第11章　合并与分割

几年前，我习惯于使用运行在终端的 PICK操作的UNIX集合，我实际使用 PICK应用的大部分时间花费在分类与连接过程中，且使用极其频繁。很幸运我没有成为一个全职的 PICK操作员。

有几种工具用来处理文本文件分类、合并和分割操作，本章详细介绍这些工具。

本章内容有：

- 实用的分类（sort）操作。
- uniq。
- join。
- cut。
- paste。
- split。

11.1　sort用法

sort命令将许多不同的域按不同的列顺序分类。当查阅注册文件或为另一用户对下载文件重排文本列时， sort工具很方便。实际上，使用其他 UNIX工具时，已假定工作文件已经被分过类。无论如何，分类文件比不分类文件看起来更有意义。

11.1.1　概述

UNIX/LINUX自带的sort功能很强大。尽管有时在使用 sort各种不同的选项时人们已经很小心，但仍会产生意想不到的结果。 sort选项很长，甚至有时在各种不同开关的实际功能和结果进行比较时也会遇到麻烦，原因可能是在结合使用 sort的不同选项时有些概念模糊不清。

本章不讨论各种不同的 sort方法（不能说sort不够强大；它很慢，但观察数值交替变化是很有趣的）也不讨论各种不同开关的结合使用功效。本章只讲到主要的 sort选项，伴随有大量实例。与sort结合使用的uniq、join、cut和paste方法与split方法也将会涉及到。

上面提到， sort命令选项很长，下面介绍本章使用的各种选项。

11.1.2　sort选项

sort命令的一般格式为：

```
sort -cmu -o output_file [other options] +pos1 +pos2 input_files
```

下面简要介绍一下sort的参数：

-c 测试文件是否已经分类。

-m 合并两个分类文件。

-u 删除所有复制行。

-o 存储sort结果的输出文件名。

其他选项有：

-b 使用域进行分类时，忽略第一个空格。

-n 指定分类是域上的数字分类。

-t 域分隔符；用非空格或tab键分隔域。

-r 对分类次序或比较求逆。

+n n为域号。使用此域号开始分类。

n n为域号。在分类比较时忽略此域，一般与 +n一起使用。

post1 传递到m，n。m为域号，n为开始分类字符数；例如4，6意即以第5域分类，从第7个字符开始。

11.1.3 保存输出

-o选项保存分类结果，然而也可以使用重定向方法保存。下面例子保存结果到 results.out：

```
$ sort video.txt >results.out
```

11.1.4 sort启动方式

缺省情况下，sort认为一个空格或一系列空格为分隔符。要加入其他方式分隔，使用 -t选项。

sort执行时，先查看是否为域分隔设置了 -t选项，如果设置了，则使用它来将记录分隔成域0、域1等等；如果未设置，用空格代替。缺省时 sort将整个行排序，指定域号的情况例外。

下面是文件video.txt的清单，包含了上个季度家电商场的租金情况。各域为：（1）名称，（2）供货区代码，（3）本季度租金，（4）本年租金。域分隔符为冒号。为此对此例需使用‘ -t ’选项。文件如下：

```
$ pg video.txt
Boys in Company C:HK:192:2192
Alien:HK:119:1982
The Hill:KL:63:2972
Aliens:HK:532:4892
Star Wars:HK:301:4102
A Few Good Men:KL:445:5851
Toy Story:HK:239:3972
```

11.1.5 sort对域的参照方式

关于sort的一个重要事实是它参照第一个域作为域 0，域1是第二个域，等等。sort也可以使用整行作为分类依据。为防止混淆，对于此文件用户应按如下方式参照域并做分类依据：

Field 0	Field 1	Field 2	Field 3
Star Wars	HK	301	4102
A Few Good Men	KL	445	5851

sort将定位各域，因此应把域0作为分类键0，域1作为分类键1等等。

11.1.6 文件是否已分类

怎样分辨文件是否已分类？如果只有30行，看看就知道了，但如果是400行呢，使用sort-c

通知sort文件是否按某种顺序分类。

```
$ sort -c video.txt
sort: disorder on video.txt
```

结果显示未分类，现在将之分类，再试一次：

```
$ sort -c video.txt
$
```

返回提示符表明已分类。然而如果测试成功，返回一个信息行会更好。

11.1.7 基本sort

最基本的sort方式为sort filename，按第一域进行分类（分类键0）。实际上读文件时sort操作将行中各域进行比较，这里返回基于第一域sort的结果，如下所示：

```
$ sort -t: video.txt
Alien:HK:119:1982
Aliens:HK:532:4892
Boys in Company C:HK:192:2192
A Few Good Men:KL:445:5851
Star Wars:HK:301:4102
The Hill:KL:63:2972
Toy Story:HK:239:3972
```

11.1.8 sort分类求逆

如果要逆向sort结果，使用-r选项。在通读大的注册文件时，使用逆向sort很方便。下面是按域0分类的逆向结果。

```
$ sort -t: -r video.txt
Toy Story:HK:239:3972
The Hill:KL:63:2972
Star Wars:HK:301:4102
A Few Good Men:KL:445:5851
Boys in Company C:HK:192:2192
Aliens:HK:532:4892
Alien:HK:119:1982
```

11.1.9 按指定域分类

有时需要只按第2域（分类键1）分类。这里为重排报文中供应区代码，使用 t1，意义为按分类键1分类。下面的例子中，所有供应区代码按分类键 1分类；注意分类键2和3对应各域也被分类。

```
$ sort -t: +1  video.txt
Alien:HK:119:1982
Boys in Company C:HK:192:2192
Toy Story:HK:239:3972
Star Wars:HK:301:4102
Aliens:HK:532:4892
A Few Good Men:KL:445:5851
The Hill:KL:63:2972
```

11.1.10 数值域分类

依此类推，要按第三分类键分类，使用t3。但是因为这是数值域，即为数值分类，可以使

用-n选项。下面例子为按年租金分类命令及结果：

```
$ sort -t: +3n  video.txt
Alien:HK:119:1982
Boys in Company C:HK:192:2192
The Hill:KL:63:2972
Toy Story:HK:239:3972
Star Wars:HK:301:4102
Aliens:HK:532:4892
A Few Good Men:KL:445:5851
```

如果不加-n，结果会怎样？这里假定按第 3域分类，找出最好的季度租金。因为是分类键2，所以使用t2。

```
$ sort -t: +2 video.txt
Alien:HK:119:1982
Boys in Company C:HK:192:2192
Toy Story:HK:239:3972
Star Wars:HK:301:4102
A Few Good Men:KL:445:5851
Aliens:HK:532:4892
The Hill:KL:63:2972
```

观察结果，分类进行了，但不是预想的结果，因为第 3域为数值域。当然这个结果也是某种类型的排列，录像机The Hill应该在第二行，但结果是：sort只查看第3域每个数值的第一个数，并按其分类，然后再按第二个数依次下去。

记住按数值域分类要加-n，这样才会得到预想结果。

```
$ sort -t: +2n video.txt
The Hill:KL:63:2972
Alien:HK:119:1982
Boys in Company C:HK:192:2192
Toy Story:HK:239:3972
Star Wars:HK:301:4102
A Few Good Men:KL:445:5851
Aliens:HK:532:4892
```

现在对了，可以看出本季度卖点最高的是 Aliens。如果使用-r选项，将会把Aliens放在第一行。

11.1.11 唯一性分类

有时，原文件中有重复行，这时可以使用 -u选项进行唯一性（不重复）分类以去除重复行，本例中Alien有相同的两行。带重复行的文件如下，其中 Alien插入了两次：

```
$ pg video.txt
Boys in Company C:HK:192:2192
Alien:HK:119:1982
The Hill:KL:63:2972
Aliens:HK:532:4892
Star Wars:HK:301:4102
A Few Good Men:KL:445:5851
Alien:HK:119:1982
```

使用-u选项去除重复行，不必加其他选项， sort会自动处理。

```
$ sort -u video.txt
Alien:HK:119:1982
Aliens:HK:532:4892
Boys in Company C:HK:192:2192
```

```
A Few Good Men:KL:445:5851
Star Wars:HK:301:4102
The Hill:KL:63:2972
```

11.1.12 使用k的其他sort方法

sort还有另外一些方法指定分类键。可以指定 k选项，第1域（分类键）以1开始。不要与前面相混淆。我经常使用这个选项。因为我习惯于第一域为数值 1，这样使用sort时用同样的数值做分类依据会更有意义。其他选项也可以使用 k，主要用于指定分类域开始的字符数目。

要在第1域进行分类，可以使用-k4，这是按年租金分类的次序。

```
$ sort -t: -k4  video.txt
Alien:HK:119:1982
Boys in Company C:HK:192:2192
The Hill:KL:63:2972
Star Wars:HK:301:4102
Aliens:HK:532:4892
A Few Good Men:KL:445:5851
```

11.1.13 使用k做分类键排序

可以指定分类键次序。先以第 4域，再以第1域分类，命令为-k4 -k1，也可以反过来，以便在文件首行显示最高年租金，方法如下：

```
$ sort -t: -r -k4 -k1 video.txt
A Few Good Men:KL:445:5851
Aliens:HK:532:4892
Star Wars:HK:301:4102
The Hill:KL:63:2972
Boys in Company C:HK:192:2192
Alien:HK:119:1982
```

11.1.14 指定sort序列

可以指定分类键顺序，也可以使用 -n选项指定不使用哪个分类键进行查询。看下面的 sort命令：

```
sort +0 -2 +3
```

该命令意即开始以域0分类，忽略域2，然后再使用域3分类。

11.1.15 pos用法

指定开始分类的域位置的另一种方法是使用如下格式：

```
sort +field_number .characters_in
```

意即从field_number开始分类，但是要在此域的第 characters_in个字符开始。

这里是一个例子，供应区代码加入一些后缀。如：

```
$ pg video.txt
Boys in Company C:HK48:192:2192
Alien:HK57:119:1982
The Hill:KL23:63:2972
Aliens:HK11:532:4892
```

```
Star Wars:HK38:301:4102
A Few Good Men:KL87:445:5851
Toy Story:HK65:239:3972
```

要只使用供应区代码后缀部分将文件分类，其命令为 +1.2，意即以第1域最左边第3个字符开始分类，其具体含义及脚本如下：

	Field 0	Field 1	Field 2	Field 3
	Aliens	H K 1 1	532	4892
Characters in:		0 1 2 3		

```
$ sort -t: +1.2 video.txt
Aliens:HK11:532:4892
The Hill:KL23:63:2972
Star Wars:HK38:301:4102
Boys in Company C:HK48:192:2192
Alien:HK57:119:1982
Toy Story:HK65:239:3972
A Few Good Men:KL87:445:5851
```

11.1.16 使用head和tail将输出分类

分类操作时，不一定要显示整个文件或一页以查看 sort结果中的第一和最后一行。如果只显示最高年租金，按第4域分类-k4并求逆，然后使用管道只显示sort输出的第一行，此命令为head，可以指定查阅行数。如果只有第一行，则为 head -1：

```
$ sort -t: -r -k4  video.txt | head -1
A Few Good Men:KL:445:5851
```

要查阅最低年租金，使用tail命令与head命令刚好相反，它显示文件倒数几行。 1为倒数一行，2为倒数两行等等。查阅最后一行为 tail -1。结合上述的sort命令和tail命令显示最低年租金：

```
$ sort -t: -r -k4 video.txt | tail -1
Alien:HK:119:1982
```

可以使用head或tail查阅任何大的文本文件， head用来查阅文件头，基本格式如下：

head [how_many_lines_to_display] file_name

Tail用来查阅文件尾，基本格式为：

tail [how_many_lines_to_display] file_name

如果使用head或tail时想省略显示行数，缺省时显示10行。

要查阅文件前20行：

```
$ head -20 file_name
```

要查阅文件后7行：

```
$ tail -7 file_name
```

11.1.17 awk使用sort输出结果

对数据分类时，对sort结果加一点附加信息很有必要，对其他用户尤其如此。使用 awk可以轻松完成这一功能。比如说采用上面最低租金的例子，需要将 sort结果管道输出到awk，不要忘了用冒号作域分隔符，显示提示信息和实际数据。

```
$ sort -t: -r -k4 video.txt|tail -1 | awk -F: '{print "Worst rental
",$1,"has been rented "$3}'

Worst rental Alien has been rented 119
```

11.1.18 将两个分类文件合并

将文件合并前，它们必须已被分类。合并文件可用于事务处理和任何种类的修改操作。下面这个例子，因为忘了把两个家电名称加入文件，它们被放在一个单独的文件里，现在将之并入一个文件。分类的合并格式为‘ sort -m sorted_file1 sorted_file2，下面是包含两个新家电名称的文件列表，它已经分类完毕：

```
$ pg video2.txt
Crimson Tide:134:2031
Die Hard:152:2981
```

使用-m +o。将这个文件并入已存在的分类文件 video.sort，要以名称域进行分类，实际上没有必要加入+o，但为了保险起见，还是加上的好。

```
$ sort -t: -m +0 video2.txt video.sort
Alien:HK:119:1982
Aliens:HK:532:4892
Boys in Company C:HK:192:2192
Crimson Tide:134:2031
Die Hard:152:2981
A Few Good Men:KL:445:5851
Star Wars:HK:301:4102
The Hill:KL:63:2972
```

11.2 系统sort

sort可以用来对/etc/passwd文件中用户名进行分类。这里需要以第 1域即注册用户名分类，然后管道输出结果到awk，awk打印第一域。

```
$ cat passwd | sort -t: +0 | awk -F":" '{print $1}'
adm
bin
daemon
...
...
```

sort还可以用于df命令，以递减顺序打印使用列。下面是一般 df输出。

```
$ df
Filesystem   1024-blocks Used   Available Capacity Mounted on
/dev/hda5    495714      291027 179086    62%      /
/dev/hda1    614672      558896 55776     91%      /dos
```

使用-b选项，忽略分类域前面的空格。使用域 4（+4），即容量列将分类求逆，最后得出文件系统自由空间的清晰列表。

```
$ df | sort -b -r +4
Filesystem   1024-blocks Used   Available Capacity Mounted on
/dev/hda1    614672      558896 55776     91%      /dos
/dev/hda5    495714      291027 179086    62%      /
```

在一个文本文件中存入所有 IP地址的拷贝，这样查看本机 IP地址更容易一些。有时如果在管理员权限下，就需要将此文件分类。将 IP地址按文件中某种数值次序分类时，需要指定

域分隔符为句点。这里只需关心IP地址的最后一段。分类应从此域即域3开始，未分类文件如下：

```
$ pg iplist
193.132.80.123  dave tansley
193.132.80.23   HP printer 2nd floor
193.132.80.198  JJ. Peter's scanner
193.132.80.38   SPARE
193.132.80.78   P.Edron
```

分类后结果如下：

```
$ sort -t.  +3n iplist
193.132.80.23   HP printer 2nd floor
193.132.80.38   SPARE
193.132.80.78   P.Edron
193.132.80.123  dave tansley
193.132.80.198  JJ. Peter's scanner
```

11.3 uniq用法

uniq用来从一个文本文件中去除或禁止重复行。一般uniq假定文件已分类，并且结果正确。我们并不强制要求这样做，如果愿意，可以使用任何非排序文本，甚至是无规律行。

可以认为uniq有点像sort命令中唯一性选项。对，在某种程度上讲正是如此，但两者有一个重要区别。sort的唯一性选项去除所有重复行，而uniq命令并不这样做。重复行是什么？在uniq里意即持续不断重复出现的行，中间不夹杂任何其他文本，现举例如下：

```
$ pg myfile.txt
May Day
May Day
May Day
Going Down
May Day.
```

uniq将前三个May Day看作重复副本，但是因为第4行有不同的文本，故并不认为第五行持续的May Day为其副本。uniq将保留这一行。

命令一般格式：

uniq -u d c -f input-file output-file

其选项含义：

-u 只显示不重复行。

-d 只显示有重复数据行，每种重复行只显示其中一行

-c 打印每一重复行出现次数。

-f n为数字，前n个域被忽略。

一些系统不识别-f选项，这时替代使用-n。

使用本节开始时的文本，创建文件myfile.txt，在此文件上运行uniq命令。

```
$ uniq myfile.txt
May Day
Going Down
May Day
```

注意第5行保留下来，其文本为最后一行May Day。如果运行sort -u，将只返回May Day和Going Down。

连续重复出现

使用-c选项显示行数，即每个重复行数目。本例中，行 May Day重复出现三次。

```
$ uniq -c myfile.txt
      3 May Day
      1 Going Down
      1 May Day
```

1. 不唯一

使用-d显示重复出现的不唯一行：

```
$ uniq -d myfile.txt
May Day
```

2. 对特定域进行测试

使用-n只测试一行一部分的唯一性。例如 -5意即测试第5域后各域唯一性。域从 1开始记数。

如果忽略第1域，只测试第2域唯一性，使用-n2，下述文件包含一组数据，其中第 2域代表组代码。

```
$ pg parts.txt
AK123 OP
DK122 OP
EK999 OP
```

运行uniq，将返回所有行。因为这个文件每一行都不同。

```
$ uniq -c parts.txt
      1 AK123 OP
      1 DK122 OP
      1 EK999 OP
```

如果指定测试在第1域后，结果就会不同。uniq会比较三个相同的OP，因此将返回一行。

```
$ uniq -f2 parts.txt
AK123 OP
```

如果‘ -f ’返回错误，替代使用：

```
$ uniq -n2 parts.txt
AK123 OP
```

11.4 join用法

join用来将来自两个分类文本文件的行连在一起。如果学过 SQL语言，可能会很熟悉join命令。

下面讲述join工作方式。这里有两个文件 file1和file2，当然已经分类。每个文件里都有一些元素与另一个文件相关。由于这种关系， join将两个文件连在一起，这有点像修改一个主文件，使之包含两个文件里的共同元素。

文本文件中的域通常由空格或 tab键分隔，但如果愿意，可以指定其他的域分隔符。一些系统要求使用join时文件域要少于20，为公平起见，如果域大于20，应使用DBMS系统。

为有效使用join，需分别将输入文件分类。

其一般格式为：

join [options] input-file1 input-file2.

让我们看看它的可用选项列表：

an n为一数字，用于连接时从文件 n中显示不匹配行。例如， -a1显示第一个文件的不匹配行，-a2为从第二个文件中显示不匹配行。

o n.m n为文件号， m为域号。1.3表示只显示文件1第三域，每个n，m必须用逗号分隔，如1.3，2.1。

j n m n为文件号， m为域号。使用其他域做连接域。

t 域分隔符。用来设置非空格或tab键的域分隔符。例如，指定冒号做域分隔符 -t：。

现有两个文本文件，其中一个包含名字和街道地址，称为 name.txt，另一个是名字和城镇，为town.txt。

```
$ pg names.txt
M.Golls 12 Hidd Rd
P.Heller The Acre
P.Willey 132 The Grove
T.Norms 84 Connaught Rd
K.Fletch 12 Woodlea
```

```
$ pg town.txt
M.Golls Norwich NRD
P.Willey Galashiels GDD
T.Norms Brandon BSL
K.Fletch Mildenhall MAF
```

连接两个文件

连接两个文件，使得名字支持详细地址。例如 M.Golls记录指出地址为12 Hidd Rd。连接域为域0 —— 名字域。因为两个文件此域相同， join将假定这是连接域：

```
$ join names.txt town.txt
M.Golls  12 Hidd Rd Norwich NRD
P.Willey 132 The Grove Galashiels GDD
T.Norms  84 Connaught Rd Brandon BSL
K.Fletch 12 Woodlea Mildenhall MAF
```

好，工作完成。缺省join删除或去除连接键的第二次重复出现，这里即为名字域。

1. 不匹配连接

如果一个文件与另一个文件没有匹配域时怎么办？这时 join不可以没有参数选项，经常指定两个文件的-a选项。下面的例子显示匹配及不匹配域。

```
$ join -a1  -a2 names.txt town.txt
M.Golls  12 Hidd Rd Norwich NRD
P.Heller The Acre
P.Willey 132 The Grove Galashiels GDD
T.Norms  84 Connaught Rd Brandon BSL
K.Fletch 12 Woodlea Mildenhall MAF
```

输出表明P.Heller不匹配第二个文件中任何一个记录。再运行这个命令，但指定只显示第一个文件中不匹配行：

```
$ join -a1 names.txt town.txt
```

2. 选择性连接

使用-o选项选择连接域。例如要创建一个文件仅包含人名及城镇， join执行时需要指定显示域。方式如下：

使用1.1显示第一个文件第一个域，2.2显示第二个文件第二个域，其间用逗号分隔。命令为：

```
$ join  -o 1.1,2.2 names.txt town.txt
M.Golls Norwich
P.Willey Galashiels
T.Norms Brandon
K.Fletch Mildenhall
```

要创建此新文件，将输出结果重定向到一个文件即可。

```
$ join  -o 1.1,2.2 names.txt town.txt >towns.txt
```

使用-jn m进行其他域连接，例如用文件1域3和文件域2做连接键，命令为：

```
join -j1 3 -j2 2 file1 file2
```

下面观察一个具体实例。有两个文件：

```
$ pg pers
P.Jones Office Runner   ID897
S.Round UNIX admin      ID666
L.Clip  Personl Chief   ID982

$ pg pers2
Dept2C  ID897   6 years
Dept3S  ID666   2 years
Dept5Z  ID982   1 year
```

文件pers包括名字、工作性质和个人ID号。文件pers2包括部门、个人ID号及工龄。连接应使用文件pers中域4，匹配文件pers2中域2，命令及结果如下：

```
$ join -j1 4 -j2 2 pers pers2
ID897 P.Jones Office Runner Dept2C 6 years
ID666 S.Round UNIX admin Dept3S    2 years
ID982 L.Clip Personl Chief Dept5Z  1 year
```

使用join应注意连接域到底是哪一个，比如说你认为正在访问域4，但实际上join应该访问域5，这样将不返回任何结果。如果是这样，用awk检查域号。例如，键入$ awk '{print $4}'文件名，观察其是否匹配假想域。

11.5 cut用法

cut用来从标准输入或文本文件中剪切列或域。剪切文本可以将之粘贴到一个文本文件。下一节将介绍粘贴用法。

cut一般格式为：

```
cut [options] file1 file2
```

下面介绍其可用选项：

-c list 指定剪切字符数。

-f field 指定剪切域数。

-d 指定与空格和tab键不同的域分隔符。

-c用来指定剪切范围，如下所示：

-c1，5-7 剪切第1个字符，然后是第5到第7个字符。

-c1-50 剪切前50个字符。

-f 格式与-c相同。

-f 1，5 剪切 第1域，第5域。

-f1，10-12 剪切第1域，第10域到第12域。

参照上一节中的文件‘ pers ’，现在从'pers'文件中剪切文本。使用冒号做其域分隔符。

```
$ pg pers
P.Jones:Office Runner:ID897
S.Round:UNIX admin:ID666
L.Clip:Personl Chief:ID982
```

11.5.1 使用域分隔符

文件中使用冒号“ :”为域分隔符，故可用-d选项指定冒号，如-d :。如果有意观察第3域，可以使用-f3。要抽取ID域。可使用命令如下：

```
$ cut -d: -f3 pers
ID897T
ID666
ID982
```

11.5.2 剪切指定域

cut命令中剪切各域需用逗号分隔，如剪切域1和3，即名字和ID号，可以使用：

```
$ cut -d: -f1,3 pers
P.Jones:ID897
S.Round:ID666
L.Clip:ID982
```

要从文件/etc/passwd中剪切注册名及缺省根目录，需抽取域1和域3：

```
$ cut -d: -f1,6 /etc/passwd
gopher:/usr/lib/gopher-data
ftp:/home/ftp
peter:/home/apps/peter
dave:/home/apps/dave
...
```

使用-c选项指定精确剪切数目。这种方法需确切知道开始及结束字符。通常我不用这种方法，除非在固定长度的域或文件名上。

当信息文件传送到本机时，查看部分文件名就可以识别文件来源。要得到这条信息需抽取文件名后三个字符。然后才决定将之存在哪个目录下。下面的例子显示文件名列表及相应cut命令：

```
2231DG
2232DP
2236DK

$ ls 223*|cut -c4-6
1DG
2DP
6DK
```

如果使用ls -l命令作部分输出，情况将不同。需使用 -c选项。

```
-rw-r--r--   1 dave   admin        56 Apr 26 20:40 tr2.txt
-rw-r--r--   1 dave   admin        71 Apr 26 21:20 trpro.txt
```

要剪切字符，须计算 ls -l列表中的字符数。如显示权限用 cut -c1-10。然而这种方法可能相当慢，因此需要使用其他工具将相应信息抽取出来。要剪切谁正在使用系统的用户信息，方法如下：

```
$ who -u| cut -c1-8
root
dave
```

```
peter
```

11.6 paste用法

cut用来从文本文件或标准输出中抽取数据列或者域，然后再用 paste可以将这些数据粘贴起来形成相关文件。粘贴两个不同来源的数据时，首先需将其分类，并确保两个文件行数相同。

paste将按行将不同文件行信息放在一行。缺省情况下， paste连接时，用空格或tab键分隔新行中不同文本，除非指定-d选项，它将成为域分隔符。

paste格式为；

```
paste -d -s -file1 file2
```

选项含义如下：

-d 指定不同于空格或tab键的域分隔符。例如用@分隔域，使用-d@。

-s 将每个文件合并成行而不是按行粘贴。

- 使用标准输入。例如ls -l |paste ， 意即只在一列上显示输出。

从前面的剪切中取得下述两个文件：

```
$ pg pas1
ID897
ID666
ID982


$ pg pas2
P.Jones
S.Round
L.Clip
```

基本paste命令将之粘贴成两列：

```
$ paste pas1 pas2
ID897   P.Jones
ID666   S.Round
ID982   L.Clip
```

11.6.1 指定列

通过交换文件名即可指定哪一列先粘：

```
$ paste pas2 pas1
P.Jones ID897
S.Round ID666
L.Clip  ID982
```

11.6.2 使用不同的域分隔符

要创建不同于空格或tab键的域分隔符，使用-d选项。下面的例子用冒号做域分隔符。

```
$ paste -d: pas2 pas1
P.Jones:ID897
S.Round:ID66
```

要合并两行，而不是按行粘贴，可以使用 -s选项。下面的例子中，第一行粘贴为名字，第二行是ID号。

```
$ paste -s pas2 pas1
P.Jones S.Round L.Clip
ID897   ID666   ID982
```

11.6.3 paste命令管道输入

paste命令还有一个很有用的选项（ -)。意即对每一个（ -)，从标准输入中读一次数据。使用空格作域分隔符，以一个4列格式显示目录列表。方法如下：

```
$ pwd
$ /etc
$ ls | paste -d" " - - - -
init.d rc rc.local rc.sysinit
rc0.d rc1.d rc2.d rc3.d
rc4.d rc5.d rc6.d
```

也可以以一列格式显示输出：

```
$ ls | paste -d"" -
init.d
rc
rc.local
rc.sysinit
rc0.d
rc1.d
...
```

11.7 split用法

split用来将大文件分割成小文件。有时文件越来越大，传送这些文件时，首先将其分割可能更容易。使用vi或其他工具诸如sort时，如果文件对于工作缓冲区太大，也会存在一些问题。因此有时没有选择余地，必须将文件分割成小的碎片。

split命令一般格式：

```
split -output_file-size input-filename output-filename
```

这里output-file-size指的是文本文件被分割的行数。split查看文件时，output-file-size选项指定将文件按每个最多1000行分割。如果有个文件有2800行，那么将分割成3个文件，分别有1000、1000、800行。每个文件格式为x[aa]到x[zz]，x为文件名首字母，[aa]、[zz]为文件名剩余部分顺序字符组合，下面的例子解释这一点。

假定文件bigone.txt有2800行，split命令产生下列文件：

```
$ split bigone.txt
xaa
xab
xac
```

文件大小为：

Size	Filename
1000	xaa
1000	xab
800	xac

可以使用output-file-size选项来分割文件。以下为一个6行文件。

```
$ pg split1
this is line1
```

```
this is line2
this is line3
this is line4
this is line5
this is line6
```

按每个文件2行分割，命令为：

```
$ split -2 split1
```

观察其结果。

```
$ ls -lt |head
total 205
-rw-r--r--   1 dave     admin          28 Apr 30 13:12 xaa
-rw-r--r--   1 dave     admin          28 Apr 30 13:12 xab
-rw-r--r--   1 dave     admin          28 Apr 30 13:12 xac
    ...
```

文件有6行，split按每个文件两行进行了分割，并按字母顺序命名文件。为进一步确信操作成功，观察一个新文件内容：

```
$ pg xac
this is line5
this is line6
```

11.8 小结

本章讲述了对文本文件进行基本的合并和分割处理的各种工具。诸如 sort、join、split、uniq、cut和paste，并附有大量实例。使用这些工具将使你事半功倍。现在如果遇到一个未处理文件，相信你已知道使用什么工具将数据转化为更有意义的信息。