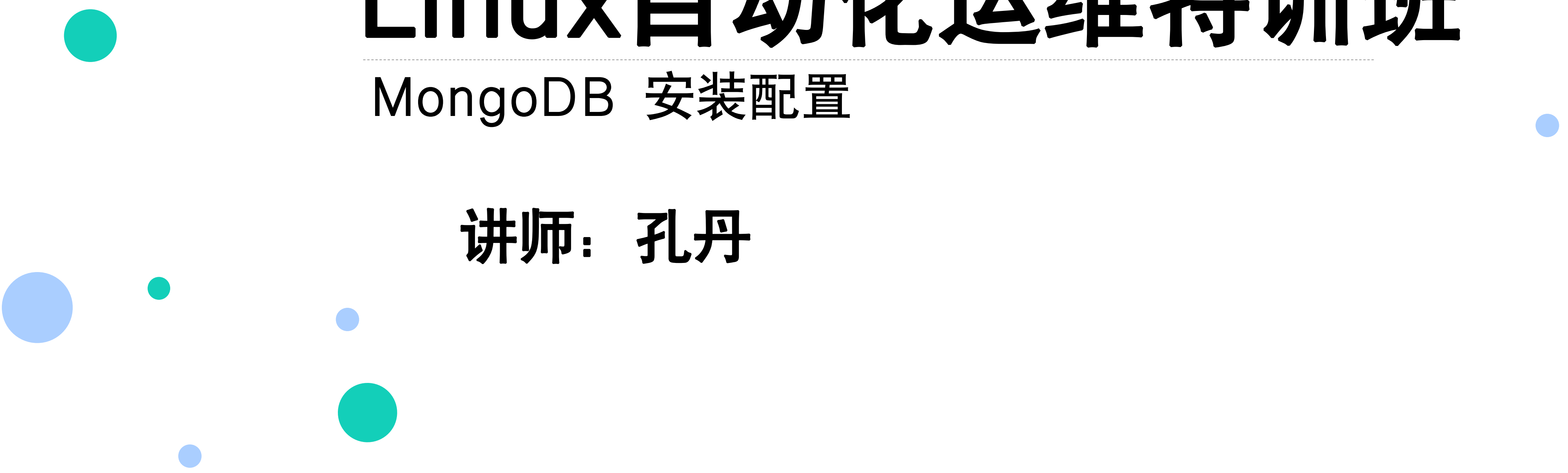




# Linux自动化运维特训班

Shell特性回顾

讲师：孔丹



# 大纲

➤ Bash 的基本功能

➤ 用户登录shell类型

# Bash 的基本功能

## □ 1、历史命令

### 1) 历史命令的查看

```
[root@localhost ~]# history [选项] [历史命令保存文件]
```

选项:

-c: 清空历史命令

-w: 把缓存中的历史命令写入历史命令保存文件。如果不手工指定历史命令保存文件, 则放入默认历史命令保存文件 ~/.bash\_history 中

```
[root@localhost ~]# vi /etc/profile
```

...

```
HISTSIZE=1000
```

为什么history 命令查看的历史命令和 ~/.bash\_history 文件中保存的历史命令是不同?

因为当前登录操作的命令并没有直接写入 ~/.bash\_history 文件, 而是保存在缓存当中的。需要等当前用户注销之后, 缓存中的命令才会写入 ~/.bash\_history 文件。如果我们需要把内存中的命令直接写入 ~/.bash\_history 文件, 而不等用户注销时再写入, 就需要使用 “-w” 选项了。

# Bash 的基本功能

## □ 1、历史命令

### 2) 、历史命令的调用

如果想要使用原先的历史命令有这样几种方法：

- 使用上、下箭头调用以前的历史命令

- 使用 “!n” 重复执行第 n 条历史命令

- 使用 “!!” 重复执行上一条命令

- 使用 “!字串” 重复执行最后一条以该字串开头的命令

- 使用 “!\$” 重复上一条命令的最后一个参数

## □ 2、命令与文件的补全

tab

Bash 命令补全增强软件包 bash-completion，这个包支持对  
systemctl 命令服务名称的补全

# Bash 的基本功能

## □ 3、命令别名

命令格式：

```
[root@localhost ~]# alias
```

#查询命令别名

```
[root@localhost ~]# alias 别名='原命令'
```

#设定命令别名

别名的优先级比命令高，那么命令执行时具体的顺序是什么呢？

命令执行时的顺序是？

- 1、第一顺位执行用绝对路径或相对路径执行的命令。
- 2、第二顺位执行别名。
- 3、第三顺位执行 Bash 的内部命令。
- 4 第四顺位执行按照\$PATH 环境变量定义的目录查找顺序找到的第一个命令。

为了让这个别名永久生效，可以把别名写入环境变量配置文件“~/.bashrc”。

# Bash 的基本功能

## □ 4、Bash 常用快捷键

ctrl+A 把光标移动到命令行开头

ctrl+E 把光标移动到命令行结尾

ctrl+C 强制终止当前的命令

ctrl+L 清屏，相当于 clear 命令

ctrl+U 删除或剪切光标之前的命令

ctrl+K 删除或剪切光标之后的内容

ctrl+Y 粘贴 ctrl+U 或 ctrl+K 剪切的内容

ctrl+R 在历史命令中搜索

ctrl+D 退出当前终端

ctrl+Z 暂停，并放入后台

ctrl+S 暂停屏幕输出

ctrl+Q 恢复屏幕输出

# Bash 的基本功能

## □ 5、输入输出重定向

### 1) Bash 的标准输入输出

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

### 2) 输入重定向

<



# Bash 的基本功能

## □ 5、输入输出重定向

### 3) 输出重定向

类 型	符 号	作用
标准输出重定向	命令 > 文件	以覆盖的方式，把命令的正确输出输出到指定的文件或设备当中。
	命令 >> 文件	以追加的方式，把命令的正确输出输出到指定的文件或设备当中。
标准错误输出重定向	错误命令 2>文件	以覆盖的方式，把命令的错误输出输出到指定的文件或设备当中。
	错误命令 2>>文件	以追加的方式，把命令的错误输出输出到指定的文件或设备当中。
正确输出和错误输出同时保存	命令 > 文件 2>&1	以覆盖的方式，把正确输出和错误输出都保存到同一个文件当中。
	命令 >> 文件 2>&1	以追加的方式，把正确输出和错误输出都保存到同一个文件当中。
	命令 &>文件	以覆盖的方式，把正确输出和错误输出都保存到同一个文件当中。
	命令 &>>文件	以追加的方式，把正确输出和错误输出都保存到同一个文件当中。
	命令>>文件 1 2>>文件 2	把正确的输出追加到文件 1 中，把错误的输出追加到文件 2 中



# Bash 的基本功能

## □ 6、命令执行顺序

； --命令的顺序执行# date ; ls -l /etc/passwd

&& --前面命令执行不成功，后面的命令不执行

# mkdir /mnt/iso && mount /dev/sr0 /mnt/iso

|| --如果前面命令成功，后面就不执行，如果前面不成功后面就执行

## □ 7、管道

当在两个命令之间设置管道时，管道符|左边命令的输出就变成了右边命令的输入。只要第一个命令向标准输出写入，而第二个命令是从标准输入读取，那么这两个命令就可以形成一个管道。

# Bash 的基本功能

## □ 8、通配符

? 匹配一个任意字符

\* 匹配 0 个或任意多个任意字符，也就是可以匹配任何内容

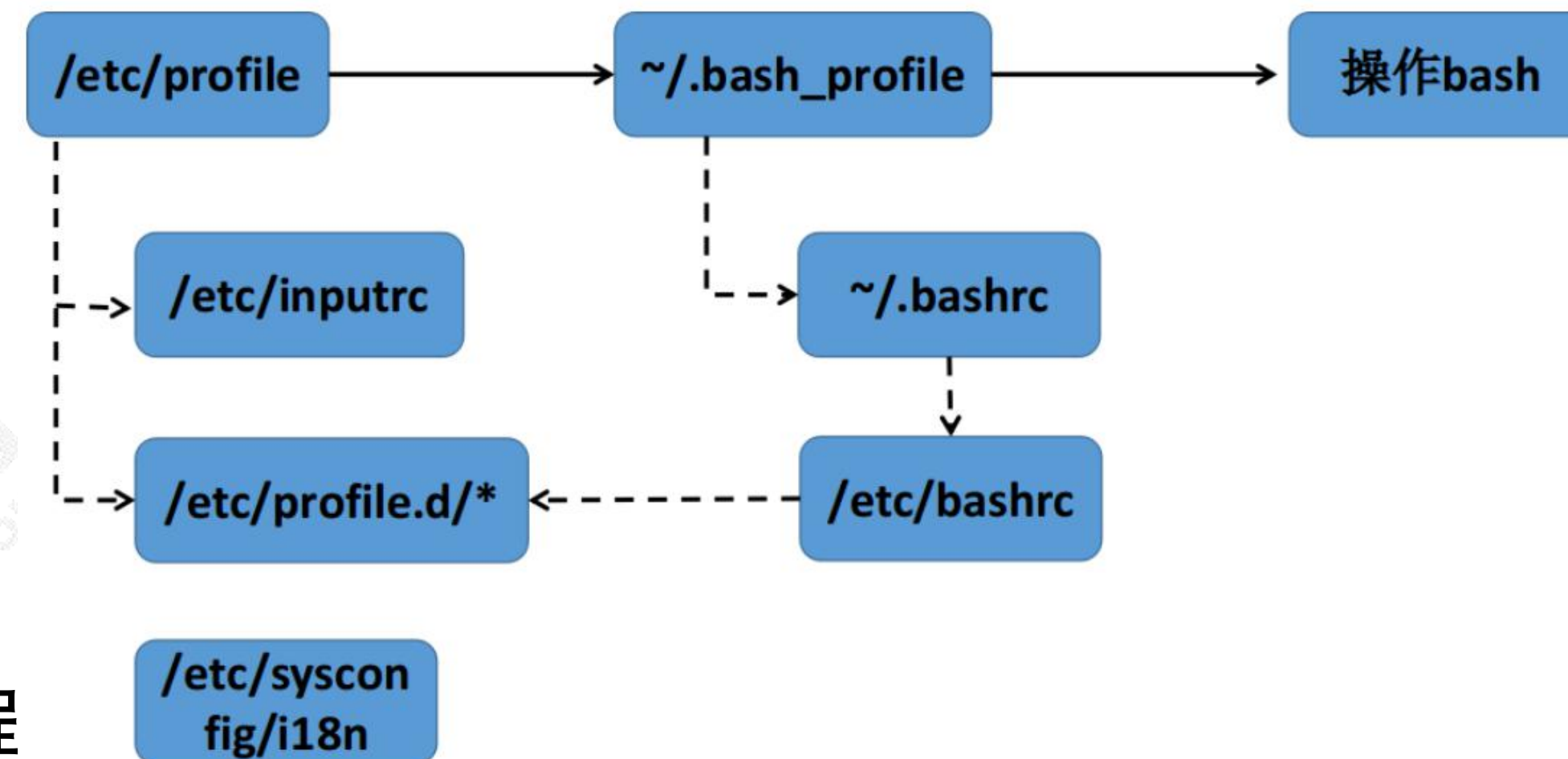
[] 匹配中括号中任意一个字符。

[-] 匹配中括号中任意一个字符， -代表一个范围。

[^] 逻辑非，表示匹配不是中括号内的一个字符。

# 用户登录shell类型

- login shell 取得bash 时需要完整的登录流程，需要读取完整的配置文件，例如我们通过tty1-tty6通过用户名和密码登录的shell就属于login shell，ssh 远程登录的用户的shell也是login shell。
- login shell 主要读取两个配置文件/etc/profile和~/.bash\_profile  
首先/etc/profile在系统登录的时候进行配置环境  
~/.bash\_profile是login shell 登录时进行读取，且该文件会读取  
~/.bash\_profile  
~/.bash\_login  
~/.profile



实线的方向是主线流程

# 用户登录shell类型

- non-login shell 取得bash的接口不需要重复登录，比如在图形化界面的ctl+alt+t出来的shell就属于non-login shell 直接在命令行中通过su username也属于non-login shell。
- 非登录式shell如何读取配置文件？  
~./bashrc-->/etc/bashrc-->/etc/profile.d/\*.sh

# 总结

□ shell基础回顾

# 作业

- 1、重定向和管道的区别？
- 2、查看指定程序httpd的进程运行状态，并将输出重定向到文件中。
- 3、统计系统中当前登录的用户数。
- 4、使用 mount 命令显示当前挂载的文件系统的信息，并使用 column 命令格式化列的输出，最后将输出结果保存到一个文件中。





# 谢谢观看

更多好课，请关注[万门大学APP](#)

