Linux 系统：由 Linux 内核+GUN 工具+其他自由软件构成的系统。Linux 的主体部分只是一个内核，采用单内核结构模式，支持动态加载内核模块（驱动）

单内核模式：整个内核作为一个大的 Project 来实现，OS 的主要模块:进程管理、内存管理、设备管理，都是其中的一个个模块函数，可以直接相互调用。---高效、实用。

微内核模式：内核层微小化，仅包括贴近硬件 I/O 操作、中断管理、同步原语、进程间通讯控制，以提高可扩展性和可移植性。进程管理、内存管理、设备驱动、文件管理等作为服务层软件，运行在用户态。但内核层与服务层之间通讯开销大。

混合内核—微内核把部分核心服务移到内核态运行

命令解释器：shell：

Bourne Shell 简称 sh。是 UNIX 默认的 shell;脚本编程功能强,但交互性差。Bourne Again Shell，即 bash, GNU linux 默认 Shell，是前者的扩展。

bash 功能特点：bash 占用资源较少，且来自 GNU，与 Linux kernel 相容性高；Bash scripts 能直接设计 program，便于高效管理 Linux 主机；提供了：命令行编辑功能，及命令补全功能键 “Tab”；命令历史记录 键入历史命令部分字符串+ESC 键+ TAB 键 ‘允许应用别名代替命令关键字：Alias <alinam>= ‘命令’

bash 主要功能：执行命令；I/O 重定向、管道；环境控制，允许用户定制环境以满足自己需要环境文件.bash_profile .bashrc .bash_logout 通过配置合适的环境变量，可以改变主目录、命令提示符、命令搜索路径等用户工作环境。；台处理 &；支持编写复杂功能的 Shell 脚本程序

root 的重要性：系统管理、安全管理、套件管理

linux 技术特点：符合 POSIX 标准，具有良好的开放性、可移植性和跨平台性。 真正的多用户、多任务 OS。不仅继承了 UNIX 特征，而且在许多方面超过了 UNIX。运行高效、稳定，可靠性好、安全性高。

Linux 系统的引导分区：/boot 该目录放置系统内核及驱动模块引导程序，需要独立分区。

Linux 系统交换分区—/swap Swap 空间的作用:作为虚拟内存使用的一段磁盘空间(设为物理磁盘的 2-3 倍)。用于暂存物理内存中因暂时不用而被交换出的程序或数据；

Linux 系统根分区：

/root（超级用户主目录）：/bin(基本命令)、/etc(各种配置文件)、/home(各个用户的主目录)、/tmp(临时文件)、/var(可变信息区：file spool,logs,requests,mail,etc)、/proc(进程信息)、/dev(设备)、/boot(kernel 和 boot 配置文件)、/usr(用户程序)、/sbin(程序管理员执行程序)、/mnt(设备/文件系统挂载点)、/lib(用于存放系统共享库文件和核心模块)、/opt(附加的应用软件包)

init 内核启动时会有一个 init_task 进程，它是系统的 0 号进程/idle 进程。后续所有的进程都是由它 clone 出来的。

管理 init：用$runlevel 命令观察运行级别；

用 init 或 telinit 改变运行级别:$init 3

进入与退出系统：虚拟控制台： ALT+1...4 **关机：** shutdown –h now——立刻关机；Shutdown –h +1---1 分钟以后关机；init 0；halt——立即停止系统，需要人工关闭电源 Poweroff--立即停止系统，并且关闭电源。**重新启动：** reboot；init 6 **注销** Logout/exit。

grub:是一个多重操作系统启动管理器。

用来引导不同系统,在 X86 架构机器中, GRUB、LILO 最为常用。它的第 1 阶段存放在 MBR 扇区(0 柱面 0 磁头 01 扇区); 第 2 阶段存放在 boot 目录下，即/boot/grub 文件

多用户环境:每个账号均有自己的 home 主目录、mail box，以及其它与账号相关的环境设定。

Linux 内部使用的账号相关 ID 标识：UID, GID 两种 UID：0(root.ID)，1~65534(一般用户 ID)；三种身份：owner, group, others 与文档权限密切相关；

账户基本配置文件:

/etc/passwd:user 基本配置文件，每行存放一用户信息，：用户名、口令、UID、GID、home 目录等域。

/etc/shadow: user 影子口令文件, 每行存一用户密码相关信息：：用户名、MD5 口令，以及各类时间等域。

/etc/group: group 配置文件,每行存放一用户组信息，

：组名、组口令、GID、组成员列表等域.

/etc/gshadow: group 影子文件,每行存放一用户组口令等信息，：组名、组口令、管理员账号、组成员列表等域。

用户管理基本命令：useradd(添加用户)、userdel(删除用户)、passwd (为当前用户设置密码)、usermod (修改用户命令，可用来修改登录名、用户 home 目录等)、finger (查看用户信息)、id (查看用户的 UID、GID 及所归属的用户组)、chfn (更改用户信息工具)、su (用户切换工具)、sudo (用另一个用户身份来执行命令)、visudo(编辑 /etc/sudoers 的命令；也可直接用 vi 来编辑)

useradd [参数] 用户名

Useradd 命令参数：d home_dir：设置用户登录的默认工作目录， 缺省为/home/<username>-e expired-date ：设置账户的过期日期。-g initial gname：设置新用户初始组 (主组)，默认为与用户同名的私有组。-G groupNames：设置用户的二级组，多个用’，’分隔；-s shell_name ：默认为/bin/bash-u uid: 指定用户的 uid，必须大于 500；

修改用户账户命令，usermod [参数] 用户名

除了 useradd 参数外，还增加以下参数选项：-l <new_login_name>：修改用户登录名 (注，不允许更改在线用户账号名称；改变 uid，不会自动更改 crontab,at 等工作档中的 uid)

设置账号口令，passwd [参数] 用户名 只有 root 才有权限使用，常用参数选项：-S: 查询指定用户口令。-l: 锁定用户口令。-u： 解锁用户口令。-d： 删除指定用户口令。

用户(组 group)管理基本命令 groupadd [-r] <组名> # 添加用户组,-r 指示建立系统组；**groupmod [-r] <组名>** #修改用户组信息；**groupdel <组名>** #删除用户组；**groups** #显示用户所属的用户组；

用户管理相关文件：/etc/default/useradd 文件

查看用户信息$ who [选项]

-H 附加显示列标题-m 只显示运行本命令的用户

-q 只显示已登录用户账户名及用户总数量

$ whoami(显示当前用户名)

$w 命令查询当前登录系统用户行为;能提供比 who 更详细的信息，包括已登录时间、占用 CPU 时间等统计信息。

文件的概念：是一个抽象的概念。是由创建者定义的、有确定名称的：字节序列

文件分类：0(未知)、1(普通文件[-])、2(目录文件[d])、3(字符设备[c])、4(块设备[b])、5(命名管道文件[p]、6(网络文件[n])、7(符号链接文件[l])。

引入特殊文件“目录文件”具有普通文件的三要素：目录项、inode 和“数据体”；目录文件“数据体”存储其下级各类文件的文件目录项记录集

文件分类

一、普通文件

文本文件: ASCII 码形式存储 “–” 开头，如：-rw-r--r-- 1 root root 39599 Mar 8 12:15 x

二进制文件: 以二进制形式存储在计算机中，不可直接读，要通过相应的软件读取 “–”开头，如：-rwxrwxrwx 1 root root 46888960 Dec 9 2005 x.sh

二、目录文件：

d 字母开头 如：drwxr-xr-x 2 root root 4096 Aug 2 2006 bin

三、设备文件

块设备文件: b 字母开头 如：brw-rw---- 1 root disk 3, 1 Jan 30 2003 hda1 *字符设备文件*: c 字母开头 如：crw------- 1 root root 4, 1 Jul 31 13:49 tty1

特殊目录：根目录 / 当前目录 .

父目录 .. Home 目录 ~

路径名：绝对路径名：从根目录开始的、以/分隔的各级目录名；文件说明：路径名+文件名 例如，/usr/bin/cp

相对路径名：从当前目录开始的路径， 例如，当前路径为/usr/src,则前例文件说明为 ../bin/cp

Linux 文件系统概念

磁盘分区卷布局格式，简称文件系统·卷格式。

常见的文件系统卷格式类型（file-system-type）

文件系统(卷）的创建

用 fdisk 或其他工具定义/创建磁盘分区；在指定的分区上创建文件系统（卷）也称为文件系统格式化，通过一些格式化工具来进行；每类操作系统都有这方面的工具；

在 Linux 中有 mkfs 系列工具。创建方法，以系统第二块硬盘为例：/dev/sdb 分区：$>fdisk /dev/sdb 分区后，获得若干分区，其主分区为/dev/sdb1；创建：$>mkfs –t ext2 /dev/sdb1 $>mke2fs.ext2 /dev/sdb1 查看结果：$>dumpe2fs [选项] /dev/sdb1

文件系统挂载（mount）不像 windows 系统下有 C:、D:、…等不同逻辑盘符，Linux VFS 只有一棵目录树，它通过 mount 把所有分区绑定/组织到这棵目录树上。指定的挂载目录，称为挂载点；指定文件系统类型。umount 命令将未被使用的文件系统从目录树上卸载掉。

每个 partition 就是一个完整的 filesystem；使用 mount 来挂载每一个 partition，还需指定 mount point (directory)、 filesystemtype；命令使用举例，

$> mount -t vfat /dev/hda5 /mnt/windows

$> mount -t ntfs /dev/hda6 /mnt/winnt

$> mount -t ext3 /dev/hdb1 /mnt/linux2

挂载/dev/cdrom 到/mnt：**mount /dev/cdrom /mnt**

虚拟机安装的 Linux 共享 windows 目录

在 windows 环境下，将要共享的目录，比如 linux-share-dir 设置为共享目录；

创建挂载点目录，例如,windir,执行 mount 命令，$>mount –t cifs –o username=<uname>, password=xx,rw //<ip>/linux-share-dir /mnt/windir

文档权限管理:(所有者、所属者、其他人)

特殊执行权限位 setuid、setgid 和 sticky

setuid/setgid 位，允许普通用户以 root 角色运行只有 root 才能运行的程序或命令。 sticky 位则指示程序运行结束后，在内存中驻留一份副本，再运行启动更快，但会占有内存。setuid、setgid 和 sticky 八进制位值分别为：4000,2000,1000 九字母法叠加在相应的 x 位上

用户[rwx] + suid = [rws] ；[rw-] + suid = [rwS] ；

组 [rwx] + sgid = [rws] ；[rw-] + sgid = [rwS]

其它[r-x] + sticky= [r-s] ；[r--] + sticky = [r-S]

应用举例，$>chmod 4755 /bin/rm #设置 rm 的权限为 4755 -rwsr-xr-x 1 root root 26556 Feb 19 2003 /bin/rm $>chmod ug + rws, o -wx fil1 #给拥有者用户组加上 rws 权，其他用户删除写/执行权

目录及文件安全命令 chmod 命令（改变文件或目录的权限）$>chmod 4755 /bin/rm #设置 rm 的权限为 4755 -rwsr-xr-x 1 root root 26556 Feb 19 2003 /bin/rm #给拥有者用户组加上 rws 权，其他用户删除写/执行权 $>chmod ug + rws, o -wx fil1

chown 命令（改变文件或目录的所有者）$> chown [选项] 用户[:用户组] 文件名 只有 root 才能改变所有者所有者可以更改用户组为包含自己的用户组 用户可用名字或 uid，用户组可用名字或 gid d$> chgrp [选项] 用户组 文件名

文件的默认权限与 umask 掩码

系统默认 umask=0022，可通过 umask 命令显示或修改：$>umask #不带参数时为显示 0022

$> umask 0002 #将 umask 修改为 0002

新建文件的默认权限（所有者无执行权）

0666 – umask = 0666 – 0022 = 0644；

新建目录的默认权限为：0777 – umask = 0744

suid 程序的系统安全隐患

设/bin/bash 的 suid 位，会给系统安全带来巨大隐患。普通用户利用这个空子，通过如下两个命令操作，可变把自己权限提升为超级用户权限：

$> cp /bin/bash /home/a/.rootshell

$> chmod 4777 /home/a/.rootshell

系统管理员可用下面的组合命令定期检查被设置 suid/guid 位的隐患程序：

$> find / -type f \(-perm -04000 –o –perm 02000 \) –exec ls –la {} \;

小结

用 fdisk 划磁盘分区;用 dd 命令创建虚拟磁盘分区；

$>fdisk –l #仅显示分区信息 $>fdisk [磁盘名]

用 mkfs 命令格式化建立文件系统；

用 dumpe2fs 命令查看 ext2/3 的文件系统信息；

用 mount/umount 命令挂载/卸载文件系统；

通过 fdsik –l 查看磁盘分区；硬盘容量计算公式= 磁面数 x 扇区数 x 每扇区大小 512 x 柱面个数

fdisk 分区管理

[root@node1 ~]# fdisk /dev/sdb

Command (m for help): m

Command action a toggle a read only flag b edit bsd disklabel c toggle the mountable flag d delete a partition l list known partition types m print this menu n add a new partition o create a new empty DOS partition table p print the partition table q quit without saving changes s create a new empty Sun disklabel t change a partition's system id u change display/entry units v verify the partition table w write table to disk and exit x extra functionality (experts only)

Command (m for help):

用 df 统计整个分区文件系统情况

因只需要查看每个块组的块位图，执行非常快。$> df [-h | -k | -i] [文件系统] 未指定文件系统，则显示所有文件系统；选项：-h 自动采用合适的大小单位，-k 总以 k 为单位 -i 显示文件系统 inode 的使用情况 用 du 命令查看一个目录的已用空间执行不快，因为要搜遍整个目录的所有文件。$> du [-h | -k] [目录或文件名]

常用核心命令 pwd（显示当前目录命令）cal [-y|-3] [月份] [年份] （显示日历）默认月份、年份是当月当年选项-y 显示全年 12 个月，-3 显示本月及前后月

date （显示日期） man （联机帮助命令）

ls （列目录清单命令）$> ls [选项] [路径名或文件名]

常用选项：-a 列出所有类型文件，包括隐藏的；-l 以长格式显示详细信息；-h 以合适的单位显示文件大小；-R 递归列出所有下级子目录中文件；-F 在列出的文件名后，附加一个特殊类型符号 如 “/” 表目录、“*” 表可执行 “@” 表链接、l” 表管道. (普通文件无)

cat （显示查看文本文件内容命令）

$> cat [选项] [文件名 1[文件名 2 ..]]常用选项：

-n 对所有行编号；-b 只对非空行编号；

高级应用：与管道、I/O 重定向等结合应用；

$> cat test* > alltest.txt $> cat test* | more

wc （统计文件的字数、行数、单词数）

$> wc [-c | -l | -w] 文件名

more （分页显示查看文本文件内容命令）

每次显示一屏幕内容暂停，并显示--more—

空格键显示下一页，回车显示下一行。

$> more [选项] [文件名]

less （类似 more，但功能似乎更强些）

可用方向键前后翻译不同的屏幕页；按 Q 退出

/<字词>，可在文中快速搜索; /继续找下一个

!<其它简单命令>，例如，!wc [file]

find （查找文件或目录命令）

$> find 路径名 查找选项 [操作选项]

路径名是搜索的顶层目录，会自动递归搜索子目录

常用查找选项 -name 文件名 -type 文件类型 [f|d|b|c|l] -user 用户名 -atime/mtime/ctime n 指定天数以前被访问/修改/改状态，+是大于，-是小于 -size n 指定文件大小；-perm <八进制权限值>可对查到的文件，进行附加操作，

-exec 命令 {}\; （-ok 会对每个查到文件提问是否执行） -print 显示查找匹配到的每个文件路径名；查找条件可用 and(-a),or(-o),not(!)组合

locate （查询文件名中包含指定字符串文件）

使用系统预先建立的文件名数据库，且仅查文件名，故执行速度很快。(/var/lib/mlocate/mlocate.db)root 可用 update 命令更新数据库。文件名可用通配符；选项-i 忽略大小写。$> locate [选项] [文件名]

whereis （对指定命令，查其对应的二进制文件）

例如，$>whereis find 结果 find: /usr/bin/find /usr/man/man1/find.1.gz

grep （在一个或系列文件内容中，搜索匹配字符串）

$> grep [选项] [正则模式] [文件名]可用正则表达式(模板)描述需要查找的字符串；常用查找选项

-c 只显示匹配行的数目；-i 比较时忽略大小写

-l 只显示匹配行所属文件名，文件中多次匹配只显示 1 次 -n 显示匹配行在文件中的行号 应用举例，$> grep -l -i linux /etc/*输出结果： file1: LINUX file1: linux system file2: linux

正则表达式语法 \:解除特殊字符的特殊含义。

例如，在待搜索字符串中含有？则用\?解除？的特殊性

^:匹配一行的开始位置，例如，“^b”，匹配以 b 开头的行**$:匹配一行的结尾位置**，例如，“b $”，匹配以 b 结尾的行 ***:匹配前面字符的零次或多次**，例如，” zo*” ，能匹配” z” ,” zoo” **+:匹配前面字符的 1 次或多次**，例如，” zo+” ，能匹配” zo” ,” zoo” ，不能匹配” z” **? 匹配前面字符的 0 次或 1 次**，例如，” zo? ” ，只能匹配” zo” ，” z” **.: 匹配任意单个字符 x|y :匹配字符 x 或 y**，例如，” z|food” 能匹配” z” 或” food” **[xyz]: 匹配括号中的某一个字符**

[^xyz]:不匹配括号中的某一个字符 [a~z] 匹配指定范围中的任一个字符

touch 命令功能：生成一个空文件或修改文件的存取/修改的时间记录值。 格式：touch [参数] <文件名>

[test@linux test]$ touch * 将当前目录下的文件时间修改为系统的当前时间 [test@linux test]$ touch –d 20070806 test 将 test 文件的日期改为 20070806

[test@linux test]$ touch abc 若 abc 文件存在，则修改为系统的当前时间；若不存在，则生成一个为当前时间的空文件

ln 命令功能：在文件和目录之间建立链接

格式：ln [参数] <源文件或目录> <目标文件或目录>

链接分 “软链接” 和 “硬链接” 1.软链接：[root@linux test]# ln –s /home/xie2/t1 xdoc 创建一个链接文件 xdoc,并指向目录/home/xie2/t1 2.硬链接：[root@linux test]# ln /home/xie2/t1 hdoc 创建对应 t1 的一个硬链接文件 hdoc。

文件拷贝/移动命令 cp/mv

$>cp [-f|-i|-r] 源文件或目录名 目标文件或目录名

文件名，可以使用” ?” 、” *” 等通配符；

选项：-f 强制复制；-i 交互式提问复制；-r 递归目录文件应用举例： $> cp a.txt d1 //文件 a.txt 复制到 d1 目录下 $> cp a.txt d1/b.txt //复制到 d1 目录下并更名 b.txt $> cp -r d1 bkup $> cp –r /etc/*.* 目标文件夹允许将数个不同文件，复制到同一目录，此时格式： $>cp 文件 1 文件 2 … 目标目录名 mv(完成后会自动删除源文件或目录)$>mv [-f|-i|-r] 源文件或目录名 目标文件或目录名

删除文件与目录命令

rm 命令$> rm [-i | -f | -r] 文件或目录名 联合选项-fr，既高效也危险。rmdir 目录名 （只能删除单级空目录）mkdir 创建目录命令$> mkdir [-p] 目录名

加选项-p 可自动创建不存在的父目录

举例：$>mkdir –p /home/xie/par/ch1

其他杂项操作$>uname [-a|-s|-r|-m] 显示系统有关信息选项：-a 所有信息； -s OS 名称；-m 机器硬件 -r OS 发行版本 $>hostname 查看主机名

$>history |less 查看系统的历史操作记录

dmesg 命令显示系统诊断信息、操作系统版本号、物理内存的大小以及其它信息

free 命令用于查看系统内存，虚拟内存（交换空间）的大小占用情况

Linux 系统的基本网络配置

Linux 中，TCP/IP 网络是通过若干文本配置文件进行配置的，这些文件在/etc 目录或它的子目录下。

启动时，内核可检测出所有的网络接口卡(NIC)

可用 lspci 命令可列举内核已检测到的 PCI 设备

NIC 不作为文件处理（在/dev 下无 NIC 对应设备节点）

系统启动时，将执行 shell 脚本程序 /etc/rc.d/init.d/network 来完成网络的配置。

脚本程序调用了网路配置命令/sbin/ifconfig，由它读取各网卡的配置文件来配置接口。

网络基本配置

$>cd /etc/sysconfig/network-scripts

$>vi ifcfg-eth0 #写入网络 IP 等

分配 IP 地址： IPADDR=<IP>

子网掩码： NETMASK=255.255.255.0

配置默认网关: GATEWAY=<网关 IP>配置一个或以上的 DNS 在/etc/resolv.conf 中添加；设置完 IP 后，重启网络系统 $> /etc/rc.d/init.d/network restart

用 ifconfig 命令配置网络接口

ifconfig ethx [选项] [地址]用 ifconfig –a 命令显示所有已识别的 NIC 配置后用 up 直接激活 $> Ifconfig eth0 up(激活网卡)$> ifconfig eth0 192.160.1.1 netmask 255.255.255.0 up 禁用网卡 down

$> ifconfig eth1 down

配置 DNS DNS 客户端配置文件：/etc/resolv.conf 样例文件：search whucs.com nameserver 192.168.0.1 nameserver 192.168.0.2

主机未联网域名服务器不能用时，使用/etc/hosts 文件，样例如下：127.0.0.1 Localhost server.whucs.com 192.168.0.3 station1.whucs.com

用 netstat 查看整个网络状况格式： netstat [-acCeFghilMnNoprstuvVwx][-A][--ip]

主要参数 -a--all：显示所有连线中的 Socket。

-c--continuous：持续列出网络状态。 -e--extend：显示网络其它相关信息。 -g--groups：显示多重广播功能群组组员名单。 -i--interfaces：显示网络界面信息表单。 -l--listening：显示监控中的服务器的 Socket。

-n--numeric：直接使用 IP 地址，而不通过域名服务器。

-r--route：显示 Routing Table。 -s--statistice：显示网络工作信息统计表。 -t--tcp：显示 TCP 传输协议的连线状况。 -u--udp：显示 UDP 传输协议的连线状况。

-v--verbose：显示指令执行过程。 -w--raw：显示 RAW 传输协议的连线状况。

用 ping 查询机器联网是否正常 ping [选项] 主机名|IP 地址|子网地址（广播）命令中各选项的含义如下：

-c 数目 在发送指定数目的包后停止.-f 大量且快速地送网络封包给一台机器，看它的回应.-l 次数 在指定次数内，以最快的方式送封包数据到指定机器 (只有超级用户可以使用此选项) -q 不显示任何传送封包的信息，只显示最后的结果 -r 不经由网关而直接送封包到一台机器，看本机的 NIC 是否有问题

用 traceroute 检测目标主机是否有路可达

traceroute [-nwig] IP

选项参数-n ：可不必进行主机名解析，单纯用 IP，速度较快!-w ：若对方主机在几秒钟内没回声就宣告不可达...预设是 5 秒-i ：用在比较复杂的环境,比如有两条 ADSL--有两个 ppp，可以用-i 来选择是 ppp0 还是 ppp1

-g ：与-i 的参数相仿，用来选择不同 gateway 的 IP

显示和修改本机路由（route 命令）route [-f] [-p] [Command] [Destination] [mask Netmask] [metric Metric] [Gateway] dev [Interface]-f 清除所有网关入口的路由表；-p 与 add 命令一起使用时使路由具有永久性；Command 包括：Add/Change/Delete/Print；mask Netmask 指定子网掩码；metric Metric 为路由指定一个整数成本值标（1-9999），在路由表(与转发数据包目标地址最匹配)的多个路由中进行选择时使用；Gateway 指定网络目标地址集和子网掩码可达的前进或下一跃点 IP 地址

显示和修改本机路由（route 命令）使用 route 编辑路由表 $> route –n $> route del default // 删除默认网关 $> route –n 指定默认的新网关

$> route add default gw 192.168.0.254

用 arp -a 命令查本机 arp 缓存信息
典型 arp 缓存信息如下：
Interface: 192.168.11.3 --- 0x2
Internet Address Physical Address Type
192.168.11.1 00-0d-0b-43-a0-2e dynamic
192.168.11.2 00-01-4a-03-5b-ed dynamic
其他网络命令$> hostname 显示和设置系统的主机名
$> hostname station.exam.com //动态分配
$> cat /etc/sysconfig/network 输出样式如下，
NETWORKING=yes HOSTNAME=station.exam.com
NISDOMAIN=RHA131 $> host server1.exam.com
server1.exam.com has address 192.168.0.254
$> host 192.168.0.254 192.168.0.254 in_addr.arpa domain name pointer ...server1.exam.com
FTP 服务管理（1）
vsftpd 是一个典型的 FTP 服务器程序。检测与安装 vsftpd$> rpm –qa | grep ftp 如检测不到，则用如下命令安装，$> rpm –ivh vsftpd-version.rpm
也可以用更简单的 yum 命令安装$> yum install vsftpd vsftpd 服务的启动、停止及状态检测
vsftpd 服务的启动、停止及状态检测$> service vsftpd status $> service vsftpd start $> service vsftp restart 要在引导时自动启动 vsftp 服务器，可用如下命令：$>chkconfig --level 5 vsftpd on $>chkconfig --list | grep vsftpd vsftpd 的配置文件 /etc/vsftpd/conf/vsftpd.conf 主配置文件
/etc/vsftpd/conf/ftpusers 指定那些本地用户不能访问 ftp 服务 /etc/vsftpd/conf/user_list 指定那些允许访问 ftp 服务的本地用户 （要求 vsftpd.conf 中选项 userlist_deny=NO）
FTP 登录匿名登录
$> ftp 127.0.0.1 或从浏览器以 ftp://127.0.0.1 登录 登录后的默认目录为：/var/ftp/pub 匿名用户 ftp 不仅不能离开默认目录，也不能上传文件。
本地权限用户登录 $> ftp <username>@<ip>
若通过验证就可登录访问该用户的/home/<username>
创建用户或修改用户，可以改变用户 home 目录。此外，还应该给用户 home 设置合宜的读写权限。
Apache WEB 服务器管理
检测与安装 $> rpm –qa | grep httpd 如检测不到，则用如下命令安装，$> rpm –ivh httpd-version.rpm 更简单的安装方法：$> yum install httpd 启动、停止及状态检测$> service httpd status $> service httpd start $> service httpd restart Apache 服务器的配置默认配置文件：/etc/httpd/conf/httpd.conf
配置文件中，一些重要的默认配置项

服务器的根目录 ServerRoot /etc/httpd
管理员 Email 地址 ServerAdmin. root@localhost
根文档目录 DocumentRoot /var/www/html
站点主页文件 DirectoryIndex index.html indexhtm.
HTTP 端口号 Listen 80
个人 Web 站点 /home/*/public html
配置默认的 WEB 站点默认 Web 站点主目录 /var/www/html 将一个已建好的网页文件，比如 default.htm 复制到该目录下；
将 default.htm 添加到配置的站点主文件行。
配置个人（系统用户比如 user1）Web 站点修改配置文件，按说明激活其中一些被注释行$> su – user1 // 转到 user1 的 home 目录下$> mkdir public_html $> cd //退到父目录$> chmod 711 user1 //设置目录权限 复制或编辑一个站点主页文件 index.html 到 /home/user1/public_html/下;
Linux 中查看网络接口
用 ifconfig -a 命令，可查看系统检测到的网络接口。eth0、eth1、..是以太网接口 lo 是 loopback 接口
每一种数据链路层协议，都有一个 MTU 定义。可用 netstat –in 命令查看 MTU 信息
网络层负责把来自传输层的数据进行 MTU 分割，将每个 MTU 单元封装成 IP 数据报。对 IP 数据报进行无连接传送、寻径以及差错处理。本层主要协议 IP 协议、ARP/RARP 协议和 ICMP 协议 IP 协议是点对点的不可靠协议，不保证 IP 数据报一定送达目的地。
ICMP(Internet Control Message Protocol)
进程概念 将能完整记录“运行中任务”状态信息的数据结构，称为任务描述符（Task Descriptor）。当把“运行中任务”用一个更简洁的名词“进程（Process）”来指代时，任务描述符也常被称为进程控制块(Process Control Block, PCB)
系统虚拟存储器指（主存 + (被映射到虚空间的)部分辅存）构成的比实际主存更大的存储空间。
进程虚拟存储器
在 32 位系统中，系统为每个进程实现了 4G 空间的虚拟存储器，其中，仅有部分空间被映射到物理主存。
Linux 进程分类 系统进程 OS 启动时自动加载，并在启动后一直在环境中运行的系统级进程。这类进程通常不与终端关联。**用户进程**通常是与终端关联的交互进程，它们由一个用户登录的 shell 启动运行，绑定用户 ID。交互进程既可以在前台运行，也可以在后台运行。
守护进程一类在系统开机时通过脚本或开机后 root 手动启动的进程，在后台运行且总是活跃的。
例如，$> chkconfig --level 35 httpd on
进程属性进程标识（PID）、父进程标识（PPID）、启动进程的用户标识（UID）和所归属组（GID）
进程状态：状态分为运行 R、休眠 S、僵尸 Z。
进程执行的调度优先级；rt_priority 实时优先级
nice 静态优先级 进程所连接的终端名； 进程资源占用比如占用资源大小（内存、CPU 占用量）等
ps 命令（进程查看）常用选项：a(显示所有用户进程)、x(显示没有终端控制的进程)、l(显示进程的详细列表)、u(显示进程的用户名和启动时间等信息)、e(显示包括系统进程的所有进程)、f(显示进程的详细信息)
例子：**$>ps -au**

ps -aux 显示的进程状态
STAT 进程状态 名目含义 D Uninterruptible sleep (因 IO 等原因) R 正在运行可中在队列中可过行的；S 处于休眠状态；T 停止或被追踪； W 进入内存交换（从内核 2.6 开始无效）； X 死掉的进程 (从来没见过)； Z 僵尸进程；< 优先级高的进程 N 优先级较低的进程 s 进程的领导者（在它之下有子进程）；l 是多线程进程 + 位于后台的进程组；
top 命令动态显示系统当前的进程和系统处理器状况。通过命令 top –d <秒数>指定刷新时间间隔。如果在前台执行该命令，它将独占前台，直到用户按 q 键结束退出。
对各单进程信息行，可通过 p(CPU 使用时间)/m(内存使用量)/t(执行时间)键，选择不同行排序方式。
top 命令用法及参数 b 以批量模式运行，但不能接受命令行输入；-c 显示命令行，而不仅仅是命令名；-i 禁止显示空闲进程或僵尸进程；-p PID 仅监视指定进程的 ID；PID 是一个数值；-q 不经任何延时就刷新；-d N 显示两次刷新时间的间隔，比如 -d 5，表示两次刷新间隔为 5 秒；-n NUM 显示更新次数，然后退出。 比如 -n 5，表示 top 更新 5 次数据就退出；-S 累积模式，输出每个进程的总的 CPU 时间，包括已死的子进程；
用程序名查询进程的 grep 命令
常用来判断某个程序或服务是否正在运行。命令格式：$>pgrep [参数选项] 程序名 常用参数 -l 列出程序名和进程 ID； -o 进程最小的 ID； -n 进程最大的 ID； （不能与 o 联用）举例：$>pgrep -l httpd
4557 httpd $>pgrep –l –n $>pgrep –l -o
Kill 命令杀死进程：向指定进程发送终止信号。
命令格式: kill -signal PID 可用选项
-<signal> 指定发送信号类型，默认为 SIGTERM
-l 显示信号名称列表 应用举例$> kill -9 <PID> # 无条件强制终止 $> kill -15 <PID> # 正常方式关闭进程 $> kill -1 <PID> #重新读取可执行程序文件
一般是先用 ps 或 pgrep 查 pid 或程序名,再用 kill $>killall 正在运行的程序名
进程的前、后台控制前台运行进程：可共享键盘、鼠标和显示器等，执行交互终端运行模式；通常优先级高；执行的时间片多但较短；后台运行进程不能执行终端交互；优先级较低；获得执行时间片较长
在 shell 命令行下，用命令名后附加 “&” 启动的进程，按后台方式运行。
可用 jobs 命令列出查看后台方式运行的进程
可用 fg <pid> 命令，把指定进程调到前台运行。
可用 bg <pid> 命令，把指定进程放到后台运行。
按 Ctrl_Z 键，把当前获得焦点的进程放到后台运行
周期性任务管理工具 AT 和 CRON Linux 通过 at 与 cron 两个命令来实现自动化执行作业。at 命令指定作业只能在指定时间执行一次，而 cron 命令则可根据用户指定的计划表进行周期性重复执行。
at 命令语法格式：at [-f script] [-m -l -r] [time] [date]-f: script 是所要提交的脚本或命令。
-l: 列出当前所有等待运行的作业，含作业标识号 jid。
-r: 清除某个已提交的 at 作业，要提供相应的 jid
-m: 作业执行完成后给用户发邮件。 time：at 命令的时间格式非常灵活；可以是 H、H H、 HHMM、HH：MM or H:M，其中 H 和 M 分别是小时和分钟。还可以使用 am 或 pm。date：日期格式可以是月份数或日期数，且可用诸如 today、tomorrow 类的词。
cron 命令每个用户都可用一个 crontab 执行计划表文件来保存作业调度信息。允许指定任意一个 shell 脚本或程序命令，并灵活地指定周期性的执行时间。
创建：$>vi xcronjob_file 启动：$> crontab xcron_file 成功后自动生成：/var/spool/cron/username cron 提供了针对已启动作业的编辑(edit)、列出(list)查看和删除(remove)命令。 $> crontab [-e | -l | -r]
以下是 **crontab 文件的格式**：
{minute} {hour} {day-of-month} {month} {day-of-week} {命令或脚本} minute: 区间为 0 – 59 hour: 区间为 0 – 23 day-of-month: 区间为 0 – 31 month: 区间为 1 – 12. 1 是 1 月. 12 是 12 月. Day-of-week: 区间为 0 – 7. 周日可以是 0 或 7.
用*缺省某区间，表示每区间单位都执行示例
每天 02:00 执行任务： 0 2 * * * /bin/sh backup.sh# 每天 5:00 和 17:00 执行任务：0 5,17 * * * /scripts/script.sh 每周日 17:00 执行任务：0 17 * * sun /scripts/script.sh 每 10min 执行一次任务： */10 * * * * /scripts/monitor.sh
备份与压缩命令 tar 命令
主选项-c 创建(create)-x 释放 辅助选项-f 后面跟归档文件名字 -z 使用 gzip 应用示例$> tar -cvf a.tar <目标目录或文件> //非压缩打包 $> tar -zcvf a.tar <目标目录或文件> //带压缩打包 $>tar-xcvfa.tar //释放压缩包
gzip 和 gunzip 除了.zip 文件的压缩格式外，在 Linux 系统下更常见的是.gz 文件的压缩格式，这种文件一般是由 gzip 命令所产生。zip 命令具有将许多文件压缩成一个文件的功能；gzip 一般与 tar 结合一起使用：用 tar 将所有文件打包成一个文件；再用 gzip 进行压缩，形成扩展名为.tar.gz 或.tgz 的文件。应用示例
$>gzip test.txt #压缩文件时，不需要任何参数
$>gzip –l test.txt.gz #压缩文件时显示压缩率$>gunzip test.txt.gz #解压缩
RPM 软件包管理
Rpm -qa 查询系统中已安装的所有 RPM 包
rpm -q<包名> 查询是否已安装指定的 RPM 包
rpm-qp<包名> 用于安装前，了解包的基本信息
rpm-qi<包名> 查询系统中已安装 RPM 包的描述信息 rpm-ql<包名> 列出系统中已安装 RPM 包里所含的文件 rpm-qf <文件名>查询系统中指定文件所属的软件包

RPM 软件包管理 RPM 全称是 Red hat Package Manager，它是针对红帽公司软件包管理标准的实现。
安装与删除 RPM 包 安装：$> rpm –ivh <RPM 包名>
可选参数：i(install)安装，v(verify)安装中显示详细信息，h(horizontal)显示安装进度条。删除：$> rpm –e <RPM 包名> 升级与验证 升级：$> rpm –Uvh <包名> $>rpm –Uvh vsftpd-2.6.2-x86-64.rpm
验证：$> rpm –V <包名>
yum 命令 YUM 是“Yellow do Updater, Modified”的缩写，可执行程序名为 yum。
列出软件包 $> yum list [all | expr1] [expr2] [...] 列出仓库中可用的包 yum list available [all | expr1] [expr2] [...] 按同类分组形式软件仓库中的软件包
yum grouplist 显示安装包信息 yum info package1 安装软件包 $>yum install [package1] [package2] [...]安装示例：$> yum install gcc
删除软件包 $>yum remove [package1] [package2] [...] 或 $> yum erase [package1] [package2] [...] $> yum groupremove group1 //删除程序组 roup1 查看包依赖情况 $> yum deplist package1 检查新版本 $> yum check-update
升级软件包 $> yum update [package1] [package2] [...] 清除缓存 $>yum clean all
yum 的常用选项-y 对所有的提问表示肯定的回答（yes）-c [config-file] 指定 yum 的配置文件
config file 的路径支持 http、ftp、urls and local file paths -d [number] 指定调试信息显示级别 [0-10]，默认为 2。等价于 debuglevel=n -e [number] 指定错误信息显示级别 [0-10]， 0 表示只显示重要的错误信息，默认为 2,等价于 errorlevel=0,1...10
-x, --exclude=package 排除哪些安装包。等价于 exclude=....
yum、apt-get 与 wget 用法及区别
yum 包格式为 rpm，安装 rpm 包的命令是“rpm – 参数”,Debian(ubuntu\debian)安装包格式为 deb
安装 deb 包的命令是“dpkg – 参数” 包管理工具 apt-get wegt 不安装，只是下载工具，可通过 http\https\ftp 下载
src 源代码包的编译与安装源码安装优点配置灵活，可随意去掉某些不重要模块，移植性也更好。有利于程序开发人员学习和进行二次开发。源码安装缺点难度偏大，容易出一些初学者感觉无所适从的错误提示信息。
基于 src 安装的步骤下载并用 tar 命令释放 src-tar 包;看安装说明(install|readme)。 例如，$>tar -zxf httpd-2.2.17.tar.gz -C /usr/src/执行如下命令，进行编译准备， $> ./configure –prefix /usr/local/apache #指定应用安装目录该命令用来设置编译器，并确定相关参数，产生 MakeFile 文件。执行如下命令，进行编译
$> make #要利用前一步生成的 MakeFile 文件用如下命令，进行软件安装$> make install
执行会把编译产生的可执行文件复制到正确的位置，比如 /usr/local/apache 目录下。删除安装过程产生的临时文件$> make clean
设备驱动程序 Linux 中，允许进程将设备驱动程序当作“文件”，以文件方式与它们通信交流。
每个已安装的设备驱动程序都对应一个文件名，对应着一个被称为设备节点(device node)的内存索引节点。
内核探测到一个新设备，并找到驱动后，就会在/dev 目录下添加一个设备节点。Linux 所有设备驱动，以文件形式存储在文件系统的/lib/modules/version 目录下，由系统或用户“按需”加载。当内核首次访问某个设备时，就会从文件系统装载这个设备驱动。
运行 lsmod 命令可查看已加载的内核驱动模块列表，直接查看目录/proc/modules 可获得同样信息（lsmod 可读性更好些）。
Linux 硬件配置信息系统探测到的硬件信息，包括引导启动时显示的所有信息，都被记录在一个不断滚动更新的缓冲日志文件/var/log/dmesg 中；可用 dmesg 命令查看。
内核还把探测到的一些关键硬件信息，以虚拟文件方式，分类组织在虚拟文件系统目录/proc 下。通过查看/proc 下的不同文件内容，可以获得一些关键硬件信息。
查看 CPU 支持及配置
探查机器的硬件配置$> cat cpuinfo 显示 CPU 型号、速度、内部缓存大小等。查看内存配置 $> cat /proc/meminfo 查看硬盘配置$> ls /proc/ide
$> ls /proc/scsi$> fdisk –l #查看磁盘分区情况
查看 PCI 设备 lspci –v 命令列出了所有探测到的 PCI 设备
运行脚本的三种方法：1）在指定的 Shell 下执行，将脚本程序名作为 Shell 的第一个参数$>bash showinfo [脚本程序自带参数]；（2）使用“.”命令执行脚本， “.” 后要有空格$>. showinfo [脚本程序自带参数]; (3) 将脚本设置为可执行，然后当做外部命令执行$>chmod a+x showinfo$>./showinfo [脚本程序自带参数];
分隔符：；按命令出现的先后，顺序执行、&&先执行前面的命令，若成功，才接着执行后面命令，若失败，不执行后面命令、||先执行前面的命令，若成功，不执行后面命令，若失败才执行后面命令、后缀&后台方式执行命令
变量：**用户自定义变量**：定义 abc=“he is”，引用 echo $abc 可用 unset 删除变量或变量的值、**环境变量**：HOME：用户主目录，默认值为/home/<用户名>；
LOGNAME：登录用户名； MAIL：用户邮件存放路径，默认值为/var/spool/mail/<用户名>； PATH：命令搜索路径； PWD：用户当前工作目录路径； SHELL：默认 Shell 的路径名； TERM：使用的终端名；、位置变量:例如 set one two 则$0 为 bash(a.sh m 则为 a.sh)，$1 为 one、预定义特殊变量：shell 已经定义好的：$# 命令行上的参数个数，不包括$0。
$? 最后执行命令的退出代码，0 表示成功，其它值表示失败、$$ 当前进程的 pid、$! 最后一个后台运行进程的进程号、$* 命令行所有参数构成的一个字符串、$@ 用双引号括起的命令行各参数拼接构成的一个字符串

变量替换：设置变量默认值${var:-defaultValue}替换变量时， var 值为空 ，则取 defaultValue，但不改变 var 的值。使用格式${var:=defaultValue}， var 值为空，则取 defaultValue，同时会改变设置 var 的值。 使用格式 ${var:+defaultValue} ， var 值非空 ，才取 defaultValue，但不改变 var 的值。**带错误检查的变量替换**使用${var:?message}替换变量时，如果 var 值非空则取 var，否则若为空则显示错误信息提示 message。
使用特殊字符 1）双引号， 仅能消除空格、制表符的特殊含义；2）单引号，能消除所有特殊字符的特殊含义；3）反单引号（`）括起的字符串， 被 Shell 解释为命令，执行时用命令输出结果代替被反单引号对界限的那个部分。4）反斜杠，消除单个字符的特殊含义。
变量的算术运算：只能进行整数间的运算，例：let sum=5/2 在 expr 中，运算符两边与操作数之间必须有空格，小括号要转义:**echo `expr 1 + 2`**；但 let 则没有这个要求，运算符前后有无空格均可，小括号不需转义，但=前后不能有空格。
条件表达式测试语法：test 条件表达式，或 [条件表达式]（注意要在条件两边加空格）；**字符串**：string1 = string2（相等为真），string1 != string2 (不相等为真)、test -n string (长度不为 0 为真)，test string 或 test -z string(长度为 0 为真) **数值**：Int1 -eq int2(ne:不等，ge:greater than or equal\（需用括号(())括起来)<:((“$n” < “$m”))**复合**:逆否!;与-a;或:-o 例：大于 0 且小于 10 表达式为: [x -ge 0 -a x -lt 10]
控制语句
if [条件表达式 1]
then
<命令序列 1> # 序列中可嵌套 if 语句
exit 1（结束？）
elif [条件表达式 2] #允许有多个 elif
<命令序列 2>
else
<命令序列 x>
fi
case <变量名引用> in
v1|v12) 命令列表 1;; (# 选项值必须以右括号结尾；若匹配多个离散值，用|分隔)
v2|v22) 命令列表 2;;
09|9) 命令列表 3;;
10) 命令列表 4;;
esac
for 循环变量名 in 列表
do
echo “In the loop for ${循环变量名} “
done
while [$n -gt 0]
do
echo “In the loop for ${循环变量名} “
done
break [n] # 跳出当前循环，带 n 则跳出 n 层循环；
continue [n] # 跳回循环体第 1 条，带 n 跳过内部的 n 层循环体执行；
变量的作用域
局部变量只在用户当前的 Shell 生命期中才有意义。如果在 Shell 中启动另一
个进程或退出，局部变量的值将变为无效。
全局变量通过 export 命令来定义，可以被任何运行的子 Shell 引用，子 Shell 无法改变父 Shell 中全局变量的值， 只能改变其副本的值；子 Shell 中局部变量的使用优先于全局变量
函数 function 函数名 () #function 可以省略
{
语句序列;
}
函数在使用前必须定义，因此应将函数定义放在脚本开始的部分。在函数中使用 return 结束本次函数执行，而使用 exit 会直接结束退出包含函数的当前脚本程序。当 return/exit 不带参数时，返回值有函数中最后一条命令决定
调用函数仅使用其函数名，例子:
#!/bin/sh
# A function
hello()
{
echo “hello, today’s date is `date`”
}
echo “going to call test function:”
hello
运行脚本， $> .xprog
结果： going to call test function:
hello, today’s date is <当前日期显示输出串>