



北京邮电大学

Beijing University of Posts and Telecommunications

复杂筛选及加工awk

蒋砚军 北京邮电大学计算机学院



► awk: 逐行扫描进行文本处理的一门语言



a.w.k分别为该程序的三位设计者姓氏的第一个字母

■ 用法

- ◆ awk '程序' 文件名列表
- ◆ awk -f 程序文件名 文件名列表

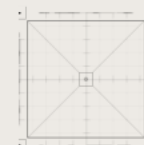
程序 **条件 {动作}**

awk自动对每行文本执行**条件**判断，满足条件执行**动作**（内置循环）

允许多段程序：多段程序间用空格或分号隔开

■ 处理方式

- ◆ 输入文件的每行作为一个“记录”，变量NR就是行号
- ◆ 每行用空格分隔开的部分，叫做记录的“域”
内置变量\$1是第1域内容，依次，\$2是第2域内容，.....
特别的，\$0指的是整个这一行的内容
- ◆ awk的处理为：符合**条件**的行，执行相应的**动作**



► awk描述条件的方法



■ 使用与C语言类似的关系算符

< 小于 <= 小于或等于 == 等于
!= 不等于 > 大于 >= 大于或等于

■ 使用与C语言类似的逻辑算符

|| 条件或 && 条件与 ! 条件非

■ 正则表达式的模式匹配 */regexpr/*

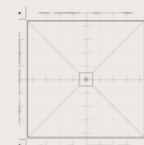
◆ 包含该模式的行，执行动作

■ 特殊的条件

◆ 不指定任何条件，对所有文本行执行动作

◆ BEGIN 开始处理所有文本行之前执行动作

◆ END 处理完所有文本行之后执行动作



► awk描述动作的方法



■ 描述“动作”时，简单的用法有：

- ◆ 自定义变量

- ◆ 加减乘除等算术逻辑运算

- ◆ 正则表达式匹配运算符（用作条件判断） ~ !~

例如： \$2 ~ "[1-9][0-9]*"

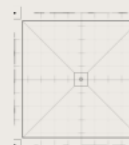
- ◆ 流程控制（与C语言类似）

 - 条件判断 if

 - 循环控制 for

- ◆ print 变量1, 变量2,

printf("格式串", 变量1, 变量2,)



► awk举例



```
$ ps -ef | grep guest
  guest   669   668   0 11:27:13   ttyt1    00:00:00  -sh
  guest   678   669   0 11:27:18   ttyt1    00:00:00  vi
$ ps -ef | awk '/guest/{ printf("%s ",$2); }'
669 678
$ awk '{printf("%d: %s\n",NR,$0); }' test.c
1: main()
2: {
3:     printf("Hello!\n");
4: }
$ date
Thu May 27 22:02:22 BEIDT 2004
$ date | awk '{print $4}'
22:02:42
$ ls -s | awk '$1 > 2000 { print $2 }'
disk.img
document.pdf
linux-src.tar.Z
pppd.log
```



北京邮电大学

Beijing University of Posts and Telecommunications



谢谢