



# U2 Linux Shell 基础

姜厚云

jianghy@dlut.edu.cn



# 课程目标

---

- 了解 Linux 文件系统结构
- 掌握 Linux Shell 基本格式和常用命令
- 掌握 Linux 文件管理
- 掌握 Linux Shell 元字符
- 掌握 Linux 常用工具
- 了解 Linux Shell 编程

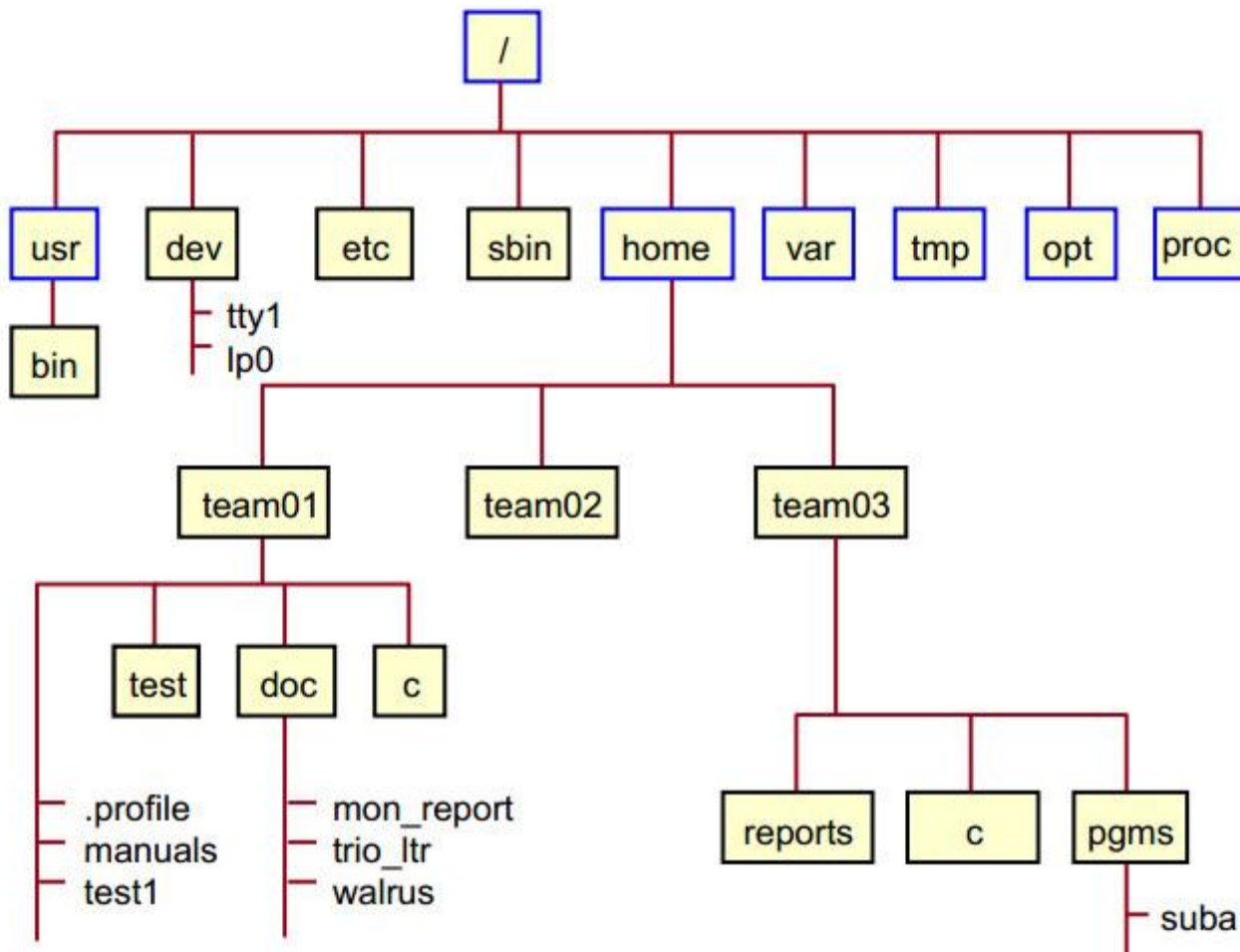


# Linux 文件类型

- 普通文件
  - /etc/hosts
- 目录文件
  - /home
- 特殊文件
  - /dev/cdrom



# Linux 文件系统目录结构





# Linux 文件访问

- **绝对**路径 (以 / 开头访问文件)
  - `/home/team01/test1`
  - `/home/team03/pgms/suba`
- **相对**路径 (假设当前目录是 `/home/team01`)
  - `test1` 或者 `./test1` (点号)
  - `../team03/pgms/suba` (双点号)
  - `.././dev/lp0`



# Shell 命令基本格式

➤ Command [ options ] [ arguments ]

例如:      ls

ls /

ls -l

ls -l -a 或 ls -la

ls -l /



# man 帮助手册

---

- 格式: `man Command`
- 例如: `man ls`



# 常用命令

- `logout` 、 `exit`、 `ctrl+d`
- `passwd`
- `date [ +%Y%m%d ]`
- `cal [ [ month ] year ]`
- `echo`
- `clear`





# 目录命令

- pwd
- ls [ -l | -a | -R | -d ]
- cd [ - | ~ ]
- mkdir [ -p ]
- rmdir



# 文件命令

---

- touch
- cp [ -r ]
- mv
- cat [ -v ] [ -t ] [ -e ]
- more
- rm [ -r ]
- ln [ -s ]



# 文本编辑器vi

➤ <http://www.runoob.com/linux/linux-vim.html>

version 1.1  
April 1st, 06  
翻译:2006-5-21

**Esc**  
命令  
模式

## vi / vim 键盘图

~ 切换大小写	! 外部过滤器	@ 运行宏	# prev ident	\$ 行尾	% 括号匹配	^ "软"行首	& 重复 :s	* next ident	( 句首	) 下一句首	"soft" bol down	+ 后一行行首
. 跳转到标注	1	2	3	4	5	6	7	8	9	0 "硬"行首	- 前一行行首	= 自动格式化
Q 切换至 ex 模式	W 下一单词	E 词尾	R 替换模式	T back 'till	Y 拷贝行	U 撤销行内命令	I 到行首插入	O 分段(前)	P 粘贴(前)	{ 段首	}	段尾
q 录制宏	w 下一单词	e 词尾	r 替换字符	t 'till	y 拷贝	u 撤销命令	i 插入模式	o 分段(后)	p 粘贴(后)	[ 杂项	]	杂项
A 在行尾附加	S 删除行并插入	D 删除至行尾	F 行内字符反向查找	G 文尾/行号	H 屏幕顶行	J 合并两行	K 帮助	L 屏幕底行	:	ex 命令	" 寄存器标识	行首/列
a 附加	s 删除字符并插入	d 删除	f 行内字符查找	g 附加命令	h ←	j ↓	k ↑	l →	;	重复 u/T/t/F	' 跳转到标注的行首	\ 未用!
Z 退出	X 退格	C 修改至行末	V 可视行模式	B 前一单词	N 查找上一处	M 屏幕中间行	< 反缩进	> 缩进	?	向前搜索		
Z 附加命令	X 删除(字符)	c 修改	v 可视模式	b 前一单词	n 查找下一处	m 设置标注	, 反向 u/T/t/F	.	重复命令	/	向后搜索	



# 文件权限

## ➤ chmod

- chmod 755 filename
- chmod u=rwx,g+w,o-rx filename

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
<b>d</b>	<b>rwX</b>			<b>r-X</b>			<b>r-X</b>		
目录 文件	读	写	执行	读	写	执行	读	写	执行

## ➤ chown

- chown hadoop filename
- chown -R hadoop /opt/hadoop

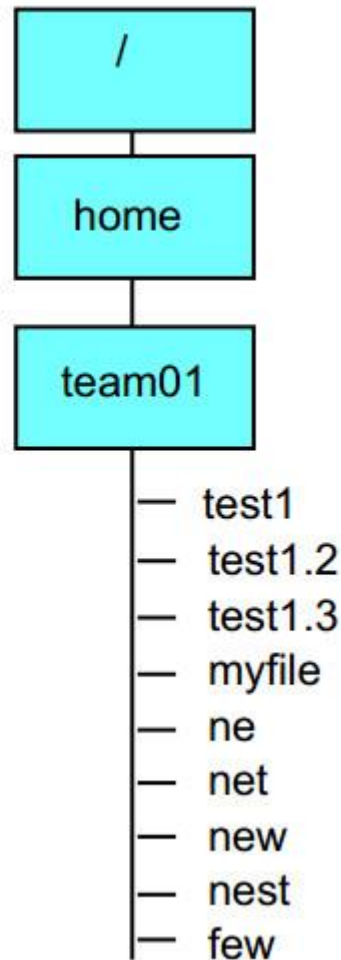


# 通配符

➤ \* ? [] [-] !

```
$ ls ne?  
net  new  
  
$ rm ?e?  
few  net  new
```

```
$ cp n* /tmp  
ne net  new  nest  
  
$ echo test1*  
test1  test1.2  test1.3
```





# 通配符

[ ] ! [-]

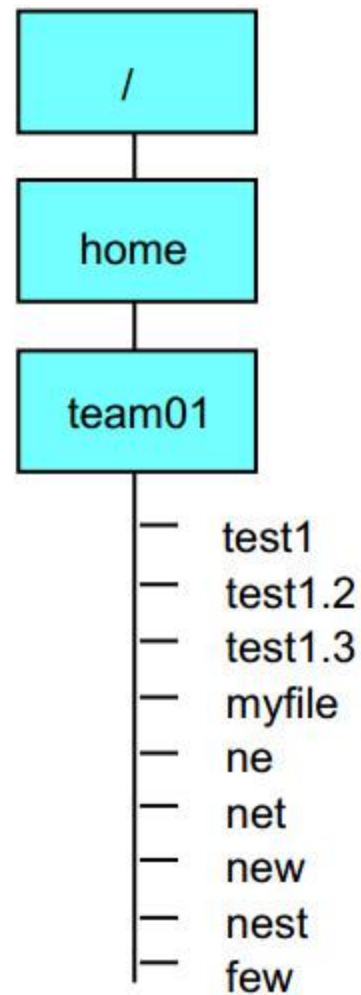
```
$ ls ne[stw]
net  new

$ rm [fghjdn]e[tw]
few  net  new

$ ls *[1-5]
test1  test1.2  test1.3

$ qprt [!tn]*
myfile  few

$ cat ?[!y]*[2-5]
test1.2  test1.3
```





# 重定向符

➤ `> >> 2> 2>> <`

➤ `date > date.log`

➤ `cat file1 > file2`

➤ `cat file1 file2 > file3`

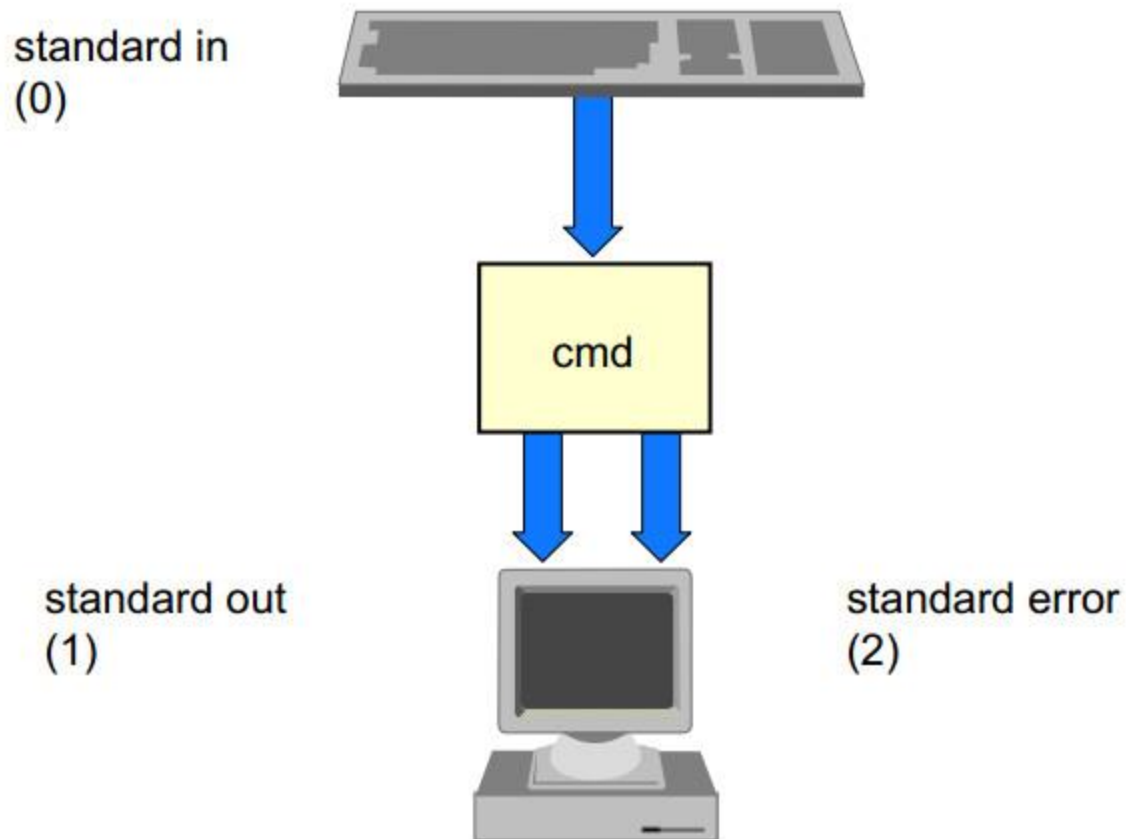
➤ `date >> date.log`

➤ `data 2> err.log`

➤ `data 2> /dev/null`



# 重定向符







# 命令连接符

- `command1 ; command2` (顺序执行)
  - `date ; cal`
- `command1 && command2` (条件执行)
  - `date && cal`
  - `data && cal`
- `command1 || command2` (条件执行)
  - `date || cal`
  - `data || cal`
- `command1 | command2` (管道)
  - `ls -R / | more`



# 变量

## ➤ 自定义变量

- `export JAVA_HOME=/usr/java/jdk_1.8.0-131`
- `echo $JAVA_HOME`

## ➤ 环境变量

- HOME、PWD、PATH、PS1、RANDOM .....
- `export PS1='[$PWD]> '`
- `Export PATH=$JAVA_HOME:$PATH`

## ➤ 查看和删除变量

- `set / unset`

## ➤ 只读变量

- `readonly`



# 命令替换

- 将命令的运行结果保存到变量或作为命令的参数
  - `variable=$( command )` 或者 `variable=`command``
- 例如:
  - `year=$( date +%Y )`
  - `sum=$( expr $a + $b )`
  - `ls -l $( which mkdir )`



# 进程

## The process environment

**Program**

**User and group ID**

**Data**

**Process ID (PID)**

**Open files**

**Parent Process ID (PPID)**

**Current directory**

**Program variables**

➤ **ps** [ -e -f ]



# 父子进程

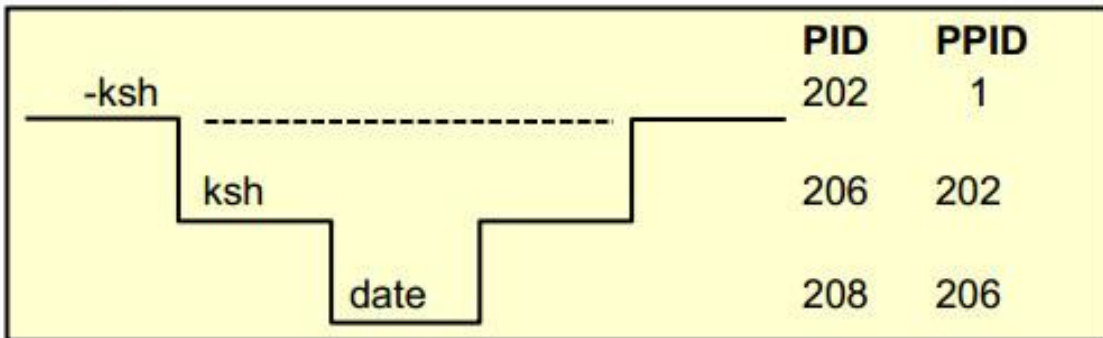
```
$ echo $$
202

$ ksh
                                (Create a subshell)

$ echo $$
206
$ date
                                (Run a command)
Tue Jan 4 11:18:26 GMT 2011
$ <ctrl-d>
                                (Exit the subshell)

$ echo $$
202
```

Subshell





# 进程管理

- kill
  - kill [ -9 ] pid
  - kill -l
- Ctrl + z
  - 发送SIGSTOP信号，进程只是被停止，再送SIGCONT信号，进程继续运行
- Ctrl + c
  - 发送SIGINT信号，默认进程会结束
- 后台进程 &
- jobs [ -l ]
- fg、bg



# 工具

- file
- wc
- alias
- find
- sort    head    tail
- grep
- tar
- gzip / gunzip



# 配置文件

---

- `$HOME/.bash_profile`
- `$HOME/.bashrc`
- `/etc/.bash_profile`
- `/etc/.bashrc`





# Shell 脚本语法

## ➤ 顺序

## ➤ 分支

➤ if elif else then test case esac

## ➤ 循环

➤ for while until do done

## ➤ 其它

➤ continue break



# 执行Shell 脚本

- `bash scriptname`
- `chmod +x scriptname`  
`./scriptname`
- `. scriptname`