TOP -2

Introduction to the `top` Command

The `top` command is used to display real-time information about system resource usage. It provides a dynamic and continually updated overview of the system's processes, memory usage, and CPU usage, which is especially useful for system administrators.

Basic Syntax

```
top [options]

```

Common options:

- `-b`: Batch mode (non-interactive, used in scripts).
- `-n`: Number of iterations before exiting.
- `-p <PID>`: Monitor specific process(es).
- `-u <USER>`: Display processes for a specific user.

Default `top` Output

When you run `top` without any options, you'll see a display like this:

```
top - 12:34:56 up 5 days, 10:23,  2 users,  load average: 0.15, 0.10, 0.05
Tasks:  250 total,   1 running, 248 sleeping,   0 stopped,   1 zombie
%Cpu(s):  2.0 us,  0.5 sy,  0.0 ni, 97.0 id,  0.5 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  16384.0 total,  12345.0 free,   2048.0 used,   2991.0 buff/cache
MiB Swap:  8192.0 total,   6144.0 free,   2048.0 used.  14236.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1234 root      20   0  101000   5000   1500 S   0.5   0.1   0:00.10 nginx

```

Header Fields Explanation

1. System Summary Information (Top Section)

Example:

```
top - 12:34:56 up 5 days, 10:23,  2 users,  load average: 0.15, 0.10, 0.05

```

- `12:34:56`: Current time.

- `up 5 days, 10:23`: System uptime (days, hours, and minutes).
- `2 users`: Number of logged-in users.
- `load average: 0.15, 0.10, 0.05`: Average system load over the last 1, 5, and 15 minutes (ideally less than the number of CPUs).

---

2. Task Summary

Example:

```
Tasks:  250 total,   1 running, 248 sleeping,   0 stopped,   1 zombie

```

- `250 total`: Total number of processes.
- `1 running`: Number of processes actively running on the CPU.
- `248 sleeping`: Number of idle processes waiting for an event.
- `0 stopped`: Processes that have been stopped.
- `1 zombie`: Dead processes that are not yet removed from the process table.

---

3. CPU Usage

Example:

```
%Cpu(s):  2.0 us,  0.5 sy,  0.0 ni, 97.0 id,  0.5 wa,  0.0 hi,  0.0 si,  0.0 st

```

- `us`: User space CPU usage.
- `sy`: System/kernel space CPU usage.
- `ni`: Nice CPU usage (low-priority processes).
- `id`: Idle CPU time.
- `wa`: Time spent waiting for I/O.
- `hi`: Hardware interrupt CPU time.
- `si`: Software interrupt CPU time.
- `st`: Time stolen by a hypervisor (used in virtualized systems).

---

4. Memory Usage

Example:

```
MiB Mem :  16384.0 total,  12345.0 free,   2048.0 used,   2991.0 buff/cache
```

- `total`: Total physical memory.
- `free`: Free memory available.
- `used`: Memory currently in use.
- `buff/cache`: Memory used for buffers and cache.

---

5. Swap Usage

Example:

```
MiB Swap:   8192.0 total,   6144.0 free,   2048.0 used.  14236.0 avail Mem

```

- `total`: Total swap space.
- `free`: Free swap space.
- `used`: Swap space in use.
- `avail Mem`: Total memory available for new processes.

Process Table (Bottom Section)

Each row represents a process running on the system.

Example:

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1234 root      20   0  101000   5000   1500 S   0.5   0.1   0:00.10 nginx

```

Field Explanations

1. `PID`: Process ID.
2. `USER`: User who owns the process.
3. `PR`: Priority (lower value = higher priority).
4. `NI`: Nice value (affects process priority).
5. `VIRT`: Total virtual memory used by the process.
6. `RES`: Resident memory (non-swapped physical memory used).
7. `SHR`: Shared memory used by the process.
8. `S`: Process state:
   - `R`: Running.

   - `S`: Sleeping.
   - `D`: Uninterruptible sleep.
   - `Z`: Zombie.
   - `T`: Stopped.
9. `%CPU`: Percentage of CPU used by the process.
10. `%MEM`: Percentage of memory used by the process.
11. `TIME+`: Total CPU time used by the process.
12. `COMMAND`: Command or name of the process.

Example Use Cases

1. Find Top CPU Consuming Processes

```
top -b -n 1 | head -n 20

```

- Lists the top 20 CPU-consuming processes.

2. Monitor Specific Process

```
top -p <PID>

```

- Monitors a specific process by its PID.

3. Batch Mode Output

```
top -b -n 1 > top_output.txt

```

- Saves the output of `top` to a file for later analysis.

4. Filter by User

```
top -u username

```

- Shows processes belonging to a specific user.

Let me know if you'd like additional commands or examples!