

cut 命令可以从一个文本文件或者文本流中提取文本列。

命令用法：

```
cut -b list [-n] [file ...]
```

```
cut -c list [file ...]
```

```
cut -f list [-d delim][-[s]][file ...]
```

l 上面的-b、-c、-f 分别表示字节、字符、字段（即 byte、character、field）；

l list 表示-b、-c、-f 操作范围，-n 常常表示具体数字；

l file 表示的自然是要操作的文本文件的名称；

l delim（英文全写：delimiter）表示分隔符，默认情况下为 TAB；

l -s 表示不包括那些不含分隔符的行（这样有利于去掉注释和标题）

上面三种方式中，表示从指定的范围中提取字节（-b）、或字符（-c）、或字段（-f）。

范围的表示方法：

N

只有第 N 项

N-

从第 N 项一直到行尾

N-M

从第 N 项到第 M 项(包括 M)

-M

从一行的开始到第 M 项(包括 M)

-

从一行的开始到结束的所有项

下面是实例，先以较简单的“命令用法”中提及的第二条开始讲起：

```
[root@server1 ~]# whatis ls
ls: nothing appropriate
```

interrupts 文件中的字符排列非常齐整，正适合我们切豆腐。

但这里我们只对两个数字列感兴趣，用法如下：

```
[root@server1 ~]# man whatis_
```

里面还有一些不需要的内容，精减一下：

```
[root@SkyFly /]# grep '[[:digit:]]:' /proc/interrupts
0:          830      IO-APIC-edge      timer
1:         9384      IO-APIC-edge      i8042
6:           3      IO-APIC-edge      floppy
7:           0      IO-APIC-edge      parport0
8:           1      IO-APIC-edge      rtc
9:           0      IO-APIC-fasteoi    acpi
12:        1560      IO-APIC-edge      i8042
14:           0      IO-APIC-edge      libata
15:          39      IO-APIC-edge      libata
16:        5319      IO-APIC-fasteoi    eth1
17:           0      IO-APIC-fasteoi    ehci_hcd:usb1
18:           0      IO-APIC-fasteoi    uhci_hcd:usb2, Ensoniq AudioPCI
19:       3074      IO-APIC-fasteoi    ioc0
```

合到一起：

```
[root@SkyFly ~]# grep '[[digit:]]:' /proc/interrupts |cut -c1-15
0:      830
1:     9534
6:         3
7:         0
8:         1
9:         0
12:     1573
14:         0
15:        39
16:     5330
17:         0
18:         0
19:     3074
```

哇，果然够帅!!

不相邻列的截选又应该如何做呢？

```
[root@SkyFly ~]# grep '[[digit:]]:' /proc/interrupts |cut -c1-4,34-
0:    timer
1:    i8042
6:    floppy
7:    parport0
8:    rtc
9:    acpi
12:   i8042
14:   libata
15:   libata
16:   eth1
17:  ehci_hcd:usb1
18:  uhci_hcd:usb2, Ensoniq AudioPCI
19:   ioc0
```

这种方式需要事先确定占多少个字符位置，不仅麻烦，而且容易出错。

下面的问题该怎么去做？

```
[root@SkyFly ~]# head -15 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

这就是第二讲：使用 cut -f 提取文本中的字段。

cut -c 主要是用来在固定字符位置或个数的文本文件中提取，对于上面的例子就显得无能为力了。仔细观察，发现 passwd 文件有个规律，就是以冒号来区分不同的段的文本，于是。。。

```
[root@SkyFly proc1# cut -f1 -d':' /etc/passwd |head -15
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
gopher
ftp
```

怎么样，好玩吧~!

继续，创建一个文本文件，名为 a.txt，名字有点土，凑合着用吧。

```
[root@SkyFly pub1# cat a.txt
#test_only

A1      A2      A3      A4
B1      B2      B3      B4
C1      C2      C3      C4
D1      D2      D3      D4
```

A1、B1、C1 所代表的行字符之间均以 TAB 分隔，D1 却是以空格来分开的。

```
[root@SkyFly pub1# cut -f1- -s a.txt
A1      A2      A3      A4
B1      B2      B3      B4
C1      C2      C3      C4
```

看到-s 的作用了吗？(因为第一行不含有任何 TAB 字符，所以直接被剔除了)，而最后一行(即 D1 行)，是以空格区分间距，所以也不合要求。

```
[root@SkyFly pub1# cut -f1- -s --output-delimiter='/' a.txt
A1/A2/A3/A4
B1/B2/B3/B4
C1/C2/C3/C4
```