awk 不一样的分隔符 - 空格分隔符

今天用 awk 格式化字符串的时候，发现了一个奇怪现象，查看了 awk 手册后，特以此文记录。

示例文本内容

后文所有 awk 语名中出现的 file.txt 内容均如下：

```
# cat -A file.txt
     1^Iroot:x:0:0:root:/root:/bin/bash$
     2^Ibin:x:1:1:bin:/bin:/sbin/nologin$
     3^Idaemon:x:2:2:daemon:/sbin:/sbin/nologin$
```

登录后复制

- 1.
- 2.
- 3.
- 4.

现象描述

通过 awk -F 的 “[]” 指定多个分隔符（包含空格）的时候，连续的空格被分隔成了多个字段。

awk 默认以空白字符（包含空格、TAB 字符、换行符）做为分隔符，为了更直观对比，此处示例直接通过 -F 参数指定。简单示例对比下：

我们先指定空格做为分隔符来获取第二个字段

```
# awk -F " " '{print NF, $2}' file.txt
2 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
2 daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

再通过 [] 指定空格分隔符来获取

```
# awk -F "[ ]" '{print NF, $6}' file.txt
6 1     root:x:0:0:root:/root:/bin/bash
6 2     bin:x:1:1:bin:/bin:/sbin/nologin
6 3     daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

是不是好奇怪，我们通过 -F " " 做为分隔符的时候，每行只有 2 个字段，而通过 -F "[ ]" 做分隔符的时候，每行共有 6 个字段。$1-$5 获取的值为空，而 $6 确打印了全部内容。

查看 awk 手册：

4.5.1 Whitespace Normally Separates Fields

awk interpreted this value in the usual way, each space character would separate fields, so two spaces in a row would make an empty field between them. The reason this does not happen is that a single space as the value of FS is a special case—it is taken to specify the default manner of delimiting fields. If FS is any other single character, such as ",", then each occurrence of that character separates two fields. Two consecutive occurrences delimit an empty field. If the character occurs at the beginning or the end of the line, that too delimits an empty field. The space character is the only single character that does not follow these rules.

4.5.2 Using Regular Expressions to Separate Fields

There is an important difference between the two cases of ‘FS = " "’ (a single space) and ‘FS = "[\t\n]+"’ (a regular expression matching one or more spaces, TABs, or newlines). For both values of FS, fields are separated by runs (multiple adjacent occurrences) of spaces, TABs, and/or newlines. However, when the value of FS is " ", awk first strips leading and trailing whitespace from the record and then decides where the fields are.

awk 手册

这两段内容刚好解释了这个奇怪的现象。大概意思就是：

- 行中的连续空格不会分隔空字段。当 FS 的值为 " " 时，awk 首先从记录中去除行首和行尾的空白，然后再分割字段。
- 如果 FS 是其他字符，比如”，“，连续两次出现将分隔一个空字段。如果字符出现在行首或行尾，也会分隔空字段。空格字符做为默认分隔符，是唯一不遵守这些规则的字符。
- 如果通过 -F "[ ]" 指定，执表示通过单个空格分隔，此时，将失去其做为默认分隔符的特性，与其它字符一样，遵守同样的分隔规则。

总结

结合上面内容，我们再来看几个示例，对今天的内容做个总结。

示例：

```
[root@nginx01 ~]# awk '{print $1}' file.txt
1
2
3
[root@nginx01 ~]# awk -F " " '{print $1}' file.txt
1
2
3
[root@nginx01 ~]# awk -F "[:\t]+" '{print $1}' file.txt
     1
     2
     3
[root@nginx01 ~]# awk -F "[ :\t]+" '{print $2}' file.txt
1
2
3
```


总结：

- 示例一，没有指定分隔符，用的默认分隔符，此时行首的连续空白字符被自动去除。
- 示例二，指定分隔符为空格，等价于默认分隔符。
- 示例三，指定分隔符为一个或多个连续的“冒号或 tab 键“，此时行首多个连续空白字符被一起计入第一个字段。
- 示例四，指定分隔符为一个或多个连续的”空白字符或冒号或 tab 键“，此时行首多个连续的空白字符被分隔为一个独立的字段。