

走进流编辑器
的王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

走进流编辑器的王国(Sed,AWK,...)

冯忆¹

¹北邮信息网络中心

介绍Sed, AWK等流编辑器, 网络中心117
December 20, 2008

Outline

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

1 老鸟们都会用Sed和AWK

- Unix文化

2 Sed王后

- Sed概述
- 编辑命令
- Sed进阶

3 AWK国王

- AWK与Sed
- 编辑命令
- AWK杂碎

4 经典小问题

- 小问题举例
- 效率对比

5 状元们

什么是流编辑器？

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

1 共同特性

- 工作：非交互 “输入->加工->输出” vs 交互 “WYSIWYG”
- 配合：相互交流的语言—文本

2 王国成员

- 王后 Sed
- 国王 AWK
- 状元 cut, tr, head, tail...

用流编辑器能让我们...

走进流编辑器
的王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

❶ 融入传统的Unix文化 流编辑器的使用反映了如下的文化背景:

- 命令行界面——文本显示，文本配置...
- K.I.S.S & Pipe...
eg. `ls -l|awk '{print $7}'`

❷ 帮助你更好的完成工作

- “K.I.S.S & Pipe” 和 `popen()` eg. 统计表项
- 如果有一个上G的文本文件？
- 如果有1000个上G的文本文件？
- 文件虽小，要求很多- 删除行，删除列，替换，插入...

Sed的故事

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

- 1 Sed全称 Stream Editor
- 2 Sed是早期Unix的工具之一，快40年历史了

Sed的直观例子

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

file.txt

- daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
saned:x:111:123::/home/saned:/bin/false
echo:x:1000:1000::/home/echo:/bin/bash
root:x:0:0:root:/root:/bin/bash

打印以“root”开头的行：
`sed -n '/^root/{p}' file.txt`

Sed是如何工作的

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

其实很简单...

Sed工作模式：执行如下循环

- 读入下一行
- 用n个“编辑命令”依次处理该行，前者输出为后者输入
- 将处理后的结果输出到屏幕

Sed做的 和 我们要做的：

- Sed自动的**循环**读入每一行
- 每一行的处理是由Sed**顺序**执行编辑命令
- 我们要做的就是设计“编辑命令”

Sed的命令行写法

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

为了与AWK对比，我们采用这种写法：

- `sed [-n] '编辑命令1; 编辑命令2' 文件名`
- 其中，“编辑命令”格式为 `'模式{操作;...}'` 所以展开如下：
`sed [-n] '模式1{操作a;操作b}; 模式2{操作c;操作d}' 文件名`

其他允许的写法：

- ① `sed -n -e '编辑命令1' -e '编辑命令2' -e ... 文件名`
- ② `sed -n -e '编辑命令1;编辑命令2;...' 文件名`
- ③ `sed -n -e '编辑命令1
编辑命令2
编辑命令3' 文件名`
- ④ `sed -n -f 编辑命令文件 文件名`

“编辑命令”的格式

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

编辑命令

模式{操作}

其中，模式可以为单个“行号”或“正则式”，或者两者组合：

- ① 默认的地址 `sed -n '{p}' file.txt`
- ② 单行号地址 `sed -n '3{p}' file.txt`
- ③ 双行号地址 `sed -n '1,3{p}' file.txt`
- ④ 单正则地址 `sed -n '/^saned/{p}' file.txt`
- ⑤ 双正则地址 `sed -n '/^saned/,/^root/{p}' file.txt`
- ⑥ 混合的地址 `sed -n '1,/^root/{p}' file.txt`
- ⑦ 特殊\$地址 `sed -n '${p}' file.txt`

“编辑命令”中的“操作”——就是内置函数

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

修改函数：

① **d 删除本行，进入下一轮** `sed -e '/root/{d}' file.txt`

② **a 后插n行**
`sed -e '/echo/a HaHaHa\nHiHiHi' file.txt`
`sed -e '/echo/{a\`
`HaHaHa\`
`HiHiHi`
`}' file.txt`

③ **i 前插n行**
`sed -e '/echo/i HaHaHa\nHiHiHi' file.txt`

④ **c 替换一行**
`sed -e '/echo/c HaHaHa\nHiHiHi' file.txt`

“编辑命令”中的“操作”——就是内置函数

走进流编辑器
的王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

替换函数：

① s/正则式/替代式/标志

```
sed '/echo/{s/1000/2000/}' file.txt  
sed '/echo/{s/1000/2000/g}' file.txt  
sed -n '/echo/{s/1000/2000/p}' file.txt  
sed '/echo/{s/1000/2000/w result.txt}' file.txt  
sed '/echo/{s/1000/9&9/}' file.txt  
sed '/echo/{s/(10\)00/1/}' file.txt
```

② y/被替换字母/替换字母/

```
sed 'y/abcdefghijklmnopqrstuvwxyz/  
ABCDEFGHIJKLMNOPQRSTUVWXYZ/' c.txt
```

“编辑命令”中的“操作”——就是内置函数

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

输入输出函数：

- 1 **p 输出到屏幕** 和-n连用，改变Sed的默认打印特性
`sed -n '/root/{p}' file.txt`
- 2 **w 输出到文件**
`sed -e '/root/{w result.txt}' file.txt`
- 3 **r 从文件读入** 插入方式同a函数
`sed -e '/echo/{r input.txt}' file.txt`

“编辑命令”中的“操作”——就是内置函数

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

杂项函数：

❶ **！不匹配才执行** `sed '/root/!{d}' file.txt`

❷ **q 退出Sed** `sed '/root/{q}' file.txt`

❸ **{ } 组合命令** `sed
'/echo/{s/echo/ECHO;/s/1000/2000/}' file.txt`

❹ **= 打印行号** `sed -n '/echo/{=}' file.txt`

回顾一下Sed到底能做啥...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

Sed工作模式：执行如下循环

- 读入下一行
- 用n个“编辑命令”依次处理该行，前者输出为后者输入
 - 编辑命令的模式：说明了是否处理该行
 - 编辑命令的操作：说明了怎么处理该行，核心的是d,a,i,c,s
- 将处理后的结果输出到屏幕
 - -n参数和p函数连用

Sed做的 和 我们要做的：

- Sed自动的**循环**读入每一行
- 每一行的处理是由Sed**顺序**执行编辑命令
- 我们要做的就是设计“编辑命令”

顺序处理模式是不够的...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

file.txt

```
● daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
saned:x:111:123::/home/saned:/bin/false
echo:x:1000:1000::/home/echo:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

❶ 如何消除file.txt中重复的root行？

Sed支持变量，分支和循环...

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

对读入的每一行，Sed支持如下特性：

- ❶ “全局”变量：“模式空间”和“保存空间”
 - **n N** 读入下一行，且**n**会输出，**N**不会
打印奇数行 `sed '{n;d}' file.txt`
 - **h H g G x** 交换两个变量
`sed '1{h}; 1{s/./ */}; 1{x}; {G}; {s/\n/:/}' file.txt`
 - **D P** 删除和打印
打印奇数行 `sed 'N;P;d' file.txt`
- ❷ 分支和循环：靠跳转实现，类似“jmp”
 - **:tag** 标签
 - **b** 条件跳转
死循环 `sed -e ':tag;b tag' file.txt`
 - **d** 删除本行进入下一轮
 - **t s** 命令成功则跳转

支持变量，顺序，分支和循环意味着——程序设计！...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

1 C/汇编程序设计

- 操作对象 是各种寄存器（CPU，硬件）变量
- 变量类型 电平值——从一个电平到另一个电平
- 操作手段 对寄存器的操作，如关系，算术运算...
- 操作效果 硬件按要求进行工作，比如图形显示
- 工作模式 循环的“读取指令，执行指令”

2 Sed程序设计

- 操作对象 是**两个变量**—— 模式空间 和 保存空间
- 变量类型 **字符串**——从一种模式的字符串到另一种模式
- 操作手段 采用固定的**内置函数** d,a,i,c,s 和 n,N,D,P,h,H,g,G,x...
- 操作效果 变量(字符串)被改变，**无关系、算术运算**
- 工作模式 Sed自动的**循环**读入每一行
对每一行的处理是由Sed**顺序**执行编辑命令
在执行编辑命令时，允许**跳转**，以形成分支和循环，
也允许打破Sed循环，**提前读入**下n行

对我们的启发...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

从而我们可以粗略估量：

- ① 某个问题是否能用Sed来做
- ② 是否用AWK...等更容易

利用变量，顺序，分支和循环的小例子...

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

file.txt

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
saned:x:111:123::/home/saned:/bin/false
echo:x:1000:1000::/home/echo:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

消除重复的root行:

```
1 sed -n -e '/^root/!b t1; G;/\nroot/d;h;P;d; :t1;p' file.txt
```

长编辑命令可写入文件 -f 形式...

走进流编辑器
的王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

cmd_sed.txt

```
● /^root/!b t1
  G
  /\nroot/d
  h
  P
  d
  :t1
  p
```

同上例，消除重复的root行：

```
① sed -n -f cmd_sed.txt file.txt
```

更多参考

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

man sed

比如你会发现 更多的模式，如 '2,+2'

AWK的故事

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

- 1 AWK名字来自 Alfred Aho 、 Peter Weinberger 和 Brian Kernighan 三位作者的首字母
- 2 我的Linux的AWK是mawk版本, man mawk如下:
mawk conforms to the Posix 1003.2 (draft 11.3) definition of the AWK language which contains a few features not described in the AWK book, and mawk provides a small number of extensions.

AWK和Sed是如此的默契——工作模式

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

AWK的工作模式完全同先前Sed的描述：

Sed工作模式：执行如下循环

- 读入下一行
- 用n个“编辑命令”依次处理该行，前者输出为后者输入
- 将处理后的结果输出到屏幕

Sed做的 和 我们要做的：

- Sed自动的**循环**读入每一行
- 每一行的处理是由Sed**顺序**执行编辑命令
- 我们要做的就是设计“编辑命令”

AWK和Sed是如此的默契——命令行写法

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

AWK的命令行写法完全同先前Sed的写法：

Sed的命令行写法：

- `sed [-n] '编辑命令1; 编辑命令2' 文件名`
- 其中，“编辑命令”格式为 `'模式{操作;...}'` 所以展开如下：
`sed [-n] '模式1{操作a;操作b}; 模式2{操作c;操作d}' 文件名`

Sed和AWK的简单对比例子

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

file.txt

- daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
saned:x:111:123::/home/saned:/bin/false
echo:x:1000:1000::/home/echo:/bin/bash
root:x:0:0:root:/root:/bin/bash

sed -n '/^root/{p}' file.txt

awk '/^root/{print}' file.txt

AWK与Sed不同的地方...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

1 Sed程序设计

- 操作对象 是**两个变量**—— 模式空间 和 保存空间
- 变量类型 **字符串**—— 从一种模式的字符串到另一种模式
- 操作手段 采用固定的**内置函数** d,a,i,c,s 和 n,N,D,P,h,H,g,G,x...
- 操作效果 变量(字符串)被改变, **无关系、算术运算**
- 工作模式 Sed自动的**循环**读入每一行
对每一行的处理是由Sed**顺序**执行编辑命令
在执行编辑命令时, 允许**跳转**, 以形成分支和循环,
也允许打破Sed循环, **提前读入**下n行

2 AWK程序设计

- 操作对象 是**预定义变量**和**自定义变量**
- 变量类型 **字符串、数值** 和 **数组**
- 操作手段 除了类似Sed中的**内置函数**外, 采用了**C语言**编程模式, 支持**关系**和**算术运算**, 还有**字段**操作,
对**字符串处理**功能丰富
- 操作效果 变量**被改变** 或 求出**统计结果**
- 工作模式 同上Sed

“变量”之 预定义变量

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

方便用户直接获取重要信息：

❶ \$0 和\$1 - \$n awk

```
'BEGIN{FS=":";OFS="..";ORS="\n\n"};
{sub(/root/,"BOSS",$1);print}' file.txt
```

❷ FS OFS

❸ RS ORS

❹ NR FNR

连接两个文本对应行 awk

```
'NR==FNR{a[NR]=$0};NR>FNR{print $0,a[FNR]}'
file.txt file2.txt
```

❺ NF

打印每行字段数 awk '{print NF,\$0}' file.txt

❻ FILENAME

注意：

预定义变量的值，是可以改变的：

```
awk 'BEGIN{FS=":";OFS=":"}{$1=NR;print}' file.txt
```

“变量”之 自定义变量

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

方便用户存储信息。直接使用，不需定义。

自定义变量接受以下三种类型的值，且可以随时改变类型：

① 数值

统计第3列的和 awk

```
'BEGIN{FS=":"};{s1+= $3};END{print s1}' file.txt
```

② "字符串"

③ 数组

反序打印各行 awk

```
'{Array[NR]=$0};END{for(i=NR;i>=1;i--){print Array[i]}}' file.txt
```

注意：

在算术运算中，非数字的字符串被认为是0值。

如x="a"+2，x获得的值为2

“编辑命令”的格式

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

编辑命令

模式{操作}

“编辑命令”中的模式

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

建立在“基本模式”中“关系运算”之上的两种模式：

- 1 **组合模式** 将关系运算加上 && || 和 ! awk
'BEGIN{FS=":"};(\$3 >= 100 && \$4 > 200)||\$1 == "root"
{print}' file.txt
- 2 **范围模式** 用逗号将两个关系模式隔开 awk
'NR==2,NR==4{print}' file.txt

“编辑命令”中的模式

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

Sed的模式在AWK中的对应：

- ① **Sed行号** `sed -n '2p' file.txt -> awk 'NR==2{print}' file.txt`
- ② **Sed模式** `sed -n '/root/p' file.txt -> awk '$0~/root/{print}' file.txt`

“编辑命令”中的“操作”

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

“操作”在Sed中只有一种类型——内置函数
而在AWK中和C语言编程很相同！并有如下类型：

- ❶ 赋值运算 $=$ $+=$ $-=$ $*=$ $/=$ $\%=$ $++$ $--$
- ❷ 算术运算 $+$ $-$ $*$ $/$ $\%$
- ❸ 关系运算 $<$ $<=$ $==$ $!=$ $>=$ $>$
- ❹ 内置函数
 - 数学函数 `sqrt` `sin` `cos` `atan` `log` `exp` `int` `rand` `srand`...
 - 输出函数 `print` `printf` `sprintf`...
 - 有模式无操作时，自动打印匹配的行
 - 有操作时，仅当用户主动print时会打印
 - 允许用 `>` 输出到文件
 - 字符串处理 `index` `length` `match` `sub` `gsub` `substr` `split` `tolower` `toupper`
- ❺ 自定义函数
- ❻ 控制流程
 - 处理一行 `if` `while` `for` `break` `continue`
 - AWK行循环 `next` `exit` `getline`

“编辑命令”中的“操作”

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后
Sed概述
编辑命令
Sed进阶

AWK国王
AWK与Sed
编辑命令
AWK杂碎

经典小问题
小问题举例
效率对比

状元们

Summary

file.txt

- daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/bash
saned:x:111:123:./home/saned:/bin/false
echo:x:1000:1000:./home/echo:/bin/bash
root:x:0:0:root:/root:/bin/bash

例子 仅仅展示一些用法:

```
awk 'BEGIN{FS=":"}; $0~/^echo/{print foo($3,$4)}; function  
foo(n,m){if(n > 10){n=sqrt(n)+m;}return n;}' file.txt
```

回顾AWK与Sed不同的地方...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

1 Sed程序设计

- 操作对象 是**两个变量**—— 模式空间 和 保存空间
- 变量类型 **字符串**—— 从一种模式的字符串到另一种模式
- 操作手段 采用固定的**内置函数** d,a,i,c,s 和 n,N,D,P,h,H,g,G,x...
- 操作效果 变量(字符串)被改变, **无关系、算术运算**
- 工作模式 Sed自动的**循环**读入每一行
对每一行的处理是由Sed**顺序**执行编辑命令
在执行编辑命令时, 允许**跳转**, 以形成分支和循环,
也允许打破Sed循环, **提前读入**下n行

2 AWK程序设计

- 操作对象 是**预定义变量**和**自定义变量**
- 变量类型 **字符串**、**数值** 和 **数组**
- 操作手段 除了类似Sed中的**内置函数**外, 采用了**C语言**编程模式, 支持**关系**和**算术**运算, 还有**字段**操作,
对**字符串处理**功能丰富
- 操作效果 变量**被改变** 或 求出**统计结果**
- 工作模式 同上Sed

对我们的启发...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

从而我们可以粗略估量：

- 1 某个问题是否能用AWK来做
- 2 是否用Sed... 更简洁

print函数

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

print函数和sqrt, sub等函数不同, 参数列表不用括号
且有如下的使用形式:

① 字符串连接

- 直接挨着 `awk 'BEGIN{FS=":"};{print $1":"$2}' file.txt`
- 逗号连接 `awk 'BEGIN{FS=":";OFS=".."};{print $1,$2}' file.txt`

② 使用管道 `awk 'BEGIN{FS=":"};{print $1 | "sort"}' file.txt`

printf的参数

- ① 逗号隔开 `awk 'BEGIN{FS=":"};{printf "%-5d %-5d\n", $3, $4}' file.txt`

如果编辑命令很长很长...

走进流编辑器
的王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

cmd_awk.txt

```
BEGIN{FS=":"}  
$0~/^echo/{print foo($3,$4)}  
function foo(n,m) {  
    if(n > 10) {  
        n=sqrt(n)+m;  
    }  
    return n;  
}
```

❶ awk -f cmd_awk.txt file.txt

FS, 哈希表 和 看电影

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

- 1 FS为空 空输入行作为记录分割符，空格、tab和换行为字段分割符
- 2 哈希表？ `awk '/^root/{Array["root"]++};END{print Array["root"]}' file.txt`
- 3 看电影？ `awk '/root/{system("gmplayer");}' file.txt`

小问题举例一

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK
Unix文化

Sed王后

Sed概述
编辑命令
Sed进阶

AWK国王

AWK与Sed
编辑命令
AWK杂碎

经典小问题

小问题举例
效率对比

状元们

Summary

1 打印偶数行

- `awk 'NF % 2 == 0{print}' file.txt`
- `sed -n '{n;p}' file.txt`
- `sed '2~2{p}'`

2 打印从2行开始的3行（即2行到4行）

- `awk 'NR>=2 && NR<=4{print}' file.txt`
- `sed -n '2,+2{p}' file.txt`

3 加上行号

- `awk '{print NR" "$0}' file.txt`
- `sed '=' file.txt | sed -n 'N;s/\n/ /p'`

4 打印前4行

- `awk 'NR<=4{print};NR>4{exit}' file.txt`
- `sed '4q' file.txt`

小问题举例二

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

1 交换两行

- `sed -n '{h;n;G;p}' file.txt`

- `awk '{getline var; print var"\n"$0}' file.txt`

2 消除空行

- `sed '/^$/d' file.txt`

- `awk 'NR > 0{print}' file.txt`

3 合并多行

- `sed ':t1;N;s/\n/./;b t1' file.txt`

- `awk 'BEGIN{ORS=".."};{print}' file.txt`

效率对比

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

不用比了，sed通常比AWK快！

小是小，方便...

走进流编辑器的
王国

冯忆

老鸟们都会
用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

cut • fmt • head • iconv • sort
tac • tail • tr • uniq • wc

更多学习资料

走进流编辑器的王国

冯忆

老鸟们都会用Sed和AWK

Unix文化

Sed王后

Sed概述

编辑命令

Sed进阶

AWK国王

AWK与Sed

编辑命令

AWK杂碎

经典小问题

小问题举例

效率对比

状元们

Summary

- 1 BOSS网站 <http://buptoss.3322.org/>
- 2 UnixCenter <http://www.unix-center.net/>