

# Sort 命令

1. 功能说明: 将文本文件内容加以排序, sort 可针对文本文件的内容, 以行为单位来排序

2. 格式:

```
sort [选项] filename
```

3. sort 命令选项

- m** • -m 将已排序的输入文件, 合并为一个排序后的输出数据流。
- n** • -n 以整数类型比较字段
- o** • -o outfile 将输入写到指定文件, 而非标准输出。如果该文件为输入文件之一, 则sort 在进行排序写到输入文件之前, 会先将它复制到一个临时文件
- r** • -r 倒置排序的顺序为 由大至小 (descending), 而非默认的由小至大 (ascending)
- t** • -t char 使用单个字符char作为默认的字段分割字符, 取代默认的空白字符。
- u** • -u 只有唯一的记录, 丢弃所有具有相同键值的记录, 只留其中的第一条。只有键值字段是重要的, 也就是说: 被丢弃的记录其他部分可能是不同值。
  - 行为模式: sort 会读取指定的文件, 如果未给定文件, 则读取标准输入, 在将排序好的数据写至标准输出。
- b** • -b 忽略开头的空白
- c** • -c 检查输入是否已正确排序, 如输入未经排序, 但退出码 (exit code) 为非零值, 则不会有任何输出
- d** • -d 字典顺序: 仅文字数字与空白才有意义
- g** • -g 一般数值: 以浮点数字类型比较字段。这个选项的运作有点类似 -n, 差别仅在于这个选项的数字可能有小数点及指数。(仅GNU版本提供此功能)
- f** • -f 以不管字母大小写的方式排序
- i** • -i 忽略无法打印的字符
- k** • -k 定义排序键值字段 (该选项后接一个字段编号, 或则是一对数字。有时-k之后可用空白分隔。每个编号后都可以接一个点号的字符位置, 及/ 或 修饰符(modifier)字母之一。且当出现多个-k选项时候, 会先从第一个键值开始排序, 找出匹配该键值的记录后, 再进行第二个键值字段的排序, 以此类推。)

4. sort将文件的每一行作为一个单位, 相互比较, 比较原则是从首字符向后, 依次按ASCII码值进行比较, 最后将他们按升序输出

5. 实例

```
- cat seq          - sort seq
- banana           - apple
  apple            banana
  pear             orange
  orange            pear
```

sort的-u 选项它的作用很简单, 就是在输出行中去除重复行

```
- cat seq          - sort -u seq
- banana           - apple
  apple            banana
  pear             orange
  orange
```

apple	pear
pear	

sort的-n选项，按数值排序

- cat number	- sort number	- sort -n number
1	1	1
3	11	2
2	2	3
5	3	4
78	4	5
11	5	11
4	78	78

sort的-o选项，想把排序结果输出到原文件中，用重定向可不行

数值从大到小

```
[root@zhoucentos log]# cat number
78
5
4
3
2
11
-
[root@zhoucentos log]# sort -nr number -o number
[root@zhoucentos log]# cat number
78
11
5
4
3
2
1
-
```

```
[root@zhoucentos log]# cat date
2017-12-02
2017-01-09
2017-10-23
2017-04-24
```

这个文件有三列，列与列之间用“-”隔开了，第一列表示年，第二列表示月，第三列表示日。那么想以月来排序，也就是以第二列来排序，如何利用sort实现？  
方法：sort提供了-t选项，后面可以设定间隔符。指定了间隔符之后，就可以用-k来指定列数了。

① ② ③  
年 月 日

```
[root@zhoucentos log]# sort -n -k 2 -t '-' date
```

按数字排序

利用空格分隔符  
对第2列排序

```
jlgao:x:1116:100::/users5/jlgao:/bin/bash
yuzhouzhang:x:1117:100::/users8/yuzhouzhang:/bin/bash
yjtian:x:1118:100::/users8/yjtian:/bin/bash
sychen:x:1119:100::/users5/sychen:/bin/bash
```

- (1)-k2,4,5,6 指的是从第二个字段的第4个字符开始比较, 一直比到第五个字段的第六个字符。
- (2) sort -t: -k1,1 /etc/passwd 以冒号隔开的第一个字段: 用户名称 对 /etc/passwd进行排序
- (3)sort -t: -k3nr /etc/passwd 以冒号隔开的第3个字段 uid 反向 (由大到小) 排序
- (4)sort -t: -k4n -k3n /etc/passwd 以冒号隔开的第4个字段GID, 以及第3个字段uid排序,先以k4n排序, 在其基础上再以k3n排序

## • 2.从公司英文名称的第二个字母开始进行排序

```
[root@FDMdevBI opt]# sort -t ' ' -k1.2 testsort.txt
baidu 100 5000
sohu 100 4500
google 110 5000
guge 50 3000
[root@FDMdevBI opt]# sort -t ' ' -k1.3 testsort.txt
guge 50 3000
sohu 100 4500
baidu 100 5000
google 110 5000
```

→ 必须空格  
~  
→ 第1个的第3个字母

## • 3.按照公司人数排序, 人数相同的只保留一个

```
[root@FDMdevBI opt]# cat testsort.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
[root@FDMdevBI opt]# sort -n -k2 testsort.txt
guge 50 3000
baidu 100 5000
sohu 100 4500
google 110 5000
[root@FDMdevBI opt]# sort -n -k2 -u testsort.txt
guge 50 3000
baidu 100 5000
google 110 5000
```

## 4.让testsort.txt按照公司人数排序, 人数相同的按照员工平均工资降序排序

```
[root@FDMdevBI opt]# sort -n -t ' ' -k3r -k2 testsort.txt
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
```

→ 位置调换

1 1 . / . , 110 100 5000 100 4500

## Uniq 命令 <删除相邻的重复行, 只保留一行>

1. 功能: 删除相邻的重复行, 只保留一行

2. 格式: `uniq [选项] filename`

3. 命令

-c 在输出行前面加上每行在输入文件中出现的次数。

-d 仅显示重复行

-u 仅显示不重复的行

## Wc 命令 <统计行 字节 字数>

- 功能: 统计给定文件中的行数、字节数和字数。
- 字是指由空格字符区分的最大字符串。
- 格式: `wc [option] filename`
  - -c: 统计字节数, 使用的是UTF-8编码, 每个汉字占3个字节, 行尾回车占一个字节。
  - -l: 统计行数
  - -w: 统计字数
- 默认都统计, 输出顺序总是行数, 字数, 字节数
- `wc -cl test`与`wc -lc test`是一样的。
- 例 用cat命令建立一个文件, 然后后wc命令统计文件信息。

## Comm 命令 <比较> → 排好序

- 功能: 对两个已经排好需的文件进行比较, 若没有排序好, 则出错。
- 格式: `comm [option] file1 file2`
  - option: 1, 2, 3控制相应的列是否显示
  - -12: 表示1, 2列不显示, 默认都显示
- 输出3列:

- 第1列：仅在file1中出现的行
- 第2列：仅在file2中出现的行
- 第3列：两个文件都存在的行

■ 演示：a文件{1\n2\n3\n4\n5}, b文件{1\n3\n4\n5\n6}

\$comm a b

*diff 命令*

*<比较, 列出不同, 但不排序>*

1.

- 功能：逐行比较两个文本文件，列出其不同之处，不要求实现对文件进行排序。
- 格式：diff [option] file1 file2
- Option
  - -b：忽略行尾的空格，且字符串中的一个或多个空格都视为相等。
  - -c：采用上下文输出格式，提供三行上下文。
  - -C n：采用上下文输出格式，提供n行上下文。
  - -e：产生一个合法的ed脚本作为输出。
- 输出格式：如何将file1转变成file2
  - n1 a n3,n4：a表示增加
  - n1,n2 d n3：d表示删除
  - n1,n2 c n3,n4：c表示修改
  - n1,n2
  - -12：表示1, 2列不显示，默认都显示
- n1,n2是针对file1的，n3,n4是针对file2的
- 每一行后面跟随受到影响的若干行，以"<"开头的行属于file1，">"开头的行属于file2。
- 演示实例：

2. 实例

*file1*

```
file1.txt
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
int count=0;
printf("hello,world!\n");
}
```

*file2*

```
file2
#include <stdio.h>
#include <string.h>
void main()
{
int count;
char *s="hello,world";
printf("hello,world!\n");
}
```



**\$diff file1.txt file2.txt**

直接输出不同的地方

```
3d2
< #include <stdlib.h>
6c5,6
< int count=0;
---
> int count;
> char *s="hello,world";
```