Linux Beginner

From Matchi Wiki

Contents

Basics

How can you see which kernel version a system is currently is use?

Use `uname`:

Example:

```
$ uname -a
Linux z2 4.4.0-sabayon #1 SMP Thu Dec 22 13:47:06 UTC 2016 x86_64 Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz GenuineIntel GN
```

What is the system's current IP address?

Use `ifconfig` or `ip addr show`:

Examples:

```
$ ifconfig
eth0:   flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.23  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fdc2:4b54:3728:0:ce3d:82ff:fe20:8834  prefixlen 64  scopeid 0x0<global>
        inet6 2a02:c7f:7005:9300:ce3d:82ff:fe20:8834  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::ce3d:82ff:fe20:8834  prefixlen 64  scopeid 0x20<link>

$ ip addr show
```

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether cc:3d:82:20:88:34 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.23/24 brd 192.168.0.255 scope global dynamic wlp3s0
       valid_lft 49545sec preferred_lft 49545sec
    inet6 fdc2:4b54:3728:0:ce3d:82ff:fe20:8834/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
```

How to check for free disk space on a system?

Use `df`

Example:

```
$  df
Filesystem     1K-blocks      Used Available Use% Mounted on
none             8174884      1624   8173260   1% /run
udev               10240         0     10240   0% /dev
tmpfs            8174884    188676   7986208   3% /dev/shm
/dev/sda10     209135104 195447948   3040580  99% /
tmpfs            8174884         0   8174884   0% /sys/fs/cgroup
tmpfs            8174884       112   8174772   1% /tmp
/dev/sda9      235364160  43815328 179569928  20% /data
tmpfs            1634980        16   1634964   1% /run/user/1000
```

How does one check the total size of a given directory?

Use `du`

Example:

```
$ du -sh /data
81.5G /data
```

How can you see if a service is running?

On older system, use `</etc/init.d/[servicename] status/code>`:

`On Centos 7.x <code>systemctl status [service name]`:

Examples:

```
$ systemctl status mysqld
● mysqld.service - MySQL database server
   Loaded: loaded (/usr/lib64/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
```

Or, if is running:

```
$ systemctl status mysqld
● mysqld.service - MySQL database server
   Loaded: loaded (/usr/lib64/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2017-02-07 18:37:44 GMT; 2s ago
  Process: 14107 ExecStartPost=/usr/libexec/mysqld-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
 Main PID: 14106 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─14106 /usr/sbin/mysqld --basedir=/usr
```

How does one start or stop a service?

On Centos 7.x `sudo systemctl start [service name]`:

Examples:

```
$ sudo systemctl start mysqld
```

Why do we need to use `sudo` to manage services?

Only user *root* can start or stop services.

How does one check for open, listening ports on the currently logged-on system?

Use the `netstat` command:

Examples:

```
$ sudo netstat -tulpn
Password:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:32294           0.0.0.0:*               LISTEN      2639/skype
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      14106/mysqld
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2164/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      2071/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      2164/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      2071/cupsd
```

Why do we need to use `sudo` for the `netstat` command?

Services with ports below 1024 are owned by user *root* and the actual program and PID would normally not be displayed.

How does one check for open ports on a remote machine?

Use the `nmap` command:

Examples:

```
$ nmap [server IP address]

Starting Nmap 7.12 ( https://nmap.org ) at 2017-02-07 22:31 GMT
Nmap scan report for mapp02 (87.106.145.156)
Host is up (0.037s latency).
Not shown: 990 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
```

```
993/tcp  open  imaps
995/tcp  open  pop3s
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 4.80 seconds
```

How does one check for the resource consumption of a process?

Use the `ps` command and filter the output for the process name.

Examples:

```
$ ps aux | grep mysqld
mysql    14106  0.0  0.7 817528 126796 ?       Ssl  18:37   0:00 /usr/sbin/mysqld --basedir=/usr
```

This shows the process's owner (mysql), the process's PID, the relative and averaged CPU usage, the relative memory consumption (0.7% of the total memory on the machine) and virtual memory storage.

How does one check which process is consuming the most resources?

Use the `top` command or the more colourful `htop` command:

Examples:

```
$ top
top - 19:04:09 up 2 days,  1:31,  5 users,  load average: 1.12, 1.31, 1.37
Tasks: 250 total,   2 running, 248 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.9 us,  0.9 sy,  0.0 ni, 93.1 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 16349772 total,   889820 free,  3817408 used, 11642544 buff/cache
KiB Swap: 16694268 total, 16694268 free,        0 used. 11655036 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2572 gerrit    20   0 1904544 545732  96544 R  17.3  3.3  16:06.41 thunderbird
 2193 root      20   0  614060 150964  87520 S   3.7  0.9  56:00.14 X
 2630 gerrit    20   0  720372  66552  52668 S   2.7  0.4   2:18.49 konsole
 2639 gerrit    20   0  677052 222076  80660 S   2.0  1.4  65:21.47 skype
 2510 gerrit    20   0 3299704 133808  71640 S   1.3  0.8  15:07.34 kwin_x11
 4486 gerrit    20   0 1506684 457456  71996 S   1.0  2.8   8:29.45 chrome
11482 root      20   0 1301720  43828  13532 S   0.7  0.3   0:12.27 python2.7
32617 gerrit    20   0 1496572 451124  71096 S   0.7  2.8   9:01.80 chrome
  696 gerrit    20   0 1477808 379912 111044 S   0.3  2.3  14:32.37 chrome
14614 gerrit    20   0  140196   3860   3052 R   0.3  0.0   0:00.03 top
...
```

Processes are sorted by default in descending CPU usage, so the first-listed item in the list is worst resource consumer.

How can I mount a new storage device (USB drive / harddrive partition) on a machine?

Use the `mount` command, the `umount` command to unmount the device later.

Examples:

The canonical place where devices are mounted is on the /mnt directory. Devices, as well as well as other file directories, can be mounted anywhere else too. Physical devices are accessed via the /dev directory, but need to be mounted before they can be read or written to. Here, we mount the 4th partition (the 3 in sda3) of the first

SATA (the s in sda3) drive (the a in sda3) on the /mnt/data directory. The mount command does its best to guess what type of file system it is, and if it fails to detect it then it needs to be indicated with a `-t [FS-type]` added to the command:

```
$ sudo mkdir /mnt/data
$ sudo mount /dev/sda3 /mnt/data
```

Show all the mounted devices and directories on a system

Use the `mount` command:

Examples:

This lists all the mounted devices, logical volumes and directories.

```
$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
none on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=10240k,nr_inodes=2040656,mode=755)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
/dev/sda10 on / type ext4 (rw,noatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
...
```

How does one check for open file on a machine?

Use the `lsof` command:

This will list 1000's of open files, regardless of who has them open. It includes files opened by the kernel.

Examples:

Filter the output to only show files opened by the *skype* program, and then only the first 10:

```
$ lsof | grep "^skype" | head
skype      2639                gerrit  cwd       DIR               8,10      12288    8392112 /home/gerrit
skype      2639                gerrit  rtd       DIR               8,10       4096          2 /
skype      2639                gerrit  txt       REG               8,10   36499124    4758911 /opt/bin/skype
skype      2639                gerrit  mem       REG               0,16   67108904      13243 /dev/shm/pulse-shm-34976629
skype      2639                gerrit  DEL       REG                0,5                 15047 /memfd:pulseaudio
skype      2639                gerrit  mem       REG               8,10   16504512    3217997 /usr/share/fonts/noto-cjk/N
skype      2639                gerrit  DEL       REG                0,5              13762561 /SYSV00000000
skype      2639                gerrit  mem       REG               8,10    1336044    3016815 /usr/share/fonts/unifont/un
skype      2639                gerrit  mem       REG               8,10   12293172    3016810 /usr/share/fonts/unifont/un
skype      2639                gerrit  mem       REG               8,10   10385096    3043913 /usr/share/fonts/baekmuk-fo
```

What is the Butter File System?

It is a file system known as btrfs and based on the B-Tree and uses the Copy-On-Write principle to aid speed, pooling of multiple devices for scaling storage.

What are other commonly-encountered file system

- NTFS - Used on Windows machines
- ETX2, EXT3, EXT4
- ReiserFS - now largely defunct because the inventor killed his wife (she was a bitch, though) and is languishing in jail
- XFS
- Swap - not strictly a file system
- FAT32 - used on SD cards and very old Windows systems