

模块 1 Linux 操作系统的安装与启动

本模块从零基础入门者的视角讲解 Linux 操作系统(简称 Linux 系统或 Linux)的来历和发展历程,尽量简明扼要地进行叙述,清楚地交代学习本课程需要使用的软件,同时完整地演示 VMware Workstation 15 虚拟机的安装与配置,以及 Red Hat Enterprise Linux(以下简称 RHEL 8)系统的安装、配置与 systemd 初始化进程,最后验证采用 VMware Workstation 虚拟机学习的好处,即可以进行系统快照的备份与恢复。

通过本模块的学习,学生将达到以下职业能力目标和要求:

- (1)了解 Linux 操作系统的来历和发展历程;
- (2)掌握虚拟机(VMware Workstation、VirtualBox 等)的安装及使用方法;
- (3)掌握在虚拟机中安装 Linux 的方法;
- (4)掌握 RHEL 8/CentOS 8 操作系统的安装与启动方法;
- (5)掌握在虚拟机中备份与恢复操作系统的方法。

1.1 Linux 操作系统的来历和发展历程

Linux 操作系统作为一款免费、自由、开放的操作系统,它的发展势不可挡,而它的前世今生在很多教材和专业书籍中都有充分的介绍,本书将用一段话来总结 Linux 操作系统的来历和发展历程。

Linux 的前身 UNIX 系统于 1969 年在 AT&T 的贝尔实验室诞生,20 世纪 70 年代的 UNIX 系统由于开源且免费而逐步盛行,但在 1979 年时,AT&T 公司宣布了 UNIX 系统的商业化计划,随之使得开源软件业转变成了版权式软件产业,源代码被当作商业机密,成为专利产品,人们再也不能自由地享受科技成果。于是在 1984 年,Richard Stallman 针对如此封闭的软件创作环境,发起 GNU 源代码开放计划并制定了著名的 GPL 许可协议。1987 年,GNU 计划获得了一项重大突破——gcc 编译器的发布,使得程序员可以基于该编译器编写属于自己的开源软件。

芬兰赫尔辛基大学的在校研究生 Linus Torvalds 在 1991 年 10 月编写了一款称为 Linux 的操作系统,因其较高的代码品质且基于 GNU GPL 许可协议的开放源代码特性,迅速引起了 GNU 计划和一大批程序员的支持,随后 Linux 系统便开始火热地发展。到了 1994 年 1 月,Bob Young 基于 Linux 系统内核集成了众多的源代码和程序软件,创办了 RedHat 公司及红帽系统并开始出售技术服务,进一步推动了 Linux 系统的普及。1998 年后,随着基于 GNU 源代码开放计划的 Linux 系统的火热,以 IBM 和 Intel 为代表的大量 IT 企业开始大力推动开放源代码软件的发展。到了 2020 年,Linux 内核已经发展到了 4.18 版



本,并已经有了数百个 Linux 系统版本,但它们依然统一使用 Linus Torvalds 开发并维护的 Linux 系统内核。RedHat 公司也成为开源行业及 Linux 系统使用的带头公司。

Linux 系统中所说的免费只是说 Linux 的内核免费,因此在 Linux 内核的基础上产生了众多的 Linux 版本。Linux 的发行版说简单点就是将 Linux 内核与应用软件打一个包。目前较知名的发行版有 RedHat、CentOS、Ubuntu、Debian、Fedora、SUSE 等,当然 Linux 发行版还有很多,这里仅列举出几个比较常用的 Linux 发行版本。

1.2 安装前的工具准备

本书是基于 RHEL 8 系统编写的,书中的内容及实验也适用于 CentOS、Fedora 等系统,其实所有版本的 Linux 都大同小异,只要学会了其中一个,学习其他发行版就相当简单了。

正所谓“工欲善其事,必先利其器”,要想学好 Linux 系统,得有一台装有 Linux 操作系统的计算机,学习者也不太可能再去买一台计算机来单独安装 Linux 操作系统进行学习,所以建议采用虚拟机软件来安装 Linux 操作系统供学习使用,使用虚拟机的好处有不少,如下仅罗列了 3 条:

- (1)可以在一台物理机上安装多个操作系统。
- (2)可以通过虚拟机在一台物理机上同时运行多个操作系统。
- (3)可以利用虚拟机进行软件测试,也可进行开机/关机状态快照备份。

不管采用何种虚拟机,最终目的都是通过虚拟机来模拟真实系统环境进行学习,本书采用的虚拟机是 VMware Workstation 15, Linux 操作系统镜像是 RHEL 8.0,工具准备好后接下来将进行 VMware Workstation 虚拟机的安装。

1.3 VMware Workstation 虚拟机的安装与配置

如果是第一次操作计算机与安装软件,请参照以下步骤进行虚拟机的安装与配置,如果比较熟悉 VMware Workstation、VirtualBox 等虚拟机软件的安装与配置,可以跳过这一部分内容,直接进行后面的 Linux 操作系统的安装与学习。

运行下载的 VMware Workstation 15 虚拟机安装软件,将会看到图 1-1 所示的程序安装向导初始界面。

在“欢迎使用 VMware Workstation Pro 安装向导”界面中,单击“下一步”按钮,如图 1-2 所示。



图 1-1 虚拟机程序安装向导初始界面



图 1-2 “欢迎使用 VMware Workstation Pro 安装向导”界面

在“最终用户许可协议”界面中，选择“我接受许可协议中的条款”复选框，然后单击“下一步”按钮，如图 1-3 所示。

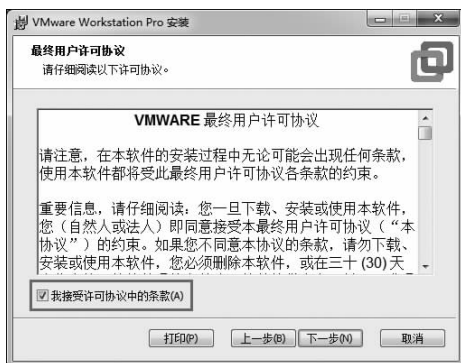


图 1-3 “最终用户许可协议”界面

在“自定义安装”界面中，可以采取默认设置的方式进行安装，如果要更改软件的安装路径，单击“更改”按钮进行安装路径的选择，然后单击“下一步”按钮，如图 1-4 所示。

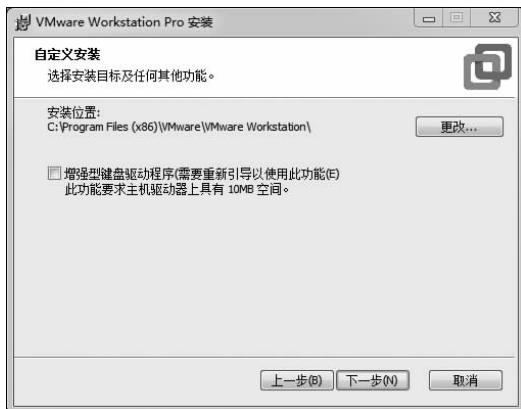


图 1-4 “自定义安装”界面

在“用户体验设置”界面中,可以采取软件默认的选择方式,也可以自由选择,最后单击“下一步”按钮,如图 1-5 所示。

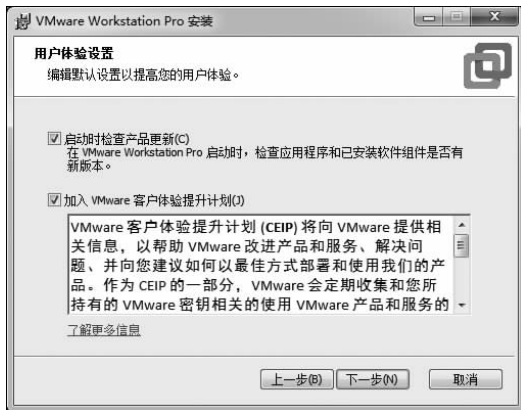


图 1-5 “用户体验设置”界面

在“快捷方式”界面中,可以采取软件默认选择的方式,也可以自由选择,最后单击“下一步”按钮,如图 1-6 所示。

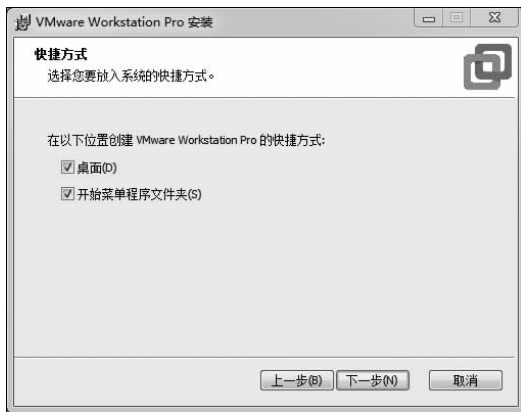


图 1-6 “快捷方式”界面



一切准备就绪后,单击“安装”按钮,即进入虚拟机安装等待过程,如图 1-7 和图 1-8 所示。



图 1-7 已准备好安装虚拟机



图 1-8 虚拟机安装等待过程

很快虚拟机显示安装完成(具体视计算机配置而定),然后单击“完成”按钮退出安装向导,或者单击“许可证”按钮,如图 1-9 所示。



图 1-9 虚拟机安装完成界面

如果在图 1-9 所示界面中单击“完成”按钮,虚拟机将有 30 天的试用期;如果单击“许可证”按钮,将弹出“输入许可证密钥”界面,输入正确的许可证密钥,然后单击“输入”按钮,如图 1-10 所示。



图 1-10 “输入许可证密钥”界面

最后单击“完成”按钮,将进入虚拟机软件的管理界面,如图 1-11 所示。



图 1-11 虚拟机软件的管理界面

这里要强调一点的是,不管采取 VMware Workstation、VirtualBox 还是其他虚拟机软件,最终的目的都是学习 Linux 系统相关知识,所以这里仅仅起一个抛砖引玉的作用,学生在学的过程中遇到问题可以先自行查阅相关资料尝试解决。例如,想学习 VirtualBox 虚拟机的安装,可以自行查阅相关资料进行安装学习。

在图 1-11 所示的管理器界面中单击“创建新的虚拟机”选项,然后在弹出的“新建虚拟机向导”对话框中选择“典型(推荐)”单选选项,如图 1-12 所示。

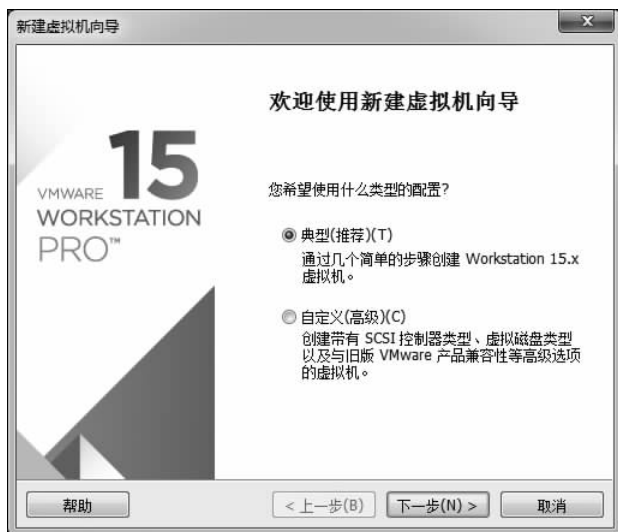


图 1-12 “新建虚拟机向导”对话框

单击“下一步”按钮,进入“安装客户机操作系统”界面,如图 1-13 所示。



图 1-13 “安装客户机操作系统”界面

选择“稍后安装操作系统”单选项,然后单击“下一步”按钮,进入“选择客户机操作系统”界面,如图 1-14 所示。

在图 1-14 中,选择客户机操作系统的类型为“Linux”,版本为“其他 Linux 4. × 或更高版本内核 64 位”,然后单击“下一步”按钮,进入“命名虚拟机”界面,如图 1-15 所示。



图 1-14 “选择客户机操作系统”界面

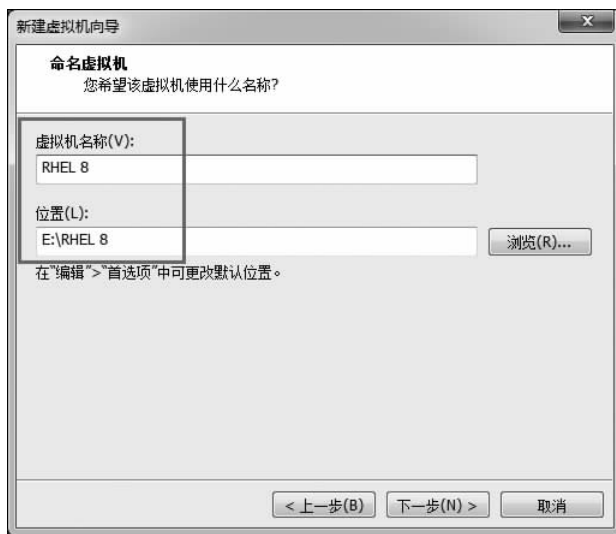


图 1-15 “命名虚拟机”界面

填写“虚拟机名称”，建议将“位置”设置在剩余空间比较多的磁盘，单击“下一步”按钮，进入“指定磁盘容量”界面。

将虚拟机的磁盘大小设置为 20 GB，并选择“将虚拟磁盘存储为单个文件”单选选项(目的是使文件不那么凌乱)，如图 1-16 所示。单击“下一步”按钮，进入“已准备好创建虚拟机”界面，如图 1-17 所示。

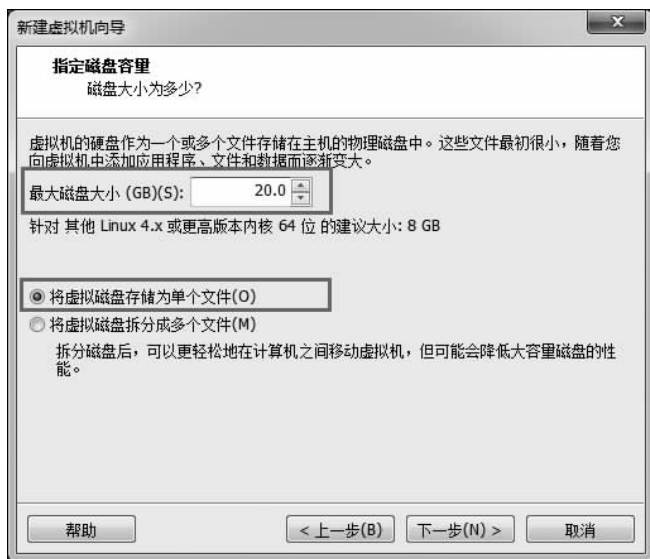


图 1-16 “指定磁盘容量”界面



图 1-17 “已准备好创建虚拟机”界面

单击“自定义硬件”按钮，弹出“硬件”对话框，将虚拟机系统内存设置最低为 1 GB，具体视计算机配置而定。选择光驱设备“新 CD/DVD(IDE)”，在“使用 ISO 映像文件”选项区中选中下载好的系统镜像文件，如图 1-18 所示。

VMware Workstation 虚拟机软件提供了多种可选的网络模式：“桥接模式”“NAT 模式”和“仅主机模式”，下面将解释。这里采用默认的“NAT 模式”，如图 1-19 所示。

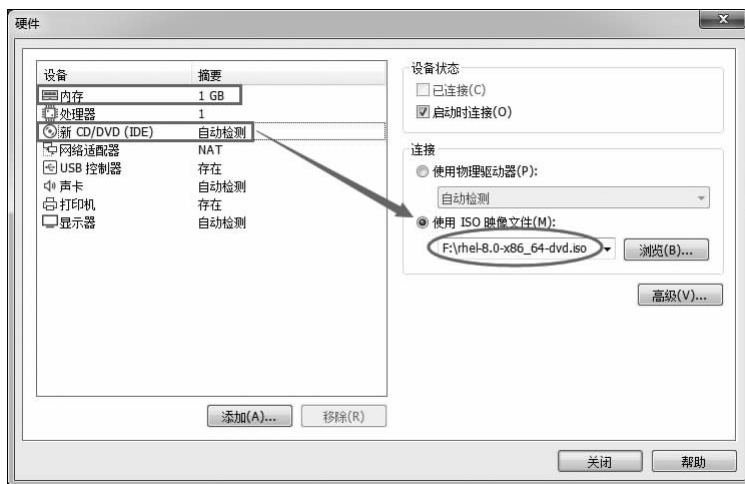


图 1-18 “硬件”对话框



图 1-19 设置虚拟机网络适配器

桥接模式就是将主机网卡与虚拟机的虚拟网卡利用虚拟网桥进行通信。在桥接的作用下,类似于把物理主机虚拟为一个交换机,所有桥接设置的虚拟机连接到这个交换机的一个接口上,物理主机也同样插在这个交换机中,所以所有桥接下的计算机网卡间都是交换模式,可以相互访问而不干扰。在桥接模式下,虚拟机 IP 地址需要与主机在同一个网段,如果需要联网,则网关与 DNS 需要与主机网卡一致,即虚拟机对外界来说就好比独立的物理计算机,其网络结构如图 1-20 所示。

如果网络 IP 资源紧缺,但是又希望虚拟机能够联网,这时 NAT 模式是最好的选择, NAT 模式借助虚拟 NAT 设备和虚拟 DHCP 服务器,使得虚拟机可以联网。在连接 VMnet8 虚拟交换机时,虚拟机会将虚拟 NAT 设备及虚拟 DHCP 服务器连接到 VMnet8 虚拟交换机上,同时也会将主机上的虚拟网卡 VMware Network Adapter8 连接到 VMnet8 虚拟交换机上,虚拟网卡只是作为主机与虚拟机通信的接口,虚拟机并不是依靠虚拟网卡



VMware Network Adapter VMnet8 来联网的,其网络结构如图 1-21 所示。

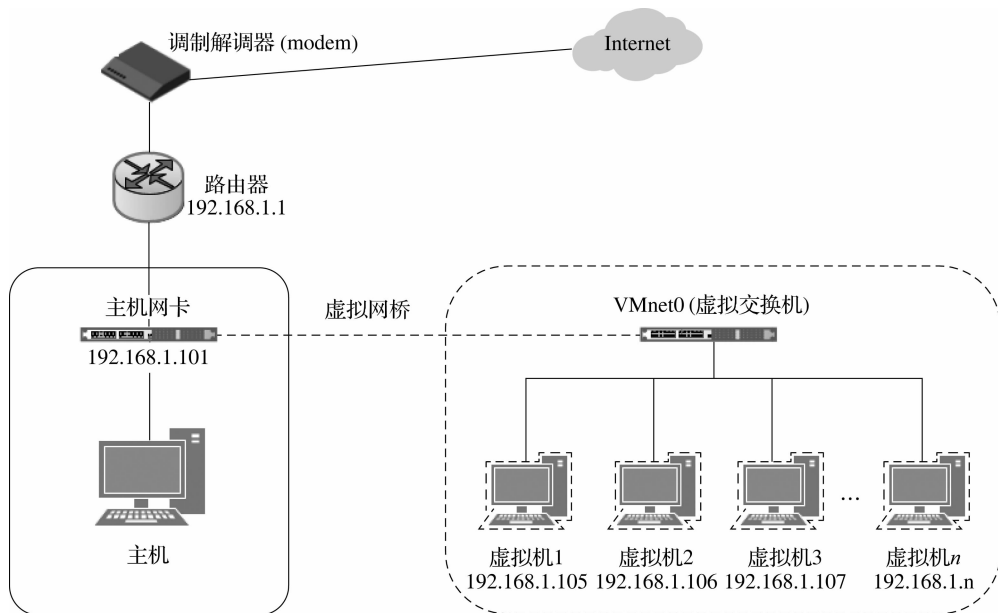


图 1-20 虚拟机桥接模式的网络结构

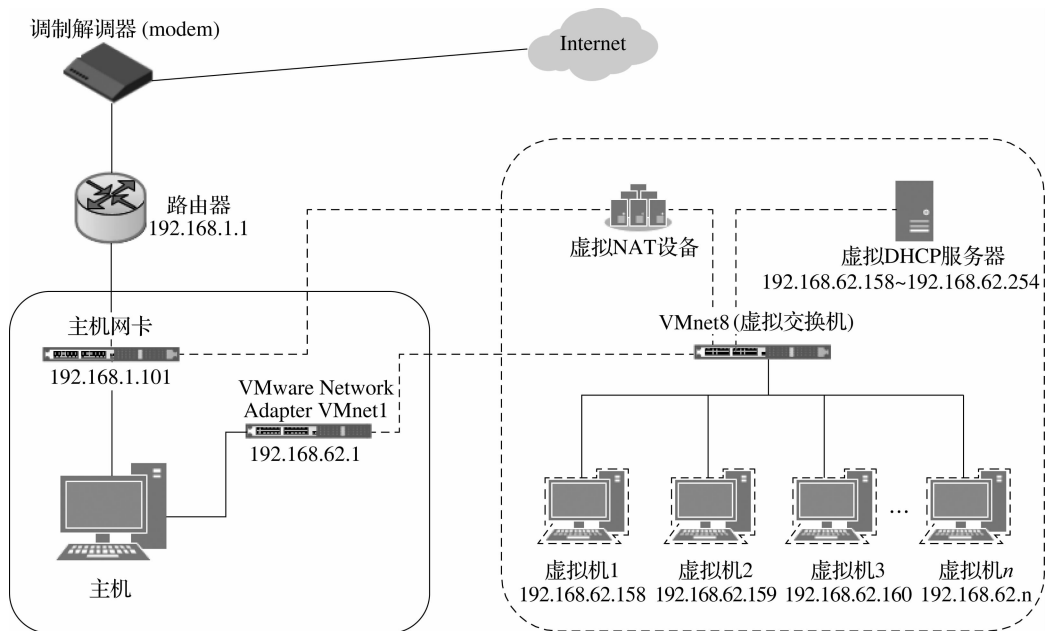


图 1-21 虚拟机 NAT 模式的网络结构

仅主机模式其实就是 NAT 模式去除了虚拟 NAT 设备,然后使用 VMware Network Adapter VMnet1 虚拟网卡连接 VMnet1 虚拟交换机来与虚拟机通信,仅主机模式将虚拟机与外网隔开,使得虚拟机成为一个独立的系统,只与主机相互通信,如果想要在仅主机模式下联网,可以将能联网的主机网卡与 VMware Network Adapter VMnet1 共享,这样就可以



实现虚拟机联网,其网络结构如图 1-22 所示。

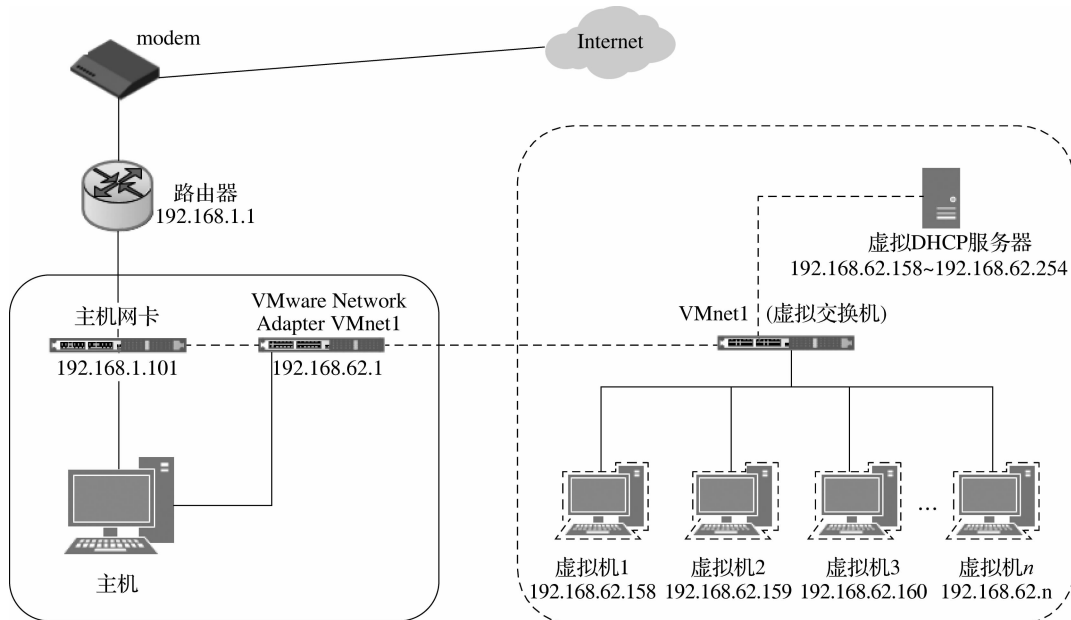


图 1-22 虚拟机仅主机模式的网络结构

根据实际情况做好配置,然后单击“关闭”按钮,返回到已准备好创建虚拟机界面后单击“完成”按钮,将进入虚拟机配置成功界面,如图 1-23 所示。

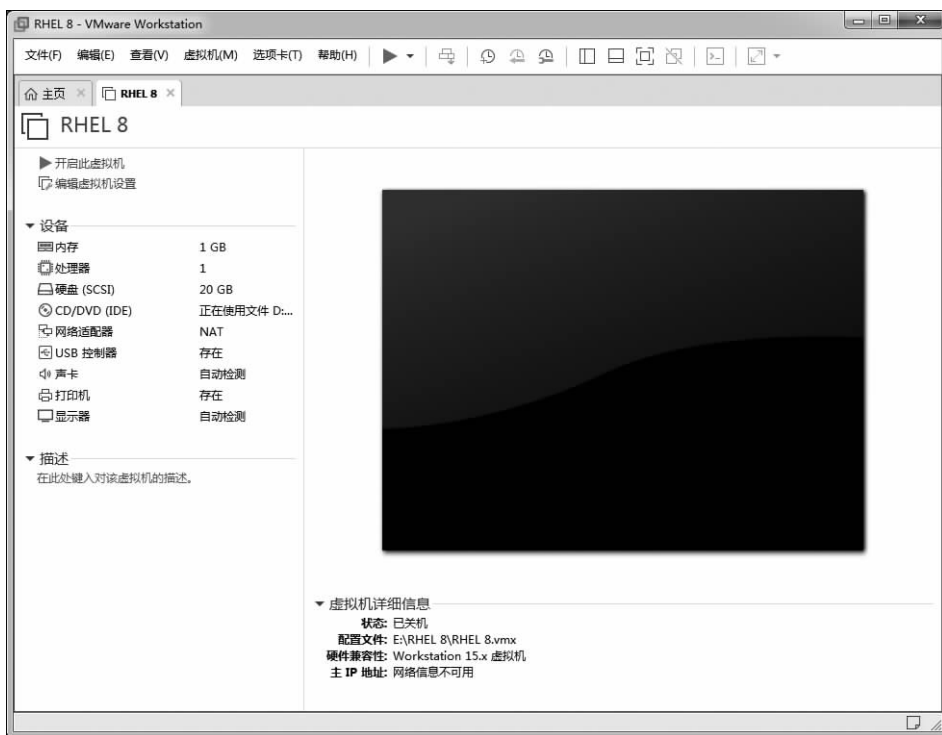


图 1-23 虚拟机配置成功界面



1.4 Linux 操作系统的安装与启动

1.4.1 Linux 系统的安装

虚拟机系统的前期配置顺利完成后,就可以在虚拟机管理器界面中单击“开启此虚拟机”按钮,稍后就能看到 RHEL 8 系统安装的界面,如图 1-24 所示。在界面中有 3 个选项,分别是安装 RHEL 8 系统、校验光盘并安装 RHEL 8 系统和系统救援模式。此时单击进入虚拟机系统中,用方向键选择第一个选项“Install Red Hat Enterprise Linux 8.0.0”来安装 Linux 系统。

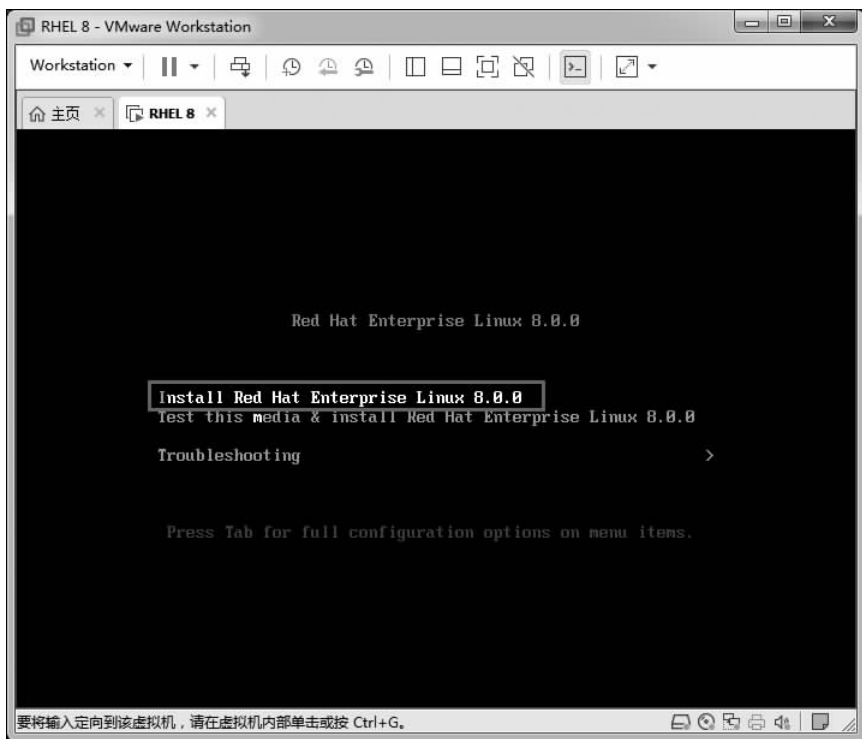


图 1-24 RHEL 8 系统安装界面

注:很多初学者单击进入虚拟机系统后,会发现鼠标指针不能移动,键盘被虚拟计算机独占了,此时只需要按系统默认设置的热键来取消独占键盘和鼠标状态即可,默认的设置是按 Ctrl+Alt 快捷键。快捷键可以根据使用习惯来设置,并且在第一次操作时都会有相应的提示。

接下来按 Enter 键加载系统镜像并进行安装,如图 1-25 所示。

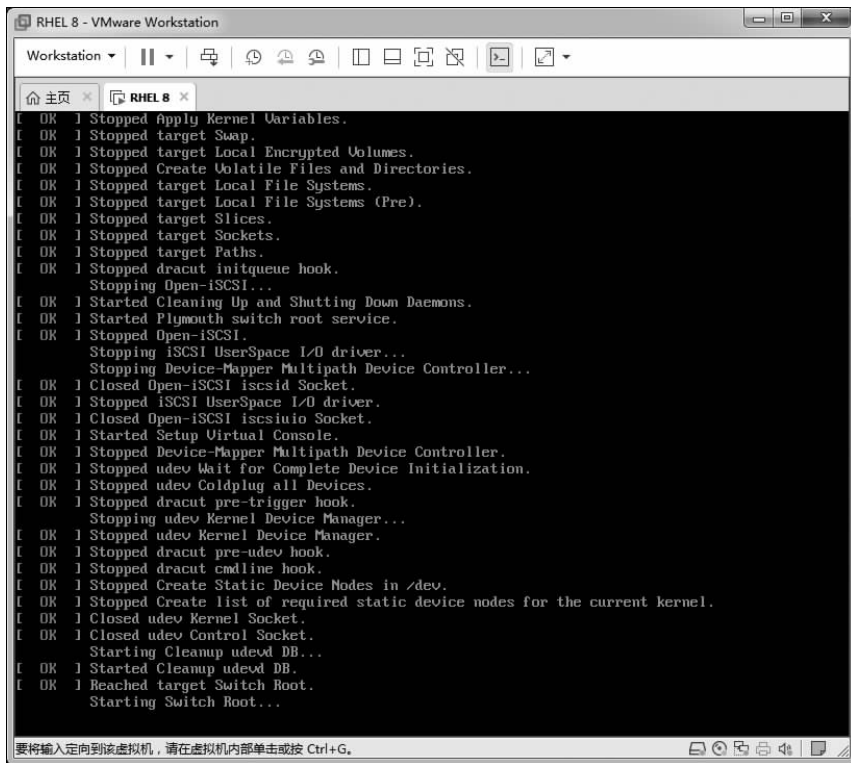


图 1-25 系统安装初始化

选择在安装过程中的语言信息,可以采取默认,也可以选择中文,然后单击“继续”按钮,如图 1-26 所示。

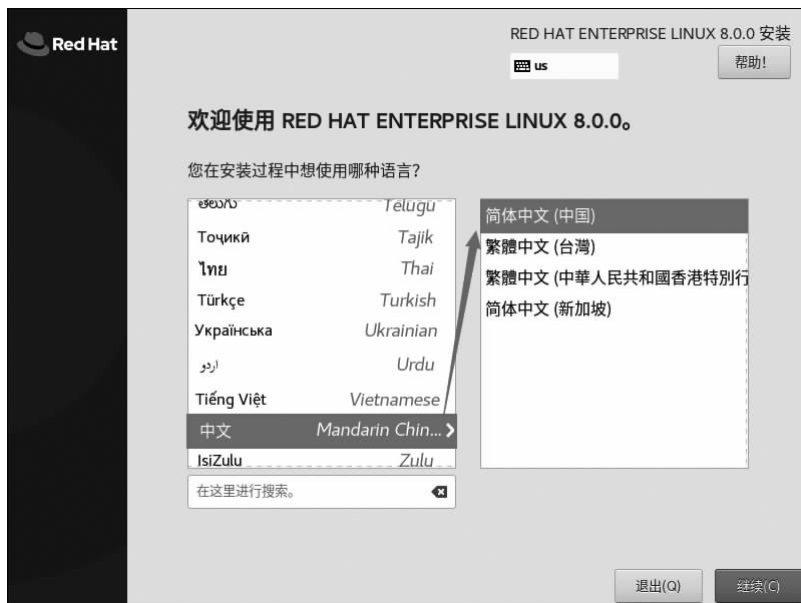


图 1-26 选择安装过程中的语言信息



在系统安装主界面中有三大选项,如图 1-27 所示,“本地化”选项分别是键盘、语言支持及时间和日期,时间和日期内容修改为“亚洲/上海时区”,如图 1-28 所示;“软件”选项分别是安装源和软件选择,采用默认设置即可;“系统”选项分别是安装目的地、KDUMP、网络和主机名、安全策略和系统目的。



图 1-27 系统安装主界面



图 1-28 系统时间和日期的设置



注:应尽量按照所要求的步骤做,否则在后期的排错中会花费大量的时间和精力,这里需要强调的是“系统”选项下的“安装目的地”选项,前期可以不做设置,后期会具体讲解,在图 1-29 所示界面中只需要单击左上角的“完成”按钮即可。“网络和主机名”选项可以按照图 1-30 来设置主机名“fishyoung.com”(可以自行修改),设置网络连接为“打开”状态,如果正确地安装了虚拟机软件,就能获取到 IP 地址(可能与图中的不同)。



图 1-29 “安装目标位置”界面



图 1-30 “网络和主机名”界面

在图 1-27 所示界面中单击“开始安装”按钮,将显示系统安装进度并进入重要的用户设置即“根密码”设置,如图 1-31 所示。

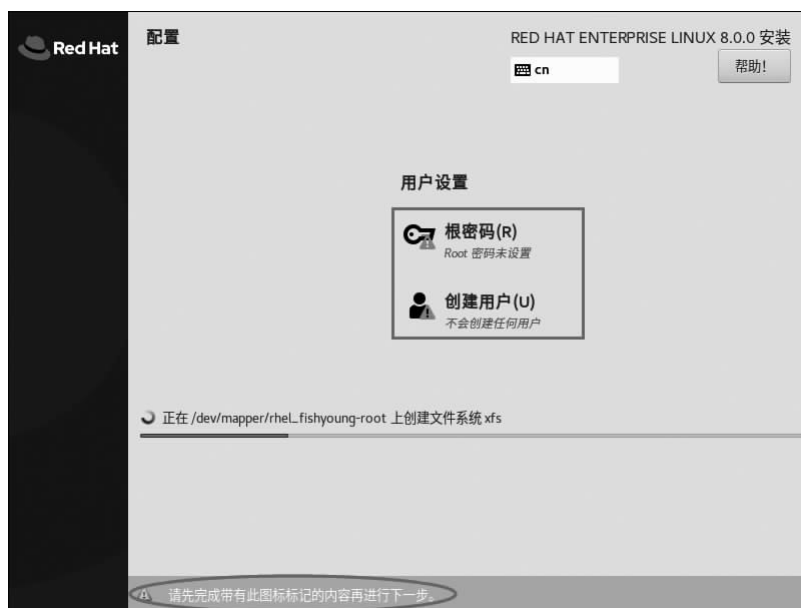


图 1-31 RHEL 8 系统安装配置界面

选择“根密码”选项,进入 ROOT 密码设置界面,如图 1-32 所示,做实验时可以输入弱密码,如 123456,在真正的应用环境中应该设置安全的密码,若采用的是弱密码,则需要单击两次左上角的“完成”按钮才能完成设置。



图 1-32 “ROOT 密码”界面

Linux 系统的安装进度视计算机的配置情况而定,安装完成后单击“重启”按钮,如图 1-33 所示。



图 1-33 系统安装完成界面

重启系统后将会看到系统初始化设置界面,如图 1-34 所示。



图 1-34 系统初始化设置界面

选择“License Information”选项,在打开的界面中选择“我同意许可协议”复选框,然后单击左上角的“完成”按钮,如图 1-35 所示。

返回图 1-34 所示的初始化设置界面后单击“结束配置”按钮,系统将会重启。重启后在出现的“隐私”和“输入”界面中,采用默认设置,单击“前进”按钮,如图 1-36 所示。

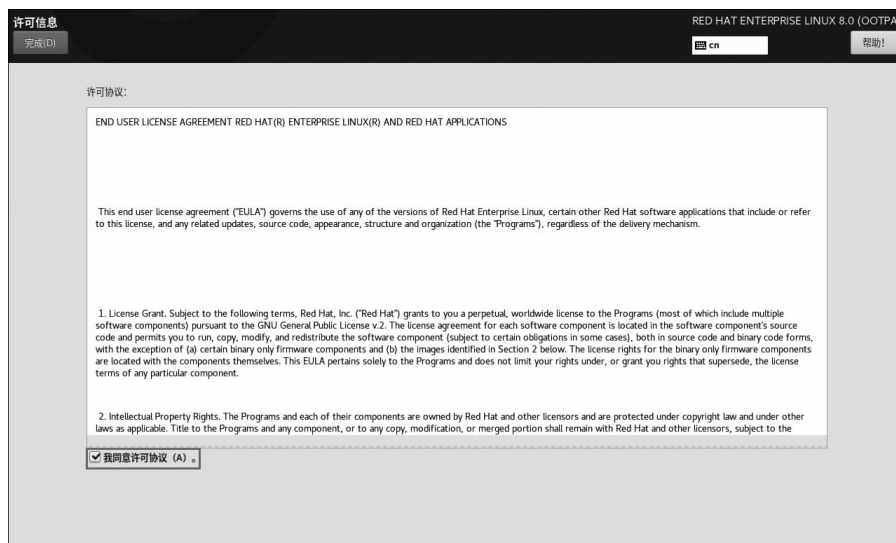


图 1-35 接受许可协议



图 1-36 默认设置界面

在“在线账号”界面中直接单击“跳过”按钮进入下一步,如图 1-37 所示。



图 1-37 “在线账号”界面



在“关于您”界面中,设置用户名为 fishyoung,然后单击“前进”按钮,如图 1-38 所示。



图 1-38 “关于您”界面

在“密码”界面中,设置密码为 123456,然后单击“前进”按钮,如图 1-39 所示。

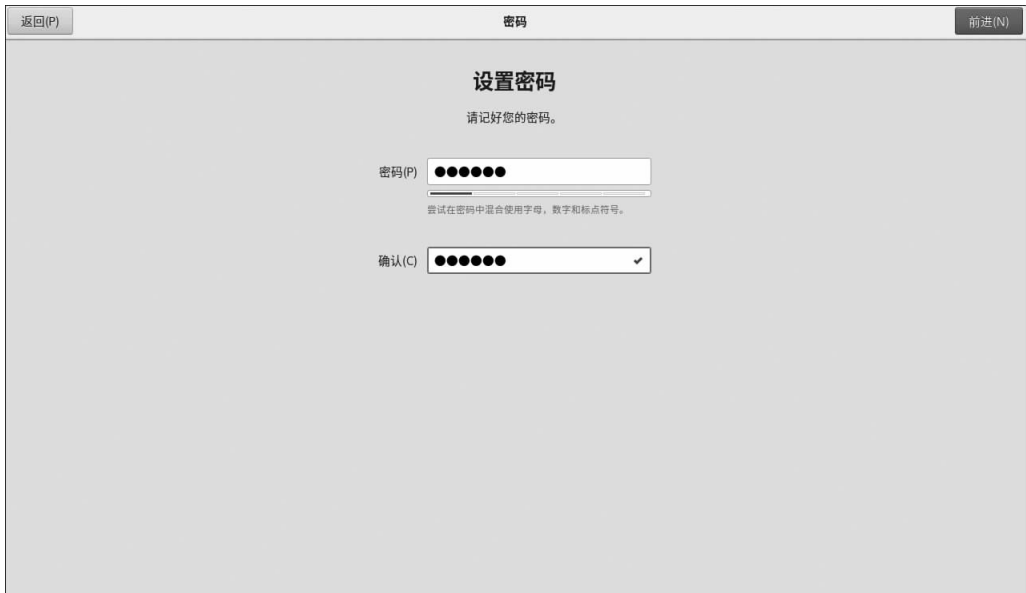


图 1-39 “密码”界面

在图 1-40 所示的界面中单击“开始使用 Red Hat Enterprise Linux(S)”按钮,系统将会重启,利用添加好的 root 或者 fishyoung 账户进行登录,如图 1-41 所示。



图 1-40 系统初始化完成界面

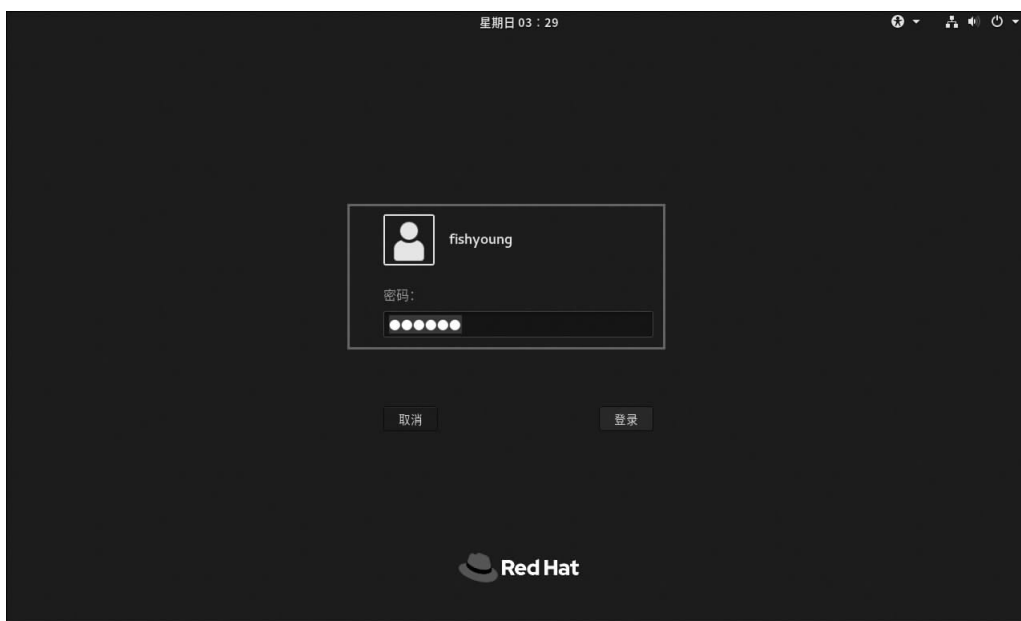


图 1-41 Linux 系统登录

登录成功后,将会出现图 1-42 所示的界面。完成 RHEL 8 系统的安装与设置,接下来就是 Linux 系统的学习。

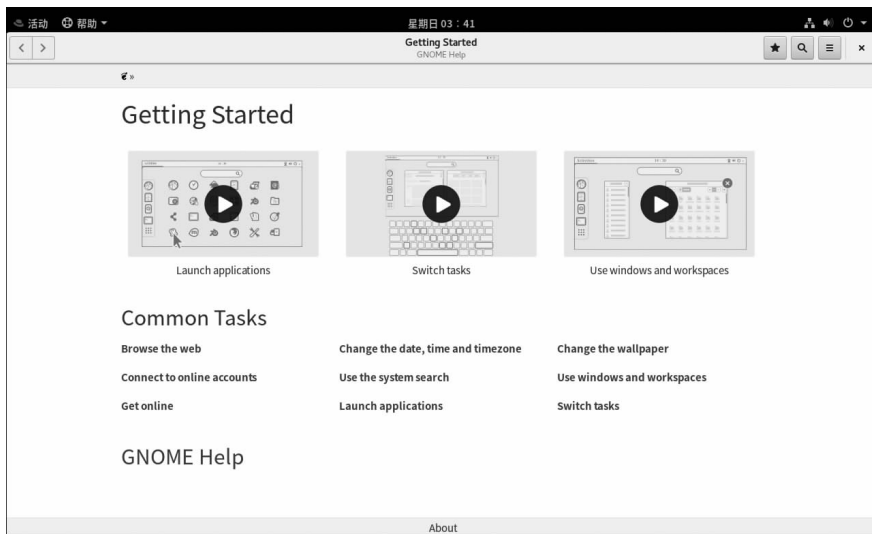


图 1-42 RHEL 8 系统安装完成界面

1.4.2 Linux 初始化进程之 systemd

Linux 操作系统的开机是自检 BIOS—Boot Loader 引导—加载系统内核—内核初始化—启动初始化进程这样一个过程,而早期 RHEL 5、RHEL 6 版本的 Linux 系统初始化 init 进程在 RHEL 7 系统及其以后版本中已经被替换成了 systemd,初始化进程是 Linux 操作系统中第一个启动的进程,它起着重要的作用,如为系统提供初始化工作,为用户提供初始化环境等。systemd 用目标(target)替代了运行级别的概念,提供了更大的灵活性。例如,systemd 目标可以继承一个已有的目标,并添加其他服务来创建自己的目标。

表 1-1 列举了 System V init 运行级别和 systemd 目标的对应关系。

表 1-1 System V init 运行级别和 systemd 目标的对应关系

System V init 运行级别	systemd 目标	备 注
0	runlevel0. target, poweroff. target	关闭系统
1, s, single	runlevel1. target, rescue. target	单用户模式
2	runlevel2. target, multi-user. target	用户定义/域特定运行级别。默认等同于 3
3	runlevel3. target, multi-user. target	多用户,非图形化。用户可以通过多个控制台或网络登录
4	runlevel4. target, multi-user. target	用户定义/域特定运行级别。默认等同于 3
5	runlevel5. target, graphical. target	多用户,图形化。通常为运行级别 3 的服务外加图形化登录
6	runlevel6. target, reboot. target	重启
emergency	emergency. target	紧急 Shell



从图 1-42 可知, Linux 系统安装完成后, 第一次进入的是图形化界面, 如果想设置开机默认进入“多用户, 非图形化”的文本模式, 可直接用命令 `ln` 把多用户模式目标文件连接到 `/etc/systemd/system/` 目录下并命名为 `default.target` (可以理解为创建一个快捷方式), 在 Linux 系统中单击左上角的“活动”图标, 再单击“Terminal”图标, 进入图 1-43 所示界面。



图 1-43 打开 Linux 系统终端界面

在 Linux 系统终端界面中执行“`ln -sf /lib/systemd/system/multi-user.target/etc/systemd/system/default.target`”命令, 要注意这条命令中的空格。得到图 1-44 所示的错误提示, 这主要是由权限不够导致的, 后续会进行讲解, 这里先按照命令输入。



图 1-44 权限不够错误提示

注:ln 是 link 的意思,注意第一个字母是字母 L 的小写。

为了解决权限不够的问题,我们需要切换到系统的管理员账号 root(后续学习中可以指定用户为系统管理员),然后执行上面的命令,如图 1-45 所示。



图 1-45 切换 root 账户再执行命令

注:在图 1-45 中的框线部分,需要着重强调的是,在用“su-root”命令切换用户时,下面需要输入该用户的密码,此时输入的密码是不可见的,只需要通过键盘输入正确的密码并按“Enter”键即可。关于 su 命令,将在模块 3 中进行详细讲解。

重启计算机进行测试,在 Linux 系统的文本模式登录界面输入用户名和密码并按“Enter”键即可进入系统,如图 1-46 所示。

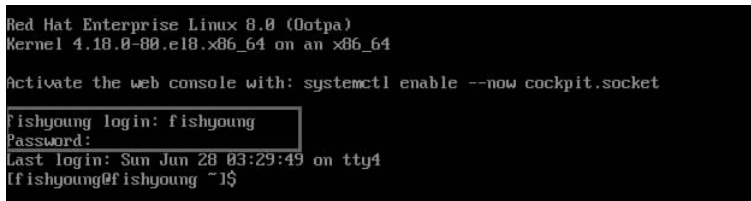


图 1-46 Linux 文本模式登录

上面介绍的是修改 Linux 系统的默认登录方式,即文本模式和图形模式,但是如果想在操作 Linux 系统中随时在多种模式之间进行切换又该如何操作呢?

systemd 的主要命令行工具是 systemctl,熟悉 RHEL 5 或 RHEL 6 的系统管理员应该非常熟悉系统服务和 init 系统的管理,如 service、chkconfig 及 telinit 命令的使用,当然 systemd 也可以完成同样的管理任务,只是命令工具 systemctl 的语法有所不同,systemctl 与传统的系统管理命令的对比见表 1-2 和表 1-3,其中,sn 仅代表服务名。

表 1-2 设置服务的 System V init 命令和 systemd 命令对照表

System V init 命令	systemd 命令	备注
service sn start	systemctl start sn.service	用来启动一个服务(不会重启现有的服务)



(续表)

System V init 命令	systemd 命令	备 注
service sn stop	systemctl stop sn.service	用来停止一个服务（不会重启现有的服务）
service sn restart	systemctl restart sn.service	用来停止并重新启动一个服务
service sn reload	systemctl reload sn.service	重新装载配置文件而不中断服务
service sn status	systemctl status sn.service	汇报服务是否正在运行

表 1-3 查看服务的 System V init 命令和 systemd 命令对照表

System V init 命令	systemd 命令	备 注
chkconfig sn on	systemctl enable sn.service	在下次启动时或满足触发条件时设置服务为启用
chkconfig sn off	systemctl disable sn.service	在下次启动时或满足触发条件时设置服务为禁用
chkconfig sn	systemctl is-enabled sn.service	用来检查一个服务在当前环境下是否开机自动启用
chkconfig --list	systemctl list-unit-files --type=service	输出在各个运行级别下服务的启用和禁用情况
telinit 3 init 3	systemctl isolate multi-user.target systemctl isolate runlevel3.target	改变至多用户运行级别

除了表 1-2 和表 1-3 所列出的常见用法,系统管理员还需要了解其他一些系统配置和管理任务的命令,首先了解 systemd 如何处理电源管理,命令见表 1-4。

表 1-4 systemd 电源管理命令

命 令	操 作
systemctl reboot	重启机器
systemctl poweroff	关机
systemctl suspend	待机

注:关机不是每个登录用户在任何情况下都可以执行的,一般只有系统管理员账户才可以关机。

通过上面的对照表命令,我们在 RHEL 8 系统中做一个练习,用 root 账户来操作,要求如下:

(1)查看系统防火墙服务(firewalld.service)开启状态。

```
systemctl status firewalld.service
```

(2)设置系统防火墙开机自启动。

```
systemctl enable firewalld.service
```

(3)停止系统防火墙服务。

```
systemctl stop firewalld.service
```



(4)再次查看系统防火墙开启状态。

```
systemctl status firewalld.service
```

(5)查看系统防火墙是否开机自启动。

```
systemctl is-enabled firewalld.service
```

(6)设置系统防火墙不开机自启动。

```
systemctl disable firewalld.service
```

(7)重启 RHEL 8 系统后查看系统防火墙服务是否开启。

```
systemctl reboot // 重启  
systemctl status firewalld.service // 登录后,再次查看
```

(8)将 RHEL 8 系统的文本模式切换到图形模式,前提是已安装图形化界面。

方法一:

```
systemctl isolate graphical.target
```

方法二:

```
startx
```

(9)将 RHEL 8 系统的图形模式切换成文本模式。

方法一:

```
systemctl isolate multi-user.target
```

方法二:可以用“Ctrl+Alt+F2”~“Ctrl+Alt+F6”快捷键切换成五个虚拟终端(文本模式)中的任意一个,用“Ctrl+Alt+F1”快捷键返回图形模式。

(10)设置系统开机默认进入指定目标(target)。

```
systemctl set-default multi-user.target // 多用户,非图形化目标  
systemctl set-default graphical.target // 多用户,图形化目标
```

1.5 Linux 操作系统的快照备份与恢复

在安装完 Linux 系统后,若系统崩溃了,则又要从头开始安装。此时我们要做的就是将刚才安装完的 Linux 系统做一个备份,以备不时之需,当出现系统故障时,可以快速地还原至出错前快照备份的环境状态,进而减少重装系统的时间。

在 Linux 系统所在虚拟机 VMware Workstation 窗体工具栏中选择“虚拟机”→“快照”→“拍摄快照”选项,如图 1-47 所示。将弹出图 1-48 所示的“RHEL 8-拍摄快照”对话框,需要填写快照“名称”和“描述”,最后单击“拍摄快照”按钮进行保存。

接下来将进行备份效果验证,如图 1-49 所示,在桌面上增加一个 Create Test 目录。

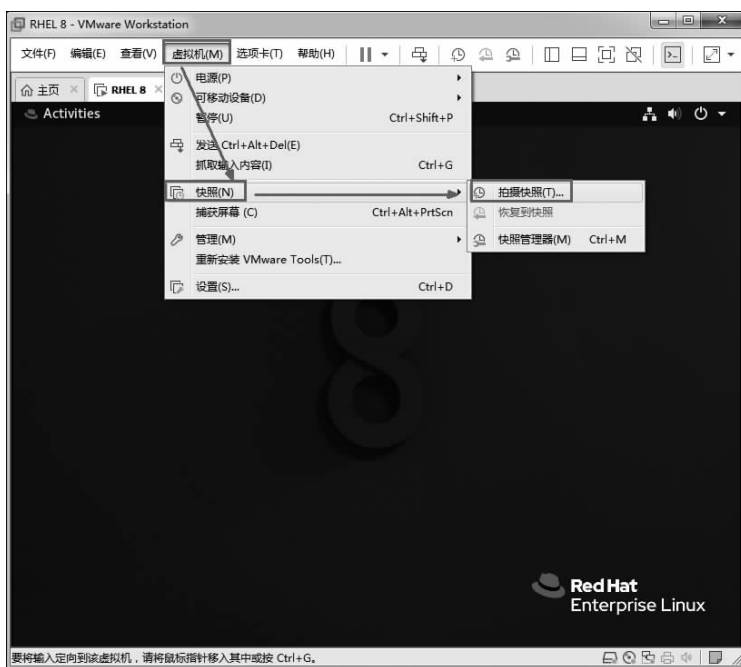


图 1-47 选择“虚拟机”→“快照”→“拍摄快照”选项

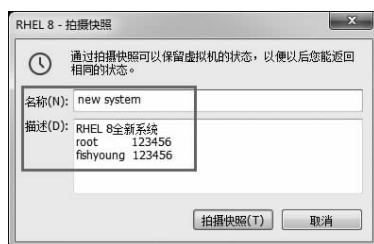


图 1-48 “RHEL 8-拍摄快照”对话框

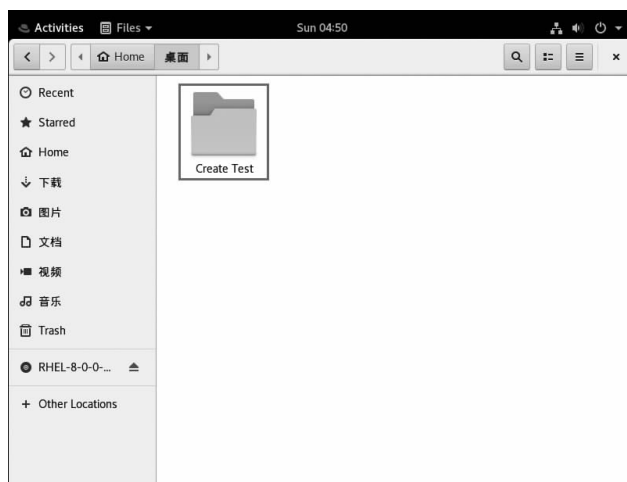


图 1-49 增加一个 Create Test 目录



关闭虚拟系统后,在虚拟机 VMware Workstation 窗体工具栏中的“虚拟机”→“快照”→“快照管理器”选项下有刚创建的新系统快照,如图 1-50 所示。

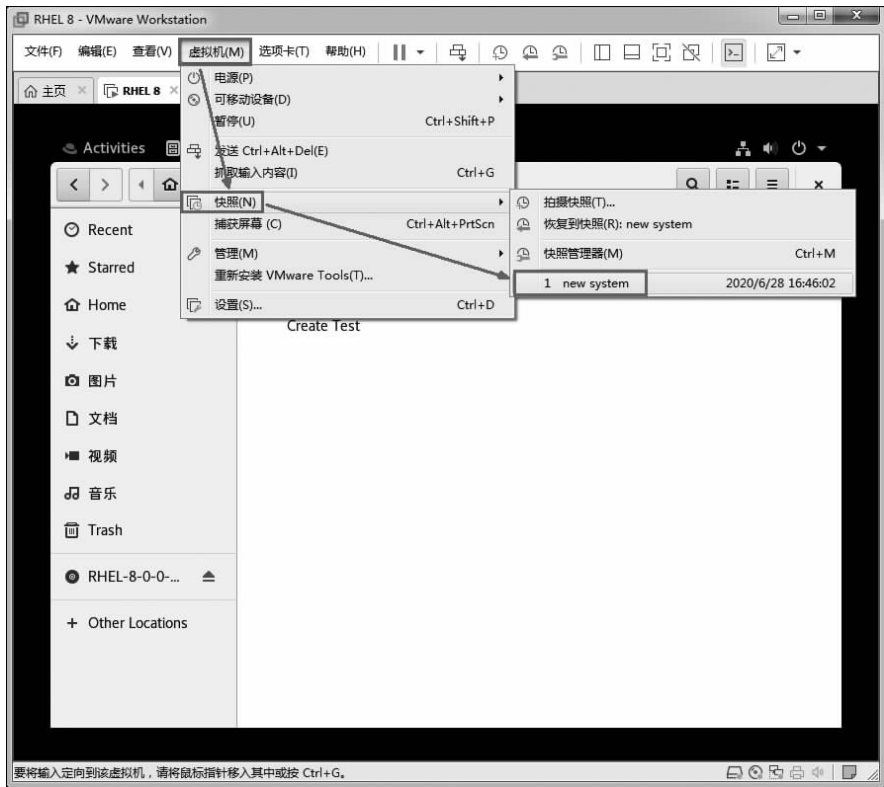


图 1-50 选择快照状态进行恢复

选择快照恢复虚拟计算机到所选快照时的状态,系统将进行几秒钟的还原,如图 1-51 所示。

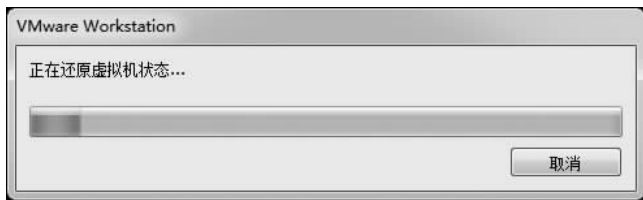


图 1-51 系统快照还原状态

还原后的界面如图 1-52 所示,可以发现之前在图 1-49 中创建的 Create Test 目录是不存在的,已经还原到最开始备份的 new system 干净系统了。

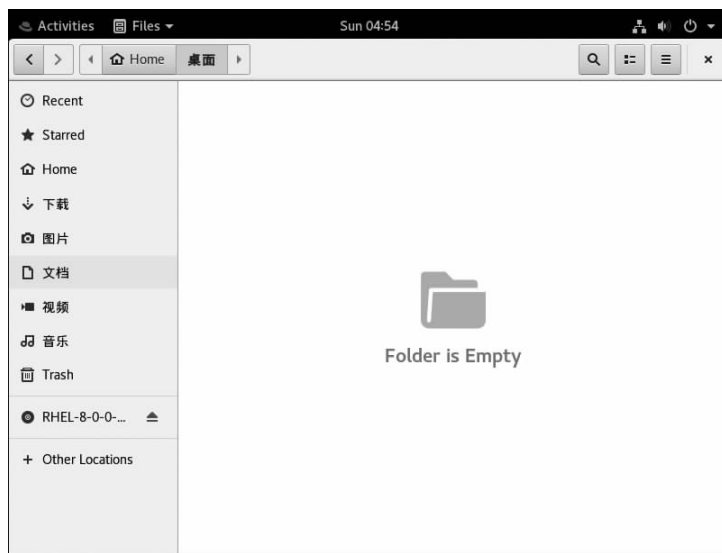


图 1-52 系统快照还原后的界面

1.6 实训 1 Linux 操作系统的安装启动和快照备份恢复

1. 实训目的

- (1)掌握虚拟机(VMware Workstation、VirtualBox 等)的安装及使用方法。
- (2)掌握在虚拟机中安装 Linux 操作系统的方法。
- (3)掌握 RHEL 8/CentOS 8 操作系统的安装与启动方法。
- (4)掌握在虚拟机中备份与恢复操作系统的方法。

2. 实训内容

- (1)将 Linux 系统安装在一个物理容量充足的非系统盘。
- (2)设置 Linux 系统磁盘为 20 GB。
- (3)Linux 系统内存推荐 1 GB,可以根据自己的物理设备进行调整。
- (4)虚拟机中的网络采用默认方式,暂不更改,仅启用网卡。
- (5)Linux 系统安装完成后,进行快照备份与恢复实验。

3. 实训要求

- (1)按题目要求写出相应的步骤和命令(“文字+截图”方式)。
- (2)总结实训心得与体会。

模块 2 Linux 基本命令

本模块首先介绍 Linux 命令的特点及 Linux 系统特有的文件系统目录结构,然后提出操作 Linux 系统一般采取远程连接的方式,推荐使用 Xshell 连接工具。通过 4 个基本命令很好地解释了 Linux 系统下的绝对路径和相对路径,这个知识点看似非常简单,实则易错。接着结合实际工作需要和学生的学习特点,将基本命令分为五大类(共 20 多个基础命令),它们与系统工作、系统状态、工作目录、文件、目录、打包压缩与搜索等主题相关,包含了 Linux 系统常用的基础操作,同时每个命令都配有相应的注释,为今后学习更复杂的命令和服务做好铺垫。

通过本模块的学习,读者将达到以下能力目标和要求:

- (1)熟悉 Linux 系统的操作环境;
- (2)掌握 Linux 远程连接工具的使用;
- (3)掌握 Linux 系统中的基本命令;
- (4)能够运用所学的基本命令进行组合操作。

2.1 远程连接

Linux 系统大多应用于服务器,而服务器不可能像 PC 一样放在办公室,而是放在 IDC 机房的,所以登录 Linux 系统一般都是通过远程登录的方式。Linux 系统是通过 SSH 服务实现远程登录功能的,当安装完系统时,这个服务已安装好且随机启动,默认 sshd 服务开启 22 端口,所以不需要额外配置就能直接远程登录 Linux 系统。sshd 服务的配置文件为 `/etc/ssh/sshd_config`,可以修改这个配置文件来实现想要的 sshd 服务,如更改启动端口为 22222。

如果是 Windows 操作系统,那么 Linux 远程登录需要在机器上额外安装一个终端软件。目前比较常见的终端登录软件有 Xshell、SecureCRT、PuTTY 等,不管使用哪一个客户端软件,最终的目的只有一个,就是远程登录 Linux 服务器。这些软件网上有很多免费版的,可以下载一个试用,这里选择使用 Xshell 软件,Xshell 软件的下载与安装比较简单,此处略过。

Xshell 软件安装完成后,为使远程连接后汉字不乱码,需要设置软件编码格式为 Unicode (UTF-8),如图 2-1 所示。

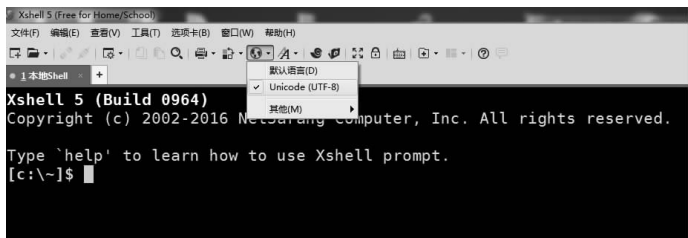


图 2-1 设置软件编码格式

要想远程连接 Linux 服务器,首先需要知道服务器的 IP 地址,如果是按照前面步骤一步一步安装的 Linux 系统,那么 Linux 系统会通过 DHCP 获取 IP 地址,在 Linux 系统中执行命令来获取 IP 地址。

```
[root@fishyoung~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:f2:a9:53 brd ff:ff:ff:ff:ff:ff
    inet 192.168.244.128/24 brd 192.168.244.255 scope global dynamic ens33
        valid_lft 1457sec preferred_lft 1457sec
    inet6 fe80::b220:409e:5e48:aald/64 scope link
        valid_lft forever preferred_lft forever
```

通过命令的执行得到 Linux 系统的 IP 地址为 192.168.244.128,在图 2-1 所示的界面中输入“ssh 192.168.244.128”并按 Enter 键登录,此时会打开图 2-2 所示的提示框。

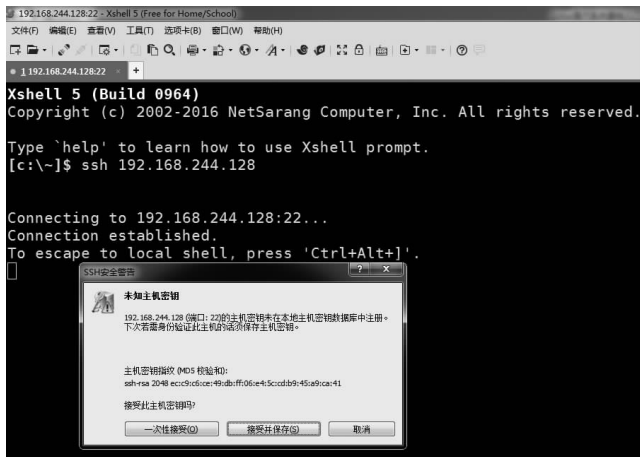


图 2-2 未知主机密钥提示



这里是主机密钥验证,只有在第一次连接时才会出现这个提示框,单击“接受并保存”按钮,弹出图 2-3 所示的对话框。



图 2-3 提示输入用户名

在文本框中输入用户名 root,单击“确定”按钮,弹出图 2-4 所示的对话框。



图 2-4 提示输入 root 用户密码

在“密码”文本框中输入 root 用户的密码,弹出登录成功后的 Linux Shell 命令行提示窗口。当进入系统后,我们会看到这样的字符:[root@fishyoung ~]#,其中#代表当前是管理员 root 用户登录,如图 2-5 所示;如果是\$,则表示当前账号为普通用户。

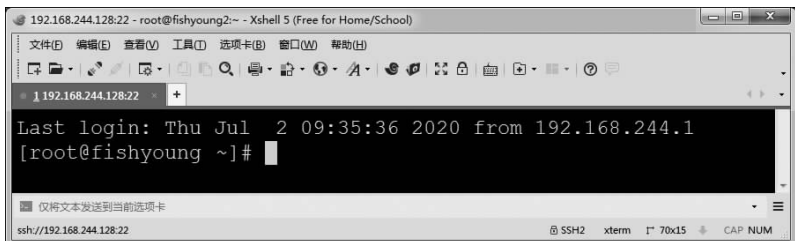


图 2-5 正确登录系统后的窗口及命令行

此时就可以像在 VMware 虚拟机里或物理机连接显示器那样管理 Linux 系统了,甚至感受不到是远程管理,因为都是文本操作,操作非常流畅,哪怕距离服务器有几千米。



2.2 了解 Linux 命令

2.2.1 Linux 命令的特点

(1) 在 Linux 系统中, 命令区分大小写, 文件执行情况与后缀名没有太大的关系, 主要看文件的属性, 即不像 Windows 系统通过后缀名判定文档。

(2) 在命令行中, 可以使用“Tab”键来自动补齐命令, 即可以只输入命令的前几个字母, 然后按“Tab”键, 系统将自动补齐命令, 若命令不止一个, 则显示出所有与输入字符相匹配的命令。例如, 在命令提示符后输入“hi”, 然后按“Tab”键, 系统将自动补全命令为“history”; 如果在命令提示符后只输入“h”, 然后按“Tab”键, 此时系统会警鸣一声, 再次按“Tab”键, 系统将显示所有以“h”开头的命令, 如图 2-6 所示。

```
[root@fishyoung ~]# h 此时须按2次 Tab 键
halt      hash      head      hexdump   hostid    hostnamectl
hardlink  hdsploder help      history    hostname  hwclock
[root@fishyoung ~]# history 输入“hi”, 按1次 Tab 键则自动补全为“history”
```

图 2-6 “Tab”键的使用效果

(3) 利用向上或向下方向键, 可以翻查曾经执行过的历史命令, 并且可以再次执行命令。

(4) 如果要在一个命令行上输入并执行多条命令, 可以使用分号来分隔命令。例如:

```
cd /etc; ls -l
```

(5) 如果屏幕上的内容较多, 可以按“Ctrl+L”快捷键清屏。

(6) 断开一个长命令行, 可使用反斜杠“\”, 以实现将一个较长的命令分成多行表达, 以增加命令的可读性。换行后, Shell 自动显示提示符“>”, 表示正在输入一个长命令, 此时可继续在新行上输入命令的后续部分。

(7) 一个文本控制台或仿真终端在同一时刻只能运行一个程序或命令, 在未执行结束前, 一般不能进行其他操作, 此时可将程序在后台运行, 以释放控制台或终端, 使其能进行其他操作。要使程序以后台方式运行, 只需在要执行的命令后加一个“&.”即可, 如“# cat &.”。

(8) 可以通过 jobs 命令查看序列号[num], 要让后台的程序到前台执行可以通过命令“fg [num]”实现; 要让前台执行的程序暂时调到后台, 可以按“Ctrl+Z”快捷键; 要结束后台的程序, 可以执行“kill %[num]”命令。

2.2.2 Linux 文件系统目录结构

Linux 文件系统采用带链接的树型目录结构, 即只有一个根目录(通常用“/”表示), 其中含有下级子目录或文件的信息, 子目录中又可含有其下级的子目录或文件的信息, 这样一层一层地延伸下去, 如图 2-7 所示。

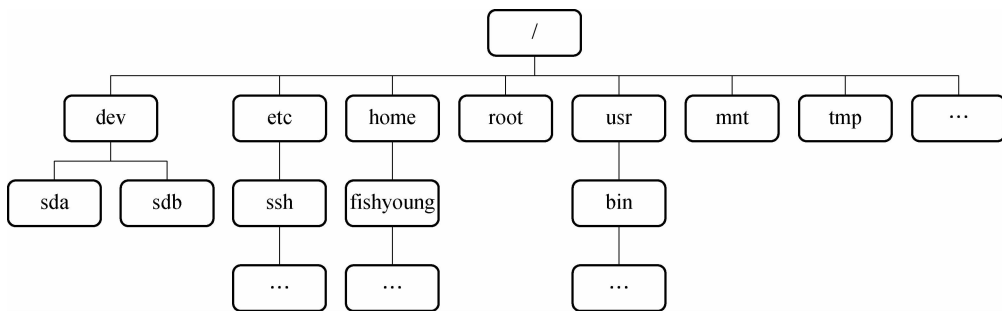


图 2-7 Linux 树型目录结构 1

在安装 Linux 时,系统会建立一些默认的目录,而每个目录都有其特殊的功能。同时 Linux 的设计思想是一切皆文件,把几乎所有资源统统抽象为文件形式,包括硬件设备、通信接口等。表 2-1 列举了 Linux 中部分默认目录功能。

表 2-1 Linux 中部分默认目录功能

目 录	说 明
/	Linux 系统的根目录
/bin	存放必要命令的目录
/dev	任何硬件与接口设备以文件的形式存放在这个目录下
/etc	存放系统配置文件的目录
/home	系统默认的普通用户家目录
/lib	存放必要运行库的目录
/mnt	各项设备的文件系统挂载点(mount)
/proc	存放存储进程和系统信息的目录
/root	系统管理员(root)的家目录
/sbin	存放系统管理程序的目录
/tmp	临时文件的存放位置,可供所有用户执行写入操作的特有权限
/usr	UNIX software resource 的缩写,是操作系统软件资源所默认放置的目录

2.3 绝对路径与相对路径

Windows 下的路径样式为“F:\rhel 7\”,Linux 下的路径可分为绝对路径与相对路径。

(1)绝对路径是从“/”开始的路径,如/usr/bin、/dev/sda。

(2)相对路径不是从“/”开始,而是从当前目录开始的路径,如 dev/sdb,.. /home/fishyoung。

2.3.1 pwd 命令

pwd 命令用于显示用户当前所处的工作目录,如果用户不清楚当前所处的目录,就可以



使用此命令。例如：

```
[root@fishyoung~]# pwd
/root
```

2.3.2 cd 命令

cd 命令用于切换工作路径,如果用户想切换到其他目录,就可以使用此命令。例如：

```
[root@fishyoung~]# pwd
/root
[root@fishyoung~]# cd /home/fishyoung //切换到 fishyoung 用户家目录
[root@fishyoung fishyoung]# cd //切换到 root 用户家目录
[root@fishyoung~]# cd - //返回到上一次的目录(fishyoung 家目录)
/home/fishyoung
[root@fishyoung~]#
```

2.3.3 ls 命令

ls 命令用于显示目录中的文件信息,该命令操作的语法格式为：

```
ls [参数选项] [文件或路径]
```

说明:ls 是 LS 的小写,“[]”里面的参数选项不是必需的,应根据实际情况搭配使用。例如：

```
[root@fishyoung~]# ls //查看当前目录下的文件及目录
anaconda-ks.cfg initial-setup-ks.cfg
[root@fishyoung~]# ls -a //查看当前目录下所有的文件及目录(含隐藏文件及目录)
. . . anaconda-ks.cfg .bash_history .bash_logout 省略部分.....
[root@fishyoung~]# ls -l //以长格式查看文件的属性、大小等详细信息
total 8
-rw-----.1 root root 1655 Mar 26 23:06 anaconda-ks.cfg
-rw-r--r--.1 root root 1731 Mar 26 23:09 initial-setup-ks.cfg
[root@fishyoung~]# ls -R //显示当前目录及子目录下的文件名
.:
anaconda-ks.cfg initial-setup-ks.cfg
[root@fishyoung~]# ls -ld //查看当前目录的属性信息
dr-xr-x---.5 root root 205 Mar 27 20:42.
[root@fishyoung~]#
```

说明:在 Linux 系统中,用“.”代表当前目录,用“..”代表当前目录的父目录。

2.3.4 echo 命令

echo 命令用于在屏幕上输出信息,该命令操作的语法格式为：

```
echo [参数选项] 字符串或变量
```

echo 将后面的字符串、变量等显示到标准输出,常用参数及作用如表 2-2 所示。

表 2-2 echo 命令参数及作用

参 数	作 用
-n	不换行输出内容
-e	解析转义字符
\n	换行
\r	回车
\t	制表符

例如:

```
[root@fishyoung~]# echo fishyoung; echo $ PATH           //输出字符串和 PATH 环境变量
fishyoung
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@fishyoung~]# echo -n fishyoung; echo $ PATH       //不换行输出
fishyoung/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@fishyoung~]# echo "hellwold\n"                  //输出字符串,并没有转义
hellwold\n
[root@fishyoung~]# echo -e "hellwold\n"               //输出字符串,并换行
hellwold
[root@fishyoung~]# echo -e "hellwold\r"               //输出字符串,并按 Enter 键
hellwold
[root@fishyoung~]# echo -e "hellwold\tlinuxstudy"     //输出字符串,并用制表符分隔
hellwold linuxstudy
[root@fishyoung~]#
```

试一试:在图 2-8 所示的 Linux 树型目录结构中,假设当前所处的目录是 ssh,完成下面两个要求:

(1)切换到 fishyoung 目录,分别采用绝对路径和相对路径的方法。

(2)列出 lvm 目录下所有文件(包括隐藏)的属性、大小等信息,分别采用绝对路径和相对路径的方法。

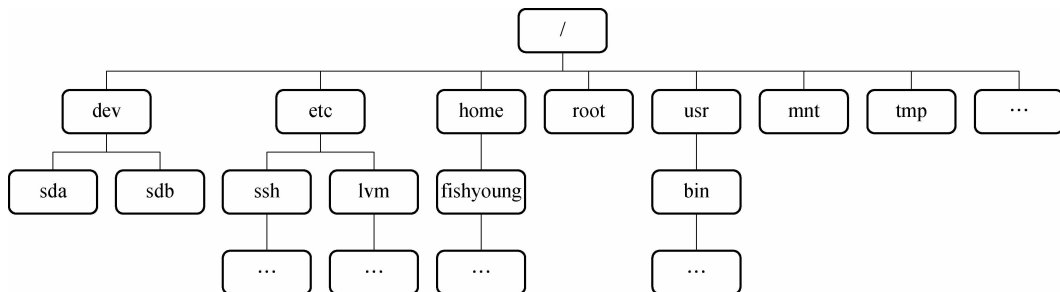


图 2-8 Linux 树型目录结构 2



2.4 基本命令

Shell 命令行和 Shell、Python 编程是日常 Linux 系统管理工作中必不可少的,为了后续的命令操作和脚本编程,学习一些常用与实用的系统工作命令是有必要的。Linux 系统是由很多目录文件构成的,下面将结合实际的工作环境来进行基本命令的学习,在学习过程中如果不记得命令的用法或参数信息,可以借助 man 命令来获得详细的帮助信息,如“man pwd”。

2.4.1 目录操作类命令

(1)mkdir 命令用来创建目录,该命令操作的语法格式为:

```
mkdir [参数选项] [目录名称]
```

例如:

```
[root@fishyoung~]# ls //查看当前目录下的文件及目录
anaconda-ks.cfg
[root@fishyoung~]# mkdir A //在当前目录下创建目录 A
[root@fishyoung~]# ls
A anaconda-ks.cfg //A 目录创建成功
[root@fishyoung~]# mkdir A/B //在 A 目录下再创建 B 目录
[root@fishyoung~]# ls A //查看 A 目录下的文件及目录
B //A 目录下的 B 目录创建成功
[root@fishyoung~]# mkdir AA/BB //在当前目录下创建 AA 目录,在 AA 目录下创建 BB 目录
mkdir: cannot create directory 'AA/BB': No such file or directory //错误
[root@fishyoung~]# mkdir -p AA/BB //用参数“-p”再次创建
[root@fishyoung~]# ls;ls AA //查看当前目录和 AA 目录下的文件及目录
A AA anaconda-ks.cfg //AA 目录创建成功
BB //AA 目录下的 BB 目录创建成功
[root@fishyoung~]#
```

当想要创建 AA/BB 目录时,系统提示不能创建,原因是 AA 目录不存在,在 Linux 系统中,如果要创建目录的上一级目录不存在就会报错。此时就需要加入“-p”参数选项,利用该选项可以创建一大串级联目录,还有一个好处就是当创建一个已经存在的目录时,不会报错。

(2)rmdir 命令用来删除空目录,该命令操作的语法格式为:

```
rmdir [参数选项] [目录名称]
```

例如:

```
[root@fishyoung~]# ls //查看当前目录下的文件及目录
A AA anaconda-ks.cfg
```



```

[root@fishyoung~]# ls A/B //查看并确认 B 目录下没有文件及目录
[root@fishyoung~]# rmdir A //删除 A 目录
rmdir: failed to remove 'A': Directory not empty //错误,不能删除
[root@fishyoung~]# rmdir A/B //删除 A 目录下的 B 目录
[root@fishyoung~]# ls A //删除成功,A 目录下没有 B 目录
[root@fishyoung~]# ls
A AA anaconda-ks.cfg
[root@fishyoung~]# rmdir A //删除 A 目录
[root@fishyoung~]# ls //删除成功,当前目录下没有 A 目录
AA anaconda-ks.cfg
[root@fishyoung~]#

```

rmkdir 命令只能删除空目录,当想要删除 A 目录时,提示不能删除,原因是 A 目录下有 B 目录,只有清空 A 目录下的所有文件及目录,才能删除 A 目录。即使加入“-p”参数选项也只能删除一大串级联空目录,这个命令有局限性,所以通常采用 rm 命令。

(3)rm 命令用来删除文件或目录,该命令操作的语法格式为:

```
rm [参数选项] [文件或目录名称]
```

例如:

```

[root@fishyoung~]# ls //查看当前目录下的文件及目录
AA anaconda-ks.cfg
[root@fishyoung~]# mkdir -p A/{B..D} //创建 A 目录及 A 目录下的 B、C、D 目录
[root@fishyoung~]# ls A //查看 A 目录下的文件及目录
B C D
[root@fishyoung~]# rm anaconda-ks.cfg //删除当前目录下的文件
rm: remove regular file 'anaconda-ks.cfg'? y //输入“y”并确认删除
[root@fishyoung~]# ls //查看当前目录下的文件及目录
A AA
[root@fishyoung~]# rm A/B //删除 B 目录
rm: cannot remove 'A/B': Is a directory //错误,不能删除
[root@fishyoung~]# rm -r A //加入参数“-r”删除 A 目录
rm: descend into directory 'A'? y //输入“y”并确认删除 A 目录下的文件及目录
rm: remove directory 'A/B'? y //输入“y”并确认删除 B 目录
rm: remove directory 'A/C'? y //输入“y”并确认删除 C 目录
rm: remove directory 'A/D'? y //输入“y”并确认删除 D 目录
rm: remove directory 'A'? y //输入“y”并确认删除 A 目录
[root@fishyoung~]# mkdir -p A/{B..D} //创建 A 目录及 A 目录下的 B、C、D 目录
[root@fishyoung~]# ls //查看当前目录下的文件及目录
A AA
[root@fishyoung~]# rm -rf A //加入参数“-rf”,删除 A 目录
[root@fishyoung~]# ls //查看当前目录下的文件及目录

```



AA

```
[root@fishyoung~]#
```

rm 命令只能删除文件,当想要删除目录时,提示不能删除,只有加入“-r”参数才能删除目录,同时会有很多的提示需要输入“y”或“n”,此时可以加入“-f”参数来强制删除。

2.4.2 文件操作类命令

(1) touch 命令用来创建空白文件,如果该文件存在,就表示修改当前文件时间,该命令操作的语法格式为:

```
touch [参数选项] [文件名]
```

例如:

```
[root@fishyoung~]# ls //查看当前目录下的文件及目录
AA
[root@fishyoung~]# touch 1.txt //创建空白文件 1.txt
[root@fishyoung~]# ll //以长格式查看目录下的文件及目录
total 0
-rw-r-r-. 1 root root 0 Mar 28 11:44 1.txt //注意修改时间 11:44
drwxr-xr-x. 4 root root 26 Mar 28 10:41 AA
[root@fishyoung~]# touch 1.txt //等一段时间,再次使用命令
[root@fishyoung~]# ll
total 0
-rw-r-r-. 1 root root 0 Mar 28 11:48 1.txt //修改时间为 11:48
drwxr-xr-x. 4 root root 26 Mar 28 10:41 AA
[root@fishyoung~]#
```

在创建空白文件方面,touch 命令相当简洁。例如,“touch 1.txt”命令可以创建一个名为 1.txt 的空白文件。对 touch 命令来讲,有难度的操作主要体现在设置文件内容的修改时间(mtime)、文件权限或属性的更改时间(ctime)和文件的读取时间(ctime)上,后续将通过更多的命令结合来讲解。

(2) cp 命令用来复制文件或目录,该命令操作的语法格式为:

```
cp [参数选项] 源文件 目标文件
```

在 Linux 系统中,复制操作具体又分为以下三种情况:

- ①如果目标文件是目录,就把源文件复制到该目录中。
- ②如果目标文件不存在,就执行正常的复制操作。
- ③如果目标文件是普通文件,就会询问是否覆盖。

例如:

```
[root@fishyoung~]# ls //查看当前目录下的文件及目录
1.txt AA
[root@fishyoung~]# cp 1.txt AA //复制 1.txt 到 AA 目录下
[root@fishyoung~]# ls AA //查看 AA 目录,确认 1.txt 已被复制
```



```

1.txt BB CC
[root@fishyoung~]# cp 1.txt 2.txt //复制 1.txt 为 2.txt
[root@fishyoung~]# ls //查看当前目录,确认 1.txt 复制为 2.txt
1.txt 2.txt AA
[root@fishyoung~]# cp 1.txt 2.txt //复制 1.txt 为 2.txt,弹出覆盖询问
cp: overwrite '2.txt'? y
[root@fishyoung~]# ls //查看当前目录
1.txt 2.txt AA
[root@fishyoung~]# cp AA TestDir //复制 AA 目录为 TestDir 目录,弹出错误
cp: omitting directory 'AA'
[root@fishyoung~]# cp -r AA TestDir //加参数“-r”复制 AA 目录
[root@fishyoung~]# ls;ls TestDir //查看当前目录和 TestDir 目录,确认成功
1.txt 2.txt AA TestDir
1.txt BB CC
[root@fishyoung~]#

```

在复制文件或目录时,如果要复制一个目录,必须加入“-r”参数进行递归复制,否则会出错;如果在复制时要保留原始文件或目录的原始属性,同时保留软链接属性,需要加入“-a”参数(类似-dpR 组合);如果在复制时要保留原始文件或目录的原始属性,需要加入“-p”参数。为演示“-p”参数,需要先切换到 fishyoung 账户创建一个普通文件,然后用 root 账户进行复制测试,具体命令如下:

```

[root@fishyoung~]# su - fishyoung //切换到 fishyoung 账户
[fishyoung@fishyoung~]$ pwd //查看当前所处路径
/home/fishyoung
[fishyoung@fishyoung~]$ touch test.txt //创建 test.txt 普通文件
[fishyoung@fishyoung~]$ exit //退回到 root 账户
logout
[root@fishyoung~]# cd /home/fishyoung //切换到 fishyoung 家目录
[root@fishyoung fishyoung]# ll //以长格式显示文件属性
total 0
-rw-rw-r--. 1 fishyoung fishyoung 0 Mar 28 13:33 test.txt
[root@fishyoung fishyoung]# cp test.txt root1.txt //复制一份为 root1.txt
[root@fishyoung fishyoung]# ll
total 0
-rw-r--r--. 1 root root 0 Mar 28 13:34 root1.txt
-rw-rw-r--. 1 fishyoung fishyoung 0 Mar 28 13:33 test.txt
[root@fishyoung fishyoung]# cp -p test.txt root2.txt //加“-p”参数再复制一份为 root2.txt
[root@fishyoung fishyoung]# ll
total 0-rw-r--r--. 1 root root 0 Mar 28 13:34 root1.txt
-rw-rw-r--. 1 fishyoung fishyoung 0 Mar 28 13:33 root2.txt
-rw-rw-r--. 1 fishyoung fishyoung 0 Mar 28 13:33 test.txt
[root@fishyoung~]#

```




(3)mv 命令用来重命名或者移动文件或目录,如果在同目录中对文件或目录进行 mv 操作,就相当于重命名,该命令操作的语法格式为:

```
mv [参数选项] 源文件 [目标文件|目标路径]
```

例如:

```
[root@fishyoung~]# ls;ls AA //查看当前目录和 AA 目录下的文件
1.txt 2.txt AA TestDir
1.txt BB CC
[root@fishyoung~]# mv 2.txt AA //将 2.txt 移动到 AA 目录
[root@fishyoung~]# ls;ls AA //再次查看当前目录和 AA 目录下的文件
1.txt AA TestDir //少了 2.txt 文件
1.txt 2.txt BB CC //多了 2.txt 文件
[root@fishyoung~]# mv TestDir Test //将 TestDir 移动为 Test,即重命名
[root@fishyoung~]# ls //再次查看当前目录
1.txt AA Test
[root@fishyoung~]#
```

(4)dd 命令用来按照指定大小和个数的数据块来复制文件或转换文件,该命令操作的语法格式为:

```
dd [参数选项]
```

例如:

```
[root@fishyoung~]# ls //查看当前目录下的文件
1.txt AA Test
[root@fishyoung~]# dd if=/dev/zero of=file.txt count=10 bs=10M //生成 100 MB 大小的 file.txt 文件
10+0 records in
10+0 records out
104857600 bytes (105 MB) copied,2.81269 s,37.3 MB/s
[root@fishyoung~]# ls -lh //以可视的长格式查看目录下的文件
total 100M
-rw-r--r--.1 root root 0 Mar 28 11:48 1.txt
drwxr-xr-x.4 root root 52 Mar 28 13:59 AA
-rw-r--r--.1 root root 100M Mar 28 15:16 file.txt
drwxr-xr-x.4 root root 39 Mar 28 13:21 Test
[root@fishyoung~]#
```

dd 命令可以根据需要生成一个指定大小的文件,具体的参数及作用如表 2-3 所示。

表 2-3 dd 命令参数及作用

参 数	作 用
if	输入文件名,默认为标准输入,即指定源文件<if=input file>
of	输出文件名,默认为标准输出,即指定目标文件<of=output file>



(续表)

参 数	作 用
count	仅读取指定的区块数
bs	设置读入/输出的块大小为 bs 个字节

执行“dd if=/dev/zero of=file.txt count=10 bs=10M”命令,即从/dev/zero 设备文件中取出 10 个大小为 10 MB 的数据块(100 MB),然后保存成 file.txt 文件。

(5)ln 命令用来为文件创建链接,链接类型分为硬链接(hard link)和符号链接(symbolic link)两种,默认的链接类型是硬链接。如果要创建符号链接,必须使用“-s”参数选项。该命令操作的语法格式为:

```
ln [参数选项] 源文件或目录 目标文件或目录
```

例如:

```
[root@fishyoung~]# ln 1.txt AA/3.txt //创建普通文件硬链接
[root@fishyoung~]# ln AA Test/ //创建目录硬链接报错
ln: 'AA': hard link not allowed for directory
[root@fishyoung~]# ll 1.txt AA/3.txt //查看对比原文件和硬链接文件
-rw-r--r--.2 root root 0 Mar 28 11:48 1.txt
-rw-r--r--.2 root root 0 Mar 28 11:48 AA/3.txt
[root@fishyoung~]# ln -s /root/1.txt AA/4.txt //创建普通文件符号链接
[root@fishyoung~]# ln -s /root/AA Test //创建目录符号链接
[root@fishyoung~]# ll 1.txt AA/4.txt //查看对比原文件和符号链接文件
-rw-r--r--.2 root root 0 Mar 28 11:48 1.txt
lrwxrwxrwx.1 root root 5 Mar 30 16:28 AA/4.txt-> /root/1.txt
[root@fishyoung~]# ll Test/ //查看目录符号链接文件
lrwxrwxrwx.1 root root 2 Mar 30 16:28 AA-> /root/AA
[root@fishyoung~]#
```

建立硬链接时,链接文件和被链接文件必须位于同一个文件系统中,并且不能建立指向目录的硬链接,而符号链接则不存在这个问题。

在硬链接的情况下,参数中的“目标”被链接至[链接名]。如果[链接名]是一个目录名,系统将在该目录之下建立一个或多个与“目标”同名的链接文件,链接文件和被链接文件的内容完全相同。如果[链接名]为一个文件,用户将被告知该文件已存在且不进行链接。如果指定了多个“目标”参数,那么最后一个参数必须为目录。

如果给 ln 命令加上“-s”选项,则建立符号链接。如果[链接名]已经存在但不是目录,将不做链接。[链接名]可以是任何一个文件名(可包含路径),也可以是一个目录,并且允许它与“目标”不在同一个文件系统中。如果[链接名]是一个已经存在的目录,系统将在该目录下建立一个或多个与“目标”同名的文件,此新建的文件实际上是指向原“目标”的符号链接文件。值得注意的是,在做符号链接时一定要使用绝对地址。符号链接类似于 Windows 操作系统下的快捷方式。



2.4.3 文件查看类命令

(1)cat 命令用来查看一个文件的内容并显示在屏幕上,一般用于查看内容较少的文件,该命令操作的语法格式为:

```
cat [参数选项] [文件名]
```

例如:

```
[root@fishyoung~]# cat 1.txt           //查看 1.txt 的内容
welcome
to
my
site
[root@fishyoung~]# cat -n 1.txt       //文件内容前显示行号
1  welcome
2  to
3  my
4  site
[root@fishyoung~]#
```

在用 cat 命令阅读文件时,若要在前面显示行号,则需要加上“-n”参数。

(2)more/less 命令也用来查看一个文件的内容,后面直接跟文件名,当文件内容太多,一屏放不下时,使用 more 命令可以解决这个问题。当看完一屏后按空格键继续看下一屏。但看完所有内容后就会退出,如果想提前退出,只需按“q”键即可。less 命令的作用与 more 相同,后面直接跟文件名,按空格键同样可以翻页,还可以用上下键进行上翻和下翻,该命令操作的语法格式为:

```
more/less [参数选项] [文件名]
```

例如:

```
[root@fishyoung~]# more /etc/inputrc
# do not bell on tab-completion
# set bell-style none
set meta-flag on
set input-meta on
set convert-meta off
set output-meta on
# Completed names which are symbolic links to
# directories have a slash appended.
set mark-symlinked-directories on
$ if mode=emacs
--More--(34%)
[root@fishyoung~]#
```

(3)head/tail 命令用来查看文件的前/后十行,如果加上“-n”参数,则显示文件前/后 n 行,该命令操作的语法格式为:



```
head/tail [参数选项] [文件名]
```

例如:

```
[root@fishyoung~]# head /etc/inputrc           //默认显示前 10 行
# do not bell on tab-completion
# set bell-style none
set meta-flag on
set input-meta on
set convert-meta off
set output-meta on
# Completed names which are symbolic links to
# directories have a slash appended.
[root@fishyoung~]# head -2 /etc/inputrc        //显示前 2 行
# do not bell on tab-completion
# set bell-style none
[root@fishyoung~]#
```

tail 命令与 head 命令一样,后面直接跟文件名,如果加上“-f”参数,则动态地显示文件最后 10 行,特别适用于查看实时更新的日志,如“tail -f /var/log/messages”。

(4)file 命令用来查看文件类型,在 Linux 系统中,由于文本、目录、设备等都统称为文件,又不能单凭后缀就知道具体的文件类型,这时就需要使用 file 命令来查看文件类型,该命令操作的语法格式为:

```
file [参数选项] [文件名]
```

例如:

```
[root@fishyoung~]# file 1.txt                 //查看 1.txt 类型为纯文本文件
1.txt: ASCII text
[root@fishyoung~]# file AA                   //查看 AA 类型为目录
AA: directory
[root@fishyoung~]#
```

2.4.4 文件压缩和归档类命令

(1)gzip/gunzip 命令用来压缩/解压缩文件,文件压缩类命令还有 bzip2、bunzip2 等,该命令操作的语法格式为:

```
gzip/gunzip [参数选项] 文件
```

例如:

```
[root@fishyoung~]# cp /etc/inputrc.         //复制文件到当前目录
[root@fishyoung~]# ls                       //查看当前目录
1.txt AA file.txt inputrc Test
[root@fishyoung~]# gzip inputrc            //对 inputrc 进行压缩
[root@fishyoung~]# ls                       //查看当前目录,生成压缩文件
1.txt AA file.txt inputrc.gz Test
```



```

[root@fishyoung~]# zcat inputrc.gz //查看压缩文件里面的内容
# do not bell on tab-completion
# set bell-style none
set meta-flag on
set input-meta on
.....省略部分信息.....
[root@fishyoung~]# gunzip inputrc.gz //解压压缩文件
[root@fishyoung~]# ls //查看当前目录
1.txt AA file.txt inputrc Test
[root@fishyoung~]# gzip AA //压缩目录 AA,报错
gzip: AA is a directory-ignored
[root@fishyoung~]# gzip -r AA //加“-r”参数压缩 AA 目录里的文件
[root@fishyoung~]# ls AA //查看 AA 目录
1.txt.gz 2.txt.gz BB CC
[root@fishyoung~]# gunzip AA //解压缩目录 AA,报错
gzip: AA is a directory-ignored
[root@fishyoung~]# gunzip -r AA //加“-r”参数解压缩 AA 目录里的文件
[root@fishyoung~]# ls AA //查看 AA 目录
1.txt 2.txt BB CC
[root@fishyoung~]#

```

gzip 程序用来压缩文件,gunzip 程序用来还原 gzip 压缩的文件,zcat 程序用来显示 gzip 压缩的文件。如果被压缩/解压缩的是目录,但没有“-r”参数选项,则 gzip/gunzip 会报告错误消息并忽略该目录。使用“-r”参数选项,gzip/gunzip 会递归地压缩/解压缩目录层次结构中的文件,同时压缩后会删除原始文件,压缩过的文件后缀为“.gz”。

(2)tar 命令用来归档/解归档文件,在 Linux 系统中一般结合压缩命令一起使用,后缀一般为“.tar.gz”,该命令操作的语法格式为:

```

tar [参数选项] 归档文件名 须归档的原文件或目录 //归档语法
tar [参数选项] 归档文件名 [-C 目标目录] //解归档语法

```

例如:

```

[root@fishyoung~]# ls //查看当前目录
1.txt AA file.txt inputrc Test
[root@fishyoung~]# tar-czvf AA.tar.gz AA //对 AA 目录进行压缩归档
AA/
AA/BB/
AA/CC/
AA/1.txt
AA/2.txt
[root@fishyoung~]# ls //存在 AA.tar.gz 归档文件
1.txt AA AA.tar.gz file.txt inputrc Test
[root@fishyoung~]# ls Test //查看 Test 目录
1.txt BB CC

```



```
[root@fishyoung~]# tar -xzvf AA.tar.gz -C Test //将归档文件解压到 Test 目录
[root@fishyoung~]# ls Test //查看 Test 目录
l.txt AA BB CC
[root@fishyoung~]#
```

tar 命令能够处理各种类型的归档文件,并且可接收许多带选项的归档过滤器,常用参数及作用如表 2-4 所示。

表 2-4 tar 命令常用参数及作用

参 数	作 用
-c	小写 c,创建“.tar”格式的包文件
-C	大写 C,解包时指定目标目录
-f	指定文档名
-r	追加文件至“.tar”格式的文件结尾
-t	列表查看包内的文件
-v	输出详细信息
-x	解开“.tar”格式的包文件
-z	调用 gzip 程序进行压缩或解压缩

特别要注意的是,在参数的选择中,c、r、x、t 参数仅能存在一个,因为不可能同时压缩与解压缩。同时,“-f”参数选项要留意,在 f 之后要立即接文档名,不要再加参数。例如,“tar -czfv tfile sfile”是错误的写法,“tar -czvf tfile sfile”才是正确的。

2.4.5 文件搜索和查找类命令

(1)grep 命令用来查找普通文件中包含指定字符串的行,该命令操作的语法格式为:

```
grep [参数选项] 查找字符串 普通文件名
```

例如:

```
[root@fishyoung~]# ls //查看当前目录
l.txt AA AA.tar.gz file.txt inputrc Test
[root@fishyoung~]# grep console inputrc //查找文件中包含 console 的行
# for linux console and RH/Debian xterm
# for freebsd console
[root@fishyoung~]# grep -c console inputrc //统计包含 console 的行数
2
[root@fishyoung~]# grep -n console inputrc //输出 console 所在的行号
15:# for linux console and RH/Debian xterm
39:# for freebsd console
[root@fishyoung~]# grep -vn console inputrc //输出不包含 console 的行
.....省略部分信息.....
13:$ if mode=emacs
14:
```



```

16: "\e[1~": beginning-of-line
17: "\e[4~": end-of-line
.....省略部分信息.....
37: "\eOF": end-of-line
38:
40: "\e[H": beginning-of-line
.....省略部分信息.....
[root@fishyoung~]# grep -n ^sole inputrc           // 查询以 sole 开头的行
[root@fishyoung~]# grep -n sole$ inputrc           // 查询以 sole 结尾的行
39: # for freebsd console
[root@fishyoung~]#

```

grep 命令是在文件中搜索满足条件的行,如果查找的字符串中带有空格,可用单引号或双引号引起来,同时可以结合一些参数,以便于快速查找定位。

(2) find 命令用来查找系统中的文件或目录,结合参数使用的功能非常强大,该命令操作的语法格式为:

```
find 路径 匹配表达式
```

例如:

```

[root@fishyoung~]# ls                               // 查看当前目录
1.txt AA AA.tar.gz file.txt inputrc Test
[root@fishyoung~]# touch test.txt                   // 创建普通文件 test.txt
[root@fishyoung~]# ls                               // 查看当前目录
1.txt AA AA.tar.gz file.txt inputrc Test test.txt
[root@fishyoung~]# find . -name "* est *"           // 查找命名包含 est 的文件
./Test
./test.txt
[root@fishyoung~]# find . -type d -name "* est *"   // 查找命名包含 est 的目录
./Test
[root@fishyoung~]# find . -type f -name "* est *"   // 查找命名包含 est 的普通文件
./test.txt
[root@fishyoung~]# find . -type f -name "* est *" -exec mv {} ./Test \;
// 查找命名包含 est 的普通文件,并将其移动到 Test 目录中
[root@fishyoung~]# ls;ls Test                       // 查看当前目录和 Test 目录
1.txt AA AA.tar.gz file.txt inputrc Test
1.txt AA BB CC test.txt
[root@fishyoung~]#

```

find 命令能够按照不同的文件特征进行特定搜索,同时可以搭配通配符“*”使用,常用匹配表达式及作用如表 2-5 所示。

表 2-5 find 命令常用匹配表达式及作用

匹配表达式	作用
-name filename	查找指定名称的文件



(续表)

匹配表达式	作用
-user username	查找指定用户名的文件
-group groupname	查找指定用户组的文件
-type b/c/d/f/l	查找指定类型的文件,如 b(块设备)、c(字符设备)、d(目录)、f(普通文件)、l(符号连接)
-perm mode	查找指定权限的文件
-exec command {} \;	对匹配条件的文件执行 command 命令

特别要注意的是,对匹配条件的文件执行 command 命令时,command 是基础命令,“{}”代表的是前面匹配条件找到的文件,“\;”是固定结尾格式写法。例如:

```
[root@fishyoung~]# find . -name "* est" -exec ls -l {} \;
total 0
-rw-r--r--.1 root root 0 Mar 28 13:21 1.txt
drwxr-xr-x.4 root root 52 Mar 31 10:57 AA
drwxr-xr-x.2 root root 6 Mar 28 13:21 BB
drwxr-xr-x.2 root root 6 Mar 28 13:21 CC
-rw-r--r--.1 root root 0 Apr 1 15:21 test.txt
```

上述命令的含义为查找当前目录中以 est 结尾的文件,并按照长格式输出显示。要注意的是,“{}”所处的位置应该符合命令的语法格式要求。例如:

```
[root@fishyoung~]# ls AA; find . -name "* est" -exec cp -r {} AA \;
1.txt 2.txt BB CC
[root@fishyoung~]# ls AA //查看 AA 目录
1.txt 2.txt BB CC Test
```

上述命令的含义为查找当前目录中以 est 结尾的文件,并复制到 AA 目录,因为是目录,所以加上“-r”参数。

find 命令除了“-exec”选项外,还可以搭配 xargs 命令使用,在使用 find 命令的“-exec”选项处理匹配到的文件时,find 命令将所有匹配到的文件一起传递给“-exec”执行。但有些系统对传递给“-exec”的命令长度是有限制的,因此在 find 命令运行几分钟后有可能出现溢出错误,错误信息通常是“参数列太长”或“参数列溢出”。而 xargs 命令每次只获取一部分文件而不是全部,这样它就可以先处理最先获取的那一部分文件,然后处理下一批,并如此继续下去。xargs 命令的具体用法通过实例来完成。例如,用 find 命令查找/data 目录下以“.txt”结尾的文件并将其复制到/tmp 下。

方法一:“-t”参数指定目标目录。

```
[root@fishyoung~]# mkdir Find //创建 Find 目录
[root@fishyoung~]# ls //查看当前目录
1.txt AA AA.tar.gz file.txt Find inputrc Test
[root@fishyoung~]# find . -type f -name "*.txt" | xargs cp -t Find
[root@fishyoung~]# ls;ls Find //查看当前目录和 Find 目录
```




```
1.txt AA AA.tar.gz file.txt Find inputrc Test
1.txt 2.txt file.txt test.txt
```

方法二：“-i”参数指定将找到的结果放到{}中。

```
[root@fishyoung~]# mkdir Find2 //创建 Find2 目录
[root@fishyoung~]# ls //查看当前目录
1.txt AA AA.tar.gz file.txt Find Find2 inputrc Test
[root@fishyoung~]# find . -type f -name "*.txt" | xargs -i cp {} Find2
[root@fishyoung~]# ls;ls Find2 //查看当前目录和 Find2 目录
1.txt AA AA.tar.gz file.txt Find Find2 inputrc Test
1.txt 2.txt file.txt test.txt
```

2.4.6 输入/输出重定向和管道命令符的使用

1. 输入/输出重定向

当执行一个指令时,该指令可能会由文件读入数据,经过处理之后,再将数据输出到屏幕上,这个过程中可能会有标准输入、标准错误输入、标准输出、标准错误输出。Linux 系统下使用标准输入 stdin(0,默认是键盘)和标准输出 stdout(1,默认是终端屏幕)来表示每个命令的输入和输出,还使用一个标准错误输出 stderr(2,默认是终端屏幕)来输出错误信息。输入/输出重定向符号及作用如表 2-6 所示。

表 2-6 输入/输出重定向用到的符号及作用

符 号	作 用
command < file	将 file 文件作为 command 命令的标准输入
command > file	将 command 命令的结果输出到 file 文件中,若有则覆盖,若没有则新建
command >> file	将 command 命令的结果输出到 file 文件中,若有则追加,若没有则新建
command 2> file	将 command 命令结果的错误信息输出到 file 文件中,若有则覆盖,若没有则新建
command &> file	将 command 命令结果的所有信息(正确和错误)输出到 file 文件中,若有则覆盖,若没有则新建

例如:

```
[root@fishyoung~]# ls //查看当前目录
1.txt AA AA.tar.gz file.txt Find Find2 inputrc Test
[root@fishyoung~]# ls A //查看 A 文件或目录,将错误信息输出到屏幕
ls: 无法访问 A: 没有那个文件或目录
[root@fishyoung~]# ls > 2.txt //查看并把信息重定向到 2.txt(覆盖)
[root@fishyoung~]# cat 2.txt //查看 2.txt 文档
1.txt
2.txt
AA
```



```

.....省略部分信息.....
[root@fishyoung~]# ls A 2> 2.txt
//查看A文件或目录,并把错误信息输出重定向到2.txt,且无信息输出到屏幕(覆盖)
[root@fishyoung~]# cat 2.txt //查看2.txt文档
ls: 无法访问A: 没有那个文件或目录
[root@fishyoung~]# ls &>> 2.txt //查看并把信息输出重定向到2.txt(追加)
[root@fishyoung~]# cat < 2.txt //把2.txt文档输入重定向到cat命令
ls: 无法访问A: 没有那个文件或目录
1.txt
2.txt
AA
.....省略部分信息.....
[root@fishyoung~]#

```

在输入/输出重定向的过程中,要注意“>”(覆盖)和“>>”(追加)的用法。

2. 管道命令符

管道命令符“|”的作用是将前一个命令的标准输出作为后一个命令的标准输入,格式为“命令 A | 命令 B | 命令 C”。前面讲解与 find 命令搭配使用的 xargs 命令时,已经使用过管道命令符“|”。

```
grep -c console inputrc //统计包含 console 的行数
```

之前的思路是直接使用“-c”参数来完成,这里采取“|”的方式来完成,思路如下:

(1)先找出 inputrc 文件中包含 console 的所有行数。

```
grep console inputrc
```

(2)统计行数。

```
wc -l
```

例如,使用管道命令符则为:

```

grep console inputrc | wc -l
[root@fishyoung~]# ls //查看当前目录
1.txt AA AA.tar.gz file.txt Find Find2 inputrc Test
[root@fishyoung~]# grep console inputrc | wc -l //结合管道命令符和wc命令
2
[root@fishyoung~]#

```

再如,使用非交互式方式更改 fishyoung 用户密码为123456。

```

[root@fishyoung~]# echo "123456" | passwd --stdin fishyoung
Changing password for user fishyoung.
passwd: all authentication tokens updated successfully.
[root@fishyoung~]#

```

echo 命令用于在终端输出字符串或变量的值,“-stdin”选项用于从标准输入管道读入新的值,passwd 命令用于更改密码。

注:非交互式方式就是在重置密码时不需要人工干预,特别适合在 Shell 编程中使用。



2.4.7 系统信息和进程管理类命令

(1) `uname` 命令用来查看系统内核与系统版本等信息,格式为:

```
uname [参数选项]
```

在使用 `uname` 命令时,一般会固定搭配“-a”参数来完整地查看当前系统的内核名称、主机名、内核发行版本、节点名、系统时间、硬件名称、硬件平台、处理器类型及操作系统名称等信息。

```
[root@fishyoung~]# uname -a
Linux fishyoung.com 3.10.0-693.el7.x86_64 #1 SMP Thu Jul 6 19:56:57 EDT 2017 x86_64 x86_64
x86_64 GNU/Linux
[root@fishyoung~]#
```

通过查看 `redhat-release` 文件来获取当前系统版本的详细信息,其命令及相应的结果为:

```
[root@fishyoung~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.4 (Maipo)
[root@fishyoung~]#
```

(2) `whereis` 命令用来寻找命令的可执行文件所在的位置,格式为:

```
whereis [参数选项] 命令名称
```

例如:

```
[root@fishyoung~]# whereis uname
uname: /usr/bin/uname /usr/share/man/man1/uname.1.gz
[root@fishyoung~]# whereis find
find: /usr/bin/find /usr/share/man/man1/find.1.gz
[root@fishyoung~]#
```

通过命令查出 `uname` 命令和 `find` 命令所在的绝对路径分别为 `/usr/bin/uname` 和 `/usr/bin/find`。

(3) `history` 命令用来显示历史执行过的命令,格式为:

```
history [参数选项]
```

在使用 `history` 命令时,如果使用“-c”参数,就会清空所有的命令历史记录。还可以使用“!编码数字”的方式来重复执行某一次的命令。

```
[root@fishyoung~]# history
.....省略部分信息.....
514 cat l.txt
515 ls
516 uname -a
517 cat /etc/redhat-release
```