**awk使用**

awk工具可以把需要处理的文件按行读取到内存中，每行默认用空格作为分隔符来吧内容分割成多个部分，每个部分依此定义为$1、$2依此类推，$0代表整行。

awk命令格式：

awk -选项 '操作符/模式/ {所需操作命令}' 文件名

**awk命令常用选项**：

-F：指定分割符

-v：自定义变量，每一个“-v”定义一个变量

**awk常用操作符：**

~：代表含有后面匹配条件的行，为模糊操作。

例如：

awk -F: '$5~/root/{print $0}' /etc/passwd表示打印出第5部分包含root的行

结果：显示/etc/passwd文件的第一行。

!~：与~相反，代表不匹配后面的内容。

实例：

awk -F: '$5!~/root/{print $0}' /etc/passwd

结果：显示/etc/passwd文件除去第一行以外的所有行。

==：完全匹配，费模糊搜索。

!=：与==相反

>=：大于等于

例如：打印第3部分大于等于1000的行

awk -F: '$3>=1000{print $0}' /etc/passwd

<=：小于等于

条件1 && 条件2：只有条件1和条件2都成立才执行{}里的操作

条件1 || 条件2：满足其中1个条件即可。

**模式：**

//：匹配条件，如/root/则是匹配一行文本里存在root这个关键字才符合条件。

BEGIN{}：在开始处理文本之前执行一次

实例：

awk -F: 'BEGIN{print "username uid\n"}{print $1, $3}' /etc/passwd

END{}：文本处理完成后执行一次。

**操作：**

{}里的常用操作主要是print和printf，如print $1就是打印第一列的内容。

**awk判断语句：**

实例：

awk -F: '{if ($3>=10) {print $1, $3}}' /etc/passwd

awk -F: '{if ($3==0) {print $1, "the user is root"} else if ($3>=500) {print $1, "is common user"}}' /etc/passwd

**awk常用内置变量：**

FS：输入字段分隔符，默认为空白字符，作用和-F选项类似。

实例：

awk -v FS=':' '{print $1}' /etc/passwd

OFS：输出字段分隔符，默认为空白字符

实例：

awk -v OFS=':' '{print $2, $3}' /etc/passwd

NF：记录awk打印出来的每行一共有几列。比如一行文字被默认分隔符空格分为7列，那么{print NF}出的结果会显示成7，在awk中引用变量无需加$符号，否则{print $NF}的结果是就是第7列的值（也就是最后一部分）。

NR:打印出每一行的行号。

实例：

awk -F: '{print NR, $1}' /etc/passwd