|  |
| --- |
| LINUX操作基础及shell简介（第二周） |
| 计划学时 | 4 | 授课时间 | 第 2 周  |
| 步骤 | 教学内容 |
| 明确任务 | 1、LINUX启动过程2、shell和命令基础3、LINUX常用操作命令 |
| 项目引导 | 1、LINUX启动过程有几个阶段？2、什么是shell？3、LINUX命令有哪些？ |
| 操作训练 | 1、LINUX启动过程linux启动时我们会看到许多启动信息。Linux系统的启动过程并不是大家想象中的那么复杂，其过程可以分为5个阶段：·  内核的引导。·  运行 init。·  系统初始化。·  建立终端 。·  用户登录系统。*init程序的类型：*·  **SysV:** init, CentOS 5之前, 配置文件： /etc/inittab。·  **Upstart:** init,CentOS 6, 配置文件： /etc/inittab, /etc/init/\*.conf。·  **Systemd：** systemd, CentOS 7,配置文件： /usr/lib/systemd/system、 /etc/systemd/system。**内核引导**当计算机打开电源后，首先是BIOS开机自检，按照BIOS中设置的启动设备（通常是硬盘）来启动。操作系统接管硬件以后，首先读入 /boot 目录下的内核文件。**运行init**init 进程是系统所有进程的起点，你可以把它比拟成系统所有进程的老祖宗，没有这个进程，系统中任何进程都不会启动。init 程序首先是需要读取配置文件 /etc/inittab。**运行级别**许多程序需要开机启动。它们在Windows叫做"服务"（service），在Linux就叫做"守护进程"（daemon）。init进程的一大任务，就是去运行这些开机启动的程序。但是，不同的场合需要启动不同的程序，比如用作服务器时，需要启动Apache，用作桌面就不需要。Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"运行级别"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。**Linux系统有7个运行级别(runlevel)：**·  运行级别0：系统停机状态，系统默认运行级别不能设为0，否则不能正常启动·  运行级别1：单用户工作状态，root权限，用于系统维护，禁止远程登陆·  运行级别2：多用户状态(没有NFS)·  运行级别3：完全的多用户状态(有NFS)，登陆后进入控制台命令行模式·  运行级别4：系统未使用，保留·  运行级别5：X11控制台，登陆后进入图形GUI模式·  运行级别6：系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动 **系统初始化**在init的配置文件中有这么一行： si::sysinit:/etc/rc.d/rc.sysinit　它调用执行了/etc/rc.d/rc.sysinit，而rc.sysinit是一个bash shell的脚本，它主要是完成一些系统初始化的工作，rc.sysinit是每一个运行级别都要首先运行的重要脚本。它主要完成的工作有：激活交换分区，检查磁盘，加载硬件模块以及其它一些需要优先执行任务。l5:5:wait:/etc/rc.d/rc 5这一行表示以5为参数运行/etc/rc.d/rc，/etc/rc.d/rc是一个Shell脚本，它接受5作为参数，去执行/etc/rc.d/rc5.d/目录下的所有的rc启动脚本，/etc/rc.d/rc5.d/目录中的这些启动脚本实际上都是一些连接文件，而不是真正的rc启动脚本，真正的rc启动脚本实际上都是放在/etc/rc.d/init.d/目录下。而这些rc启动脚本有着类似的用法，它们一般能接受start、stop、restart、status等参数。/etc/rc.d/rc5.d/中的rc启动脚本通常是K或S开头的连接文件，对于以 S 开头的启动脚本，将以start参数来运行。而如果发现存在相应的脚本也存在K打头的连接，而且已经处于运行态了(以/var/lock/subsys/下的文件作为标志)，则将首先以stop为参数停止这些已经启动了的守护进程，然后再重新运行。这样做是为了保证是当init改变运行级别时，所有相关的守护进程都将重启。至于在每个运行级中将运行哪些守护进程，用户可以通过chkconfig或setup中的"System Services"来自行设定。**建立终端**rc执行完毕后，返回init。这时基本系统环境已经设置好了，各种守护进程也已经启动了。init接下来会打开6个终端，以便用户登录系统。在inittab中的以下6行就是定义了6个终端：1:2345:respawn:/sbin/mingetty tty12:2345:respawn:/sbin/mingetty tty23:2345:respawn:/sbin/mingetty tty34:2345:respawn:/sbin/mingetty tty45:2345:respawn:/sbin/mingetty tty56:2345:respawn:/sbin/mingetty tty6从上面可以看出在2、3、4、5的运行级别中都将以respawn方式运行mingetty程序，mingetty程序能打开终端、设置模式。同时它会显示一个文本登录界面，这个界面就是我们经常看到的登录界面，在这个登录界面中会提示用户输入用户名，而用户输入的用户将作为参数传给login程序来验证用户的身份。**用户登录系统**一般来说，用户的登录方式有三种：·  （1）命令行登录·  （2）ssh登录·  （3）图形界面登录对于运行级别为5的图形方式用户来说，他们的登录是通过一个图形化的登录界面。登录成功后可以直接进入 KDE、Gnome 等窗口管理器。而本文主要讲的还是文本方式登录的情况：当我们看到mingetty的登录界面时，我们就可以输入用户名和密码来登录系统了。Linux 的账号验证程序是 login，login 会接收 mingetty 传来的用户名作为用户名参数。然后 login 会对用户名进行分析：如果用户名不是 root，且存在 /etc/nologin 文件，login 将输出 nologin 文件的内容，然后退出。这通常用来系统维护时防止非root用户登录。只有/etc/securetty中登记了的终端才允许 root 用户登录，如果不存在这个文件，则 root 用户可以在任何终端上登录。/etc/usertty文件用于对用户作出附加访问限制，如果不存在这个文件，则没有其他限制。<="" p="" data-filtered="filtered" style="color: rgb(51, 51, 51); font-family: "Helvetica Neue", Helvetica, "PingFang SC", "Hiragino Sans GB", "Microsoft YaHei", "Noto Sans CJK SC", "WenQuanYi Micro Hei", Arial, sans-serif; font-size: 12px; font-style: normal; font-variant-ligatures: normal; font-variant-caps: normal; font-weight: 400; letter-spacing: normal; orphans: 2; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 2; word-spacing: 0px; -webkit-text-stroke-width: 0px; background-color: rgb(255, 255, 255); text-decoration-style: initial; text-decoration-color: initial;">**图形模式与文字模式的切换方式**Linux预设提供了六个命令窗口终端机让我们来登录。默认我们登录的就是第一个窗口，也就是tty1，这个六个窗口分别为tty1,tty2 … tty6，你可以按下Ctrl   Alt   F1 ~ F6 来切换它们。如果你安装了图形界面，默认情况下是进入图形界面的，此时你就可以按Ctrl   Alt   F1 ~ F6来进入其中一个命令窗口界面。当你进入命令窗口界面后再返回图形界面只要按下Ctrl   Alt   F7 就回来了。如果你用的vmware 虚拟机，命令窗口切换的快捷键为 Alt   Space   F1~F6. 如果你在图形界面下请按Alt   Shift   Ctrl   F1~F6 切换至命令窗口。**Linux 关机**在linux领域内大多用在服务器上，很少遇到关机的操作。毕竟服务器上跑一个服务是永无止境的，除非特殊情况下，不得已才会关机。正确的关机流程为：sync > shutdown > reboot > halt关机指令为：shutdown ，你可以man shutdown 来看一下帮助文档。例如你可以运行如下命令关机：sync 将数据由内存同步到硬盘中。shutdown 关机指令，你可以man shutdown 来看一下帮助文档。例如你可以运行如下命令关机：shutdown –h 10 ‘This server will shutdown after 10 mins’ 这个命令告诉大家，计算机将在10分钟后关机，并且会显示在登陆用户的当前屏幕中。shutdown –h now 立马关机shutdown –h 20:25 系统会在今天20:25关机shutdown –h 10 十分钟后关机shutdown –r now 系统立马重启shutdown –r 10 系统十分钟后重启reboot 就是重启，等同于 shutdown –r nowhalt 关闭系统，等同于shutdown –h now 和 poweroff最后总结一下，不管是重启系统还是关闭系统，首先要运行 **sync** 命令，把内存中的数据写到磁盘中。关机的命令有 **shutdown –h now halt poweroff** 和**init 0** , 重启系统的命令有 **shutdown –r now reboot init 6**。**远程登录LINUX：**LINUX一般作为服务器使用，而服务器一般放在机房，不可能在机房操作LINUX服务器，这时我们就需要规程登录到LINUX服务器来管理维护系统。LINUX系统中是通过SSH服务实现的远程登录，默认SSH服务器端口号为22。Window系统上 Linux 远程登录客户端有SecureCRT, Putty, SSH Secure Shell等，我们以SecureCRT来登录远程服务器。**LINUX忘记密码解决方法：**进入单用户模式更改一下root密码即可。步骤：重启LINUX系统：3秒之内要按下回车，出现如下界面：然后输入e：在 第二行最后边输入 single，有一个空格。具体方法为按向下尖头移动到第二行，按"e"进入编辑模式：后面加上single回车：最后按"b"启动，启动后就进入了单用户模式了：此时已经进入到单用户模式了，你可以更改root密码了。更密码的命令为 passwd：**2、shell和命令基础**主要介绍Shell的相关概念、命令格式、通配符的使用方法，然后分别介绍了各类LINUX常用命令的使用Shell。**2.1shell简介****（1）什么是shell** Shell是系统的用户界面，提供了用户与内核进行交互操作的一种接口(命令解释器), Shell接收用户输入的命令并把它送入内核执行，在用户与系统之间进行交互。Shell 在Linux 系统中具有极其重要的地位。外层应用程序命令解释器Shell系统核心硬件**（2）Shell的功能**命令解释器是Shell 最重要的功能。Linux 系统中的所有可执行文件都可以作为Shell命令来执行。将Linux的可执行文件进行分类，如下表所示。

|  |
| --- |
| Linux系统上可执行文件的分类 |
| 类 别 | 说 明 |
| Linux命令 | 存放在/bin、/sbin目录下的命令 |
| 内置命令 | 出于效率的考虑，将一些常用命令的解释段序构造在Shell内部 |
| 实用程序 | 存放在/usr/bin、/usr/sbin、/usrlocal/bin、/usrlocal/sbin等目录下的实用程序 |
| 用户程序 | 用户程序经过编译生成可执行文件后可作为Shell命令运行 |
| Shell脚本 | 由Shell语言编写的批处理文件 |

Shell是如何完成命令解释的呢？命令解释过程如下：是外部命令或实用程序用户输入命令提交给Shell是否为内置命令在系统中查找该命令的文件并调入内存执行内核中的系统功能调用是内置命令当用户提交了一个命令后，Shell 首先判断是否为内置命令(由Shell自身负责解释)，如果是就通过Shell的解释器将其解释为系统功能调用并转交给内核执行；若是外部命令或实用程序，就试图在硬盘中查找该命令并将其调入内存，再将其解释为系统功能调用并转交给内核执行。在查找该命令时分为两种情况:①用户给出了命令的路径，Shell 就沿着用户给出的路径进行查找，若找到则调入内存，若没找到则输出提示信息。②用户没有给出命令的路径，Shell 就在环境变量PATH所制定的路径中依次进行查找，若找到则调入内存，若没找到则输出提示信息。此外，Shell 还具有如下功能。●通配符、命令补全、别名机制、命令历史。●重定向、管道、命令替换、Shell编程等。**（3）Shell 的主要版本**下表中列出了几种常见的Shell版本。RHEL/CentOS 下默认的Shell是bash。

|  |  |
| --- | --- |
| 版本 | 说明 |
| Bourme Again Shell(bash、bsh的扩展) | bash是大多数Linux系统的默认Shell，bash与bsh完全向后兼容，并且在bsh的基础上增加和增强了很多特性。bash也包含了很多C Shell和Kom Shell中的优点。bash有很灵话和强大的编程接口，同时又有很友好的用户界面 |
| Kom Shell(ksh) | Kom Shell(ksh)由Dave Korm所写。是UNIX系统上的标准Shell。另外，在Linux环境下有一个专门为Linux系统编写的Kom Shell的扩展版本，即Public Domain Korm Shell(pdksh) |
| tcsh(csh的扩展) | tcsh是C Shell的扩展。tcsh与csh完全向后兼容。但它包含了更多的使用户感觉方便的新特性，其最大的提高是在命令行编辑和历史浏览方面 |

**（4）Shell的元字符**在Shell 中有一些具有特殊的意义字符，称为Shell 元字符(Shell Metacharacters)。若不以特殊方式指明，Shell 并不会把它们当作普通文字符使用。下表中简单介绍了常用的Shell元字符的含义。

|  |  |
| --- | --- |
| 元字符 | 含 义 |
| · | 代表任意字符串 |
| ? | 代表任意字符 |
| / | 代表根目录或作为路径间隔符使用 |
| \ | 转义字符。当命令的参数要用到保留字时，要在保留字前面加上转义字符 |
| \<Enter> | 续行符。可以使用续行符将一个命令行分写在多行上 |
| $ | 变量值置换，如$PATH表示环境变量PATH的值 |
| ' | 在'... '中间的字符均被当作文字处理。指令、文件名、保留字等都不再具有原来的意义 |
| " | 在"..."中间的字符公被当作文字处理并允许变量值置换 |
| ` | 命令替换，置换`...`中命令的执行结果 |
| < | 输入重定向字符 |
| > | 输出重定向字符 |
| | | 管道字符 |
| & | 后台执行字符。在一个命令之后加上字符“&"。该命令就会以后台方式执行 |
| ; | 分割顺序执行的多个命令 |
| ( ) | 在子Shell中执行一组命令 |
| {} | 在当前Shelll中执行一组命令 |
| ! | 执行命令历史记录中的命令 |
| ~ | 代表登录用户的宿主目录(自家目录) |

**2.2命令格式和通配符****（1）命令格式**Shell命令的一般格式为:**cmd [-options] [arguments]**其中，cmd是命令名； options是选项；arguments是参数，即操作对象。说明:●最简单的Shell命令只有命令名，复杂的Shell命令可以有多个选项和参数。●选项和参数都作为Shell命令执行时的输入，它们之间用空格分隔开。●单字符参数前使用一个减号(-)， 单词参数前使用两个减号(--)。●多个单字符参数前可以只使用一个减号。●操作对象(arguments)可以是文件也可以是目录，有些命令必须使用多个操作对象，如cp命令必须指定源操作对象和目标操作对象。●并非所有命令的格式都遵从以上规则，如dd、 find 等。例如:$ls$ls -lra /home$ ls -help$cat /etc/passwd ./myfile具有以上格式的字符串习惯地称为命令行，命令行是用户与Shell间对话的基本单位。**（2）目录和文件名的命名规则**在Linux下可以使用长文件或目录名，可以给目录和文件取任何名字，但必须遵循下列规则:●除了/之外，所有的字符都合法。●有些字符最好不用，如空格符、制表符、退格符和字符: ? ，@ # $ & 0 \ | ； ‘’ “” <>等。●避免使用+、-或.来作为普通文件名的第一个字符。●大小写敏感。@以.开头的文件或目录是隐含的。**（3）通配符**通配符主要用于用户方便描述目录或文件。下表中是常用的通配符及其说明。

|  |  |  |  |
| --- | --- | --- | --- |
| 通配符 | 说 明 | 通配符 | 说 明 |
| **·** | 匹配任何字符和任何数目的字符 | […] | 匹配任何包含在括号里的单字符 |
| ? | 匹配任何单字符 | [!...] | 匹配任何不包含在括号里的单字符 |

**注意：**\*能匹配文件或目录名中的.，但不能匹配首字符是的文件或目录名。要匹配隐含文件应该使用.\*。通配符在指定一系列的文件名时非常有用，下表列举了一些使用通配符的例子。

|  |  |
| --- | --- |
| 举 例 | 说 明 |
| ls \*.c | 列出当前目录下的所有C语言源文件 |
| ls /home/\*/\*.c | 列出/home目录下所有子目录中的所有C语言源文件 |
| ls n\*.conf | 列出当前目录下的所有以字母n开始的conf文件 |
| ls test?.dat | 列出当前目录下以test开始的，随后一个字符是任意的.dat文件 |
| ls [abc]\* | 列出当前目录下首字符是a或b或c的所有文件 |
| ls [!abc]\* | 列出当前目录下首字符不是a或b或c的所有文件 |
| ls [a-zA-Z]\* | 列出当前目录下首字符是字母的所有文件 |

**2.3文件及Linux目录结构****（1）什么是文件**在Linux系统上，文件被看作是字节序列。这种概念使得所有的系统资源有了统一的标识，这些资源包括普通文件或目录、磁盘设备、控制台（键盘、显示器）、打印机等。对这些资源的访问和处理都是通过字节序列的方式实现的。Linux系统下的文件类型包括：* 普通文件（-）
* 目录（d）
* 符号链接（l）
* 字符设备文件（c）
* 块设备文件（b）
* 套接字（s）
* 命名管道（p）

**（2）普通文件**普通文件就是字节序列，Linux 并没有对其内容规定任何的结构。普通文件可以是程序源代码(C、 C++、Python、Perl 等)、可执行文件(文件编辑器、数据库系统、出版工具、绘图工具等)、图片、声音、图像等。Linux 不会区别对待这些文件，只有处理这些文件的应用程序才会根据文件的内容为它们赋予相应的含义。在DOS或Winodws环境中，所有的文件名后缀就能表示该文件的类型，如\* .exe表示可执行文件，\*.bat 表示批处理文件。在Linux 环境下，只要是可执行的文件并具有可执行属性则就能执行，不管其文件名后缀是什么。但是对一些数据文件一般也遵循一些文件名后缀规则，下表中列出了一些常用的文件后缀。

|  |  |
| --- | --- |
| 举 例 | 说明 |
| \*.txt | 文本文件 |
| \*.conf | 配置文件 |
| \*htm/\*xm/\*yml/\*sql | HTMLXMLYAMLSQL文件 |
| \*.cl\*.cpp | C/C++语言源程序文件 |
| \*.so/\*.ko/\*.lib | 模块文件、库文件 |
| \*sh/\*.php/\*.py/\*.pV\*.rb | Shell/PHP/Python/Per/Ruby脚本文件 |
| \*.rpm | RPM包文件 |
| \*.tar | tar存档文件 |
| \*.gz/\*.bz2/\*xz | 由gzip/bzip2/xz生成的压缩文件 |
| \*.tar.gz/\*.tgz/\*.tar.bz2/\*.tbz/\*.tar.xz/\*.bcz | 压缩后的tar包文件 |
| \*.lock | 用于表示某个程序或某种服务正在运行的锁文件 |
| \*~ | 备份文件 |

**（3）目录和硬链接**目录文件是由一组目录项组成，目录项可以是对其他文件的指向也可以是其下的子目录指向。实际上，一个文件的名称是存储在其父目录中的，而并非同文件内容本身存储在一起。将两个文件名(存储在其父目录的目录项中)指向硬盘上一个存储空间，对两个文件中的任何一个的内容进行修改都会影响到另一个文件， 这种链接关系称为硬链接。硬链接文件实际上就是在某目录中创建目录项，从而使不止一个目录可以引用到同一个文件。它可以由ln命令建立。首先查看一下目录中的文件情况。使用ln命令建立文件textfile1的硬链接文件textfile4。该命令产生一个新的文件textfile4,和已经存在的文件textfile1建立起硬链接关系:可以看出，textfile4和textfilel的大小相同，内容相同。再看详细信息的第2列，原来textfile1的链接数是1，说明这一块硬盘存储空间有textfile1 一个文件指向它，而建立起textfile1和textfile4的硬链接关系之后，这块硬盘空间就有textfilel 和textfile4两个文件同时指向它，所以textfile1和textfile4的链接数就都变为了2。因为两个文件指向一块硬盘空间，所以如果现在修改textfile4的内容为This is file2.，再查看textfile1的内容，就会有:如果删除其中的一个文件(不管是哪一个)， 就是删除了该文件和硬盘空间的指向关系，该硬盘空间不会释放，另外一个文件的内容也不会发生改变，但是目录详细信息中的链接数会减少，见如下信息。硬链接并不是一种特殊类型的文件，只是在同一个文件系统中允许多个目录项指向同一个文件的一种机制。**（4）符号链接**符号链接又称软链接，是指将一个文件指向另外一个文件的文件名。这种符号链接的关系由ln -s命令行建立。首先查看一下目录中的文件信息。使用ln命令和-s选项建立文件textfilel的符号链接文件textfile5。该命令产生一个新的文件textfile5，和已经存在的文件textfile1建立起符号链接关系。可以看出textfile5这个文件很小，因为它只是记录了要指向的文件名而已，请注意从文件textfile5指向文件textfile1的指针。为什么cat命令显示的textfile5的内容与textfile1相同呢?因为cat命令在寻找textfile5的内容时，发现textfile5是一个符号链接文件，根据textfile5记录的文件名找到了textfile1文件，然后将textfile1的内容显示出来。明白了textfile1和textfile5的符号链接关系，就可以理解为什么textfile1的链接数仍然为1，这是因为textfile1指向的硬盘空间仍然只有textfile1一个文件在指向。如果现在删除了textfile5,对textfile1并不产生任何影响；而如果删除了textfile1，那么textfile5就因无法找到文件名称为textfile1的文件而成为死链接。**（5）设备文件**设备是指计算机中的外围硬件装置，即除了CPU和内存以外的所有设备。通常，设备中含有数据寄存器或数据缓存器、设备控制器，用于完成设备同CPU或内存的数据交换。在Linux下，为了屏蔽用户对设备访问的复杂性，采用了设备文件，即可以通过像访问普通文件一样的方式对设备进行读写访问。设备文件用来访问硬件设备，包括硬盘、光驱、打印机等。每个硬件设备至少与一个设备文件相关联。设备文件分为字符设备(如键盘)和块设备(如磁盘)。Linux 下设备名以文件系统中的设备文件的形式存在。所有的设备文件存放在/dev目录下。下面对常用设备列表说明，如下表所示。

|  |  |
| --- | --- |
| 设备文件 | 说 明 |
| /dev/sd\* | SCSI/SAS、PATA/SATA、USB硬盘设备，如sdal表示第1块硬盘的第1个分区；sdb2表示第2块硬盘的第2个分区 |
| /dev/sr0 | 光驱设备 |
| /dev/console | 系统控制台 |
| /dev/tty\* | 本地终端设备 |
| /dev/pts/\* | 伪终端设备 |
| /dev/ppp\* | ppp设备。PPP(Point-to-Point)协议设备，用于传统的拨号上网 |
| /dev/lp\* | 表示并口设备，如IpO表示第1个并口设备；lpl表示第2个并口设备 |
| /dev/null | 空设备。可将其视为“黑洞"，所有写入它的内容都会丢失，通常用于屏蔽命令行输出 |
| /dev/zero | 零设备。可以产生连续不断的进制的零流，通常用于创建指定长度的空文件 |

在/dev目录下有许多链接文件，使用这些链接能够方便地使用系统中的设备。例如，可以通过/dev/cdrom而不是/dev/sr0来访问光驱。**（6）套接字和命名管道**套接字和命名管道是Linux环境下实现进程间通信(IPC)的机制。命名管道(FIFO)文件允许运行在同一台计算机上的两个进程之间进行通信。套接字(socket)允许运行在不同计算机上的进程之间相互通信。套接字和命名管道通常是在进程运行时创建或删除的，一般无需系统管理员干预。**（7）熟悉Linux的目录结构**Linux的目录结构遵从文件系统层次结构标准(File system Hierarchy Standard, FHS)。下表中解释了由FHS所规定的存放特定类型的文件位置。

|  |  |
| --- | --- |
| 目录名 | 内容说明 |
| bin | 存放二进制的可执行程序 |
| boot | 存放用于系统引导时使用的各种文件 |
| dev | 用于存放设备文件，用户可以通过这些文件访问外部设备 |
| etc | 存放系统的配置文件 |
| home | 存放所有用户文件的根目录，有一个用户在该目录下就有一个与该用户名相对应的子目录，当用户登录时就进入其用户名对应的子目录 |
| lib/lib64 | 存放根文件系统中的程序运行所需要的共享库及内核模块 |
| lost+found | 存放一些系统检查结果，发现不合法的文件或数据都存放在这里，通常此目录是空的，除非硬盘遭受了不明的损坏 |
| mnt | 临时文件系统的挂载点目录 |
| media | 即插即用型存储设备的挂我点自动在这个目录下创建，如CD/DVD等 |
| opt | 第三方软件的存放目录 |
| proc | 是一个虚拟文件系统，存放当前内存的映射，主要用于在不重启机器的情况下管理内核 |
| root | 超级用户目录 |
| sbin | 类似bin目录，也存放二进制可执行文件，但是只有root才能访问 |
| srv | 系统对外提供服务的目录，如Web虚拟主机等 |
| tmp | 用于放置各种临时文件 |
| usr | 用于存放系统应用程序 |
| var | 用于存放需要随时改变的文件，如系统日志、脱机工作目录等 |

 |
| 知识深化 | **注意：**（1）在Linux环境下，文件是归类存放的。初学Linux的朋友应该熟悉特定类型的文件的存放位置。（2）对于Linux的初学者而言，在不知道自己究竟在做什么的情况下，不要轻易操作系统目录，如/proc、 /boot、 /etc、 /usr、 /var 等。（3）用户可以使用如下命令获得Linux文件层次结构的说明:$ man hier |
| 归纳总结 | Linux操作系统和windows有着根本的区别，需要我们好好考虑。 |
| 课堂作业 | 1.熟练掌握Linux系统分区的方法。（思考：/ 、/boot、 /home和/swap如何划分，分多少空间为宜？）2.Linux操作系统安装完毕后，熟悉系统界面及基本操作。3.配置网络并测试网络环境，让虚拟机中的Linux能上外网。 |
| 课后作业 | (１)如何获得命令帮助? help命令和--help命令选项的作用分别是什么?(２)常用的Linux信息获取命令有哪些?各自的功能是什么?(３)如何正确地关闭和重新启动Linux系统? |
| 教学体会 | 学生的学习积极性很高，对于linux很感兴趣。但是对于命令和一些操作还是需要加强练习。 |