PPID</i>	<i>C</i>	<i>PRI</i>	<i>NI</i>	<i>ADDR</i>	<i>SZ</i>	<i>WCHAN</i>	<i>TTY</i>	<i>TIME</i>	<i>COMMAND</i>
100	S	1008	5230	1	5	30	20	10D924	37		TTYP1	0:01	-sh

- F: indicador hexadecimal, que especifica características del proceso, por ejemplo, 00 proceso terminado, 01 proceso del sistema, 08 proceso en memoria principal, 10 proceso bloqueado
- S: Estado del proceso , que será
- UID: Número de usuario que inició el proceso .
- PPID: Número de proceso superior.
- C: Cantidad de recursos de cpu utilizados por el proceso.
- PRI: Prioridad de ejecución del proceso.



2. Comandos...

Kill

Para terminar un proceso que está activamente ejecutándose se puede hacer presionando la tecla **break** o **supr**, pero no se pueden terminar de este modo procesos subordinados o asociados a terminales diferentes, para estos, se usa la orden **kill**. Proporcionándole como parámetro del Id de proceso.

La orden kill envía una señal al proceso, se pueden enviar distintas señales a los procesos, con kill, se envía la señal de “terminación de software”, si el proceso no termina con esta señal, se puede utilizar

kill -9 proceso

La opción **-9** envía la señal de “terminación incondicional”. Con esta opción se pueden terminar sesiones, útil por ejemplo cuando no se hace logout para salir y se arranca un nuevo terminal



2. Comandos...

killall

- Termina un proceso junto con los distintos subprocesos que tenga asociados. Permite eliminar el árbol de procesos.

Killall nombre-aplicación

killall gnome-terminal

La orden anterior mata el terminal abierto.



2. Comandos...

top

Es el equivalente al Monitor del sistema en la línea de comandos. Muestra una lista de procesos que se pueden ordenar de diferentes formas, por defecto muestra los procesos que más consumen.

-u usuario Muestra la lista de procesos del usuario indicado

Ejem: top -u alumno



2. Comandos...

top

Es el equivalente al Monitor del sistema en la línea de comandos. Muestra una lista de procesos que se pueden ordenar de diferentes formas, por defecto muestra los procesos que más consumen.

-u usuario Muestra la lista de procesos del usuario indicado

Ejem: top -u alumno



ACTIVIDADES

- **Realiza la actividad:**

Actividad 1:
Gestión de procesos.

Otras...