cut 是一个选取命令，就是将一段数据经过分析，取出我们想要的。一般来说，选取信息通常是针对“行”来进行分析的，并不是整篇信息分析的。

（1）其语法格式为：
cut [-bn] [file] 或 cut [-c] [file] 或 cut [-df] [file]

使用说明
cut 命令从文件的每一行剪切字节、字符和字段并将这些字节、字符和字段写至标准输出。
如果不指定 File 参数，cut 命令将读取标准输入。必须指定 -b、-c 或 -f 标志之一。

主要参数
-b ：以字节为单位进行分割。这些字节位置将忽略多字节字符边界，除非也指定了 -n 标志。
-c ：以字符为单位进行分割。
-d ：自定义分隔符，默认为制表符。
-f ：与-d 一起使用，指定显示哪个区域。
-n ：取消分割多字节字符。仅和 -b 标志一起使用。如果字符的最后一个字节落在由 -b 标志的 List 参数指示的
范围之内，该字符将被写出；否则，该字符将被排除。

（2）cut 一般以什么为依据呢? 也就是说，我怎么告诉 cut 我想定位到的剪切内容呢?

cut 命令主要是接受三个定位方法：

第一，字节（bytes），用选项-b

第二，字符（characters），用选项-c

第三，域（fields），用选项-f

（3）以“字节”定位

举个例子吧，当你执行 ps 命令时，会输出类似如下的内容：

[rocrocket@rocrocket programming]$ who
rocrocket :0 2009-01-08 11:07
rocrocket pts/0 2009-01-08 11:23 (:0.0)
rocrocket pts/1 2009-01-08 14:15 (:0.0)
如果我们想提取每一行的第 3 个字节，就这样：

[rocrocket@rocrocket programming]$ who|cut -b 3
c
c
c

（4）如果“字节”定位中，我想提取第 3，第 4、第 5 和第 8 个字节，怎么办?

-b 支持形如 3-5 的写法，而且多个定位之间用逗号隔开就成了。看看例子吧：

[rocrocket@rocrocket programming]$ who|cut -b 3-5,8
croe
croe
croe
但有一点要注意，cut 命令如果使用了-b 选项，那么执行此命令时，cut 会先把-b 后面所有的定位进行从

小到大排序，然后再提取。可不能颠倒定位的顺序哦。这个例子就可以说明这个问题：

```
[rocrocket@rocrocket programming]$ who|cut -b 8,3-5
croe
croe
croe
```

（5）还有哪些类似“3-5”这样的小技巧，列举一下吧!

```
[rocrocket@rocrocket programming]$ who
rocrocket :0          2009-01-08 11:07
rocrocket pts/0       2009-01-08 11:23 (:0.0)
rocrocket pts/1       2009-01-08 14:15 (:0.0)
[rocrocket@rocrocket programming]$ who|cut -b -3
roc
roc
roc
[rocrocket@rocrocket programming]$ who|cut -b 3-
crocket :0          2009-01-08 11:07
crocket pts/0       2009-01-08 11:23 (:0.0)
crocket pts/1       2009-01-08 14:15 (:0.0)
```

想必你也看到了，-3 表示从第一个字节到第三个字节，而 3-表示从第三个字节到行尾。如果你细心，你可以看到这两种情况下，都包括了第三个字节“c”。

如果我执行 who|cut -b -3,3-，你觉得会如何呢？答案是输出整行，不会出现连续两个重叠的 c 的。看：

```
[rocrocket@rocrocket programming]$ who|cut -b -3,3-
rocrocket :0          2009-01-08 11:07
rocrocket pts/0       2009-01-08 11:23 (:0.0)
rocrocket pts/1       2009-01-08 14:15 (:0.0)
```

（6）给个以字符为定位标志的最简单的例子吧!

下面例子你似曾相识，提取第 3，第 4，第 5 和第 8 个字符：

```
[rocrocket@rocrocket programming]$ who|cut -c 3-5,8
croe
croe
croe
```

不过，看着怎么和-b 没有什么区别啊？莫非-b 和-c 作用一样? 其实不然，看似相同，只是因为这个例子举的不好，who 输出的都是单字节字符，所以用-b 和-c 没有区别，如果你提取中文，区别就看出来了，来，看看中文提取的情况：

```
[rocrocket@rocrocket programming]$ cat cut_ch.txt
星期一
星期二
星期三
星期四
[rocrocket@rocrocket programming]$ cut -b 3 cut_ch.txt

```




[rocrocket@rocrocket programming]$ cut -c 3 cut_ch.txt
一
二
三
四
看到了吧，用-c 则会以字符为单位，输出正常；而-b 只会傻傻的以字节（8 位二进制位）来计算，输出就是乱码。
既然提到了这个知识点，就再补充一句，如果你学有余力，就提高一下。
当遇到多字节字符时，可以使用-n 选项，-n 用于告诉 cut 不要将多字节字符拆开。例子如下：

[rocrocket@rocrocket programming]$ cat cut_ch.txt |cut -b 2




[rocrocket@rocrocket programming]$ cat cut_ch.txt |cut -nb 2

[rocrocket@rocrocket programming]$ cat cut_ch.txt |cut -nb 1,2,3
星
星
星
星
（7）域是怎么回事呢？解释解释:)

为什么会有“域”的提取呢，因为刚才提到的-b 和-c 只能在固定格式的文档中提取信息，而对于非固定格式的信息则束手无策。这时候“域”就派上用场 了。如果你观察过/etc/passwd 文件，你会发现，它并不像 who 的输出信息那样具有固定格式，而是比较零散的排放。但是，冒号在这个文件的每一行 中都起到了非常重要的作用，冒号用来隔开每一个项。

我们很幸运，cut 命令提供了这样的提取方式，具体的说就是设置“间隔符”，再设置“提取第几个域”，就 OK 了！

以/etc/passwd 的前五行内容为例：

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5
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
[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1
root
bin

daemon
adm
lp
看到了吧，用-d 来设置间隔符为冒号，然后用-f 来设置我要取的是第一个域，再按回车，所有的用户名就都列出来了！呵呵 有成就感吧！
当然，在设定-f 时，也可以使用例如 3-5 或者 4-类似的格式：

[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1,3-5
root:0:0:root
bin:1:1:bin
daemon:2:2:daemon
adm:3:4:adm
lp:4:7:lp
[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f 1,3-5,7
root:0:0:root:/bin/bash
bin:1:1:bin:/sbin/nologin
daemon:2:2:daemon:/sbin/nologin
adm:3:4:adm:/sbin/nologin
lp:4:7:lp:/sbin/nologin
[rocrocket@rocrocket programming]$ cat /etc/passwd|head -n 5|cut -d : -f -2
root:x
bin:x
daemon:x
adm:x
lp:x
（8）如果遇到空格和制表符时，怎么分辨呢？我觉得有点乱，怎么办？

有时候制表符确实很难辨认，有一个方法可以看出一段空格到底是由若干个空格组成的还是由一个制表符组成的。

[rocrocket@rocrocket programming]$ cat tab_space.txt
this is tab finish.
this is several space finish.
[rocrocket@rocrocket programming]$ sed -n l tab_space.txt
this is tab\tfinish.$
this is several space finish.$
看到了吧，如果是制表符（TAB），那么会显示为\t 符号，如果是空格，就会原样显示。
通过此方法即可以判断制表符和空格了。
注意，上面 sed -n 后面的字符是 L 的小写字母哦，不要看错。

（9）我应该在 cut -d 中用什么符号来设定制表符或空格呢?

其实 cut 的-d 选项的默认间隔符就是制表符，所以当你就是要使用制表符的时候，完全就可以省略-d 选项，而直接用 - f 来取域就可以了。

如果你设定一个空格为间隔符，那么就这样：

[rocrocket@rocrocket programming]$ cat tab_space.txt |cut -d ' ' -f 1

this
this
注意，两个单引号之间可确实要有一个空格哦，不能偷懒。
而且，你只能在-d 后面设置一个空格，可不许设置多个空格，因为 cut 只允许间隔符是一个字符。

[rocrocket@rocrocket programming]$ cat tab_space.txt |cut -d ' ' -f 1
cut: the delimiter must be a single character
Try `cut --help' for more information.

（10）cut 有哪些缺陷和不足？

猜出来了吧？对，就是在处理多空格时。
如果文件里面的某些域是由若干个空格来间隔的，那么用 cut 就有点麻烦了，因为 cut 只擅长处理“以一个字符间隔”的文本内容