



北京邮电大学

Beijing University of Posts and Telecommunications

shell中的逻辑判断

蒋砚军 北京邮电大学计算机学院



► shell中的条件判断



■ 条件判断的依据

判定一条**命令**是否执行成功。方法:命令执行的返回码, 0表示成功, 非0表示失败。可以把命令执行结束后的“返回码”理解为“出错代码”

■ 命令执行结束后的返回码

主函数:

```
int main(void) { ... }
```

```
int main(int argc, char **argv) { ... }
```

```
int main(int argc, char **argv, char **env) { ... }
```

main()函数的返回值, 或者程序调用了系统调用exit(code)导致进程终止, exit函数的参数值code。取值0~255

如果代码中main()函数没有return一个确定的值, 返回码就是随机值, 不可用来做条件判断

► shell内部变量\$?



```
$ ls -d xyz
```

```
xyz
```

```
$ echo $?    $?上一命令的返回码，shell自定义变量
```

```
0
```

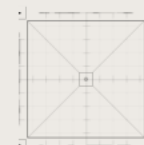
```
$ ls -d xyz1
```

```
xyz1: not found
```

```
$ echo $?
```

```
2
```

- 用管道线连接在一起的若干命令，进行条件判断时以最后一个命令执行的返回码为准



► 复合逻辑



用 **&&** 或 **||** 连结两个命令

可以利用复合逻辑中的“**短路计算**”特性实现最简单的条件

■ *cmd1* **&&** *cmd2*

若*cmd1*执行成功(返回码为0)则执行*cmd2*，否则不执行*cmd2*

■ *cmd1* **||** *cmd2*

*cmd1*执行失败(返回码不为0)则执行*cmd2*，否则不执行*cmd2*

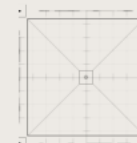
\$ **ls -d xdir >/dev/null && echo FOUND**

FOUND

若没有目录ydir

\$ **ls -d ydir >/dev/null 2>&1 || echo No ydir**

No ydir



► 命令true与false



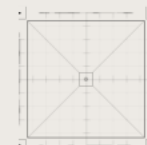
■ /bin/true

◆ 返回码总为0

■ /bin/false

◆ 返回码总不为0

■ 有的shell为了提高效率，将true和false设置为内部命令



► 自己编写的程序用作条件判断



```
$ cat odd.c
#include <stdlib.h>
int main(int argc, char **argv)
{
    int a = argc < 2 ? 0 : atoi(argv[1]);
    return a % 2 == 0 ? 1 : 0;
}
$ a=7
$ ./odd $a && echo ODD
ODD
$ a=90
$ ./odd $a && echo ODD
```



北京邮电大学

Beijing University of Posts and Telecommunications



谢谢