



专题7-Ubuntu系统配置

1 引言

本专题主要介绍了Ubuntu网络配置方法，日期和时间管理方法，使用ps命令查看进程信息，使用top和free命令查看CPU、内存等系统信息的方法。

2 网络配置

2.1 关于网络的基础知识（IP地址、子网掩码、网关、NAT等）

感兴趣可以学习《计算机网络》课程或者书籍，这里不再赘述。

2.2 查看Ubuntu网络配置

1、ifconfig命令（network interfaces configuring）

`ifconfig`：不加任何选项表示只显示接口为up状态的接口信息。

`Ifconfig -a`：表示显示所有接口的接口信息，不管该接口是up还是down都显示。这个用的多。

输入 `ifconfig` 命令，查看服务器的ip地址：

```
ifconfig -a
```

如果有报错，先下载安装 `net-tools`，然后再运行 `ifconfig` 命令即可看到：

```
sudo apt install net-tools
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.64.130 netmask 255.255.255.0 broadcast 192.168.64.255
    inet6 fe80::20c:29ff:feab:b83d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:b8:3d txqueuelen 1000 (Ethernet)
    RX packets 344 bytes 353322 (353.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 206 bytes 27129 (27.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 160 bytes 16336 (16.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 160 bytes 16336 (16.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

从上面可以看到，我这台服务器的网络配置为：

IPv4:

ip地址 (inet) : 192.168.64.130 ， 这是一个保留给内部使用的C类地址。

子网掩码 (netmask) : 255.255.255.0

局域网广播地址 (broadcast) : 192.168.64.255

IPv6:

ip地址 (inet6) : fe80::20c:29ff:feab:b83d

物理地址 (ether) : 00:0c:29:ab:b8:3d

"注意"

ether是指ethernet，即以太网，这是我们现在使用的LAN（local area network，局域网）技术。现在局域网都是基于以太网技术。

2、ping命令

使用ping命令，可以和网络上的其他设备打招呼，看彼此之间是否处于连通状态，例如，我想测试一下当前主机和网关是否连通，输入下列命令：

```
ping 192.168.64.130
```

ping之后会一直收到对方主机的回复，可以使用 **组合键Ctrl+Z** 将Ping命令挂起，或者按 **组合键Ctrl+C** 中止命令执行。

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ping 192.168.64.130
PING 192.168.64.130 (192.168.64.130) 56(84) bytes of data.
64 bytes from 192.168.64.130: icmp_seq=1 ttl=64 time=0.337 ms
64 bytes from 192.168.64.130: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 192.168.64.130: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 192.168.64.130: icmp_seq=4 ttl=64 time=0.060 ms
64 bytes from 192.168.64.130: icmp_seq=5 ttl=64 time=0.052 ms
64 bytes from 192.168.64.130: icmp_seq=6 ttl=64 time=0.048 ms
64 bytes from 192.168.64.130: icmp_seq=7 ttl=64 time=0.065 ms
64 bytes from 192.168.64.130: icmp_seq=8 ttl=64 time=0.128 ms
64 bytes from 192.168.64.130: icmp_seq=9 ttl=64 time=0.073 ms
64 bytes from 192.168.64.130: icmp_seq=10 ttl=64 time=0.226 ms
64 bytes from 192.168.64.130: icmp_seq=11 ttl=64 time=0.091 ms
^C
--- 192.168.64.130 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10248ms
rtt min/avg/max/mdev = 0.048/0.112/0.337/0.085 ms
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

??? tip "ping命令说明"

ping 命令用于测试与目标主机之间的连接。它向目标主机发送一个ICMP（Internet Control Message Protocol）Internet控制报文协议回显请求，并等待它的回应。如果目标主机正常工作并且连接畅通，则会收到一个回显响应。如果没有收到响应，则说明有问题，需要进一步排除故障。ping可以用来测试本机与目标主机是否联通、联通速度如何、稳定性如何。

使用ping命令作用：

- **检测网络连接问题**：如果我们无法访问某个网站或主机，可以使用ping命令来检测网络连接是否正常。如果ping命令能够成功收到目标主机的回复，说明网络连接正常；如果无法收到回复，可能是网络故障或目标主机不可达。
- **测量网络延迟**：通过ping命令可以测量网络延迟，即从发送ICMP请求到接收到回复所需的时间。可以使用ping命令的-c选项指定发送的ICMP请求次数，然后计算平均延迟时间。较高的延迟可能会导致网络连接缓慢，需要进一步排查网络问题。
- **检测丢包率**：ping命令还可以用于检测网络丢包率，即发送的ICMP请求在传输过程中丢失的比例。可以观察ping命令的输出结果中的丢包率字段，如果丢包率较高，可能是网络拥堵或目标主机负载过高。

3、ip命令

Linux ip 命令与 ifconfig 命令类似，但比 ifconfig 命令更加强大，主要功能是用于显示或设置网络设备。

ip 命令是 Linux 加强版的网络配置工具，用于代替 ifconfig 命令。

其语法结构如下：

```
ip [ OPTIONS ] OBJECT { COMMAND | help }
```

OBJECT 为常用对象，值可以是以下几种：

```
OBJECT={ link | addr | addrlabel | route | rule | neigh | ntable | tunnel | maddr | mroute | mrule
```

常用对象的取值含义如下：

- link：网络设备
- addr (address)：设备上的协议（IP或IPv6）地址
- addrlabel：协议地址选择的标签配置
- route：路由表条目
- rule：路由策略数据库中的规则

OPTIONS 为常用选项，值可以是以下几种：

```
OPTIONS={ -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] | -h[uman-readable] | -iec | -f[ami
```

常用选项的取值含义如下：

- V：显示命令的版本信息；
- s：输出更详细的信息；
- f：强制使用指定的协议族；
- 4：指定使用的网络层协议是IPv4协议；
- 6：指定使用的网络层协议是IPv6协议；
- 0：输出信息每条记录输出一行，即使内容较多也不换行显示；
- r：显示主机时，不使用IP地址，而使用主机的域名。

输入以下命令，显示网络接口信息

```
ip link show
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:ab:b8:3d brd ff:ff:ff:ff:ff:ff
    altname enp2s1
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

输入下面命令，显示网卡IP信息

```
ip addr show
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ab:b8:3d brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.64.10/24 brd 192.168.64.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet 192.168.64.130/24 brd 192.168.64.255 scope global secondary dynamic noprefixroute ens33
        valid_lft 1217sec preferred_lft 1217sec
    inet6 fe80::20c:29ff:feab:b83d/64 scope link
        valid_lft forever preferred_lft forever
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

输入下面的命令，显示系统路由

```
ip route show
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ip route list
default via 192.168.64.2 dev ens33 proto static metric 100
default via 192.168.64.2 dev ens33 proto dhcp src 192.168.64.130 metric 100
192.168.64.0/24 dev ens33 proto kernel scope link src 192.168.64.10 metric 100
192.168.64.0/24 dev ens33 proto kernel scope link src 192.168.64.130 metric 100
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

4、Windows查看网络配置

在Windows系统中，在命令提示符模式下查看网络配置的命令是

```
ipconfig -all
```

以下是我的电脑的部分网络配置截图：

```
C:\Users\lenovo>ipconfig -all

Windows IP 配置

    主机名 . . . . . : YCC
    主 DNS 后缀 . . . . . :
    节点类型 . . . . . : 混合
    IP 路由已启用 . . . . . : 否
    WINS 代理已启用 . . . . . : 否

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :
    描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
    物理地址 . . . . . : 10-A5-1D-96-44-3A
    DHCP 已启用 . . . . . : 是
    自动配置已启用 . . . . . : 是

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :
    描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
    物理地址 . . . . . : 12-A5-1D-96-44-39
    DHCP 已启用 . . . . . : 否
    自动配置已启用 . . . . . : 是

以太网适配器 VMware Network Adapter VMnet1:

    连接特定的 DNS 后缀 . . . . . :
    描述 . . . . . : VMware Virtual Ethernet Adapter for VMnet1
    物理地址 . . . . . : 00-50-56-C0-00-01
    DHCP 已启用 . . . . . : 否
    自动配置已启用 . . . . . : 是
    本地链接 IPv6 地址 . . . . . : fe80::78c6:b3a9:26af:d3b8%17(首选)
    IPv4 地址 . . . . . : 192.168.174.1(首选)
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :
    DHCPv6 IAID . . . . . : 184569942
    DHCPv6 客户端 DUID . . . . . : 00-01-00-01-2D-39-85-D1-10-A5-1D-96-44-39
    TCP/IP 上的 NetBIOS . . . . . : 已启用

以太网适配器 VMware Network Adapter VMnet8:

    连接特定的 DNS 后缀 . . . . . :
    描述 . . . . . : VMware Virtual Ethernet Adapter for VMnet8
    物理地址 . . . . . : 00-50-56-C0-00-08
```

电脑上因为有无线网卡、蓝牙、wmware虚拟机的虚拟网卡等设备，显示信息可能会比上面的截图长很多！

Windows系统同样可以使用ping命令。

```
C:\Users\lenovo>ping 192.168.64.130
```

```
正在 Ping 192.168.64.130 具有 32 字节的数据:  
来自 192.168.64.130 的回复: 字节=32 时间<1ms TTL=64  
来自 192.168.64.130 的回复: 字节=32 时间=9ms TTL=64  
来自 192.168.64.130 的回复: 字节=32 时间=21ms TTL=64  
来自 192.168.64.130 的回复: 字节=32 时间<1ms TTL=64
```

```
192.168.64.130 的 Ping 统计信息:  
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
    最短 = 0ms, 最长 = 21ms, 平均 = 7ms
```

2.3 修改Ubuntu网络配置

注意：Ubuntu的不同版本网络配置方法差别很大，这里的网络配置方法仅限于ubuntu-24.04版本，其他版本可能会有区别。

我们来给虚拟机设置静态IP地址。在以前使用虚拟机Vmware安装Ubuntu时，网络类型我们设置的是“NAT”模式。

新建虚拟机向导

网络类型
要添加哪类网络?

网络连接

☐ 使用桥接网络(R)
为客户机操作系统提供直接访问外部以太网网络的权限。客户机在外部网络上必须有自己的 IP 地址。

☒ 使用网络地址转换(NAT)(E)
为客户机操作系统提供使用主机 IP 地址访问主机拨号连接或外部以太网网络连接的权限。

☐ 使用仅主机模式网络(H)
将客户机操作系统连接到主机上的专用虚拟网络。

☐ 不使用网络连接(T)

帮助

< 上一步(B)

下一步(N) >

取消

我们来看一下Vmware给Ubuntu分配的IP网段是什么。

打开VMware：编辑 -> 虚拟网络编辑器 -> 更改设置，记住子网ip和子网掩码，我这里子网ip是192.168.64.0，C类网络IP地址0是指网络号，子网掩码是255.255.255.0，表示网络号为24位，这表示NAT子网IP地址可以在192.168.64.1~192.168.64.255中分配。

虚拟网络编辑器

×

名称	类型	外部连接	主机连接	DHCP	子网地址
VMnet1	仅主机...	-	已连接	已启用	192.168.174.0
VMnet8	NAT	NAT	已连接	已启用	192.168.64.0

添加网络(E)...

移除网络(O)

重命名网络(W)...

VMnet 信息

桥接模式(将虚拟机直接连接到外部网络)(B)

已桥接至(G):

自动设置(U)...

NAT 模式(与虚拟机共享主机的 IP 地址)(N)

NAT 设置(S)...

仅主机模式(在专用网络内连接虚拟机)(H)

将主机虚拟适配器连接到此网络(V)

主机虚拟适配器名称: VMware 网络适配器 VMnet8

使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)

DHCP 设置(P)...

子网 IP (I):

192 . 168 . 64 . 0

子网掩码(M):

255 . 255 . 255 . 0

⚠ 需要具备管理员特权才能修改网络配置。

更改设置(C)

还原默认设置(R)

导入(T)...

导出(X)...

确定

取消

应用(A)

帮助

如果需要修改，可以选择“更改设置”。

点击NAT设置，记住网关地址。

正常情况下VMware的网关IP是以 192.168.x.2结尾的，因为 x.1是绑定在物理机的虚拟网卡上的，而 x.2是用于转发数据的。

因此我这里网关IP地址是192.168.64.2。

网络: vmnet8
子网 IP: 192.168.64.0
子网掩码: 255.255.255.0
网关 IP(G): 192 . 168 . 64 . 2

端口转发(F)

主机端口	类型	虚拟机 IP 地址	描述
------	----	-----------	----

添加(A)... 移除(R) 属性(P)

高级

☒ 允许活动的 FTP(T)
☒ 允许任何组织唯一标识符(O)

UDP 超时(以秒为单位)(U): 30

配置端口(C): 0

☐ 启用 IPv6(E)
IPv6 前缀(6): fd15:4ba5:5a2b:1008::/64

DNS 设置(D)... NetBIOS 设置(N)... 确定 取消 帮助

在ubuntu中执行ifconfig，记住网卡名称，我的是ens33。

```

caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.64.130 netmask 255.255.255.0 broadcast 192.168.64.255
    inet6 fe80::20c:29ff:feab:b83d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:b8:3d txqueuelen 1000 (Ethernet)
    RX packets 604 bytes 422717 (422.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 402 bytes 60483 (60.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 189 bytes 19119 (19.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 189 bytes 19119 (19.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

接下来给Ubuntu配置静态IP地址

注意：Ubuntu18固定IP的方式跟Ubuntu18之前版本的配置方式不同，Ubuntu18之前在 `/etc/network/interfaces` 进行配置，Ubuntu18及之后版本在 `/etc/netplan/*.yaml` 进行配置，如 `/etc/netplan/01-network-manager-all.yaml`，如果 `/etc/netplan` 目录下没有 `yaml` 文件，则可以新建一个。

可以看到当前目录下没有该文件，默认有一个 `50-cloud-init.yaml`：

```

caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ cd /etc/netplan
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$ ls -l
total 4
-rw----- 1 root root 390 Sep 24 19:44 50-cloud-init.yaml
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$

```

这里使用Chat基础介绍一下该文件以及其文件相关命名规则：

在 Ubuntu 系统中，`/etc/netplan/50-cloud-init.yaml` 是 Netplan 网络配置工具的一个 YAML 文件，用于定义网络接口的配置。它通常是系统初始安装或配置时由 Cloud-Init 工具生成的，特别是在云环境中部署 Ubuntu 系统时。

`/etc/netplan/50-cloud-init.yaml` 是 Netplan 的配置文件之一，Netplan 用于管理网络接口，抽象出了一种简单的 YAML 配置方式，可以生成适配不同网络后端（如 NetworkManager 或 systemd-networkd）的配置。

数字优先级： `50-cloud-init.yaml` 文件名中的数字 50 表示优先级，Netplan 会按数字顺序加载多个配置文件。如果有多个文件，数字较小的配置文件会被较早解析，后续配置文件可能覆盖前

面的设置。

该文件内容通常类似如下：

```
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens33:
      dhcp4: true
  version: 2
~
~
~
~
~
~
~
1,1 All
```

- `network`：Netplan 配置的顶级键。
- `version: 2`：Netplan 配置的版本号，2 是当前常用版本。
- `ethernets`：定义以太网接口的设置。
- `ens33`：网络接口的名称（如 ens33、eth0 等）。
- `dhcp4: true`：启用 IPv4 的 DHCP。

使用下面命令新建并进行修改：

```
sudo vim /etc/netplan/01-network-manager-all.yaml
```

修改成以下形式：

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    ens33:
      dhcp4: no
      dhcp6: no
      addresses:
        - 192.168.64.10/24
      routes:
        - to : default
          via : 192.168.64.2
      nameservers:
        addresses:
          - 192.168.64.2
```

```
❏ Let Network Manager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    ens33:
      dhcp4: no
      dhcp6: no
      addresses:
        - 192.168.64.10/24
      routes:
        - to: default
          via: 192.168.64.2
      nameservers:
        addresses:
          - 192.168.64.2
~
~
~
~
~
"01-network-manager-all.yaml" [readonly] 16L, 332B      1,1      All
```

修改完成保存退出，执行

```
sudo netplan apply
```

出现警告，提醒权限问题，这是文件权限开的太高所致：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$ sudo netplan apply

** (generate:4194): WARNING **: 10:43:39.799: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:4193): WARNING **: 10:43:40.698: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:4193): WARNING **: 10:43:41.003: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$
```

执行

```
sudo chmod 0600 /etc/netplan/01-network-manager-all.yaml
```

再重新执行：

```
sudo netplan apply
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$ sudo chmod 0600 01-network-manager-all.yaml
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$ sudo netplan apply
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$
```

成功！可以使用 `ifconfig` 查看新的ip配置：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/netplan$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.64.10 netmask 255.255.255.0 broadcast 192.168.64.255
    inet6 fe80::20c:29ff:feab:b83d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:b8:3d txqueuelen 1000 (Ethernet)
    RX packets 3147 bytes 2060820 (2.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1610 bytes 191847 (191.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 214 bytes 22108 (22.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 214 bytes 22108 (22.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

可以看到ip地址已经变成了192.168.64.10了。

3 日期和时间管理

3.1 显示当前日期、时间和时区

输入date命令，显示当前日期、时间和时区

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date
Tue Nov 26 10:50:50 AM CST 2024
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$
```

其中，CST(China Standard Time)表示中国标准时间的时区缩写。

3.2 设定日期和时间格式

通过使用+和格式字符串，可以高度自定义时间的显示格式。格式字符串中的指令以%开头，后跟一个字符标识符。常见的格式化选项如下：

- %Y：年 (Year)，四位数的年份，即YYYY格式，如 2024
- %y：年 (year)，两位数的年份，即YY格式，如 24
- %m：月 (month)，两位数的月份，即MM格式，如 09
- %d：日 (day)，两位数的日期，即DD格式，如 29
- %H：小时 (Hour)，24小时制，即HH格式，如 14
- %M：分钟 (Minute)，分钟，即MM格式，如 35
- %S：秒 (Second)，即SS格式，如 02

输入带格式的date命令，显示日期：

```
date +%Y-%m-%d
date +%y-%m-%d
```

可以简化使用下列命令，其中 %F代表 YYYY-MM-DD date format 。

```
date +%F
```

输入带格式的date命令，显示时间：

```
date +%H:%M:%S
```

可以简化使用下列命令，其中 %T代表 HH:MM:SS time format。

```
date +%T
```

输入带格式的date命令，显示日期+时间，注意必须不要忘记命令中的引号：

```
date +"%Y-%m-%d %H:%M:%S"
```

可以简化使用下列命令：

```
date +"%F %T"
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +%Y-%m-%d
2024-11-26
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +%y-%m-%d
24-11-26
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +%F
2024-11-26
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +%H:%M:%S
10:57:10
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +%T
10:57:19
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +"%Y-%m-%d %H:%M:%S"
2024-11-26 10:57:44
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ date +"%F %T"
2024-11-26 10:58:06
```

3.3 把日期和时间写入文件

我们把日期和时间写入文件，输入下列命令

```
echo date +"%F %T" >> log.txt
ls -l
cat log.txt
```



```

caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ echo date +"%F %T" >> log.txt
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ls -l
total 28
drwxrwxr-x 2 caicloudcat caicloudcat 4096 Nov 13 21:23 demo
-rw-r--r-- 1 caicloudcat caicloudcat 0 Nov 19 10:58 example.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 31 Nov 19 10:30 hardLink.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 12 Nov 26 11:03 log.txt
lrwxrwxrwx 1 caicloudcat caicloudcat 11 Nov 19 09:59 symbolicLink.txt -> example.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 253 Oct 29 11:05 test.txt
drwxrwxrwx 3 caicloudcat caicloudcat 4096 Nov 19 14:00 web
-rw-rw-r-- 1 caicloudcat caicloudcat 53 Oct 22 10:51 web1.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 47 Oct 22 10:52 web2.txt
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ cat log.txt
date +%F %T
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$

```

可以看到在 `log.txt` 中保存是命令本身，而不是date的执行结果。

可以使用反引号` 和 `()`表示执行命令或表达式的值，其中`()不仅可以引用命令的输出，还可以引用变量的值、命令的参数等。

```
echo `date +"%F %T"` >> log.txt
```

推荐使用下面的命令：

```
echo $(date +"%F %T") >> log.txt
```

```

-rw-rw-r-- 1 caicloudcat caicloudcat 12 Nov 26 11:03 log.txt
lrwxrwxrwx 1 caicloudcat caicloudcat 11 Nov 19 09:59 symbolicLink.txt -> example.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 253 Oct 29 11:05 test.txt
drwxrwxrwx 3 caicloudcat caicloudcat 4096 Nov 19 14:00 web
-rw-rw-r-- 1 caicloudcat caicloudcat 53 Oct 22 10:51 web1.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 47 Oct 22 10:52 web2.txt
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ cat log.txt
date +%F %T
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ echo $(date +"%F %T") >> log.txt
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ls -l
total 28
drwxrwxr-x 2 caicloudcat caicloudcat 4096 Nov 13 21:23 demo
-rw-r--r-- 1 caicloudcat caicloudcat 0 Nov 19 10:58 example.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 31 Nov 19 10:30 hardLink.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 32 Nov 26 11:07 log.txt
lrwxrwxrwx 1 caicloudcat caicloudcat 11 Nov 19 09:59 symbolicLink.txt -> example.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 253 Oct 29 11:05 test.txt
drwxrwxrwx 3 caicloudcat caicloudcat 4096 Nov 19 14:00 web
-rw-rw-r-- 1 caicloudcat caicloudcat 53 Oct 22 10:51 web1.txt
-rw-rw-r-- 1 caicloudcat caicloudcat 47 Oct 22 10:52 web2.txt
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$

```

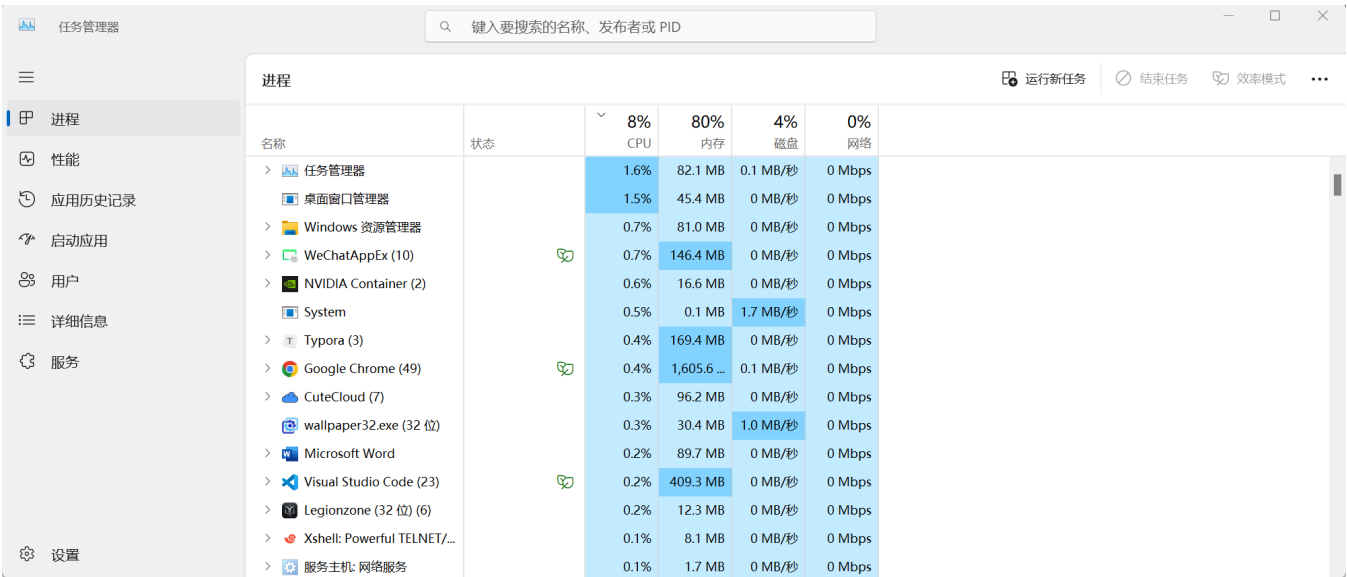
可以看到当前时间被保存到 `log.txt` 文件中了。

4 查看进程信息

4.1 显示进程信息

进程是指在计算机中正在运行的程序。

在windows系统中，在任务栏的空白处点右键，选择"任务管理器--进程"，可以查看系统的进程信息。



名称	状态	8% CPU	80% 内存	4% 磁盘	0% 网络
任务管理器		1.6%	82.1 MB	0.1 MB/秒	0 Mbps
桌面窗口管理器		1.5%	45.4 MB	0 MB/秒	0 Mbps
Windows 资源管理器		0.7%	81.0 MB	0 MB/秒	0 Mbps
WeChatAppEx (10)		0.7%	146.4 MB	0 MB/秒	0 Mbps
NVIDIA Container (2)		0.6%	16.6 MB	0 MB/秒	0 Mbps
System		0.5%	0.1 MB	1.7 MB/秒	0 Mbps
Typora (3)		0.4%	169.4 MB	0 MB/秒	0 Mbps
Google Chrome (49)		0.4%	1,605.6 ...	0.1 MB/秒	0 Mbps
CuteCloud (7)		0.3%	96.2 MB	0 MB/秒	0 Mbps
wallpaper32.exe (32 位)		0.3%	30.4 MB	1.0 MB/秒	0 Mbps
Microsoft Word		0.2%	89.7 MB	0 MB/秒	0 Mbps
Visual Studio Code (23)		0.2%	409.3 MB	0 MB/秒	0 Mbps
Legionzone (32 位) (6)		0.2%	12.3 MB	0 MB/秒	0 Mbps
Xshell: Powerful TELNET/...		0.1%	8.1 MB	0 MB/秒	0 Mbps
服务主机: 网络服务		0.1%	1.7 MB	0 MB/秒	0 Mbps

Ubuntu的ps（Process Status）命令可以显示当前终端会话中运行的进程信息。输入下列命令，显示当前终端会话中运行的进程的基本信息：

```
ps

caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps
  PID TTY          TIME CMD
 2980 pts/0    00:00:00 bash
 5054 pts/0    00:00:00 ps
```

输入下列命令，显示当前终端会话中运行的进程的详细信息，“-f”参数代表全格式/完整格式全格式/完整格式（full format）：

```
ps -f
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
caiclou+ 2980    2977  0  10:07 pts/0        00:00:00 -bash
caiclou+ 5056    2980  99  11:19 pts/0        00:00:00 ps -f
```

其中,

- UID (User ID) : 进程用户ID
- PID (Process ID): 进程ID
- PPID (Parent Process ID) : 父进程ID
- C(CPU) : 进程占用 CPU 的百分比
- STIME (Start Time) : 进程启动的具体时间
- TTY (Teletypewriter) : 与进程交互的终端设备
- TIME: 启动进程花费的 CPU 时间
- CMD: 启动进程的命令

输入下列命令, 显示系统中所有正在运行的进程的详细信息, 包括系统进程和用户进程, “-e”参数代表every:

```
ps -ef
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps -ef
UID      PID     PPID  C  STIME TTY          TIME CMD
root         1         0  0  10:06 ?           00:00:07 /sbin/init splash
root         2         0  0  10:06 ?           00:00:00 [kthreadd]
root         3         2  0  10:06 ?           00:00:00 [pool_workqueue_release]
root         4         2  0  10:06 ?           00:00:00 [kworker/R-rcu_g]
root         5         2  0  10:06 ?           00:00:00 [kworker/R-rcu_p]
root         6         2  0  10:06 ?           00:00:00 [kworker/R-slub_]
root         7         2  0  10:06 ?           00:00:00 [kworker/R-netns]
root         9         2  0  10:06 ?           00:00:00 [kworker/0:0H-events_highpri]
root        11         2  0  10:06 ?           00:00:00 [kworker/u256:0-floppy]
root        12         2  0  10:06 ?           00:00:00 [kworker/R-mm_pe]
root        13         2  0  10:06 ?           00:00:00 [rcu_tasks_kthread]
root        14         2  0  10:06 ?           00:00:00 [rcu_tasks_rude_kthread]
root        15         2  0  10:06 ?           00:00:00 [rcu_tasks_trace_kthread]
root        16         2  0  10:06 ?           00:00:00 [ksoftirqd/0]
root        17         2  0  10:06 ?           00:00:01 [rcu_preempt]
root        18         2  0  10:06 ?           00:00:00 [migration/0]
root        19         2  0  10:06 ?           00:00:00 [idle_inject/0]
root        20         2  0  10:06 ?           00:00:00 [cpuhp/0]
root        21         2  0  10:06 ?           00:00:00 [cpuhp/1]
```

注意: 这个页面很长, 这里没有完全截屏下来。

如果你觉得显示的进程数量太多, 可以使用管道符号缩小查找范围。| 是管道符号, 表示 将前一个命令的 "输出" 作为后一个命令的 "输入"。这样就可以一步一步缩小范围了。grep ps 会在

`ps -ef` 命令的 "输出" 中查找包含 `ps` 的进程信息。

```
ps -ef | grep ps
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps -ef | grep ps
root      511      2  0 10:06 ?        00:00:00 [psimon]
root     1077      1  0 10:06 ?        00:00:00 /usr/sbin/cupsd -l
root     1090      1  0 10:06 ?        00:00:00 sshd: /usr/sbin/sshd -D [listener] 0 of
10-100 startups
cups-br+  1116      1  0 10:06 ?        00:00:00 /usr/sbin/cups-browsed
kernoops  1123      1  0 10:06 ?        00:00:00 /usr/sbin/kerneloops --test
kernoops  1128      1  0 10:06 ?        00:00:00 /usr/sbin/kerneloops
root     4909      2  0 10:47 ?        00:00:00 [psimon]
caiclou+  5063    2980  0 11:20 pts/0    00:00:00 ps -ef
caiclou+  5064    2980  0 11:20 pts/0    00:00:00 grep --color=auto ps
```

输入下列命令，显示当前终端会话中当前用户正在运行的进程信息，“-u”参数代表user：

```
ps -u
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
caiclou+  2022  0.0  0.1 244336  6016 tty2      Ssl+  10:07   0:00 /usr/libexec/gdm-wayland
caiclou+  2032  0.0  0.4 306876 16512 tty2      Sl+   10:07   0:00 /usr/libexec/gnome-sessi
caiclou+  2980  0.0  0.1  19740  5120 pts/0     Ss   10:07   0:00 -bash
caiclou+  5065  300  0.1 22284  4608 pts/0     R+   11:23   0:00 ps -u
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$
```

其中，

- USER：用户名
- PID (Process ID)：进程ID
- CPU：CPU使用率
- MEM：(Memory) 内存使用率
- VSZ：(Virtual Size) 虚拟内存大小(以KB为单位)
- RSS：(Resident Set Size) 常驻内存大小(以KB为单位)
- TTY：(Teletypewriter) 与进程交互的终端设备
- STAT：(State) 进程状态
 - **R** 正在运行状态（正在使用CPU）
 - **S** 睡眠状态
 - **D** 不可中断睡眠，即在睡眠的过程中不可以接收信号唤醒=》执行的IO操作得不到硬件设备的响应
 - **T** 停止的进程

- **Z** 僵尸进程
 - **X** 死掉的进程(几乎看不见, 因为死了就立即回收了)
 - **<** 代表优先级较高的进程
 - **N** 代表优先级较低的进程
 - **s** 包含子进程
 - **+** 表示在前台运行的进程 (前台运行就是指在命令行中运行)
 - **l** 小写字母l, 代表以线程的方式运行, 即多线程
 - **|** 管道符号代表多进程
- **START**: 进程启动时间
 - **TIME**: 启动进程花费的 CPU 时间
 - **COMMAND**: 启动进程的命令

ps 命令中:

- **-a** 显示一个终端的所有进程
- **-u** 选择有效的用户id或者是用户名
- **-x** 显示没有控制终端的进程, 同时显示各个命令的具体路径。

aux参数很多情况下可以一起使用, 输入下列命令:

```
ps -aux
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.1  0.3  23248 14136 ?        Ss   10:06   0:07 /sbin/init splash
root         2  0.0  0.0      0      0 ?        S    10:06   0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        S    10:06   0:00 [pool_workqueue_release
root         4  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/R-rcu_p]
root         6  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/R-slub_]
root         7  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/R-netns]
root         9  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/0:0H-events_hi
root        11  0.0  0.0      0      0 ?        I    10:06   0:00 [kworker/u256:0-floppy]
root        12  0.0  0.0      0      0 ?        I<   10:06   0:00 [kworker/R-mm_pe]
root        13  0.0  0.0      0      0 ?        I    10:06   0:00 [rcu_tasks_kthread]
root        14  0.0  0.0      0      0 ?        I    10:06   0:00 [rcu_tasks_rude_kthread
root        15  0.0  0.0      0      0 ?        I    10:06   0:00 [rcu_tasks_trace_kthrea
root        16  0.0  0.0      0      0 ?        S    10:06   0:00 [ksoftirqd/0]
root        17  0.0  0.0      0      0 ?        I    10:06   0:01 [rcu_preempt]
root        18  0.0  0.0      0      0 ?        S    10:06   0:00 [migration/0]
root        19  0.0  0.0      0      0 ?        S    10:06   0:00 [idle_inject/0]
root        20  0.0  0.0      0      0 ?        S    10:06   0:00 [cpuhp/0]
root        21  0.0  0.0      0      0 ?        S    10:06   0:00 [cpuhp/1]
root        22  0.0  0.0      0      0 ?        S    10:06   0:00 [idle_inject/1]
root        23  0.0  0.0      0      0 ?        S    10:06   0:00 [migration/1]
```



"-ef 和 -aux 之间的区别："

- `-ef` 会打印出父级PID
- `-aux` 会打印出CPU、内存相关的使用情况

4.2 强制结束进程

强制结束 id=4857 的进程

```
kill -9 4857
```

-9 表示发送 SIGKILL[signal kill] 信号给进程 id 为 4857 的进程，SIGKILL 信号是一种强制停止进程的信号

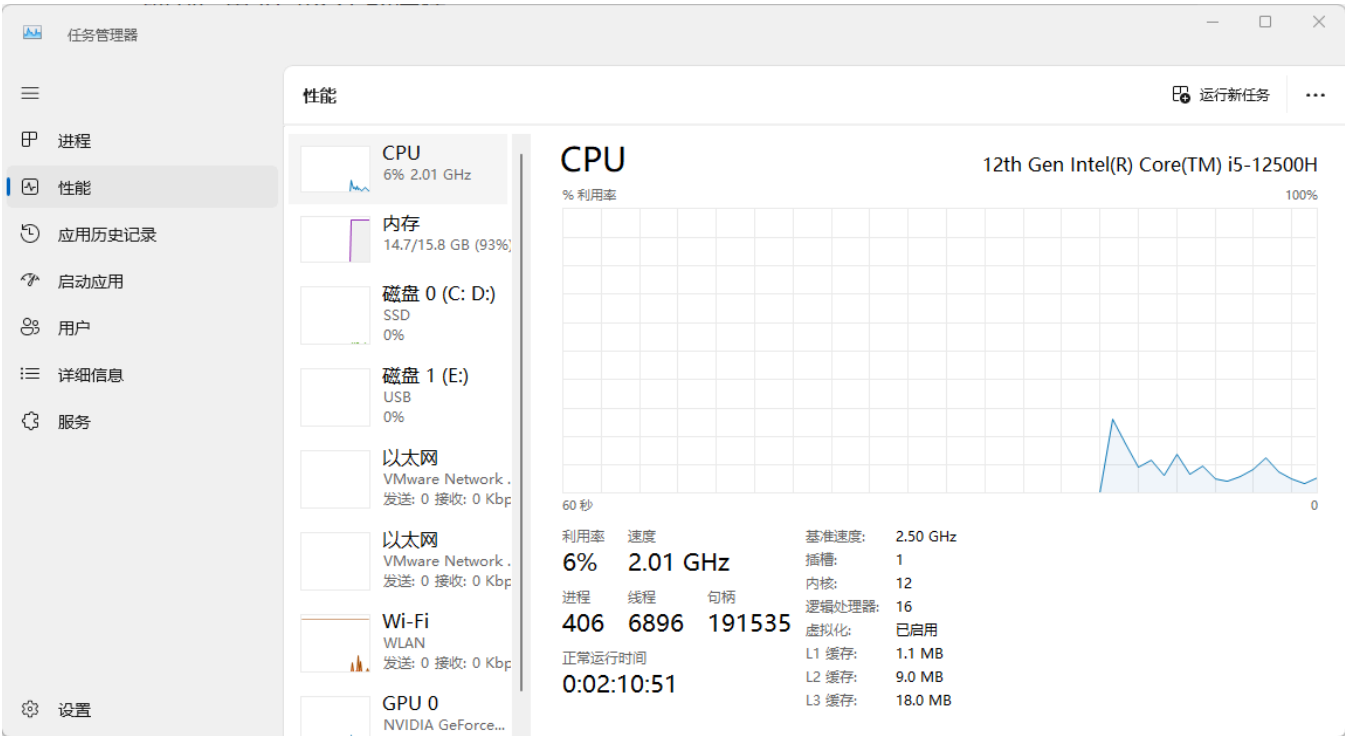
```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ kill -9 4857
-bash: kill: (4857) - No such process
```

我的系统中没有4857的进程，所以提示No such process。但是 kill 命令要记住，以后如果有哪个进程运行出现问题，就可以直接kill掉。

5 查看cpu、内存等系统信息

5.1 使用top命令查看系统的实时信息

在windows系统中，在任务栏的空白处点右键，选择“任务管理器”--“性能”，可以查看系统CPU、内存、磁盘等设备的实时信息。



在Ubuntu中，top是一个非常实用的动态实时显示系统运行状态的命令行工具。它能够提供系统概览，包括正在运行的进程信息、CPU 使用率、内存使用情况等。输入下列命令：

```
top
```

```
top - 11:44:33 up 1:37, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 306 total, 1 running, 305 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.5 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3868.2 total, 1587.7 free, 1183.2 used, 1365.9 buff/cache
MiB Swap: 3769.0 total, 3769.0 free, 0.0 used. 2685.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 2171 caiclou+  20   0 4302204 261892 114300 S   1.0   6.6   0:37.43 gnome-shell
 5188 caiclou+  20   0  23204    5760   3584 R   1.0   0.1   0:00.22 top
   939 root      20   0     0     0     0 I   0.7   0.0   0:21.45 kworker/2:3-mpt_poll_0
 5164 root      20   0     0     0     0 I   0.7   0.0   0:01.99 kworker/3:2-events
   17 root      20   0     0     0     0 I   0.3   0.0   0:01.75 rcu_preempt
 4121 root      20   0     0     0     0 I   0.3   0.0   0:12.23 kworker/1:1-mm_percpu_wq
 4950 root      20   0     0     0     0 I   0.3   0.0   0:08.52 kworker/0:0-events
 5179 root      20   0     0     0     0 I   0.3   0.0   0:00.14 kworker/u257:2-events_unbound
    1 root      20   0  23248  14136   9400 S   0.0   0.4   0:07.27 systemd
    2 root      20   0     0     0     0 S   0.0   0.0   0:00.03 kthreadd
    3 root      20   0     0     0     0 S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/R-netns
    9 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   11 root      20   0     0     0     0 I   0.0   0.0   0:00.00 kworker/u256:0-floppy
   12 root      0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
   13 root      20   0     0     0     0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   14 root      20   0     0     0     0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   15 root      20   0     0     0     0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   16 root      20   0     0     0     0 S   0.0   0.0   0:00.10 ksoftirqd/0
   18 root      rt   0     0     0     0 S   0.0   0.0   0:00.28 migration/0
```

上面显示信息的具体说明如下：

(1) 顶部区域

top - 11:44:33 up 1:37, 3 users, load average: 0.00, 0.00, 0.00

11:44:33 : 当前系统时间。

1:37 : 系统运行时间，格式是小时:分钟。

3 users : 当前登录的用户数量

load average: 0.00, 0.00, 0.00 : 系统负载平均值，分别是过去1分钟、5分钟和15分钟的平均值。

(2) 任务（进程）区域

Tasks: 306 total, 1 running, 305 sleeping, 0 stopped, 0 zombie

306 total : 系统中总的任务（进程）数。

1 running : 正在运行的任务数。

305 sleeping : 处于睡眠状态的任务数。

0 stopped : 被停止的任务数。

0 zombie : 僵尸进程数。僵尸进程是指在进程已经结束运行但其相关资源（如内存空间等）尚未被释放的情况下，该进程的状态是一个已经死亡但仍然存在于系统中的进程。

(3) CPU 状态

%Cpu(s): 0.2 us, 0.5 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

us : user, 用户空间占用CPU的百分比。

sy : system, 内核空间占用CPU的百分比。

ni : nice, 改变过优先级的进程占用CPU的百分比。nice 是指优先级调整值,用于调整进程的优先级，当一个进程被赋予较高的nice值时,它将以较低的优先级运行，从而使其他进程能够获得更多的 CPU 资源。

id : idle, 空闲CPU百分比。99.3 id表示CPU大部分时间处于空闲状态, 说明系统负载较低。

wa : wait, 等待输入输出 (I/O) 的CPU时间百分比。I/O[Input/Output] 操作是指系统中的输入和输出操作, 常见的 I/O 操作有: 读取和写入文件、键盘输入、打印机输出等。

hi : hardware interrupt, 处理硬件中断的时间百分比。**硬中断** 是由硬件设备(如键盘、鼠标等)发出的信号, 会打断 CPU 正在执行的任务,使其跳转到中断处进行处理。0.0 hi 表示没有硬中断发生。

si : software interrupt, 处理软件中断的时间百分比。**软中断** 是由软件程序发出的信号,会打断 CPU正在执行的任务, 使其跳转到中断处进行处理, 0.0 si 表示没有软中断发生。

st : 被虚拟机偷去的时间百分比。0.0 st 表示没有虚拟机从物理机中窃取CPU时间, 即虚拟化对 CPU 的占用为0.0%。

(4) 内存状态

MiB Mem : 3868.2 total, 1587.7 free, 1183.2 used, 1365.9 buff/ cache

MiB Mem (Megabytes of Memory) , 内存的兆字节数。

total: 总内存。

free: 空闲内存。

used: 已使用的内存。

buff/cache: 用作缓冲和缓存的内存。

(5) 交换空间 (Swap) 状态

MiB Swap: 3769.0 total, 3769.0 free, 0.0 used. 2685.0 avail Mem

MiB Swap (Megabytes of Swap) , 交换空间兆字节数。**交换空间** (Swap Space) 是系统中的虚拟内存技术,用于在物理内存不足时, 将部分内存数据暂时存储到磁盘上。

total: 总交换空间。

free: 空闲交换空间。

used: 已使用的交换空间。

avail Mem: 可用于启动新应用的内存量。

(6) 进程信息区

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

- PID: 进程编号
- USER: 用户名
- PR(Priority): 优先级
- NI(Nice): 优先级调整值,用于调整进程的优先级
- VIRT(Virtual Memory): 虚拟内存大小
- RES(Resident Memory): 物理内存大小
- SHR(Shared Memory): 共享内存大小
- %CPU(CPU Usage): CPU使用率
- %MEM(Memory Usage): 内存使用率
- TIME+(Time Elapsed): 进程使用的CPU时间
- COMMAND: 命令

在 top 命令的交互模式下:

按键盘上1键, 切换CPU显示模式, 显示每个CPU的使用情况

```
top - 14:22:33 up 1:54, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 308 total, 1 running, 307 sleeping, 0 stopped, 0 zombie
%Cpu0 :  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1 :  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2 :  0.7 us,  0.7 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 3868.2 total, 1603.6 free, 1158.5 used, 1375.6 buff/cache
MiB Swap: 3769.0 total, 3769.0 free,  0.0 used. 2709.7 avail Mem
```

按键盘上大写字母M, 按照物理内存(RES)使用率对进程进行排序。

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2171	caiclou+	20	0	4302204	261892	114300	S	0.3	6.6	0:42.76	gnome-shell
3141	caiclou+	30	10	757888	180604	101032	S	0.0	4.6	0:05.52	update-manager
2333	caiclou+	20	0	970340	63108	49540	S	0.0	1.6	0:00.51	evolution-alarm
2261	caiclou+	20	0	1271856	42880	34688	S	0.0	1.1	0:00.52	evolution-sourc
2766	caiclou+	20	0	713084	41292	31216	S	0.0	1.0	0:00.30	xdg-desktop-por
4149	root	20	0	585264	40912	34236	S	0.0	1.0	0:02.65	fwupd
3081	caiclou+	20	0	577384	32636	25212	S	0.0	0.8	0:01.21	update-notifier
795	root	20	0	1991632	32332	20224	S	0.0	0.8	0:02.63	snapd
2673	caiclou+	20	0	429612	30484	22016	S	0.0	0.8	0:00.42	snapd-desktop-i
2556	caiclou+	20	0	834092	29696	25856	S	0.0	0.7	0:00.26	evolution-addre
2477	caiclou+	20	0	429620	28296	17204	S	0.0	0.7	0:01.69	ibus-extension-
2726	caiclou+	39	19	890820	28256	18432	S	0.0	0.7	0:00.95	tracker-miner-f
2745	caiclou+	20	0	2671248	27052	22016	S	0.0	0.7	0:00.24	gjs
2263	caiclou+	20	0	2671288	26564	22016	S	0.0	0.7	0:00.13	gjs
2839	caiclou+	20	0	425872	24576	18048	S	0.0	0.6	0:00.23	xdg-desktop-por
2523	caiclou+	20	0	1374668	24192	20864	S	0.0	0.6	0:00.35	evolution-calen
2304	caiclou+	20	0	750216	24120	20664	S	0.0	0.6	0:00.23	gsd-media-keys
2472	caiclou+	20	0	555320	24064	20480	S	0.0	0.6	0:00.12	goa-daemon
2306	caiclou+	20	0	532164	23104	19264	S	0.0	0.6	0:00.28	gsd-power
1085	root	20	0	120904	22784	13568	S	0.0	0.6	0:00.14	unattended-upgr

按键盘上大写字母P, 按照CPU占用率对进程进行排序, 用来查看CPU占用率最高的进程。

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5307	caiclou+	20	0	23180	5888	3712	R	1.4	0.1	0:01.16	top
666	systemd+	20	0	17556	7552	6656	S	0.5	0.2	0:14.19	systemd-oomd
4121	root	20	0	0	0	0	I	0.5	0.0	0:16.51	kworker/1:1-events
5167	root	20	0	0	0	0	I	0.5	0.0	0:04.60	kworker/2:0-events
5197	root	20	0	0	0	0	I	0.5	0.0	0:01.86	kworker/0:2-events
5200	root	20	0	0	0	0	I	0.5	0.0	0:02.25	kworker/3:0-events
1	root	20	0	23248	14136	9400	S	0.0	0.4	0:07.46	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u256:0-floppy
12	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_pe
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
16	root	20	0	0	0	0	S	0.0	0.0	0:00.10	ksoftirqd/0
17	root	20	0	0	0	0	I	0.0	0.0	0:01.93	rcu_preempt

输入下列命令, 查看指定PID的进程信息:

```
top -p 1
```

```
top - 14:26:39 up 1:58, 3 users, load average: 0.00, 0.01, 0.00
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.0 us, 4.0 sy, 0.0 ni, 94.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3868.2 total, 1599.4 free, 1162.6 used, 1375.7 buff/cache
MiB Swap: 3769.0 total, 3769.0 free, 0.0 used. 2705.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	23248	14136	9400	S	0.0	0.4	0:07.48	systemd

5.2 使用free命令显示系统内存

如果只需要显示内存信息，可以使用free命令。free 命令是 Linux 和类 Unix 操作系统中用于显示内存使用情况的工具。通过 free 命令，可以查看系统中物理内存（RAM）、交换空间（swap）以及缓存和缓冲区的使用情况。输入下列命令显示内存的使用情况：

```
free
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ free
              total        used        free      shared  buff/cache   available
Mem:           3961004       1190688       1637584          26060       1408740       2770316
Swap:          3859452           0       3859452
```

这里默认以 KB 为单位显示内存使用情况，各字段的说明如下：

- total 总内存大小
- used 已使用的内存大小
- free 空闲内存大小
- shared 共享内存大小
- buff/cache 缓冲和缓存区内内存大小
- available 可用的内存大小 = 空闲内存 + 缓冲和缓存区内内存

在输入free命令时，常用选项如下：

- **-b**：以字节为单位显示内存使用情况。
- **-k**：以 KB 为单位显示内存使用情况（默认）。
- **-m**：以 MB 为单位显示内存使用情况。
- **-g**：以 GB 为单位显示内存使用情况。
- **-h**：以人类可读的格式显示内存使用情况（自动选择合适的单位，如 KB、MB、GB）。
- **-t**：显示总计行，即在输出的开头显示内存和交换空间的总计。
- **-s [秒]**：每隔指定的秒数重复显示内存使用情况。

- `-c` 计数：重复显示内存使用情况指定的次数。

例如，输入下列命令，以兆为单位显示内存的使用情况：

```
free -m
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Desktop$ free -m
```

	total	used	free	shared	buff/cache	available
Mem:	3868	1162	1598	25	1375	2705
Swap:	3768	0	3768			

上图中，虽然空闲内存为1598MB，缓冲和缓存区内存为 1375 MB，但实际可用的内存大小只有2705MB，这是因为还有其他一些内存区域被占用，不能立即用于新的进程。