

Task:

Linux 系统典型面试题与解决方案，如，如何分析日志文件 请求 ip 前十的 ip 等等 20-30 道场景.

Overview:

1. 获取一个进程的内存并监控
2. Linux 中如何打印所有网络接口
3. 有效电话号码 (leetcode)
4. 第十行 (leetcode)
5. 转置文件 (leetcode)
6. 词频 (leetcode)
7. shell 中 \${} 与 \$() 各是什么意思
8. 如何在 linux 中打印所有环境变量
9. 多服务器的系统时间不一致如何解决
10. 服务器的平均负载如何计算
11. 如何查看 node_modules(某一文件夹) 的体积有多大
12. Linux 文件系统的 mtime, ctime 有什么区别?
13. 如何查看一个文件的 inode number
14. 在 linux 中如何查看 cron 执行的情况
15. 如何检查可用磁盘空间?
16. 如何查看机器的内核版本?
17. 如何在 Linux 系统上启动服务?
18. 如何检查 Linux 系统的 IP?
19. 假设您有一个在 Linux 上运行的 Apache Web 服务器，在哪里可以找到该服务器的 index.html 文件?
20. 如何将文件的全部权限授予读取，写入和执行。
21. 统计 Nginx 日志里访问次数最多的前十个 IP
22. 把 /data 目录及其子目录下所有以扩展名 .txt 结尾的文件中包含 magedu 的字符串全部
23. 用 Shell 脚本或命令将/opt 目录下的大于 15KB 的文件都移到/tmp 目录下

1. 获取一个进程的内存并监控

Source: <https://github.com/shfshanyue/Daily-Question/issues/4>

在编写脚本时，有时会出现内存过大发生 (out of memory) OOM 的事情，那我们如何得知某个进程的内存？另外又如何监控它

ps = process status

通过 ps aux 列出所有进程及其状态和资源使用情况

a = show processes for all users

u = display the process's user/owner

x = also show processes not attached to a terminal

有以下三个命令

1. ps PID (ps -o rss PID)
2. pidstat -r
3. htop/top -pid PID

通过 ps 可以获知一个进程所占用的内存

ps -o rss PID

(VSZ: 虚拟内存集，进程占用的虚拟内存空间。

RSS: 物理内存集，进程占用实际物理内存空间。)

```
# 查看 2579 PID 的内存
# -o rss 代表附加 RSS 信息进行打印
$ ps -o rss 2579
PID    RSS S TTY      TIME COMMAND
2579 19876 S pts/10 00:00:03 node index.js
```

如果要监控内存，肯定使用对进程万能的命令 pidstat

pidstat -r -p PID

```
# 查看 23097 PID 的内存信息，每隔一秒打印一次
# -r: 查看进程的内存信息
# -s: 查看进程的 stack 信息
# -p: 指定 PID
# 1: 每间隔 1s 打印一次
# 2: 共打印 2 组
$ pidstat -sr -p 23097 1 2
Linux 3.10.0-693.2.2.el7.x86_64 (shanyue)      07/18/19      _x86_64_      (2 CPU)

18:56:07      UID      PID minflt/s  majflt/s     VSZ    RSS   %MEM StkSize  StkRef
Command
18:56:08        0    23097      0.00      0.00 366424 95996   2.47    136     80 node
18:56:08      UID      PID minflt/s  majflt/s     VSZ    RSS   %MEM StkSize  StkRef
Command
```

```
18:56:09      0    23097    0.00    0.00 366424 95996  2.47   136    80 node
```

htop/top -pid PID

```
 1 [||||]          ] Tasks: 169; 1 running
 2 [||||]          ] Load average: 2.70 1.87 1.00
Mem[|||||||||||||8.59G/15.5G] Uptime: 277 days(!), 02:02:01
Swp[               ]

START  PID USER  PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
17:20 11401 root    20   0 593M 48048 14136 S  0.0  0.3  0:14.57 node index.js shanyue
```

2. Linux 中如何打印所有网络接口

Source: <https://github.com/shfshanyue/Daily-Question/issues/31>

有三种方式可以实现

1. `ifconfig`
2. `netstat -i`
3. `ip`

`ifconfig` 是最简单最常用，但是打印信息太多了

```
$ ifconfig
cni0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
    inet 10.244.0.1  netmask 255.255.255.0  broadcast 0.0.0.0
    ether 9e:f3:9e:47:9d:55  txqueuelen 1000  (Ethernet)
    RX packets 14733665  bytes 1007493238 (960.8 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 15040396  bytes 4534954611 (4.2 GiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
    inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
    ether 02:42:09:fe:1d:e5  txqueuelen 0  (Ethernet)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

`netstat -i` 和 `ip` 也挺好用，它们还可以打印路由表

```
$ netstat -i
Kernel Interface table

```

Iface	MTU	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
cni0	1450	14733471	0	0 0	15040202	0	0	0	0	BMRU
docker0	1500	0	0	0 0	0	0	0	0	0	BMU
eth0	1500	42833038	0	0 0	40961577	0	0	0	0	BMRU
flannel.1	1450	4761436	0	0 0	4091581	0	0	0	0	BMRU
lo	65536	376447249	0	0 0	376447249	0	0	0	0	LRU
veth2052e2d6	1450	7358995	0	0 0	7518642	0	0	0	0	BMRU
veth3190806e	1450	7363232	0	0 0	7510100	0	0	0	0	BMRU

`ip` (ip COMMAND CHEAT SHEET for Red Hat Enterprise Linux)

(https://access.redhat.com/sites/default/files/attachments/rh_ip_command_cheatsheet_1214_jcs_print.pdf)

`ip link`: Show information for all interfaces, `ip addr`: Show information for all addresses

```
$ ip link
$ ip addr
```

3. 有效电话号码

Source: <https://leetcode-cn.com/problems/valid-phone-numbers/>

给定一个包含电话号码列表（一行一个电话号码）的文本文件 file.txt，写一个 bash 脚本输出所有有效的电话号码。

你可以假设一个有效的电话号码必须满足以下两种格式： (xxx) xxx-xxxx 或 xxx-xxx-xxxx。（x 表示一个数字）

你也可以假设每行前后没有多余的空格字符。

示例:

假设 file.txt 内容如下:

987-123-4567

123 456 7890

(123) 456-7890

你的脚本应当输出下列有效的电话号码:

987-123-4567

(123) 456-7890

有三种方式可以实现

1. 基本正则表达式 (Basic regex BRE)
2. 扩展正则表达式 (extended regex ERE)
3. 与 Perl 兼容的正则表达式 (Perl-compatible regex PCRE)

```
$ grep "^(\([0-9]\{3\}\) \|[0-9]\{3\}-\)[0-9]\{3\}-[0-9]\{4\})$" file.txt # BRE
$ grep -E "^(\([0-9]{3}\) \|^([0-9]{3}-)[0-9]{3}-[0-9]{4})$" file.txt # ERE
$ grep -P "^(\([0-9]{3}\) \|[0-9]{3}-)[0-9]{3}-[0-9]{4})$" file.txt # PCRE
```

默认为基本正则表达式，加上 -E 变成 ERE，-P 变成 PCRE

在 grep 中，^ 标记着单词的开始，\$ 标记着单词的结束。-q 选项 禁止 grep 产生任何输出。

4. 第十行

Source: <https://leetcode-cn.com/problems/tenth-line/>

给定一个文本文件 file.txt，仅打印文件 的第 10 行。

例：

假定 file.txt 具有以下内容：

```
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Line 9
Line 10
```

您的脚本应输出第十行，即：

```
Line 10
```

注意：

- 1.如果文件少于 10 行，您应该输出什么？
- 2.至少有三种不同的解决方案。尝试探索所有可能性。

`head -10 file.txt | tail -1` 如果文件少于 10 行，则自然会被视为无法输出第 10 行的候选项。但是，可以通过以下方式轻松修复。

```
tail -n+10 file.txt | head -1
```

另外，`sed`，`awk`，`cut` 和 `mapfile` 可以做的工作，以及（例如，以下）

```
# Solution 1
$awk 'FNR == 10 {print }' file.txt
# OR
$awk 'NR == 10' file.txt
# Solution 2
$sed -n '10' file.txt
# Solution 3
$mapfile -s 9 -n 1 array < file.txt
$echo "${array[0]}" | head -1
# Solution 4
$cut -f10 -d$'\n' file.txt
```

5. 转置文件

Source: <https://leetcode.com/problems/transpose-file/>

给定一个文本文件 file.txt，转置其内容。

您可以假设每一行具有相同的列数，并且每个字段都由' '字符分隔。

例：

如果 file.txt 具有以下内容：

```
name age
alice 21
ryan 30
```

输出以下内容：

```
name alice ryan
age 21 30
```

```
awk '
{
    for (i = 1; i <= NF; i++) {
        if(NR == 1) {
            s[i] = $i;
        } else {
            s[i] = s[i] " " $i;
        }
    }
}
END {
    for (i = 1; s[i] != ""; i++) {
        print s[i];
    }
}' file.txt
```

仅在读取最后一行后才执行带有“END”前缀的代码块；同样，将在读取任何行之前执行带有“BEGIN”前缀的代码块。

AWK 是基于行的：主代码块（不带前缀的代码块）一次处理一行输入。

NR：一个变量，指示在读取的多个文件中累积的记录数（即当前行号）。FNR 与相似 NR，但每个读取的文件都会重置。由于在这个问题中我们只需要处理一个文件，所以使用都可以。

NF：一个变量，指示输入行上的字段数（即“列”数）。

\$i：输入行的第 i 个字段。

t[]：用于保存转置表的数组。

6. 词频

Source: <https://leetcode.com/problems/word-frequency/>

编写一个 bash 脚本来计算文本文件中每个单词的出现频率 words.txt。

为简单起见，您可以假设：

- words.txt 仅包含小写字符和空格 ' ' 字符。
- 每个单词只能包含小写字符。
- 单词由一个或多个空格字符分隔。

例：

假设 words.txt 具有以下内容：

```
the day is sunny the the
the sunny is is
```

您的脚本应输出以下内容，并按降序排列：

```
the 4
is 3
sunny 2
day 1
```

注意：

不必担心处理纽带，可以确保每个单词的频率计数都是唯一的。

您可以使用 Unix 管道将其写成一吗？

Solution:

```
$ cat words.txt | tr -s ' ' '\n' | sort | uniq -c | sort -r | awk '{ print $2, $1 }'
```

tr -s: 使用目标字符串截断字符串，但仅保留一个实例（例如，多个空格）

sort: 连续产生相同的字符串，以便 uniq 可以完全正确地计数相同的字符串。

uniq -c: uniq 用于过滤连续的重复行，-c 表示计数

sort -r: -r 表示按降序排序

awk '{print \$ 2, \$ 1}': 格式化输出

7. shell 中 `{}` 与 `()` 各是什么意思

`{}` 变量

`()` 命令

8. 如何在 linux 中打印所有环境变量

`$ printenv`

9. 多服务器的系统时间不一致如何解决

Source: https://support-it.huawei.com/docs/zh-cn/hcs-6.5.0/troubleshooting-guide/o_troubleshooting_NTPtime.html

Solution: 利用 ntp 服务

10. 服务器的平均负载如何计算

Source: <https://q.shanyue.tech/base/linux/299.html>

load average 指单位时间内运行态进程及不可中断进程的平均进程数, 运行态进程指正在使用或者 等待使用 CPU 的进程, 不可中断进程指正等待一些 IO 操作的进程。可使用 *uptime* 查看此指标。

```
$ uptime
16:48:09 up 2 days, 23:43, 2 users, load average: 0.01, 0.21, 0.20
```

11. 如何查看 node_modules(某一文件夹) 的体积有多大

Source: <https://q.shanyue.tech/base/linux/278.html>

`du` 命令可以查看磁盘的使用情况, 从它可以看出来文件及目录的大小

```
# -d 搜索深度
$ du -hd 0 node_modules
132M    node_modules
```

-h

--human-readable

追加的尺寸字母如“M”为兆字节到每个大小。

使用的是 1024 的幂, 而不是 1000 的幂; 'M'代表 1,048,576 字节。

如果您希望使用 1000 的幂, 请使用 '-H'或 '--si'选项。

12. Linux 文件系统的 mtime, ctime 有什么区别?

Source: <https://www.zhihu.com/question/29778427>

mtime: modified time 指文件内容改变的时间戳

ctime: change time 指文件属性改变的时间戳, 属性包括 mtime。

而在 windows 上, 它表示的是 creation time

(mtime: 最后一次更新档案数据的时间; ctime: 最后一次更新状态的时间, 如权限和属性;)

所以 ctime 会比 mtime 要大一些, 使用 stat 查看文件属性如下

```
$ stat hello.txt
  File: 'hello.txt'
  Size: 30          Blocks: 8          IO Block: 4096   regular file
Device: fd01h/64769d Inode: 917526     Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2019-12-10 16:15:55.253325208 +0800
Modify: 2019-12-10 16:15:52.740653330 +0800
Change: 2019-12-10 16:15:52.742653069 +0800
 Birth: -
```

而 http 服务选择 Last_Modified 时一般会选择 mtime

13. 如何查看一个文件的 inode number

Source: <https://q.shanyue.tech/base/linux/115.html>

可以使用 ls 或者 stat

```
$ stat hello.txt
  File: 'hello.txt'
  Size: 30          Blocks: 8          IO Block: 4096   regular file
Device: fd01h/64769d Inode: 917526     Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2019-12-10 16:15:55.253325208 +0800
Modify: 2019-12-10 16:15:52.740653330 +0800
Change: 2019-12-10 16:15:52.742653069 +0800

$ ls -li hello.txt
917526 hello.txt
```

14. 在 linux 中如何查看 cron 执行的情况

Source: <https://q.shanyue.tech/base/linux/99.html>

在 centos 中, 查看日志文件

```
$ tail -f /var/log/cron
Dec  5 19:30:01 8 CROND[24068]: (root) CMD (echo hello, world)
Dec  5 19:31:01 8 CROND[24084]: (root) CMD (echo hello, world)
Dec  5 19:31:01 8 CROND[24083]: (root) CMD (echo hello)
Dec  5 19:32:01 8 CROND[24094]: (root) CMD (echo hello, world)
Dec  5 19:32:01 8 CROND[24093]: (root) CMD (echo hello)
Dec  5 19:33:01 8 CROND[24104]: (root) CMD (echo hello, world)
Dec  5 19:33:01 8 CROND[24103]: (root) CMD (echo hello)
Dec  5 19:34:01 8 CROND[24113]: (root) CMD (echo hello)
```

```
Dec  5 19:34:01 8 CROND[24114]: (root) CMD (echo hello, world)
```

15. 检查某台主机是否可达

```
$ ping ADDRESS
```

ADDRESS 可以是主机名、域名或者 IP 地址。

如果主机可达，那么会输出如下信息：

```
$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=1.44 ms
^C
--- 192.168.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.440/1.440/1.440/0.000 ms
$ ping google.com
PING google.com (209.85.153.104) 56(84) bytes of data.
64 bytes from bom01s01-in-f104.1e100.net (209.85.153.104):
icmp_seq=1 ttl=53 time=123 ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 123.388/123.388/123.388/0.000 ms
```

如果主机不可达，则输出如下信息：

```
$ ping 192.168.0.99
PING 192.168.0.99 (192.168.0.99) 56(84) bytes of data.
From 192.168.0.82 icmp_seq=1 Destination Host Unreachable
From 192.168.0.82 icmp_seq=2 Destination Host Unreachable
```

在主机不可达时， ping 返回错误信息 Destination Host Unreachable 。

15~20: source: <https://linuxsecurityblog.com/2020/07/10/top-linux-interview-questions/>

15. 如何检查可用磁盘空间?

```
df -ah
```

```
qubits@LAPTOP-70LNOM71:~$ df -ah
Filesystem      Size  Used Avail Use% Mounted on
rootfs          917G   77G  841G   9% /
none            917G   77G  841G   9% /dev
sysfs            0      0      0    - /sys
proc             0      0      0    - /proc
devpts           0      0      0    - /dev/pts
none            917G   77G  841G   9% /run
none            917G   77G  841G   9% /run/lock
none            917G   77G  841G   9% /run/shm
none            917G   77G  841G   9% /run/user
binfmt_misc      0      0      0    - /proc/sys/fs/binfmt_misc
cgroup           917G   77G  841G   9% /sys/fs/cgroup
cgroup           0      0      0    - /sys/fs/cgroup/devices
C:\              917G   77G  841G   9% /mnt/c
D:\              14G    13G   1.7G  89% /mnt/d
F:\              932G   61G  871G   7% /mnt/f
```

16. 如何查看机器的内核版本?

```
uname -a
```

```
qubits@LAPTOP-70LNOM71:~$ uname -a
Linux LAPTOP-70LNOM71 4.4.0-18362-Microsoft #836-Microsoft Mon May 05 16:04:00 PST 2020 x86_64 GNU/Linux
qubits@LAPTOP-70LNOM71:~$
```

17. 如何在 Linux 系统上启动服务?

```
service apache2 start
```

```
qubits@LAPTOP-70LNOM71:~$ sudo service apache2 start
Starting Apache httpd web server: apache2.
qubits@LAPTOP-70LNOM71:~$
```

18. 如何检查 Linux 系统的 IP?

```
ifconfig
```

```

qubits@LAPTOP-70LNOM71:~$ ifconfig
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.1 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::165:2100:8441:d642 prefixlen 64 scopeid 0xfd<compat,link,site,host>
    ether 0a:00:27:00:00:08 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 1500
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0xfe<compat,link,site,host>
    loop (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.6 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2001:bb6:307b:c358:b0de:9079:75b7:1022 prefixlen 64 scopeid 0x0<global>
    inet6 2001:bb6:307b:c358:21c2:dee2:46b7:b918 prefixlen 128 scopeid 0x0<global>
    inet6 fe80::b0de:9079:75b7:1022 prefixlen 64 scopeid 0xfd<compat,link,site,host>
    ether 3c:95:09:e2:14:d3 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

19. 假设您有一个在 Linux 上运行的 Apache Web 服务器，在哪里可以找到该服务器的 index.html 文件？

```
cd /var/www/html
```

```

qubits@LAPTOP-70LNOM71:~$ cd /var/www/html
qubits@LAPTOP-70LNOM71:/var/www/html$ ls
index.html
qubits@LAPTOP-70LNOM71:/var/www/html$

```

20. 如何将文件的全部权限授予读取，写入和执行。

```
sudo chmod 777 hello.py
```

```

qubits@LAPTOP-70LNOM71:~$ sudo chmod 777 hello.py
qubits@LAPTOP-70LNOM71:~$ ls
Desktop  Downloads  hello.py  msfinstall  Pictures  searchsploit2
Documents  Empire  mimikatz  Music  Public  Templates
qubits@LAPTOP-70LNOM71:~$

```

21. 统计 Nginx 日志里访问次数最多的前十个 IP

Source: https://mp.weixin.qq.com/s/kC_Ny3gCh1m7ppFdXiXEhg

https://blog.csdn.net/cooleyes_yu/article/details/90439181

统计一下 /var/log/nginx/access.log 日志中访问量最多的前十个 IP?

```
$ awk '{print $1}' /var/log/nginx/access.log | sort | uniq -c |  
sort -nr -k1 | head -n 10
```

Nginx:

```
awk '{print $1}' urlogfile | sort | uniq -c | sort -nr -k1 | head -n 10  
awk '{print $1}' /usr/local/nginx/logs/localhost.access.log | sort |  
uniq -c | sort -nr -k1 | head -n 10
```

Apache:

```
cd /var/log/httpd/&&\  
cat access_log | awk '{print $1}' | uniq -c | sort -rn -k1 | head -n  
10
```

说明:

awk '{print \$1}': 取数据的低 1 域 (第 1 列)

sort: 对 IP 部分进行排序。

uniq -c: 打印每一重复行出现的次数。(并去掉重复行)

sort -nr -k1: 按照重复行出现的次序倒序排列,-k1 以第一列为标准排序。

head -n 10: 取排在前 5 位的 IP 。

22. 把 /data 目录及其子目录下所有以扩展名 .txt 结尾的文件中包含 magedu 的字符串全部替换成 magestudy

Source: <https://www.twblogs.net/a/5b95800a2b717750bda4d81e/?lang=zh-cn>

```
$ find /data -type f -name "*.txt" | xargs sed -i 's/magedu/magestudy/g'
```

注: | 和 xargs 区别:

echo '--help' | cat 输出 --help

echo '--help' | xargs cat 输出为 cat --help 的结果

思考: 把/data 目录和子目录下以扩展名 .txt 文件名中包含 magedu 替换成 magestudy

```
$ find /data -type f -name "*magedu*.txt" | awk -F "magedu" '{print "mv "  
$1"magedu"$2 " " $1"magestudy"$2}' | sh
```

注: -F "magedu" 是将 magedu 当作分隔符

|sh 是将前面的输出的内容当作命令执行。

23. 用 Shell 脚本或命令将/opt 目录下的大于 15KB 的文件都移到/tmp 目录下

Source: <https://www.twblogs.net/a/5b95800a2b717750bda4d81e/?lang=zh-cn>

方法 1: (判断文件和判断文件大小分开)

```
#!/usr/bin/bash
for i in `ls /opt`
do
    if test -f $i ; then
        if test `ls -l /opt/$i | awk '{print $5}'` -gt 15000 ;then
            mv /opt/$i /tmp
        fi
    fi
done
```

方法 2: (判断文件和判断大小在同一条命令里)

注: grep -v ^d 不能改成 grep -v ^- 因为 ls -l 显示的第一个字段的首字母不一定是-, 还可能是 l 链接等文件类型。

```
#!/usr/bin/bash
for i in `ls -l /opt | grep -v ^d | awk '{if($5>15000){print $9}}'`
do
    mv /opt/$i /tmp
done
```

扩展:

用 find 命令可以查找/opt 整个目录下包含子目录下的所有于 15k 的文件, 而上面两种方法用的是 ls -l 命令, 仅仅是对/opt 目录下 (不包括子目录) 的文件操作。

```
find /opt -type f -size +15k -exec mv {} /tmp \;
```