

SHELL DAY05



# Shell脚本编程

NSD SHELL

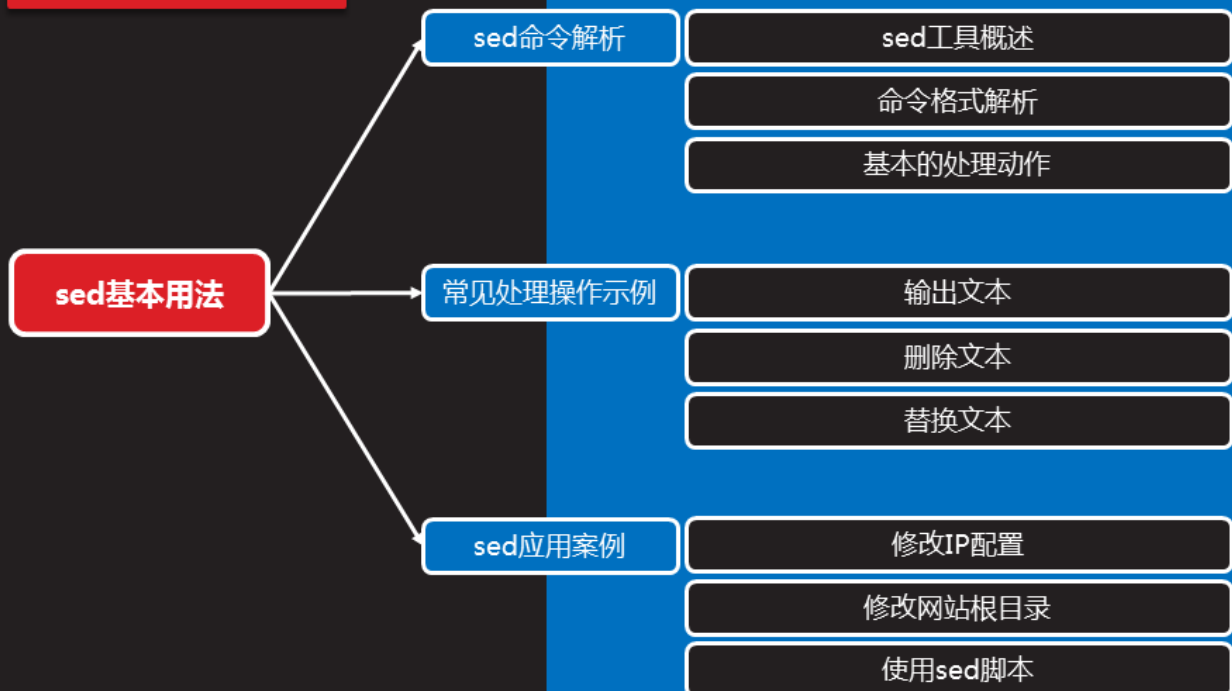
DAY05

# 内容

上午	09:00 ~ 09:30	作业讲解与回顾
	09:30 ~ 10:20	sed基本用法
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	sed文本块处理
	15:00 ~ 15:50	sed高级应用
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



## sed基本用法



# sed命令解析

## sed工具概述

- Stream Editor , 流式编辑器
  - 非交互, 基于模式匹配过滤及修改文本
  - 逐行处理, 并将结果输出到屏幕
  - 可实现对文本的输出、删除、替换、复制、剪切、导入、导出等各种操作

# 命令格式解析

知识讲解

- 主要用法

- 格式1：前置命令 | sed [选项] '编辑指令'
- 格式2：sed [选项] '编辑指令' 文件.. ..

└── [定址符]处理动作

└── [地址1,[地址2]]

```
[root@svr5 ~]# grep ^id /etc/inittab
```

```
id:5:initdefault:
```

```
[root@svr5 ~]# sed -n '/^id/p' /etc/inittab
```

```
id:5:initdefault:
```

//列出以id开头的行



## 命令格式解析（续1）

知识讲解

- 常见命令选项

- -n：屏蔽默认输出（全部文本）
- -i：直接修改文件内容
- -r：启用扩展的正则表达式，若与其他选项一起使用，应作为首个选项



## 命令格式解析（续2）

### 知识讲解

- 条件，可以是行号或/正则/
  - 行号可以使用单个数字表示单行
  - 或者3,5表示连续的多行
  - 省略条件，则默认逐行处理全部文本
  - 匹配正则时，需要使用//

```
[root@svr5 ~]# sed -n '2,4p' /etc/passwd
[root@svr5 ~]# sed -n 'p' /etc/passwd
[root@svr5 ~]# sed -n '/root/p' /etc/passwd
```

```
//输出第2-4行
//输出全部
//输出包含root的行
```



## 基本的处理动作

- 常用动作指令

### 知识讲解

操作符	用途	指令示例
p	打印行	2,4p 输出第2、3、4行
		2p;4p 输出第2行、4行
d	删除行	2,4d 删除第2、3、4行
s	字符串替换	s/old/new/ 将每行的第1个old替换为new
		s/old/new/3 将每行的第3个old替换为new
		s/old/new/g 将所有的old都替换为new

替换操作的分隔 “/” 可改用其他字符，如#、&等，便于修改文件路径



# 常见处理操作示例

## 输出文本

知识讲解

示 例	含义解析
<code>sed -n 'p' a.txt</code>	输出所有行，等同于 <code>cat a.txt</code>
<code>sed -n '4p' a.txt</code>	输出第4行
<code>sed -n '4,7p' a.txt</code>	输出第4~7行
<code>sed -n '4,+10p' a.txt</code>	输出第4行及其后的10行内容
<code>sed -n '/^bin/p' a.txt</code>	输出以bin开头的行
<code>sed -n '\$=' a.txt</code>	输出文件的行数



## 删除文本

### 知识讲解

示 例	含义解析
<code>sed '3,5d' a.txt</code>	删除第3~5行
<code>sed '/xml/d' a.txt</code>	删除所有包含xml的行
<code>sed '/xml/!d' a.txt</code>	删除不包含xml的行, !符号表示取反
<code>sed '/^install/d' a.txt</code>	删除以install开头的行
<code>sed '\$d' a.txt</code>	删除文件的最后一行
<code>sed '/^\$/d' a.txt</code>	删除所有空行

此例中只作输出, 不更改原文件 ( 若需要更改, 应添加选项 `-i` )



## 替换文本

### 知识讲解

示 例	含义解析
<code>sed 's/xml/XML/' a.txt</code>	将每行中第一个xml替换为XML
<code>sed 's/xml/XML/3' a.txt</code>	将每行中的第3个xml替换为XML
<code>sed 's/xml/XML/g' a.txt</code>	将所有的xml都替换为XML
<code>sed 's/xml//g' a.txt</code>	将所有的xml都删除 ( 替换为空串 )
<code>sed '4,7s/^/#/' a.txt</code>	将第4~7行注释掉 ( 行首加#号 )
<code>sed 's/^#an/an/' a.txt</code>	解除以#an开头的行的注释 ( 去除行首的#号 )

此例中只作输出, 不更改原文件 ( 若需要更改, 应添加选项 `-i` )



## 案例1：sed基本用法

课堂练习

1. 练习sed命令的p、d、s等操作
2. 拷贝/etc/hosts文件测试，利用sed完成以下任务：
  - 1) 删除文件中每行的第二个、最后一个字符
  - 2) 将文件中每行的第一个、倒数第1个字符互换
  - 3) 删除文件中所有的数字
  - 4) 为文件中每个大写字母添加括号



## sed应用案例



## 修改IP配置

- 修改虚拟机XML文件的名称值<name>

知识讲解

```
[root@svr5 ~]# grep ^IPADDR /etc/sysconfig/network-scripts/ifcfg-eth0  
IPADDR=192.168.1.24 //修改前
```

```
[root@svr5 ~]# sed -ri '/^IPADDR/s/192.168.1.(.*)/172.16.0.\1/' \  
/etc/sysconfig/network-scripts/ifcfg-eth0 //修改操作
```

使用 **\1** 标签可调用本组表达式内第1个用 **()** 保存的匹配结果



## 修改网站根目录

- 任务目标
  - 将 /var/www/html 修改为 /opt/wwwroot

知识讲解

```
[root@svr5 ~]# grep "^DocumentRoot" /etc/httpd/conf/httpd.conf  
DocumentRoot "/var/www/html" //修改前
```

```
[root@svr5 ~]# sed -i 's#/var/www/html#/opt/wwwroot#' \  
/etc/httpd/conf/httpd.conf //修改操作
```



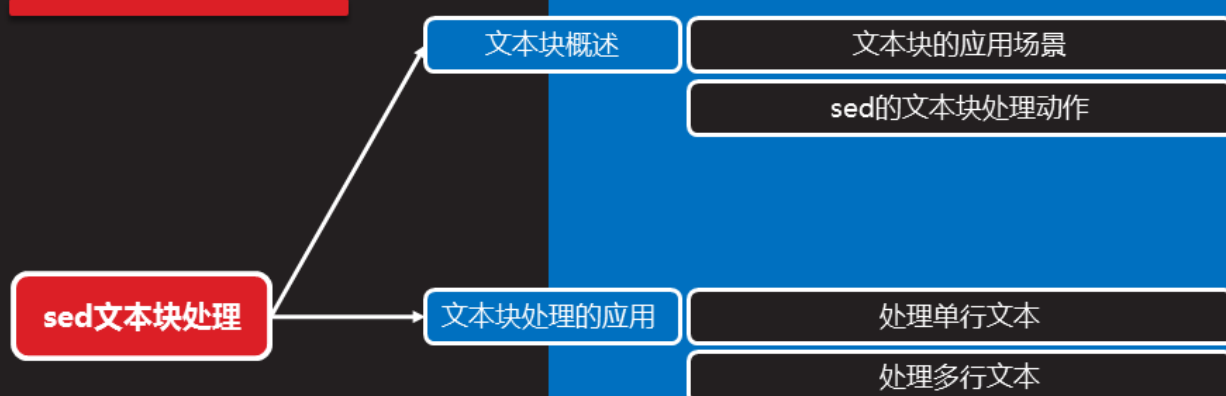
## 案例2：使用sed修改系统配置

课堂练习

1. 练习课上的sed应用案例
2. 编写脚本anonftp.sh，实现的功能如下：
  - 1) 通过yum安装vsftpd软件包
  - 2) 修改vsftpd服务配置，开启匿名上传
  - 3) 调整/var/ftp/pub目录权限，允许ftp写入
  - 4) 启动vsftpd服务，并设置开机自运行



### sed文本块处理



# 文本块概述



## 文本块的应用场景

- 如何在文件内插入新行？
- 如何插入多行文本？
- 如何替换掉指定的整行文本？

知识讲解



# sed的文本块处理动作

知识讲解

操作符	用途	指令示例
i	行前插入文本	2iYY 在第2行之前添加文本行 "YY"
		4,7iYY 在第4-7行的每一行前添加文本行
a	行后插入文本	2aYY 在第2行之后添加文本
		/^XX/aYY 在以XX开头的行之后添加文本
c	替换当前行	2cYY 将第2行的内容修改为 "YY"



## 文本块处理的应用

## 处理单行文本

知识讲解

```
[root@svr5 ~]# sed '2iXX' m.txt
11111111 Tarena
XX
222 IT Group
```

//插入到行前

```
[root@svr5 ~]# sed '2aXX' m.txt
11111111 Tarena
222 IT Group
XX
```

//插入到行后

```
[root@svr5 ~]# sed '2cXX' m.txt
11111111 Tarena
XX
```

//替换指定行



## 处理多行文本

知识讲解

- 修改后的文本有多行时
  - 以换行符 \n 分隔
  - 或者，使用 \ 强制换行

```
[root@svr5 ~]# sed -n '2cXX\nYY' m.txt
XX
YY
```

```
[root@svr5 ~]# sed -n '2cXX\
> YY' m.txt
XX
YY
```



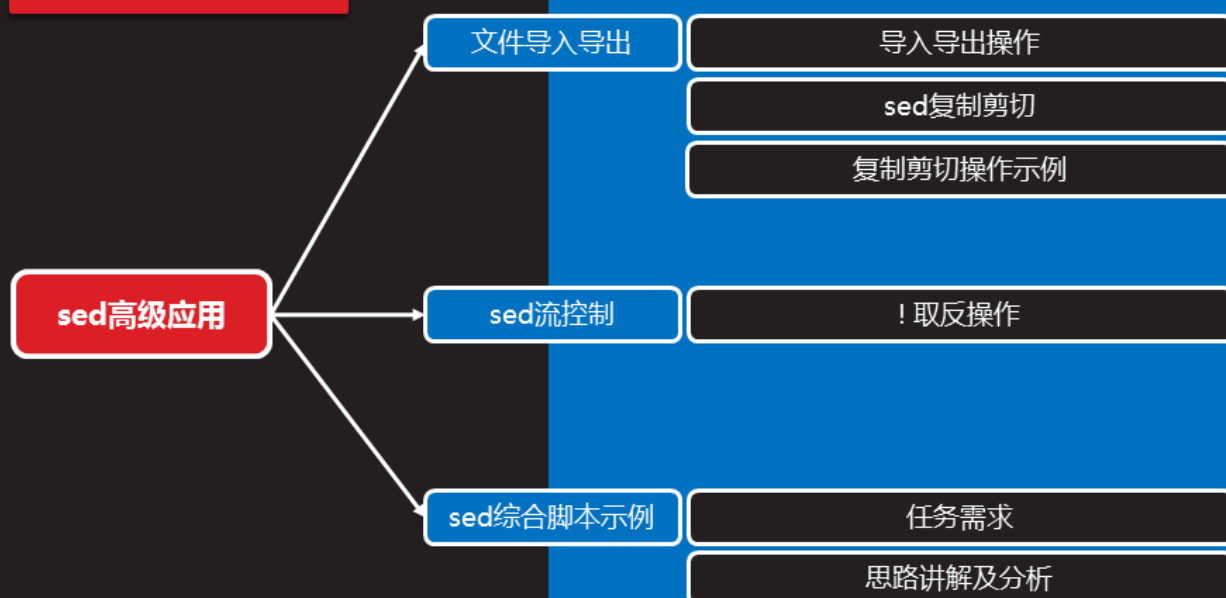
## 案例3：sed多行文本处理

课堂练习

1. 修改主机名配置文件
2. 修改hosts文件，添加两条映射记录
  - 192.168.4.5      svr5.tarena.com    svr5
  - 119.75.217.56      www.baidu.com



### sed高级应用



# 文件导入导出

## 导入导出操作

- 基本动作
  - r 动作应结合 -i 选项才会存入，否则只输出
  - w 动作以覆盖的方式另存为新文件

知识讲解

操作符	用途	指令示例
r	读取文件	3r b.txt 在第3行下方插入文件b.txt
		4,7r b.txt 在第4-7每一行后插入文件b.txt
w	保存到文件	3w c.txt 将第3行另存为文件c.txt
		4,7w c.txt 将第4-7行另存为文件c.txt



## 导入导出操作（续1）

知识讲解

```
[root@svr5 ~]# sed '2r m.txt' reg.txt
```

```
abcd XX
```

```
XX
```

//在第2行下方插入m.txt

```
11111111 Tarena
```

```
222 IT Group
```

```
abcabcd XX
```

```
[root@svr5 ~]# sed '/^abca/r m.txt' reg.txt
```

```
abcd XX
```

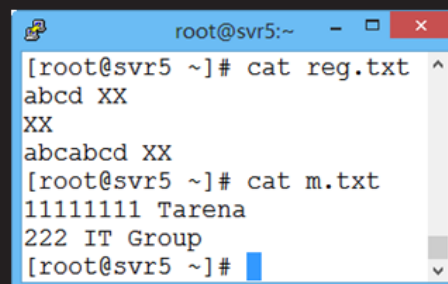
```
XX
```

```
abcabcd XX
```

//在以abca开头的行下方插入m.txt

```
11111111 Tarena
```

```
222 IT Group
```



```
root@svr5:~  
[root@svr5 ~]# cat reg.txt  
abcd XX  
XX  
[root@svr5 ~]# cat m.txt  
11111111 Tarena  
222 IT Group  
[root@svr5 ~]#
```



## 导入导出操作（续2）

知识讲解

```
[root@svr5 ~]# sed -n '1,2w c.txt' reg.txt
```

```
[root@svr5 ~]# cat c.txt
```

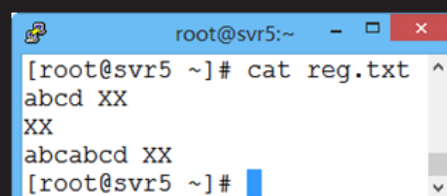
```
abcd XX
```

```
XX
```

```
[root@svr5 ~]# sed -n '/^XX/w d.txt' reg.txt
```

```
[root@svr5 ~]# cat d.txt
```

```
XX
```



```
root@svr5:~  
[root@svr5 ~]# cat reg.txt  
abcd XX  
XX  
[root@svr5 ~]#
```

与以下操作效果相同：sed -n '/^XX/p' reg.txt > d.txt





## sed复制剪切

知识讲解

- 模式空间
  - 存放当前处理的行，将处理结果输出
  - 若当前行不符合处理条件，则原样输出
  - 处理完当前行再读入下一行来处理
- 保持空间
  - 作用类似于“剪贴板”
  - 默认存放一个空行（换行符 \n）



## sed复制剪切（续1）

知识讲解

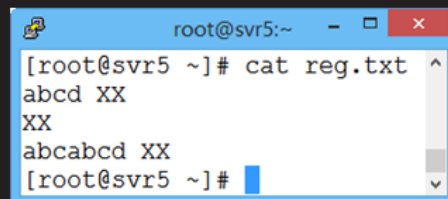
- 基本动作
    - H：模式空间 ---[追加]--->保持空间
    - h：模式空间 ---[覆盖]--->保持空间
    - G：保持空间 ---[追加]--->模式空间
    - g：保持空间 ---[覆盖]--->模式空间
- 复制
- 粘贴



# 复制剪切操作示例

- 任务目标
  - 把第1-3行复制到文件末尾

```
[root@svr5 ~]# sed '1,3H;$G' reg.txt
abcd XX
XX
abcabcd XX
```



```
root@svr5:~
[root@svr5 ~]# cat reg.txt
abcd XX
XX
abcabcd XX
[root@svr5 ~]#
```

```
abcd XX
XX
abcabcd XX
```

追加的保持空间内容

改用 '1h;2,3H;\$G' 可避免出现空行

知识讲解

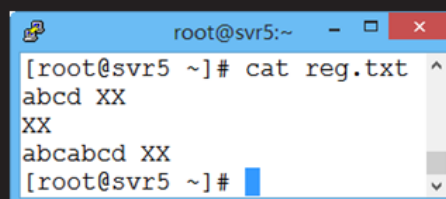


## 复制剪切操作示例（续1）

- 任务目标
  - 把第1行剪切到文件末尾
  - 把第1-2行剪切到文件末尾

```
[root@svr5 ~]# sed '1h;1d;$G' reg.txt
XX
abcabcd XX
abcd XX
```

//原第1行内容



```
root@svr5:~
[root@svr5 ~]# cat reg.txt
abcd XX
XX
abcabcd XX
[root@svr5 ~]#
```

```
[root@svr5 ~]# sed '1h;2H;1,2d;$G' reg.txt
abcabcd XX
abcd XX
XX
```

//原第1行内容

//原第2行内容

知识讲解



# sed流控制

## ！取反操作

知识讲解

- 用途
  - 根据定址条件取反
- 示例
  - 列出不使用bash的用户账号记录

```
[root@svr5 ~]# sed -n '/bash$/!p' /etc/passwd  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
... ..
```

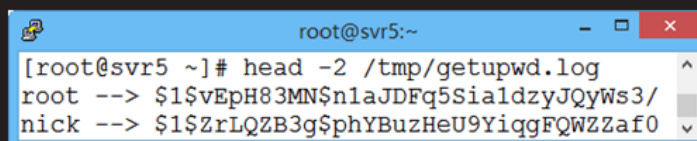


# sed综合脚本示例

## 任务需求

- 根据/etc/passwd提取密码串
  - 找到使用bash作登录Shell的本地用户
  - 列出这些用户的shadow密码记录
  - 按每行“用户名 --> 密码记录” 保存结果

知识讲解



```
root@svr5:~  
[root@svr5 ~]# head -2 /tmp/getupwd.log  
root --> $1$vEpH83MN$nlajDFq5SialdzyJQyWs3/  
nick --> $1$ZrLQZB3g$phYBuzHeU9YiqgFQWZzaf0
```



# 思路讲解及分析

## 参考思路、分析

### 知识讲解

```

root@svr5:~
[root@svr5 ~]# cat getupwd.sh
#!/bin/bash
> /tmp/getupwd.log                                ## 创建空文件
sed -n '/:\/bin\/bash$/w /tmp/urec.tmp' /etc/passwd ## 提取符合条件的账号记录
UNUM=$(egrep -c '.' /tmp/urec.tmp)                 ## 取得记录个数
while [ ${i:=1} -le $UNUM ]                         ## 从第1行开始，遍历账号记录
do
    UREC=$(sed -n "${i}p" /tmp/urec.tmp)             ## 取指定行数的记录
    NAME=${UREC%%:*}                                 ## 截取用户名（记录去尾）
    PREC=$(sed -n "/^$NAME:/p" /etc/shadow)          ## 查找与用户名对应的shadow记录
    PASS=${PREC#*:}                                  ## 掐头
    PASS=${PASS%:*}                                  ## 去尾，只留下密码记录
    echo "$NAME --> $PASS" >> /tmp/getupwd.log        ## 保存结果
    let i++                                           ## 自增1，转下一次循环
done
/bin/rm -rf /tmp/urec.tmp                           ## 删除临时文件
echo "用户分析完毕，请查阅文件 /tmp/getupwd.log"    ## 完成后提示
[root@svr5 ~]#
[root@svr5 ~]# head -2 /tmp/getupwd.log
root --> $1$vEpH83MN$nlajDFq5SialdzyJQyWs3/
nick --> $1$ZrLQZB3g$phYBuzHeU9YiqgFQWZZaf0
[root@svr5 ~]#
  
```



## 案例4：sed综合脚本应用

编写脚本getupwd.sh，实现以下需求：

- 1) 找到使用bash作登录Shell的本地用户
- 2) 列出这些用户的shadow密码记录
- 3) 按每行“用户名 --> 密码记录”保存到getupwd.log

### 课堂练习



## 总结和答疑



## sed替换

## 问题现象

- 故障错误信息

```
[root@svr5 ~]# sed "s/a/b" tesh.txt  
sed: -e expression #1, char 5: unterminated `s' command
```

知识讲解



## 原因分析

- 分析故障
  - 报错信息：unterminated `s' command
- 分析故障原因
  - s替换没找到合法的结束符
  - s/旧/新/，使用s替换时，最后需要使用/结尾

知识讲解



