

第二章 UNIX的软件工具

- 2.1 Unix软件开发工具简介
- 2.2 BACKUS系统
- 2.3 vi
- 2.4 sed
- 2.5 awk
- 2.6 make
- 2.7 SVN

➤ 2.4 sed

➤ sed概述

sed(**S**treamer **E**ditor)称为字符流编辑程序，是一个非交互式的文本编辑程序，作者为Lee E. McMahon(李•麦克马洪)。

适用于sed的情况为：

➤ 不适合vi等交互编辑程序

有的文件行数太多或某些行的字符太多，超出vi和ex能容纳的限度，或者使用vi时读入文件和写出文件的时间很长。

➤ 全局编辑操作

1) 能够事先确定的全局编辑操作。

2) 命令中字符数较多，为避免在键入命令时出错而不宜于进行交互编辑。

➤ 流水线

更多的情况是利用sed可以实现流水线操作。

➤ 2.4 sed

➤ sed的执行方式

➤ 方式1 `sed '命令' [输入文件]` 或者

➤ 方式2 `sed -f 命令文件 [输入文件]`

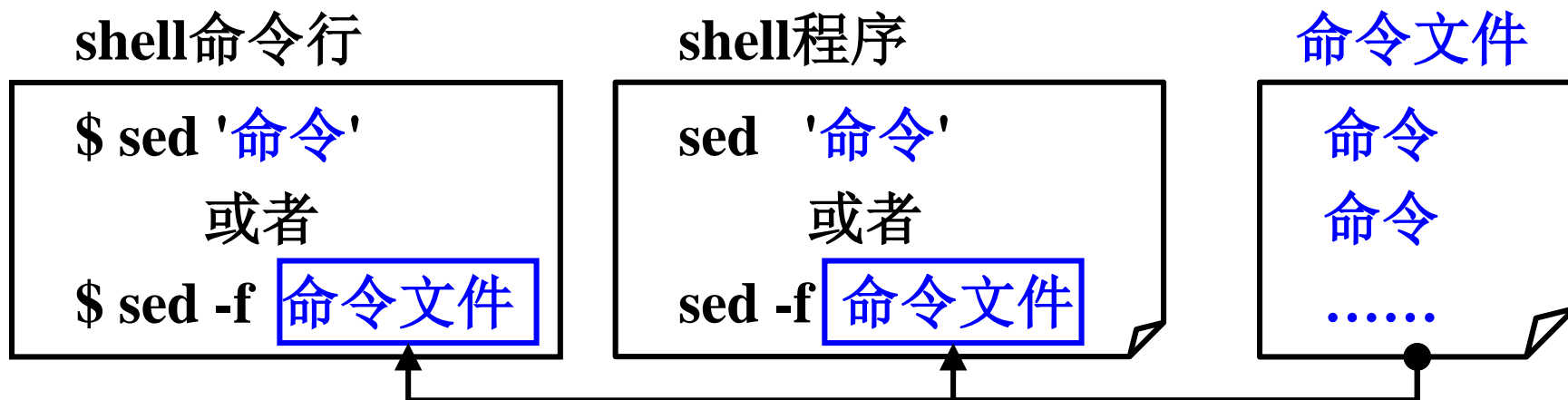
➤ 功能

用单引号括起的'命令'是sed命令。

-f表示指定的命令文件中含有sed命令，称为sed命令文件。

如果缺省输入文件，sed的输入将指向标准输入。

sed的缺省输出是标准输出，但也可以在'命令'中定义。



※sed的命令格式

[*pattern*] [!] *function* [/*cmd*]

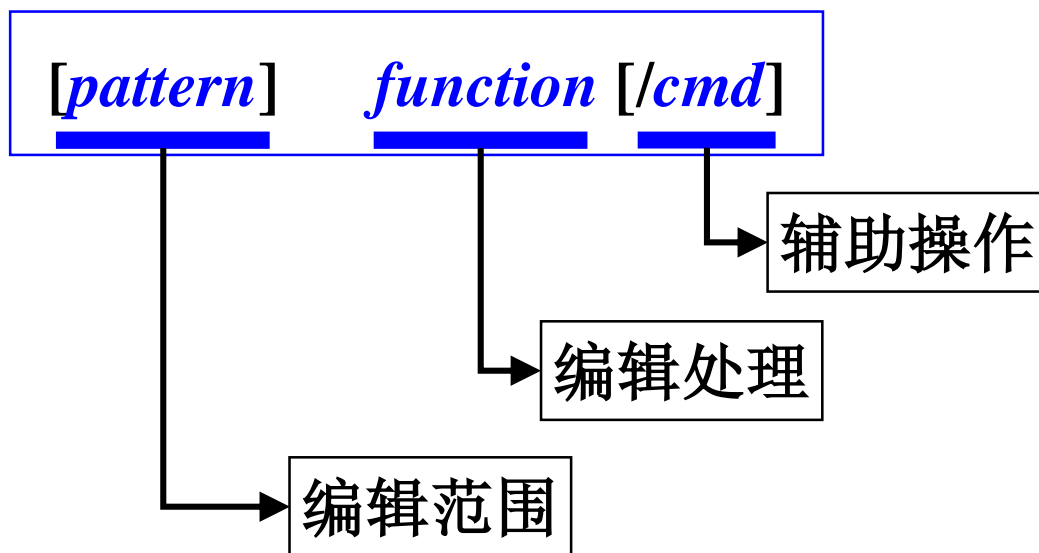
表示用*function*(命令)处理由*pattern*(地址表达式)匹配的行。

⊙如果缺省*pattern*，表示对所有行执行*function*。

⊙如果加!，表示对不匹配*pattern*的行执行*function*。

⊙*cmd*表示命令的辅助功能。

由于sed的许多基本操作也是由ed程序的函数实现的，因此sed的*pattern*和*function*均与vi的全局操作命令相同。



对*pattern*描述的文本用*function*处理
选择某个*cmd*增加辅助操作

※sed的命令格式

[*pattern*] [!] *function* [/cmd]

表示用*function*(命令)处理由*pattern*(地址表达式)匹配的行。

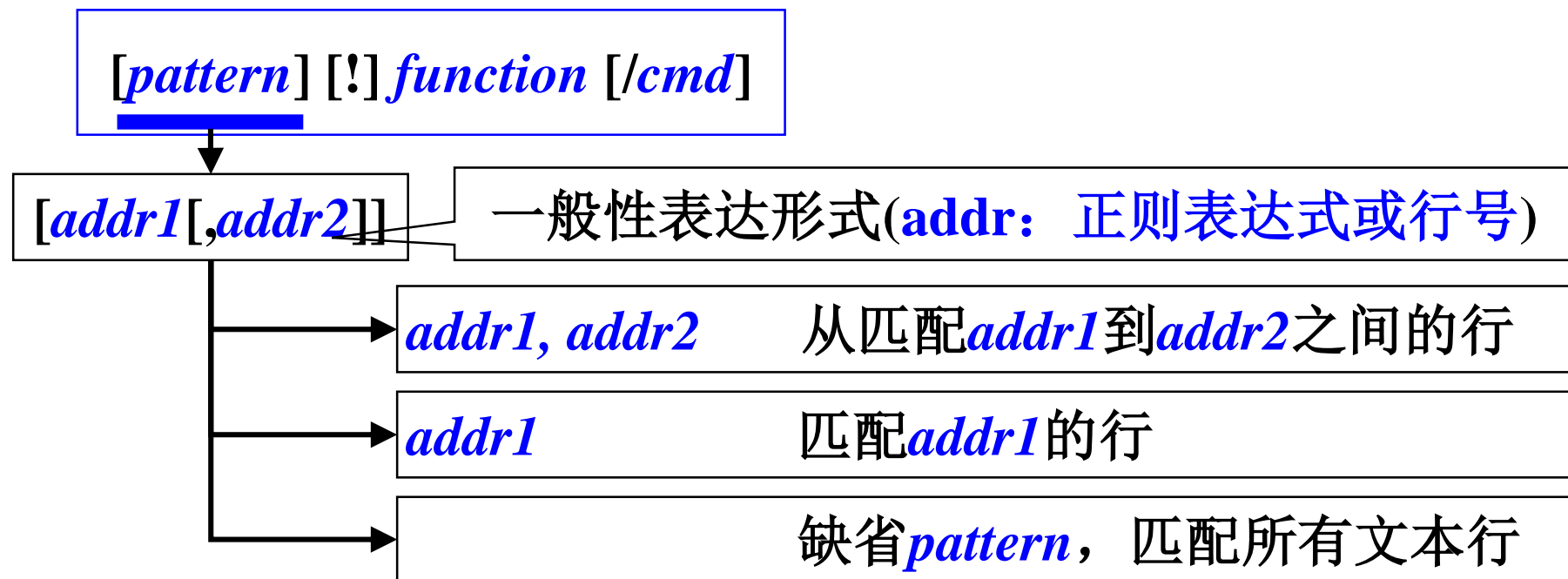
⊙如果缺省*pattern*，表示对所有行执行*function*。

⊙如果加!，表示对不匹配*pattern*的行执行*function*。

⊙*cmd*表示命令的辅助功能。

由于sed的许多基本操作也是由ed程序的函数实现的，因此sed的*pattern*和*function*均与vi的全局操作命令相同。

⊙*pattern*(地址表达式， 又称行地址address)



※sed的命令格式

[*pattern*] [!] *function* [/cmd]

表示用*function*(命令)处理由*pattern*(地址表达式)匹配的行。

⊙如果缺省*pattern*，表示对所有行执行*function*。

⊙如果加!，表示对不匹配*pattern*的行执行*function*。

⊙*cmd*表示命令的辅助功能。

由于sed的许多基本操作也是由ed程序的函数实现的，因此sed的*pattern*和*function*均与vi的全局操作命令相同。

⊙常用的function(命令)

[*pattern*] [!] *function* [/cmd]

→ **d命令** 删除由*pattern*匹配的行

→ **s命令** 替换操作

※sed的命令格式

[*pattern*] [!] *function* [/*cmd*]

表示用*function*(命令)处理由*pattern*(地址表达式)匹配的行。

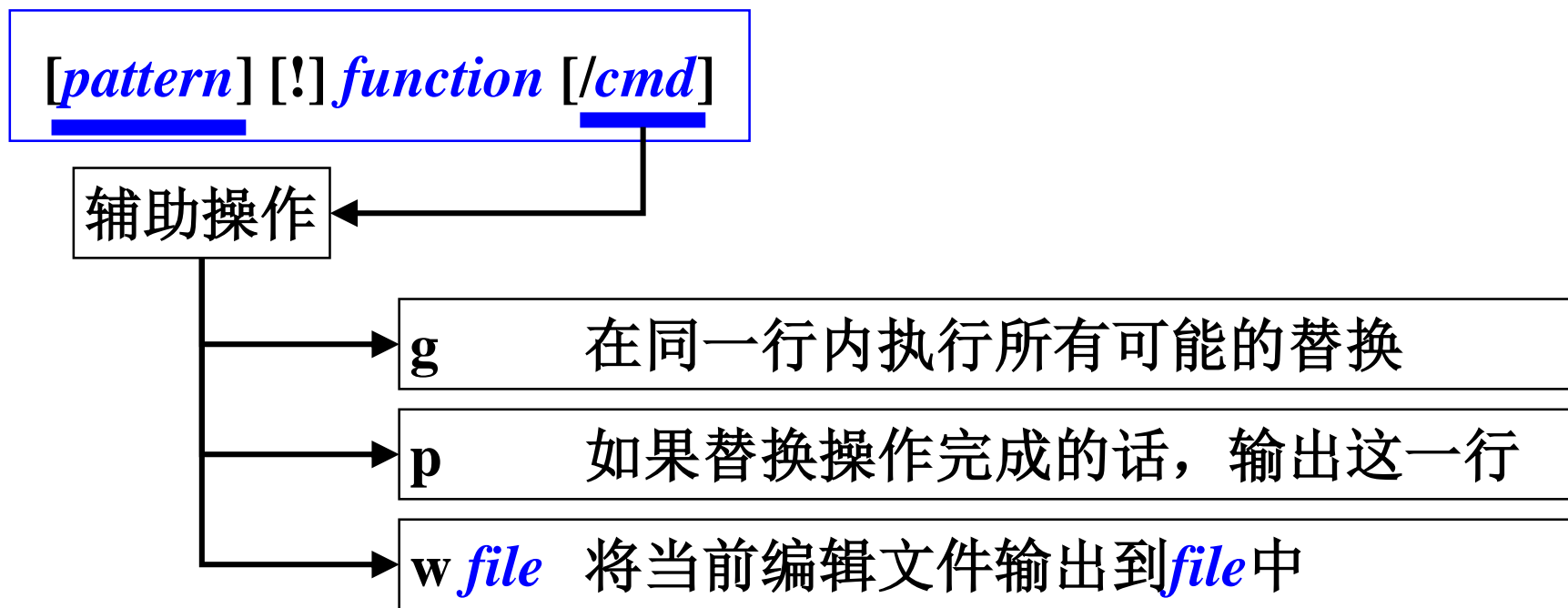
⊙如果缺省*pattern*，表示对所有行执行*function*。

⊙如果加!，表示对不匹配*pattern*的行执行*function*。

⊙*cmd*表示命令的辅助功能。

由于sed的许多基本操作也是由ed程序的函数实现的，因此sed的*pattern*和*function*均与vi的全局操作命令相同。

⊙cmd(s命令的辅助功能)



【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

The Abstract
A novel design
TheStructured
the methodology for

A novel design
the methodology for

行(不删除)

删除)

sed '/^#/, \$d' a.txt

删除从第1个shell注释行到最后一行

sed '/\/* /d' a.txt

删除含/*的行(C语言注释行)

sed '/^\$/d' a.txt

删除所有空行

sed '/^#/s/Q/DFF/g' a.txt

以#开头的行将所有的Q替换为DFF

sed 's/\$/\\/' a.txt

**在每个行尾都加上反斜杠 **

sed 's/^/t/' a.txt

在每行前插入一个制表符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

```
("%s);  
exe1B.c  
printf("input name");  
1230713("%s);
```



```
("%s);  
printf("input name");
```

行(不删除)

删除)

行到最后一行

sed '/\/* /d' a.txt

删除含/*的行(C语言注释行)

sed '/^\$/d' a.txt

删除所有空行

sed '/^#/s/Q/DFF/g' a.txt

以#开头的行将所有的Q替换为DFF

sed 's/\$/\/' a.txt

在每个行尾都加上反斜杠 \

sed 's/^/\t/' a.txt

在每行前插入一个制表符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

sed '/[a-z]/!d' a.txt

输出所有含小写字母的行(不删除)

**Fudan UNI
1230713;
FUDAN UNI
hardware file**



**Fudan UNI
hardware file**

删除)
行到最后一行
释行)

sed '/^\$/d' a.txt

删除所有空行

sed '/^#/s/Q/DFF/g' a.txt

以#开头的行将所有的Q替换为DFF

sed 's/\$/\\/' a.txt

在每个行尾都加上反斜杠 \

sed 's/^/\t/' a.txt

在每行前插入一个制表符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

sed '/[a-z]/!d' a.txt

输出所有含小写字母的行(不删除)

sed '1,10d' a.txt

删除第1行到第10行

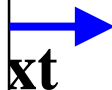
sed '5,\$d' a.txt

输出第1行到第4行(不删除)

sed '/^#/, \$d' a.txt

删除从第1个shell注释行到最后一行

**The Abstract
A novel design
Structured file
the methodology for**



**The Abstract
A novel design**

替换为DFF

替换为DFF

杠 \

sed 's/^\\t/' a.txt

在每行前插入一个制表符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

sed '/[a-z]/!d' a.txt

输出所有含小写字母的行(不删除)

sed '1,10d' a.txt

删除第1行到第10行

sed '5,\$d' a.txt

输出第1行到第4行(不删除)

sed '/^#/, \$d' a.txt

删除从第1个shell注释行到最后一行

sed '/\/* /d' a.txt

删除含/*的行(C语言注释行)

```
#include<stdio.h>
/* main function */
void main()
```

xt



```
#include<stdio.h>
void main()
```

替换为DFF
杠\
符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

sed '/[a-z]/!d' a.txt

输出所有含小写字母的行(不删除)

sed '1,10d' a.txt

删除第1行到第10行

sed '5,\$d' a.txt

输出第1行到第4行(不删除)

sed '/^#/, \$d' a.txt

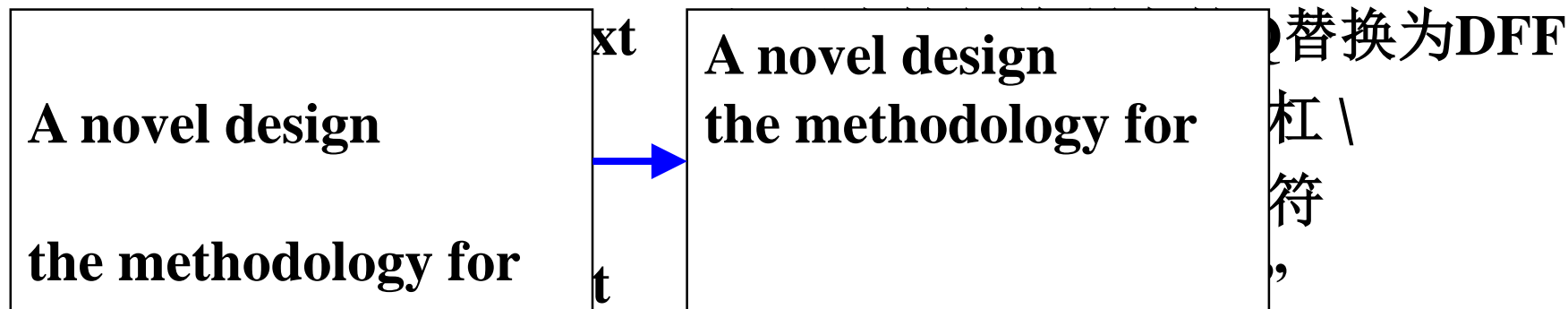
删除从第1个shell注释行到最后一行

sed '/\/* /d' a.txt

删除含/*的行(C语言注释行)

sed '/^\$/d' a.txt

删除所有空行



sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt	删除所有含字符串The的行
sed '/[0-9]/d' a.txt	删除所有含有数字的行
sed '/[a-z]/!d' a.txt	输出所有含小写字母的行(不删除)
sed '1,10d' a.txt	删除第1行到第10行
sed '5,\$d' a.txt	输出第1行到第4行(不删除)
sed '/^#/, \$d' a.txt	删除从第1个shell注释行到最后一行
sed '/\/* /d' a.txt	删除含/*的行(C语言注释行)
sed '/^\$/d' a.txt	删除所有空行
sed '/^#/s/Q/DFF/g' a.txt	以#开头的行将所有的Q替换为DFF
sed 's/\$/\\/' a.txt	在每个行尾都加上反斜杠 \

The Abstract
A novel design
TheSstructured

t
→
/' a.

The Abstract\
A novel design\
TheSstructured\
符
,
“fprintf(stderr,”

【例2-14】 sed命令示例（以a.txt为输入文件）。

sed '/The/d' a.txt

删除所有含字符串The的行

sed '/[0-9]/d' a.txt

删除所有含有数字的行

sed '/[a-z]/!d' a.txt

输出所有含小写字母的行(不删除)

sed '1,10d' a.txt

删除第1行到第10行

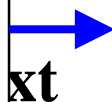
sed '5,\$d' a.txt

输出第1行到第4行(不删除)

sed '/^#/, \$d' a.txt

删除从第1个shell注释行到最后一行

Abstract
Novel design
Structured
methodology



Abstract
Novel design
Structured
methodology

释行)

替换为DFF

杠 \

sed 's/^/\t/' a.txt

在每行前插入一个制表符

sed 's/if□[/if□(/g' a.txt

将“if□[”替换为 “if□(”

sed 's/printf(/f&stderr,/' a.txt

将“printf(”改为“fprintf(stderr,”

➤ 【例2-15】 改变文件后缀示例

题意要求：要求将当前目录下所有具有某个后缀的文件全部重新命名，改为另一个后缀。为简单起见，将后缀为“.a”的文件改为后缀为“.b”的文件。

例如，执行命令“ls *.a”，显示当前命令下的文件为：

abc.a

de.a

xyz.a

如果执行以下命令：

mv abc.a abc.b

mv de.a de.b

mv zxy.a zxy.b

可以改变文件的后缀，获得以下文件：

abc.b

de.b

zxy.b

这样的操作属于手动执行shell命令。利用sed命令，编写shell程序(脚本文件)，可以实现非交互操作(batch program，批处理程序)。这是UNIX系统文件管理的一种基本操作。

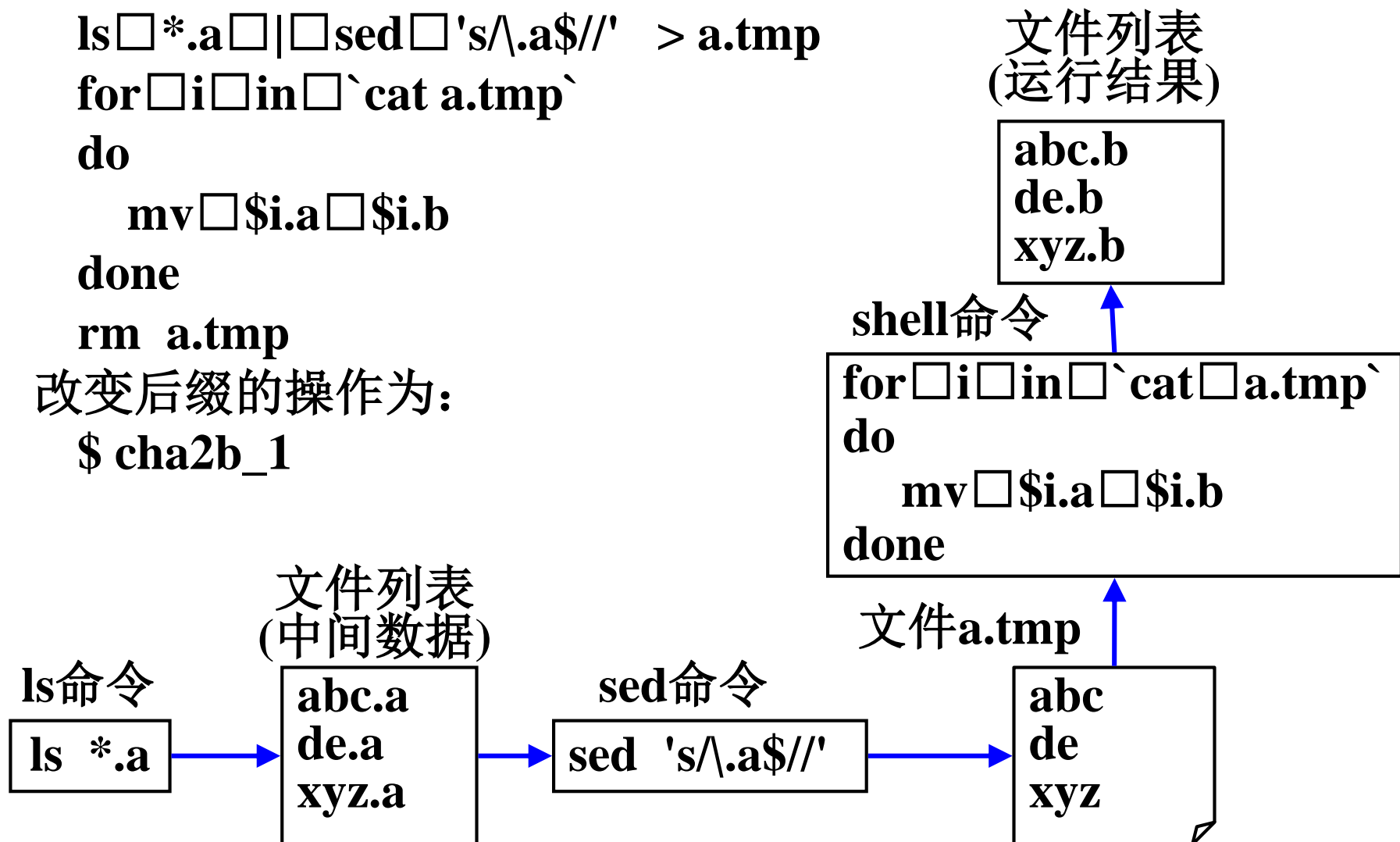
以下讨论和介绍几种利用sed命令编程的方案。

➤ 【例2-15】 改变文件后缀示例

➤ 方案1。编写shell文件cha2b_1。

```
ls *.a | sed 's/\.a$//' > a.tmp
for i in `cat a.tmp`
do
    mv $i.a $i.b
done
rm a.tmp
```

改变后缀的操作为：
\$ cha2b_1



➤ 【例2-15】 改变文件后缀示例

➤ 方案2。将cha2b_1改写为cha2b_2。

如果不希望产生临时文件a.tmp，可以把ls命令和sed命令产生的结果赋给一个变量，例如为list，则cha2b_2为：

```
list=`ls *.a | sed 's/\.a$//'`
```

```
for i in $list
```

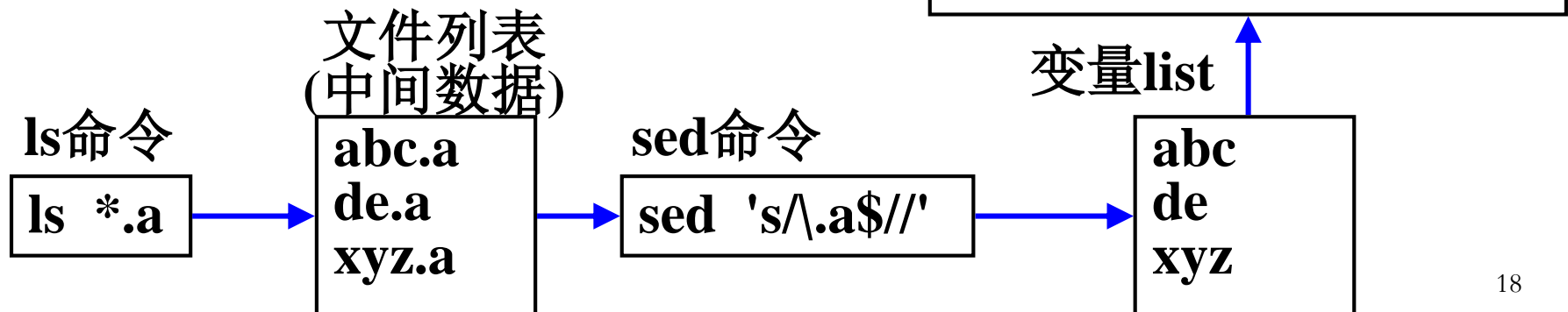
```
do
```

```
    mv $i.a $i.b
```

```
done
```

改变后缀的操作为：

```
$ cha2b_2
```



➤ 【例2-15】 改变文件后缀示例

➤ 方案3。将cha2b_2改写为cha2b_3。

考虑到for循环执行的命令为：

```
mv abc.a abc.b
```

```
mv de.a de.b
```

```
mv zxy.a zxy.b
```

可以用sed替代for循环来产生shell文件， cha2b_3的内容为：

```
ls *.a | \
```

```
sed 's/\.a$//' | \
```

(1)

```
sed 's/.* /mv &.a &.b/' > cha2b_3.sh
```

(2)

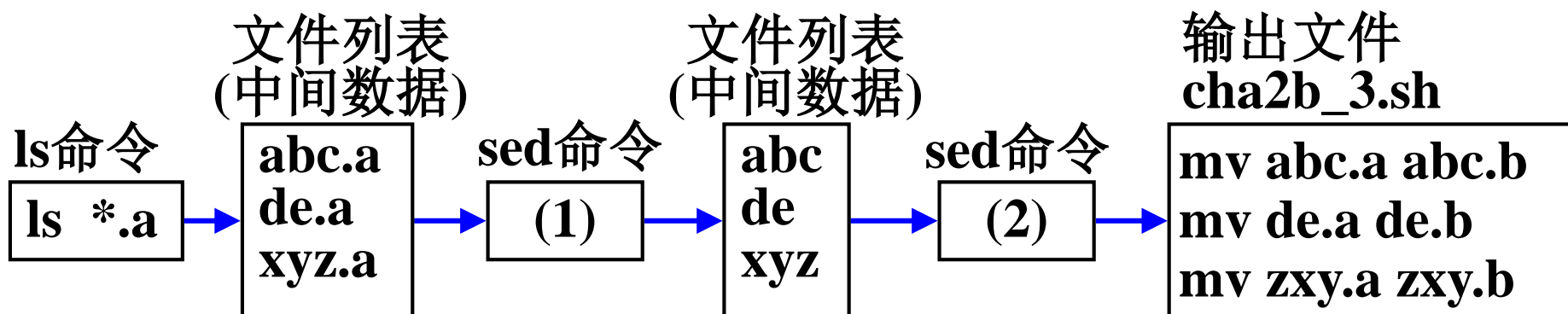
其中，“*****”表示一行中的所有文字，替换结果中的“**&**”是引用无名宏。执行命令为：

```
$ cha2b_3          生成shell文件cha2b_3.sh
```

```
$ cha2b_3.sh       运行shell文件cha2b_3.sh
```

➤ 【例2-15】 改变文件后缀示例

➤ 方案3。将cha2b_2改写为cha2b_3。（续）



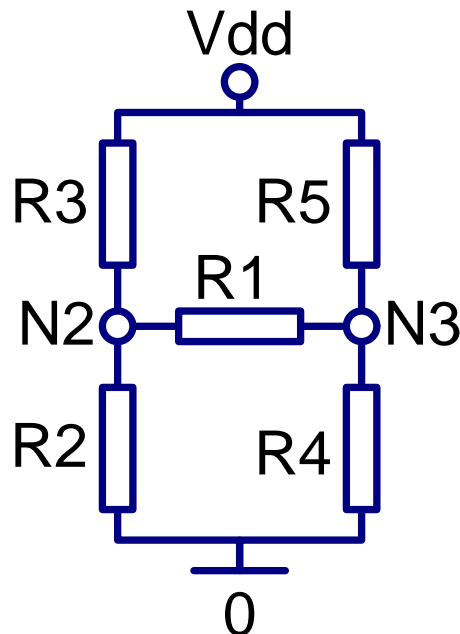
【例2-17】电路描述文件转换为电路网表文件

➤ 题意分析

定义电路描述文件或者电路网表文件都可以描述电路图。

例如，电路描述文件circuit描述了一个用节点名或者节点号混合命名节点的电路图，以及节点名及其节点号的定义。

```
Vdd□=□12  
R1□N2□N3  
R2□0□N2  
N2□=□9  
R3□Vdd□N2  
R4□N3□0  
N3□=□20  
R5□N3□Vdd
```



```
Vdd□=□12  
R1□N2□N3
```

节点号定义
电阻链接信息

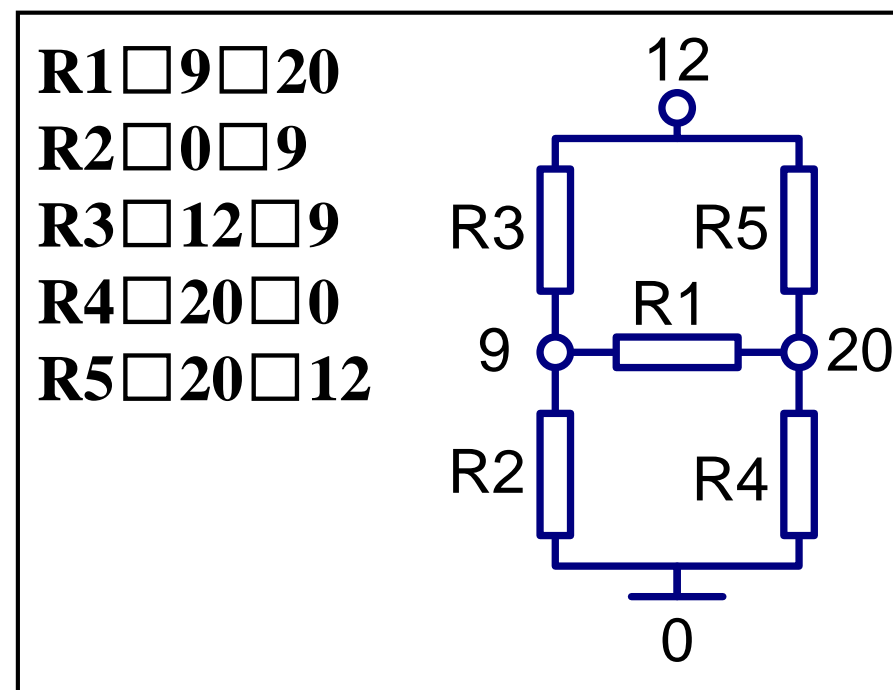
电路描述文件circuit及其电路图

【例2-17】电路描述文件转换为电路网表文件

➤ 题意分析（续）

定义电路描述文件或者电路网表文件都可以描述电路图。

例如，电路网表文件net描述了全部用节点号来描述的电路图。



电路网表文件net及其电路图

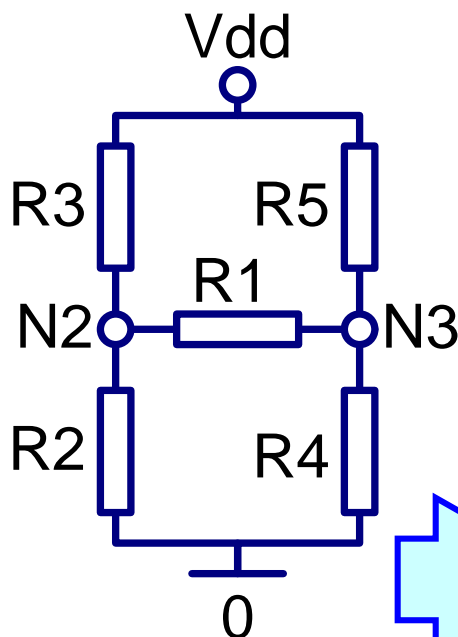
【例2-17】电路描述文件转换为电路网表文件

► 题意分析（续）

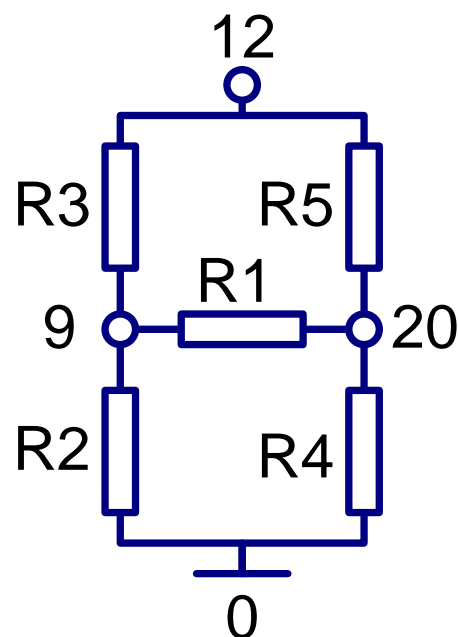
本题用sed编程，根据电路描述文件中关于节点名的定义，将其转换为节点号，从而生成电路网表文件。

例如，将节点名Vdd、N2和N3分别转换为节点号12、9和20。

```
Vdd□=□12  
R1□N2□N3  
R2□0□N2  
N2□=□9  
R3□Vdd□N2  
R4□N3□0  
N3□=□20  
R5□N3□Vdd
```



```
R1□9□20  
R2□0□9  
R3□12□9  
R4□20□0  
R5□20□12
```



电路描述文件circuit及其电路图

电路网表文件net及其电路图

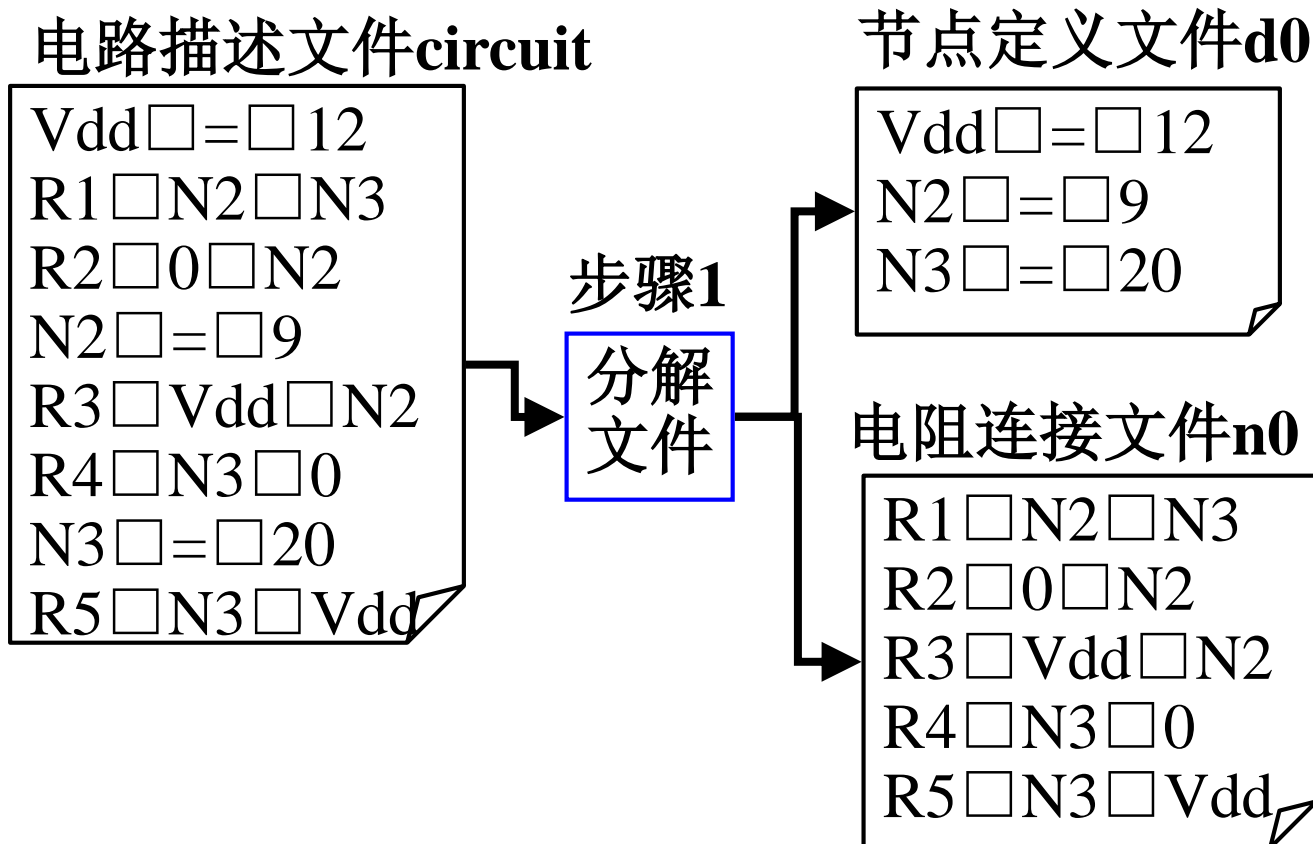
➤ 方案1

步骤1: 分解电路描述文件。

命名节点定义文件和电阻连接文件分别为d0和n0。执行sed命令如下:

```
$ sed '/=!d' < circuit > d0
```

```
$ sed '/=/d' < circuit > n0
```



➤ 方案1

步骤2: 根据节点定义文件，生成sed命令文件，目的在于此后将电阻连接文件中的节点名替代为节点号。

执行sed命令为：

```
$ sed 's/^/s\\/' <d0 | \
```

行首加 “s/”

```
> sed 's/□=□/\\/' | \
```

等号改为 “/”

```
> sed 's/$/\\/' >s0
```

行尾加 “/”

以期获得sed命令文件s0的内容为：

```
s/Vdd/12/
```

```
s/N2/9/
```

```
s/N3/20/
```

节点定义文件d0

```
Vdd□=□12  
N2□=□9  
N3□=□20
```

步骤2

生成
s0

sed命令文件s0

```
s/Vdd/12/  
s/N2/9/  
s/N3/20/
```

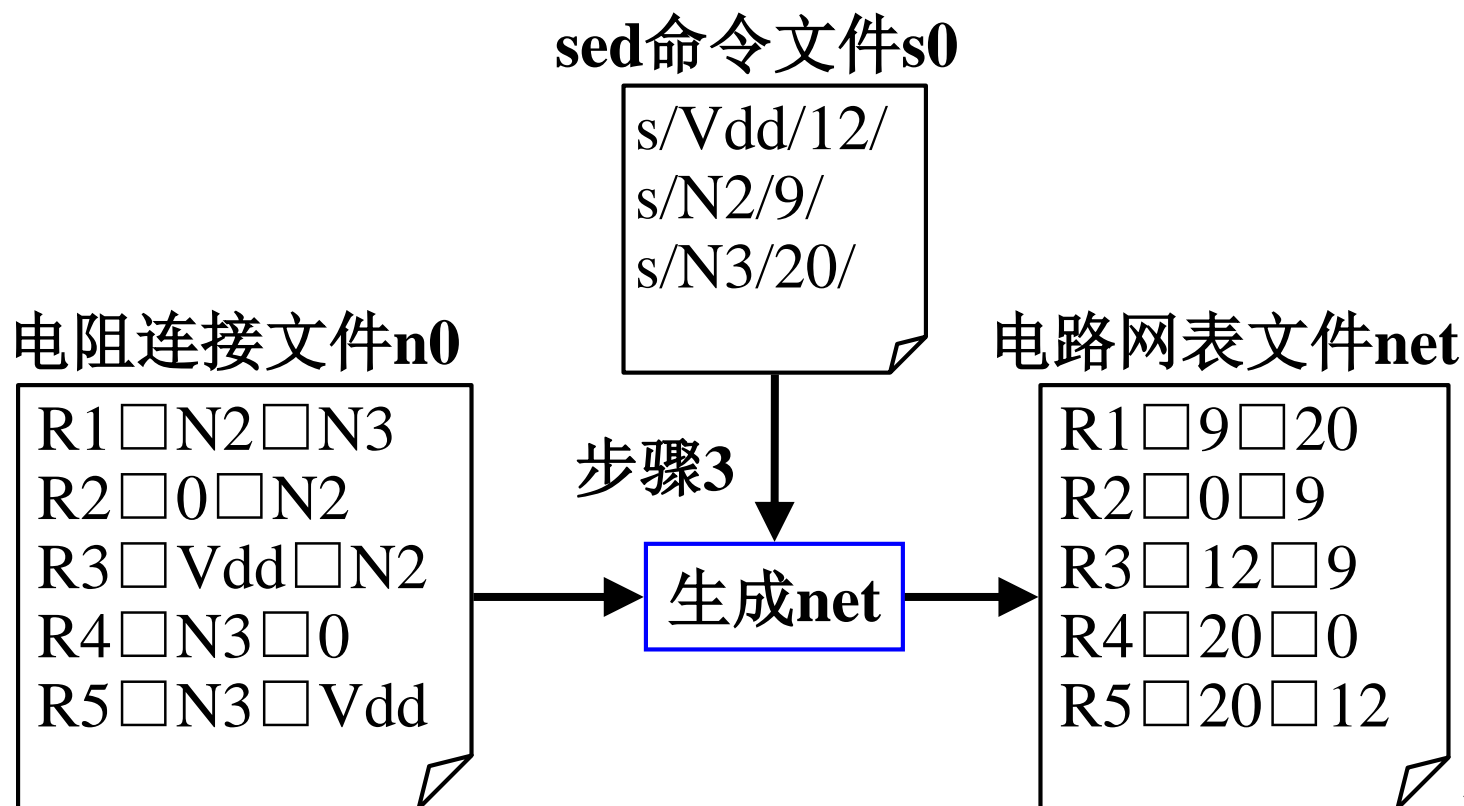
➤ 方案1

步骤3: 用sed命令文件s0, 将电阻连接文件n0转换为电路网表文件net。

执行sed命令为:

```
$ sed -f s0 < n0 > net
```

以期获得电路网表文件net。

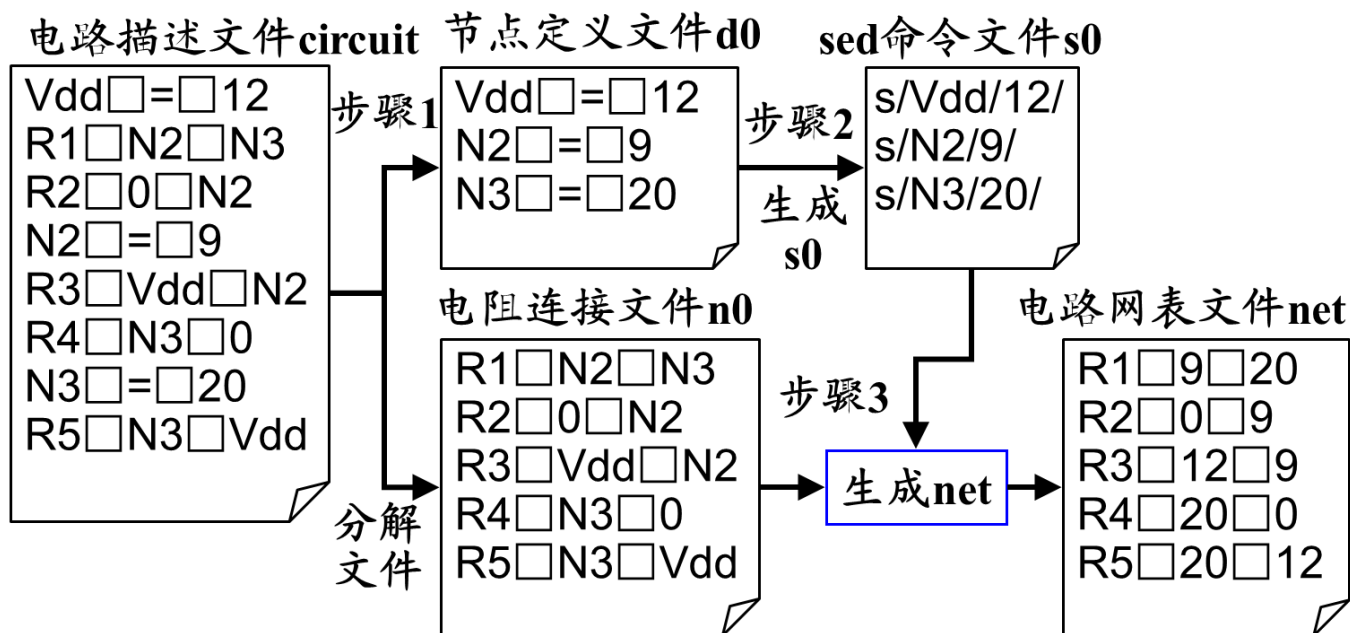


➤ 方案1

综合以上步骤，将所有的sed命令在一个shell文件中实现，取名为n2c_1，文件如下：

```
sed '/=!d' < $1 > $4  
sed '/=/d' < $1 > $5  
sed 's/^s\\/' < $4 |\  
sed 's/□=□/\\/' |\  
sed 's/$/\\/' > $3  
sed -f $3 < $5 > $2
```

运行shell文件n2c_1的命令为：
\$ n2c_1 circuit net s0 d0 n0



➤ 方案2

利用shell的流水线操作，用两个步骤生成电路网表文件。

步骤1: 分解电路描述文件，生成sed命令文件s0。

执行sed命令如下：

```
$ sed '/=/'!d' < circuit | \                               省去了中间文件d0  
> sed 's/\(.*)\[\]=\[\]\(.*)/s/\1\2\1/' > s0
```

其中的两个“\(.*)”都是定义有名宏，“\1”和“\2”是引用有名宏，三个“\”都是将斜杠转义。

电路描述文件circuit

```
Vdd\[\]=\12  
R1\[\]N2\[\]N3  
R2\[\]0\[\]N2  
N2\[\]=\9  
R3\[\]Vdd\[\]N2  
R4\[\]N3\[\]0  
N3\[\]=\20  
R5\[\]N3\[\]Vdd
```

分解
文件

节点
定义
信息

生成
s0

sed命令文件s0

```
s/Vdd/12/  
s/N2/9/  
s/N3/20/
```

➤ 方案2

步骤2: 分解电路描述文件，获得电阻连接信息。

用sed命令文件s0，将其转换为电路网表文件net。

执行sed命令为：

```
$ sed '/=/d' < circuit | \  
> sed -f s0 > n0
```

省去了中间文件n0

电路描述文件circuit

```
Vdd = 12  
R1 N2 N3  
R2 0 N2  
N2 = 9  
R3 Vdd N2  
R4 N3 0  
N3 = 20  
R5 N3 Vdd
```

分解
文件

电阻
连接
信息

sed命令文件s0

```
s/Vdd/12/  
s/N2/9/  
s/N3/20/
```

生成net

电路网表文件net

```
R1 9 20  
R2 0 9  
R3 12 9  
R4 20 0  
R5 20 12
```

➤ 方案2

综合以上步骤，采用流水线方法，以省去中间临时文件(节点定义文件和电阻连接文件)的产生，并且将所有的sed命令在一个shell文件中实现，取名为n2c_2，文件如下：

`sed '/=!d' < $1 \|` 用流水线省去节点定义文件d0

`sed 's/(.*)□=□\(.*)/s/V\1/V\2/' > $3`

`sed '/=/d' < $1 \|` 用流水线省去电阻连接文件n0

`sed -f $3 > $2`

运行shell文件n2c_2的命令为：

`$ n2c_2 circuit net s0`

电路描述文件circuit

```
Vdd□=□12
R1□N2□N3
R2□0□N2
N2□=□9
R3□Vdd□N2
R4□N3□0
N3□=□20
R5□N3□Vdd
```

步骤1

分解
文件

节点
定义
信息

步骤1

生成
s0

sed命令文件s0

```
s/Vdd/12/
s/N2/9/
s/N3/20/
```

步骤2

生成net

电路网表文件net

```
R1□9□20
R2□0□9
R3□12□9
R4□20□0
R5□20□12
```

➤ 作 业

- 习题 2-14.1, 2-14.2
- 上机及习题 2-4, 2-5, 2-6, 2-7