

200.1 Medición y resolución de problemas de uso de recursos - Ejercicios

Nota: Estos ejercicios implican ejecutar comandos de monitorización y, opcionalmente, simular carga para observar su efecto. Realízalos en tu VM de prueba.

Ejercicio 1.1.1: Usando **top** y **htop** para Monitorizar (Dinámico)

- **Objetivo:** Obtener una vista en tiempo real del estado del sistema y el uso de recursos por proceso.
- **Requisitos:** Acceso a la línea de comandos. Instalar **htop** si no está (`sudo apt install htop` o `sudo dnf install htop`).
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Ejecuta **top**:** Observa la sección superior (resumen del sistema: tareas, CPU, memoria, swap) y la lista de procesos. Identifica las columnas (PID, USER, %CPU, %MEM, COMMAND, etc.).
 3. **Dentro de **top**:** Presiona **P** para ordenar por uso de CPU, **M** para ordenar por uso de memoria. Presiona **1** para ver el uso de CPU por cada núcleo. Presiona **q** para salir.
 4. **Ejecuta **htop**:** Observa su interfaz más colorida y amigable. Las barras de uso de CPU, memoria y swap en la parte superior. Usa las teclas de función (F1-F10) en la parte inferior.
 5. **Dentro de **htop**:** Presiona **F6** para ordenar por una columna (selecciona CPU o MEM). Presiona **F2** para configuraciones. Presiona **F10** o **q** para salir.
 6. **(Opcional) Genera algo de carga de CPU y observa **top/htop**:** En otra terminal, ejecuta `dd if=/dev/zero of=/dev/null &` (esto crea un proceso que consume CPU). Vuelve a la terminal con **top/htop**. Deberías ver el proceso **dd** en la parte superior de la lista, consumiendo CPU. Detén el proceso **dd** (busca su PID en **top/htop** y usa `kill <PID>`, o simplemente mátalos con `kill %<numero_job>` si lo lanzaste con `&`).
 7. **(Opcional) Genera algo de E/S de disco y observa **top/htop**:** En otra terminal, ejecuta `dd if=/dev/zero of=temp_io_test bs=1M count=1000 &` (esto crea un archivo grande). Vuelve a **top/htop**. Observa el valor de **wa** (wait I/O) en el resumen de CPU; debería aumentar mientras **dd** escribe. Elimina el archivo después (`rm temp_io_test`).

Ejercicio 1.1.2: Verificando Memoria con **free**

- **Objetivo:** Obtener un resumen rápido del uso de memoria RAM y swap.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Muestra el uso de memoria (bytes por defecto):** Ejecuta **free**.

3. **Muestra el uso de memoria en formato legible por humanos:** Ejecuta `free -h`. Identifica la memoria total, usada, libre, compartida, en buffer/caché, y el total y uso de swap.

Ejercicio 1.1.3: Obteniendo Instantáneas con `vmstat`

- **Objetivo:** Obtener estadísticas de procesos, memoria, paginación, E/S de bloques y CPU en un intervalo.
- **Requisitos:** Acceso a la línea de comandos. El paquete `sysstat` puede ser necesario (`sudo apt install sysstat` o `sudo dnf install sysstat`).
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Muestra una instantánea (promedio desde el arranque):** Ejecuta `vmstat`.
 3. **Muestra estadísticas cada 2 segundos, 5 veces:** Ejecuta `vmstat 2 5`. Observa cómo los valores (especialmente `r` - procesos ejecutables, `wa` - wait I/O, columnas de memoria y swap) cambian entre intervalos.
 4. **(Opcional) Genera carga de E/S de disco en otra terminal y observa `vmstat`:** Ejecuta `dd if=/dev/zero of=temp_io_test2 bs=1M count=500`. Mientras se ejecuta, observa la columna `wa` en la salida de `vmstat`.

Ejercicio 1.1.4: Monitorizando E/S de Disco con `iostat`

- **Objetivo:** Ver estadísticas detalladas de E/S para tus dispositivos de almacenamiento.
- **Requisitos:** Acceso a la línea de comandos. El paquete `sysstat` debe estar instalado.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Muestra estadísticas de E/S extendidas por dispositivo (promedio desde el arranque):** Ejecuta `iostat -x`. Identifica tus dispositivos de disco (ej: `sda`, `sdb`, `dm-0`). Observa columnas como `r/s` (lecturas por segundo), `w/s` (escrituras por segundo), `rMB/s`, `wMB/s` (velocidad en MB/s), `await` (tiempo promedio de espera para operaciones), `%util` (porcentaje de tiempo que el dispositivo está ocupado).
 3. **Muestra estadísticas extendidas cada 2 segundos para un disco específico:** Ejecuta `iostat -x 2 <nombre_disco>` (ej: `iostat -x 2 sda`).
 4. **(Opcional) Genera carga de E/S de disco en otra terminal y observa `iostat`:** Ejecuta `dd if=/dev/zero of=temp_io_test3 bs=1M count=1000 oflag=direct`. Mientras se ejecuta, observa la salida de `iostat -x 2 <tu_disco>` en la otra terminal. Deberías ver altos valores en las columnas de escritura (`w/s`, `wMB/s`) y `%util`. El `oflag=direct` ayuda a asegurar que los datos se escriban directamente al disco en lugar de quedarse en caché.

Ejercicio 1.1.5: Viendo la Carga Promedio del Sistema con `uptime`

- **Objetivo:** Entender el significado de la carga promedio.
- **Requisitos:** Acceso a la línea de comandos.

- **Desarrollo Paso a Paso:**

1. Abre una terminal.
2. **Muestra el uptime y la carga promedio:** Ejecuta `uptime`.
3. **Observa los tres números al final:** Son la carga promedio para los últimos 1, 5 y 15 minutos.
4. **Identifica el número de núcleos de CPU:** Ejecuta `nproc` o `grep -c '^processor' /proc/cpuinfo`. Este es el número de "unidades de CPU" que tu sistema puede ejecutar concurrentemente.
5. **Interpreta la carga promedio:** Compara los números de carga promedio con el número de núcleos. Una carga consistentemente mayor que el número de núcleos indica que el sistema está sobrecargado.

Ejercicio 1.1.6: Monitorizando Uso de CPU por Núcleo con `mpstat`

- **Objetivo:** Ver cómo se distribuye la carga de CPU entre los núcleos.
- **Requisitos:** Acceso a la línea de comandos. El paquete `sysstat` debe estar instalado.
- **Desarrollo Paso a Paso:**
 1. Abre una terminal.
 2. **Muestra estadísticas de CPU para todos los núcleos en un intervalo:** Ejecuta `mpstat -P ALL`. 2. Observa la línea `all` (promedio) y las líneas para cada núcleo (ej: 0, 1, etc.). Identifica las columnas `%usr`, `%sys`, `%idle`, `%iowait`.
 3. **(Opcional) Genera carga de CPU en otra terminal (dd o un script de bucle infinito: `while true; do :; done`) y observa `mpstat`:** Verás que uno o más núcleos tienen un alto `%usr` o `%sys`. Si tienes un sistema con múltiples núcleos, verás cómo la carga se distribuye.