A+

awk从放弃到入门（3）：awk变量

在本博客中，AWK是一个系列文章，本人会尽量以通俗易懂的方式递进的总结awk命令的相关知识点。

awk系列博文直达链接：AWK命令总结之从放弃到入门（通俗易懂，快进来看）

在阅读这篇文章之前，最好先阅读之前的文章，以之前的知识点作为基础，再看这篇文章会容易理解很多。

之前的文章在使用到"输入分隔符"和"输出分隔符"的时候，我们都提到了一个名词："变量"。

这篇文章我们就来详细的总结一下awk中的变量，我们会先对概念进行描述，如果概念中有不明白的地方，不要着急，对应其示例，你自然就会明白。

对于awk来说"变量"又分为"内置变量" 和 "自定义变量"，"输入分隔符FS"和"输出分隔符OFS"都属于内置变量。

内置变量就是awk预定义好的、内置在awk内部的变量，而自定义变量就是用户定义的变量。

我们先看看awk常用的一些内置变量，此处先大致列出其概念，只看概念并不容易理解其意思，不懂没关系，等到示例时你自然会明白。

awk常用的内置变量以及其作用如下

FS：输入字段分隔符， 默认为空白字符

OFS：输出字段分隔符， 默认为空白字符

RS：输入记录分隔符(输入换行符)， 指定输入时的换行符

ORS：输出记录分隔符（输出换行符），输出时用指定符号代替换行符

NF：number of Field，当前行的字段的个数(即当前行被分割成了几列)，字段数量

NR：行号，当前处理的文本行的行号。

FNR：各文件分别计数的行号

FILENAME：当前文件名

ARGC：命令行参数的个数

ARGV：数组，保存的是命令行所给定的各参数

上面描述到的"输入字段分隔符FS和输出字段分隔符OFS在之前的文章中已经解释过了，字段数量NF也大致说了。

RS、ORS、NR、FNR、FILENAME、ARGC、ARGV这些术语对于我们来说是新接触的，但是触类旁通，RS其实与FS类似，ORS与OFS类似，FS是字段输入分隔符分隔符，OFS是字段输出分隔符，ORS是行输出分隔符，它们的原理都很相似。不要着急，我们来慢慢解释

内置变量

内置变量NR

NR比较简单，我们先看NR的例子。

首先，如下图所示，test1文件中一共有两行文本，使用空格隔开，第1行有4列，第2行有5列

而内置变量NR表示每一行的行号，内置变量NF表示每一行中一共有几列，那么，也就是说，我们可以通过下例中的方法，得到test1文本中，每一行的行号以及每一数量。

或者，利用NR内置变量，先打印出行号，再打印出整行的内容，相当于为test1中的每一行都添加了行号以后再进行输出，示例如下。

好了，现在每一行的开头都有行号了，简单吧。

细心如你一定注意到了一个细节，就是在打印 $0，$1，$2 这些内置变量的时候，都有使用到"$"符号，但是在调用 NR，NF 这些内置变量的时候，就没有使用"$"，习惯，那么可能是因为你已经习惯了使用bash的语法去使用变量，在bash中，我们在引用变量时，都会使用$符进行引用，但是在awk中，只有在引用$0、$1等内候才会用到"$",引用其他变量时，不管是内置变量，还是自定义变量，都不使用"$",而是直接使用变量名。

内置变量FNR

FNR这个内置变量是什么意思呢？我们一起来看看。

当我们使用awk同时处理多个文件，并且使用NR显示行号的时候，效果如下图。

从返回结果可以看出，awk处理多个文件的时候，如果使用NR显示行号，那么，多个文件的所有行会按照顺序进行排序。

可是，如果我们想要分别显示两个文件的行号，该怎么办呢，这个时候就会用到内置变量FNR，效果如下。

我想，对比完上述两个示例，你肯定明白了FNR内置变量的作用，没错，它的作用就是当awk处理多个文件时，分别对每个文件的行数进行计数。

内置变量RS

现在，我们来看看RS这个变量，我们说了，RS是输入行分隔符，如果不指定，默认的"行分隔符"就是我们所理解的"回车换行"。

假设，我们不想以默认的"回车换行"作为"行分隔符"，而是想使用空格作为所谓的行分隔符，也就是说，我们想让awk认为，每遇到一个空格，就换行，换句话说，以为每次遇到一个空格就是新的一行。那么我们该怎么做呢？示例如下。

如上图所示，我们先使用了默认的"回车换行"作为"行分隔符"输出了test1文本，这时显示文本一共有2行。

而后来，我们又指定了使用"空格"作为"行分隔符"输出test1文本，这时显示文本一共有8行。

看到了吗？当我们指定使用空格作为"行分隔符"时，在awk解析文本时，每当遇到空格，awk就认为遇到的空格是换行符，于是awk就将文本换行了，而此时人类理
行"，对于awk来说并不是所谓的换行符，所以才会出现上图中第4行的现象，即使从人类的角度去看是两行文本，但是在awk的世界观里，它就是一行。

如果你还是没有理解，那么我们换个方式描述，再来啰嗦一遍。

默认情况下，awk使用"回车换行"作为"行分隔符(换行符)"，此时，人类的世界观与awk的世界观是一致的，因为我们和awk都认为，遇到回车换行，就表示当前行结
一行。

而当我们指定了特定的"行分隔符"时，比如空格，那么当awk遇到空格时，就认为当前行结束了，新的一行开始了，此时，awk的世界观与人类的世界观已经不同，为"回车换行"才是新的一行，awk却认为"回车换行"并不是新的一行的开始，所以，从上图中返回的信息中，我们可以看到，人类所以为的"两行"，共用了一个行号们就是第4行。

这就是输入行分隔符的使用方法。同理，我们来看看"输出行分隔符"，理解输出行分隔符之前，请做好心理准备，最好不要以正常的思维去理解换行，才能比较容易行分隔符。

内置变量ORS

在理解"输出行分隔符"ORS之前，请先理解刚才描述的"输入行分隔符"RS，否则理解起来可能比较困难。

默认情况下，awk将人类眼中的"回车换行"，当做"输出行分隔符"，此时，awk的"世界观"与人类的"世界观"是相同的。

现在，我们改变一下awk的想法，我们让awk认为，"+++"才是真正的输出行分隔符，示例如下图

看懂了吗，我们再啰嗦的解释一遍，在没有指定输出行分隔符之前，awk跟人类的逻辑思维是一样一样的，当人类想要换行的时候，就会"另起一行"（回车换行），的，当它在输出文字的时候，如果想要换行，就会"另起一行"（回车换行），可是，如果我们指定了"输出行分隔符"为"+++",那么，当awk在输出文字的时候，如果会"另起一行"（+++），所以，对于awk来说，它完成了"另起一行"的动作，只不过，它所认为的"另起一行"的动作就是输出"+++",而不再是原来的输出" 回车换行人类看到的"表象上"，awk并没有换行，那是因为我们还是以"回车换行"作为换行的标准，而awk已经变了，它认为,"+++"就是换行的标准。

这次明白了吧，真的明白了吗？

我们把刚才学到的"输入换行符"和"输出换行符"同时使用，看看是什么效果，示例如下。

如果你能明白awk为什么会将test1的文本输出成上图中的模样，那么你已经彻底理解了RS与ORS两个内置变量。

如果你又懵逼了，那么，从RS内置变量开始，再看一遍吧。

内置变量FILENAME

FILENAME这个内置变量，从字面上，就能看出是什么意思，没错，就是显示文件名，演示效果如下。

内置变量ARGC与ARGV

ARGC内置变量表示命令行参数的个数，什么意思呢？我们先不解释ARGC，先看看ARGV是什么。

别眼花了。

一个是ARGC，

一个是ARGV，

先说说ARGV。

ARGV内置变量表示的是一个数组，这个数组中保存的是命令行所给定的参数。这样解释还是很模糊，不容易理解，我们来看看示例。

上图中，我们先使用BEGIN模式，输出一个字符串"aaa"，然后，传入两个文件的文件名作为参数，我们发现，BEGIN模式正常执行了打印操作，输出了"aaa"字符串
样的命令，同样使用BEGIN模式，只不过，这次不只打印"aaa",还打印ARGV这个数组中的第二个元素的值。

我说已经说过，ARGV内置变量表示的是一个数组，既然是数组，就需要用上图中的下标的方式，引用对应元素的值，因为数组的索引都是从0开始的，所以，ARG
RGV数组中的第二个元素的值，从返回结果可以看出，ARGV[1]对应的值为test1，同理，我们又使用第三条命令，多打印了一个ARGV[2]的值，发现ARGV[2]对应
这个时候，你明白ARGV内置变量的含义了吗，说白了，ARGV内置变量表示的是：所有参数组成的数组。那么细心的你一定会问了，ARGV[0]对应的是哪个参数呢
一下。

我擦，第一个参数竟然是awk这个命令本身？？太神奇了，有没有很出乎意料…

好吧，awk就是这么规定的，'pattern{ action }'并不被看做是参数，awk被看做为参数。

好了，说明了ARGV变量以后，再说ARGC变量的作用，就容易多了。

在刚才的例子中，应该有三个参数，awk、test1、test2，这三个参数作为数组的元素存放于ARGV中，现在，而ARGC则表示参数的数量，也可以理解为ARGV数组
如下

自定义变量

好了，内置变量解释完了，现在我们来看看自定义变量，自定义变量，顾名思义，就是用户定义的变量，有两种方法可以自定义变量。

方法一：-v varname=value 变量名区分字符大小写。

方法二：在program中直接定义。

我们来看一些小例子，即可明白上述两种方法。

通过方法一自定义变量。

这种方式，与设置内置变量的值的方法是一样的。

使用方法二自定义变量，直接在program中定义即可，但是注意，变量定义与动作之间需要用分号";"隔开。

当然，我们也可以一次性定义多个变量

第一种方法虽然看上去比较麻烦，但是这种方法也有自己的优势

当我们需要在awk中引用shell中的变量的时候，则可以通过方法一间接的引用。举例如下

好了，awk中变量的使用方法就暂时总结到这里，希望这篇文章会对你有所帮助。