TOP

Lesson: Monitoring System Health with the `top` Command

The `top` command in Linux provides real-time monitoring of system resource usage, including CPU, memory, processes, and system load. It is one of the most powerful tools for gaining an instant snapshot of system performance.

1. Starting the `top` Command

To launch `top`, simply open a terminal and type:

```
top
```

This will present an interface displaying real-time updates about system resource usage.

2. Understanding the `top` Output

The `top` display is divided into two main sections:

- **Summary Area** (at the top): Provides an overview of system health, including uptime, load averages, and memory usage.
- **Process List** (below the summary): Shows running processes, ordered by resource usage.

Sample `top` Command Output

```
top - 11:23:11 up  5:12,  3 users,  load average: 0.24, 0.32, 0.29
Tasks: 250 total,   1 running, 248 sleeping,   1 stopped,   0 zombie
%Cpu(s):  2.3 us,  0.7 sy,  0.0 ni, 96.9 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  7945.4 total,  2329.5 free,  2031.6 used,  3584.3 buff/cache
MiB Swap:  2048.0 total,  2048.0 free,     0.0 used.  5000.3 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1234 john      20   0  205000  19444   7984 S   5.5   1.2   0:05.24 firefox
 5678 alice     20   0  156000  13200   5400 S   3.8   1.0   0:02.10 chrome
 9876 root      20   0  512000  48000  12000 S   2.3   3.5   0:12.44 mysqld
 5432 bob       20   0  410000  35400   7800 S   1.5   2.9   0:08.39 java
 2468 tom       20   0  380000  29000   9400 S   0.9   2.2   0:01.15 sshd
```

Let’s break down this output into its two key sections:

2.1. The Summary Area

The summary section provides high-level information about system performance.

1. **Uptime and Load Average**:
   - Example: `11:23:11 up  5:12,  3 users,  load average: 0.24, 0.32, 0.29`
     - **11:23:11**: Current time.
     - **up 5:12**: System uptime (how long the system has been running).
     - **3 users**: Number of users currently logged in.
     - **load average**: The average system load over the last 1, 5, and 15 minutes. Lower values indicate less demand on the CPU.
2. **Task Information**:
   - Example: `Tasks: 250 total,   1 running, 248 sleeping,   1 stopped,   0 zombie`
     - **250 total**: Total number of tasks (processes) on the system.
     - **1 running**: Number of currently running tasks.
     - **248 sleeping**: Tasks that are idle or waiting for an event.
     - **1 stopped**: Processes that have been stopped.

    - **0 zombie**: Zombie processes (defunct processes).

3. **CPU Usage**:
    - Example: `%Cpu(s):  2.3 us,  0.7 sy,  0.0 ni, 96.9 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st`
        - **us**: Time spent on user-level processes (not kernel processes).
        - **sy**: Time spent on system-level processes (kernel processes).
        - **ni**: Time spent on processes with changed priority (nice values).
        - **id**: Idle time (time when the CPU is doing nothing).
        - **wa**: Time spent waiting for I/O operations to complete.
        - **hi**: Time spent servicing hardware interrupts.
        - **si**: Time spent servicing software interrupts.
        - **st**: Time lost due to virtual machine "steal time" (when the VM's CPU is stolen by the hypervisor for other tasks).
4. **Memory Usage**:
    - Example: `MiB Mem :  7945.4 total,  2329.5 free,  2031.6 used,  3584.3 buff/cache`
        - **total**: Total physical memory available.
        - **free**: Free memory available.
        - **used**: Memory currently being used by processes.
        - **buff/cache**: Memory used for buffers and cache (freed as needed).
5. **Swap Memory Usage**:
    - Example: `MiB Swap:  2048.0 total,  2048.0 free,     0.0 used.  5000.3 avail Mem`
        - **total**: Total swap space available.
        - **free**: Swap space not used.
        - **used**: Amount of swap memory being used.
        - **avail Mem**: Memory that can be allocated to processes without swapping.

2.2. The Process List

The process list shows details about running processes, ordered by CPU usage by default.

Field	Description
PID	Process ID, a unique identifier for each process.
USER	The user account that owns the process.
PR	Process priority.
NI	Nice value (the lower the value, the higher the priority).
VIRT	Virtual memory size (includes swapped out memory and allocated memory that may not be used).
RES	Resident memory size (physical memory the process is using).
SHR	Shared memory size (shared with other processes).
S	Process state (`R` for running, `S` for sleeping, `D` for disk sleep, `Z` for zombie, etc.).
%CPU	Percentage of CPU time the process is using.
%MEM	Percentage of RAM the process is using.

TIME+	Total CPU time the process has used since it started.
COMMAND	The command or process name.

---

3. Navigating and Customizing `top`

The `top` command allows for dynamic interaction and customization to focus on specific metrics.

1. **Changing Sort Order**:
   - By default, `top` sorts by `%CPU` usage. You can change this by pressing the following keys:
     - `M`: Sort processes by memory usage.
     - `P`: Sort processes by CPU usage (default).
     - `T`: Sort processes by time (cumulative CPU time).
     - `N`: Sort by process ID.
2. **Killing a Process**:
   - You can kill a process directly from `top` by pressing `k`, then entering the **PID** of the process and confirming.
3. **Filtering by User**:
   - Press `u` to filter processes by a specific user. Enter the username to see only their processes.
4. **Changing Refresh Interval**:
   - By default, `top` refreshes every 3 seconds. You can change this by pressing `d` and specifying a new interval (in seconds).
5. **Displaying Specific Fields**:
   - Press `f` to bring up a list of fields that can be displayed or hidden. You can toggle fields on or off by selecting their corresponding letter.
6. **Saving Custom Configuration**:
   - Once you have customized the display, press `W` to save your configuration. This will load the same settings the next time you run `top`.

---

4. Useful `top` Command Options

The `top` command also supports several useful options that can be passed when launching it.

- **Batch Mode (`-b`)**:
  Runs `top` in batch mode, useful for logging or scripting.

  ```
  top -b -n 1
  ```

- **Set Refresh Interval (`-d`)**:
  Set a custom delay (in seconds) between screen updates.

  ```
  top -d 5
  ```

- **Show Only Specific User’s Processes (`-u`)**:
  Filter processes by a specific user.

  ```
  top -u username
  ```

---

5. Practical Use Cases of `top`

1. **

Diagnosing High CPU Usage**:
If your system feels slow, run `top` to identify which processes are consuming the most CPU resources by sorting by `%CPU`. Investigate the process ID (PID) and user responsible for the resource hog.

2. **Monitoring Memory Usage**:
   Use `top` to monitor the system's memory usage, paying attention to the **RES** and **%MEM** fields to track the processes consuming the most memory. This is useful for diagnosing memory leaks.
3. **Tracking System Load**:
   The **load average** values in the summary area provide insight into how busy the system is. If these values are consistently higher than the number of CPU cores, it indicates that the system is overloaded.
4. **Identifying and Killing Misbehaving Processes**:
   If a process is using excessive resources, you can kill it directly from `top` by pressing `k`, entering its PID, and confirming.

---

6. Summary

- `top` is a powerful, real-time tool for monitoring system health, providing detailed information about CPU, memory, and process usage.
- It is highly customizable, allowing users to filter, sort, and view only the most relevant information.
- Knowing how to use `top` effectively can help troubleshoot system performance issues quickly.

Would you like further customization examples, or perhaps help with saving `top` reports for later analysis?