

进程管理

查询进程

free：打印系统情况和内存情况

```
root@iZ2zeaxmpy0c41v2i7xvz2Z ~ # free
              total        used        free      shared  buff/cache   available
Mem:           1929508        267372        246240          4240        1415896        1470456
Swap:              0              0              0
root@iZ2zeaxmpy0c41v2i7xvz2Z ~ # free -h
              total        used        free      shared  buff/cache   available
Mem:             1.8Gi         261Mi         240Mi          4.0Mi         1.3Gi         1.4Gi
Swap:              0B              0B              0B
```

输出简介

Mem 行(第二行)是内存的使用情况。

Swap 行(第三行)是交换空间的使用情况。

total 列显示系统总的可用物理内存和交换空间大小。

used 列显示已经被使用的物理内存和交换空间。

free 列显示还有多少物理内存和交换空间可用使用。

shared 列显示被共享使用的物理内存大小。

buff/cache 列显示被 buffer 和 cache 使用的物理内存大小。

available 列显示还可以被应用程序使用的物理内存大小。

buffer和cache

buffer:在操作系统中指 buffer cache，中文一般翻译为 "缓冲区",它只是把磁盘上的块直接搬到内存中而不关心块中究竟存放的是什么格式的文件

cache 在操作系统中指 page cache，中文一般翻译为 "页高速缓存"。页高速缓存是内核实现的磁盘缓存。它主要用来减少对磁盘的 I/O 操作。页高速缓存缓存的是内存页面，当访问的数据在cache中存在时就不需要再去磁盘读取，可以提高性能。

free和available

free 是真正尚未被使用的物理内存数量。available 是从应用程序的角度看到的可用内存数量。为了提升磁盘操作的性能会用一部分内存去缓冲磁盘数据，也就是buffer和cache。因此会损失一部分内存。

swap

swap space: 是磁盘上的一块区域，可以是一个分区，也可以是一个文件。所以具体的实现可以是 swap 分区也可以是 swap 文件。当系统物理内存吃紧时，Linux 会将内存中不常访问的数据保存到 swap 上，这样系统就有更多的物理内存为各个进程服务，而当系统需要访问 swap 上存储的内容时，再将 swap 上的数据加载到内存中，这就是常说的换出和换入。交换空间可以在一定程度上缓解内存不足的情况，但是它需要读写磁盘数据，所以性能不是很高

ps：报告当前进程状态

```
ps [options] [--help]
```

常用参数:

-ef: 查询正在运行的进程信息

```
root@iZ2zeaxmpy0c4lv2i7xvz2Z ~ # ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root          1        0  0 Sep06 ?        00:00:04 /sbin/init noibrs
root          2        0  0 Sep06 ?        00:00:00 [kthreadd]
root          3        2  0 Sep06 ?        00:00:00 [rcu_gp]
root          4        2  0 Sep06 ?        00:00:00 [rcu_par_gp]
root          6        2  0 Sep06 ?        00:00:00 [kworker/0:0H-kblockd]
root          9        2  0 Sep06 ?        00:00:00 [mm_percpu_wq]
root         10        2  0 Sep06 ?        00:00:00 [ksoftirqd/0]
root         11        2  0 Sep06 ?        00:02:06 [rcu_sched]
root         12        2  0 Sep06 ?        00:00:00 [migration/0]
root         13        2  0 Sep06 ?        00:00:00 [idle_inject/0]
root         14        2  0 Sep06 ?        00:00:00 [cpuhp/0]
root         15        2  0 Sep06 ?        00:00:00 [cpuhp/1]
```

-aux: 显示所有包含其他使用者的行程

```
root@iZ2zeaxmpy0c4lv2i7xvz2Z ~ # ps -aux
USER          PID  %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1   0.0  0.7 169496 13764 ?        Ss   Sep06   0:04 /sbin/init noibrs
root           2   0.0  0.0      0      0 ?        S    Sep06   0:00 [kthreadd]
root           3   0.0  0.0      0      0 ?        I<   Sep06   0:00 [rcu_gp]
root           4   0.0  0.0      0      0 ?        I<   Sep06   0:00 [rcu_par_gp]
root           6   0.0  0.0      0      0 ?        I<   Sep06   0:00 [kworker/0:0H-kblockd]
root           9   0.0  0.0      0      0 ?        I<   Sep06   0:00 [mm_percpu_wq]
root          10   0.0  0.0      0      0 ?        S    Sep06   0:00 [ksoftirqd/0]
root          11   0.0  0.0      0      0 ?        I    Sep06   2:06 [rcu_sched]
root          12   0.0  0.0      0      0 ?        S    Sep06   0:00 [migration/0]
root          13   0.0  0.0      0      0 ?        S    Sep06   0:00 [idle_inject/0]
root          14   0.0  0.0      0      0 ?        S    Sep06   0:00 [cpuhp/0]
root          15   0.0  0.0      0      0 ?        S    Sep06   0:00 [cpuhp/1]
root          16   0.0  0.0      0      0 ?        S    Sep06   0:00 [idle_inject/1]
```

pgrep: 查找进程ID

常用参数:

-l 查找匹配进程, 同时显示进程名和PID

-o 当匹配多个进程时, 显示进程号最小的那个

-n 当匹配多个进程时, 显示进程号最大的那个

```
root@iZ2zeaxmpy0c4lv2i7xvz2Z ~ # pgrep -l re
2 kthreadd
23 rcu_tasks_kthre
26 oom_reaper
83 devfreq_wq
89 ecryptfs-kthrea
459 systemd-resolve
root@iZ2zeaxmpy0c4lv2i7xvz2Z ~ # pgrep -l re -o
2 kthreadd
root@iZ2zeaxmpy0c4lv2i7xvz2Z ~ # pgrep -l re -n
459 systemd-resolve
```

pstree: 以树状显示进程派生关系

常用参数

-a 显示每个程序的完整指令, 包含路径, 参数或是常驻服务的标示

-c 不使用精简标示法

-G 使用VT100终端机的列绘图字符

-h 列出树状图时, 特别标明现在执行的程序

```
root@iZ2zeaxmpy0c4lv2i7xvz2Z:~# nstree -h
systemd├─AliSecGuard─6*[{AliSecGuard}]
│├─AliYunDun─24*[{AliYunDun}]
│├─AliYunDunUpdate─5*[{AliYunDunUpdate}]
│├─CmsGoAgent.linux├─exe─11*[{exe}]
││└─5*[{CmsGoAgent.linux}]
│├─ModemManager─2*[{ModemManager}]
│├─NetworkManager─2*[{NetworkManager}]
│├─accounts-daemon─2*[{accounts-daemon}]
│├─acpid
│├─2*[agetty]
│├─aliyun-service─7*[{aliyun-service}]
│├─assist_daemon─7*[{assist_daemon}]
│├─atd
│├─avahi-daemon─avahi-daemon
│├─chronyd─chronyd
│├─cron
│├─cups-browsed─2*[{cups-browsed}]
│├─cupsd
│├─dbus-daemon
│├─2*[kerneloops]
│├─networkd-dispatcher
│├─polkitd─2*[{polkitd}]
│├─rsyslogd─3*[{rsyslogd}]
│├─rtkit-daemon─2*[{rtkit-daemon}]
│├─sshd├─sshd─zsh─su─zsh─su─zsh─su─zsh
││├─2*[sshd─sftp-server]
││├─sshd─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─pstore
││├─sshd─sshd─zsh
││└─sshd─sshd─sftp-server
```

```
systemd├─AliSecGuard─6*[{AliSecGuard}]
│├─AliYunDun─24*[{AliYunDun}]
│├─AliYunDunUpdate─5*[{AliYunDunUpdate}]
│├─CmsGoAgent.linux├─exe─11*[{exe}]
││└─5*[{CmsGoAgent.linux}]
│├─ModemManager─2*[{ModemManager}]
│├─NetworkManager─2*[{NetworkManager}]
│├─accounts-daemon─2*[{accounts-daemon}]
│├─acpid
│├─2*[agetty]
│├─aliyun-service─7*[{aliyun-service}]
│├─assist_daemon─7*[{assist_daemon}]
│├─atd
│├─avahi-daemon─avahi-daemon
│├─chronyd─chronyd
│├─cron
│├─cups-browsed─2*[{cups-browsed}]
│├─cupsd
│├─dbus-daemon
│├─2*[kerneloops]
│├─networkd-dispatcher
│├─polkitd─2*[{polkitd}]
│├─rsyslogd─3*[{rsyslogd}]
│├─rtkit-daemon─2*[{rtkit-daemon}]
│├─sshd├─sshd─zsh─su─zsh─su─zsh─su─zsh
││├─2*[sshd─sftp-server]
││├─sshd─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─su─zsh─pstore
││├─sshd─sshd─zsh
││└─sshd─sshd─sftp-server
```

终止进程

kill：删除执行中的程序和工作

-l 信号，若果不加信号的编号参数，则使用“-l”参数会列出全部的信号名称

-a 当处理当前进程时，不限制命令名和进程号的对应关系

-p 指定kill 命令只打印相关进程的进程号，而不发送任何信号

-s 指定发送信号

-u 指定用户

- kill 9 强制删除程序

pkill批量按照进程死进程

```
pkill [OPTIONS] <PATTERN>
```

1 (HUP)：重新加载进程

9 (KILL)：杀死进程

15 (TERM)：完美地停止一个进程

进程监控

top：显示当前系统进程情况, 内存,CPU等信息

使用格式：

top [OPTIONS]

参数说明：

-d: 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变之。

-p:通过指定监控进程ID来仅仅监控某个进程的状态。

-q:该选项将使top没有任何延迟的进行刷新。如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。

-S: 指定累计模式。

-s: 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。

-i: 使top不显示任何闲置或者僵死进程。

-c:显示整个命令行而不只是显示命令名。

```
top - 11:09:14 up 4 days, 7:04, 3 users, load average: 0.04, 0.02, 0.00
Tasks: 154 total, 1 running, 152 sleeping, 1 stopped, 0 zombie
%Cpu(s): 0.5 us, 0.3 sy, 0.0 ni, 99.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1884.3 total, 308.9 free, 262.3 used, 1313.0 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 1434.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
932	root	10	-10	113480	21568	14812	S	1.0	1.1	73:05.00	AliYunDun
572	root	20	0	928940	21232	11952	S	0.7	1.1	10:18.90	exe
1042	root	10	-10	424028	5968	5476	S	0.3	0.3	2:47.21	AliSecGuard
22375	root	20	0	11852	3796	3148	R	0.3	0.2	0:00.01	top
1	root	20	0	169496	13764	8476	S	0.0	0.7	0:04.84	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.44	ksoftirqd/0
11	root	20	0	0	0	0	I	0.0	0.0	2:06.84	rcu_sched
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.82	migration/0
13	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1

dstat: 实时监控磁盘,CPU等

简答题

- 假设你在你的云主机上安装了 MySQL 数据库，请问你要如何确认它是否在正确运行，以及如果它在运行的话，应该如何结束它？
 - 在 `ps -ef | grep sql` 查找Mysql相关进程，记录pid
 - 利用 `kill 9 pid` 将进程结束