### Awk过滤筛选

格式： awk 【选项】 ‘【条件】{指令}’ 文件

其中，print 是最常用的编辑指令；若有多条编辑指令，可用分号分隔。

处理文本时，若未指定分隔符，则默认将空格、制表符等作为分隔符。

直接过滤文件内容

选项：-F 指定分隔符

awk -F: '{print NR,NF}' sucai.txt 指定：为分隔符

awk -F[:/] '{print NR,NF}' sucai.txt 指定：或者/为分隔符

NR：文件当前行的行号

NF：文件当前行的列数

在脚本中加入这两行并放到/var/www/cgi-bin就能在网页上运行脚本

echo "Context-type:text/html"

echo ""

完整的格式：

awk ‘BEGIN{指令} {指令} END{指令}’

BEGIN{}行前处理，读取文件内容之前执行，指令执行一次

{}逐行处理，读取文件过程中执行，指令执行n次

END{}行后处理，读取文件结束后执行，指令执行一次

Column -t 文本对齐（制作表格）

Sort 从小到大排序 不 +n 是一个字符i一个字符比

Sort -n 按数字来比

1. 字符和数字的比较 == != > >= < =<

[root@svr7 ~]# awk -F: '$1=="root"' /etc/passwd(字符需要加双引)

[root@svr7 ~]# awk -F: '$3==1000' /etc/passwd

1. 逻辑比较 && ||

[root@svr7 ~]# awk -F: '$3>10 && $3<20' /etc/passwd

1. Awk的if语句

单分支if （判断） {指令}

awk -F: '{if($3>1000){x++}} END{print x}' /etc/passwd

多分支If(判断）{指令} else {指令}

1. for循环语句

[root@svr7 ~]# awk 'BEGIN{z[1]=11;z[2]=22;z[3]=33;z[4]=44; for(i in z){print z[i]}}'

DOS攻击 （deny of service）

ab -c 100 -n 100000 <http://192.168.4.207/> （ab模拟）

模仿100个人访问一个网页10w次

[root@pc207 ~]# awk '{ip[$1]++}END{for(i in ip){print ip[i],i}}' /var/log/httpd/access\_log 筛选访问的次数（可以算是一个公式）