Linux 运维常用命令

Nohup

nohup 命令运行由 Command 参数和任何相关的 Arg 参数指定的命令，忽略所有挂断（SIGHUP）信号。在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令，添加 & （ 表示“and”的符号）到命令的尾部。

nohup 是 no hang up 的缩写，就是不挂断的意思。
缺省情况下该作业的所有输出都被重定向到一个名为 nohup.out 的文件中

案例

1. nohup command > myout.file 2>&1 &

在上面的例子中，0 – stdin (standard input)，1 – stdout (standard output)，2 – stderr (standard error)；
2>&1是将标准错误（2）重定向到标准输出（&1），标准输出（&1）再被重定向输入到myout.file文件中。

2. 0 22 * * * /usr/bin/python /home/pu/download_pdf/download_dfcf_pdf_to_oss.py > /home/pu/download_pdf/download_dfcf_pdf_to_oss.log 2>&1

这是放在crontab中的定时任务，晚上22点时候怕这个任务，启动这个python的脚本，并把日志写在download_dfcf_pdf_to_oss.log文件中

nohup 和&的区别

1. & ： 指在后台运行：指在后台运行，但当用户推出(挂起)的时候，命令自动也跟着退出
2. nohup ： **不挂断**的运行，注意并没有后台运行的功能，，就是指，用 nohup 运行命令可以使命令永久的执行下去，**和用户终端没有关系**，例如我们断开 SSH 连接都不会影响他的运行，注意了 nohup 没有后台运行的意思；&才是后台运行

我们可以巧妙的吧他们结合起来用就是

nohup COMMAND &

这样就能**使命令永久的在后台执行**

系统监控

free 命令能够显示系统中物理上的空闲和已用内存，还有交换内存，同时，也能显示被内核使用的缓冲和缓存

top 命令可以实时动态地查看系统的整体运行情况，是一个综合了多方信息监测系统性能和运行信息的实用工具，TOP 命令是 Linux 下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，有点像 window 系统的任务管理器

df 命令用于显示磁盘分区上的可使用的磁盘空间。默认显示单位为 KB。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

ps 命令用于查看进程统计信息

tail 命令可用于查看文件的内容，语法为

less
more

sz 命令是利用 ZModem 协议来从 Linux 服务器传送文件到本地，一次可以传送一个或多个文件。

上传文件
rz

netstat 命令是用于监控进出网络的包和网络接口统计的命令行工具

curl 命令是一个**利用 URL 规则在 shell 终端命令行下工作的文件传输工具**；curl 命令作为一款强力工具，curl 支持包括 HTTP、HTTPS、ftp 等众多协议，还支持 POST、cookies、认证、从指定偏移处下载部分文件、用户代理字符串、限速、文件大小、进度条等特征；

uname 命令用于查看内核版本

查看 linux 的 ip 地址：可以用命令

```
ip addr
```

不管在 window 还是 linux 系统要校验某台服务器是否可以 ping 通，都可以使用命令，如果要加上断口的，linux 可以使用 telnet 命令
语法：telnet ip port

```
telnet 127.0.0.1 8080
```

Awk

awk 是一个强大的文本分析工具，相对于 grep 的查找，sed 的编辑，awk 在其对数据分析并生成报告时，显得尤为强大。简单来说 **awk 就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。**

awk 有 3 个不同版本：awk、nawk 和 gawk，未作特别说明，一般指 gawk，gawk 是 AWK 的 GNU 版本。

awk 其名称得自于它的创始人 Alfred Aho 、Peter Weinberger 和 Brian Kernighan 姓氏的首个字母。实际上 AWK 的确拥有自己的语言： AWK 程序设计语言 ， 三位创建者已将它正式定义为“样式扫描和处理语言”。它允许您创建简短的程序，这些程序读取输入文件、为数据排序、处理数据、对输入执行计算以及生成报表，还有无数其他的功能。

```
awk '{pattern + action}' {filenames}
```

pattern 表示 AWK 在数据中查找的内容，而 action 是在找到匹配内容时所执行的一系列命令。花括号({})不需要在程序中始终出现,但它们用于根据特定的模式对一系列指令进行分组。pattern

就是要表示的正则表达式，用斜杠括起来。

有三种方式调用 awk:
1.命令行方式
awk [-F field-separator] 'commands' input-file(s)
其中，commands 是真正 awk 命令，[-F 域分隔符]是可选的。 input-file(s) 是待处理的文件。在 awk 中，文件的每一行中，由域分隔符分开的每一项称为一个域。通常，在不指名-F 域分隔符的情况下，默认的域分隔符是空格。

2.shell 脚本方式
将所有的 awk 命令插入一个文件，并使 awk 程序可执行，然后 awk 命令解释器作为脚本的首行，一遍通过键入脚本名称来调用。
相当于 shell 脚本首行的: #!/bin/sh
可以换成: #!/bin/awk

3.将所有的 awk 命令插入一个单独文件，然后调用:
awk -f awk-script-file input-file(s)
其中，-f 选项加载 awk-script-file 中的 awk 脚本，input-file(s)跟上面的是一样的。

awk 工作流程是这样的: 读入有'\n'换行符分割的一条记录，然后将记录按指定的域分隔符划分域，填充域,$0 则表示所有域,$1 表示第一个域,$n 表示第 n 个域。默认域分隔符是"空白键" 或 "[tab]键",所以$1 表示登录用户，$3 表示登录用户 ip,以此类推。

```
awk -F 'w' '{print $1; print $2}' log.txt
awk -F 'w' '{print $1;}' log | sort
```

sort:用于排序

```
cat  test.log|awk -F" " '{print $2}'|sort|uniq -c|sort -nrk 1 -t' '|awk -F" " '{print $2}'|head -10
```

问题剖析:

1.cat *.log 将文本内容打印到屏幕

2.使用 awk 命令可以按照分割符将一行分割为多个列，第一列用$1 表示，第二列用$2 表示，依次类推

```
   awk -F" " '{print $2}     //表示用空格作为分隔符进行分割，打印出第 2 列
```

3.sort 进行排序，默认是按照 ascii 码进行排序的

4.**uniq -c 统计相邻的行的重复数量**，结果是类似 3 127.13.13.13,前面的数字代码重复的行数

```
  sort|uniq -c   //统计重复的行数
```

5.**sort -n 是按照数值进行由小到大进行排序， -r 是表示逆序，-t 是指定分割符，-k 是执行按照第几列进行排序**

```
sort -nrk 1 -t' '
```

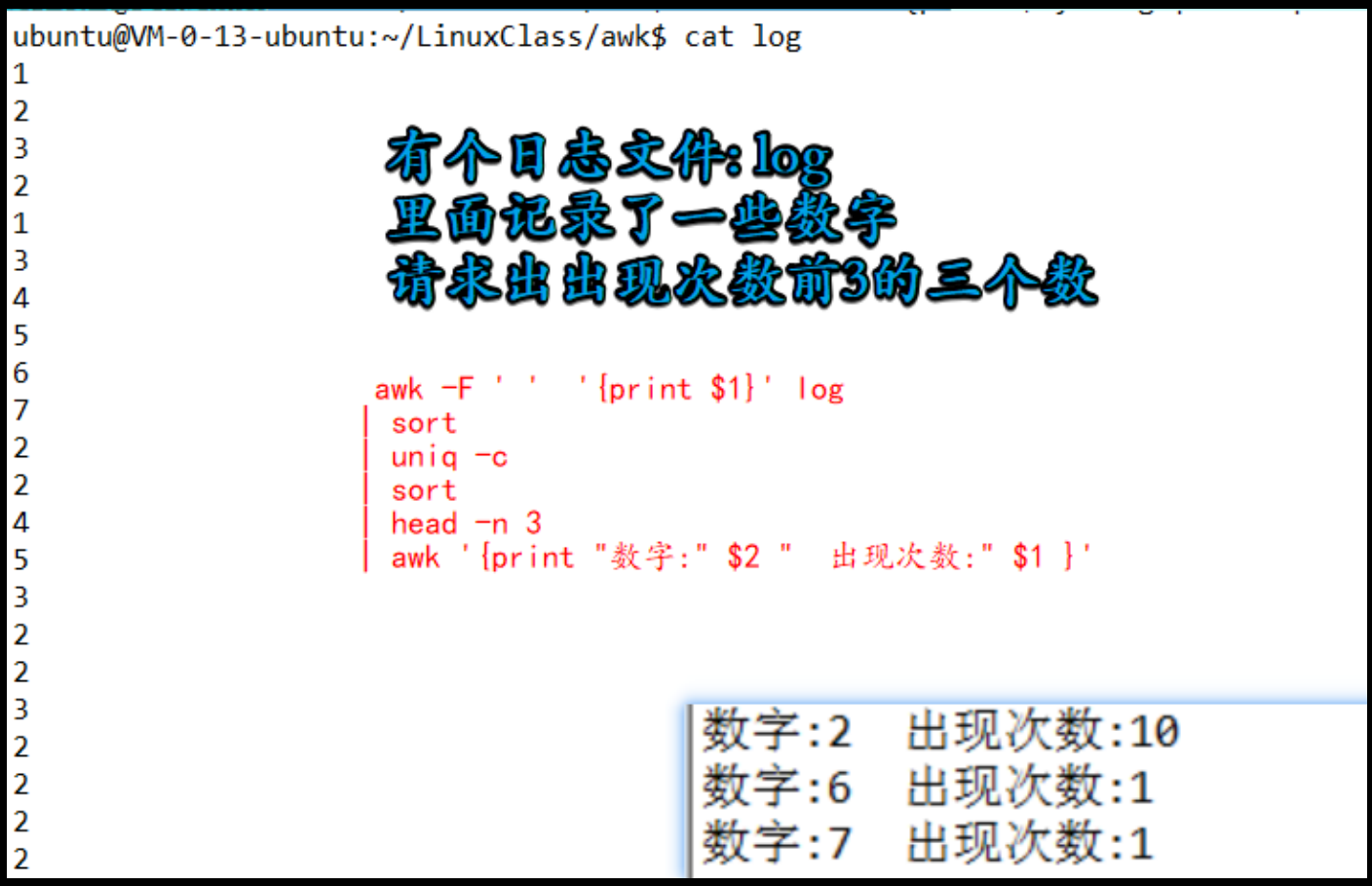
6.使用 awk 命令可以按照分割符将一行分割为多个列，第一列用$1 表示，第二列用$2 表示，依次类推

```
awk -F" " '{print $2}'    //表示用空格作为分隔符进行分割，打印出第 2 列
```

7.head -n 表示取前 n 个

head -10

例题：




```
awk -F ' ' '{print $1}' log | sort | uniq -c |sort | head -n 3 | awk '{print "数字:" $2 "  出现次数:" $1 }'
```

重定向

- 输入：
  - <<以键盘作为输入
  - <以文件作为输入
- 输出
  - >覆盖输出
  - >>追加输出
  - 2>标准错误
  - 2>>标准错误
  - &>标准错误+标准输出
  - &>> 追加形式