# [Linux下文件的切割、合并](http://www.blogjava.net/bluebillow/archive/2008/11/11/239829.html)，三种方法

Linux下文件切割、合并 的 三种方法：

$ split + cat

$ dd + cat

$ knife + cat

## split+cat,简单,推荐。

**Linux下文件的切割与合并(split+cat)**   
当前目录下有sp4.exe文件容量129M ，我用64M的U盘把spe.exe考到另一台机器上。

这个U盘一次装不下129M ，只能将文件分割成多个小的部分，多次拷贝。

**文件切割：  
# split -b 60m sp4.exe partial //**文件切割,分配每个切割大小为60MByte。用-b或-C指定分块大小。-b,--bytes=SIZE,SIZE值为每一输出档案的大小，**单位为byte**。-C,--line-bytes=SIZE。

SIZE可加入单位: b 代表 512，k 代表 1K，m 代表1Meg。  
# ls //按partialaa、partialab、partialac的格式生成如下文件,生成的文件自动添加了后缀aa/ab/ac  
-rw-r--r--    1 root     root     62914560  5月 18 00:40 partialaa   
-rw-r--r--    1 root     root     62914560  5月 18 00:41 partialab   
-rw-r--r--    1 root     root      8576176  5月 18 00:41 partialac  //partialaa/partialab为60mbyte，剩下的给partialac。  
拷贝到另一台机器上，将多个部分**合并：**

**# cat partialaa partialab partialac > sp4.exe 或 cat partiala\* > sp4.exe //文件合并。**用jion也可以合并。

附注：

#split -b 60m sp4.exe //如果不指定输出名字，那么输出文件名将会是xaa,xab,…的形式。

#split -b 60m sp4.exe split//指定输出文件名为split，那么输出文件将会是splitaa,splitab…,

**文件1.Android开发.zip,大小为3.9G，下面对它进行切割、合并操作。**

$ ls -l 1.Android开发.zip

-rwx------+ 1 Administrators None 4196507906 Oct 13 22:41 1.Android开发.zip

**切割：**

**$ split -b 2048m 1.Android\_origin.zip partial //按2048M=2G大小切割。**

$ ls -l partiala\*

-rw-r--r--+ 1 lenovo None 2147483648 Oct 14 00:13 partialaa

-rw-r--r--+ 1 lenovo None 2049024258 Oct 14 00:14 partialab

**合并：**

**$ cat partialaa partialab > backup\_now.zip**

$ ls -l backup\_now.zip

-rw-r--r--+ 1 lenovo None 4196507906 Oct 14 00:19 backup\_now.zip

**split命令参考：总结,Linux命令.doc -> split,文件分割**

[**Linux下切割与合并文件的方法(split+cat)**](http://www.blogjava.net/bluebillow/archive/2008/11/11/239829.html)

切割合并文件在linux下用split和cat就可以完成。下面举些实例进行说明。

**1.文件切割**文件切割模式分为两种：文本文件 和 二进制模式。

**1.1文本模式**  
文本模式只适用于文本文件，**用这种模式切割后的每个文件都是可读的**。文本模式又分为两种：按文件大小切割 和 按文本行数切割。

**1.1.1按文件大小切割**

split -C 5k duanxin split//指定输出文件名为splitaa, splitab…，如果不指定输出名字，那么输出文件名将会是xaa,xab,…的形式。

将文本文件duanxin按每块最大5k的大小进行切割，不打碎行。  
**1.1.2 按文本行数切割**

split -l 100 duanxin split

每个分块100行，不考虑大小。日志分析时应该有用。  
**1.2 二进制模式**

split -b 5k duanxin split

每个分块大小都是5k(最后一个不保证)，基本不可读。任何类型文件都可以用这种切割模式。 **2.文件合并**

cat splitaa splitab >newduanxin / cat split\* >newduanxin

不管用什么方式切割，合并方法不变。  
**3.其它**split可以用-a选项指定输出文件名的长度。如  
split -l 100 -a 3 duanxin split  
则输出文件出类似于splitaaa,splitaab。不指定时默认为2。  
**用-b或-C指定分块大小**时，可用的单位有，b:512bytes, k:1Kbytes, m:1Megbytes.  
<http://www.blogjava.net/bluebillow/archive/2008/11/11/239829.html>

参考资料<http://herofit.blogspot.com/2006/10/linux_10.html>

找东西的时候发现网上都是讲这个的，就随手贴一下，据说分割还可以用DD命令。

我的需求用split好像是搞不定的，希望**从文本的指定行数进行分割**，最终找到一个方法，见下：

**sed -n '1001,2000p' ur-file >newfile**

随后就发现，sed超级有用，有兴趣的自己了解吧，附送一个介绍sed用法的文章：<http://blog.chinaunix.net/u/15010/showart_243352.html>

## dd+cat

**Linux文件切割、合并(dd+cat)**

将一个140kbyte的文件originFile分割成两部分，一部分为part1，大小为60kbyte，剩下的分给另一部分part2。

**文件切割**

# dd if=originFile bs=1024 count=60 skip=0 of=part1 //count\*bs=60\*1024byte=60kbyte

# dd if=originFile bs=1024 count=80 skip=60 of=part2 //skip=60,忽略file1已经切割的大小

第一部分，count\*bs=60\*1024byte=60kbyte,

第二部分，skip=60\*1024byte=60kbyte=第一部分的大小,剩下140kbyte-60kbyte=80kbyte=80\*1024byte。

**文件合并**

# cat part1 part2 > bakFile

**split命令参考：总结,Linux命令.doc -> dd,复制、转化、磁盘读写速度监测**

**文件切割**

将一个6.3G的压缩文件android\_linaro\_4.4.4\_r2.tar.bz2分割成两部分，一部分为3G，剩下的分给另一部分。

#dd if=android\_linaro\_4.4.4\_r2.tar.bz2 bs=1024 count=3145728 skip=0 of=android\_linaro\_4.4.4\_r2\_tar\_bz2\_part1

#dd if=android\_linaro\_4.4.4\_r2.tar.bz2 bs=1024 count=3670016 skip=3145728 of=android\_linaro\_4.4.4\_r2\_tar\_bz2\_part2

第一部分，count\*bs=3145728\*1024byte=3G,

第二部分，skip=3145728\*1024byte=3G,就是第一部分的大小,剩下6.3G-3G=3.3G=3670016\*1024byte。

**文件合并**

# cat android\_linaro\_4.4.4\_r2\_tar\_bz2\_part1 android\_linaro\_4.4.4\_r2\_tar\_bz2\_part2 > the\_origin\_file.tar.bz2

**dd的作用是转换和拷贝文件，我们可以利用它来分割文件，相关的选项如下：**

**if=filename：输入的文件名**

**of=finename：输出的文件名**

**bs=bytes：一次读写的字节数(block size)，默认是512bytes。(以字节byte为单位)**

**skip=blocks : 拷贝前，跳过的输入文件的前blocks块，块的大小由bs决定**

**count=blocks：只拷贝输入文件的前blocks块**

例如，现在有一个文件file，大小为116616字节：

1. [root]# du -b file
2. 116616  file   //**116616byte**

将其分割为两文件file1和file2，那我们就设置每块为1024字节，将file的前60块放入file1，余下的放入file2：

1. [root]# dd if=file bs=1024 count=60 skip=0  of=file1 //file1大小为：count\*bs=60\*1024byte=60kbyte
2. [root]# dd if=file bs=1024 count=80 skip=60 of=file2 //file2，skip=60，忽略掉file1已经切割的大小。file2大小为：file大小-file1大小。

然后用cat将两个文件合并为file.bak，要注意文件的顺序：

1. [root]# cat file1 file2 > file.bak

可以用md5sum验证一下file和file.bak：

1. [root]# md5sum file
2. 3ff53f7c30421ace632eefff36148a70  file
3. [root]# md5sum file.bak
4. 3ff53f7c30421ace632eefff36148a70  file.bak

可以证明两个文件时完全相同的。

为了方便分割、合并文件，我写了两个脚本：

**ddf.sh：**

1. #ddf.sh：分割文件，分割后的文件以数字结尾，例如file分割为两个文件：file1和file2
2. #!/bin/sh
4. #使用脚本是第一参数是要分割的文件名
5. Filename=$1
6. Filesize=0
8. Path=`pwd`
10. #验证文件名是否正确，然后计算文件的大小
11. **if** [ -z $Filename ];then
12. echo "Error:The file name can not be empty"
13. exit
14. fi
15. **if** [ -e $Filename ];then
16. Filesize=`du -b $Filename | awk '{print $1}'`
17. **if** [ $Filesize == 0 ];then
18. echo "Error:The File size is zero!"
19. exit
20. fi
21. echo "The file size is $Filesize Byte"
22. echo "Plese enter the subfile size(KB):"
23. **else**
24. echo "Error:$Filename does not exist!"
25. exit
26. fi
28. #输入分割后每个文件的大小，单位是KB
29. read Subfilesize
30. **if** [ -z $Subfilesize ];then
31. echo "Error:Input can not be empty"
32. exit
33. fi
35. echo $Subfilesize | grep '^[0-9]\+$' >> /dev/null
36. **if** [ $? -ne 0 ];then
37. echo "Error:The Input is not a number!"
38. exit
39. **elif** [ $Subfilesize -eq 0 ];then
40. echo "Error:The Subfile size is zero!"
41. exit
42. fi
44. #计算需要分割为几个文件
45. SubfileByte=`expr $Subfilesize \\* 1024`
46. Subfilenum=`expr $Filesize / $SubfileByte`
47. **if** [ `expr $Filesize % $Subfilesize` -ne 0 ];then
48. Subfilenum=`expr $Subfilenum + 1`
49. fi
51. #将文件分割
52. echo "$Filename will be divided into $Subfilenum"
53. i=1
54. skipnum=0
55. **while** [ $i -le $Subfilenum ]
56. do
57. echo "$Filename$i"
58. dd **if**=$Filename of="$Path/$Filename$i" bs=1024 count=$Subfilesize skip=$skipnum
59. i=`expr $i + 1`
60. skipnum=`expr $skipnum + $Subfilesize`
61. done
62. echo "$Filename has been divided into $Subfilenum"
63. echo "Done !"

**caf.sh：**

1. #caf.sh:合并文件，需要合并的文件要放在一个文件夹里
2. #       文件名分为两个部分，第一部分都相同，第二部分必须是从1开始的连续数字，例如file1，file2，file3
3. #       合并后的文件名为file.bak
4. #!/bin/sh
6. #输入文件名的第一部分
7. echo "Please enter file name:"
8. read Filename
9. **if** [ -z $Filename ];then
10. echo "Error:File name can not be empty"
11. exit
12. fi
14. #输入待合并文件的个数
15. echo "Please enter the number of subfiles:"
16. read Subfilenum
17. **if** [ -z $Subfilenum ];then
18. echo "Error:The number of subfiles can not be empty"
19. exit
20. fi
21. echo $Subfilenum | grep '^[0-9]\+$' > /dev/null
22. **if** [ $? -ne 0 ];then
23. echo "Error:Input must be a number"
24. exit
25. fi
26. **if** [ $Subfilenum -eq 0 ];then
27. echo "Error:The number of subfiles can not be zero"
28. exit
29. fi

32. #合并文件
33. i=1
34. Newfile=$Filename\.bak
35. **while** [ $i -le $Subfilenum ]
36. do
37. Subfilename=$Filename$i
38. **if** [ -e $Subfilename ];then
39. echo "$Subfilename done!"
40. cat $Subfilename >> $Newfile
41. i=`expr $i + 1`
42. **else**
43. echo "Error:$Subfilename does not exist"
44. rm -rf $Newfile
45. exit
46. fi
47. done

50. echo "Subfiles be merged into $Newfile"
51. echo "Success!"

用这两个脚本完成对file的分割、合并：

1. [root]# ./ddf.sh file
2. The file size **is** 116616 Byte
3. Plese enter the subfile size(KB):
4. 60
5. file will be divided into 2
6. file1
7. 记录了60+0 的读入
8. 记录了60+0 的写出
9. 61440字节(61 kB)已复制，0.0352612 秒，1.7 MB/秒
10. file2
11. 记录了53+1 的读入
12. 记录了53+1 的写出
13. 55176字节(55 kB)已复制，0.0316272 秒，1.7 MB/秒
14. file has been divided into 2
15. Done !
16. [root]# ls
17. caf.sh  ddf.sh  file  file1  file2
18. [root]# ./caf.sh
19. Please enter file name:
20. file
21. Please enter the number of subfiles:
22. 2
23. file1 done!
24. file2 done!
25. Subfiles be merged into file.bak
26. Success!
27. [root]# ls
28. caf.sh  ddf.sh  file  file1  file2  file.bak

## 总结和技巧

**字节为最小存储单位，byte。**

**ls/split/dd等linux命令，使用的数据单位都是字节(byte)。**

$ ls -l | grep 1.Android

-rwx------+ 1 Administrators None **4196507906** Oct 13 22:41 1.Android开发.zip

**4196507906指的是字节(bytes)。**

技巧，dd命令中，**bs/count/skip参数的计算**

参考：shell总结，$[] 整数计算

$ echo 2+3 **不能达到计算的目的**

$ echo $[2+3] **用于+-\*/%**

$ echo $[4196507906-2147483648]

2049024258

$ echo $[(4196507906-2147483648)/1024] # **$[]，中括号中写计算表达式**。

2001000

浮点数用 $ echo "scale=3; 10/3" | bc -l (bc用于计算的)

在我的使用中，dd命令分割windows下的zip文件时，合并后的文件损坏。split命令却可以很好的工作。

是不是dd命令只能处理linux格式的文件？

split命令切割，切割后的每块大小一样(最后一块大小可能不一样，最后一块的大小为剩余大小)。使用**简单方便，推荐。**

dd命令切割，可以指定切割的每一块的大小。使用的时候要进行大小计算，比较麻烦。

## knife+cat

knife工具到http://see.online.sh.cn/ch/sw/dl/split.htm下载 ，解压后   
[root@rhas3 knife-2.0.1]# ls   
knife  link  linkfiles.pl  sp4.exe   
**分割：**[root@rhas3 knife-2.0.1]# ./knife -c sp4.exe 500   ( 500：字节大小)   
Cutting......Cutted Over :-)   
[root@rhas3 knife-2.0.1]# ls -l   
总用量 44   
-rwxr-xr-x    1 root     root        12949 2000-02-28  knife   
-rwxr-xr-x    1 root     root           17 2000-02-28  link   
-rwxr-xr-x    1 root     root         1088 2000-02-28  linkfiles.pl   
-rw-r--r--    1 root     root         1543 2000-02-28  sp4.exe   
-rw-r--r--    1 root     root          500  5月 18 00:49 sp4.exe.k00   
-rw-r--r--    1 root     root          500  5月 18 00:49 sp4.exe.k01   
-rw-r--r--    1 root     root          500  5月 18 00:49 sp4.exe.k02   
-rw-r--r--    1 root     root           43  5月 18 00:49 sp4.exe.k03   
  
**合并：**[root@rhas3 knife-2.0.1]# ./link sp4.exe.k00 sp4.exe.k01 sp4.exe.k02 sp4.exe.k03                                                                                    
Please intput the file name you want to LINK up, one by one.   
Just input OK when you have finished inputting.(一定要全输完在ok)   
ok   
In this order:   
    
Now it's time you type the name of the output file.   
    
-->;sp4.exe   
[root@rhas3 knife-2.0.1]#   
OK!