第 3 章　构建极简版 Linux 发行版

本章学习目标：

- 掌握磁盘镜像文件的创建和分区方法。
- 掌握格式化分区和挂载分区的方法。
- 掌握将 GRUB 引导加载程序安装到虚拟硬盘的方法。
- 掌握 Linux 内核的配置、编译。
- 掌握制作 initrd 的方法。
- 理解 grub.cfg 的作用和语法。
- 掌握 VirtualBox 中运行 Linux 的方法。

3.1　创建磁盘映像文件

磁盘映像文件可以用来模拟硬盘。磁盘映像文件是一个包含磁盘内容的文件，包括分区表、文件系统和数据等信息，可以被视为一个完整的硬盘副本。执行如下命令可以在文件夹/root/minilinux 中创建一个大小为 128MB 的磁盘映像文件 minilinux_disk.img。

```
[root@ztg ~]# mkdir /root/minilinux
[root@ztg ~]# cd /root/minilinux/
[root@ztg minilinux]# dd if=/dev/zero of=minilinux_disk.img bs=1M count=128
```

提示：本章使用的所有命令都在 all-commands-4-minilinux.sh 文件中，该文件在本书配套资源中。

3.2　对磁盘分区

对磁盘进行分区可以将一个物理硬盘分成多个逻辑部分，并分别为每部分分配一个驱动器号。通常情况下，用户会将操作系统及其相关文件、程序文件和系统配置文件存储在一个单独的分区中，同时将用户数据存储在另一个分区，这样可以更好地组织和管理数据，并避免由于操作系统或程序文件故障而导致用户数据丢失的风险。通过磁盘分区可以更好地组织和管理磁盘空间，提高数据存储、查找和访问的效率，并且提高性能和安全性。另外，将关键数据存储在单独的分区中可以使数据备份更加容易和有效。

执行如下命令对虚拟硬盘 minilinux_disk.img 进行分区。

```
[root@ztg ~]# cd /root/minilinux/
[root@ztg minilinux]# fdisk minilinux_disk.img
命令(输入 m 获取帮助): n
分区类型
   p   主分区 (0个主分区，0个扩展分区，4空闲)
   e   扩展分区 (逻辑分区容器)
选择 (默认 p): p
```

```
分区号 (1-4, 默认  1): 1
第一个扇区 (2048-262143, 默认 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-262143, 默认 262143):

创建了一个新分区 1，类型为“Linux”，大小为 127 MiB。

命令(输入 m 获取帮助): w
分区表已调整。
正在同步磁盘。

[root@ztg minilinux]#
```

3.3 关联磁盘分区

losetup 命令用于将一个块设备文件(如硬盘分区或者磁盘映像文件)关联到系统中的一个 loopback 设备(环回设备)上，从而使得这个块设备文件可以以文件的形式访问。这个命令通常被用来在 Linux 系统上挂载磁盘镜像文件。使用 losetup 命令时，需要指定一个未被占用的 loopback 设备，并将其与目标块设备文件相关联。通过这种方式，就可以像访问普通文件一样访问该块设备文件。同时，也可以使用其他工具(如 mount 命令)将其挂载到文件系统的某个目录下进行读写操作。

执行如下命令将磁盘映像文件 minilinux_disk.img 关联到 loopback 设备/dev/loop0 上。此后，就可以像访问普通磁盘一样访问该磁盘映像文件。当不再需要访问时，可使用 losetup -d 命令取消关联。

```
[root@ztg ~]# cd /root/minilinux/
[root@ztg minilinux]# fdisk -l minilinux_disk.img
Disk minilinux_disk.img: 128 MiB, 134217728 字节, 262144 个扇区
单元: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x78388192

设备                   启动   起点   末尾   扇区     大小 Id 类型
minilinux_disk.img1        2048 262143 260096  127M 83 Linux

[root@ztg minilinux]# umount /dev/loop0
[root@ztg minilinux]# losetup -o 1048576 /dev/loop0 minilinux_disk.img
```

3.4 格式化分区和挂载分区

文件系统是用于管理硬盘上数据的一种机制。它定义了如何在硬盘上存储和组织数据，并提供访问这些数据的接口。不同的操作系统支持不同的文件系统类型，例如，Windows 系统通常使用 NTFS 文件系统，Linux 系统通常使用 ext4、btrfs 或 XFS 等文件系统。

格式化分区是指在硬盘上创建一个新的文件系统，以便该分区可以用于存储数据。格式化分区的主要作用是清除分区上的所有数据，并将文件系统结构写入分区中。这样做可以确保分区的可靠性和稳定性，并允许操作系统正确地读取和写入数据。格式化分区还可以使用户更有效地使用硬盘空间。当分区被格式化时，文件系统会对硬盘上的空间进行组

织并标记为可用或已使用的状态。这样，用户就可以知道哪些空间已经被使用，哪些空间还可以用来存储数据。需要注意的是，在格式化分区之前一定要备份分区中的重要数据。因为格式化分区会将其中的所有数据都删除，如果没有备份，这些数据将无法恢复。另外，需要考虑选择哪种文件系统类型来适配操作系统和需求。不同的文件系统类型具有不同的特点和限制，例如，某些文件系统类型可能更擅长处理大文件或小文件，或更适合高可用性、高速读写等场景。

在 Linux 中，挂载分区是将一个文件系统与操作系统的目录结构进行关联的过程。当挂载一个分区时，这个分区的内容就会成为 Linux 文件系统的一部分，存储在该分区的文件和文件夹就可以被访问。Linux 之所以需要挂载分区，是因为它采用了一种“一切皆文件”的哲学，将所有设备和资源都视为文件或文件夹。对于硬盘等存储设备来说，这些文件和文件夹是通过文件系统进行组织和管理的。因此，当要访问某个分区上的文件时，需要先将该分区挂载到 Linux 文件系统中，才能够访问其中的数据。在挂载分区时，需要指定挂载点，即将该分区挂载到 Linux 文件系统中的一个目录下。

需要注意的是，在挂载分区之前，必须先格式化该分区，并且该分区必须支持 Linux 所使用的文件系统类型。否则，该分区将无法挂载到 Linux 文件系统中。

下面第 1 行的命令格式化分区/dev/loop0。第 2 行的命令创建挂载点/mnt/minilinux，挂载点通常是一个空目录，用来承载该分区下的文件和文件夹。第 3 行的命令将分区/dev/loop0 挂载到/mnt/minilinux 目录下，然后就可以在该目录下创建文件和文件夹了。

```
1 [root@ztg minilinux]# mkfs.ext4 /dev/loop0
2 [root@ztg minilinux]# mkdir /mnt/minilinux
3 [root@ztg minilinux]# mount -t ext4 /dev/loop0 /mnt/minilinux
```

3.5 安装 GRUB

GRUB 是一款开源的引导加载程序，它被广泛用于大部分 Linux 操作系统的安装程序中。在安装 Linux 操作系统时，GRUB 通常会被安装到硬盘的 MBR（主引导记录）或 EFI 系统分区上，并且它可以管理和引导多个操作系统。命令 grub-install 是 GRUB 软件包中的一个工具，它的作用是将 GRUB 引导加载程序安装到计算机硬件上，例如，MBR 或 EFI 系统分区。命令 grub-install 是安装和配置 GRUB 引导加载程序的关键命令，它能够确保计算机在启动时可以正确引导所需的操作系统。

执行如下命令将 GRUB 引导加载程序安装到虚拟硬盘 minilinux_disk.img 的 MBR。

```
1 # cd /root/minilinux/
2 # grub-install --boot-directory=/mnt/minilinux/boot/ --target=i386-pc \
3               --modules=part_msdos minilinux_disk.img
```

3.6 下载、配置、编译 Linux 内核

Linux 内核是操作系统的核心，它提供了管理系统硬件资源和处理软件程序的基本功能。它是整个操作系统的最底层，通常负责操作系统的启动、设备驱动程序、进程管理、内存

管理和系统调用等重要任务。Linux 内核是操作系统的基石，它提供了操作系统所需的核心功能，使得各种应用程序和工具可以在上面运行。

1. 下载 Linux 内核

可以执行如下命令从 Linux 内核官方网站或清华大学开源软件镜像站下载所需版本的内核源代码，使用的内核版本号是 6.1.11。源代码文件 linux-6.1.11.tar.xz 被下载到/root/minilinux 文件夹中，然后使用 tar 命令将下载的源代码文件解压到/root/minilinux/linux-6.1.11 目录中。

```
1# cd /root/minilinux/
2# wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.1.11.tar.xz
3# wget https://mirror.tuna.tsinghua.edu.cn/kernel/v6.x/linux-6.1.11.tar.xz
4# tar -xvf linux-6.1.11.tar.xz
```

2. 配置、编译 Linux 内核

执行下面第 1 行的命令，进入源代码目录/root/minilinux/linux-6.1.11。第 2 行的命令 make x86_64_defconfig 的作用是生成一个 x86_64 架构的默认配置文件，该文件包含 Linux 内核在 x86_64 架构上的所有必要配置选项，以便能够顺利地进行编译和安装。在编译 Linux 内核时，可以使用这个默认配置文件来快速生成一个可用的内核镜像。当然，在实际应用中，可能会根据具体的需求对内核的功能进行定制和修改。如果需要自定义配置选项，可以执行第 3 行的命令 make menuconfig 进行修改。运行 make menuconfig 命令来打开内核配置界面，此时会出现一个文本界面，其中包含内核编译选项，参考下面第 5～7 行所示选项对内核进行配置。在内核配置界面中，可以使用箭头键和回车键来浏览和设置不同的选项，可以选择需要编译进内核的驱动程序、文件系统和功能等。如果不确定如何设置某个选项，可以按 F1 键查看帮助信息。当完成配置后，连续按两次 Esc 键返回上一级菜单，当菜单界面提示是否保存配置，输入 Y 即可保存并退出。

```
 1[root@ztg ~]# cd /root/minilinux/linux-6.1.11/
 2[root@ztg linux-6.1.11]# make x86_64_defconfig
 3[root@ztg linux-6.1.11]# make menuconfig
 4
 5Device Drivers  --->
 6    [*] Block devices  --->
 7        <*>   RAM block device support
 8
 9[root@ztg linux-6.1.11]# make -j4 bzImage
10[root@ztg linux-6.1.11]# cp arch/x86/boot/bzImage /mnt/minilinux/boot/
```

内核配置选项 RAM block device support 允许内核将一块内存区域模拟为一个块设备。这意味着，可以在内存中创建一个文件系统，并将其挂载为一个块设备。该功能通常用于测试或嵌入式设备开发。当启用此选项时，内核将创建一个 RAM 硬盘(ramdisk)，并将其视为一个块设备。需要注意的是，在某些情况下，使用 RAM 磁盘可能会导致系统性能下降或内存不足。因此，建议仅在需要时才启用此选项，并且要确保有足够的内存可用来支持 RAM 硬盘。

上面第 9 行的 make -j4 bzImage 命令的作用是编译 Linux 内核源代码并生成 bzImage 文件。编译内核时使用 4 个(-j4，通常设置为 CPU 核心数的两倍)并发任务进行编译，以加快编译速度。bzImage 文件是 Linux 内核经过压缩后的可引导映像文件，可以用于启动系

统，它通常被存储在 boot 目录中，作为启动加载程序（如 GRUB）的引导映像文件。第 10 行的命令将 bzImage 文件复制到/mnt/minilinux/boot 文件夹中。

在编译内核期间，如果出现了错误提示，可以按 Ctrl＋C 组合键来中止编译过程，并修复相关问题后重新开始编译。

3. 阅读 Linux 内核源代码

阅读 Linux 内核源代码需要一定的基础知识和经验，并且对 Linux 内核的基本概念和结构有一个初步的了解，包括进程、线程、调度、文件系统、网络等。此外，还需要掌握 Linux 内核的主要组成部分，例如，处理器架构相关的代码、驱动程序、文件系统和网络协议栈等。Linux 内核主要使用 C 语言编写，因此掌握 C 语言编程技巧和语法是必不可少的。需要熟悉指针、数组、结构体等基本概念和操作，以及对内存管理、指针运算等方面有深入的理解。此外，理解操作系统和计算机体系结构基本原理也有助于更好地理解内核。Linux 内核由多个模块和子系统组成，每个模块和子系统都有各自的任务和功能。由于 Linux 内核非常庞大和复杂，阅读代码的过程可能会比较漫长和艰难，因此需要耐心和毅力。建议先从简单的模块或子系统入手，例如，进程管理、文件系统等，逐步深入到复杂的模块。同时也要注意理解内核的数据结构和算法等核心概念。

在阅读源代码时，需要定位到与自己所学知识相关的关键代码路径，以便能够更加深入地研究代码实现细节。在定位到关键代码路径之后，可以逐步深入理解代码的实现细节，包括数据结构、算法、函数调用关系、各种配置选项等。需要注意的是，在阅读代码的过程中，应该注重理解其核心思想和实现逻辑，而不是过于关注细节上的问题。

Linux 内核有详细的文档，包括注释、手册和文档，可以帮助理解内核的各部分。在阅读代码时，将注释和文档作为重要的参考，并利用它们来明确代码的目的和实现方式。特别是在阅读开发者提交的 patch 时，注释和文档能够帮助读者更好地理解其他人的意图和讨论过程。

为了加深对内核的理解，可以尝试编译并运行内核，以验证自己对源代码的理解和推测是否正确。在调试时，可以使用一些工具，如 gdb、kgdb、kdb、systemtap 等，这些工具可以帮助我们进一步了解代码执行流程和调试信息。可以使用调试器跟踪内核运行时的状态。通过设置断点、打印变量的值等方式，可以更深入地了解内核的实现和执行流程。学习内核调优可以了解内核运行性能的瓶颈和调优方法。可以了解内核参数、内存管理、I/O 调度等方面的知识，在了解内核架构的基础上进行调优。

Linux 内核开发是一个开放的社区，有许多邮件列表和社区活动供开发者交流和讨论。可以参加这些活动来了解内核开发的最新动态和技术进展，以及从其他开发者那里获取帮助和反馈。同时也可以贡献自己的代码、提交 Bug 报告等方式来参与内核开发。内核开发者通常会在邮件列表中讨论最新的技术进展和内核更新。关注这些内容可以帮助了解内核的最新动态和热门话题。

3.7 制作 initrd

initrd（initial ram disk）是一个临时根文件系统，它被加载到内存中，并在 Linux 操作系统的启动过程中使用。在 Linux 系统启动时，内核需要找到并加载必要的驱动程序和文件

系统才能继续引导过程。但是，在这些驱动程序和文件系统被加载之前，内核需要访问硬件设备和其他资源。这就需要一个临时的根文件系统，该文件系统包含必要的驱动程序和其他文件，以便内核可以启动并继续执行引导过程。因此，initrd 的作用是为 Linux 内核提供一个包含必要的驱动程序和文件系统的临时根文件系统，以便内核可以完成初始化过程。一旦初始化完成，内核将卸载 initrd 并转向真正的根文件系统。

BusyBox 是一个精简的 Linux 工具集合，包括一些常见的 Linux 工具，如 ls、cp、mv、cat 等。它通常用于嵌入式设备中，因为它非常小巧且功能强大。构建临时根文件系统通常需要一个基本的 Linux 工具集，而 BusyBox 正好满足了这个需求。因此，可以使用 BusyBox 构建临时根文件系统。

下面第 1～5 行的命令在/root/minilinux 文件夹中创建磁盘映像文件 initrd.img，然后以环回设备的方式将 initrd.img 挂载到/mnt/rootfs。第 6～9 行的命令下载 BusyBox 源代码压缩包文件 busybox-1.36.1.tar.bz2，然后解压。接着运行 make menuconfig 命令来打开配置界面，此时会出现一个文本界面，其中包含编译选项，参考下面第 11～13 行所示选项进行配置。第 15 行的命令编译 BusyBox。第 16 行的命令将 BusyBox 安装到/mnt/rootfs 文件夹中。创建启动初始化脚本文件/mnt/rootfs/etc/init.d/rcS，rcS 文件内容如第 21～22 行所示。第 24 行的命令为 rcS 脚本文件添加可执行权限。第 25～27 行的命令创建设备文件。第 28 行的命令卸载 initrd.img，此时生成的 initrd.img 文件就是临时根文件系统。第 29 行的命令将 initrd.img 文件复制到文件夹/mnt/minilinux/boot 中，此时该文件夹中包含 3 个文件：bzImage、grub、initrd.img。

```
 1 [root@ztg ~]# cd /root/minilinux
 2 [root@ztg minilinux]# dd if=/dev/zero of=initrd.img bs=1M count=4
 3 [root@ztg minilinux]# mkfs.ext4 initrd.img
 4 [root@ztg minilinux]# mkdir /mnt/rootfs
 5 [root@ztg minilinux]# mount -o loop initrd.img /mnt/rootfs
 6 [root@ztg minilinux]# wget https://busybox.net/downloads/busybox-1.36.1.tar.bz2
 7 [root@ztg minilinux]# tar xjvf busybox-1.36.1.tar.bz2
 8 [root@ztg minilinux]# cd busybox-1.36.1
 9 [root@ztg busybox-1.36.1]# make menuconfig
10
11 Settings  --->
12     --- Build Options
13     [*] Build static binary (no shared libs)
14
15 [root@ztg busybox-1.36.1]# make -j8
16 [root@ztg busybox-1.36.1]# make CONFIG_PREFIX=/mnt/rootfs install
17 [root@ztg busybox-1.36.1]# mkdir -p /mnt/rootfs/etc/init.d/
18 [root@ztg busybox-1.36.1]# gedit /mnt/rootfs/etc/init.d/rcS
19 [root@ztg busybox-1.36.1]# cat /mnt/rootfs/etc/init.d/rcS
20
21 #!/bin/busybox sh
22 echo "Hello miniLinux!"
23
24 [root@ztg busybox-1.36.1]# chmod +x /mnt/rootfs/etc/init.d/rcS
25 [root@ztg busybox-1.36.1]# mkdir /mnt/rootfs/dev
26 [root@ztg busybox-1.36.1]# mknod /mnt/rootfs/dev/console c 5 1
27 [root@ztg busybox-1.36.1]# mknod /mnt/rootfs/dev/ram b 1 0
28 [root@ztg busybox-1.36.1]# umount /mnt/rootfs
29 [root@ztg busybox-1.36.1]# cp /root/minilinux/initrd.img /mnt/minilinux/boot/
30 [root@ztg busybox-1.36.1]# ll -h /mnt/minilinux/boot/
31 -rw-r--r-- 1 root root  11M  6月  3 10:42 bzImage
32 drwxr-xr-x 5 root root 4.0K  6月  3 15:02 grub
33 -rw-r--r-- 1 root root 4.0M  6月  3 14:39 initrd.img
```

3.8 编写 grub.cfg

需要在引导程序配置文件/mnt/minilinux/boot/grub/grub.cfg 中添加 Linux 内核(bzImage)和根文件系统(initrd.img)的信息,以及其他必要的引导参数,确保启动过程可以正确地加载内核和根文件系统。执行如下命令创建配置文件 grub.cfg,其内容如第 3～7 行所示。第 4 行是一个 GRUB 引导程序的引导命令,它告诉计算机引导程序去加载位于硬盘 hd0 的第一个分区(msdos1)中的/boot/bzImage 文件作为 Linux 内核,并将根文件系统挂载在一个内存设备上并以读写模式(rw)挂载。接着,它将运行 init 程序,并且指定了它的路径为/linuxrc。第 5 行为注释行。第 6 行是用来指定引导时加载的 initrd 镜像文件的位置。它告诉系统在硬盘 hd0 的第一个分区(msdos1)中找 initrd.img 文件,然后将该文件加载到内存中,以便在启动 Linux 操作系统时使用。

```
1 [root@ztg ~]# cd /mnt/minilinux/boot/grub
2 [root@ztg grub]# cat > grub.cfg << EOF
3 menuentry "minilinux" {
4     linux (hd0,msdos1)/boot/bzImage root=/dev/ram rw init=/linuxrc
5     #linux (hd0,msdos1)/boot/bzImage console=tty0
6     initrd (hd0,msdos1)/boot/initrd.img
7 }
8 EOF
9 [root@ztg grub]#
```

3.9 VirtualBox 中运行 Linux

当完成上述步骤后,就可以在 VirtualBox 中安装并测试这款极简版 Linux 发行版了。

下面第 2 行的命令的作用是将 minilinux_disk.img 文件转换为原始磁盘映像格式,并将输出保存到 minilinux_disk.raw 文件中。qemu-img 是 QEMU 虚拟机的一个工具,用于创建、转换和管理虚拟磁盘镜像。在这条命令中,convert 参数表示要执行转换操作,-O raw 参数指定目标格式为原始磁盘映像格式,minilinux_disk.img 是要转换的源文件名,而 minilinux_disk.raw 是输出文件名。第 3 行的命令是使用 VirtualBox 中的 VBoxManage 命令工具将一个基于原始磁盘(raw disk)的磁盘映像文件 minilinux_disk.raw 转换为 VirtualBox 可以使用的 VDI(VirtualBox Disk Image)格式的文件。这样,在 VirtualBox 中就可以直接使用这个 minilinux_disk.vdi 文件作为虚拟机的硬盘,而无须重新创建虚拟硬盘或导入其他格式的映像文件。

```
 1 [root@ztg ~]# cd /root/minilinux/
 2 [root@ztg minilinux]# qemu-img convert minilinux_disk.img -O raw minilinux_disk.raw
 3 [root@ztg minilinux]# VBoxManage convertdd minilinux_disk.raw minilinux_disk.vdi
 4 [root@ztg minilinux]# ll -h
 5 drwxr-xr-x 36 root root 4.0K  6月  3 11:13 busybox-1.36.1
 6 -rw-r--r--  1 root root 2.5M  5月 19 06:37 busybox-1.36.1.tar.bz2
 7 -rw-r--r--  1 root root 4.0M  6月  3 14:38 initrd.img
 8 drwxrwxr-x 26 root root 4.0K  6月  3 10:40 linux-6.1.11
 9 -rw-r--r--  1 root root 129M  2月  9 18:36 linux-6.1.11.tar.xz
10 -rw-r--r--  1 root root 128M  6月  3 15:04 minilinux_disk.img
11 -rw-r--r--  1 root root 128M  6月  3 15:22 minilinux_disk.raw
12 -rw-------  1 root root  36M  6月  3 15:22 minilinux_disk.vdi
```

打开 VirtualBox 并创建一个新的虚拟机，如图 3.1 所示，在“新建虚拟电脑”对话框中，输入虚拟计算机的名称，并选择操作系统类型和版本。根据需要为虚拟机分配内存空间。在“虚拟硬盘”页面上，选择“使用已有的虚拟硬盘文件”选项，并单击“选择虚拟硬盘文件”按钮。在打开的对话框中，找到 minilinux_disk.vdi 文件所在的位置，并选择它。最后单击“创建”按钮完成虚拟机的创建。

图 3.1　在 VirtualBox 中安装极简版 Linux 发行版

创建虚拟机后，打开该虚拟机的设置窗口，系统相关的设置如图 3.2 所示，在“处理器”选项卡中，处理器数量选择 2 或 4(根据自己计算机中 CPU 核数而定)。在“显示”设置中，选中“屏幕”选项卡，显存大小设置为 128MB。

图 3.2　虚拟机的设置

启动虚拟机，显示 GRUB 的操作系统选择界面，如图 3.3 所示。选择 minilinux 菜单项，按回车键，即可正常启动 Linux 操作系统，命令行界面如图 3.4 所示，此时就可以使用极简版 Linux 发行版了。

图 3.3　GRUB 的操作系统选择界面

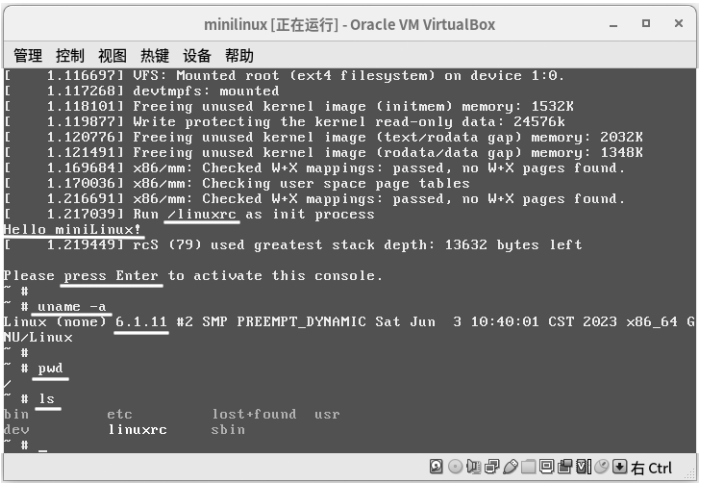

图 3.4　Linux 操作系统的命令行界面