Linux 复习资料

目录

1 试卷题型.....3
2 第二章重点.....3
2.1 Linux 特点概述.....3
2.2 常用命令的功能与使用.....4
3 第三章重点.....6
3.1 vim 介绍.....6
3.2 vim 工作模式.....6
3.3 vim 使用方法.....6
4 第四章重点.....8
4.1 用户和组概念.....8
4.2 用户和组管理相关文件.....9
4.3 用户和组管理相关命令.....10
5 第五章重点.....10
5.1 基本权限的概念.....10
5.2 ACL 权限的概念.....11
5.3 特殊权限的概念.....12
5.4 权限的字符表示法与数字表示法.....13
5.5 权限管理命令与权限规划.....13
6 第六章重点.....14
6.1 文件系统的概念.....14
6.2 分区设备命名规则.....14
6.3 分区命令的基本功能.....14
7 第七章重点.....15
7.1 软件安装的两种模式.....15
7.2 软件安装常用命令的使用.....15
8 第八章重点.....16

8.1 shell 编程中各种变量的概念和作用 16
8.2 if 流程控制语句 17
8.3 数值判断与文件类型判断 18

1 试卷题型

题型	数量	分值
判断	10	1
单选	20	1.5
填空	10	1
简答	4	不明
操作	2	不明

2 第二章重点

2.1 Linux 特点概述

1. Linux 内核最初由李纳斯·托瓦兹（Linus Torvalds）编写，如今其开源。
2. Linux 严格区分大小写。
3. Linux 中所有内容（包括硬件设备）以文件形式保存。
4. Linux 不靠扩展名区分文件类型

通过权限位标识来确定文件类型，常见的文件类型只有普通文件、目录、链接文件、块设备文件、字符设备文件等。

5. 用户与 Linux 系统的交互

计算机由运算器、控制器、存储器、I/O 设备等共同组成，让各种硬件设备各司其职且有能协同运行的东西就是系统内核。Linux 系统的内核负责完成对硬件资源的分配、调度等管理任务。系统内核对计算机的正常运行来说非常重要，因此一般不建议直接去编辑内核中的参数，而是让用户通过基于系统调用接口开发出来的程序或服务来管理计算机，以满足日常工作的需要。

6. 终端提示符

[root@linuxprobe~]#

上述式子中，**[]**为提示符的分隔符，**root** 显示当前登录用户名，**@**分隔符，**linuxprobe** 为当前系统的简要主机名（此外还有主机全名），~表示用户当前所在目录（这里的~指用户家目录），**#**提示符，用于识别用户权限等级，**?**或**$**为普通用户，权限较小。

2.2 常用命令的功能与使用

1. 命令基本格式

命令名称 [命令参数选项] [命令作用对象]

命令名称：表达想要做的事情；

命令参数选项：用于对命令进行调整，参数可以使用长格式或短格式。

命令作用对象：一般指要处理的文件、目录、用户等资源名称。

2. 常用命令列表

命令名称	语法格式	功能描述
man	man [命令或配置文件]	获取对象的帮助信息手册
info	info [命令]	查看命令帮助信息
help	help [Shell 内置命令]	获取 shell 内置命令帮助信息
history	history [参数] [历史命令]	显示执行过的历史命令
ls	ls [参数] [文件名称]	用于显示目录中的文件信息
mkdir	mkdir [参数] 目录名称	创建空白的目录
rmdir	rmdir [目录名]	删除空目录
rm	rm [参数] 文件名称	删除文件或目录
cp	cp [参数] 源名 目标文件名	复制文件或目录
mv	mv 源名 目标文件名	剪切或重命名文件
touch	touch [参数] 文件名称	创建空白文件或设置时间
cat	cat [参数] 文件名称	查看纯文本文件（内容少）
more	more 文件名称	查看纯文本文件（内容多）
less	less 文件名称	浏览文本内容，类似 more
head	head [参数] 文件名称	查看纯文本文件前 n 行
tail	tail [参数] 文件名称	查看纯文本文件后 n 行
wc	wc [参数] 文件名称	统计文件行数、字或字节数
grep	grep [参数] 文件名称	按行提取文本内容
cut	cut [参数] 文件名称	按列提取文本内容
diff	diff [参数] 文件 A 文件 B	比较多个文件之间的内容差
uniq	uniq [参数] 文件名称	去除文本中连续的重复行

sort	sort [参数] 文件名称	对文本行进行再排序
pwd	pwd	显示当前所处的工作目录
whereis	whereis 命令名称	命令的二进制/源码/手册路径
which	which 命令名称	搜索二进制程序对应的位置
locat	locate 文件名称	按照搜索文件所对应的位置
find	find [查找范围] 寻找条件	按照条件查找文件位置
zip	zip [参数] 压缩包名 文件	创建 ZIP 格式压缩文件
unzip	unzip [参数] 压缩包名	解压 ZIP 格式文件
gzip	gzip [参数] 文件	压缩文件（生成.gz）
gunzip	gunzip [参数] 文件.gz	解压 GZIP 压缩文件
bzip2	bzip2 [参数] 被压缩文件	压缩文件（生成.bz2）
bunzip2	bunzip2 [参数] 文件.bz2	解压 bzip2 压缩文件
tar	tar [参数] 压文件 被压目录	打包/解包归档文件
echo	echo [参数] [字符串/&变量]	输出字符串到标准输出
date	date [+指定的格式]	显示或设置系统时间日期
reboot	reboot	重启系统
poweroff	poweroff	关闭系统电源
shutdown	shutdown [参数] 时间	定时关机/重启系统
logout	logout	退出当前登录的 Shell 会话
who	who	显示当前登录用户信息
last	last	显示用户登录历史记录
lastlog	lastlog [选项]	显示所有用户最近登录时间
wget	wget [参数] 网址	从网络下载文件
free	free [-h]	显示内存使用情况
ps	ps [参数]	显示进程状态信息
kill	kill [参数] PID	向进程发送信号终止进程
ifconfig	ifconfig [参数] [网络设备]	配置/显示网络接口参数
ping	ping [参数] IP 地址	测试网络连通性

3 第三章重点

3.1 vim 介绍

vim 是一个基于文本界面的编辑工具，使用简单，是所有 Linux 发行版本的默认文本编辑器。

3.2 vim 工作模式

1. 工作模式类型：命令模式、输入模式和末行模式。
2. 三种工作模式的关系

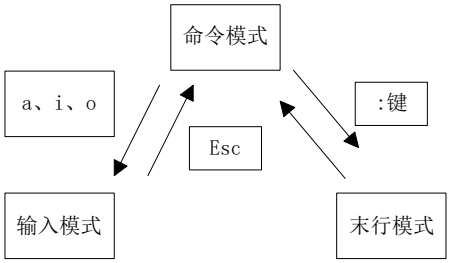


图 1 vim 关系图

3.3 vim 使用方法

1. 打开文件: **vim [参数] 文件名称**

若文件存在则打开该文件，若文件不存在则打开一个空文件。

2. 命令模式下的移动操作

字符	作用
w	移动光标到下一个单词的单词首
b	移动光标到上一个单词的单词首（begin）
e	移动到下一个单词的单词尾（end）
$	移动光标到行尾
n$	移动光标到当前光标所在行之后 n 行的行尾
0 或^	移动光标到行首
gg	光标移动到第一行
G	光标移动到最后一行
ngg 或 nG	（n 为数字）移动光标到第 n 行

3. 命令模式下的增删改查操作

vim 命令	作用
?字符串	在文本中从下至上搜索该字符串
/字符串	在文本中从上至下搜索该字符串
x	删除光标所在处的字符
nx	删除光标所在处后 n 个字符
dd	删除光标坐在的行
ndd	删除光标所在行的后 n 行
dG	删除光标所在行到文件末尾所有内容
D	删除光标虽在出到该行行尾的内容
yy	复制当前行
nyy	复制当前行以下 n 行
dd	剪切当前行
ndd	剪切当前行以下 n 行
p	将之前删除或复制过的数据粘贴在光标后面
P	将之前删除或复制过的数据粘贴到光标前面
r	取代光标所在处字符
R	从光标所在处开始替换字符，按 Esc 结束
u	取消上一步操作（删除、替换、剪切等）

4. 插入模式下的操作

字符	作用
a	在光标所在字符后插入
A	在光标所在行的行尾插入
i	在光标所在字符前插入
I	在光标所在行的行首插入
o	在光标所在行的下面插入新行
O	在光标所在行的上面插入新行

5. 末行模式下的操作

vim 命令	作用
w	保存修改
w 新文件名	另存为指定文件
q	退出
!	强制执行
zz	快捷键，保存修改并退出
:set nu	设置文本显示行号
:set nonu	设置文本取消行号
:set ic	设置不区分字母大小写
:set noic	设置区分大小写
:n	光标移动到第 n 行
:n1,n2d	删除 n1 到 n2 范围内的所有行
:s/one/two	将当前光标所在行的第一个词替换
:s/one/two/g	将当前光标所在行的所有单词替换
:%s/one/two/g	替换全文的单词
:n1,n2s/one/two/g	将 n1 到 n2 范围内的单词替换

4 第四章重点

4.1 用户和组概念

1. root 管理员

root 管理员就是存在于所有类 UNIX 系统中的超级用户，它拥有最高的系统所有权，能够管理系统的各项功能。

2. 系统用户（伪用户）

系统用户或伪用户是系统或服务正常运行所必需的用户，不能登陆系统的，也不能被删除。

3. 用户组

创建用户时将自动创建一个与其同名的基本用户组，多用户为扩展用户组。

4.2 用户和组管理相关文件

1. /etc/passwd

用户名:密码标志:UID:GID:描述性信息:主目录:默认 shell

该文件保存的是系统中所有的用户和用户的主要信息；每行代表一个用户；用“:”作为分隔符，将每行划分为 7 个字段。其中要注意的是 UID0 为超级用户 UID，1~999 为系统用户 UID，1000~以后为普通用户 UID。

2. /etc/shadow

该文件保存着用户的实际加密密码和密码有效期等参数，其权限为 000，所以保存的内容除 root 用户外其他用户是不能查看的；该文件中每行代表一个用户，用“:”作为分隔符，将每行划分为 9 个字段。

3. /etc/group

组名:组密码标志:GID:该用户组中的用户列表

该文件是记录组 ID（GID）和组名的对应文件；该文件中每一行代表一个用户组，各用户组中以“:”作为字段之间的分割，分为 4 个字段。

4. /etc/gshadow

组名:加密密码:组管理员:组附加用户列表

该文件保存组密码的文件，组管理员可以利用这个密码管理这个用户组。该文件每行代表一个组用户的密码信息，隔行信息用“:”作为分隔符分为 4 个字段。

4.3 用户和组管理相关命令

命令	语法格式	功能描述
id	id 用户名	显示用户详细信息
whoami	whoami	查看自己的用户身份
useradd	useradd [参数] 用户名	添加用户
passwd	passwd [参数] 用户名	用于修改用户密码、过期时间
usermod	usermod [参数] 用户名	修改用户信息
userdel	userdel [参数] 用户名	删除已有的用户账户
su	su [参数] 用户名	切换不同的用户身份
groupadd	groupadd [参数] 组名	添加用户组
groupmod	groupmod [参数] 组名	修改用户组的相关信息
goupdel	goupdel 组名	删除用户组
gpasswd	gpasswd [参数] 组名	设置组密码并指定组管理员
newgrp	newgrp 组名	切换用户的有效组

5 第五章重点

5.1 基本权限的概念

1. 权限身份

文件的所有者、所属组、其他人

文件的所有者一般是文件的建立者，所属组为该文件或目录归属的用户组，除所有者、所属组中用户外都是其他人。

2. 文件的权限位

使用 ls -l 命令获得如下类似的文件信息：

```
lin@lin-B150M-D3H:~/shiyan6workspace$ ls -l t29.sh
-rw-rw-r-- 1 lin lin 375  4月  24 19:41 t29.sh
```

字段 1，表示文件的权限；字段 2，表示文件的链接数；字段 3，所有者；字段 4，所属组；字段 5，表示文件的磁盘占用大小；字段 6、7，表示最近一次修改时间；字段 8，表示问及那的名称。

3. 权限位的类型和含义

符号	说明	含义
-		普通文件或无该权限
d	Document	目录文件
l	Link	链接文件
p	Pipeline	管道文件
b	Block	块设备文件
c	Character	字符设备文件
s	Socket	套接字文件
r	read	可读
w	write	可写
x	execute	可执行

4. chmod 命令更改权限

chmod [ugoa][[+-=][perms]] 文件名称

用户身份

符	全称	含义
u	user	所有者
g	group	所属组
o	other	其他人
a	all	全部

赋予方式

符	含义
+	加入权限
-	减去权限
=	设置权限

权限

符	含义
r	读
w	写
x	执行
s	特殊权限
t	特殊权限

5.2 ACL 权限的概念

1. 概念

如果对某个指定的用户进行单独的权限控制，就需要用到文件的访问控制列表（FACL），解决用户身份不足的问题。基于普通文件或目录设置 ACL 访问控制其实是针对指定的用户或用户组设置文件或目录的操作权限，更加精确的派发权限。

2. 默认 ACL 权限

如果针对某个目录设置了默认 ACL，则目录中的文件会继承其权限；如果针对某个文件设置了默认 ACL，则文件不再继承其所在目录的权限。

3. 两个常用命令

1) getfacl 命令

功能描述：用于查看文件的 ACL 权限规则。

语法格式：**getfacl 文件名称**

2) setfacl 命令

ACL 权限提供的是在所有者、所属组、其他人的读/写/执行权限之外的特殊权限控制，使用 setfacl 命令可以针对单一用户或用户组、单一文件或目录来进行读写执行权限的控制。

语法格式：**setfacl [参数] 文件名称**

参数	作用
-m	“u:用户名:权限”或“g:组名:权限”的格式进行赋予 ACL 权限
-M	从文件中读取权限
-x	删除某个指定的 ACL 权限
-b	删除全部 ACL 权限
-d	设定默认 ACL 权限，仅对目录生效
-k	删除默认 ACL 权限
-R	递归子目录设定 ACL 权限

5.3 特殊权限的概念

1. SetUID 功能

SetUID 是对二进制程序进行设置的特殊权限，能够让二进制程序的执行者临时拥有属主的权限；只有可执行的二进制程序才能设置 SetUID 权限；命令执行者要对该程序拥有执行权限；SetUID 权限只在该程序执行过程中有效，程序执行结束后自动回复原身份。

2. SetUID 权限位标志“s”

passwd 命令所有者的权限由 rwx 编程了 rws，其中 x 改编成小写“s”就意

味着该文件被赋予了 SUID 权限。如果原先权限位上没有 x 执行权限，那么被赋予 SetUID 特殊权限后将办成大写的“S”。

3. SetGID 特殊权限

当 s 标志在所属组的 x 位置时是 SetGID，简称 SGID；当 SGID 对二进制程序进行设置时，能够让执行者临时获取到文件所属组的权限；当 SGID 对目录进行设置时，则是让目录内新创建的文件自动继承该目录原有用户组的名称。

4. SBIT（Sticky Bit，粘滞位）特殊权限

粘滞位目前只对目录有效，普通用户对该目录拥有 w 和 x 全新，如果没有粘滞位，那么因为普通用户拥有 w 权限，所以可以删除此目录下的所有文件，包括其他用户建立的文件。但是被赋予了粘滞位，除了 root 可以删除所有文件，普通用户只能删除自己建立的文件，不能删除其他用户建立的文件。

5.4 权限的字符表示法与数字表示法

1. 特殊权限的数字法表示

SUID、SGID|与 SBIT 也有对应的数字法表示，分别为 4、2、1。文件权限的数字表示法是由“特殊权限+一般权限”构成的，其中 7777 为满权限，第一个数字就是代表特殊权限位。

2. 转换方法

SUID、SGID 和 SBIT 分别对应着所有者、所属组和其他人对文件或目录的 x 权限，若原本权限的 x 位没有权限，且有特殊权限，则对应使用 S 与 T，若原本权限的 x 位有权限，且有特殊权限，则使用 s 与 t。

5.5 权限管理命令与权限规划

命令	语法	功能描述
chmod	[ugoa][[+-=][perms]] 文件名称	更改文件权限
chattr	chattr [+-=][参] 文件/目录	设置文件的隐藏权限
lsattr	lsattr [参数] 文件/目录名称	用于查看文件的隐藏权限
sudo	sudo [参数] 用户名	给普通用户提供额外的权限

6 第六章重点

6.1 文件系统的概念

1. 文件系统

不同的操作系统，管理系统中文件的方式也不尽相同（给文件设定的属性和权限也不完全一样），因此，为了使硬盘有效存放当前系统中的文件数据，就需要将硬盘进行格式化，令其使用和操作系统一样（或接近）的文件系统格式，格式化过程的主要目的是向硬盘中写入文件系统。

2. 硬盘分区

Windows 或 Linux 系统，可识别的分区类型包括以下三种：主分区、扩展分区和逻辑分区。其中一块硬盘最多只能分成 4 个主分区，扩展分区只能有一个并且也算作主分区的一周，所以主分区加扩展分区最多有 4 个，但是扩展分区不能存储数据和进行格式化，必须再划分成逻辑分区才能使用。

6.2 分区设备命名规则

1. 使用“sd”代表 SCSI 或 SATA 硬盘，使用“hd”代表 IDE 硬盘。
2. 硬盘编号：系统采用 a~z 来表示 24 块不同的硬盘（默认从 a 开始）
3. 分区编号：使用“1~4”代表主分区或扩展分区，“5~59”嗲表逻辑分区
4. 命名格式：/硬件目录/硬盘类型硬盘编号分区编号，例如“/dev/sda5”

6.3 分区命令的基本功能

1. 挂载

挂载指将硬件设备的文件系统和 Linux 系统种的文件系统通过指定目录（作为挂载点）进行关联，所有的硬件设备必须挂载之后才能使用，且挂载点必须是目录。

2. 分区命令

命令	语法格式	功能描述
df	df [参数] [挂载点]	查看已挂载的文件系统信息
du	du [参数] [目录或文件名]	统计目录或文件占用磁盘大小
fdisk	fdisk -l / fdisk 设备文件名	列出系统分区/指定硬盘分区
mkfs	mkfs [参数] 分区设备文件名	对硬盘分区格式化
mke2fs	mke2fs [参数] 设备文件名	格式化分区，可指定特殊设置
mount	mount [参数] 设备名 挂载点	用于挂载 Linux 系统外的文件
unmount	unmount 设备文件名或挂载点	卸载已经挂载的硬件设备

7 第七章重点

7.1 软件安装的两种模式

1. 源码包

是使用特定的格式和语法所书写的文本代码，是人写的计算机语言指令，一般由英文单词组成。源码包的安装因为要把源代码编译为二进制语言，所以安装时间较长。

2. 二进制包

是源码包经过编译之后的包。这种包因为编译过程在发布之前已经完成，所以用户安装时速度快而且报错也大大减少。二进制包是 Linux 下的默认安装软件包，所以又是也把二进制包称作默认安装软件包。

7.2 软件安装常用命令的使用

1. RPM 包

包名-版本号-发布次数-发行商-Linux 平台-适合的硬件-包扩展名

安装命令格式：rpm -ivh 包全名

包的升级：rpm –(U/F)vh 包全名；U 为可直接安装最新版本，F 为仅当有较低版本时才能升级。

查询软件包：rpm 选项 查询对象

RPM 包的卸载：rpm -e 包名

2. yum 命令

命令	作用
yum repolist all	列出所有仓库
yum list all	列出仓库种所有软件包
yum info 软件包名称	查看软件包信息
yum install 软件包名称	安装软件包
yum reinstall 软件包名称	重新安装软件包
yum update 软件包名称	升级软件包
yum remove 软件包名称	移除软件包
yum clean all	清除所有仓库缓存
yum check-update	检查可更新的软件包
yum grouplist	查看系统中已经安装的软件包组
yum groupinstall 软件包组	安装指定的软件包组
yum groupremove 软件包组	移除指定的软件包组
yum groupinfo 软件包组	查询指定的软件包组信息

8 第八章重点

8.1 shell 编程中各种变量的概念和作用

1. 变量概述

变量时计算机系统用于保存可变值的内存单元；shell 脚本可将信息如文件名、数值等存入变量；变量定义方式：**变量名=变量值**；引用方式：**$变量名**

2. 自定义变量

用户用于存储信息的变量，变量名称可以自定义，变量值和所用可更改，由用户决定。

3. 环境变量

环境变量主要保存的时和系统环境相关的数据，变量名称不可更改，变量值可以更改，用户可以添加新的环境变量名称及变量值。

环境变量声明：export 变量名=变量值　或　export 用户自定义变量。

4. 预定义变量

预定义变量时 Bash 中已经定义好的变量，变量名称不能自定义，变量作用也是固定的，变量值可以更改。

预定义变量	作用
$?	最后一次执行的命令的返回状态
$$	当前进程的进程号（PID）
$!	后台运行的最后一个进程的进程号（PID）

5. 位置参数变量

位置参数变量主要是向脚本当中传递参数或数据的，变量名称不能自定义，变量作用时固定的，变量值可更改。

符号	作用
$n	n 为数字，$n 代表命令本身，传入命令行后第 n 个位置的参数
$*	将命令行中所有的参数堪称一个整体
$@	将命令行中的所有参数每个都区别对待
$#	代表命令行中所有参数的个数

8.2 if 流程控制语句

1. 单分支结构

语法格式 1

```
if [ 条件判断式 ];then
   命令序列
fi
```

语法格式 2

```
if [ 条件判断式 ]
   then
      命令序列
fi
```

2. 双分支结构

语法格式 1

```
if [ 条件判断式 ];then
    条件成立时，执行的命令序列
  else
    条件不成立时，执行的命令序列
fi
```

语法格式 2

```
if [ 条件判断式 ]
  then
   条件成立时，执行的命令序列
  else
    条件不成立时，执行的命令序列
fi
```

3. 多分支结构

语法格式 1

```
if [ 条件判断式 1 ];then
    条件 1 成立时，执行的命令序列
elif [ 条件判断式 2 ];then
    条件 2 成立时，执行的命令序列
else
    条件都不成立时，执行的命令序列
fi
```

语法格式 2

```
if [ 条件判断式 1 ]
  then
    条件 1 成立时，执行的命令序列
elif [ 条件判断式 2 ]
  then
    条件 2 成立时，执行的命令序列
else
  条件都不成立时，执行的命令序列
fi
```

8.3 数值判断与文件类型判断

1. 数据类型

shell 编程中，所有的变量默认都是字符串，如果不手动指定变量的类型，那么所有的数值都不能进行数值运算。

进行数值运算的三种方法

1) declare：把变量声明为指定类型（如数值型、数组型）

语法格式：**declare [+/-] [参数] 变量名**

命令参数：数组型（a）、整数型（i）、只读（r）、环境变量（x）、显示指定变量的被声明类型（-p）

2) expr：$(expr $numa + $numb)运算符号两侧必须有空格

3) let：let 变量名=$numa+$numb 与 expr 类似，但格式要求更宽松

4) “$((运算式))”或“$[运算式]”

5) 通过管道符使用 bc 命令计算任意精度的数

如：**echo 2.3*2 | bc** 或 **echo $num1+$num2 | bc**

2. 文件测试语句

文件类型判断

选项	作用
-b	判断文件是否存在，并且是否为块设备文件
-c	判断文件是否存在，并且是否为字符设备文件
-d	判断文件是否存在，并且是否为目录文件
-e	测试文件是否存在
-f	判断文件是否存在，并且是否为普通文件
-L	判断文件是否存在，并且是否为符号链接文件
-p	判断文件是否存在，并且是否为管道文件
-s	判断文件是否存在，并且是否为非空
-S	判断文件是否存在，并且是否为嵌套字文件

文件权限判断

操作符	作用
-r	判断文件是否存在，并且是否有读权限
-w	判断文件是否存在，并且是否有写权限
-x	判断文件是否存在，并且是否有执行权限
-u	判断文件是否存在，并且是否有 SUID 权限
-g	判断文件是否存在，并且是否有 SGID 权限
-h	判断文件是否存在，并且是否有 SBIT 权限

文件比较操作符

操作符	作用
F1 -nt F2	判断 F1 的修改时间是否比 F2 的新
F1 -ot F2	判断 F1 的修改时间是否比 F2 的旧
F1 -ef F2	判断 F1 和 F2 的 inode 号是否一致（两个文件是否同一）

3. 整数值比较

操作符	作用
整数 1 -eq 整数 2	是否等于，相等为真
整数 1 -ne 整数 2	是否不等于，不相等为真
整数 1 -gt 整数 2	是否大于，大于为真
整数 1 -lt 整数 2	是否小于，小于为真
整数 1 -le 整数 2	是否小于等于，小于等于为真
整数 1 -ge 整数 2	是否大于等于，大于等于为真

4. 字符串比较

操作符	作用
字符串 1 == 字符串 2	比较字符串内容是否相同
字符串 1 == 字符串 2	比较字符串内容是否不同
-z 字符串	判断字符串内容是否为空/
-n 字符串	判断字符串内容是否为非空

5. 多重条件判断

测试选项	作用
判断 1 -a 判断 2	与，1 与 2 都成立为真
判断 1 -o 判断 2	或，1 和 2 有一个成立结果为真
！判断	非，判断取反

6. read 输入命令

功能描述：接收标准输入的输入，或者其他文件描述符的输入，得到输入后将数据放入一个标准变量中。

语法格式：**read [参数] [变量名]**

参数	含义
-p “提示信息”	在等待输入时，输出提示信息
-t 秒数	指定等待输入的时间
-n 字符数	设置仅接收的字符数
-s	隐藏输入的数据