

### 103.5 Crear, monitorizar y terminar procesos - Ejercicios

*Nota: Algunos de estos ejercicios implican terminar procesos. Ten cuidado y asegúrate de que entiendes qué procesos estás afectando, especialmente si usas comandos con **sudo**.*

#### Ejercicio 3.5.1: Ejecutando Procesos en Primer y Segundo Plano

- **Objetivo:** Iniciar un proceso en primer plano, enviarlo a segundo plano, traerlo de vuelta y usar la gestión de jobs.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Ejecuta un comando en primer plano que no termine inmediatamente:** Ejecuta `sleep 60`. La terminal quedará bloqueada durante 60 segundos.
  3. **Mientras `sleep 60` se ejecuta, presiona `Ctrl+Z`:** El comando `sleep` se detendrá y la shell te mostrará un mensaje como `[1]+ Stopped sleep 60`. La terminal ahora está libre. El número entre corchetes (1 en este ejemplo) es el número de job.
  4. **Verifica los jobs detenidos/en segundo plano:** Ejecuta `jobs`. Verás la lista de jobs gestionados por la shell, incluyendo `sleep 60` marcado como `Stopped`.
  5. **Envía el job a segundo plano:** Ejecuta `bg`. El comando `sleep 60` ahora se ejecutará en segundo plano (`[1]+ sleep 60 &`). La terminal sigue libre.
  6. **Verifica los jobs de nuevo:** Ejecuta `jobs`. Ahora `sleep 60` debería estar marcado como `Running`.
  7. **Trae el job de vuelta a primer plano:** Ejecuta `fg`. `sleep 60` volverá a bloquear la terminal. Espera a que termine (o presiona `Ctrl+C` para terminarlo, o `Ctrl+Z` para detenerlo de nuevo).
  8. **Ejecuta un comando directamente en segundo plano:** Ejecuta `sleep 120 &`. La shell mostrará el número de job y el PID (ej: `[2] 12345`) y la terminal quedará inmediatamente libre.
  9. **Verifica los jobs y procesos:** Ejecuta `jobs` para ver el job de `sleep 120`. Ejecuta `ps aux | grep sleep` para ver el proceso `sleep 120` listado con su PID.
  10. **Mata el proceso `sleep 120` desde otra terminal o después de que termine (o usa `kill` como en el siguiente ejercicio):** Si no lo matas, terminará solo después de 120 segundos. Si lo matas, el job saldrá de la lista de `jobs`.

#### Ejercicio 3.5.2: Monitorizando Procesos con `ps`, `top` y `htop`

- **Objetivo:** Usar las herramientas estándar para ver qué procesos se están ejecutando en el sistema.
- **Requisitos:** `htop` puede requerir instalación.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.

2. **Muestra tus propios procesos asociados a la terminal:** Ejecuta `ps`. Verás `ps` y la propia shell (`bash`, etc.).
3. **Muestra todos los procesos en formato BSD:** Ejecuta `ps aux`. Desplázate por la larga lista. Identifica columnas como `USER`, `PID`, `%CPU`, `%MEM`, `COMMAND`.
4. **Muestra todos los procesos en formato extendido y con árbol:** Ejecuta `ps -efH`. Busca procesos padre e hijo en la columna `CMD`. Identifica el `PID 1` (`systemd` o `init`).
5. **Usa `top` para ver procesos dinámicamente:** Ejecuta `top`. Observa cómo la lista se actualiza. Identifica los procesos que consumen más CPU o memoria. Presiona `k` para intentar "matar" un proceso (te pedirá el `PID` - ¡no mates procesos críticos!). Presiona `q` para salir.
6. **Usa `htop` (si está instalado):** Ejecuta `htop`. Explora su interfaz más interactiva. Usa las teclas de función en la parte inferior. Identifica cómo matar procesos (`F9`). Presiona `F10` o `q` para salir.
7. **Encuentra el PID de un proceso específico:** Abre otra terminal. En la primera terminal, ejecuta `sleep 300`. En la segunda terminal, ejecuta `ps aux | grep sleep`. Busca la línea que muestra `sleep 300`. El número en la segunda columna es su `PID`. Anótalo.
8. **Ahora usa `ps` filtrando por PID (opcional):** En la segunda terminal, ejecuta `ps -p <PID>` (reemplaza `<PID>` por el número que anotaste). Debería mostrar solo la línea del proceso `sleep 300`.

### Ejercicio 3.5.3: Terminando Procesos con `kill` y `killall`

- **Objetivo:** Terminar procesos utilizando señales.
- **Requisitos:** Continúa con el proceso `sleep 300` del ejercicio anterior, o lanza uno nuevo (`sleep 300 &` en una terminal). Necesitas su `PID`. Necesitas privilegios de superusuario (`sudo`) si el proceso es de otro usuario o es un proceso del sistema que necesita permisos.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y asegúrate de tener el `PID` del proceso `sleep 300` (ej: usando `ps aux | grep sleep`).
  2. **Intenta terminar el proceso de forma amable (`SIGTERM` por defecto):** Ejecuta `kill <PID>` (reemplaza `<PID>`).
  3. **Verifica si el proceso terminó:** Ejecuta `ps aux | grep sleep`. Si el proceso desapareció de la lista, significa que terminó. Si todavía aparece (posiblemente como "Terminated" o "Zombie" brevemente), no respondió a `SIGTERM`.
  4. **Lanza el proceso de nuevo si terminó:** Ejecuta `sleep 300 &` y obtén su nuevo `PID`.
  5. **Termina el proceso de forma forzada (`SIGKILL`):** Ejecuta `kill -9 <PID>` o `kill -SIGKILL <PID>`.

6. **Verifica si el proceso terminó:** Ejecuta `ps aux | grep sleep`. Debería haber desaparecido (quizás brevemente como zombie). SIGKILL es una señal que no se puede ignorar.
7. **Lanza múltiples instancias de un programa:** Ejecuta `sleep 100 &, sleep 110 &, sleep 120 &`.
8. **Verifica que están corriendo:** Ejecuta `ps aux | grep sleep`. Verás los tres procesos con diferentes PIDs.
9. **Mata todas las instancias por nombre con killall:** Ejecuta `killall sleep`.
10. **Verifica que todos los procesos terminaron:** Ejecuta `ps aux | grep sleep`. No deberías ver ninguno (o solo la línea del propio grep).