

**RED HAT®  
TRAINING**



# Red Hat

## **RH124 红帽系统管理 I**

**RH124-04-创建、查看和编辑文本文件**



# Red Hat

## 一、将输出重定向到文件或程序

# 标准输入、标准输出和标准错误输出

## 交互式硬件设备：

标准输入：从该设备接受用户输入的数据

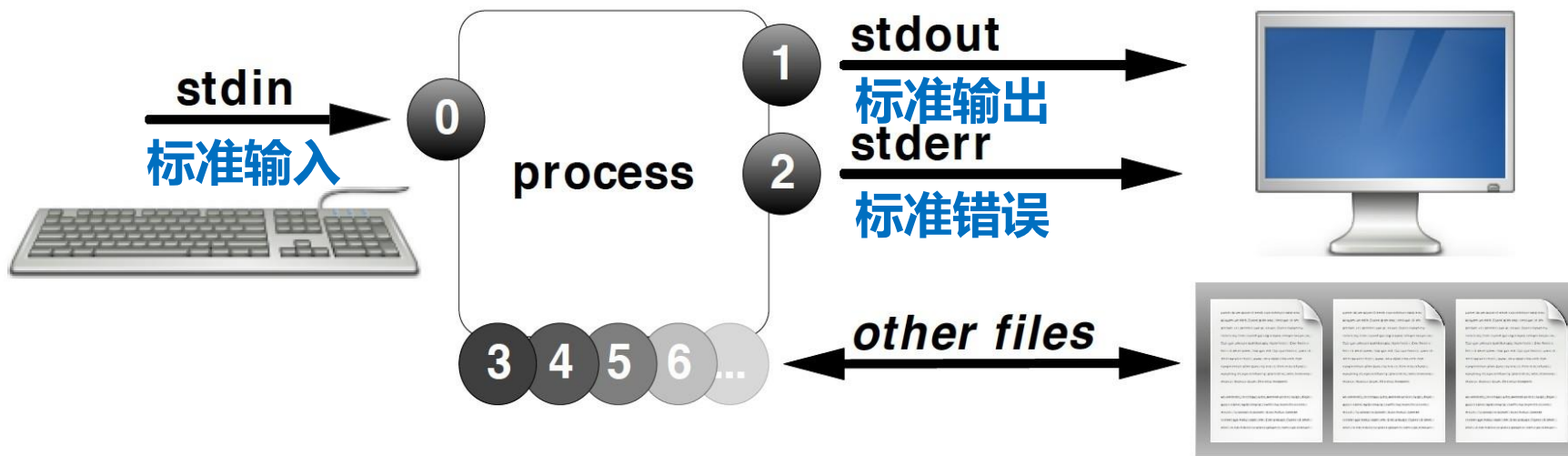
标准输出：通过该设备向用户输出的数据

标准错误输出：通过该设备报告执行出错信息

类型	设备文件	文件描述编号	默认设备
标准输入	/dev/stdin	0	键盘
标准输出	/dev/stdout	1	显示器
标准错误输出	/dev/stderr	2	显示器

# 标准输入、标准输出和标准错误

- ◆ 构建了一个带有编号标记的通道（文件描述符）的进程结构来管理打开的文件。通过用于通道**0、1、2**（标准输入、标准输出和标准错误）的默认连接创建进程
- ◆ 进程使用**3号及以上编号**的通道连接其他文件



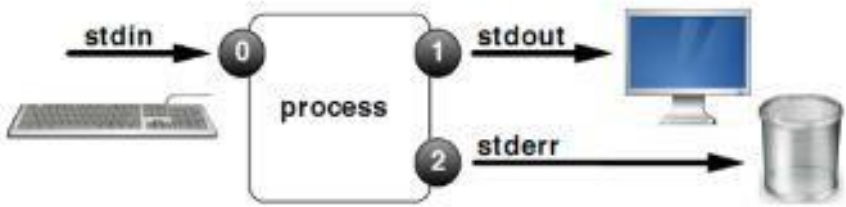
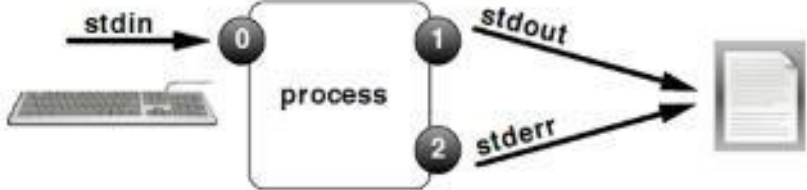
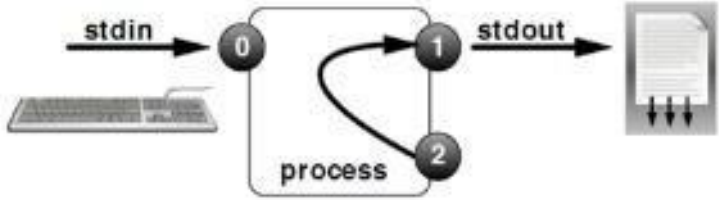
# 重定向输出到文件-1

- ◆ **> file**                      标准输出到文件
- ◆ **>> file**                    标准输出追加到文件
- ◆ **2> file**                    标准错误输出到文件

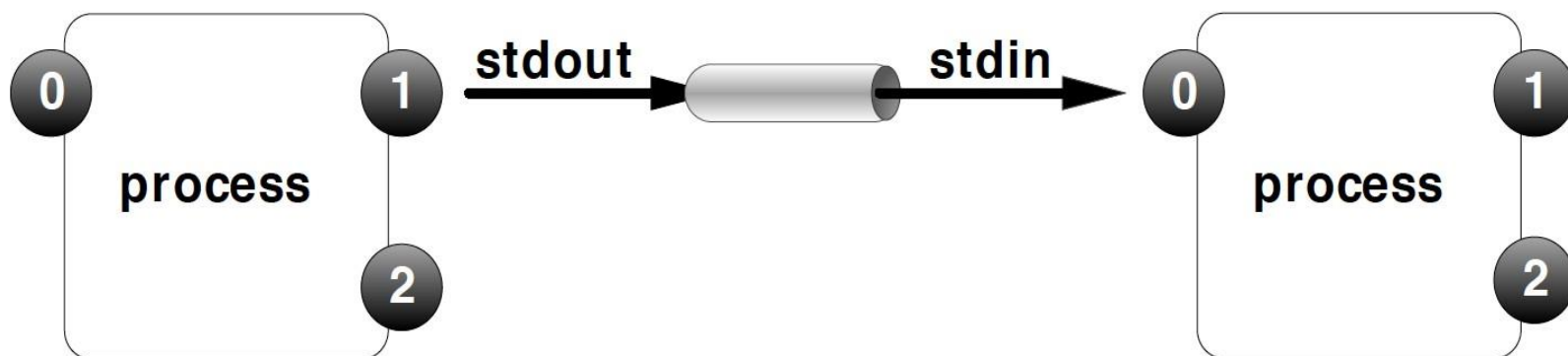
<code>&gt;file</code>	redirect <b>stdout</b> to a file <sup>(1)</sup>	
<code>&gt;&gt;file</code>	redirect <b>stdout</b> to a file, append to current file content <sup>(2)</sup>	
<code>2&gt;file</code>	redirect <b>stderr</b> to a file <sup>(1)</sup>	

## 重定向输出到文件-2

- ◆ `2>/dev/null` 丢弃标准错误输出
- ◆ `&>file` 标准输出和标准错误输出都重定向到文件
- ◆ `>>file2>&1` 合并标准输出和错误输出并追加到文件  
或 `&>>file`

<code>2&gt;/dev/null</code>	discard <b>stderr</b> error messages by redirecting to <b>/dev/null</b>	
<code>&amp;&gt;file</code>	combine <b>stdout</b> and <b>stderr</b> to one file <sup>(1)</sup>	
<code>&gt;&gt;file 2&gt;&amp;1</code>	combine <b>stdout</b> and <b>stderr</b> , append to current file content <sup>(2) (3)</sup>	

# 构建管道



## 实例：

`ls -l /usr/bin | less`

#分页命令的长输出

`ls -l | wc -l >/tmp/count`

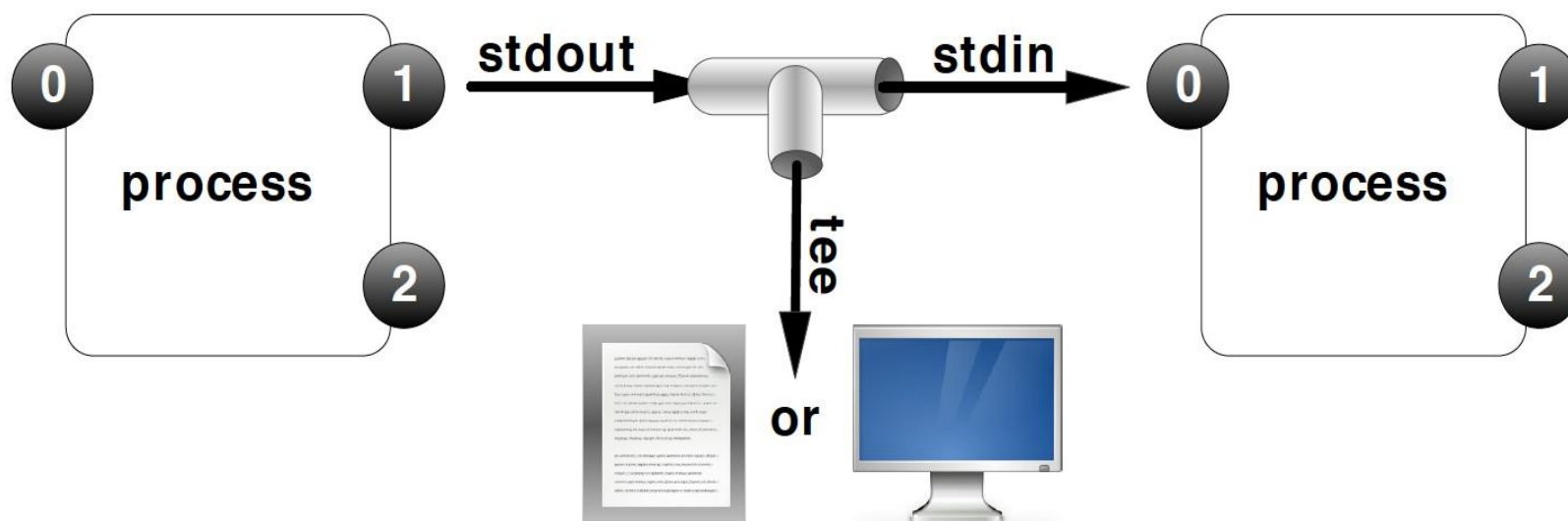
#计算输出或列表中的行数

`ls -t | head -n 10 >/tmp/a`

#抓取命令输出的前几行、后几行或选定行



# tee



**tee**:将标准输出结果同时输出到屏幕和文件

**实例：**

```
ls -l | tee /tmp/1.txt
```





redhat®

## 二、从shell提示符编辑文本文件

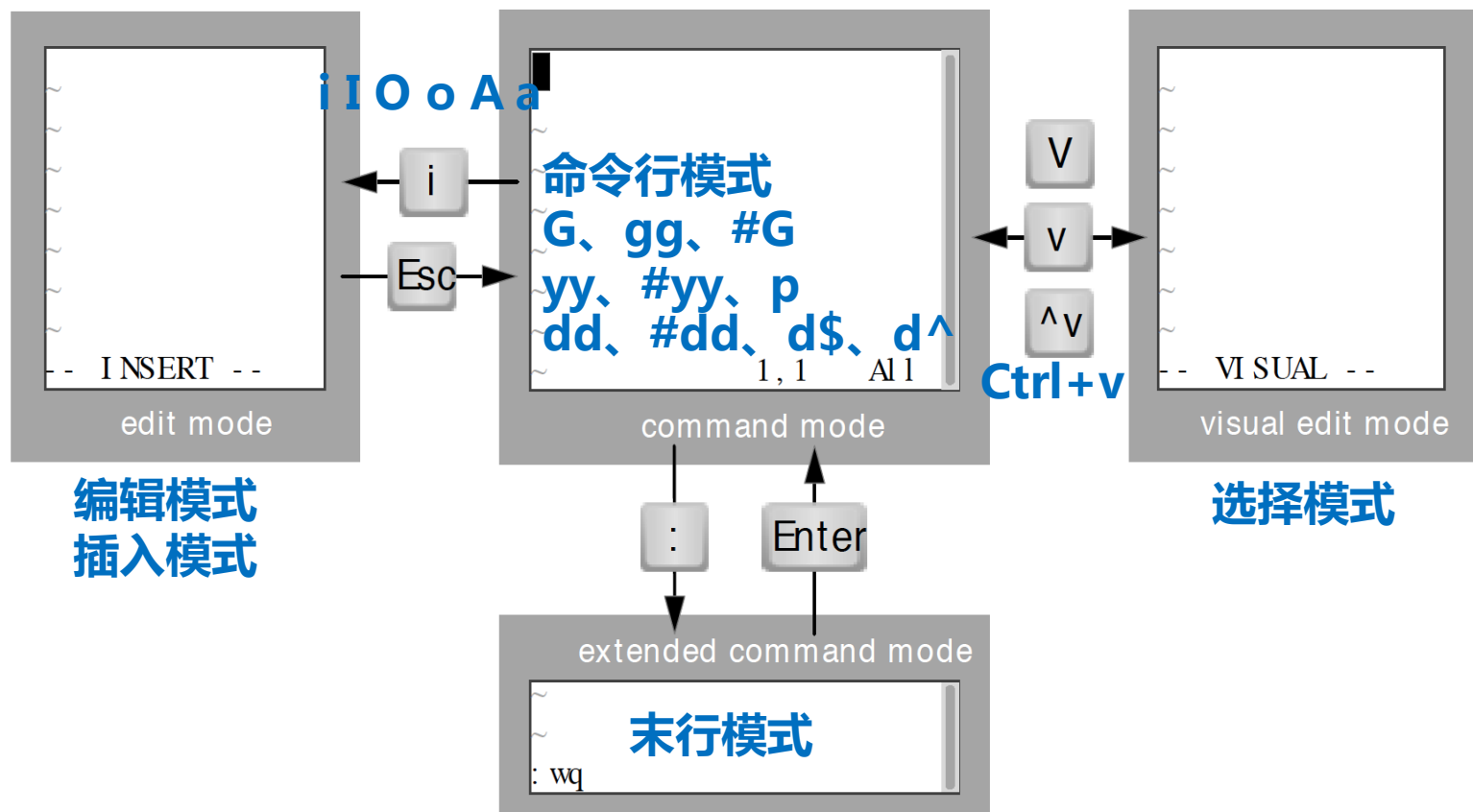
# 使用vim编辑器

- ◆ Linux下我们常用的编辑工具：  
VIM  
EMACS
- ◆ 作为一个编辑器，vi被广泛地运用在各种Unix操作系统上。  
Vi是Linux中的标准文本编辑器。VIM是vi的改进版

**打开文件命令：** **vi m** 文件名

- ◆ 如果文件已存在，则此文件被打开且显示文件内容
- ◆ 如果文件不存在，则vim在第一次存盘时自动建立在硬盘上

# 模式间切换



# 命令行模式

移动光标:

- ◆ h:左 j:下 k:上 l:右
- ◆ w:按单词移动光标到下个单词
- ◆ b:按单词移动光标到上个单词
- ◆ W:下个单词的首字母
- ◆ B:按照字符串块来移动, 并且是移动到每个块的最前
- ◆ E/e:下个单词的尾字母
- ◆ O/\$: 光标移动到当前行的第一个/最后一个字符(可以为空字符,注意与-区分)
- ◆ gg:直接跳到第一行行首
- ◆ G:跳到最后一行行首
- ◆ #G: 跳转到第#行, #代表行号

# 命令行模式

- ◆ yy : 复制当前行
- ◆ #yy : 复制#行, #为行数
- ◆ p : 粘贴
- ◆ dd : 删除当前行, 相当于剪切
- ◆ #dd : 删除#行, #为行数
- ◆ d\$ : 删除至行尾
- ◆ d^ : 删除至行首
- ◆ u : 撤销上一步操作
- ◆ U : 一次性撤销当前行的所有更改
- ◆ <ctrl-r> : 反撤销
- ◆ /xxx : 向后查找xxx内容
- ◆ n : 向下查找
- ◆ N : 向上查找

# 插入模式

- ◆ i : 光标所在位置插入
- ◆ I : 行首插入
- ◆ a : 光标后面插入
- ◆ A : 行尾插入
- ◆ o : 当前光标下一行插入
- ◆ O : 当前光标上一行插入

# 末行模式-保存于退出

从命令模式下输入 “ : ”

- ◆ :w 保存当前文件
- ◆ :q 如果未对文件做改动则退出
- ◆ :wq或:x或ZZ 保存当前文件并退出
- ◆ :q! 放弃存储并退出
- ◆ w /path 另存为
- ◆ :e! 放弃所有更动，重新编辑



# 末行模式-替换

- ◆ :s/old/new/g      替换当前行整行
- ◆ :%s/old/new/g      全文替换
- ◆ n,ms/old/new/g      替换n~m行

# 选择模式

命令模式下 **v|V|Ctrl+v**进入选择模式：

- ◆ 移动光标以选定操作块
- ◆ **d**剪切选定块或 **y**复制选定块
- ◆ p（小写）将选定内容贴在光标所在位置 **右边**
- ◆ P（大写）将选定内容贴在光标所在位置 **左边**

## 随堂练习

- 1、将root家目录中所有的文件及目录的详细信息，包括隐藏文件和目录信息，重定向到lab\_vim.txt文件中。
- 2、使用vim编辑lab\_vim.txt文件。
- 3、删除文件第三行。（命令行模式dd）
- 4、删除所有行的时间列。（ctrl+v 选择时间列 d删除）
- 5、复制最后3行并粘贴到文件的开头。（3yy 复制三行 P粘贴）
- 6、将文件中所有的root替换成tom。（末行模式 %s/root/tom/g）
- 7、保存并退出。（末行模式wq）
- 8、将当前时间追加到lab\_vim.txt文件中。
- 9、查看文件并确认操作。

微思网络----福建IT精英的发源地！

