[GNU Awk 4.0: Teaching an Old Bird Some New Tricks](http://blog.chinaunix.net/uid-14293861-id-2977155.html)

简单翻译:  
  
gawk 4.0中加入的新特性:  
  
1.现在,gawk提供了引用外部脚本文件的机制.@include "文件名"这样的语句可以让gawk把该文件包含进来.和命令行的-f 参数类似,gawk会在当前目录寻找该文件.嵌套引用也是支持的,而且gawk不会引用相同的脚本两次.同时,伴随gawk很多年的igawk脚本将会被废弃  
  
2.两个新的模式BEGINFILE和ENDFILE被添加进来,BEGINFILE模式匹配的行为在每个文件的第一个记录被读取之前运行,如果一个文件不存在或者不能被打开,gawk会因致命错误退出程序,如果在BEGINFILE中添加检测语句,发现文件打不开时,执行nextfile进入到下一个文件,则程序会继续运行.ENDFILE行为可以在每个文件读取完之后做一些善后工作.  
  
3.你可以通过一个字符串变量间接的调用一个函数,例如  
  
function foo(a,b){ .... }  
function bar(a,b){ .... }  
BEGIN{  
fun="foo";@fun(1, 2) #间接调用foo(1,2)  
fun="bar";@fun(1, 2) #间接调用bar(1,2)  
}  
  
4.gawk已经支持真正的多位数组a[x][y],而不是以前通过不可见字符连接多个索引来模拟多维数组.并且和C以及其他编译型语言不同,awk的多维数组不要求是矩形的.  
  
5.默认支持了switch/case语句.gawk很久以前就有了switch语句,不过除非在编译的时候指定选项,否则是不可用的.  
  
6.通过FPAT匹配字段内容确定字段而不仅仅是通过FS匹配字段间隔来确定字段.前者类似于match函数匹配$0得到的数组,后则类似于split函数分割$0得到数组  
  
7.gawk网络编程中支持ipv6  
  
8.对应每个长选项,都一个短选项与之对应.  
  
9.默认激活范围表达式:{1,9},由于为了兼容其他awk版本.gawk 4.0以前只有在指定--posix参数或者--re-interval参数时,范围表达式才可以使用  
  
  
接下来将要进行的工作:  
  
几个重要的特性将会加入到gawk中.  
  
1.XMLGawk项目是基于gawk3.1.6产生的gawk分支版本.它提供更好的方式去加载awk动态扩展库以及更好的运行他们,这些都将会分别加入到gawk以及发行版中.  
  
2.尽管gawk在很多年前就有了加载动态扩展库的能力,但是该API一直没有趋于稳定并且不易使用,我已经重新设计了一套API使得在awk程序中可以调用c语言写成的函数.下一步我将会发布这个API  
  
3.目前gawk发行版中包含三个独立的可执行文件,普通的gawk,分析awk程序的pgawk,以及调试awk程序的dgawk.新的gawk核心使得将这三个程序合成为一个并且能通过参数分别运行他们成为可能.这将会简化编译步骤并且简化安装过程.  
  
4.更新文档中过时的内容以及程序例子.

大家可以到GNU的ftp上下载下来爽一爽， <ftp://ftp.gnu.org/gnu/gawk>，粗略的看了下介绍，新版本的gawk功能更强大了！！！  
下面是4.0.0版本gawk的一些新的features（测试了下部分功能）：  
<http://lists.gnu.org/archive/html/info-gnu/2011-06/msg00013.html>

Copyright (C) 2010, 2011 Free Software Foundation, Inc.  
  
   Copying and distribution of this file, with or without modification,  
   are permitted in any medium without royalty provided the copyright  
   notice and this notice are preserved.  
  
Changes from 3.1.8 to 4.0.0  
---------------------------  
  
1. The special files /dev/pid, /dev/ppid, /dev/pgrpid and /dev/user are  
   now completely gone. Use PROCINFO instead.  
  
2. The POSIX 2008 behavior for `sub' and `gsub' are now the default.  
   THIS CHANGES BEHAVIOR!!!!

1. echo '11122211' |awk '{sub(/1{3}/,"")}1'
2. 22211

复制代码

3. The \s and \S escape sequences are now recognized in regular expressions.

1. echo '111 222  11' |awk '{gsub(/\s/,"")}1'
2. 11122211

复制代码

4. The split() function accepts an optional fourth argument which is an array  
   to hold the values of the separators.

1. echo '111-222|33' |awk '{split($0,a,/[-|]/,seps);print "a[1] = "a[1] RS "a[2] = "a[2] RS "a[3] = "a[3] RS "spes[1] = "seps[1] RS "speS[2] = "seps[2]}'
2. a[1] = 111
3. a[2] = 222
4. a[3] = 33
5. spes[1] = -
6. speS[2] = |

复制代码

5. New -b / --characters-as-bytes option that means "hands off my data"; gawk  
   won't try to treat input as a multibyte string.  
  
6. New --sandbox option; see the doc.

1. --sandbox
2. Disable the system() function, input redirections with getline, output redirections with print and printf, and dynamic extensions. This is particularly useful when you want to run awk scripts from questionable sources and need to make sure the scripts can't access your system (other than the specified input data file).

复制代码

7. Indirect function calls are now available.

1. --With indirect function calls, you tell gawk to use the value of a variable as the name of the function to call.

复制代码

8. Interval expressions are now part of default regular expressions for  
   GNU Awk syntax.  
  
9. --gen-po is now correctly named --gen-pot.  
  
10. switch / case is now enabled by default. There's no longer a need  
    for a configure-time option.

1. --Control flow in the switch statement works as it does in C.
2. seq 10 |awk '{switch ($0%2){
3. case "0":
4. print "even number: "$0;break
5. default:
6. print "odd number: "$0
7. }
8. }'
9. odd number: 1
10. even number: 2
11. odd number: 3
12. even number: 4
13. odd number: 5
14. even number: 6
15. odd number: 7
16. even number: 8
17. odd number: 9
18. even number: 10

复制代码

11. Gawk now supports BEGINFILE and ENDFILE. See the doc for details.  
  
--The body of the BEGINFILE rules is executed just before gawk reads the first record from a file. FILENAME is set to the name of the current file, and FNR is set to zero.  
--The ENDFILE rule is called when gawk has finished processing the last record in an input file. For the last input file, it will be called before any END rules. (这两个功能真的很酷,尤其是在处理多个文件时，如下面：)

1. head f1 f2
2. ==> f1 <==
3. aaa
4. bbb
5. ccc
6. ==> f2 <==
7. aaa
8. bbb
9. ccc
10. awk 'BEGIN{print"BEGIN: ---"}BEGINFILE{print "\nBEGINFILE: +++"}{print}ENDFILE{print"ENDFILE: +++\n"}END{print"END: ---"}' f1 f2
11. BEGIN: ---
12. BEGINFILE: +++
13. aaa
14. bbb
15. ccc
16. ENDFILE: +++
17. BEGINFILE: +++
18. aaa
19. bbb
20. ccc
21. ENDFILE: +++
22. END: ---

复制代码

12. Directories named on the command line now produce a warning, not  
    a fatal error, unless --posix or --traditional.  
  
13. The new FPAT variable allows you to specify a regexp that matches  
    the fields, instead of matching the field separator. The new patsplit()  
    function gives the same capability for splitting.  
  
--The value of FPAT should be a string that provides a regular expression. This regular expression describes the contents of each field.

1. echo '111-222|33' |awk -vFS="[-|]" '{print "$1 = "$1 RS "$2 = "$2 RS "$3 = "$3}'
2. $1 = 111
3. $2 = 222
4. $3 = 33
5. #如果用FPAT呢？
6. echo '111-222|33' |awk -vFPAT="[^-|]+" '{print "$1 = "$1 RS "$2 = "$2 RS "$3 = "$3}'
7. $1 = 111
8. $2 = 222
9. $3 = 33

复制代码

14. All long options now have short options, for use in `#!' scripts.  
  
15. Support for IPv6 added via /inet6/... special file. /inet4/... forces  
    IPv4 and /inet chooses the system default (probably IPv4).  
  
16. Added a warning for /[:space:]/ that should be /[[:space:]]/.  
  
17. Merged with John Haque's byte code internals. Adds dgawk debugger and  
    possibly improved performance.  
  
18. `break' and `continue' are no longer valid outside a loop, even with  
    --traditional.  
  
19. POSIX character classes work with --traditional (BWK awk supports them).  
  
20. Nuked redundant --compat, --copyleft, and --usage long options.  
  
21. Arrays of arrays added. See the doc. (这个更强！)

1. awk 'BEGIN{arr["a"]["b"]=1;arr["a"]["c"]=2;
2. for( i in arr)
3. for( j in arr[i])
4. print i,j,arr[i][j]
5. }'
6. a b 1
7. a c 2

复制代码

22. Per the GNU Coding Standards, dynamic extensions must now define  
    a global symbol indicating that they are GPL-compatible. See  
    the documentation and example extensions.  
    THIS CHANGES BEHAVIOR!!!!  
  
23. In POSIX mode, string comparisons use strcoll/wcscoll.  
    THIS CHANGES BEHAVIOR!!!!  
  
24. The option for raw sockets was removed, since it was never implemented.  
  
25. If not in POSIX mode, gawk turns ranges of the form [d-h] into  
    [defgh] before compiling a regexp.  Maybe this will stop all the  
    questions about [a-z] matching uppercase letters.  
    THIS CHANGES BEHAVIOR!!!!  
  
26. PROCINFO["strftime"] now holds the default format for strftime().  
  
27. Updated to latest infrastructure: Autoconf 2.68, Automake 1.11.1,  
    Gettext 0.18.1, Bison 2.5.  
  
28. Many code cleanups. Removed code for many old, unsupported systems:  
        - Atari  
        - Amiga  
        - BeOS  
        - Cray  
        - MIPS RiscOS  
        - MS-DOS with Microsoft Compiler  
        - MS-Windows with Microsoft Compiler  
        - NeXT  
        - SunOS 3.x, Sun 386 (Road Runner)  
        - Tandem (non-POSIX)  
        - Prestandard VAX C compiler for VAX/VMS  
        - Probably others that I've forgotten  
  
29. If PROCINFO["sorted\_in"] exists, for(iggy in foo) loops sort the  
    indices before looping over them.  The value of this element  
    provides control over how the indices are sorted before the loop  
    traversal starts. See the manual.  
  
30. A new isarray() function exists to distinguish if an item is an array  
    or not, to make it possible to traverse multidimensional arrays.  
  
31. asort() and asorti() take a third argument specifying how to sort.  
    See the doc.