

# top 命令



性能测试



汪琳仙

2周前发布

17

0

## 前言

top 命令是常用的性能分析工具，可以实时显示 Linux 系统的运行情况，例如内存、CPU、负载以及各个进程的资源占用情况。

## 语法

```
top [-] [d delay] [q] [c] [S] [s] [i] [n] [b]
```

## 参数说明

- d : 改变显示的更新速度，或是在交谈式指令列 ( interactive command ) 按 s
- q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行
- c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称
- S : 累积模式，会将已完成或消失的子进程 ( dead child process ) 的 CPU time 累积起来
- s : 安全模式，将交谈式指令取消，避免潜在的危机
- i : 不显示任何闲置 (idle) 或无用 (zombie) 的进程
- n : 更新的次数，完成后将会退出 top
- b : 批次档模式，搭配 n 参数一起使用，可以用来将 top 的结果输出重定向到指定文件

## 输出参数含义

top 界面主要分为三个部分，前 5 行展示的是系统的整体情况，第 6 行用于输入操作命令，第 7 行及下面部分是系统中每个进程的具体信息。

```
top - 15:51:12 up 55 days, 23:06, 1 user, load average: 0.17, 0.35, 0.48
Tasks: 467 total, 1 running, 385 sleeping, 0 stopped, 81 zombie
%Cpu(s): 1.5 us, 1.4 sy, 0.0 ni, 97.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32668676 total, 2020032 free, 12048796 used, 18599848 buff/cache
KiB Swap: 16449532 total, 9387960 free, 7061572 used, 19630552 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 4140 root        20   0 3897440 124040 5036  S   9.0   0.4   6051.08 cadvisor
17523 root        20   0 1125.2g 97780 34952  S   5.0   0.3   14:07.11 chrome
28900 root        20   0 9858.5m 1.1g 14160  S   3.7   3.6   4:10.61 java
1123  root        20   0 750580 21996 6068  S   3.0   0.1   1830:16 gitlab-runner
3717  root        20   0 717108 25316 3364  S   2.7   0.1   1415:40 process-exporte
28901 root        20   0 8413636 616400 13660  S   2.7   1.9   3:36.14 java
7629  netdata    20   0 58184 7652 1836  S   2.3   0.0   1892:13 apps.plugin
4063  nfsnobo+  20   0 719924 18732 4644  S   2.0   0.1   1900:32 node_exporter
1810  netdata    20   0 889112 564960 4152  S   1.7   1.7   1696:59 netdata
17474 root        20   0 32.4g 30024 12788  S   1.3   0.1   4:26.05 chrome
6691  root        20   0 15.8g 602048 7604  S   1.0   1.8   119:07.15 java
20064 root        20   0 15.8g 1.8g 7664  S   1.0   5.8   265:34.47 java
2768  root        20   0 9913.7m 208592 3608  S   0.7   0.6   186:25.68 java
19914 root        20   0 12.6g 152176 6484  S   0.7   0.5   106:36.09 java
27845 root        20   0 162500 2672 1572  R   0.7   0.0   0:00.90 top
    9  root        20   0 0 0 0  S   0.3   0.0   125:52.37 rcu_sched
   802 root        20   0 1193068 92588 3952  S   0.3   0.3   53:55.43 fileserv
 3002 root        20   0 9881.0m 171260 3752  S   0.3   0.5   176:20.59 java
```



## 第一行

```
top - 15:51:12 up 55 days, 23:06, 1 user, load average: 0.17, 0.35, 0.48
```

- top: 命令名称
- 15:51:12: 系统当前时间
- up 55 days, 23:06: 主机已经运行的时间, 表示已经运行了55天23小时6分钟 (和 uptime 命令相等, 时间越长, 说明机器越稳定)
- 1 user: 当前用户连接数
- load average: 0.17, 0.35, 0.48: 系统一定时间内的平均负载, 分别是1分钟、5分钟、15分钟。

这里的系统平均负载在性能测试分析的时候经常会用到。所谓系统负载就是: 在一段时间内, CPU 正在处理以及等待 CPU 处理的进程数之和。系统在同一时间运行的进程数和系统 CPU 核数相关, 可以作为衡量 CPU 负载的依据。当 Load Average 的数值等于总核数时, 说明 CPU 在满负荷运行, 如果长期大于总核数, 可能 CPU 存在瓶颈。

## 第二行

```
Tasks: 467 total, 1 running, 385 sleeping, 0 stopped, 81 zombie
```

- Tasks: 任务, 也就是进程
- 467 total: 当前总共有467个任务, 也就是467个进程
- 1 running: 1个进程正在运行
- 385 sleeping: 385个进程正在休眠
- 0 stopped: 0个停止的进程
- 81 zombie: 81个僵尸进程

## 第三行

```
%Cpu(s): 1.5 us, 1.4 sy, 0.0 ni, 97.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
```

在性能测试分析的时候, 经常用第三行的数据来分析 CPU 的状态, 然后进一步定位性能瓶颈。

- %Cpu(s): CPU使用率
- us: 用户空间占用 CPU 时间的百分比, 指用户进程对 CPU 的消耗 (如果数值长期超过 80%, 且上下文切换较低, 可能需要优化用户程序)
- sy: 内核空间占用 CPU 时间的百分比, 指系统上下文切换对 CPU 的消耗 (如果数值较高, 说明服务器开了太多的进程或者线程)
- ni: 用户进程空间内改变过优先级的进程占用 CPU 时间的百分比
- id: 空闲 CPU 时间的百分比
- wa: 等待输入输出的 CPU 时间百分比, 指 CPU 在等待 IO 操作完成所花费的时间 (如果数值较高, 说明服务器有频繁的 IO 操作, 可能存在 IO 瓶颈)
- hi: CPU 硬中断时间百分比 (硬中断是硬盘、网卡等硬件设备发送给 CPU 的中断消息)
- si: CPU 软中断时间百分比 (软中断是由程序发出的中断)
- st: 被强制等待 (involuntary wait) 虚拟 CPU 时间的百分比, 此时 Hypervisor 在为另一个虚拟处理器服务

CPU 时间百分比的计算: CPU 时间由分离的时间片 (ticks) 组成, 这个时间片就是 CPU 工作的最小单位。如果 CPU 有100个时间片, 其中 60 个时间片消耗在用户空间, 则 us 就是 60%。

## 第四行

```
KiB Mem : 32668676 total, 2020032 free, 12048796 used, 18599848 buff/cache
```



- 3858.4 free: 空闲内存量
- 3369.3 used: 已使用的内存量
- 828.07 buff/cache: 用作内核缓存的内存量

## 第五行

KiB Swap: 16449532 total, 9387960 free, 7061572 used. 19630552 avail Mem

- KiB Swap: 交换空间（虚拟内存，当内存不足的时候，会把一部分硬盘空间虚拟成内存使用，在做性能测试过程时最好禁用 Swap）
- 16449532 total: 交换区总量
- 9387960 free: 空闲交换区总量
- 7061572 used: 使用的交换区总量
- 19630552 avail Mem: 可用于启动一个新应用的内存（物理内存），和free不同，它计算的是可回收的page cache和memory slab。

第四行和第五行输出信息，等同于使用 free -m 命令。

## 交互命令

在 top 命令显示的页面还可以输入以下按键执行相应的功能（区分大小写）

- h: 显示帮助页面
- l: 多行/单行显示 CPU 信息
- c: 显示产生进程的完整命令，等同于 -c 参数，再次按下 c 键，变为默认显示
- f: 可以选择需要展示的项目
- M: 根据驻留内存大小（RES）排序
- N: 根据 pid 进行排序
- P: 根据CPU使用百分比大小进行排序
- T: 根据时间/累计时间进行排序
- E: 切换顶部内存显示单位
- e: 切换进程内存显示单位
- l: 切换显示平均负载和启动时间信息
- t: 切换显示CPU状态信息
- m: 切换显示内存信息
- k: 给某一个 pid 一个信号，可以用来杀死进程
- r: 给某个 pid 重新定制一个 nice 值（即优先级）
- B: 关闭/开启粗体
- Z: 切换颜色
- q: 退出 top（也可以用 ctrl+c 退出）

## 进程信息

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
-----	------	----	----	------	-----	-----	---	------	------	-------	---------

- PID: 进程 id
- USER: 进程所有者
- PR: 进程的优先级，越小优先级越高
- NI: nice值，负值表示高优先级，正值表示低优先级



- SHR: 进程使用的共享内存, 单位kb
- S: 进程状态 (S表示休眠, R表示正在运行, Z表示僵死状态, N表示该进程优先值为负数, I表示空闲状态)
- %CPU: 进程占用的CPU使用率
- %MEM: 进程使用的物理内存和总内存的百分比
- TIME+: 进程使用的CPU时间总计, 单位1/100秒
- COMMAND: 命令行

发表评论



请留下您的宝贵意见 (支持md语法)

可通过拖放/粘贴或者选择图片

预览

发布

暂无评论

