

**SEAWAY Edge**

**管理员指南**

中科海微（北京）科技有限公司

Seaway Technologies Co. Ltd

## **目录**

目录 1

文档修订目录 1

文档版本 1

Version3.2 1

第一章 查看系统信息 4

1.1查看系统信息 4

1.2查看系统相关的资源信息 4

第二章 基础配置 5

2.1 设置语言环境 5

2.2 设置键盘 5

2.3 设置日期和时间 6

2.4 设置磁盘调度算法 7

第三章 管理用户和用户组 8

3.1 管理用户 8

3.1.1 增加用户 8

3.1.2 修改用户账号 8

3.1.3 删除用户 9

3.2 管理用户组 9

3.2.1 增加用户组 9

3.2.2 修改用户组 9

3.2.3 删除用户组 10

3.2.4 将用户加入用户组或从用户组中移除 10

3.2.5 切换用户组 10

第四章 使用DNF管理软件包 10

4.1 配置DNF 10

4.1.1 DNF配置文件 10

4.1.2 创建本地软件源仓库 11

4.1.3 添加、启动和禁用软件源 11

4.2 管理软件包 12

4.2.1 搜索软件包 12

4.2.2 列出软件包清单 12

4.2.3 安装RPM包 12

4.2.4 删除软件包 12

4.3 管理软件包组 12

4.4 检查并更新 13

第五章 管理服务 14

5.1 简介 14

5.2 特性说明 15

5.2.1 更快的启动速度 15

5.2.2 提供按需启动能力 15

5.2.3 采用cgroup特性跟踪和管理进程的生命周期 15

5.2.4 启动挂载点和自动挂载的管理 16

5.2.5 实现事务性依赖关系管理 16

5.2.6 与SysV初始化脚本兼容 16

5.2.7 能够对系统进行快照和恢复 16

5.3 管理系统服务 17

5.3.1 sysvinit命令和systemd命令 17

5.3.2 显示所有当前服务 18

5.3.3 显示服务状态 18

5.3.4 运行服务 19

5.3.5 关闭服务 19

5.3.6 重启服务 20

5.3.7 启用服务 20

5.3.8 禁用服务 20

5.4 改变运行级别 20

5.4.1 Target和运行级别 20

5.4.2 查看系统默认启动目标 21

5.4.3 查看当前系统所有的启动目标 21

5.4.4 改变默认目标 21

5.4.5 改变当前目标 21

5.4.6 切换到救援模式 21

5.4.7 切换到紧急模式 22

5.5 关闭、暂停和休眠系统 22

第六章 管理进程 23

6.1 查看进程 23

6.2 调度启动进程 25

6.3 挂起/恢复进程 25

第七章 配置网络 25

7.1 配置IP 25

7.1.1 使用nmcli命令 25

7.1.2 使用ip命令 27

7.1.3 通过ifcfg文件配置网络 28

7.1.4 通过 nmtui 工具 29

7.2 配置主机名 29

7.2.1 使用hostnamectl配置主机名 29

7.2.2 使用nmcli配置主机名 29

7.2.3 通过 nmtui 工具 30

7.3 配置网络绑定 30

7.3.1 使用nmcli 30

7.3.2 使用命令行 30

7.3.3 使用 nmtui 工具 32

第八章 使用LVM管理硬盘 32

8.1 LVM简介 32

8.2 安装 32

8.3 管理物理卷 33

8.4 管理卷组 33

8.5 管理逻辑卷 34

8.6 创建并挂载文件系统 34

# 文档修订目录

## 文档版本

## Version3.2

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号 | 修订日期 | 修订内容 | 修订人 |
| V3.1 | 2023/04/29 | 初订 | 陈海鑫 |

**前言**

在使用本手册之前，请您认真阅读以下使用许可协议，只有在同意以下使用许可协议的情况下方能使用本手册中介绍的产品。

**版权声明**

中科海微（北京）科技有限公司版权所有，并保留对本文档及本声明的最终解释权和修改权。本文档中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明外，其著作权或其他相关权利均属于中科海微（北京）科技有限公司。未经中科海微（北京）科技有限公司书面同意，任何人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、翻译成其它语言、将其全部或部分用于商业用途**。**

**免责条款**

本文档依据现有信息制作，其内容如有更改，恕不另行通知。中科海微（北京）科技有限公司在编写该文档的时候已尽最大努力保证其内容准确可靠，但中科海微（北京）科技有限公司不对本文档中的遗漏、不准确、或错误导致的损失和损害承担责任。

**技术支持与信息反馈**

如果您在使用我们的产品时遇到问题,或者您认为我们的产品有某些功能缺陷,请访问我们的官网www.haiwei.tech联系我们的客服,我们将为您解决问题和反馈;或者需要技术支持指导以及有任何宝贵意见，也请您通过官网或者电话联系我们：

**管理员指南简介**

本文档提供了SeawayEdge系统常用的管理员操作，方便管理员更好地使用SeawayEdge。本文档适用于所有使用SeawayEdge系统的管理员。

**管理员指南**

本文档主要适用于使用SeawayEdge的用户。用户需要具备以下经验和技能：

* 熟悉Linux基本操作
* 对Linux有一定了解

# 查看系统信息

## 1.1查看系统信息

​例如，命令和输出如下：



## 1.2查看系统相关的资源信息

* 查看CPU信息：



* 查看内存信息：



* 查看磁盘信息：



* 查看系统资源实时信息：



# 第二章 基础配置

## 2.1 设置语言环境

您可以通过localectl修改系统的语言环境，对应的参数设置保存在/etc/locale.conf文件中。这些参数会在系统启动过程中被systemd的守护进程读取。

* 显示当前语言环境状态

显示当前语言环境，命令如下：



* 列出可用的语言环境



例如显示当前系统中所有可用的中文环境，命令和输出如下：



* 设置语言环境

要设置语言环境，在root权限下执行如下命令，其中 locale 是您要设置的语言类型，取值范围可通过localectl list-locales获取，请根据实际情况修改。



## 设置键盘

您可以通过localectl修改系统的键盘设置，对应的参数设置保存在/etc/locale.conf文件中。这些参数，会在系统启动的早期被systemd的守护进程读取。

* 显示当前设置

显示当前键盘设置，命令如下：



* 列出可用的键盘布局

显示当前可用的键盘布局，命令如下：



* 设置键盘布局

设置键盘布局，在root权限下执行如下命令，其中 map 是您想要设置的键盘类型，取值范围可通过localectl list-keymaps获取，请根据实际情况修改：



## 设置日期和时间

本节介绍如何通过timedatectl、date、hwclock命令来设置系统的日期、时间和时区等。

* 使用timedatectl命令设置

显示日期和时间

通过远程服务器进行时间同步

修改日期

修改时间

修改时区

* 使用date命令设置

显示当前的日期和时间

| 格式参数 | 说明 |
| --- | --- |
| %H | 小时以HH格式（例如 17）。 |
| %M | 分钟以MM格式（例如 37）。 |
| %S | 秒以SS格式（例如 25）。 |
| %d | 日期以DD格式（例如 15）。 |
| %m | 月份以MM格式（例如 07）。 |
| %Y | 年份以YYYY格式（例如 2019）。 |
| %Z | 时区缩写（例如CEST）。 |
| %F | 日期整体格式为YYYY-MM-DD（例如 2019-7-15），等同%Y-%m-%d。 |
| %T | 时间整体格式为HH:MM:SS（例如 18:30:25），等同%H:%M:%S。 |

修改时间



修改日期

* 使用hwclock命令设置

显示日期和时间

设置日期和时间

## 设置磁盘调度算法

本节介绍如何设置磁盘调度算法。

* 临时修改调度策略

例如将所有IO调度算法修改为mq-deadline，此修改重启后会失效。



* 永久设置调度策略

可以通过在内核启动配置文件grub.cfg中的kernel行追加：elevator=mq-deadline，重启后生效。

# 第三章 管理用户和用户组

本节介绍UADK以及UADK engine的安装、升级和卸载，其他厂商的硬件加速器设备类似。 需要鲲鹏（Kunpeng）加速器内核态驱动、UADK用户驱动态配合使用。请按照顺序进行安装。

## 3.1 管理用户

## 3.1.1 增加用户

* useradd命令增加用户：



* 用户信息文件

与用户帐号信息有关的文件如下：

/etc/passwd：用户帐号信息文件。/etc/shadow：用户帐号信息加密文件。/etc/group：组信息文件。/etc/default/useradd：定义默认设置文件。/etc/login.defs：系统广义设置文件。/etc/skel：默认的初始配置文件目录。

* 创建用户示例

例如新建一个用户名为userexample的用户，在root权限下执行如下命令：



使用id命令查看新建的用户信息，命令如下：



修改用户userexample的密码：



## 3.1.2 修改用户账号

* 修改密码

普通用户可以用passwd修改自己的密码，只有管理员才能用passwd username为其他用户修改密码。

* 修改用户shell设置

使用chsh命令可以修改自己的shell，只有管理员才能用chsh username为其他用户修改shell设置。用户也可以使用usermod命令修改shell信息，在root权限下执行如下命令，其中 new\_shell\_path 为目标shell路径，username 为要修改用户的用户名，请根据实际情况修改：

* 修改主目录



* 修改UID

修改用户ID，在root权限下执行如下命令，其中 UID 代表目标用户ID，username 代表用户名，请根据实际情况修改：



## 3.1.3 删除用户

 在root权限下，使用userdel命令可删除现有用户。

 例如，删除用户Test，命令如下：



## 3.2 管理用户组

## 3.2.1 增加用户组

* groupadd增加用户组



* 用户组信息文件

与用户组信息有关的文件如下：/etc/gshadow：用户组信息加密文件。/etc/group：组信息文件。/etc/login.defs：系统广义设置文件。

## 3.2.2 修改用户组

* 修改GID

修改用户组ID，在root权限下执行如下命令，其中 GID 代表目标用户组ID， groupname 代表用户组，请根据实际情况修改：



* 修改用户组名

修改用户组名，在root权限下执行如下命令，其中 newgroupname 代表新用户组名， oldgroupname 代表已经存在的待修改的用户组名，请根据实际情况修改：



## 3.2.3 删除用户组

在root权限下，使用groupdel命令可删除用户组。例如，删除用户组Test，命令如下：

## 3.2.4 将用户加入用户组或从用户组中移除

在root权限下，使用gpasswd命令将用户加入用户组或从用户组中移除。例如，将用户 userexample 加入用户组 Test ，命令如下：

例如，将用户 userexample 从 Test 用户组中移除，命令如下：



## 3.2.5 切换用户组

一个用户同时属于多个用户组时，在该用户登录后，使用newgrp命令可以切换到其他用户组，以便具有其他用户组的权限。例如，将用户 userexample 切换到 Test 用户组，命令如下：



# 第四章 使用DNF管理软件包

## 4.1 配置DNF

## 4.1.1 DNF配置文件

使用export命令设置队列数量：

DNF的主要配置文件是/etc/dnf/dnf.conf，该文件包含两部分：

* “main”部分保存着DNF的全局设置。
* “repository”部分保存着软件源的设置，可以有零个或多个“repository”。

另外，在/etc/yum.repos.d目录中保存着零个或多个repo源相关文件，它们也可以定义不同的“repository”。所以openEuler软件源的配置一般有两种方式，一种是直接配置/etc/dnf/dnf.conf文件中的“repository”部分，另外一种是在/etc/yum.repos.d目录下增加.repo文件。

显示当前配置

* 要显示当前的配置信息：



* 要显示相应软件源的配置，首先查询repo id：



执行如下指令查询配置



## 4.1.2 创建本地软件源仓库

 要建立一个本地软件源仓库，请按照下列步骤操作。

* 安装createrepo软件包。在root权限下执行如下命令：



* 将需要的软件包复制到一个目录下，如/mnt/local\_repo/ 。
* 创建软件源，执行以下命令：



## 4.1.3 添加、启动和禁用软件源

 本节将介绍如何通过“dnf config-manager”命令添加、启用和禁用软件源仓库。

* 添加软件源

要定义一个新的软件源仓库，您可以在 /etc/dnf/dnf.conf 文件中添加“repository”部分，或者在/etc/yum.repos.d/目录下添加“.repo”文件进行说明。建议您通过添加“.repo”的方式，每个软件源都有自己对应的“.repo”文件，以下介绍该方式的操作方法。要在您的系统中添加一个这样的源，请在root权限下执行如下命令，执行完成之后会在/etc/yum.repos.d/目录下生成对应的repo文件。其中 repository\_url 为repo源地址：



* 启动软件源

要启用软件源，请在root权限下执行如下命令，其中 repository 为新增.repo文件中的repo id（可通过dnf repolist查询）：



* 禁用软件源

要禁用软件源，请在root权限下执行如下命令：



## 4.2 管理软件包

 使用dnf能够让您方便的进行查询、安装、删除软件包等操作。

## 4.2.1 搜索软件包

 您可以使用rpm包名称、缩写或者描述搜索需要的RPM包，使用命令如下：



## 4.2.2 列出软件包清单

要列出系统中所有已安装的以及可用的RPM包信息，使用命令如下：



## 4.2.3 安装RPM包

 要安装一个软件包及其所有未安装的依赖，请在root权限下执行如下命令：

 

## 4.2.4 删除软件包

 要卸载软件包以及相关的依赖软件包，请在root权限下执行如下命令：

 

## 4.3 管理软件包组

软件包集合是服务于一个共同目的的一组软件包，例如系统工具集等。使用dnf可以对软件包组进行安装/删除等操作，使相关操作更高效。

* 列出软件包组清单

使用summary参数，可以列出系统中所有已安装软件包组、可用的组、可用的环境组的数量，命令如下：



要列出所有软件包组和它们的组ID ，命令如下：



* 安装软件包组

每一个软件包组都有自己的名称以及相应的ID（groupid），您可以使用软件包组名称或它的ID进行安装。要安装一个软件包组，请在root权限下执行如下命令：

* 删除软件包组

要卸载软件包组，您可以使用软件包组名称或它的ID，在root权限下执行如下命令：



## 4.4 检查并更新

dnf可以检查您的系统中是否有软件包需要更新。您可以通过dnf列出需要更新的软件包，并可以选择一次性全部更新或者只对指定包进行更新。

* 检查更新

如果您需要显示当前系统可用的更新，使用命令如下：



* 升级

如果您需要升级单个软件包，在root权限下执行如下命令：



* 更新所有的包和他们的依赖

要更新所有的包和它们的依赖，在root权限下执行如下命令：



# 第五章 管理服务

## 5.1 简介

systemd是在Linux下，与SysV和LSB初始化脚本兼容的系统和服务管理器。systemd使用socket和D-Bus来开启服务，提供基于守护进程的按需启动策略，支持快照和系统状态恢复，维护挂载和自挂载点，实现了各服务间基于从属关系的一个更为精细的逻辑控制，拥有更高的并行性能。

* 概念介绍

systemd开启和监督整个系统是基于unit的概念。unit是由一个与配置文件对应的名字和类型组成的（例如：avahi.service unit有一个具有相同名字的配置文件，是守护进程Avahi的一个封装单元）。unit有多种类型，如表所示。

| unit名称 | 后缀名 | 描述 |
| --- | --- | --- |
| Service unit | .service | 系统服务。 |
| Target unit | .target | 一组systemd units。 |
| Automount unit | .automount | 文件系统挂载点。 |
| Device unit | .device | 内核识别的设备文件。 |
| Mount unit | .mount | 文件系统挂载点。 |
| Path unit | .path | 在一个文件系统中的文件或目录。 |
| Scope unit | .scope | 外部创建的进程。 |
| Slice unit | .slice | 一组用于管理系统进程分层组织的units。 |
| Socket unit | .socket | 一个进程间通信的Socket。 |
| Swap unit | .swap | swap设备或者swap文件。 |
| Timer unit | .timer | systemd计时器。 |

所有的可用systemd unit类型，可在如下表所示的路径下查看。

| 路径 | 描述 |
| --- | --- |
| /usr/lib/systemd/system/ | 随安装的RPM产生的systemd units。 |
| /run/systemd/system/ | 在运行时创建systemd units。 |
| /etc/systemd/system/ | 由系统管理员创建和管理的systemd units。 |

## 5.2 特性说明

## 5.2.1 更快的启动速度

systemd提供了比UpStart更激进的并行启动能力，采用了socket/D-Bus activation等技术启动服务，带来了更快的启动速度。为了减少系统启动时间，systemd的目标是：

* 尽可能启动更少的进程。
* 尽可能将更多进程并行启动。

## 5.2.2 提供按需启动能力

当sysvinit系统初始化的时候，它会将所有可能用到的后台服务进程全部启动运行。并且系统必须等待所有的服务都启动就绪之后，才允许用户登录。这种做法有两个缺点：首先是启动时间过长；其次是系统资源浪费。某些服务很可能在很长一段时间内，甚至整个服务器运行期间都没有被使用过。比如CUPS，打印服务在多数服务器上很少被真正使用到。您可能没有想到，在很多服务器上SSHD也是很少被真正访问到的。花费在启动这些服务上的时间是不必要的；同样，花费在这些服务上的系统资源也是一种浪费。systemd可以提供按需启动的能力，只有在某个服务被真正请求的时候才启动它。当该服务结束，systemd可以关闭它，等待下次需要时再次启动它。

## 5.2.3 采用cgroup特性跟踪和管理进程的生命周期

init系统的一个重要职责就是负责跟踪和管理服务进程的生命周期。它不仅可以启动一个服务，也能够停止服务。这看上去没有什么特别的，然而在真正用代码实现的时候，您或许会发现停止服务比一开始想的要困难。服务进程一般都会作为守护进程（daemon）在后台运行，为此服务程序有时候会派生（fork）两次。在UpStart中，需要在配置文件中正确地配置expect小节。这样UpStart通过对fork系统调用进行计数，从而获知真正的精灵进程的PID号。cgroup已经出现了很久，它主要用来实现系统资源配额管理。cgroup提供了类似文件系统的接口，使用方便。当进程创建子进程时，子进程会继承父进程的cgroup。因此无论服务如何启动新的子进程，所有的这些相关进程都会属于同一个cgroup，systemd只需要简单地遍历指定的cgroup即可正确地找到所有的相关进程，将它们逐一停止即可。

## 5.2.4 启动挂载点和自动挂载的管理

传统的Linux系统中，用户可以用/etc/fstab文件来维护固定的文件系统挂载点。这些挂载点在系统启动过程中被自动挂载，一旦启动过程结束，这些挂载点就会确保存在。这些挂载点都是对系统运行至关重要的文件系统，比如HOME目录。和sysvinit一样，systemd管理这些挂载点，以便能够在系统启动时自动挂载它们。systemd还兼容/etc/fstab文件，您可以继续使用该文件管理挂载点。有时候用户还需要动态挂载点，比如打算访问DVD内容时，才临时执行挂载以便访问其中的内容，而不访问光盘时该挂载点被取消（umount)，以便节约资源。传统地，人们依赖autofs服务来实现这种功能。systemd内建了自动挂载服务，无需另外安装autofs服务，可以直接使用systemd提供的自动挂载管理能力来实现autofs的功能。

## 5.2.5 实现事务性依赖关系管理

系统启动过程是由很多的独立工作共同组成的，这些工作之间可能存在依赖关系，比如挂载一个NFS文件系统必须依赖网络能够正常工作。systemd虽然能够最大限度地并发执行很多有依赖关系的工作，但是类似“挂载NFS”和“启动网络”这样的工作还是存在天生的先后依赖关系，无法并发执行。对于这些任务，systemd维护一个“事务一致性”的概念，保证所有相关的服务都可以正常启动而不会出现互相依赖，以至于死锁的情况。

## 5.2.6 与SysV初始化脚本兼容

和UpStart一样，systemd引入了新的配置方式，对应用程序的开发也有一些新的要求。如果systemd想替代目前正在运行的初始化系统，就必须和现有程序兼容。任何一个Linux发行版都很难为了采用systemd而在短时间内将所有的服务代码都修改一遍。systemd提供了和sysvinit以及LSB initscripts兼容的特性。系统中已经存在的服务和进程无需修改。这降低了系统向systemd迁移的成本，使得systemd替换现有初始化系统成为可能。

## 5.2.7 能够对系统进行快照和恢复

systemd支持按需启动，因此系统的运行状态是动态变化的，人们无法准确地知道系统当前运行了哪些服务。systemd快照提供了一种将当前系统运行状态保存并恢复的能力。比如系统当前正运行服务A和B，可以用systemd命令行对当前系统运行状况创建快照。然后将进程A停止，或者做其他的任意的对系统的改变，比如启动新的进程C。在这些改变之后，运行systemd的快照恢复命令，就可立即将系统恢复到快照时刻的状态，即只有服务A和B在运行。一个可能的应用场景是调试：比如服务器出现一些异常，为了调试用户将当前状态保存为快照，然后可以进行任意的操作，比如停止服务等。等调试结束，恢复快照即可。

## 5.3 管理系统服务

 systemd提供systemctl命令来运行、关闭、重启、显示、启用/禁用系统服务。

## 5.3.1 sysvinit命令和systemd命令

systemd提供systemctl命令与sysvinit命令的功能类似。当前版本中依然兼容service和chkconfig命令，相关说明如表3，但建议用systemctl进行系统服务管理。

| **sysvinit命令** | **systemd命令** | **备注** |
| --- | --- | --- |
| service network start | systemctl start network.service | 用来启动一个服务 (并不会重启现有的)。 |
| service network stop | systemctl stop network.service | 用来停止一个服务 (并不会重启现有的)。 |
| service network restart | systemctl restart network.service | 用来停止并启动一个服务。 |
| service network reload | systemctl reload network.service | 当支持时，重新装载配置文件而不中断等待操作。 |
| service network condrestart | systemctl condrestart network.service | 如果服务正在运行那么重启它。 |
| service network status | systemctl status network.service | 检查服务的运行状态。 |
| chkconfig network on | systemctl enable network.service | 在下次启动时或满足其他触发条件时设置服务为启用。 |
| chkconfig network off | systemctl disable network.service | 在下次启动时或满足其他触发条件时设置服务为禁用。 |
| chkconfig network | systemctl is-enabled network.service | 用来检查一个服务在当前环境下被配置为启用还是禁用。 |
| 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。 |
| chkconfig network \-\-list | ls /etc/systemd/system/\*.wants/network.service | 用来列出该服务在哪些运行级别下启用和禁用。 |
| chkconfig network \-\-add | systemctl daemon-reload | 当您创建新服务文件或者变更设置时使用。 |

## 5.3.2 显示所有当前服务

 如果您需要显示当前正在运行的服务，使用命令如下：

 

## 5.3.3 显示服务状态

 如果您需要显示某个服务的状态，可执行如下命令：

 

相关状态显示参数说明如表所示。

| **参数** | **描述** |
| --- | --- |
| Loaded | 说明服务是否被加载，并显示服务对应的绝对路径以及是否启用。 |
| Active | 说明服务是否正在运行，并显示时间节点。 |
| Main PID | 相应的系统服务的PID值。 |
| CGroup | 相关控制组（CGroup）的其他信息。 |

如果您需要判断某个服务是否被启用，可执行如下命令：



is-enabled命令的返回结果如下：

| 状态 | 含义 |
| --- | --- |
| enabled | 已经通过 /etc/systemd/system/ 目录下的 Alias= 别名、 .wants/ 或 .requires/ 软链接被永久启用。 |
| enabled-runtime | 已经通过 /run/systemd/system/ 目录下的 Alias= 别名、 .wants/ 或 .requires/ 软链接被临时启用。 |
| linked | 虽然单元文件本身不在标准单元目录中，但是指向此单元文件的一个或多个软链接已经存在于 /etc/systemd/system/ 永久目录中。 |
| linked-runtime | 虽然单元文件本身不在标准单元目录中，但是指向此单元文件的一个或多个软链接已经存在于 /run/systemd/system/ 临时目录中。 |
| masked | 已经被 /etc/systemd/system/ 目录永久屏蔽(软链接指向 /dev/null 文件)，因此 **start** 操作会失败。 |
| masked-runtime | 已经被 /run/systemd/systemd/ 目录临时屏蔽(软链接指向 /dev/null 文件)，因此 **start** 操作会失败。 |
| static | 尚未被启用，并且单元文件的 "[Install]" 小节中没有可用于 **enable** 命令的选项。 |
| indirect | 尚未被启用，但是单元文件的 "[Install]" 小节中 Also= 选项的值列表非空(也就是列表中的某些单元可能已被启用)、或者它拥有一个不在 Also= 列表中的其他名称的别名软链接。对于模版单元来说，表示已经启用了一个不同于 DefaultInstance= 的实例。 |
| disabled | 尚未被启用，但是单元文件的 "[Install]" 小节中存在可用于 **enable** 命令的选项。 |
| generated | 单元文件是被单元生成器动态生成的。被生成的单元文件可能并未被直接启用，而是被单元生成器隐含的启用了。 |
| transient | 单元文件是被运行时API动态临时生成的。该临时单元可能并未被启用。 |
| bad | 单元文件不正确或者出现其他错误。 **is-enabled** 不会返回此状态，而是会显示一条出错信息。 **list-unit-files** 命令有可能会显示此单元。 |

## 5.3.4 运行服务

 如果您需要运行某个服务，请在root权限下执行如下命令：

 

## 5.3.5 关闭服务

 如果您需要关闭某个服务，请在root权限下执行如下命令：

 

## 5.3.6 重启服务

 如果您需要重启某个服务，请在root权限下执行如下命令：

 

## 5.3.7 启用服务

 如果您需要在开机时启用某个服务，请在root权限下执行如下命令：

 

## 5.3.8 禁用服务

 如果您需要在开机时禁用某个服务，请在root权限下执行如下命令：

 

## 5.4 改变运行级别

## 5.4.1 Target和运行级别

systemd用目标（target）替代了运行级别的概念，提供了更大的灵活性，如您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。下表列举了systemd下的目标和常见runlevel的对应关系。

| 运行级别 | systemd目标（target） | 描述 |
| --- | --- | --- |
| 0 | runlevel0.target，poweroff.target | 关闭系统。 |
| 1, s, single | runlevel1.target，rescue.target | 单用户模式。 |
| 2, 4 | runlevel2.target，runlevel4.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。 |
| 5 | runlevel5.target，graphical.target | 多用户，图形化。通常为所有运行级别3的服务外加图形化登录。 |
| 6 | runlevel6.target，reboot.target | 重启系统。 |
| emergency | emergency.target | 紧急Shell。 |

## 5.4.2 查看系统默认启动目标

 查看当前系统默认的启动目标，命令如下：

 

## 5.4.3 查看当前系统所有的启动目标

 查看当前系统所有的启动目标，命令如下：

 

## 5.4.4 改变默认目标

 改变系统默认的目标，在root权限下执行如下命令：

 

## 5.4.5 改变当前目标

 改变当前系统的目标，在root权限下执行如下命令：



## 5.4.6 切换到救援模式

 改变当前系统为救援模式，在root权限下执行如下命令：

 

## 5.4.7 切换到紧急模式

 改变当前系统为紧急模式，在root权限下执行如下命令：

 

## 5.5 关闭、暂停和休眠系统

* systemctl命令

systemd通过systemctl命令可以对系统进行关机、重启、休眠等一系列操作。当前仍兼容部分Linux常用管理命令，对应关系如下表。建议用户使用systemctl命令进行操作。

| Linux常用管理命令 | systemctl命令 | 描述 |
| --- | --- | --- |
| halt | systemctl halt | 关闭系统 |
| poweroff | systemctl poweroff | 关闭电源 |
| reboot | systemctl reboot | 重启 |

* 关闭系统

关闭系统并下电，在root权限下执行如下命令：



* 重启系统

重启系统，在root权限下执行如下命令：



* 使系统待机

使系统待机，在root权限下执行如下命令：



* 使系统休眠

使系统休眠，在root权限下执行如下命令：



# 第六章 管理进程

操作系统管理多个用户的请求和多个任务。大多数系统都只有一个CPU和一个主要存储，但一个系统可能有多个二级存储磁盘和多个输入/输出设备。操作系统管理这些资源并在多个用户间共享资源，当用户提出一个请求时，造成好像系统被用户独占的假象。实际上操作系统监控着一个等待执行的任务队列，这些任务包括用户任务、操作系统任务、邮件和打印任务等。本章节将从用户的角度讲述如何控制进程。

## 6.1 查看进程

Linux是一个多任务系统，经常需要对这些进程进行一些调配和管理。要进行管理，首先就要知道现在的进程情况：有哪些进程、进程的状态如何等。Linux提供了多种命令来了解进程的状况。

* Who命令

who命令主要用于查看当前系统中的用户情况。如果用户想和其他用户建立即时通讯，比如使用talk命令，那么首先要确定的就是该用户确实在线上，不然talk进程就无法建立起来。又如，系统管理员希望监视每个登录的用户此时此刻的所作所为，也要使用who命令。who命令应用起来非常简单，可以比较准确地掌握用户的情况，所以使用非常广泛。



* Ps命令

ps命令是最基本又非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵尸、哪些进程占用了过多的资源等，大部分进程信息都是可以通过执行该命令得到的。ps命令最常用的还是用来监控后台进程的工作情况，因为后台进程是不与屏幕、键盘这些标准输入/输出设备进行通信的，所以如果需要检测其状况，就可使用ps命令。ps命令的常见选项如下表所示。

| 选项 | 描述 |
| --- | --- |
| -e | 显示所有进程。 |
| -f | 全格式。 |
| -h | 不显示标题。 |
| -l | 使用长格式。 |
| -w | 宽行输出。 |
| -a | 显示终端上的所有进程，包括其他用户的进程。 |
| -r | 只显示正在运行的进程。 |
| -x | 显示没有控制终端的进程。 |

ps命令确定当前正在运行中的进程



* Top命令

top命令和ps命令的基本作用是相同的，显示系统当前的进程和其他状况，但是top是一个动态显示过程，即可以通过用户按键来不断刷新进程的当前状态，如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。其实top命令提供了实时的对系统处理器的状态监视。它将显示系统中CPU的任务列表。该命令可以按CPU使用、内存使用和执行时间对任务进行排序，而且该命令的很多特性都可以通过交互式命令或者在定制文件中进行设定。top命令输出的示例如图所示：

* Kill命令

当需要中断一个前台进程的时候，通常使用“Ctrl+c”组合键，而对于后台进程不能用组合键来终止，这时就可以使用kill命令。该命令可以终止前台和后台进程。终止后台进程的原因包括：该进程占用CPU的时间过多、该进程已经死锁等。kill命令是通过向进程发送指定的信号来结束进程的。如果没有指定发送的信号，那么默认值为TERM信号。TERM信号将终止所有不能捕获该信号的进程。至于那些可以捕获该信号的进程可能就需要使用KILL信号（它的编号为9），而该信号不能被捕捉。kill指令可以中断某一进程：



## 6.2 调度启动进程

* 定时运行一批程序（at）

用户使用at命令在指定时刻执行指定的命令序列。该命令至少需要指定一个命令和一个执行时间。at命令可以只指定时间，也可以时间和日期一起指定。

at命令定时运行一批程序（at命令之后，会出现提示符at>，提示用户输入命令，在此输入了slocate -u，然后按回车键。还可以输入多条命令，当所有要执行的命令输入结束后，按“Ctrl+d”键结束at命令）



* 周期性运行一批程序（cron）

cron命令周期性运行一批程序（cron命令会搜索/var/spool/cron目录，寻找以/etc/passwd文件中的用户名命名的crontab文件，被找到的这种文件将装入内存。在crontab文件中输入需要执行的命令和时间。 ）首先创建一个文件，使用crontab命令使它成为crontab文件。

## 6.3 挂起/恢复进程

作业控制允许进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按“Ctrl+Z”键，即可挂起当前的前台作业。在键盘上按“Ctrl+Z”键后，将挂起当前执行的命令cat。使用jobs命令可以显示shell的作业清单，包括具体的作业、作业号以及作业当前所处的状态。恢复进程执行时，有两种选择：用fg命令将挂起的作业放回到前台执行；用bg命令将挂起的作业放到后台执行。灵活使用上述命令，将给自己带来很大的方便。

# 第七章 配置网络

## 7.1 配置IP

## 7.1.1 使用nmcli命令

nmcli是NetworkManager的一个命令行工具，它提供了使用命令行配置由NetworkManager管理网络连接的方法.

常用命令使用举例如下：

* 显示NetworkManager状态：



* 显示所有连接



* 只显示当前活动连接，如下所示添加-a,--active：



* 显示由NetworkManager识别到的设备及其状态：



* 使用nmcli工具启动和停止网络接口，在root权限下执行如下命令：





* 设备管理

连接到设备



断开设备连接：



* 设置网络连接

列出目前可用的网络连接：



添加一个网络连接会生成相应的配置文件，并与相应的设备关联。检查可用的设备，方法如下：



* 配置动态IP连接

配置IP

要使用 DHCP 分配网络时，可以使用动态IP配置添加网络配置文件，命令格式如下：



激活连接并检查状态：



* 配置静态IP连接

配置IP

添加静态 IPv4 配置的网络连接，可使用以下命令：



激活连接并检查状态：



* 添加 Wi-Fi 连接

通过配置文件连接Wi-Fi

使用以下命令查看Wi-Fi访问点：



以下命令生成使用的静态 IP 配置，但允许自动 DNS 地址分配的 Wi-Fi连接：



使用以下命令设定 WPA2 密码，例如 “answer”：





使用以下命令更改Wi-Fi状态（on/off）:



* 配置静态路由

使用nmcli命令为网络连接配置静态路由，使用命令如下：



使用编辑器配置静态路由，使用如下命令：



## 7.1.2 使用ip命令

* 配置静态IP地址



* 配置静态路由

在root权限下使用 ip route 命令显示当前的 IP 路由表：



添加静态路由：



## 7.1.3 通过ifcfg文件配置网络

* 配置静态网络

以enp4s0网络接口进行静态网络设置为例，通过在root权限下修改ifcfg文件实现，在/etc/sysconfig/network-scripts/目录中生成名为ifcfg-enp4s0的文件中，修改参数配置，示例如下：

TYPE=EthernetPROXY\_METHOD=noneBROWSER\_ONLY=noBOOTPROTO=noneIPADDR=192.168.0.10GATEWAY=192.168.0.1PREFIX=24DEFROUTE=yesIPV4\_FAILURE\_FATAL=noIPV6INIT=yesIPV6\_AUTOCONF=yesIPV6\_DEFROUTE=yesIPV6\_FAILURE\_FATAL=noIPV6\_ADDR\_GEN\_MODE=stable-privacyNAME=enp4s0staticUUID=08c3a30e-c5e2-4d7b-831f-26c3cdc29293DEVICE=enp4s0ONBOOT=yes

* 配置动态网络

要通过ifcfg文件为em1接口配置动态网络，请按照如下操作在/etc/sysconfig/network-scripts/目录中生成名为 ifcfg-em1 的文件，示例如下：



要配置一个向DHCP服务器发送不同的主机名的接口，请在ifcfg文件中新增一行内容，如下所示：



要配置忽略由DHCP服务器发送的路由，防止网络服务使用从DHCP服务器接收的DNS服务器更新/etc/resolv.conf。请在ifcfg文件中新增一行内容，如下所示：



要配置一个接口使用具体DNS服务器，请将参数PEERDNS=no，并在ifcfg文件中添加以下行：



其中ip-address是DNS服务器的地址。这样就会让网络服务使用指定的DNS服务器更新/etc/resolv.conf。

* 配置默认网关

在确定默认网关时，首先解析 /etc/sysconfig/network 文件，然后解析 ifcfg 文件 ，将最后读取的 GATEWAY 的取值作为路由表中的默认路由。在动态网络环境中，使用 NetworkManager 管理主机时，建议设置为由 DHCP 来分配。

## 7.1.4 通过 nmtui 工具

 nmtui 工具提供了一个交互式的界面，可以用来配置网络连接。要使用 nmtui 工具，以 root 权限执行以下命令：



选择 Edit a connection 选项，然后选择要编辑的网络连接，按 Enter 键，进入编辑界面。在交互界面中，可以使用方向键选择要编辑的选项，按 Tab 键切换到下一个选项，按 Enter 键进入编辑状态，按 Esc 键退出编辑状态。可以使用方向键选择 IPV4 CONFIGURATION 或 IPV6 CONFIGURATION 的配置方式，并选择 Show 显示详细信息。

## 7.2 配置主机名

## 7.2.1 使用hostnamectl配置主机名

* 查看所有主机名：



* 设定所有主机名：



* 远程更改主机名：



## 7.2.2 使用nmcli配置主机名

 查询static主机名，使用如下命令：



在root权限下，将static主机名设定为host-server，使用如下命令：



要让系统hostnamectl感知到static主机名的更改，在root权限下，重启hostnamed服务，使用如下命令：



## 7.2.3 通过 nmtui 工具

 nmtui 提供了一个交互式的界面，可以用来配置网络连接。要使用 nmtui 工具，以 root 权限执行以下命令：



选择 Set system hostname 选项，输入新的主机名，然后按 Enter 键。选择 OK 确认修改。

## 7.3 配置网络绑定

## 7.3.1 使用nmcli

* 创建名为mybond0的绑定：



* 添加从属接口：



* 要启动绑定，首先需启动从属接口：





## 7.3.2 使用命令行

* 检查是否已安装Bonding内核模块

在系统中默认已加载相应模块。要载入绑定模块，可在root权限下使用如下命令：



显示该模块的信息，可在root权限下使用如下命令：



* 创建频道绑定接口

要创建绑定接口，可在root权限下通过在 /etc/sysconfig/network-scripts/ 目录中创建名为 ifcfg-bondN 的文件（使用接口号码替换 N，比如 0）。根据要绑定接口类型的配置文件来编写相应的内容，比如网络接口。接口配置文件示例如下：DEVICE=bond0

NAME=bond0

TYPE=Bond

BONDING\_MASTER=yes

IPADDR=192.168.1.1

PREFIX=24

ONBOOT=yes

BOOTPROTO=none

BONDING\_OPTS="bonding parameters separated by spaces"

* 创建从属接口

创建频道绑定接口后，必须在从属接口的配置文件中添加 MASTER 和 SLAVE 指令。例如将两个网络接口enp3s0 和 enp4s0 以频道方式绑定，其配置文件示例分别如下：TYPE=EthernetNAME=bond-slave-enp3s0UUID=3b7601d1-b373-4fdf-a996-9d267d1cac40DEVICE=enp3s0ONBOOT=yesMASTER=bond0SLAVE=yes

* 激活频道绑定

要激活绑定，则需要启动所有从属接口。请在root权限下，运行以下命令：



完成后，启动所有从属接口以便启动绑定（不将其设定为 “down”）。要让 NetworkManager 感知到系统所做的修改，在每次修改后，请在root权限下，运行以下命令：

查看绑定接口的状态，请在root权限下运行以下命令：



* 创建多个绑定

系统会为每个绑定创建一个频道绑定接口，包括 BONDING\_OPTS 指令。使用这个配置方法可让多个绑定设备使用不同的配置。请按照以下操作创建多个频道绑定接口：

* + 创建多个 ifcfg-bondN 文件，文件中包含 BONDING\_OPTS 指令，让网络脚本根据需要创建绑定接口。
	+ 创建或编辑要绑定的现有接口配置文件，添加 SLAVE 指令。
	+ 使用 MASTER 指令工具在频道绑定接口中分配要绑定的接口，即从属接口。

以下是频道绑定接口配置文件示例：DEVICE=bondNNAME=bondNTYPE=BondBONDING\_MASTER=yesIPADDR=192.168.1.1PREFIX=24ONBOOT=yesBOOTPROTO=noneBONDING\_OPTS="bonding parameters separated by spaces"在这个示例中，使用绑定接口的号码替换 N。例如要创建两个接口，则需要使用正确的 IP 地址创建两个配置文件 ifcfg-bond0 和 ifcfg-bond1。

## 7.3.3 使用 nmtui 工具

在 nmtui 工具中，选择 Edit a connection，然后选择 Bond，按照提示输入相关信息，即可创建绑定。返回 nmtui 的主菜单，选择 Activate a connection，然后选择刚刚创建的绑定，即可激活绑定。

# 第八章 使用LVM管理硬盘

## 8.1 LVM简介

LVM是逻辑卷管理（Logical Volume Manager）的简称，它是Linux环境下对磁盘分区进行管理的一种机制。LVM通过在硬盘和文件系统之间添加一个逻辑层，来为文件系统屏蔽下层硬盘分区布局，提高硬盘分区管理的灵活性。使用LVM管理硬盘的基本过程如下：

* 将硬盘创建为物理卷。
* 将多个物理卷组合成卷组。
* 在卷组中创建逻辑卷。
* 在逻辑卷之上创建文件系统。

通过LVM管理硬盘之后，文件系统不再受限于硬盘的大小，可以分布在多个硬盘上，也可以动态扩容。

## 8.2 安装

* 清除缓存：



* 创建缓存：



* 在root权限下安装lvm：



* 查看安装后的rpm包：



## 8.3 管理物理卷

* 创建物理卷



* 查看物理卷



* 修改物理卷属性



* 删除物理卷



## 8.4 管理卷组

* 创建卷组



* 查看卷组



* 修改卷组属性



* 扩展卷组



* 收缩卷组



* 删除卷组



## 8.5 管理逻辑卷

* 创建逻辑卷



* 查看逻辑卷



* 调整逻辑卷大小



* 扩展逻辑卷



* 收缩逻辑卷



* 删除逻辑卷



## 8.6 创建并挂载文件系统

* 创建文件系统



* 手动挂载文件系统

手动挂载的文件系统仅在当时有效，一旦操作系统重启则会不存在。可在root权限下通过mount命令挂载文件系统。

其中：

* + lvname：指定要挂载文件系统的逻辑卷设备文件名。
	+ mntpath：挂载路径。

示例：将逻辑卷/dev/vg1/lv1挂载到/mnt/data目录。

* 自动挂载文件系统

手动挂载的文件系统在操作系统重启之后会不存在，需要重新手动挂载文件系统。但若在手动挂载文件系统后在root权限下进行如下设置，可以实现操作系统重启后文件系统自动挂载文件系统。

1. 执行blkid命令查询逻辑卷的UUID，逻辑卷以/dev/vg1/lv1为例。



查看打印信息，打印信息中包含如下内容，其中 uuidnumber 是一串数字，为UUID， fstype 为文件系统。/dev/vg1/lv1: UUID=" uuidnumber " TYPE=" fstype "2. 执行vi /etc/fstab命令编辑fstab文件，并在最后加上如下内容。



3. 验证自动挂载功能。

执行umount命令卸载文件系统，逻辑卷以/dev/vg1/lv1为例。



执行如下命令，将/etc/fstab文件所有内容重新加载。



执行如下命令，查询文件系统挂载信息，挂载目录以/mnt/data为例。

