



Examen 103 - Comandos GNU y Unix

103.5 Crear, monitorizar y terminar procesos

Teoría

Un proceso es una instancia en ejecución de un programa. Cuando ejecutas un comando, el kernel de Linux crea un nuevo proceso para ejecutar ese programa. Cada proceso tiene un identificador único y está relacionado con el proceso que lo creó.

Conceptos Clave:

1. **Proceso:** Es una ejecución de un programa cargado en memoria, con su propio espacio de memoria, recursos (descriptores de archivo, etc.) e información de estado.
2. **PID (Process ID):** Cada proceso recibe un número único, el PID, al ser creado. El PID 1 suele estar reservado para el primer proceso del sistema (init o systemd), que es el "padre" de casi todos los demás procesos.
3. **PPID (Parent Process ID):** Cada proceso (excepto init/systemd) tiene un proceso padre, y el PPID es el PID de ese proceso padre.
4. **Árbol de Procesos:** Los procesos forman una jerarquía en árbol, comenzando desde el proceso init/systemd. Un proceso hijo hereda ciertas propiedades de su padre (como variables de entorno, directorio de trabajo).
5. **Procesos en Primer Plano (Foreground) y Segundo Plano (Background):**
 - **Primer Plano:** El proceso interactúa directamente con la terminal. La terminal espera a que el proceso termine para poder ejecutar otro comando.
 - **Segundo Plano:** El proceso se ejecuta de forma independiente de la terminal. La terminal queda libre para ejecutar otros comandos. Se lanza un proceso en segundo plano añadiendo un ampersand (&) al final del comando.
6. **Jobs:** La shell mantiene una lista de procesos que ha lanzado en segundo plano o que ha detenido. Estos se conocen como "jobs".
 - `jobs`: Muestra la lista de jobs activos.
 - `fg [%número_job]`: Trae un job (especificado por su número con %) al primer plano. Si se usa sin número, trae el último job que se envió al segundo plano o se detuvo.
 - `bg [%número_job]`: Envía un job detenido al segundo plano.
 - `Ctrl+Z`: Detiene un proceso en primer plano (lo pone en un estado "detenido"). Puedes usar `bg` o `fg` después.
 - `Ctrl+C`: Envía una señal de interrupción (SIGINT) al proceso en primer plano, solicitando su terminación "amable".

Monitorización de Procesos:

Estas herramientas te permiten ver qué procesos se están ejecutando, quién los ejecuta, cuántos recursos consumen, etc.

1. ps (process status):

- Muestra una instantánea de los procesos actuales.
- `ps`: Muestra los procesos asociados a la terminal actual.
- `ps aux`: Muestra todos los procesos de todos los usuarios en formato estilo BSD (muy común). Columnas clave: USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME, COMMAND.
- `ps ef`: Muestra todos los procesos en formato completo, mostrando el árbol de procesos y el comando completo con sus argumentos.
- `ps -eH`: Muestra un árbol de procesos.
- Puedes combinar opciones (`ps auxf`, `ps -efH`).

2. top:

- Muestra una vista dinámica y en tiempo real de los procesos. Actualiza periódicamente la información.
- Muestra información del sistema (carga promedio, uso de memoria/swap) y una lista de procesos ordenada por uso de CPU (por defecto).
- Dentro de `top`, puedes usar teclas para interactuar: `k` para "matar" un proceso (requiere PID), `r` para cambiar prioridad (`renice`), `M` para ordenar por memoria, `P` para ordenar por CPU, `q` para salir.

3. htop:

- Una alternativa mejorada y más amigable a `top`. Suele requerir instalación adicional (`sudo apt install htop` o `sudo dnf install htop`).
- Proporciona barras visuales de uso de CPU/memoria, permite desplazarse vertical y horizontalmente, y facilita la selección y gestión de procesos con el ratón o teclas de función (ej: `F9` para "matar").

Terminación de Procesos:

Para detener un proceso, se le envían "señales" (signals). Las señales son mensajes de software que notifican a un proceso de un evento.

1. kill:

- Envía una señal a uno o más procesos especificados por su PID.
- `kill [opciones] <PID>`
- Las señales más comunes:
 - **SIGTERM (15)**: La señal por defecto. Solicita al proceso que termine de forma limpia (guardando archivos, cerrando conexiones, etc.). El proceso puede ignorar esta señal (aunque la mayoría del software bien escrito la respeta).
 - **SIGKILL (9)**: Fuerza la terminación inmediata del proceso. El proceso *no puede* ignorar esta señal. Se usa cuando un proceso no responde a **SIGTERM**. Es una medida de último recurso, ya que el proceso no tiene oportunidad de limpiar sus recursos.

- **SIGHUP (1):** Históricamente, colgar la línea de terminal. Hoy en día se usa a menudo para indicar a los demonios (servicios en segundo plano) que vuelvan a leer sus archivos de configuración sin detenerse y reiniciarse por completo.
- Puedes especificar la señal por nombre (`kill -SIGKILL <PID>`) o por número (`kill -9 <PID>`).

2. **killall:**

- Envía una señal a **todos** los procesos con un nombre específico.
- `killall [opciones] <nombre_proceso>`
- `killall <nombre_proceso>`: Envía SIGTERM a todos los procesos con ese nombre.
- `killall -SIGKILL <nombre_proceso>`: Envía SIGKILL a todos los procesos con ese nombre.
- Es más conveniente que `kill` si quieres detener todas las instancias de un programa, pero debes usarlo con precaución para no detener procesos que no desees.

3. **pkill:**

- Envía una señal a procesos basándose en otros criterios, no solo el nombre exacto, sino también usuario, grupo, etc., y permite usar patrones (expresiones regulares).
- `pkill [opciones] <patrón>`
- `pkill -u <usuario> <patrón>`: Mata procesos del usuario especificado que coincidan con el patrón.
- `pkill -f <patrón>`: Busca el patrón en la línea de comando completa del proceso.
- `pkill -9 <patrón>`: Envía SIGKILL.

Proceso Zombie:

Un proceso zombie es un proceso hijo que ha terminado su ejecución pero cuyo estado (código de salida, etc.) aún no ha sido recogido por su proceso padre. Aparecen en la lista de procesos con estado Z. Los zombies consumen recursos mínimos, pero indican un posible problema en el proceso padre. Solo pueden ser eliminados matando al proceso padre. Un sistema bien diseñado no debería tener procesos zombie persistentes.