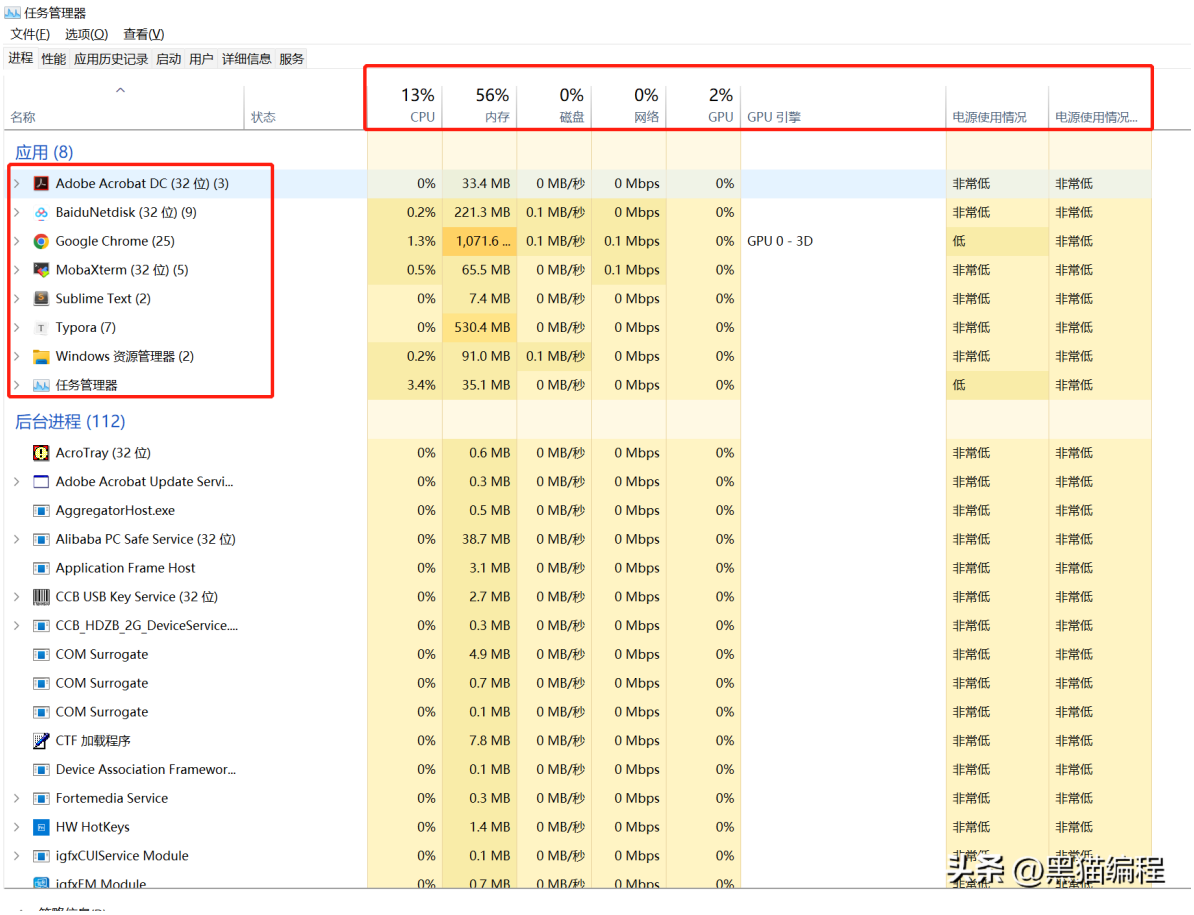
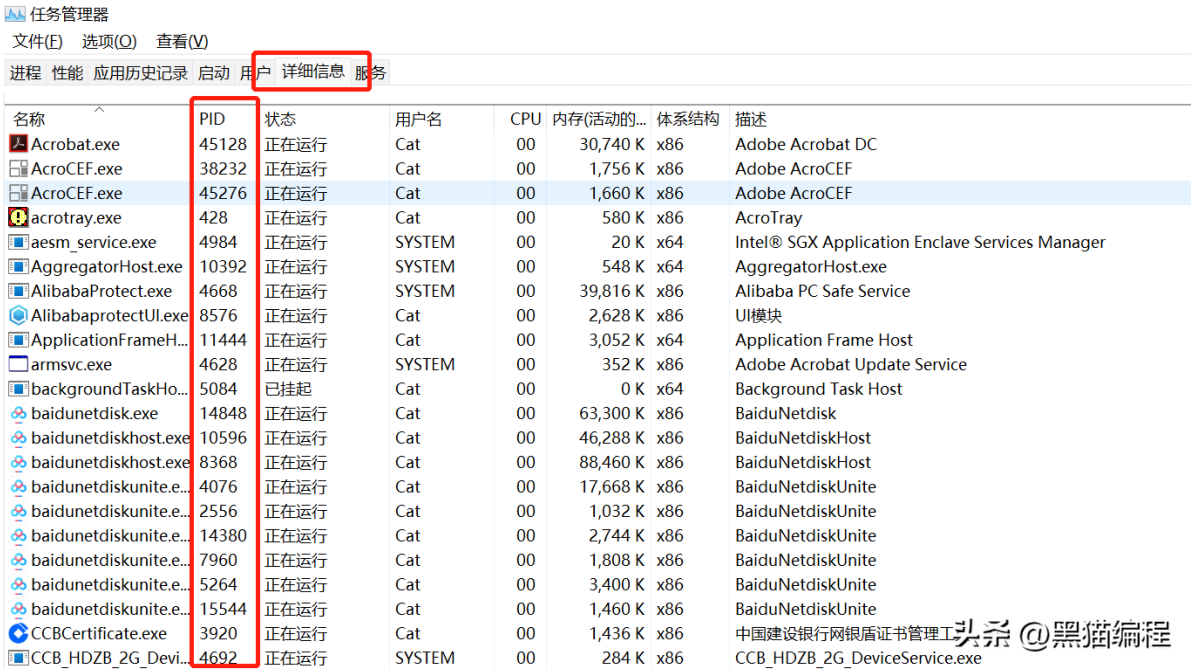
什么是进程?

当我们打开使用某个软件，操作系统都会启动一个进程。有些软件比较复杂，包括主进程和一系列子进程。比如Windows系统当中*Ctrl+Alt+Del组合键*，打开任务管理器：



应用列显示目前正在使用的软件，后台进程是用户不可见的系统服务程序，而且右侧显示每个进程的CPU、内存、磁盘、网络等详细信息。



点击详细信息，PID（process id）就是当前进程号，进程号是动态变化的，比如，我将第一个PDF阅读器关闭掉重新打开，再显示的就是新的进程号。

chrome.exe	44204	正在运行	Cat	00	7,048 K	x64	Google Chrome
chrome.exe	49044	正在运行	Cat	00	38,400 K	x64	Google Chrome
chrome.exe	10024	正在运行	Cat	00	102,248 K	x64	Google Chrome
chrome.exe	7268	正在运行	Cat	00	608 K	x64	Google Chrome
chrome.exe	12152	正在运行	Cat	00	353,744 K	x64	Google Chrome
chrome.exe	14348	正在运行	Cat	00	16,144 K	x64	Google Chrome
chrome.exe	11784	正在运行	Cat	00	3,500 K	x64	Google Chrome
chrome.exe	7336	正在运行	Cat	00	1,572 K	x64	Google Chrome
chrome.exe	5252	正在运行	Cat	00	1,604 K	x64	Google Chrome
chrome.exe	24296	正在运行	Cat	00	12,368 K	x64	Google Chrome
chrome.exe	24792	正在运行	Cat	00	233,172 K	x64	Google Chrome
chrome.exe	27592	正在运行	Cat	00	60,944 K	x64	Google Chrome
chrome.exe	23952	正在运行	Cat	00	32,224 K	x64	Google Chrome
chrome.exe	28648	正在运行	Cat	00	11,476 K	x64	Google Chrome
chrome.exe	28552	正在运行	Cat	00	11,504 K	x64	Google Chrome
chrome.exe	44208	正在运行	Cat	00	26,100 K	x64	Google Chrome
chrome.exe	39644	正在运行	Cat	00	20,544 K	x64	Google Chrome
chrome.exe	42336	正在运行	Cat	00	9,424 K	x64	Google Chrome
chrome.exe	29108	正在运行	Cat	00	83,816 K	x64	Google Chrome
chrome.exe	41360	正在运行	Cat	00	6,484 K	x64	Google Chrome
chrome.exe	40648	正在运行	Cat	00	14,808 K	x64	Google Chrome
chrome.exe	46544	正在运行	Cat	00	37,564 K	x64	Google Chrome
chrome.exe	34436	正在运行	Cat	00	20,716 K	x64	Google Chrome
chrome.exe	31328	正在运行	Cat	00	129,732 K	x64	Google Chrome
chrome.exe	50572	正在运行	Cat	00	42,648 K	x64	Google Chrome
ChsIME.exe	9084	正在运行	Cat	00	16 K	x64	Microsoft IME
ChsIME.exe	12792	正在运行	SYSTEM	00	1,272 K	x64	Microsoft IME
conhost.exe	4332	正在运行	SYSTEM	00	76 K	x64	控制台窗口主机
conhost.exe	8388	正在运行	Cat	00	272 K	x64	控制台窗口主机
conhost.exe	50380	正在运行	Cat	00	5,320 K	x64	控制台窗口主机
csrss.exe	684	正在运行	SYSTEM	00	708 K	x64	Client Server Runtime Process
csrss.exe	804	正在运行	SYSTEM	00	980 K	x64	Client Server Runtime Process
ctfmon.exe	7868	正在运行	Cat	00	8,192 K	x64	CTF 加载程序
D4Mon_CCB.exe	5400	正在运行	SYSTEM	00	396 K	x86	OnKeyMon

如上图所示，谷歌浏览器就是开启了多个子进程。

简单来说，程序是人使用计算机语言编写的，可以实现一定功能，并且可以执行的代码集合，而进程是正在执行中的程序。

top动态查看进程

```
root@cat:~# top
top - 12:10:24 up 380 days, 19:35, 18 users,  load average: 0.11, 0.16, 0.12
Tasks: 171 total,   2 running, 169 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.5 us,  2.2 sy,  0.0 ni, 95.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3792.9 total,    229.3 free,   2711.6 used,    852.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    768.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
1678653 root      10 -10  131364  34360   8004 S   1.7   0.9 256:28.88 AliYunDunMonito
3133804 root      20   0 1250268 240508   4244 S   0.7   6.2 852:29.21 python
3614417 root      20   0   40124  11784   6276 R   0.7   0.3   0:00.02 python3
    491 root      20   0 1197092  29152   4020 S   0.3   0.8   1493:07 containerd
    783 root      20   0 1357752  54140      0 S   0.3   1.4   1357:38 dockerd
  55475 root      20   0 3497136  76852   1876 S   0.3   2.0   3086:02 python3
1678643 root      10 -10   78564   9144   4556 S   0.3   0.2  39:10.46 AliYunDun
2560949 root      20   0  114100   4484   1948 S   0.3   0.1  74:29.25 containerd-shim
2575367 root      20   0  114100   4724   2340 S   0.3   0.1  68:43.39 containerd-shim
2578659 nobody    20   0   64676  45448   2396 S   0.3   1.2 101:13.83 dramatiq
2578660 nobody    20   0   64504  45580   2676 S   0.3   1.2 100:47.95 dramatiq
2578811 systemd+  20   0   21576   2656    744 S   0.3   0.1 103:14.31 redis-server
3614098 root      20   0   13896   9020   7464 S   0.3   0.2   0:00.03 sshd
```

第一行信息：

内 容	说 明
12:10:24	系统当前时间
up 380 days	系统的运行时间.本机已经运行380天
18 users	当前登录了18个用户
load average: 0.11,0.16, 0.12	系统在之前 1 分钟、5 分钟、15 分钟的平均负载。如果 CPU 是单核的，则这个数值超过 1 就是高负载：如果 CPU 是四核的，则这个数值超过 4 就是高负载。

第二行信息：

Tasks: 171 total	系统中的进程总数
2 running	正在运行的进程数
169 sleeping	睡眠的进程数
0 stopped	正在停止的进程数
0 zombie	僵尸进程数

第三行信息：

内 容	说 明
Cpu(s)	用户模式占用的 CPU 百分比
sy	系统模式占用的 CPU 百分比
ni	改变过优先级的用户进程占用的 CPU 百分比
id	idle缩写，空闲 CPU 占用的 CPU 百分比
wa	等待输入/输出的进程占用的 CPU 百分比
hi	硬中断请求服务占用的 CPU 百分比
si	软中断请求服务占用的 CPU 百分比
st	st（steal time）意为虚拟时间百分比，就是当有虚拟机时，虚拟 CPU 等待实际 CPU 的时间百分比

第四行信息：

内 容	说 明
Mem	物理内存的总量，单位为KB
used	已经使用的物理内存数量
free	空闲的物理内存数量。我们使用的是虚拟机，共分配了 628MB内存，所以只有53MB的空闲内存
buff/cache	作为缓冲的内存数量

第五行信息：

内 容	说 明
Swap	交换分区（虚拟内存）的总大小
used	已经使用的交换分区的大小
free	空闲交换分区的大小
avail Mem	可用内存

swap交换分区： 一般情况下为内存的1~2倍，但是尽量不要超过2G。

```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    762 mysql     20   0 1792064 431128      0 S   0.0  11.1 433:13.72 mysqld
3133804 root      20   0 1250268 240508   4244 S   1.0   6.2 852:36.43 python
2575835 root      20   0  824716 225552   5032 S   0.0   5.8   9:19.78 gunicorn
2575956 root      20   0  824720 194352   5092 S   0.0   5.0   9:08.12 gunicorn
2575903 root      20   0  824764 150360   5048 S   0.0   3.9   9:07.05 gunicorn
    238 root      19  -1  207124 130696 129320 S   0.0   3.4 104:02.31 systemd-journal
2575935 root      20   0  824848 129840   4684 S   0.0   3.3   9:08.25 gunicorn
  61732 root      20   0  974636  96692   3448 S   0.0   2.5   1:14.75 node
1063481 root      20   0  970324  89744    780 S   0.0   2.3   0:41.32 node
  62713 root      20   0  966852  87072   1588 S   0.0   2.2   0:29.29 node
 110208 root      20   0  964232  86116   4240 S   0.0   2.2   1:12.29 node
  59031 root      20   0  961872  84228   3912 S   0.0   2.2   0:42.11 node
  55475 root      20   0 3497136  76852   1876 S   0.3   2.0   3086:07 python3
  56861 root      20   0  957572  74888    960 S   0.0   1.9   0:04.55 node
2578656 12000     20   0   83264  68992   4956 S   0.0   1.8  67:56.40 gunicorn
2578655 12000     20   0   81372  67668   4520 S   0.0   1.7  59:58.58 gunicorn
    783 root      20   0 1357752  54140      0 S   0.0   1.4   1357:40 dockerd
2578623 70        20   0  194128  47200  45968 S   0.0   1.2   0:29.49 postgres
2578660 nobody    20   0   64504  45580   2676 S   0.3   1.2 100:49.12 dramatiq
2578659 nobody    20   0   64676  45448   2396 S   0.3   1.2 101:15.07 dramatiq
3133764 root      20   0   63356  39820   1484 S   0.0   1.0   0:00.42 python
```


输入M按照内存占用从高到低排序。

```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
1678653 root      10 -10  132492  35284   8004 S   1.7   0.9 256:43.61 AliYunDunMonito
3133804 root      20   0 1250268 240508   4244 S   0.7   6.2 852:36.85 python
    621 root      20   0 1225160  18644   5828 S   0.3   0.5   1222:17 exe
  55475 root      20   0 3497136  76852   1876 S   0.3   2.0   3086:07 python3
2578629 nobody    20   0   23108  13684   1128 S   0.3   0.4  16:09.52 dramatiq
2578659 nobody    20   0   64676  45448   2396 S   0.3   1.2 101:15.07 dramatiq
      1 root      20   0  170336   7848   3544 S   0.0   0.2 626:25.91 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:02.88 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  15:44.49 ksoftirqd/0
     11 root      20   0       0      0      0 I   0.0   0.0 140:08.80 rcu_sched
     12 root      rt   0       0      0      0 S   0.0   0.0   1:11.50 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
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     17 root      rt   0       0      0      0 S   0.0   0.0   1:11.99 migration/1
     18 root      20   0       0      0      0 S   0.0   0.0  15:48.17 ksoftirqd/1
     20 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-kblockd
```


输入P按照CPU占用从高到低排序。

free查看内存使用情况

```
root@cat:~# free
               total        used        free      shared  buff/cache   available
Mem:         3883924     2736680      120448       55772     1026796      831796
Swap:              0           0           0
root@cat:~# free -m
               total        used        free      shared  buff/cache   available
Mem:            3792        2672         120          54         999         812
Swap:              0           0           0
```

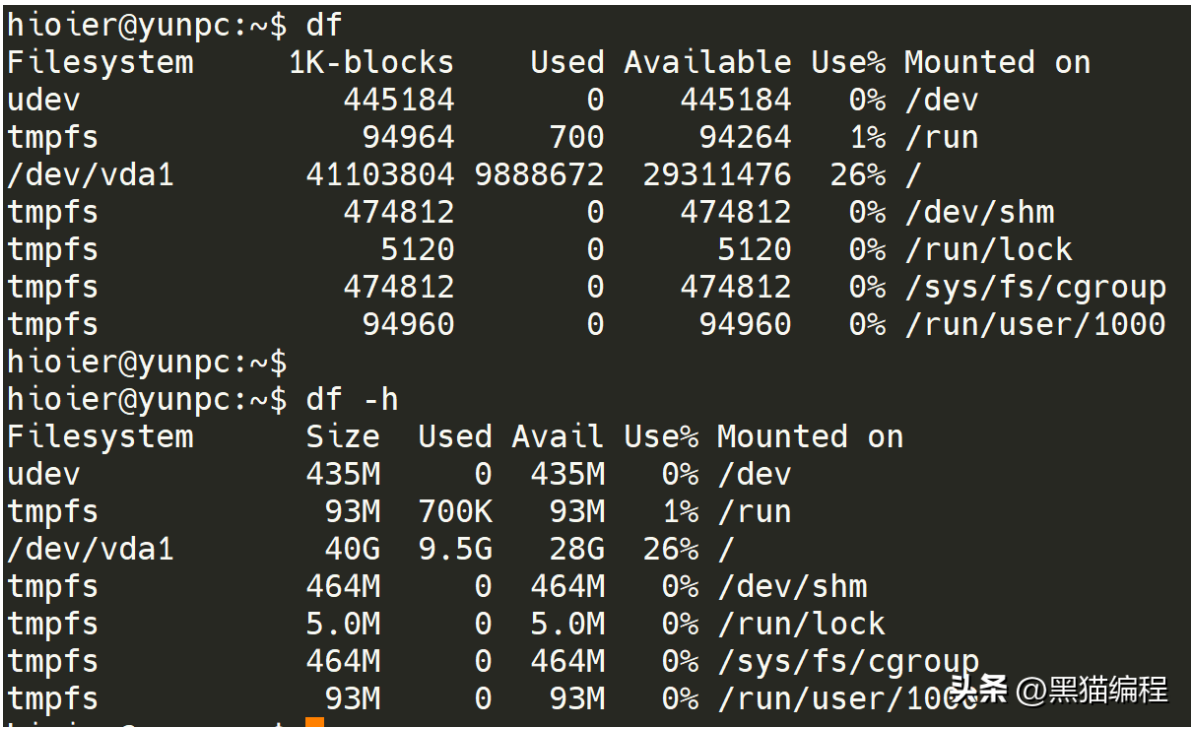
```
# free [选项]



选项说明：

-m ： 以MB的形式显示内存大小
```

df查看磁盘剩余空间




```
# df [选项]

-h ： 以较高的可读性显示磁盘剩余空间大小
```

Filesystem	磁盘名称
Size	总大小
Used	被使用的大小
Avail	剩余大小
Use%	使用百分比
Mounted on	挂载路径（相当于Windows 的磁盘符）

ps静态查看系统进程的信息

```
hioier@yunpc:~$ ps -ef | grep "cron"
root         460       1  0  2022 ?        00:00:05 /usr/sbin/cron -f
hioier   3955042 3953914  0 15:42 pts/0    00:00:00 grep --color=auto cron
hioier@yunpc:~$ ps aux | grep "cron"
root         460  0.0  0.1   9416  1892 ?        Ss    2022   0:05 /usr/sbin/cron -f
hioier   3955092  0.0  0.0   9032   720 pts/0    S+   15:42   0:00 grep --color=auto cron
```


```
# ps [选项]



选项说明：

-e ： 等价于“-A”，表示列出全部（all）的进程

-f ： 表示full，显示全部的列（显示全字段）
```

netstat查询网络访问信息

```
# netstat [选项] | grep 进程名称



选项说明：

-t：表示只列出tcp 协议的连接（tcp协议与udp协议）

-n：表示将地址从字母组合转化成ip 地址，将协议转化成端口号来显示 10.1.1.10:80

-l：表示过滤出"state（状态）"列中其值为LISTEN（监听）的连接

-p：表示显示发起连接的进程pid 和进程名称
```

```
hioier@yunpc:~$ netstat -t | grep ssh
tcp        0      0 yunpc:ssh               58.39.4.19:21023        ESTABLISHED
tcp        0     48 yunpc:ssh               58.39.4.19:21021        ESTABLISHED
```


kill终止进程

kill 进程名是正常终止进程，然而有些进程不会终止，因此可以用kill 9 进程名强制终止进程，向指定进程发送一个强制终止信号，如果进程还包括很多子进程，需要使用killall 进程名。

如果root用户想要终止某个其他用户的所有进程可以使用killall -u 用户名。