新支点电信级服务器操作系统 V6

NewStart CGSL V6

系统用户手册

编写日期	2020 年 4 月 20 日	文件版本	V1.0
文件编排	封面 1 页，目录 6 页，内容 214 页，合计 224 页		

广东中兴新支点技术有限公司

手册说明

本手册是新支点电信级服务器操作系统（以下简称 NewStart CGSL）V6 系列版本全面使用说明书，讨论了进行系统管理所需的基础知识及相关系统管理主题，能够帮助您顺利执行系统管理任务并配置和管理一个高效、安全、稳定的服务器系统。

内容介绍

章 名	概 要
第 1 章 基本命令介绍	介绍在 CGSL 系统中进行 Shell 操作的基本知识，以及系统基本命令的使用。
第 2 章 系统安装与升级	介绍 CGSL 系统如何安装和升级。
第 3 章 用户和群组管理	如何在命令行界面下完成用户账号、工作组的建立和维护。
第 4 章 文件系统管理	对 CGSL 系统的文件系统如何进行管理进行介绍。
第 5 章 软件包管理	介绍如何使用 Shell 命令安装和管理系统中的应用程序和软件包。
第 6 章 使用 Vim 编辑器	介绍如何使用 Vim 编辑器。
第 7 章 基础系统管理	介绍 CGSL 系统管理基本知识。
第 8 章 系统服务	介绍 CGSL 服务的配置与使用。
第 9 章 系统安全	介绍 CGSL 系统安全管理策略。
第 10 章 网络	介绍 CGSL 网络管理的基本配置。
第 11 章 图形环境	介绍 CGSL 系统图形环境管理的基本配置。
第 12 章 自研工具	介绍 CGSL 自研工具的使用。
第 13 章 版权说明	详细介绍 GPL 版权申明与免责申明。

版本更新说明

文档版本	发布日期	备 注
V1.0	2020 年 4 月 20 日	更新软件版本

本书约定

介绍符号的约定、键盘操作约定、鼠标操作约定以及四类标志。

1) 符号约定

- 样式按钮表示按钮名；带方括号“【属性栏】”表示人机界面、菜单条、数据表和字段名等；
- 多级菜单用“->”隔开，如【文件】->【新建】->【工程】 表示【文件】菜单下的【新建】子菜单下的【工程】菜单项；
- 尖括号<路径>表示当前目录中 include 目录下的.h 头文件,如 **<asm-m68k/io.h>**表示**/include/asm-m68k/io.h** 文件。

2) 术语约定

- NewStart CGS Linux 使用 CGSL 代替

3) 标志

本书采用二个醒目标志来表示在操作过程中应该特别注意的地方：

♣ 警告：提醒操作中的一些注意事项。

提示：介绍提高效率的一些方法。

联系方式

电　　话：　400-033-0108

电子信箱：　os@gd-linux.com

公司地址：　广州市天河区科技园高唐软件园基地高普路 1021 号 6 楼

邮　　编：　510663

目录

第 1 章

基本命令介绍

熟悉在命令行界面下工作对使用和管理 CGSL 操作系统提供了极大的方便，本章介绍在 CGSL 系统中进行 Shell 操作的基本知识。

1.1 基础知识

主要介绍关于 CGSL 系统中 Shell 及文件和目录的基础知识。

1.1.1 文件命令

CGSL 下文件名的最大长度可以是 256 个字符，通常由字母、数字、“.”（点号）、

“_”（下划线）和“-”（减号）组成。文件名中不能含有“/”符号，因为“/”在 Linux 目录树中表示根目录或路径中的分隔符（如同 DOS 中的“\”）。

Linux 系统中支持文件名中的通配符，具体如下：

星号（*）：匹配零个或多个字符；

问号（？）：匹配任何一个字符；

[ab1A-F]：匹配任何一个列举在集合中的字符。本例中，该集合是 a、b、l 或任何一个从 A 到 F 的大写字符。

1.1.2 路径

操作系统查找文件所经过的路径称为路径名。使用当前目录下的文件时可以直接引用文件名；如果要使用其他目录下的文件，就必须指明该文件在哪个目录之中。

按查找文件的起点不同可以分为两种路径：绝对路径和相对路径。从根目录开始的路径称为绝对路径，从当前所在目录开始的路径称为相对路径。

与 DOS 相同，每个目录下都有代表当前目录的“.”文件和代表当前目录父目录的“..”文件，相对路径名一般就是从“..”开始。

提示：在 Linux 目录树中，表示根目录或者路径中的分隔符是“/”。

1.1.3 文件类型

CGSL 系统支持以下文件类型：普通文件、目录文件、设备文件、命名管道文件、套接字文件以及符号链接文件。

- 普通文件：包括文本文件、数据文件、可执行的二进制程序等。
- 目录文件：简称目录，CGSL 中把目录看成是一种特殊的文件，利用它构成文件系统的分层树型结构。每个目录文件中至少包括两个文件，“..”表示上一级目录，“.”表示该目录本身。
- 设备文件：设备文件是一种特别文件，CGSL 系统利用它们来标识各个设备驱动器，内核使用它们与硬件设备通信。有两类特别设备文件：字符设备和块设备。
- 命名管道文件：用于系统进程通信的文件。
- 套接字文件：套件字文件类似于命名管道文件，用于网络通讯之间进行通信的文件。
- 符号链接：一种特殊文件，它们存放的数据是文件系统中通向某个文件的路径。当使用符号链接文件时，系统自动地访问所保存的这个路径。

1.1.4 目录结构

通过对系统目录组织结构的了解，可以在进行文件操作和系统管理时方便地知道所要的东西在什么地方。

CGSL 的文件系统采用分层的树形目录结构。即在一个根目录（通常用“/ ”表示）,含有多个下级子目录或文件；子目录中又可含有更下级的子目录或者文件的信息，这样一层一层地延伸下去，构成一棵倒置的树。树中的“根”与“杈”代表的是目录或称为文件夹，而“叶子”则是一个个的文件。

下面列出了主要的系统目录及其简单描述：

/bin ：存放普通用户可以使用的命令文件。目录/usr /bin 也可用来贮存用户命令。

/sbin ：一般用于存放非普通用户使用的命令（有时普通用户也可能会用到）。目录/usr/sbin 中也包括了许多系统命令。

/etc ：系统的配置文件。

/root ：系统管理员（root 或超级用户）的主目录。

/usr ：包括与系统用户直接相关的文件和目录，一些主要的应用程序也保存在该目录下。

/home：用户主目录的缺省位置，保存了用户文件（用户自己的配置文件、文档、数据等）。

/dev ：设备文件。在 CGSL 中设备以文件形式表现，从而可以按照操作文件的方式简便地对设备进行操作。

/mnt ：文件系统的缺省挂载点。一般用于安装移动介质、其它文件系统（如 DOS）的分区、网络共享文件系统或任何可安装文件系统。

/lib ：包含许多由/bin 和/sbin 中的程序使用的共享库文件。目录/usr/lib/中含有更多用于用户程序的库文件。

/boot ：包括内核和其它系统启动时使用的文件。

/var ：包含一些经常改变的文件。例如假脱机（spool）目录、文件日志目录、锁文件、临时文件等等。

/proc ：操作系统的内存映像文件系统，是一个虚拟的文件系统（没有占用磁盘空间）。当您查看它们时，看到的是内存里的信息，这些文件有助于了解系统内部信息。

/opt ：存放可选择安装的文件和程序。主要由第三方开发者用于安装和卸装他们的软件包。

/tmp ：用户和程序的临时目录。

/lost + found ：在系统修复过程中恢复的文件。

1.1.5 Shell 简介

用户在命令行下工作时，不是直接同操作系统内核打交道，而是由命令解释器接受命令，分析后再传给相关的程序。进入 CGSL 环境时系统将自动启动相应的 Shell，Shell 是一种命令行解释程序，它提供用户与操作系统之间的接口。CGSL 下默认的 Shell 是 bash。bash 命令的基本格式如下：

命令名【选项】【参数 1】【参数 2】...

其中方括号括起的部分表明该项对命令而言是可选的。

【选项】：对命令有特别定义，一般以“-”开始，多个选项可用一个“-”连起来，如 ls -l -a 与 ls -la 相同。

【参数】：提供命令运行的信息，或者是命令执行过程中所使用的文件名。

提示：输入用户名、口令、命令名与文件名时一定要区分大小写，因为大小写字母在 Linux 系统中代表不同的含义。

提示：在命令、选项和参数之间要用空格隔开，连续的空格会被 Shell 解释为单个空格。

键入命令

在 Shell 提示符下输入相应的命令，然后按回车键确认，Shell 会读取该命令并执行。如果系统找不到输入的命令，会显示：“command not found”，这时需要检查键入命令的拼写及大小写是否正确。

使用分号（；）可以将两个命令隔开，这样可以实现在一行中输入多个命令。命令的执行顺序和输入的顺序保持一致。

命令补齐

当要输入的命令目录很深或命令中的文件名很长时，只要按一下<Tab>键，系统会在可能的命令或文件名中找到相匹配的项，自动帮您补齐。如果有一个以上的文件符合输入的字符串，不能补齐时，可以按两下<Tab>键，系统将把所有符合的文件名列出来。

历史记录

Shell 会把过去输入过的命令记忆下来，只要按上下方向键，就可以选择以前输入过的命令了。

有了以上基础，可以运行下面列出的几个简单命令来实际使用一下：

clear ：刷新屏幕；

date ：在屏幕上显示日期和时间；

echo ：将字符或字符串回显到标准输出上；

cal ：显示月份和日历。

1.1.6 系统帮助

CGSL 具有强大的系统和网络功能，数量众多的实用工具软件和大量复杂的操作命令。

为了帮助用户顺利进行操作，系统提供了多种多样的联机帮助信息以便用户随时查询。

联机手册

通过 man 命令使用联机用户手册，系统可以显示任何命令的联机帮助信息。man 命令的语法格式为：

```
#man [ [ [-c ] [-t ] [Section] ] | [-k | -f ] ] [-F] [-m] [ -MPath ] [ -r ] [ -a ] Title
```

1. 其中选项及意义如下：

-a 显示所有匹配项。

-c 显示使用 cat 命令的手册信息。

-f 显示在关键字数据库中仅与作为最终参数给定的命令名相关的项。可以输入多个命令名，中间用空格隔开。

-F 只显示首个匹配项。

-k 显示关键字数据库中包含与作为最终参数给定的字符匹配的标题的字符串的每一行。

-m 只在 MANPATH 或 -M 中指定的路径中搜索。

-M Path 更改 man 命令搜索手册信息的标准位置。路径是用冒号隔开的路径的列表，其中，可以使用以下特殊符号：

%D 联机帮助页的缺省 AIX® 路径。

%L 与当前语言环境的 LC_MESSAGES 类别相对应的特定于语言环境的目录位置。

%L 与当前 LC_MESSAGES 类别的首 2 个字符相对应的特定于语言环境的目录位置。

-r 手册信息的远程搜索。如果出于某个原因，远程搜索失败，则 man 将执行本地搜索以获取请求的联机帮助页。

-t 使用 troff 命令格式化手册信息。如果在超文本信息基中查找到手册页面，则忽略该标志。

如果为 Section 参数指定一个段，则 man 命令在手册页面的该段中搜索 Title 参数指定的标题。

Section 参数的值可以是 1 到 8 的阿拉伯数字或字母。

1 表示用户命令和守护程序。

2 表示系统调用和内核服务。

3 表示子例程。

4 表示特殊文件、设备驱动程序和硬件。

5 表示配置文件。

6 表示游戏。

7 表示杂项命令。

8 表示管理命令和守护程序。

2. man 的一些常用参数和用法：

```
#man -a cmd
```

打开所有领域内的同名帮助，例如 man -a fam ，您首先会进入一个 fam(1M)的命令版 fam 帮助，您再按 q 键

就会进入 FAM(3X)，库函数版的帮助

```
#man -aw cmd
```

显示所有 cmd 的所有手册文件的路径，如 man -aw fam 就是

```
/usr/share/man/man1/fam.1m.gz
/usr/share/man/man3/fam.3x.gz
```

man 领域代号 cmd

直接指定特定领域内搜索手册页，如 man 3 fam 直接进入库函数版的帮助

指定手册文件的搜索路径，如 man -M /home/mysql/man mysql 显示的就是您安装的 mysql 的帮助， 而不是系统自带的旧版 mysql 的帮助

```
#man -M cmd
```

也可以将内容重定向到一个文本文件内：

```
#man cmd | col -b > cmd.txt
```

新开一个 shell 窗口（或在原窗口 shell> LANG=en_US.UTF-8)

```
shell>man /usr/share/man/zh/man1/mplayer.1.gz
```

man 的配置文件/etc/man.config

如果您不想每次 man cmd 都要用-M 指定路径，那么可以通过修改配置文件，添加内容如

```
MANPATH /home/mysql/man
```

man 在各领域的搜索次序可以通过修改 ，但一般不推荐修改。

```
MANSECT 1:8:2:3:4:5:6:7:9:tcl:n:l:p:o
```

常用的 Linux 系统帮助手册按章节分类，位于/usr/share/man 目录下。

3. 应用实例

要显示关于 grep 命令的信息，请输入：

```
#man grep
```

要显示关于 rpc_$register 库例程的信息，请输入：

```
#man rpc_\$register
```

要显示包含“mkdir”字符串的所有项(等同于 apropos 命令)，请输入：

```
#man -k mkdir
```

要显示/usr/share/man 或 /usr/share/man/local 路径中的所有与 ftp 命令相关的物件，请输入：

```
#man -M /usr/share/man:/usr/share/man/local ftp
```

要显示所有匹配项，输入以下命令：

```
#man -a Title
```

只要显示首个匹配项，输入以下命令：

```
#man -F Title
```

只要在 MANPATH 或 -M 中指定的路径中搜索，输入以下命令：

```
#man -m -M PATH Title
```

要在用户定义的 PATH 中搜索，输入以下命令：

```
#man -M PATH Title
```

1.2 目录操作命令

1.2.1 查看目录

查看目录内容的命令是 *ls* ，它默认显示当前目录的内容，可以在命令行参数的位置给出一个或多个目录名，从而可以查看这些目录。命令的语法格式为：

```
#ls [OPTION]... [FILE]...
```

ls 命令常用的选项如下：

-a ：列出所有文件，包括那些以“.”开头的文件；

-d ：如果后面接的是一个目录，那么使用该参数只输出该目录的名称；

-l ：使用长格式显示文件条目，包括连接数目、所有者、大小、最后修改时间、权限等；

其它选项及相关详细说明请参见 ls 命令的 man 手册：man ls。

1.2.2 改变工作目录

进入一个目录，或者说改变当前工作目录使用 cd 命令，其命令的语法格式为：

```
#cd DIRECTORY
```

cd 命令带有唯一的一个参数，即表示目标目录的路径名（相对路径名或绝对路径名）。

利用点点（..）把工作目录向上移动一级目录：cd..

为了从系统中的任何地方返回到用户主目录，可以使用不带任何参数的 cd 命令。

1.2.3 创建目录

使用 mkdir 命令创建一个目录或多个目录。其命令的语法格式为：

```
#mkdir [OPTION] DIRECTORY
```

mkdir 命令常用的选项如下：

-p：当指定目录存在时不会报错，并根据需要一并创建其父目录。

```
#mkdir -p DIRECTORY/SUBDIRECTORY
```

其它选项及相关详细说明请参见 mkdir 命令的 man 手册：man mkdir

1.2.4 删除目录

rmdir 命令从目录中删除一个或多个空的子目录，语法格式如下：

```
#rmdir [OPTION]... DIRECTORY...
```

子目录被删除之前应该是空目录，如果该目录中仍有其它文件，那么就不能用 rmdir 命

令把它删除。当前的工作目录必须在被删除目录之上，不能是被删除的目录本身，也不能是被删除目录的子目录。

rmdir 命令常用的选项如下：

-p 选项：递归地删除指定的目录及其父目录。例如：

```
#rmdir -p a/b/c
```

将删除 a 目录，则相当于执行：

```
#rmdir a/b/c a/b a
```

提示：rmdir 命令只能删除空目录。使用 rm -r 命令可以删除非空目录，参见后续章节中的 rm 命令介绍。

1.2.5 显示当前目录

命令 pwd 可以显示用户当前在目录树中的位置。如：

```
#pwd
/root
```

表示用户当前所在的目录是/root。

1.3 文件操作命令

1.3.1 显示文本文件

CGSL 系统中，如下的命令常用以显示文本文件。

■ **cat 命令**

cat 命令把文件串连接后传到标准输出（通常是屏幕）上显示出来。该命令的一般语法是：

```
#cat [OPTION] [FILE]...
```

常用选项如下：

-n ： 显示输出行的编号。

-b ： 只对非空输出行进行编号。

其它选项及相关详细说明请参见该命令的 man 手册：man cat

■ **more 命令**

more 命令显示文件内容，每次显示一屏。其语法是：

```
#more [OPTION]... [FILE]...
```

可在每个屏幕的底部出现一个提示信息，给出至今已显示的该文件的百分比。

如下交互式命令可控制显示结果：

- 按<space>键，显示文本的下一屏内容。
- 按<Enter>键，只显示文本的下一行内容。
- 按斜线符（/），接着输入一个模式，可以在文本中寻找下一个相匹配的模式。
- 按 h 键，显示帮助屏，该屏上有相关的帮助信息。
- 按 b 键，显示上一屏内容。
- 按 q 键，退出 more 命令。

相关其它详细信息请参见相应的 man 手册：man more

■ **less 命令**

less 命令显示文件内容，其语法是：

```
#less [OPTION]...[FILE]...
```

less 命令用于控制显示结果的交互式命令与 vi 编辑器中的交互式命令一致，详细请参见 vim 实用程序相关章节的介绍。

相关其它详细信息请参见相应的 man 手册：man less

■ **head 命令**

其命令语法如下：

```
#head [OPTION]...[FILE]...
```

head 命令在屏幕上显示指定文件前多少行和前多少个字节等，这是由-c 或者-n 选项决定的。

相关其它详细信息请参见相应的 man 手册：man head

■ **tail 命令**

其命令的语法如下：

```
#tail [OPTION]...[FILE]...
```

在屏幕上显示指定文件末尾的若干行或若干字节，这是由-c 或者-n 选项决定的；或者从指定行号开始显示，直至该文件的末尾。

相关其它详细信息请参见相应的 man 手册：man tail

1.3.2 更新文件访问和修改时间

可以利用该命令更新对文件的访问和修改时间，且可以用来创建空文件。其语法如下：

```
#touch [OPTION]... [FILE]...
```

不存在的文件名被当作空文件创建。已存在文件的时间标签会更新为当前的时间（默认方式），而数据将原封不动地保留下来。

相关其它详细信息请参见相应的 man 手册：man touch

1.3.3 拷贝文件

使用 cp 命令拷贝文件。可以使用 cp 命令把一个源文件拷贝到一个目标文件，或者把一系列文件拷贝到一个目标目录中。其语法是：

```
#cp [OPTION]...SOURSE DEST
```

如果目标文件是目录文件，那么把源文件拷贝到这个目录中，而文件名保持不变；如果目标文件不是目录文件，那么源文件就拷贝到该目标文件中，后者原有的内容将被破坏，但文件名不变。

常用选项如下：

-r，-R ：拷贝目录。

-a ：保持源文件的所有属性。

其它选项及相关详细说明请参见该命令的 man 手册：man cp

1.3.4 移动和重命名文件

mv 命令用来移动文件或对文件重命名。该命令的语法为：

```
#mv [OPTION] SOURSE DEST
```

下表是源和目标分别为文件或目录时的简单说明：

源文件	目标文件	mv 命令作用
文件	文件	将源文件重命名为目标文件名
文件	目录	将源文件移动到目标目录中
目录	目录(存在)	将源目录移动到目标目录中
目录	目录(不存在)	将源目录重命名为目标目录名

1.3.5 删除文件

用 rm 命令删除不需要的文件和目录。该命令的语法为：

```
#rm [OPTION]... FILE...
```

常用选项如下：

-r 选项：可以删除目录。当一个目录被删除时，所有文件和子目录都将被删除。

其它选项及相关详细说明请参见该命令的 man 手册：man rm

♣ 警告：这是个非常危险的命令，需慎用！

1.3.6 文件链接

CGSL 操作系统具有为一个文件起多个名字的功能，称为链接。这样只要对一个文件修改，就可以完成对所有目录下相应链接文件的修改。

ln 命令用来创建链接，常用语法为：

```
#ln [OPTION]... [-T] TARGET LINK_NAME
#ln [OPTION]...TARGET... DIRECTORY
```

其中 TARGET 为被链接的目标文件，LINK_NAME 为链接文件名，DIRECTORY 为链接文件被存放的目录。

文件链接有两种形式，即硬链接和符号链接。

■ 硬链接

默认情况下，ln 命令创建硬链接。

一个文件的硬链接数可以在目录的长列表格式的第二列中看到，无额外链接的文件链接数为 1。ln 命令会增加链接数，rm 命令会减少链接数。一个文件除非链接数为 0，否则不会物理地从文件系统中被删除。

对硬链接有如下限制：不能对目录文件作硬链接，不能在不同的文件系统之间作硬链接。

■ 符号链接

符号链接也称软链接，是将一个路径名链接到一个文件，事实上，它只是一个文本文件，其中包含它提供链接的另一个文件的路径名。另一个文件是实际包含所有数据的文件。所有读写文件内容的命令，当它们被用于符号链接时，将沿着链接方向前进去访问实际的文件。

如果给 ln 命令加上 s 选项，则建立符号链接。如下命令创建到 target 文件的符号链接文件 link：

```
#ln -s target link
```

符号链接没有硬链接的限制，可以对目录文件作符号链接，也可以在不同文件系统之间作符号链接。

相关详细说明请参见该命令的 man 手册：man ln

1.3.7 文件内容比较

■ 比较文本文件

diff 命令用于比较文本文件，并显示两个文件的不同。其一般格式是：

```
#diff [options] from-file to-file
```

如果两个文件完全一样，则不显示任何输出。如果有区别，就会分段显示两个文件的区别。

相关详细说明请参见该命令的 man 手册：man diff

■ **比较数据文件**

cmp 命令比较任何两个包含正文或数据的普通文件。其一般语法为：

```
#cmp [-l | -s] file1 file2 [skip1 [skip2]]
```

由于二进制数据不能显示到屏幕上，cmp 命令只是简单的报告从哪一个字节开始出现不同。

相关详细说明请参见该命令的 man 手册：man cmp

1.3.8 查找文件

■ **find 命令**

find 命令用来查找文件和目录的位置。该命令的语法为：

```
#find [-H] [-L] [-P] [path…] [expression]
```

其中，常用的选项有：

find 命令的最基本的用法就是列出指定目录下的所有文件和子目录：

```
#find /usr
```

-name ：按文件名查找。

如：下面的命令将查找/usr 目录下名称为 linux 的目录和文件：

```
#find /usr –name ‘linux’
```

-size ：按文件大小查找。

例如，下面的命令将查找/usr 目录下等于 100k 的文件：

```
#find /usr -size 100k
```

-user ：按文件主查找。

-type ：按文件类型查找。常见的类型有：

b 块特别文件

C 字符特别文件

f 普通文件

l 符号链接文件

d 目录文件

其它选项及相关详细说明请参见该命令的 man 手册：man find

■ **locate 命令**

locate 是一个使用方便且查询速度极快的文件和目录查找命令。该命令的语法为：

```
#locate [OPTION]... PATTERN...
```

使用 locate 命令的前提是要先创建一个用于定位文件或目录位置的 mlocate 数据库，而且该数据库应是时时更新的，这样才可以保证 locate 查找结果的准确性。

updatedb 命令用户创建和更新 mlocate 数据库，需要以 root 用户身份执行此命令。

```
#updatedb
```

数据库创建后就可以查找文件了。例如，要查找所有关于 telnet 命令的文件。可以使用：

```
#locate telnet
```

locate 命令将在其数据库中检查所有匹配于 telnet 的文件和目录并在屏幕上显示结果。相关详细说明请参见该命令的 man 手册：man locate。

1.3.9 在文件中查找文本

grep 命令用来在文本文件中查找指定模式文本，并在标准输出上显示包括给定文本所有行。grep 命令的语法为：

```
#grep [options] PATTERN [FILE...]
```

常用选项如下：

-i 选项：匹配文本时不区分大小写。

-r 选项：在目录及其子目录下的所有文件中查找。

例如，下面的命令将在/etc 目录及其子目录下的所有文件中查找所有包含“hello word”文本的行，且不区分大小写：

```
#grep -ri ‘hello world’ /etc
```

其它选项及相关详细说明请参见该命令的 man 手册：man grep

1.4 文件权限操作

在多用户操作系统中，出于安全性的考虑，需要给每个文件和目录加上访问权限，严格地规定每个用户的权限。同时，用户可以为自己的文件赋予适当的权限，以保证他人不能修改和访问。

1.4.1 改变文件主

Linux 为每个文件都分配了一个文件所有者，称为文件主，对文件的控制取决于文件主或超级用户（root）。文件或目录的创建者对创建的文件或目录拥有特别使用权。

文件的所有关系是可以改变的，chown 命令用来更改某个文件或目录的所有权。chown 命令的语法格式是：

```
#chown [OPTION]... [OWNER][:[GROUP]] FILE...
```

用户可以是用户名或用户 ID。文件是以空格分开的要改变权限的文件列表，可以用通配符表示文件名。

如下命令将/home/test 文件的文件主修改为 root 用户。

```
#chown root /home/test
```

如果改变了文件或目录的所有权，原文件主将不再拥有该文件或目录的权限。

1.4.2 改变文件用户组

在 Linux 下，每个文件又同时属于一个用户组。当创建一个文件或目录，系统会赋予它一个用户组关系，用户组的所有成员都可以使用此文件或目录。

文件用户组关系的标志是 GID 。文件的 GID 只能由文件主或超级用户（root）来修改。chgrp 命令可以改变文件的 GID ，其语法格式为：

```
#chgrp [OPTION]...GROUP FILE...
```

其中 group 是用户组 ID。文件名是以空格分开的要改变属组的文件列表，它支持通配符。

1.4.3 文件权限设置

Linux 系统中的每个文件和目录都有访问许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。

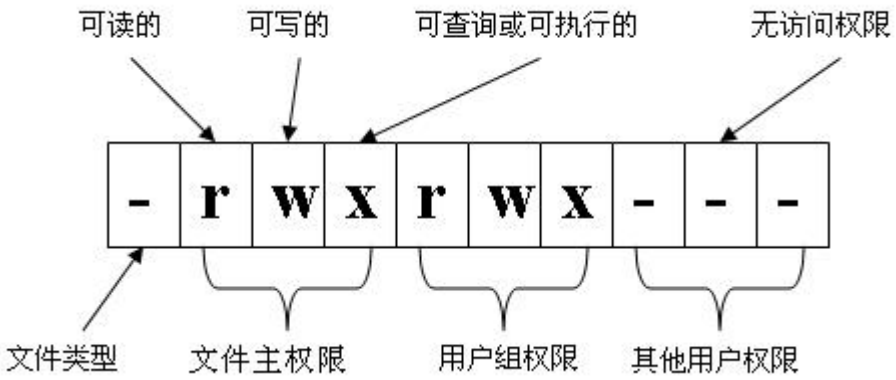
访问权限规定三种不同类型的用户：

- 文件主（owner)
- 同组用户（group)
- 可以访问系统的其他用户（others)

访问权限规定三种访问文件或目录的方式：

- 读（r)
- 写（w)
- 可执行或查找（x)

当用 ***ls -l*** 命令或 ***l*** 命令显示文件或目录的详细信息时，最左边的一列为文件的访问权限。其中各位的含义如下：



■ **文件访问权限**

- 读权限（r）

只允许指定用户读其内容，而禁止对其做任何的更改操作。将所访问的文件的内容作为输入的命令都需要有读的权限。例如：cat、more 等。

- 写权限（w）

允许指定用户打开并修改文件。例如命令 vi、cp 等。

➢ 执行权限（x）

指定用户将该文件作为一个程序执行。

■ **目录访问权限**

➢ 读权限（r）

可以列出存储在该目录下的文件，即读目录内容列表。这一权限允许 Shell 使用文件扩展名字符列出相匹配的文件名。

➢ 写权限（w）

允许从目录中删除或添加新的文件，通常只有目录主才有写权限。

➢ 执行权限（x）

允许在目录中查找，并能用 cd 命令将工作目录改到该目录。

1.4.4 改变文件权限

1.4.4.1 以符号模式改变权限

chmod 用于改变文件或目录的访问权限。用户可以用它控制文件或目录的访问权限。只有文件主或超级用户 root 才有权用 chmod 改变文件或目录的访问权限。

chmod 命令的语法为：

```
#chmod [OPTION]... key FILE...
```

key 由以下各项组成：

【who】 【操作符号】 【mode】

其中，操作对象 who 可以是下述字母中的任一个或者它们的组合：

u user ,表示用户，即文件或目录的所有者。

g group ,表示同组用户，即与文件属主有相同组 ID 的所有用户。

o others ,表示其他用户。

a all ,表示所有用户，它是系统默认值。

操作符号可以是：

+ 添加某个权限

- 取消某个权限

= 赋予给定权限并取消其他所有权限（如果有的话）

mode 所表示的权限可用相关权限标识的任意组合，常用的权限标识有：

r 可读

w 可写

x 可执行

s 在文件执行时把进程的属主或组 ID 置为该文件的文件属主

u 与文件属主拥有一样的权限

g 与和文件属主同组的用户拥有一样的权限

o 与其他用户拥有一样的权限

这三部分必须按顺序输入。可以用多个 key，但必须以逗号间隔。

如下命令添加/home/testfile 文件的同组用户的可执行权限：

```
#chmod g+x /home/testfile
```

1.4.4.2 以绝对方式改变权限

通常也可以用 ***chmod*** 命令配以不同类型的 key 直接设置权限。这时以数字代表不同的权限。这里 key 可以包括三个（或三个以上）的数字，每个数字表示对不同类型用户的权限。

数字表示的属性的含义：

0 表示禁止该权限，1 表示可执行权限，2 表示可写权限，4 表示可读权限，然后将其相加。所以数字属性的格式应为 3 个从 0 到 7 的八进制数，其顺序是（u）（g）（o）。

通常，key 是以三位八进制数字出现的，第一位表示用户权限，第二位表示组权限，第三位表示其他用户权限。

例如，要使文件 myfile 的文件主和同组用户具有读写权限，但其他用户只可读，可以用以下命令指定权限：

```
#chmod 664 myfile
```

1.4.5 默认权限

默认情况下，系统将创建的普通文件的权限设置为-rw-r-r--，即文件主对该文件可读可写（rw），而同组用户和其他用户都只可读；同样，在默认配置中，将每一个用户主目录的权限都设置为 drwx------，即只有文件主对该目录可读、写和可查询（rwx），即用户不能读其他用户目录中的内容。

用户可以修改新建文件的默认存取权限，如使用如下命令：

```
#umask u = rwx , g = , o =
```

它会在创建新文件时给文件主以全部权限，而同组用户及其他用户没有任何权限。

1.5 重定向和管道

执行一个 Shell 命令行通常会自动打开三个标准文件，即标准输入文件（stdin），通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），这两个文件通常都对应终端的屏幕。进程从标准输入文件中得到数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

下面以 cat 命令为例，cat 命令的功能是从命令行给出的文件中读取数据，并将这些数据直接送到标准输出。例如，使用以下命令：

```
#cat config
```

将会把文件 config 的内容依次显示到屏幕上。但是，如果 cat 的命令行中没有参数，它就会从标准输入中读取数据，并将其送到标准输出。例如：

```
#cat
Hello world
Hello world
```

直接使用标准输入/输出文件存在以下问题：

1、数据从标准终端输入时，输入的数据只能用一次，下次再想用这些数据时就需要重新输入；而且在终端上输入时，若输入有误修改起来也不方便；

2、输出到终端屏幕上的信息只能看不能修改。用户无法对输出的内容进行更多处理，如将输出作为另一命令的输入进行进一步的处理等。

为了解决上述问题，Linux 系统为输入、输出的传送引入了另外两种机制，即输入/输

出重定向和管道。

1.5.1 输入重定向

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以不是来自于键盘，而来自一个指定的文件。

例如，命令 ***wc*** 统计指定文件包含的行数、单词数和字符数。如果仅在命令行上键入：

```
#wc
```

wc 将等待用户的输入，从键盘键入的所有文本都出现在屏幕上，但并没有结果，直至按下<Ctrl + D> ，***wc*** 才将命令结果写在屏幕上。

如果给出一个文件名作为 ***wc*** 命令的参数，***wc*** 将返回该文件所包含的行数、单词数和字符数。

另一种把/etc/passwd 文件内容传给 ***wc*** 命令的方法是重定向 ***wc*** 的输入。输入重定向的一般形式为：

```
#command < filename
```

可以用下面的命令把 ***wc*** 命令的输入重定向为/etc/passwd 文件：

```
wc < /etc/passwd
20 23 726
```

大多数命令都以参数的形式在命令行指定输入文件的文件名，所以输入重定向并不经常使用。尽管如此，当使用一个不接受文件名作为输入参数的命令，或需要的输入内容存在于一个文件里时，就能用输入重定向解决问题。

1.5.2 输出重定向

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

输出重定向比输入重定向更常用。例如，如果某个命令的输出很多，在屏幕上不能完全显示，那么将输出重定向到一个文件中，然后再用文本编辑器打开这个文件，就可以查看输出信息；如果想保存一个命令的输出，也可以使用这种方法。还有，输出重定向可以用于把一个命令的输出当作另一个命令的输入。

输出重定向的一般形式为：

```
#command > filename
```

例如：

```
#ls > directory.out
#cat directory.out
chl.doc ch2.doc ch3.doc chimp config mail /test/
```

将 ls 命令的输出保存为一个名为 directoryout 的文件。

♣ 提示：如果>符号后的文件已存在，那么这个文件将被覆盖。

为避免输出重定向中指定文件只能存放当前命令的输出重定向的内容，Shell 提供了输出重定向的一种追加手段。

输出追加重定向与输出重定向的非常相似，区别仅在于输出追加重定向的功能是把命令（或可执行程序）的输出结果追加到指定文件的最后，而该文件原有内容不被破坏。

如果要将一条命令的输出结果追加到指定文件的后面，可以使用追加重定向操作符“> >”。形式为：

```
#command >> filename
```

例如：

```
#ls *.doc > > directory.out
#cat directory.out
chl.doc chZdoc ch3doc chimp config mail / test /
chl.doc ch2.doc ch3.doc
```

和程序的标准输出重定向一样，程序的错误输出也可以重新定向。使用符号 2>（或追加符号 2> >）表示对错误输出设备重定向。例如下面的命令：

```
#ls /usr/tmp 2> err.file
```

可在屏幕上看到程序的正常输出结果，但又将程序的任何错误信息送到文件 err.file 中，以备将来检查用。

还可以使用另一个输出重定向操作符（& >）将标准输出和错误输出同时送到同一文件中。例如：

```
#ls /usr/tmp &> output.file
```

利用重定向将命令组合在一起，可实现系统单个命令不能提供的新功能。例如使用下面的命令序列，即统计了/usr/bin 目录下的文件个数。

```
#ls /usr/bin > /tmp/dir
#wc -w < /tmp/dir
459
```

1.5.3 管道

将一个程序或命令的输出作为另一个程序或命令的输入有两种方法，一种是通过一个临时文件将两个命令或程序结合在一起，例如上节例子中的/tmp/dir 文件将 ls 和 wc 命令联在一起；另一种是 Linux 所提供的管道功能，这种方法比前一种方法更方便。

管道可以把一系列命令连接起来，这意味着第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入，以此类推。显示在屏幕上的是管道行中最后一个命令的输出。

通过使用管道符“|”来建立一个管道行。用管道重写上面的例子：

```
#ls /usr/bin | wc -w
1789
```

再如：

```
#cat sample.txt | grep “High” | wc -l
```

管道将 ***cat*** 命令的输出送给 ***grep*** 命令。***grep*** 命令在输入里查找单词 High，***grep*** 命令的输出则是所有包含单词 High 的行，这个输出又被送给 ***wc*** 命令，***wc*** 命令统计出输入中的行数。假设 sample.txt 文件的内容如下：

```
Things to do today :
Low：Go grocery shopping
High：Return movie
High：Clear level 3 in Alien vs Predator
那么该管道行的结果是 2 。
```

1.6 进程与控制作业命令

1.6.1 用 ps 命令查看系统中的进程

可以用 ps 命令观察进程状态，它会把当前瞬间进程的状态显示出来。可以根据显示的信息确定哪个进程正在运行，某个进程是被挂起，还是遇到了某些困难，进程已运行了多久，进程正在使用的资源，进程的相对优先级，以及进程的标识号（PID）。这些信息对用户很有用，对于系统管理员来说更为重要。

ps 命令的一般用法是：

```
#ps [OPTION]
```

如果不带任何选项，ps 命令列出每个与您的当前 Shell 有关的进程的 PID。结果如下：

```
PID   TTY   TIME   CMD
596 pts / 0   00 : 00 : 00 bash
627 pts / 0   00 : 00 : 00 vi
628 pts / 0   00 : 00 : 00 ps
```

其中，各字段的含义如下：

- PID ：进程标识号
- TTY ：开始该进程的终端号
- TIME ：报告进程累计使用的 CPU 时间
- CMD ：正在执行的进程名

要获得一个完整的进程信息列表，常用带有下列选项的 ps 命令：

```
#ps aux
```

或

```
#ps -ef
```

它除了列出以上字段以外，还列出 CPU 使用率（% CPU），内存使用率（% MEM），虚拟映像大小（SIZE）、驻留数据集大小（RSS ）、终端号（TTY ）、状态（STAT）等字段。其它选项及相关详细说明请参见该命令的 man 手册：man ps

1.6.2 top 命令

top 命令用于读入计算机系统的信息，这些信息包括当前的系统数据和进程的状态等。输入 *top* 命令后，屏幕输出如下：

```
top - 11:18:09 up  2:06,  3 users,  load average: 0.01, 0.00, 0.00
Tasks: 179 total,   1 running, 178 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.6%us,  0.2%sy,  0.0%ni, 99.2%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    499316k total,   458732k used,    40584k free,    37420k buffers
Swap:  1015800k total,     5036k used,  1010764k free,    88392k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1842 root      20   0  163m  31m 6056 S  1.7  6.4   0:05.57 Xorg
 2567 root      20   0  336m  14m 8816 S  0.7  2.9   0:00.65 gnome-terminal
 3712 root      20   0 14940 1240  904 R  0.7  0.2   0:00.08 top
 3675 root      20   0 14940 1248  904 S  0.3  0.2   0:00.68 top
    1 root      20   0 19236 1288 1072 S  0.0  0.3   0:03.44 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.02 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.13 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.09 migration/1
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/1
    8 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    9 root      RT   0     0    0    0 S  0.0  0.0   0:00.08 migration/2
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/2
   11 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/2
   12 root      RT   0     0    0    0 S  0.0  0.0   0:00.11 migration/3
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/3
```

图 1-1 top 命令

图中的每一列显示了系统的详细信息，图中开头几行的信息含义如下：

Uptime：

显示当前时间、自上次启动系统开始过去的时间、激活用户的数目以及在过去 1、5 和 15 分钟之内的 CPU 平均占用情况。

Process：

显示了系统所有的进程，并把进程按挂起、运行、创建和停止分类。

CPU States:

统计被用户和系统占用的当前 CPU 的状态。

Mem:

统计当前内存的占用状态。

Swap:

统计了 swap 区域的占用情况。

在 ***top*** 命令中显示了进程的列表，其中包括的内容有：PID、用户、优先级、nice 参数、所需的内存信息（SIZE、RSS、SHARE）、状态（STAT）、CPU 占用的百分比、占用的内存信息、已用的训算机时间和各目的程序调用（COMMAND ）等。

其它选项及相关详细说明请参见该命令的 man 手册：man top

1.6.3 用 kill 命令终止进程

运行过程中，可能在某一时刻，系统中有的进程出现了问题，不能正常运行，但也不能正常退出。这时可以使用 ***kill*** 命令终止进程的执行，释放这些进程占用的系统资源，常用的 ***kill*** 命令的格式为：

```
#kill [-s signal | -p ] [-a] [--] pid...
#kill -l [signal]
```

命令的选项和参数的意义如下：

pid 给出了需要结束的进程的 PID，可以通过命令 ps 获得进程的 PID 。在命令 kill 中可以一次列出许多的进程 PID。

-s signal 是一个可选参数，用来给出发给进程的信号。在默认情况下，命令 kill 给进程发 TERM 信号，该信号将通知进程退出。如果进程不接收该信号，可以通过参数-9 强行结束进程。

-l 该参数要求 kill 命令列出它可以发给进程的所有信号。

其它选项及相关详细说明请参见该命令的 man 手册：man kill

1.7 基本网络命令

CGSL 具有强大的网络功能，提供了丰富的网络应用程序，完全支持 TCP/IP 协议。在网络环境下，可以进行远程注册、远程命令调用、传送文件等操作。本节介绍了几个基本的网络操作命令。

1.7.1 基本的网络配置命令

1.7.1.1 ifconfig

可以使用 ifconfig 命令来配置并查看网络接口的配置情况。

例如：

1. 配置 eth0 的 IP 地址，同时激活该设备。

```
#ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up
```

2. 配置 eth0 别名设备 eth0:1 的 IP 地址，并添加路由。

```
#ifconfig eth0:1 192.168.1.3
#route add -host 192.168.1.3 dev eth0:1
```

3. 激活设备。

```
#ifconfig eth0 up
```

4. 禁用设备。

```
#ifconfig eth0 down
```

5. 查看指定的网络接口的配置。

```
#ifconfig eth0
```

6. 查看所有的网络接口配置。

```
#ifconfig -a
```

1.7.2 ping

ping 命令用来确定网络上的主机是否可到达和到达速率。ping 命令的格式为：

```
#ping [OPTION] host
```

例如：

```
#ping www.sina.com.cn
#ping -c4 192.168.1.12
```

ping 命令将大小固定的数据包发送给对方，并要求对方返回。当终止 ping 命令时，会显示一些统计数据。通过数据判断是否返回以及返回时间，用户可以确定对方是否可到达，是否开机，以及网络延时时间。按<Ctrl + C>中断。

1.7.2.1 route

可以使用 route 命令来配置并查看内核路由表的配置情况。

例如：

1. 添加到主机的路由。

```
#route add -host 192.168.1.2 dev eth0:0
#route add -host 10.20.30.148 gw 10.20.30.40
```

2. 添加到网络的路由。

```
#route add -net 10.20.30.40 netmask 255.255.255.248 eth0
#route add -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41
#route add -net 192.168.1.0/24 eth1
```

3. 添加默认网关。

```
#route add default gw 192.168.1.1
```

4. 查看内核路由表的配置。

```
#route
```

5. 删除路由。

```
#route del -host 192.168.1.2 dev eth0:0
#route del -host 10.20.30.148 gw 10.20.30.40
#route del -net 10.20.30.40 netmask 255.255.255.248 eth0
#route del -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41
#route del -net 192.168.1.0/24 eth1
#route del default gw 192.168.1.1
```

1.7.2.2 traceroute

可以使用 traceroute 命令显示数据包到达目的主机所经过的路由。

例如：

```
#traceroute www.sina.com.cn
```

1.7.2.3 netstat

可以使用 netstat 命令来显示网络状态信息。

例如：

1. 显示网络接口状态信息。

```
#netstat -i
```

2. 显示所有监控中的服务器的 Socket 和正使用 Socket 的程序信息。

```
#netstat -lpe
```

3. 显示内核路由表信息。

```
#netstat -r
#netstat -nr
```

4. 显示 TCP/UDP 传输协议的连接状态。

```
#netstat -t
#netstat -u
```

1.7.2.4 hostname

可以使用 hostname 命令来更改主机名。例如：

```
#hostname myhost
```

使用 hostname 命令设置的主机名是临时的，在系统重启将失效。

1.7.2.5 arp

可以使用 arp 命令来配置并查看 arp 缓存。例如：

1. 查看 arp 缓存。

```
#arp
```

2. 添加一个 IP 地址和 MAC 地址的对应记录。

```
#arp -s 192.168.33.15 00:60:08:27:CE:B2
```

3. 删除一个 IP 地址和 MAC 地址的对应缓存记录。

```
#arp -d 192.168.33.15
```

1.7.3 telnet

telnet 命令是一种远程登录工具，只要拥有合法的注册名和口令，就能像使用本地机器一样访问远程计算机了。telnet 也允许用户通过输入注册名和口令从远程网点登录到自己的计算机上，从而通过网络或电话线完成检查电子邮件、编辑文件和运行程序等操作。但

是 telnet 只能在字符终端方式下工作，不支持图形用户界面。

telnet 的基本用法是：

```
#telnet [OPTION] IP [host [port]]
```

命令键入后，telnet 即会启动一个远程会话，本命令可使用的选项参数主要有

-d：启动调试功能

-a：自动注册

-n tracefile：打开跟踪程序，把跟踪程序数据保存在 tracefile 中

-e escape_char：将会话的转义字符设置为 escape_char

-l user：把用户名发送给远程系统，以便自动注册。本参数自动包括-a 参数

port：指出与远程系统连接的端口号，如不指定，将连接到缺省端口

成功地连接到远程计算机上后，telnet 就显示登录信息，并提示用户输入注册名与口令，如注册成功，就可以开始工作了。

提示：缺省版本未开启 telnet 和 ftp 服务，需另行启动该服务。

1.7.4 ftp

FTP（文件传输协议）是在 TCP/IP 网络计算机之间传输文件的简单而有效的办法。ftp 命令的功能是在本地机和远程机之间传送文件。它允许用户传输 ASCII 文件和二进制文件。在 ftp 会话过程中，用户可以通过使用 ftp 客户程序连接到另一台计算机上。用户可以在目录中上下移动、列出目录内容、把文件从远程机拷贝到本地机上、把文件从本地机传输到远程系统中。前提当然是您必须在本地和远程文件系统中具有进行这些操作的权限。

ftp 命令的基本格式如下：

```
#ftp [OPTION] [host]
```

可以用 ***help*** 命令取得可供使用的命令清单，也可以在 help 命令后面指定具体的命令名称，获得这条命令的说明。

ls：列出远程机的当前目录

cd：在远程机上改变工作目录

lcd：在本地机上改变工作目录

ascii：设置文件传输方式为 ASCll 模式

binary：设置文件传输方式为二进制模式

close：终止当前的 ftp 会话

hash：每次传输完数据缓冲区中的数据后就显示一个#号

get（mget）：从远程机传送指定文件到本地机

put（mput）：从本地机传送指定文件到远程机

open：连接远程 ftp 站点

quit：断开与远程机的连接并退出 ftp

?：显示本地帮助信息

！：转到 Shell 中

1.7.5 finger

使用 ***finger*** 命令来查询系统用户的信息，该命令的基本格式为：

```
#finger [-lmsp] [usr...] [usr@host...]
```

运行 finger 命令后会显示系统中某个用户的用户名、主目录、停滞时间、登录时间、登录 Shell 等信息，查询远程机上的用户信息时，就需要的用户名后面加上“@主机名”的方式。

第 2 章

系统安装与升级

2.1 系统安装

使用 CGSL 系统前，需要对系统进行安装，有关 CGSL 系统安装的详细说明请参考《CGSL V6 用户安装指南》。

2.2 系统升级

2.2.1 获取升级的镜像

目标版本的镜像请联系产品线同事获取。

2.2.2 升级配置

操作系统版本的升级通过 yum 源来实现，需要对 yum 源进行配置才可以进行升级。以下例子假设要使用 CGS-Linux-MAIN.V6.01.20B5-x86_64.dvd.iso 这个镜像来升级。

1.挂载本地 ISO 至/media/CGSL 目录：

```
#mount -o loop ./CGS-Linux-MAIN.V6.01.20B5-x86_64.dvd.iso /media/CGSL/
```

2./etc/yum.repos.d/CGSL-Media.repo 文件存在如下配置，其中 BaseOS 需要手工添加：

```
name=CGSL-$releasever - Media
baseurl=file:///media/CGSL/BaseOS/
enabled = 1
gpgcheck = 1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CGSL-V6
```

2.2.3 正式升级

1.升级 yum 组件：

```
#yum update -y yum
```

2. 升级系统组件

```
#yum update -y
```

第 3 章

用户和组群管理

CGSL 是一个多用户的操作系统，用户和用户组的管理是系统管理员的重要工作之一。本章的内容包括如何在命令行界面下完成用户账号、工作组的建立和维护，并正确设置用户权限和安全性问题。

3.1 概述

在 CGSL 系统中，每个用户对应一个帐号。CGSL 安装完成后，系统本身已创建了一些特殊用户，它们具有特殊的意义，其中最重要的是超级用户，即 root。

超级用户承担了系统管理的一切任务，可以不受限制地进行任何操作，因此建议只有在完全必要的情况下才以 root 身份进行操作。

由超级用户创建允许登录系统的普通用户，一般超级用户也需要为自己建立一个用来处理一般事务的普通帐户。

下面是用户和组群管理的一些基本概念：

用户名：系统中用来标识用户的名称，可以是字母、数字组成的字符串，区分大小写。

用户标识 UID：系统中用来标识用户的数字。

用户主目录：系统为每个用户配置的单独使用环境，即用户登录系统后最初所在的目录，用户的文件都放置在此目录下。

登录 Shell：用户登录后启动以接收用户的输入并执行输入相应命令的程序，CGSL 系统中默认使用的 Shell 为/bin/bash。

用户组/组群：具有相似属性的多个用户被分配到一个组中。

组标识 GID：用来表示用户组的数字标识。

超级用户在系统中的用户 ID 和组 ID 都是 0。普通用户的用户 ID（UID）从 500 开始编号，并且默认属于与用户名同名的组。组 ID（GID）也从 500 开始编号。

3.1.1 用 su 命令改变身份

用户在系统使用过程中可以随时使用su命令来改变身份。例如，系统管理员在平时工作时可以用普通帐号登录，在需要进行系统维护时用su命令获得root权限，之后再用su回到原帐号。

su 的语法为：

```
#su [OPTION]... [-] [USER [ARG]...]
```

-,-l 选项：login 登录并改变到所切换的用户环境；

-c 选项：commmand=COMMAND 执行一个命令，然后退出所切换到的用户环境；

USER：要切换到的用户名，如果不指定用户名，则默认将用户身份切换为 root ，系统会要求给出正确的口令。

提示：su 加参数“-”，表示默认切换到 root 用户，并且改变到 root 用户的环境，切换后的 Shell 为登录 Shell，在切换用户时，建议加此参数。

3.1.2 系统中的用户管理配置文件

/etc/passwd 文件

CGSL 系统中用于管理用户账号的基本文件是/etc/passwd ，该文件中包含了系统中所有用户的用户名和它们的相关信息。每个用户帐号在文件中对应一行，并且用冒号（:）分

为七个域。每一行的形式如下：

用户名:加密口令标识位:用户 ID:组 ID:用户的全名或描述:登录目录:登录 Shell

下面是 root 用户在此文件中对应的行：

```
root:x:0:0:root:/root:/bin/bash
```

Linux 系统将每一个用户仅仅看成是一个数字，即用每个用户惟一的用户 ID 来识别，配置文件/etc/passwd 给出了系统用户 ID 与用户名之间及其他信息的对应关系。

/etc/group 文件

在 Linux 中，使用组来赋予用户访问文件的不同权限。组的划分可以采用多种标准，一个用户可同时包含在多个组内。管理用户组的基本文件是/etc/group，其中包含了系统中所有用户组的相关信息。每个用户组对应文件中的一行，并用冒号分成四个域。其中每一行的形式如下：

用户组名:加密组口令标识位:组 ID :组成员列表

下面是用户组 sys 在/etc/group 中对应的一行：

```
sys:x:3:root,bin,adm
```

代表的信息包括：系统中有一个称为 sys 的用户组，设有口令，组 ID 为 3，组中的成员有 root、bin、adm 三个用户。

CGSL 在安装中同样创建了一些标准的用户组，在一般情况下，建议您不要对这些用户组进行删除和修改，除非您完全明白它们的用途和意义。

/etc/skel 目录

一般来说，每个用户都有自己的主目录，用户成功登录后就处于自己的主目录下。主目录中存放有与用户相关的文件、命令和配置。当为新用户创建主目录时，系统会在新用户的主目录下建立一份/etc/skel 目录下所有文件的拷贝，用来初始化用户的主目录。

3.2 命令行界面下的用户和组管理

3.2.1 用户管理

3.2.1.1 添加新用户

在命令行下，超级用户 root 可以按照以下的步骤来创建新的用户帐号：

1、在 Shell 提示符下，运行命令 ***useradd*** 或 ***adduser*** 来增加一个用户。

如要在系统中加入一个名为 newuser 的新用户，可以使用以下的命令：

```
#useradd newuser
```

useradd 命令还有很多可选参数，用来设置新建用户的一些属性，详细的参数使用方法，请参考其 ***man page*** 。

2、为用户设置口令。

通过 passwd 命令可以完成为新建用户设立口令。例如，超级用户要设置或改变用户 newuser 的口令时，可使用命令：

```
#passwd newuser
```

系统会提示输入新的口令，新口令需要输入两次。出于安全的原因，键入口令时不会在屏幕上回显出来。当用户使用不带参数的 passwd 命令时，可以修改自己的口令。

useradd 命令的常用参数和选项如下表：

表 3-1 useradd 命令

选项和参数	描述
-c comment	/etc/passwd 文件中用户全名或注释域的内容
-d home-dir	指定用于取代默认的/home/username 的用户主目录
-e date	禁用帐号的日期，格式为：YYYY-MM-DD
-f days	口令过期后，帐号禁用前的天数
-g group-name	用户所属主组群的组群名或组群 ID （该组群在指定前必须存在）
-G group-list	用户是其中成员的其他组群名或组群号码（默认以外的）列表，用逗号分隔（组群在指定前必须存在）
-m	若主目录不存在，则创建它
-M	不要创建用户主目录
-N	不要为用户创建用户私人组群
-r	创建一个 UID 小于 500 的不带主目录的系统帐号
-p password	使用 crypt 加密口令
-s	指定用户登录 Shell，默认为/bin/bash
-u uid	指定用户的 UID,它必须是唯一的，且大于 499

3.2.1.2 临时禁止一个用户

有时，需要临时禁止一个用户账号的使用而不是删除它。可以采用以下两种方法：

1、把用户的记录从/etc/passwd 文件中去掉，保留其主目录和其它文件不变；

2、在/etc/passwd 文件中关于该用户的 passwd 域的第一个字符前面加上一个“*”号。

3.2.1.3 删除用户

完全删除一个用户包括：

1、删除/etc/passwd 文件中此用户的记录；

2、删除/etc/group 文件中该用户的信息；

3、删除用户的主目录；

4、删除用户所创建的或属于此用户的文件。

userdel 命令可以用来删除用户及其主目录。命令的格式为：

```
#userdel [options] LOGIN
```

若使用-r 选项，表示用户主目录及其内部的文件将被删除。

3.2.2 用户组管理

以下是用户组管理的几个常用命令。

3.2.2.1 建立组

groupadd 命令用于将新组加入系统，命令的格式为：

```
#groupadd newgroup
```

新建的组默认使用大于 500 并大于每个其他组的 ID 的最小值。如果要指定组的 ID，可以在命令中加入-g 参数，如下面的命令将在/etc/passwd 文件中产生 GID 为 503 的项目：

```
#groupadd -g 503 newgroup
```

groupadd 常用参数和选项如下表：

表 3-2 groupadd 参数和选项

选项和参数	描述
-g gid	制定用户组的 GID，它必须是唯一的，且大于 499
-r	创建小于 500 的系统用户组
-f	若用户组已存在，退出并显示错误（组不会被改变）。若指定了-g 和-f 选项，且用户组已存在，-g 选项就会被忽略。

3.2.2.2 在组中加入用户

在组中加入用户的方法是直接编辑/etc/group 文件。前面讲过，这个文件的每一行表示一个组的信息，其中第四个域代表组内用户的列表。例如：user、user2、user3 都属于组 groupl，其组的 ID 为 509，则组的信息就是：

```
groupl::509:userl,user2,user3
```

要将新用户加入组中，只需用在文件编辑器中编辑/etc/group 文件，并将用户名加入用户列表中，用逗号分隔开即可。

3.2.2.3 删除组

使用 groupdel 命令来删除组。命令的格式如下：

```
#groupdel <groupname>
```

有几点需要注意：

1、组中的文件不能自行删除，也不能自行改变所属的组；

2、如果组是用户的基本组（即/etc/passwd 文件中对应用户的组标识），则这个组无法删除。

第 4 章

文件系统管理

对于任何一个成熟的操作系统而言，文件系统管理都是一个十分重要的部分。文件系统管理的好坏直接影响到操作系统的性能和安全。

4.1 文件系统基础和相关操作

文件系统是操作系统在硬盘或者分区上保存文件信息的方法和数据结构，也就是文件在硬盘或分区上的组织方式。CGSL 系统的一个重要特征之一就是支持多种文件系统，更为灵活并可以和许多其他种类的操作系统交换数据，其中最常用的是以下几种：

1、XFS：XFS是CGSL V6默认文件系统并在所有架构中支持。XFS 是一个具有非常高性能且可扩展的文件系统，同时在大多数要求的应用程序中都会进行常规部署。XFS提供了一种健壮的、优秀的以及功能丰富的文件系统，它具有的可伸缩性能够满足最苛刻的存储需求。

2、ext4： ext3的升级版本，ext4对ext3做了很多深层次的改进，设计更合理、性能更好、更可靠，同时还引入了一些新功能。

3、ext3：ext2的升级版本，其主要优点是在ext2的基础上加入了记录数据的日志功能。

4、ext2：支持标准Unix文件类型，可用于多种存储介质，向上兼容性好。

5、vfat：Windows 9x/2000及NT操作系统使用的扩展DOS文件系统，提供了对长文件名的支持。

6、Btrfs ：Btrfs 是下一代 Linux 文件系统，可提供高级管理、可靠性和可扩展性功能。Btrfs 为文件和元数据提供 checksum 确认。它还提供快照和压缩功能以及整合的设备管理。

7、NFS：允许在多台计算机之间共享文件系统的网络文件系统。

8、iso9660：标准的CD-ROM文件系统。

4.1.1 建立文件系统

一个分区或磁盘在作为文件系统被使用之前，先要初始化将记录数据的结构写入磁盘，这个过程叫做建立文件系统。

命令 mkfs 用于创建文件系统。用 mkfs 命令可以在任何指定的块设备上建立不同类型的文件系统。mkfs 命令的语法格式如下：

```
#mkfs [-V] [-t fstype] [fs-options] device [size]
```

mkfs 命令中各项参数的意义如下：

-V：强迫产生长格式输出；

-t fstype：选择文件系统的类型；

fs-option：将要建立的文件系统选项，它可以是以下的选项：

device：将创建文件系统所在设备的设备号；

size：文件系统的大小：

例如：要在硬盘上创建一个 ext4 的文件系统，用以下命令：

```
#mkfs -t xfs /dev/sdb1
```

4.1.2 挂载文件系统

成功地建立了文件系统后，还需要将文件系统挂载或称安装（mount）到 Linux 目录树的某个位置才能使用。文件系统所连接到的目录被称为挂载点或安装点。对于系统硬件设备，Linux 并不使用设备号或驱动器来访问，而是将他们对应为/dev 目录下的一个（也可能是多个）文件。

文件系统的挂载，可以在系统引导过程中自动挂载，也可以使用命令手工挂载。

加载文件系统的命令为 mount ，该命令的语法格式如下：

```
#mount [-fnrsvw] [-t vfstype] [-o options] device | dir
```

其中：device 代表文件系统的存储设备；dir 代表文件系统将要被放置在目录系统中的位置，即挂载点。

mount 命令常用如下几个选项：

-a：加载符合要求的所有文件系统，如果不加其他参数，将加载/etc/fstab 文件中列出

的所有文件系统。

-o：用于确定文件系统的读/写限制，ro（只读）、rw（读写）等。

-f：完成操作步骤，并不真正安装文件系统。

例如：把/dev/sda8 上类型为 xfs 的文件系统挂载到目录/mnt/tmp 下，并使之按只读方式被安装。

```
#mount -t xfs -o ro /dev/sda8 /mnt/tmp
```

♣ 提示：文件系统的加载位置必须是系统中已存在的目录，否则，需要在挂载前创建这个目录。

4.1.3 卸载文件系统

除了根文件系统之外，其他文件系统都是可以拆卸的。卸载文件系统的命令是 umount，其格式如下：

```
#umount [-dflnrv] dir | device [...]
```

该命令使用设备名或挂载点为参数，用于卸载与设备名或是挂载点对应的文件系统。

例如，需要卸载已挂载在/mnt 目录的/dev/sdb1 文件系统，可使用以下命令：

```
#umount /dev/sdb1
```

或者：

```
#umount /mnt
```

♣ 警告：不能卸载当前正在使用的文件系统，这样系统会报错，正确的方法是完全退出挂载点所在目录后再执行卸载命令。

4.1.4 用fstab文件配置文件系统

一般来说，用户经常使用的文件系统是比较固定的，如果每次使用时都进行挂载是很麻烦的，而且有时候需要一次安装很多的文件系统，可以考虑定义一个在系统引导时自动

安装文件系统的方法。

Linux 中使用/etc/fstab 文件能够完成这一功能，fstab 文件中列出了引导时需安装的文件系统的类型、挂载点及可选参数。fstab 文件是一个文本文件，可以方便地通过编辑工具进行修改。

♣ 警告：请在安装前备份原来的/etc/fstab 文件，以防修改出错导致下次系统引导时文件系统不能正确加载。

以下给出一个实际的/etc/fstab 文件（您的系统不一定与之相同）：

```
#
# /etc/fstab
# Created by anaconda on Mon Oct 20 01:26:17 2014
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cgsl-root   /                       xfs     defaults        1 1
UUID=0d03864d-683c-40e1-a6c6-4f8049eb99bc /boot                   ext4    defaults        1 2
/dev/mapper/cgsl-swap   swap                    swap    defaults        0 0
```

图 4-1 /etc/fstab 文件

/etc/fstab 文件也称为文件系统安装表，它的每一行代表一个需要安装的文件系统，其格式如下：

```
device mountpoint fstype options dump passno
```

其中：

device：指定要加载的文件系统设备

mountpoint：指定文件系统的加载点

fstype：指定安装文件系统的类型

options：使用逗号隔开的安装参数列表

dump：确定文件系统两次备份之间的时间

passno：指定系统引导时检查文件系统的顺序，根文件系统为 1，其余值为 2，如果没有指定，表示引导时文件系统不被检查。

4.1.5 检查和修复文件系统

文件系统是系统数据和资源的存储位置，所以应定期检查文件系统以发现损坏的文件并及时加以修补。

对文件系统进行检查可以通过使用 fsck 工具来完成，该命令的格式为：

```
#fsck [options] filesystem
```

fsck 的-ap 选项是最常用的参数组合，能满足大部分情况下的修复需求，且修复后不会丢失数据，

例如，要对/dev/sda1 进行文件系统的扫描和修复，常使用以下命令(该命令修复不会丢失数据)：

```
#fsck -ap /dev/sda1
```

当上述命令修复失败时，可以使用如下命令进行强制修复，此命令可能会丢失数据，请考虑后执行。

```
#fsck -y /dev/sda1
```

♣ 警告：用 fsck 检查文件系统时，必须先卸载该文件系统，否则可能导致不可修复的文件系统损坏。

4.1.6 常用文件系统管理命令

df 命令

使用 df 命令可以检查文件系统的磁盘空间占用情况，提供所有映射文件系统的空闲空间信息，其命令的语法格式为：

```
#df [OPTION]... [FILE]...
```

该工具默认以 KB 为单位显示已用的和可用的磁盘空间，查看以 MB 和 GB 为单位的信息，使用 ***df -h*** 命令。

du 命令

使用 du 命令可以显示被目录占用的空间的信息，此命令可以进入被统计目录的子目录中，并显示出子目录的统计信息，常用的选项如下：

-a：同时显示出目录和文件的磁盘使用情况；

-s：只显示磁盘的总体使用情况；

使用不加目录名的 du 命令将会显示出当前目录下的所有信息。

4.1.7 使用设备

在CGSL中，可以方便地使用各种驱动器、文件系统和网络设备，包括Linux分区、MS DOS和Windows分区、USB存储设备以及CD-ROM文件系统。

使用 CD-ROM

将光盘放入光盘驱动器中，在 Shell 提示符下键入以下命令：

```
#mount /dev/cdrom /mnt/cdrom
```

它通知操作系统自动探测文件系统并安装它，被安装的设备为/dev/cdrom，安装点为/mnt/cdrom。如果命令生效，光盘中的内容将出现在目录/mnt/cdrom 下。

如果安装命令失败，首先要确认/dev/cdrom 设备存在。如果使用的是 IDE CD-ROM，对应设备文件名可能是/dev/hdb、/dev/hdc 等；如果使用 SCSI CD-ROM，对应设备文件名可能为/dev/sda，/dev/sdb...

假设/dev/cdrom 不存在，而 CD-ROM 设备文件名为/dev/hdb，可以使用如下命令创建一个到/dev/cdrom 的符号链接。

```
#ln -s /dev/hdb /dev/cdrom
```

如果系统提示“设备已经安装（mounted）或目录忙”，可能是由于当前目录是加载点/mnt/cdrom 造成的，必须切换到其它目录才能进行。

执行完对光盘的操作后，在 Shell 提示符下键入以下命令卸载它。

```
#umount /mnt/cdrom
```

关于 mtools 工具

安装了系统中提供的 mtools 工具后，就可以使用 m 系列命令实现对 DOS/Windows 格式软盘的快速访问了。这一系列命令包括：

表 4-1 mtools 命令

命令	功能
mcd	进入 DOS 子目录

mcopy	拷贝 DOS 文件
mdel	删除 DOS 文件
mdir	查看 DOS 目录内容
mformat	格式化 DOS 磁盘
mmd	创建 DOS 目录
mmove	移动 DOS 下的文件
mren	将 DOS 下的文件改名

4.2 文件系统管理实例

4.2.1 添加新硬盘

如果您给 CGSLV6 系统添加了一个新的硬盘，您可能想给这个磁盘驱动器分区，并使用 xfs(或 ext4)文件系统，则通常按如下步骤操作：

1. 使用 parted 或 fdisk 来创建分区。
2. 使用 mkfs 来把分区格式化为 xfs(或 ext4)文件系统。
3. 使用 e2label 给分区标签。
4. 创建挂载点。
5. 把分区添加到/etc/fstab 文件中。

4.2.2 ext4 转换成 xfs

将 ext4 文件系统转换为 xfs 文件系统，可以提升文件系统性能以及使用 xfs 文件系统的新功能。具体步骤如下：

（1）确认 xfsprogs 和 xfsdump 两个软件包已经正常安装：

```
# rpm -qi xfsprogs
# rpm -qi xfsdump
```

（2）执行命令：

```
#fsck -fp <设备名>
```

4.2.3 ext3 转换为 ext4

将 ext3 文件系统转换为 ext4 文件系统，可以提升文件系统性能以及使用 ext4 文件系统的新功能。使用 tune2fs 程序可以将 ext3 文件系统分区转换为 ext4 文件系统，转换过程必须在分区没有被挂载前提下进行。此操作过程不可逆（ext4 分区无法被“降级”到 ext3），转换后无法用 ext3 驱动读写此文件系统。步骤如下：

（1）对于每个需要转换的分区，确保分区没有被挂载，执行命令：

```
#tune2fs -O extents,uninit_bg,dir_index <设备名>
```

（2）执行命令：

```
#fsck -fp <设备名>
```

提示：如果不执行 fsck，分区将不可读！使用 fsck 检测磁盘能够让文件系统回到一般状态。这个过程将在 group descriptors 找到 checksum 错误，这个是被预料到的。'f'参数要求磁盘检测一定要检查，哪怕文件系统标记是正常的。'p'参数要求检测的时候能够自动修复。

上述命令中，<设备名>为要转换的设备(分区)名称，如/dev/sdb1。

提示：以上命令执行完毕后，请确定把/etc/fstab 文件中的文件系统类型从 ext3 改成 ext4。

4.2.4 ext2 转换为 ext3

tune2fs 程序能够不改变分区上的已存数据来给现存的 ext2 文件系统添加日志。如需将 ext2 文件系统转换成 ext3，以 root 用户登录后执行如下命令：

```
#/sbin/tune2fs -j <设备名>
```

其中，<设备名>要操作的设备(分区)名称，如/dev/sdb1。

提示：以上命令执行完毕后，请确定把/etc/fstab 文件中相应设备的对应行中的文件系统类型从 ext2 改为 ext3。

如果被转换的文件系统为根文件系统，则需要一个 initrd 映像（或 RAM 磁盘）来引导，需要使用 mkinitrd 命令创建。关于如何使用 mkinitrd 命令，可查阅该命令的 man 手册，另外，还需确定 GRUB 或 LILO 的相关 配置会载入新的 initrd。

4.2.5 ext3 还原为 ext2

从 ext3 文件系统还原为 ext2 文件系统的具体步骤如下：(以/dev/hdb1 为例)

1、要还原分区，必须首先卸载分区。是登录为 root 用户，然后键入：

```
#umount /dev/hdb1
```

2、执行以下命令，从文件系统的超级块中清理文件系统特性：

```
#/sbin/tune2fs -O ^has_journal /dev/hdb1
```

3、执行以下命令来检查分区的错误：

```
#/sbin/e2fsck -y /dev/hdb1
```

4、执行以下命令将分区重新挂载为 ext2 文件系统：

```
#mount -t ext2 /dev/hdb1 /mount/point
```

其中，把 /mount/point 为分区的挂载点。

5、切换到分区的挂载目录中(上述的示例中为/mount/point)执行如下命令，删除根目录下的.journal 文件。

```
#cd /mount/point
#rm -f .journal
```

提示：如果想要永久地把分区改换成 ext2，请记住更新 /etc/fstab 文件。

4.3 磁盘分区管理

♣ 警告：改变系统的硬盘分区是件非常危险的事情，即使对于经验非常丰富的管理员，我们仍建议您在改变磁盘分区前先进行磁盘的数据备份。

4.3.1 Parted 工具

利用 parted 程序可以方便地进行磁盘分区的管理和定制，如查看现存的分区表、改变分区的大小、删除分区或创建新分区。

在 Shell 提示符下以超级用户身份键入如下命令（/dev/sdb 表示要定制的设备名）。

```
#parted /dev/sdb
```

启动 parted 后，在（parted）提示下键入 help 将显示可用命令的列表。下表列出的是用户最常用的 parted 命令。

表 4-2 parted 命令

命令和参数	描述
help	显示可用的命令列表
mklabel LABEL-TYPE	为分区表创建磁盘标签
mkpart PART-TYPE [FS-TYPE] START END	制作分区，不创建新文件系统
print	显示分区表
quit	退出 parted 程序
rm NUMBER	删除分区
select DEVICE	选择另一个设备来定制，不需要重启 parted
set NUMBER FLAG STATE	在分区上设置标志，state 可以是 on 或者 off

提示：要新建、删除分区或重新划分分区大小，分区所在设备不能被正在使用，即分区不能被挂载，且交换空间不能被启用。如果分区中不包括正在被使用的文件，可以用 ***umount*** 命令来卸载分区，使用 ***swapoff*** 命令来关闭交换分区。

4.3.1.1 查看分区表

启动 parted 后，键入 ***print*** 命令来查看分区表，屏幕输入信息如下：

```
[root@localhost ~]# parted
GNU Parted 2.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: ATA WDC WD800JD-75MS (scsi)
Disk /dev/sda: 80.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  525MB   524MB   primary  ext4         boot
 2      525MB   80.0GB  79.5GB  primary               lvm

(parted) _
```

图 4-2 查看分区表信息

第一行显示了磁盘的大小，第二行显示磁盘标签类型，后面部分为分区表，其中：

Number 域表示分区设备名中的数字，例如数字 1 代表/dev/sda1；Start 和 End 分别表示对应分区在硬盘上的起止位置，以 MB 为单位；Type 代表分区类型，可以是 primary 、extended 和 logical 之一；Filesystem 是文件系统的类型，可以是 ext4、ext3、ext2、FAT、Linux -swap 等等；Flags 域列出了分区被设置的标准，可用的标志有：boot、root、swap、hidden、raid、lvm 等。

提示：要不重新启动 parted 来选择不同的设备，使用 select 命令，再紧跟设备名，如/dev/hdb。然后，您便可以查看或配置它的分区表。

4.3.1.2 创建分区

如果我们要在/dev/sdb 上创建一个新分区，那么输入以下命令启动 parted：

```
#parted /dev/sdb
```

然后用 ***print*** 命令查看当前的分区表，确认磁盘上是否有足够的空闲空间可用于新分区。

提示：不要试图在正在使用的视图上创建新分区。

根据分区表决定新分区的起止点和分区类型，每个硬盘上只能有四个主分区，如果想拥有四个以上的分区，则必须先划出一个扩展分区，再在扩展分区内创建多个逻辑分区。

例如，要在/dev/sda1 上从 18000MB 到 21000 MB 创建一个文件系统为 ext2 的主分区，键入以下命令：

```
#mkpart primary ext2 18000 21000
```

如果使用 mkpartfs 命令，分区创建后文件系统也会被创建。只要一按下<Enter>键，对分区的改变就会生效，因此在执行前请仔细检查一下命令。

创建了分区后，使用 print 命令来确认所建分区已加入分区表中，并具有正确的分区类型、文件系统类型和大小。记住新分区的设备名中的数字以方便后续操作。

使用 mkpart 划分的分区还没有格式化，用下面的命令为分区创建文件系统：

```
#mkfs -t ext4 /dev/sdb3
```

接下来，可以为新分区注明标签、在目录树中为它创建加载点，使用 mount 命令加载并使用它。还可以把它的信息添加到/etc/fstab 文件中。

提示：parted 尚不支持创建 ext3、ext4 文件系统。因此，如果想创建一个 ext3、ext4 文件系统，先使用 part 划分分区，然后使用 mkfs 来创建。

4.3.1.3 删除分区

如果要删除/dev/sdb 上的一个分区，首先输入如下命令启动 parted：

```
#parted /dev/sdb
```

然后用 ***print*** 命令查看当前的分区表，确认要删除的分区。

使用 rm 来删除分区，例如，要删除分区设备名为 sdb3 ，则在（parted）提示符下键入：

```
（parted）rm 3
```

只要一按下<Enter>键，分区就会被删除，请在命令执行前仔细检查一下！

分区删除后，使用 print 命令可以看到分区已经被从分区表删除。最后要把它从/etc/fstab 文件中删除，找到与被删除的分区相应的行，从文件中删除它。

提示：不要视图删除正在使用的设备上的分区。

4.3.2 Fdisk 工具

fdisk 也是 CGSL 系统所常用的磁盘分区管理工具，以下介绍 fdisk 的基本用法。

4.3.2.1 查看分区情况

使用 fdsik -l filesystem 命令来查看分区情况，例如以下，以 root 用户执行：

```
#fdisk -l /dev/sdb

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table
```

4.3.2.2 创建分区

对磁盘进行操作，以 root 用户执行：fdisk filesystem，例如：

```
#fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x2dfa2374.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): n                //使用命令 n 新建分区
Command action
   e    extended
   p    primary partition (1-4)
p                                      //按照提示输入 p 新建一个主分区
```

```
Partition number (1-4): 1                    //选择分区号
First cylinder (1-261, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261): +1G          // 分
区大小

Command (m for help): w                      //写入分区结果并退出
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

4.3.2.3 删除分区

```
#fdisk /dev/sdb

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p                      //查看现在的分区情况

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2dfa2374

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         132     1060258+  83  Linux
/dev/sdb2             133         197      522112+  83  Linux

Command (m for help): d                      //删除分区
Partition number (1-4): 2                    //选择分区号
```

```
Command (m for help): p                    //再次查看分区情况，是否已删除

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2dfa2374

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         132     1060258+  83  Linux

Command (m for help): w                    //写入分区结果并退出
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

4.4 交换空间

4.4.1 交换空间是什么

交换空间(Swap Space)作为物理内存的后援存储，当系统的物理内存不够用的时，会将物理内存中最近较少被访问的非关键页面交换到交换空间中去，从而释放出一部分物理内存空间，以供当前运行的程序使用。待需要访问被交换出去的页面时，再从交换空间中换入相应的数据到物理内存中。

虽然交换空间可以为带有少量内存的机器提供帮助，但是这种方法不应该被当做是对内存的取代。交换空间位于硬盘驱动器上，其访问效率比物理内存要低很多。 交换空间可以是一个专用的交换分区（推荐的方法），也可以是一个交换文件，或是两者的结合。

4.4.2 添加交换空间

添加交换空间有两种方法：添加交换分区或添加交换文件。

4.4.2.1 添加交换分区

请参照 4.3 节进行分区，但需将分区的文件类型改为 82（即 SWAP 格式）。

使用 ***mkswap*** 命令来设置交换分区。在 Shell 提示下以根用户身份键入以下命令（假设 swap 分区是/dev/sdb2）：

```
#mkswap /dev/sdb2
```

要立即启用交换分区，键入以下命令：

```
#swapon /dev/sdb2
```

要在引导时启用，编辑/etc/fstab 文件来包含以下行：

```
/dev/sdb2                swap                    swap     defaults          0 0
```

在系统下次引导时，它就会启用新建的交换分区。

新添了交换分区并启用它之后，请查看 ***cat /proc/swaps*** 或 ***free*** 命令的输出来确保交换分区已被启用了。

4.4.2.2 添加交换文件

请参照以下方法：

在 Shell 提示下以根用户身份键入以下命令，其中的 count 为需要的交换文件大小(单位为 KB)：

```
#dd if=/dev/zero of=/swapfile bs=1024 count=65536
```

使用以下命令来设置交换文件：

```
#mkswap /swapfile
```

要立即启用交换文件而不是在引导时自动启用，使用以下命令：

```
#swapon /swapfile
```

要在引导时启用，编辑/etc/fstab 文件来包含以下行：

```
/swapfile                swap                    swap     defaults          0 0
```

系统下次引导时，它就会启用新建的交换文件。

新添了交换分区并启用它之后，请查看 ***cat /proc/swaps*** 或 ***free*** 命令的输出来确保交换分区已被启用了。

4.4.3 删除交换空间

要**删除**交换分区：

1、硬盘驱动器不能再被使用（分区不能被挂载，交换分区不能被启用）。如果确认驱动器不包含任何被使用的分区，可以直接卸载(umount)这些分区，使用 ***swapoff*** 命令来关闭硬盘驱动器上的所有交换空间(以/dev/sdb2 为例)：

```
#swapoff /dev/sdb2
```

2、从/etc/fstab 中删除交换分区相关的条目。

3、删除分区(以 parted 工具为例)：

在 Shell 提示下以根用户身份键入命令：***parted /dev/sdb***。这里的 /dev/sdb 是您要删除其中的交换空间的硬盘驱动器的设备名称。

在（parted）提示下，键入 ***print*** 来查看现存的分区并判定您想删除的交换分区的次要号码。

在（parted）提示下，键入 ***rm MINOR***，这里的 MINOR 是您想删除的分区的次要号码。

♣ 警告：改变会立即发生，您必须键入正确的次要号码。

键入 ***quit*** 来退出 parted。

要**删除**交换文件：

（1）在 Shell 提示下以 root 身份执行以下命令来禁用交换文件（这里的/swapfile 是交换文件）：

```
#swapoff /swapfile
```

（2）从/etc/fstab 中删除该项目。

（3）删除实际文件：

```
#rm /swapfile
```

4.4.4 移动交换空间

要把交换空间从某处移到另一处，请首先遵循删除交换空间的说明，再遵循添加交换空间的说明。

4.5 RAID 管理

4.5.1 RAID 是什么？

RAID(独立磁盘冗余阵列)的基本目的是把多个磁盘驱动器结合成一组虚拟大容量的驱动器阵列使用，其特色是提升设备的访问性能和提供容错功能。这个驱动器阵列从用户角度看就如同一个单一的逻辑贮存单元或驱动器。

RAID 是一种在多个磁盘上分散信息的方法。它使用磁盘分条（disk striping，RAID 级别 0）、磁盘镜像（disk mirroring，RAID 级别 1）、和带有奇偶校验的磁盘分条（disk striping with parity，RAID 级别 5）之类的技术来达到冗余性，降低潜伏时间，并且（或者）增加磁盘读写的带宽，提高从硬盘崩溃中恢复的能力。

RAID 的基本原理是：数据必须使用一致的形式被分散到阵列中的驱动器上。要达到这个目的，数据必须被分割成大小一致的“块”（大小通常是 32K 或 64K，也可使用不同大小）。每一块都会根据所用的 RAID 级别而写入其中的一个硬盘驱动器。当数据要被读取时，这个进程就会反过来进行，造成多个驱动器好象是一个大驱动器的假象。

4.5.2 谁应该使用 RAID

任何需要使大量数据便于存取的用户（如一般的系统管理员）都可以从 RAID 技术中受益。使用 RAID 的主要原因包括：

- 加快速度
- 增加贮存容量
- 减少磁盘失效带来的不利影响

4.5.3 硬件 RAID 和软件 RAID

RAID 技术有两种：硬件 RAID 和软件 RAID。

4.5.3.1 硬件 RAID

基于硬件的系统从主机之外独立地管理 RAID 子系统，并且它在主机处把每一组 RAID 阵列只显示为一个磁盘。硬件 RAID 对于操作系统来说是透明的，操作系统层识别到的就是一个普通的硬盘，其管理方法与普通硬盘一致，不在此赘述。

4.5.3.2 软件 RAID

软件 RAID 由操作系统实现，因为它不需要昂贵的磁盘控制器卡或热交换底盘，软件

RAID 提供了最廉价的解决方法。它还可以用在较便宜的 IDE 盘以及 SCSI 磁盘。

CGSL 内核的 MD 驱动程序是完全独立于硬件的 RAID 解决方案的范例。基于软件的阵列性能独立于服务器 CPU 的性能和载量之外。以下列举软件 RAID 的一些最重要的特性：

- 使用线程的进程重建
- 基于内核的配置
- 不必重建而可在 Linux 机器间迁移阵列
- 使用空闲的系统资源在后台重建阵列
- 对可热交换的驱动器的支持
- 对 CPU 的自动检测以便利用某些 CPU 优化功能

提示：热交换底盘允许您不必给系统断电而移除硬盘驱动器。

4.5.3.3 RAID 级别和线形支持

RAID 支持各类配置，包括级别 0、1、4、5 和线形。这些 RAID 类型的定义如下：

级别 0 —RAID 级别 0，经常被称作“分条”，它是面向性能的分条数据映射技术。这意味着被写入阵列的数据被分割成条，然后被写入阵列中的其它磁盘成员，从而允许低费用的高度 I/O 性能，但是它不提供冗余性。级别 0 阵列的贮存能力等于硬件 RAID 所有成员磁盘的总能力或软件 RAID 中所有成员分区的总能力。

级别 1 —RAID 级别 1，或“镜像”，被使用的时期长于任何其它形式的 RAID。级别 1 通过在阵列中的每个成员磁盘上写入相同的数据（在磁盘上留一个“镜像”副本）来提供冗余性。由于镜像的简单性和高度的数据可用性，它目前仍然很流行。使用两个以上磁盘操作的级别 1 可能会在读取时使用并行访问来获得高速数据传输，但是它更常用的是独立操作以提供高速 I/O 传输率。级别 1 提供了极佳的数据可靠性，并提高了读取任务繁重的程序的执行性能，但是它相对的费用也较高。级别 1 阵列的贮存能力与硬件 RAID 中被镜像的硬盘之一或软件 RAID 中被镜像的分区之一的贮存能力相同。

级别 4 —级别 4 使用集中到单个磁盘驱动器上的奇偶校验来保护数据。它更适合于事务性的 I/O 而不是大型文件传输。由于专职的奇偶校验磁盘代有固有瓶颈，级别 4 极少在没有写回缓存之类技术陪同的情况下使用。虽然 RAID 级别 4 在某些分区方案中是一种可选项目，它在 CGSL RAID 安装中却不是一个允许的选项。硬件级别 4 的贮存能力相当于所有成员磁盘去掉一个后的贮存能力。软件级别 4 的贮存能力相当于所有成员分区去掉一个后的贮存能力（如果它们的大小相同的话）。

级别 5 —这是最普遍的 RAID 类型。通过在某些或全部阵列成员磁盘驱动器中分布奇偶校验，RAID 级别 5 避免了级别 4 中固有的写入瓶颈。唯一的性能瓶颈是奇偶计算进程。

使用现代的 CPU 和软件 RAID，这种情况通常不是什么大问题。与级别 4 一样，其结果是非对称性能，读取大大地超过了写入性能。级别 5 经常与写回缓存一起使用来降低这种非对称性。硬件级别 5 的贮存能力相当于所有成员磁盘去掉一个后的贮存能力。软件 RAID 级别 5 的贮存能力相当于所有成员分区去掉一个后的贮存能力（如果它们的大小相同）。

线形 RAID—线形 RAID 是一种用简单的驱动器聚组来创建一个较大的虚拟驱动器的方法。在线形 RAID 中，区块从一个成员驱动器到另一个成员驱动器被依次分配，只有在第一个驱动器被完全填充后，才转到下一个驱动器。这种聚组没有提供任何性能方面的利益，因为 I/O 操作不太可能在成员驱动器间被分开。线形 RAID 也没有提供任何冗余性，事实上，它降低了可靠性 — 如果任何一个成员驱动器失效了，整个阵列都不能被使用。它的贮存能力是所有成员磁盘的总和。

提示：RAID 级别 1 的代价很高，因为您把相同的信息写入阵列中的所有磁盘，这浪费了驱动器空间。譬如，如果您设立了 RAID 级别 1，因而您的根分区（/）存在于两个大小各为 40G 的驱动器上，您虽然总共有 80G 空间，却只能实际利用其中的 40G，因为另外的 40G 就如同前 40G 的镜像一样。

提示：奇偶校验的信息是基于阵列中的其它磁盘成员的内容来计算的。当阵列中的某个磁盘上的数据失效时，这则信息就会被用来重建数据。然后，在替换失效磁盘之前，被重建的数据可以用来满足失败磁盘上的 I/O 请求；在替换失效磁盘之后，它可以用来在新磁盘上重建数据。

提示：RAID 级别 4 与级别 5 所占空间相同，但是级别 5 却优于级别 4。由于这个原因，级别 4 不被支持。

4.5.4 mdadm 管理软 RAID 阵列

4.5.4.1 创建新的阵列

mdadm 使用--create(或其缩写-C)参数来创建新的阵列，并且将一些重要阵列的标识信息作为元数据可以写在每一个底层设备的指定区间。--level(或者其缩写-l)表示阵列的 RAID 级别，--chunk(或者其缩写-c)表示每个条带单元的大小，以 KB 为单位，默认为 64KB，条带单元的大小配置对不同负载下的阵列读写性能有很大影响。--raid-devices(或者其缩写-n)表示阵列中活跃的设备个数，而--spare-devices(或者其缩写-x)表示阵列中热备盘的个数，一旦阵列中的某个磁盘失效，MD 内核驱动程序自动将热备磁盘加入到阵列，然后重构丢失磁盘上的数据到热备磁盘上。

创建一个 RAID 0 设备：

```
#mdadm --create   /dev/md0 --level=0 --chunk=32 --raid-devices=3 /dev/sd[i-k]1
```

使用阵列：

MD 设备可以像普通块设备那样直接读写，也可以做文件系统格式化。

```
#mkfs.ext4 /dev/md0
#mkdir -p /mnt/md-test
#mount /dev/md0 /mnt/md-test
```

停止正在运行的阵列：

当阵列没有文件系统或者其他存储应用以及高级设备使用的话，可以使用--stop(或者其缩写-S)停止阵列；如果命令返回设备或者资源忙类型的错误，说明/dev/md0 正在被上层应用使用，暂时不能停止，必须要首先停止上层的应用，这样也能保证阵列上数据的一致性。

```
#mdadm --stop /dev/md0
```

4.5.4.2 组装曾创建过的阵列

模式--assemble 或者其缩写(-A)主要是检查底层设备的元数据信息，然后再组装为活跃的阵列。如果我们已经知道阵列由那些设备组成，可以指定使用那些设备来启动阵列。

```
#mdadm -A /dev/md0 /dev/sd[b-h]
```

配置文件：

/etc/mdadm.conf 作为默认的配置文件，主要作用是方便跟踪软 RAID 的配置，尤其是可以配置监视和事件上报选项。Assemble 命令也可以使用--config(或者其缩写-c)来指定配置文件。我们通常可以如下命令来建立配置文件。

```
#echo DEVICE /dev/sd[b-h] /dev/sd[i-k]1 > /etc/mdadm.conf
#mdadm -Ds >>/etc/mdadm.conf
#cat /etc/mdadm.conf
```

4.5.4.3 查询阵列的状态

我们可以通过 cat /proc/mdstat 信息查看所有运行的 RAID 阵列的状态，在第一行中首先是 MD 的设备名，active 和 inactive 选项表示阵列是否能读写，接着是阵列的 RAID 级别，后面是属于阵列的块设备，方括号[]里的数字表示设备在阵列中的序号，(S)表示其是热备盘，（F）表示这个磁盘是 faulty 状态。在第二行中首先是阵列的大小，单位是 KB，接着是 chunk-size 的大小，然后是 layout 类型，不同 RAID 级别的 layout 类型不同，[6/6]和[UUUUUU]表示阵列有 6 个磁盘并且 6 个磁盘都是正常运行的，而[5/6]和[_UUUUU] 表示阵列有 6 个磁盘中 5 个都是正常运行的，下划线对应的那个位置的磁盘是 faulty 状态的。

```
#cat /proc/mdstat
```

4.5.4.4 管理阵列

mdadm 可以在 Manage 模式下，对运行中的阵列进行添加及删除磁盘。常用于标识 failed 磁盘，增加 spare（热备）磁盘，以及从阵列中移走已经失效的磁盘等等。使用--fail（或者其缩写-f）指定磁盘损坏。

```
# mdadm /dev/md0 --fail /dev/sdb
```

当磁盘已经损坏时，使用--remove(或者其缩写--f)参数将这个磁盘从磁盘阵列中移走；但如果设备还正在被阵列使用，则不能从阵列中移走。

```
# mdadm /dev/md0 --remove /dev/sdb
```

4.5.4.5 监控阵列

可以使用 mdadm 对 RAID 阵列进行监控，监控程序定时查询指定的事件是否发生，然后根据配置来妥善处理。

```
#mdadm --monitor --mail=root@localhost --program=/root/md.sh
--syslog --delay=300 /dev/md0 --daemonise
```

查看系统日志信息，可以看到哪个阵列或者阵列中的哪个设备发生过的哪些事件。

```
#mdadm -f /dev/md0 /dev/sdb
```

4.5.4.6 扩展阵列

如果在创建阵列时不想使用整个块设备，可以指定用于创建 RAID 阵列每个块设备使用的设备大小。

```
#dadm -CR /dev/md0 -l5 -n6 /dev/sd[b-g] -x1 /dev/sdh --size=102400
```

然后在阵列需要扩展大小时，使用模式--grow(或者其缩写-Q)以及--size 参数(或者其缩写-z) 在加上合适的大小数值就能分别扩展阵列所使用每个块设备的大小。

```
[root@fc5 mdadm-2.6.3]#./mdadm -Q /dev/md0
/dev/md0: 500.00MiB raid5 6 devices, 1 spare. Use mdadm --detail for more
detail.
[root@fc5 mdadm-2.6.3]#./mdadm --grow /dev/md0 --size=204800
[root@fc5 mdadm-2.6.3]#cat /proc/mdstat
Personalities : [raid0] [raid10] [raid6] [raid5] [raid4]
```

```
md0 : active raid5 sdh[6](S) sdg[5] sdf[4] sde[3] sdd[2] sdc[1] sdb[0]
      1024000 blocks level 5, 64k chunk, algorithm 2 [6/6] [UUUUUU]
      [============>......]   resync = 69.6% (144188/204800) finish=0.0min
speed=10447K/sec

unused devices: <none>
[root@fc5 mdadm-2.6.3]#./mdadm -Q /dev/md0
/dev/md0: 1000.00MiB raid5 6 devices, 1 spare. Use mdadm --detail for more
detail.
```

4.5.4.7 Bitmap 记录

使用 bitmap 模式记录 RAID 阵列有多少个块已经同步(resync)。参数--bitmap(或者其缩写-b)指定记录 bitmap 信息的文件名，如果是 interval 参数表示 bitmap 记录在每个设备的元数据区。--bitmap-chunk 表示每个 bit 位代表 RAID 设备多大的数据块，单位是 KB；而--delay(或者其缩写-d)指定多长事件同步 bitmap 信息到文件或者设备上，单位是秒，默认是 5 秒。--force(或者其缩写)表示覆盖掉已经存在 bitmap 文件。而且使用参数--examine-bitmap(或者其缩写-X)能够查看存储在文件或者设备元数据中的 bitmap 记录的信息。

```
#mdadm -CR /dev/md1 -l1 -n2 /dev/sdi1 /dev/sdj1 --bitmap=internal
```

4.6 逻辑卷管理器（LVM）

LVM是一种把硬盘驱动器空间分配成逻辑卷的方法。使用LVM，硬盘驱动器或硬盘驱动器集合会分配给一个或多个物理卷（physical volumes）。物理卷无法跨越一个以上驱动器。

物理卷被合并成逻辑卷组（logical volume group），如 **图 4-3**，唯一的例外是/boot/分区。/boot/ 分区不能位于逻辑卷组，因为引导装载程序无法读取它。如果您想把/分区放在逻辑卷上，您需要创建一个分开的/boot/ 分区，它不属于卷组的一部分。

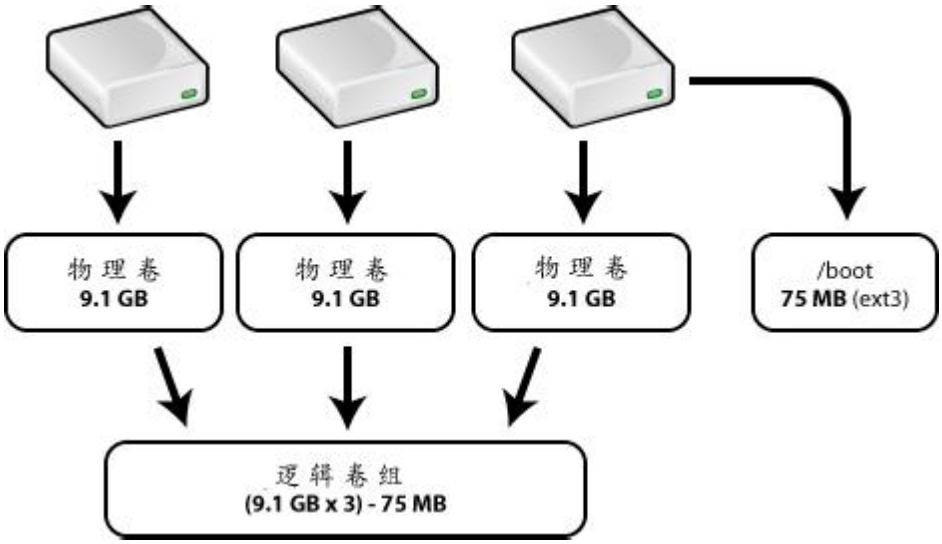


图 4-3 逻辑卷组

逻辑卷组被分成逻辑卷（logical volumes，相当于无 LVM 环境中的“分区”），如 **图 4-4**，它们被分配了挂载点（如/home 和/），以及文件系统类型（如 ext4）。当“分区”达到了它们的极限，逻辑卷组中的空闲空间就可以被添加给逻辑卷来增加分区的大小。当某个新的硬盘驱动器被添加到系统上，它可以被添加到逻辑卷组中，逻辑卷是可以扩展的分区。

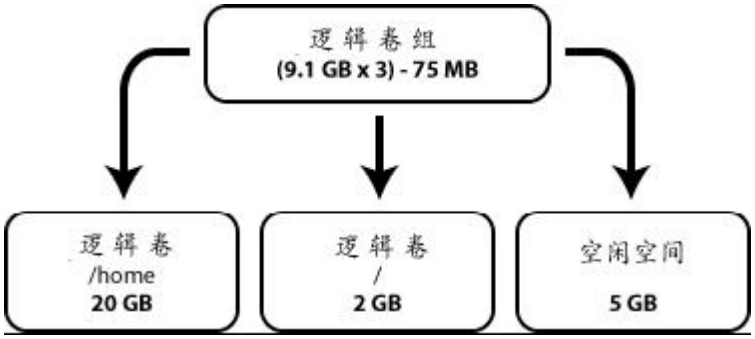


图 4-4 逻辑卷

CGSL 默认支持 LVM，LVM 命令摘要、用法说明及示例如下：

lvchange	更改逻辑卷的特性	#lvchange -t 60 /dev/vg00/lvol3
lvcreate	在卷组中创建逻辑卷	#lvcreate -L 100 /dev/vg00

lvdisplay	显示有关逻辑卷的信息	#lvdisplay -v /dev/vg00/lvol1
lvextend -m	为逻辑卷添加镜像	#lvextend -m 1 /dev/vg00/lvol3
lvextend -L	增加逻辑卷的大小	#lvextend -L 120 /dev/vg00/lvol3
lvreduce -L	减小逻辑卷的大小	#lvreduce -L 100 /dev/vg00/lvol3
lvreduce -m	减小逻辑卷的镜像副本的数量	#lvreduce -m 0 /dev/vg00/lvol3
lvremove	从卷组中删除逻辑卷	#lvremove /dev/vg00/lvol6
lvrmboot	删除到根区域、交换区域或转储区域的逻辑卷链路	#lvrmboot -d /dev/vg00/lvol2
pvchange	更改物理卷的特性	#pvchange -a n /dev/disk/disk2
pvck	对物理卷执行一致性检查	#pvck /dev/disk/disk47_p2
pvcreate	创建用作卷组的一部分的物理卷	#pvcreate /dev/rdisk/disk2
pvdisplay	显示有关物理卷的信息	#pvdisplay -v /dev/disk/disk2
pvmove	将盘区从一个物理卷移动到另一个物理卷	#pvmove /dev/disk/disk2 /dev/disk/disk3
pvremove	从物理卷中删除 LVM 数据结构	#pvremove /dev/rdisk/disk2
vgcfgbackup	保存卷组的 LVM 配置	#vgcfgbackup vg00
vgcfgrestore	恢复 LVM 配置	#vgcfgrestore -n /dev/vg00 /dev/rdisk/disk2
vgchange	打开或关闭卷组	#vgchange -a y /dev/vg00
vgcreate	创建卷组	#vgcreate /dev/vg01 /dev/disk/disk2 /dev/disk/disk3
vgdisplay	显示有关卷组的信息	#vgdisplay -v /dev/vg00
vgextend	通过添加物理卷来扩充卷组	#vgextend /dev/vg00 /dev/disk/disk2

vgexport	从系统中删除卷组	#vgexport /dev/vg01
vgimport	向系统添加现有卷组	#vgimport -v /dev/vg04
vgscan	扫描卷组的系统磁盘	#vgscan -v
vgreduce	通过从卷组中删除一个或多个物理卷来缩小卷组	#vgreduce /dev/vg00 /dev/disk/disk2
vgremove	从系统和磁盘中删除卷组定义	#vgremove /dev/vg00 /dev/disk/disk2

4.6.1 LVM 创建及配置示例

1、使用 fdisk 创建分区：

```
#fdisk /dev/sdb
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-261, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261): +500M

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e                    // 8e 为 LVM 格式的代码
Changed system type of partition 1 to 8e (Linux LVM)
```

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (66-261, default 66):
Using default value 66
Last cylinder, +cylinders or +size{K,M,G} (66-261, default 261): +500M

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2dfa2374

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              1          65      522081   8e  Linux LVM
/dev/sdb2             66         130      522112+  8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

2、将分区转换为 PV(物理卷)：

```
#pvcreate /dev/sdb1
  Writing physical volume data to disk "/dev/sdb1"
  Physical volume "/dev/sdb1" successfully created
#pvcreate /dev/sdb2
  Writing physical volume data to disk "/dev/sdb2"
  Physical volume "/dev/sdb2" successfully created
```

3、将 PV 组合成卷组 VG(卷组)

```
#vgcreate myvg1 /dev/sdb1 /dev/sdb2
  Volume group "myvg1" successfully created
```

4、创建逻辑卷 LV

```
#lvcreate -L 800M -n mylv1 myvg1
  Logical volume "mylv1" created
```

5、vgdisplay 及 lvdisplay 的相关信息

```
#vgdisplay
  --- Volume group ---
  VG Name                 myvg1
  System ID
  Format                  lvm2
  Metadata Areas          2
  Metadata Sequence No    2
  VG Access               read/write
  VG Status               resizable
  MAX LV                  0
  Cur LV                  1
  Open LV                 0
  Max PV                  0
  Cur PV                  2
  Act PV                  2
  VG Size                 1016.00 MiB
  PE Size                 4.00 MiB
  Total PE                254
  Alloc PE / Size         200 / 800.00 MiB
  Free  PE / Size         54 / 216.00 MiB
```

```
  VG UUID                 622JHS-61WH-VtKK-29i0-Ef0D-Icar-dBn83k

#lvdisplay
  --- Logical volume ---
  LV Name                  /dev/myvg1/mylv1
  VG Name                   myvg1
  LV UUID                  IdqHyQ-cEkp-Hz4X-ulPv-kvY3-nmTi-uRmGqw
  LV Write Access          read/write
  LV Status                available
  # open                   0
  LV Size                  800.00 MiB
  Current LE               200
  Segments                  2
  Allocation               inherit
  Read ahead sectors        auto
  - currently set to      256
  Block device             253:2
```

6、格式化 LVM 分区并挂载分区：

```
#mkfs.ext4 /dev/mapper/myvg1-mylv1
#mkdir /mylv1
#mount /dev/mapper/myvg1-mylv1 /mylv1
```

7、对已挂载的 LVM 文件系统进行扩容：

未扩容前：

```
/dev/mapper/myvg1-mylv1   788M    17M   731M    3% /mylv1
```

使用 lvextend 和 resize2fs 进行在线扩容：

```
#lvextend -L +100M /dev/mapper/myvg1-mylv1
  Extending logical volume mylv1 to 900.00 MiB
  Logical volume mylv1 successfully resized
# resize2fs /dev/mapper/myvg1-mylv1
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/myvg1-mylv1 is mounted on /mylv1; on-line resizing
required
old desc_blocks = 1, new_desc_blocks = 1
```

```
Performing an on-line resize of /dev/mapper/myvg1-mylv1 to 230400 (4k) blocks.
The filesystem on /dev/mapper/myvg1-mylv1 is now 230400 blocks long.
```

扩容后：

```
/dev/mapper/myvg1-mylv1   886M    17M   829M    3% /mylv1
```

4.7 设备映射多路径（DM-Multipath）

4.7.1 DM-Multipath 概述

device-mapper-multipath（简称 DM-Multipath）是 CGSL 系统自带的多路径软件，可让您将服务器节点和存储阵列间的多个 I/O 路径配置为一个单一设备。这些 I/O 路径是可包含独立电缆、交换机以及控制器的物理 SAN 连接。多路径集合了 I/O 路径，并生成由这些整合路径组成的新设备。

可使用 DM-Multipath 提供：

冗余功能：DM-Multipath 可在主动/被动配置中提供出错冗余。在主动/被动配置中，只有一半的路径在每次 I/O 时都使用。如果 I/O 路径的任意元素（电缆、交换机或者控制器）出现故障，就会将 DM-Multipath 切换到备用路径。

性能提高：可将 DM-Multipath 配置为主动/主动模式，其中将 I/O 以轮叫调度算法方式分布到所有路径中。在有些配置中，DM-Multipath 可在 I/O 路径中检测负载并动态重新平衡负载。

4.7.2 DM-Multipath 配置及管理示例

手动加载 multipath 模块

```
# modprobe dm-multipath
# modprobe dm-round-robin
```

使用 mpathconf 程序设置多路径，它可创建多路径配置文件/etc/multipath.conf。使用以下步骤为基本故障切换配置设置 DM-Multipath：

1、运行带 --enable 选项的 mpathconf 命令：

```
# mpathconf –enable
```

2、编辑/etc/multipath.conf 文件，设置相关的特性，编辑完成保存配置文件并退出。如果不清楚阵列的特性，可以使用 path_grouping_policy 的默认值 failover（倒换）。

3、重启服务：

```
# service multipathd start
```

4、清空已有的 multipath 记录

```
# mutilpath -F
```

5、重新扫描设备

```
# mutilpath -v2
```

6、查看所有设备（举例）

```
# mutilpath -ll

36001438002a56fd60000600001c60000 dm-250 HP,HSV450
[size=5.0G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=10][enabled]
\_ 2:0:5:4 sdck 69:128 [active][ready]
\_ 5:0:5:4 sdhz 134:128 [active][ready]
\_ round-robin 0 [prio=50][enabled]
\_ 2:0:0:4 sdz 69:14 [active][ready]
\_ 5:0:0:4 sdhk 134:44 [active][ready]
360060e801439ba00000139ba00002209 dm-62 HP,OPEN-V
[size=8.0G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=2][active]
\_ 5:0:4:9 sdlv 128:27 [active][ready]
\_ 2:0:2:9 sdaq 134:0 [active][ready]
\_ 5:0:7:9 sdnv 128:272 [active][ready]
\_ 2:0:11:9 sdhq 134:0 [active][ready]
```

以上是 device-mapper-mulitpath 生成的设备，分别为 dm-250 和 dm-62

其中，

dm-250 对应的物理设备是 sdck、sdhz、sdz、sdhk

dm-62 对应的物理设备是 sdlv、sdag、sdnv、sdhq

第 5 章

软件包管理

5.1 使用 rpm 命令

rpm 是一个功能十分强大的软件包管理系统，它使 Linux 下安装、升级和删除软件包的工作变得简单容易，并且具有查询、验证软件包的功能。与图形化工具相比，使用命令行可以获得更大的灵活性。

本章例子都以 example-1.2.3-1.el8.rpm 代表软件包名称。

5.1.1 安装、升级和更新

使用下面三个参数安装、升级和更新软件包：

1. rpm -i 安装一个新的软件包
2. rpm -U 升级一个软件包，如果系统中原来不存在，就进行安装
3. rpm -F 更新一个软件包，如果系统中原来不存在，就不进行安装

经常和这几个参数配合使用的参数包括：

1. -v 查看安装过程中的各种信息
2. -h 在安装过程中显示进度条

一个常用的命令格式如下：

```
#rpm -ivh example -1.2.3-1.el8.rpm
```

这个命令将安装软件包，同时显示安装信息和进度条。

5.1.2 删除

删除一个软件包的命令示例如下：

```
#rpm -e example
```

♣ 警告：删除时使用的是软件名，而不是软件包的全称。

5.1.3 查询

列出用户已经安装的 RPM 包清单

如果想查询系统中所有已经安装的 RPM 包，使用 rpm -qa 即可输出所有已安装 RPM 包的列表。

如果是查看某个已经安装的软件包，则使用 rpm -q example 命令。

查看一个 RPM 包中包括的文件

想要查看某个软件包中包含的文件清单，有下面两种方法：

如果是未安装的软件包，则使用：

```
#rpm -qlp example -1.2.3-1.el8.rpm
```

如果是已安装的软件包，请使用：

```
#rpm -ql example
```

确定某个文件属于哪个 RPM 包

如果遇到了一个不认识的文件，要找出它属于哪个软件包，则首先记录这个文件的完整路径（绝对路径），然后输入以下命令：

```
#rpm -qf filename
```

查询 RPM 包的用途

用户可以在安装或使用时查询每个软件包的用途、版本及其它信息，使用如下的命令完成查询：

```
#rpm -qip example -1.2.3-1.el8.rpm
```

5.1.4 验证

验证一个软件包，就是比较原始包和已安装软件包中文件的信息。具体来说，这些信息包括每个文件的大小、MDS 校验和、访问许可权、类型以及所属的用户和组等。

使用命令 rpm -V 可以验证一个包，下面是常用的几种情况：

验证包含某个特殊文件的软件包

```
#rpm -Vf filename
```

验证所有已安装的软件包

```
#rpm -Va
```

上面介绍是几个常用的 RPM 命令，关于 RPM 工具的更多资源，请参看相关的 man 手册页；还可以在以下的网址 **http://www.rpm.org** 获得 RPM 的最新资源

5.2 使用 yum 命令

yum 基于 rpm 包管理，能够从指定的服务器自动下载 rpm 包并且安装，可以自动解决依赖关系，一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。另外 cgslv6 版本的 yum 其实是一个软连接，连接到 dnf-3 组件。

5.2.1 配置软件仓库

以使用 CGSL V6 的 DVD 光盘作为 yum 软件仓库为例：

将 DVD 光盘插入硬盘，修改 /etc/yum.repos.d/CGSL-Media.repo 文件，将 baseurl=file:///media/CGSL/修改为：

```
baseurl=file:///media/CGSL/BaseOS
```

并保存退出。

并重新挂载至/media/CGSL/：

```
#mkdir -p /media/CGSL/
#mount /dev/cdrom /media/CGSL/BaseOS
```

5.2.2 yum 常用命令介绍

列出 yum 管理的所有软件名称和版本：

```
#yum list
```

搜索某个软件，支持关键字搜索：

```
#yum search packagename
```

安装软件包，支持通配符。-y 表示安装不需要确认：

```
#yum install -y packagename
```

删除软件包：

```
#yum remove packagename
```

第 6 章

使用 Vim 编辑器

Vim 自产生以来，历经不断革新，最新版的 Vim 已经具有很强大的功能，使用户能够更加轻松、便捷地使用它。

6.1 Vim 的工作模式

Vim 一共有三种工作模式，分别为：

- 编辑模式
- 插入模式
- 命令模式

在初始启动后首先进入编辑模式，这时用户能利用一些预先定义的按键来移动光标、删除文字、 复制或粘贴文字等。这些按键均是普通的字符，例如 l 是向上移动光标，相当于向右箭头键，k 是 向下移动光标，相当于向下箭头键。在编辑模式下，用户还能利用一些特别按键选定文字，然后 再进行删除、或复制等操作。

当用户在编辑模式下键入 i, a, o 等命令之后，可进入插入模式；键入 : 可进入命名模式。

6.1.1 命令模式

开始进入 Vim 时处于命令模式，如果已经处于插入模式或末行模式，按<ESC>键可以回到命令模式。在这种模式下，只能用按键指令，不能输入文字。

6.1.2 插入模式

插入模式就是要把文本插入到要编辑的文件，插入位置根据所用的命令不同而不同。从命令模式进入插入模式需要键入 i、a、o、r 及 I、A、O、R 等命令。在完成文本的输入

后，必须用<ESC>键返回命令模式。

6.1.3 命令模式

命令模式为 ex 模式。在命令方式下，键入“：”，光标跳到屏幕末行并显示键入的末行字符，此时键入命令后回车，Vim 会根据需要在末行显示出一定的响应信息，同时将自动回到命令状态。

6.2 Vim 编辑文件的基本过程

在命令行键入 Vim testfile，其中 testfile 代表要打开的文件名，如果文件不存在，Vim 将自动新建一个名为 testfile 文件。

进入 Vim 后，按 i 进入插入模式，就可以编写文件了，光标可以由方向键来移动。<BacksPace>键可以删去前一个字符。

如果已写好文件，就可以按<ESC>回到命令模式，然后用:w 存档（注意，是冒号命令），这时还不会离开 Vim，要离开可按:q，也可以合起来用:wq，代表保存后离开。

6.2.1 光标的移动

6.2.1.1 基本的光标移动

左	**h**	Backspace 或左方向键
下	**j**	Enter 或＋或下方向键
上	**k**	- 或上方向键
右	**l**	space 或右方向键
向下翻页	**Ctrl + f**	PageDown
向上翻页	**Crtl + b**	PageUp

6.2.1.2 复杂光标移动

0	移至行首，或是<Home>键，
~	大小写切换
$	移至行尾，或<End>键
G	移至文件尾（最后一行的第一个非空白字符处）

gg	移至文件首（第一行第一个非空白字符处）
w	移至下一个字首
W	同上，但会忽略一些标点符号
e	移至后一个字字尾
E	同上，但会忽略一些标点符号
B	移至前一个字字首
B	同上，但会忽略一些标点符号
H	移至屏幕顶部第一个非空白字符
M	移至屏幕中间第一个非空白字符
L	移至屏幕底第一个非空白字符
nl	从当前光标所在位置向右移至第 n 个字符处
:n	或 nG 移至第 n 行行首，注意 n 为具体的数字，如 1，2，3……
)	移至下一个句首
(	移至上一个句首
}	移至下一个段落首
{	移至上一个段落首

6.2.2 基本编辑指令

6.2.2.1 进入插入模式指令

i	在光标所在字符前开始输入文字（insert）
a	在光标所在字符后开始输入文字（append）
o	在光标所在行下开一新行来输入文字（open）
I	在行首开始输入文字
A	在行尾开始输入文字
O	在光标所在行上开一新行来输入文字

J	将下一行整行连接到本行（joint）

6.2.2.2 删除指令

x	删除光标所在处的字符。也可用<Del>键。
X	删除光标所在位置前的字符。
dd	删除一整行。
dw	删除一个字（delete word ）。
dG	删至文件尾。
D	删至行尾，或 d$（含光标所在处字符）。

6.2.2.3 取代及还原

r	取代光标所在处的字符。
R	取代字符直至按<Esc>为止。
cc	取代整行内容。或大写 S 亦司
cw	替换一个英文字。
~	光标所在处之大小写转换。
C	取代至行尾，即光标所在处以后的字都会被替换。
c $	同上。
c0	取代至行首，或 c~。
u	撤销前面的操作，即 undo，撤销的次数没有限制。
U	在光标没离开本行之前，恢复所有编辑动作。

6.2.2.4 复制

yy	复制光标所在行整行。或一个大写 Y 。
2yy	或 yZy 复制两行。
y$	复制至行尾。含光标所在处字符。
yG	复制至文件尾。
ylG	复制至文件首。

6.2.2.5 查找与替换

查找

/	在命令模式下，按 / 会在左下角出现一个 / ，键入要查找的字串，按回车开始查找。
?	同 / ，只是 / 是向前（下）找，? 是向后（上）找。
n	继续查找。
N	继续寻找（反向）。
*	寻找光标所在处的字（要完全符合）。
#	同上，但 * 是向前（下）找，# 则是向后（上）找。
g*	同*，但部分符合即可。
g#	同#，但部分符合即可。

替换

```
:[range]s/pattern/string/[c,e,g,i]
```

用 string 替换 Pattem。Range 指的是范围，例如 1,7 指从第一行至第七行，1,$ 指从第一行至最后一行，也就是整篇文章，也可以%代表；C 每次替换前会询问；e 不显示 error；g 不询问，整行替换；i 不分大小写。

6.2.2.6 离开

:q 如文件有修改而没保存，会警告，且无法离开。

:q! 放弃所有修改，强迫离开。

:wq 保存文件后离开，即使文件没有修改也会再保存一次。

:x 保存文件后离开，但如果文件没有修改，则不会做保存的动作。

ZZ 功能同:x，但是 ZZ 是命令模式下的命令，不是 ex 模式下的命令，不需要先输入冒号。

:w 另存，不加文件名就是写入原文件。

第 7 章

基础系统管理

本章主要介绍常用的系统管理基础操作，包括时间和日期、键盘、控制台、任务自动化、服务等管理。

7.1 时间和日期管理

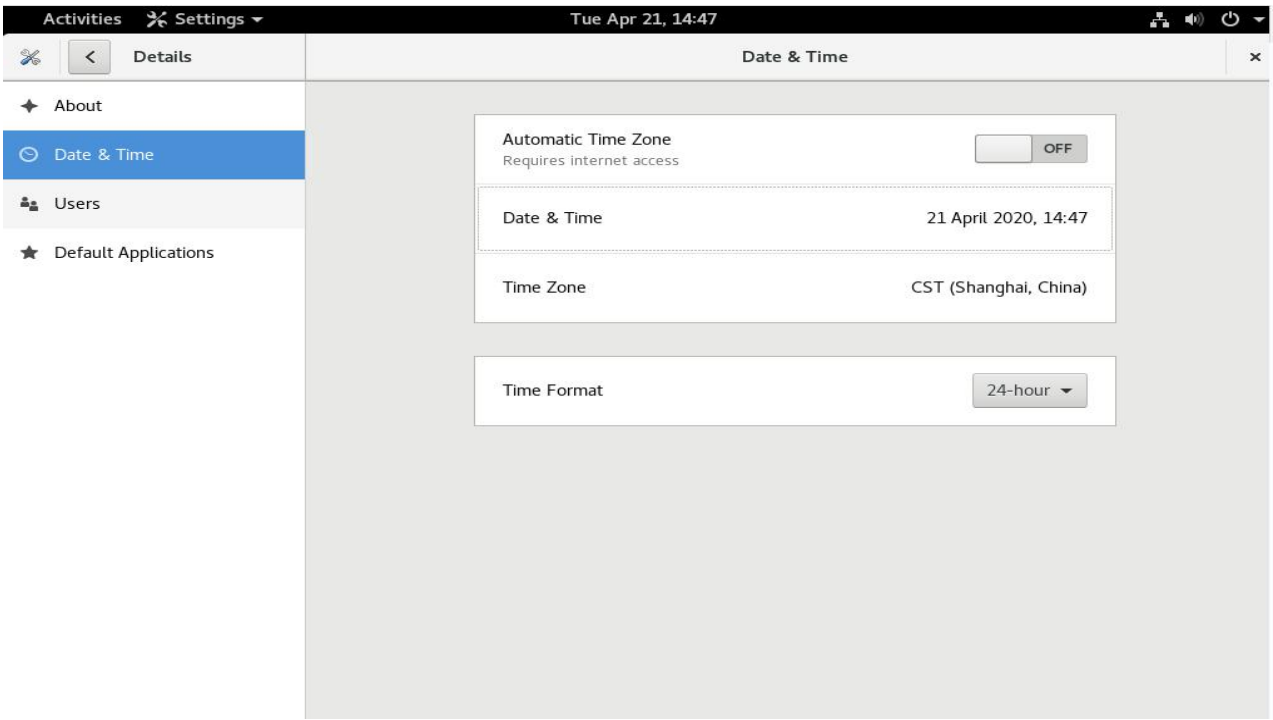
时间和日期管理工具允许用户改变系统日期和时间；配置系统使用的时区；以及设置网络时间协议（NTP）守护进程来与时间服务器的系统时钟同步。

须要进入图形环境并具备 root 用户权限才能使用该工具。要从桌面上启动这个程序，可点击【Activities】->【Show Applications】->【Settings】->【Date & Time】。

7.1.1 日期和时间属性

如图 7-1 所示，所出现的第一个带活页标签的窗口被用来配置系统日期、时间。

要改变日期和时间，需要先取消【Automatic Time Zone】（NTP），然后点击【Date & Time】标签，通过鼠标点击来修改年、月、日、时、分、秒。


Activities
Settings
Tue Apr 21, 14:47
Details
Date & Time
About
Date & Time
Users
Default Applications
Automatic Time Zone
Requires internet access
OFF
Date & Time
21 April 2020, 14:47
Time Zone
CST (Shanghai, China)
Time Format
24-hour

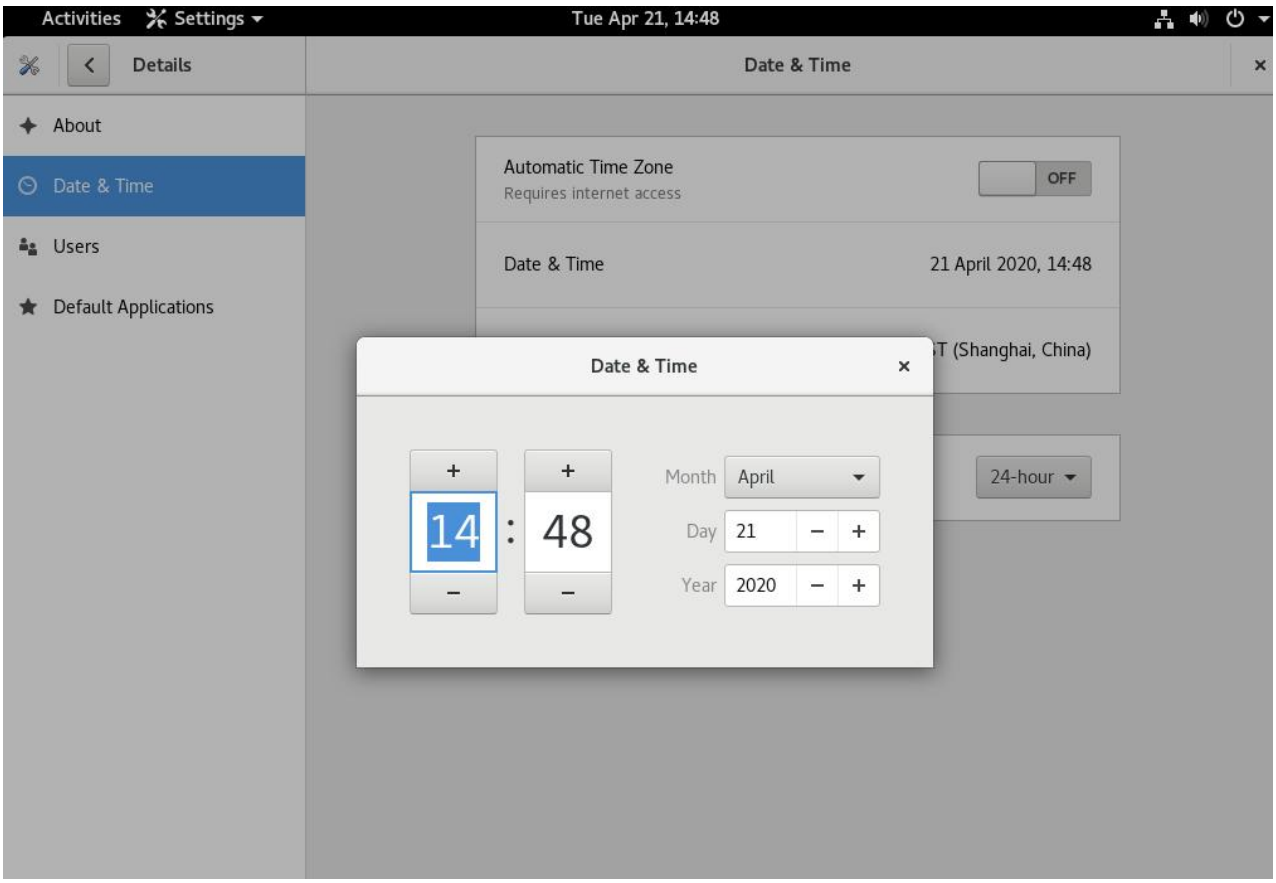

Activities
Settings
Tue Apr 21, 14:48
Details
Date & Time
About
Date & Time
Users
Default Applications
Automatic Time Zone
Requires internet access
OFF
Date & Time
21 April 2020, 14:48
T (Shanghai, China)
24-hour
Date & Time
14
48
Month
April
Day
21
Year
2020

图 7-1 时间和日期属性

要改变日期和时间，需要先取消【网络时间】（NTP），然后通过鼠标点击来修改年、月、日、时、分、秒。

7.1.2 时区配置

要配置系统时区，点击【时区】标签。时区可以通过互动地图来改变，也可以从地图下面的列表中选择想要的时区。要使用地图，点击代表您所在时区的城市，一个红色的圆点会出现，地图下的时区列表中的选择也会相应改变。

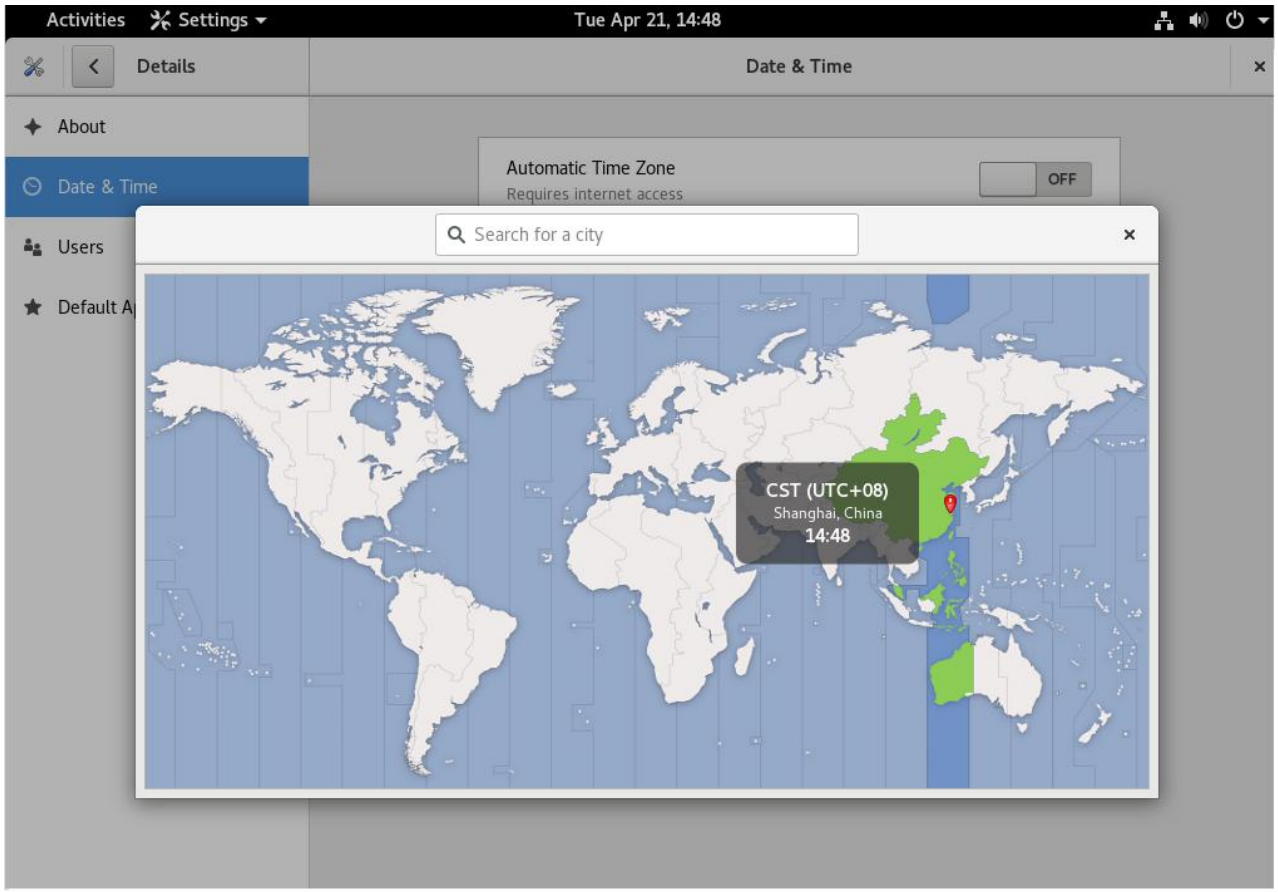


图 7-2 时区设置

7.2 键盘配置

安装程序允许用户为他们的系统配置键盘布局。

显示目前的键盘布局设置：

```
# localectl status
   System Locale: LANG=en_US.UTF-8
       VC Keymap: us
      X11 Layout: us
```

列举出可以设置的键盘布局设置：

```
# localectl list-keymaps
```

设置键盘布局设置：

```
# localectl set-keymap map
```

7.3 任务自动化

在 CGSL 中，任务可以被配置在指定的时间段、指定的日期、或系统平均载量低于指定的数量时自动运行。CGSL 预配置了对重要系统任务的运行，以便使系统能够时时被更新。譬如，被 locate 命令使用的 slocate 数据库每日都被更新。系统管理员可使用自动化的任务来执行定期备份、监控系统、运行定制脚本等等。

CGSL 随带几个自动化任务的工具：cron、at、和 batch。

7.3.1 cron

cron 是一个可以用来根据时间、日期、月份、星期的组合来调度对重复任务的执行的守护进程。

cron 假定系统持续运行。如果当某任务被调度时系统不在运行，该任务就不会被执行。CGSL 中，由 crond 服务提供相关功能，且默认安装并启用了 crond 服务。**配置 cron 任务**

cron 的主配置文件是/etc/crontab，它包括下面几行：

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
```

```
# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

前四行是用来配置 cron 任务运行环境的变量。Shell 变量的值告诉系统要使用哪个 Shell 环境（在这个例子里是 bash Shell）；PATH 变量定义用来执行命令的路径。cron 任务的输出被邮寄给 MAILTO 变量定义的用户名。如果 MAILTO 变量被定义为空白字符串（MAILTO=“ ”），电子邮件就不会被寄出。HOME 变量可以用来设置在执行命令或脚本时使用的主目录。

/etc/crontab 文件中的每一行都代表一项任务，它的格式是：

```
minute   hour   day   month   dayofweek   command
```

- minute：分钟，从 0 到 59 之间的任何整数
- hour：小时，从 0 到 23 之间的任何整数
- day：日期，从 1 到 31 之间的任何整数（如果指定了月份，必须是该月份的有效日期）
- month：月份，从 1 到 12 之间的任何整数（或使用月份的英文简写如 jan、feb 等等）
- dayofweek：星期，从 0 到 7 之间的任何整数，这里的 0 或 7 代表星期日（或使用星期的英文简写如 sun、mon 等等）
- command：要执行的命令（命令可以是 ls /proc >> /tmp/proc 之类的命令，也可以是执行您自行编写的脚本的命令。）

在以上任何值中，星号（*）可以用来代表所有有效的值。譬如，月份值中的星号意味着在满足其它制约条件后每月都执行该命令。

整数间的短线（-）指定一个整数范围。譬如，1-4 意味着整数 1、2、3、4。

用逗号（,）隔开的一系列值指定一个列表。譬如，3,4,6,8 标明这四个指定的整数。

正斜线（/）可以用来指定间隔频率。在范围后加上/<integer>意味着在范围内可以跳

过 integer。譬如，0-59/2 可以用来在分钟字段定义每两分钟。间隔频率值还可以和星号一起使用。例如，*/3 的值可以用在月份字段中表示每三个月运行一次任务。

开头为井号（#）的行是注释，不会被处理。

如您在/etc/crontab 文件中所见，它使用 run-parts 脚本来执行/etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly 和/etc/cron.monthly 目录中的脚本，这些脚本被相应地每小时、每日、每周、或每月执行。这些目录中的文件应该是 Shell 脚本。

如果某 cron 任务需要根据调度来执行，而不是每小时、每日、每周、或每月地执行，它可以被添加到/etc/cron.d 目录中。该目录中的所有文件使用和/etc/crontab 中一样的语法，请参见以下范例。

```
#record the memory usage of the system every monday
#at 3:30AM in the file /tmp/meminfo
30 3 * * mon cat /proc/meminfo >> /tmp/meminfo
#run custom script the first day of every month at 4:10AM
10 4 1 * * /root/scripts/backup.sh
```

根用户以外的用户可以使用 crontab 工具来配置 cron 任务。所有用户定义的 crontab 都被保存在/var/spool/cron 目录中，并使用创建它们的用户身份来执行。要以某用户身份创建一个 crontab 项目，登录为该用户，然后键入 ***crontab -e*** 命令，使用由 VISUAL 或 EDITOR 环境变量指定的编辑器来编辑该用户的 crontab。该文件使用的格式和/etc/crontab 相同。当对 crontab 所做的改变被保存后，该 crontab 文件就会根据该用户名被保存，并写入文件/var/spool/cron/username 中。

cron 守护进程每分钟都检查/etc/crontab 文件、etc/cron.d/目录、以及/var/spool/cron 目录中的改变。如果发现了改变，它们就会被载入内存。这样，当某个 crontab 文件改变后就不必重新启动守护进程了。

提示：CGSL 自带了 ICT(集成配置)工具，其中提供了图形化的定时任务配置功能(基于 cron 实现)，简化了 cron 定时任务的配置和使用，有关 ICT 工具的详细使用说明请参考相应的用户指南文档。

7.3.1.2 cron 权限控制

/etc/cron.allow 和/etc/cron.deny 文件被用来限制对 cron 的使用。这两个使用控制文件的格式都是每行一个用户。两个文件都不允许空格。如果使用控制文件被修改了，cron 守护进程（crond）不必被重启。使用控制文件在每次用户添加或删除一项 cron 任务时都会被读取。

无论使用控制文件中的规定如何，根用户都总是可以使用 cron。

如果 cron.allow 文件存在，只有其中列出的用户才被允许使用 cron，并且 cron.deny 文件会被忽略。

如果 cron.allow 文件不存在，所有在 cron.deny 中列出的用户都被禁止使用 cron。

7.3.1.3 crond 服务管理

执行如下命令启动 crond 服务：

```
#/sbin/service crond start
```

执行如下命令停止 crond 服务

```
#/sbin/service crond stop
```

7.3.2 at 和 batch

cron 被用来调度重复的任务，at 命令被用来在指定时间内调度一次性的任务。batch 命令被用来在系统平均载量降到 0.8 以下时执行一次性的任务。

要使用 at 或 batch 命令，您必须安装了 at RPM 软件包，并且 atd 服务必须在运行。要判定该软件包是否被安装了，使用 ***rpm -q at*** 命令。要判定该服务是否在运行，使用 ***/sbin/service atd status*** 命令。**配置 at 作业**

要在某一指定时间内调度一项一次性作业，键入 ***at time*** 命令。这里的 time 是执行命令的时间。

time 参数可以是下面格式中任何一种：

HH:MM 格式：例如，04:00 代表 4:00AM。如果时间已过，它就会在第二天的这一时间执行。

midnight：代表 12:00AM。

noon：代表 12:00PM。

teatime：代表 4:00PM。

英文月名 日期 年份 格式：例如，January 15 2002 代表 2002 年 1 月 15 日。年份可有可无。

MMDDYY、MM/DD/YY、或 MM.DD.YY 格式：例如，011502 代表 2002 年 1 月 15 日。

now + 时间：时间以 minutes、hours、days、或 weeks 为单位。例如，now+5 days 代表命令应该在 5 天之后的此时此刻执行。

时间必须要被先指定，接着是可有可无的日期。关于时间格式的详情，请阅读/usr/share/doc/at-<version>/timespec 文本文件。

键入了 ***at*** 命令和它的时间参数后，***at***>提示就会出现。键入要执行的命令，按【Enter】键，然后键入 ***Ctrl-D***。您可以指定多条命令，方法是键入每一条命令后按【Enter】键。键入所有命令后，按【Enter】键转入一个空行，然后再键入 ***Ctrl-D***。或者，您也可以在提示后输入 Shell 脚本，在脚本的每一行后按【Enter】键，然后在空行处键入 ***Ctrl-D*** 来退出。如果输入的是脚本，所用的 Shell 就会是用户的 Shell 环境变量中设置的值，用户的登录 Shell，或是/bin/sh（使用最先发现的）。

如果这组命令或脚本试图在标准输出中显示信息，该输出会用电子邮件方式被邮寄给用户。

使用命令 atq 来查看等待运行的作业。详情请参阅第 7.3.2.3 节。

at 命令的用法能够被制约。详情请参阅第 7.3.2.5 节。

7.3.2.2 配置 batch 作业

要在系统平均载量降到 0.8 以下时执行某项一次性的任务，使用 ***batch*** 命令。

键入 ***batch*** 命令后，at>提示就会出现。键入要执行的命令，按【Enter】键，然后键入 ***Ctrl-D***。您可以指定多条命令，方法是键入每一条命令后按【Enter】键。键入所有命令后，按【Enter】键转入一个空行，然后再键入 ***Ctrl-D***。或者，您也可以在提示后输入 Shell 脚本，在脚本的每一行后按【Enter】键，然后 在空行处键入 ***Ctrl-D*** 来退出。如果输入的是脚本，所用的 Shell 就会是用户的 Shell 环境变量中设置的值，用户的登录 Shell，或是</bin/sh>（使用最先发现的）。系统平均载量一降到 0.8 以下，这组命令或脚本就会被执行。

如果这组命令或脚本试图在标准输出中显示信息，该输出会用电子邮件方式被邮寄给用户。

使用命令 atq 来查看等待运行的作业。详情请参阅第 7.3.2.3 节。

at 命令的用法能够被制约。详情请参阅第 7.3.2.5 节。

7.3.2.3 查看等待运行的作业

要查看等待运行的 ***at*** 和 ***batch*** 作业，使用 ***atq*** 命令。它显示一列等待运行的作业，每项作业只占据一行。每一行的格式都是：作业号码、日期、小时、作业类别、以及用户名。用户只能查看他们自己的作业。如果根用户执行 ***atq*** 命令，所有用户的全部作业都会被显示。

7.3.2.4 其他的命令行选项

at 和 ***batch*** 的其它命令行选项包括：

选项	描述
-f	从文件中读取命令或 Shell 脚本，而非在提示后指定它们。
-m	在作业完成后，给用户发送电子邮件。
-v	显示作业将被执行的时间。

7.3.2.5 控制对 at 和 batch 的使用

/etc/at.allow 和/etc/at.deny 文件可以用来限制对 ***at*** 和 ***batch*** 命令的使用。这两个使用控制文件的格式都是每行一个用户。两个文件都不允许使用空白字符。如果使用控制文件被修改了，at 守护进程（atd）不必被重启。每次用户试图执行 ***at*** 或 ***batch*** 命令时，使用控制文件都会被读取。

不论使用控制文件如何规定，根用户都总是可以执行 ***at*** 和 ***batch*** 命令。

如果 at.allow 文件存在，只有其中列出的用户才能使用 at 或 batch 命令，at.deny 文件会被忽略。

如果 at.allow 文件不存在，所有在 at.deny 文件中列出的用户都被禁止使用 ***at*** 和 ***batch*** 命令。

7.3.2.6 启动和停止 at 服务

要启动 at 服务，使用***/sbin/service atd start*** 命令。要停止该服务，使用***/sbin/service atd stop*** 命令。建议您在引导时启动该服务。关于在引导时自动启动 at 服务的详情，请参阅其他相关介绍。

7.4 服务管理

systemd 是 CGSLV6 的系统和服务管理程序，替换了之前的发行本中使用的 SysV。

systemd 与 SysV 和 Linux 标准基本 init 脚本兼容。

systemd 比其他程序有下列优势：

1、强大的平行化功能；

2、使用插槽和 D-Bus 激活启动服务；

3、按需启动守护进程；

4、管理控制组；

5、生成系统状态快照及恢复系统状态。

systemd 组件中 systemctl 代替原来的 service 与 chkconfig 命令。

7.4.1 systemctl 命令

使用 systemctl 命令可以使某些服务立即启动/停止/重启/启用/禁用，使用如下命令：

1. 启动：

```
# systemctl start <服务>
```

2. 停止：

```
# systemctl stop <服务>
```

3. 重启：

```
# systemctl restart <服务>
```

4. 启用：

```
# systemctl enable <服务>
```

5. 禁用：

```
# systemctl disable <服务>
```

替换的服务例子：

原命令	现命令	备注
service httpd start	systemctl start httpd.service	启动 httpd 服务
service httpd stop	systemctl stop httpd.service	关闭 httpd 服务
service httpd restart	systemctl restart httpd.service	重启 httpd 服务

service httpd reload	systemctl reload httpd.service	重新驱动服务配置（须服务支持）
service httpd condrestart	systemctl condrestart httpd.service	重启 httpd 服务(如原服务是关闭状态，则命令无效)
service httpd status	systemctl status httpd.service	打印 httpd 服务运行状态
ls /etc/rc.d/init.d/	systemctl 或 systemctl list-unit-files --type=service 或 ls /lib/systemd/system/*.service 或 ls /etc/systemd/system/*.service	列出所有的服务和单元
chkconfig httpd on	systemctl enable httpd.service	开启 httpd 服务随机启动，在下次重启生效，或切换运行级别
chkconfig httpd off	systemctl disable httpd.service	关闭 httpd 服务随机启动，在下次重启生效，或切换运行级别
chkconfig httpd	systemctl is-enabled httpd.service	打印 httpd 服务的随机启动设置状态
chkconfig --list	systemctl list-unit-files --type=service(preferred) ls /etc/systemd/system/*.wants/	列出所有服务的设置（启动或关闭）
chkconfig httpd --list	ls etc/systemd/system/*.wants/httpd.service	列出 httpd 服务在所有运行级别中的设置
chkconfig httpd --add	systemctl daemon-reload	创建新配置文件或者修改配置文件后重新读取

7.5 内核模块管理

7.5.1 概述

CGSL 内核具有模块化设计。在引导时，只有少量的驻留内核被载入内存。这之后，无论何时用户要求使用驻留内核中没有的功能，某内核模块（kernel module），有时又称驱动程序（driver）就会被动态地载入内存。

在安装过程中，系统上的硬件会被探测。基于探测结果和用户提供的信息，安装程序会决定哪些模块需要在引导时被载入。安装程序会设置动态载入机制来透明地运行。

也可以通过编辑模块配置文件/etc/modprobe.conf 来手工指定这个硬件使用的模块。

例如，如果某系统包括了一个 SMC EtherPower 10 PCI 网卡，模块配置文件包含以下行：

```
alias eth0 tulip
```

如果系统上添加了第二个网卡，它和第一个网卡一模一样，可在/etc/modprobe.conf 中添加这一行：

```
alias eth1 tulip
```

7.5.2 内核模块工具

CGSL 提供了一组管理内核模块的命令。使用这些命令来判定模块是否被成功地载入了，或为新硬件试验不同的模块。

7.5.2.1 lsmod 命令

lsmod 命令显示了当前载入了的模块列表。例如：

```
[root@localhost ~]# lsmod
Module                  Size  Used by
fuse                   55488  2
vboxvideo               1168  1
drm                   161879  2 vboxvideo
vboxsf                 35719  0
autofs4                21636  3
sunrpc                198287  1
ipv6                  264062  30
dm_mirror              11748  0
dm_region_hash         10191  1 dm_mirror
dm_log                  8520  2 dm_mirror,dm_region_hash
register_ipmc_reboot     2399  0
uinput                  5962  0
ppdev                   7367  0
parport_pc             19476  0
parport                30987  2 ppdev,parport_pc
sg                     24618  0
i2c_piix4              11130  0
i2c_core               25895  2 drm,i2c_piix4
snd_intel8x0           24767  15
snd_ac97_codec         96862  1 snd_intel8x0
ac97_bus                 942  1 snd_ac97_codec
snd_seq                45323  0
```

图 7-3 当前模块列表

对每行而言，第一列是模块名称；第二列是模块大小；第三列是用量计数。

lsmod 输出和查看/proc/modules 的输出相同。

7.5.2.2 modprobe 命令

要载入内核模块，使用 modprobe 命令，然后跟着内核模块的名称。按照默认设置，modprobe 试图从</lib/modules/<kernel-version>/kernel/drivers/>子目录中载入模块。每类模块都有一个子目录，如用于网络接口驱动程序的<net/>子目录。某些内核模块有模块依赖关系，这意味着我们必须首先载入其它模块才能载入这些模块。modprobe 命令检查这些依赖关系，并在载入指定模块前载入满足这些依赖关系的模块。

例如：

```
#modprobe udf
```

这个命令载入任何满足依赖关系的模块，然后再载入 hid 模块。

要在 modprobe 执行命令的时候把它们都显示在屏幕上，使用-v 选项。例如：

```
#modprobe -v udf
```

所显示的输出和下面相似：

```
#insmod /lib/modules/4.18.0-147.3.1.el8_1.x86_64/kernel/lib/crc-itu-t.ko.xz
#insmod /lib/modules/4.18.0-147.3.1.el8_1.x86_64/kernel/fs/udf/udf.ko.xz
```

7.5.2.3 insmod 命令

还可以使用 insmod 命令来载入内核模块；不过它不解决依赖关系。因此，推荐我们使用/sbin/modprobe 命令。

7.5.2.4 rmmod 命令

要卸载内核模块，使用 rmmod 命令和模块名称。rmmod 工具只卸载不再使用的、和不是正被使用的模块所依赖的模块。

例如：

```
#rmmod udf
```

这个命令卸载 udf 内核模块。

7.5.2.5 modinfo 命令

使用 modinfo 命令来显示关于内核模块的信息。一般语法是：

```
#modinfo [options] <module>
```

包括-d 在内的选项显示了关于模块的简短描述，-p 选项列举了模块所支持的参数。要获取选项的完整列表，请参阅 modinfo 的说明书页（man modinfo）。

7.6 Kdump

Kdump 可以在系统发生崩溃的时候，转储故障现场供事后分析定位。CGSL V6 默认已开启 Kdump 功能。请查看</boot/grub2/grubenv>文件，启动参数中应有 crashkernel=auto 参数，如下：

```
# cat /boot/grub2/grubenv
# GRUB Environment Block
saved_entry=84bd0b8fd13545c58b25f182c7dfd296-4.18.0-147.8.1.el8_1.x86_64
kernelopts=root=/dev/mapper/ncl-root ro fsckroot crashkernel=auto
resume=/dev/mapper/ncl-swap rd.lvm.lv=ncl/root rd.lvm.lv=ncl/swap rhgb quiet
boot_success=0
```

可执行如下命令检查 Kdump 服务是否开启：

```
# systemctl status kdump.service
```

当系统出现内核 panic 时，系统会自动重启（重启过程会完成内核转储，时间比正常重启稍长，请耐心等待）。重启完成后会在/var/crash/目录中生成相应的 vmcore 文件。

7.7 系统信息收集

在学习如何配置系统之前，应该学习如何收集基本的系统信息。譬如，应该知道如何找出关于空闲内存的数量、可用硬盘驱动器空间的数量，硬盘分区方案，以及正在运行进程的信息。

本节将介绍如何使用几个简单程序来从 CGSL 系统中检索这类信息。

7.7.1 进程信息

ps 命令显示一个当前系统进程的列表，常用的命令(带选项)为 ps aux 和 ps -ef 命令。命令结果列表是一个静态列表，即在启用这项命令时正在运行的进程的快照。如果您需要一个时刻更新的运行进程列表，可使用下面描述的 top 命令。

ps 的输出会很长。要防止它快速从屏幕中滑过，您可以把它管道输出给 less 命令：

```
#ps aux | less
```

也可以使用 ps 命令和 grep 命令的组合来查看某进程是否在运行。譬如，要判定 Emacs 是否在运行，使用下面这个命令：

```
#ps aux | grep emacs
```

top 命令显示了当前正运行的进程以及关于它们的重要信息，包括它们的内存和 CPU 用量。该列表既是真实时间的也是互动的。以下提供了一个 top 的输出示例：

```
root@localhost:~
File  Edit  View  Search  Terminal  Help
top - 13:05:16 up 5 min,  2 users,  load average: 0.36, 0.35, 0.16
Tasks: 149 total,   1 running, 148 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  0.7%sy,  0.0%ni, 99.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    511508k total,   352676k used,   158832k free,    28940k buffers
Swap:  1048568k total,        0k used,  1048568k free,   163112k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1622 root      20   0 36264  19m 6636 S  1.0  3.8   0:05.22 Xorg
 1714 root      20   0  5784 2392 1992 S  0.3  0.5   0:00.16 devkit-power-da
 1944 root      20   0 21448 9204 7864 S  0.3  1.8   0:00.20 metacity
 2394 root      20   0  2660 1104  848 R  0.3  0.2   0:00.32 top
    1 root      20   0  2828 1368 1164 S  0.0  0.3   0:01.76 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.03 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.04 migration/1
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/1
    8 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.01 events/0
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 events/1
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 cpuset
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns
```

图 7-4 top 显示

要退出 top，按[q]键。

可以和 top 一起使用的互动命令包括：

表 7-1 互动的 top 命令

命令	描述
[Space]	立即刷新显示
[h]	显示帮助屏幕
[k]	杀死某进程，会被提示输入进程 ID 以及要发送给它的信号
[n]	改变要显示的进程数量，会被提示输入数量
[u]	按用户排序
[M]	按内存用量排序
[P]	按 CPU 用量排序

♣ 提示：按默认设置，ps 和 top 只显示进程信息，如果要查看所有线程信息，请使用 ps -elLf 命令或在 top 中键入[Shift]-[H]组合键。

另外，CGSL 还提供了图形化的系统监视器。要从桌面上启动它，选择面板上的【系统】->【管理】->【系统监视器】或在图形环境中的 Shell 提示下键入 gnome-system-monitor。然后选择【进程】标签，如图 7-5 所示。

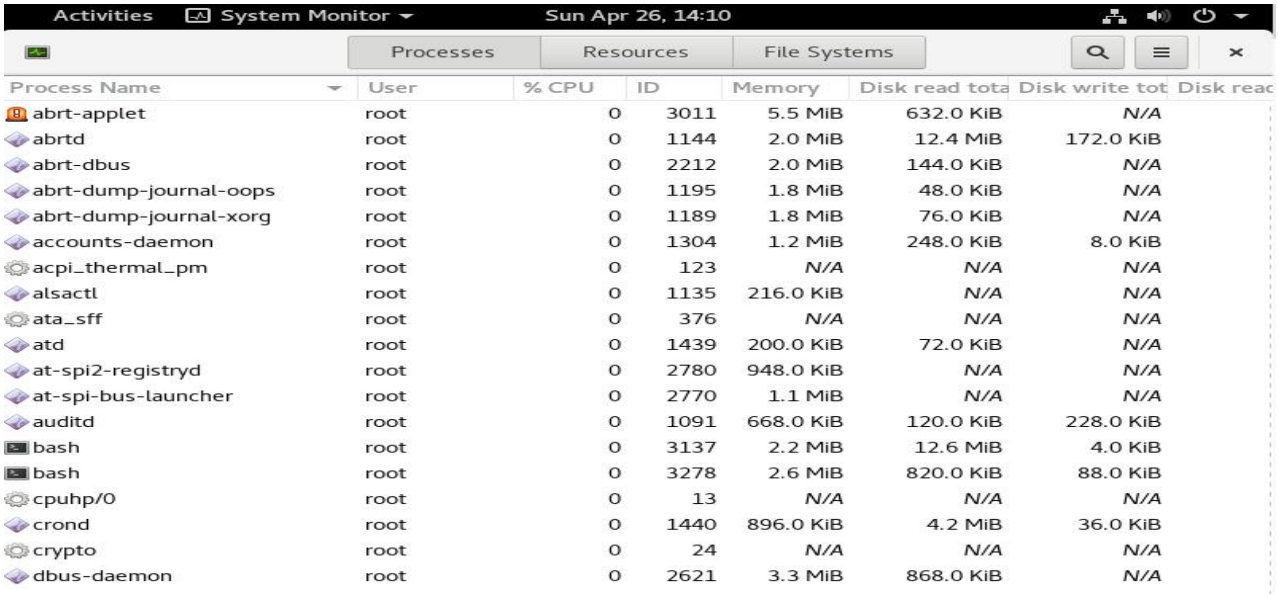


图 7-5 GNOME 系统监视器（1）

系统监视器允许在正运行的进程列表中通过【查看】里的各标签搜索进程，包括查看所有进程、您拥有的进程、或活跃的进程。 要停止某进程，选择该进程，然后点击【结束进程】。这有助于结束对用户输入已不再做出反应的进程。

若要按指定列的信息来排序，点击该列的名称。信息被排序的那一列会用深灰色显示。

若需要要改变 GNOME 系统监控器的默认设置，选择【编辑】->【首选项】，点击【进程】和【资源】标签，可以允许配置更新间隔，每个进程默认显示的信息，以及系统监视器图表的颜色。

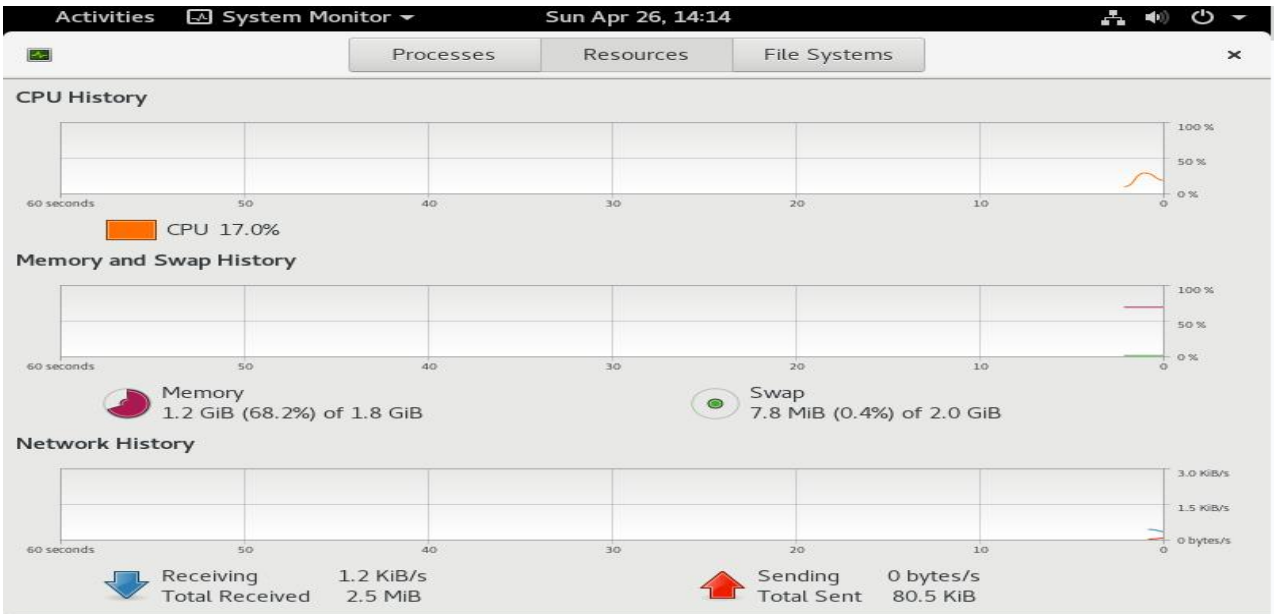
7.7.2 内存信息

free 命令显示系统的物理内存和交换区的总量，以及已使用的、空闲的、共享的、在内核缓冲内的、和被缓存的内存数量。如下命令结果中单位为 KB。

```
[root@localhost ~]# free
             total       used       free     shared    buffers     cached
Mem:        511508     487800      23708          0      80604     216780
-/+ buffers/cache:     190416     321092
Swap:      1048568          0    1048568
```

图 7-6 free 命令行显示

♣ 提示：free 命令第 3 行中的 used=第 2 行的 used-buffers-cached，第 3 行中的 free=第 2



行的 free+buffers+cached。因为 buffers 和 cached 为系统缓存，用于提高系统效率，通常可在需要时由系统自动回收，所以，通常应该以第 3 行的 used 和 free 表示当前系统内存实际使用的情况。

另外，CGSL 提供的图形化的系统监视器(上节介绍)工具中也可以查看内存信息，在系统监视器工具中选择【资源】标签，如图 7-7 所示。

图 7-7 GNOME 系统监视器（2）

7.7.3 文件系统信息

df 命令报告系统的磁盘空间用量。如果在 Shell 提示下键入了 df 命令，它的输出与下

面相似：

```
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda2 10325716 2902060 6899140 30% /
/dev/hda1 15554 8656 6095 59% /boot
/dev/hda3 20722644 2664256 17005732 14% /home
none 256796 0 256796 0% /dev/shm
```

按照默认设置，已用的和可用的磁盘空间以 KB 为单位显示。如需以易读方式显示，可以使用 df -h 命令。输出类似于：

```
Filesystem Size Used Avail Use% Mounted on
/dev/hda2 9.8G 2.8G 6.5G 30% /
/dev/hda1 15M 8.5M 5.9M 59% /boot
/dev/hda3 20G 2.6G 16G 14% /home
none 251M 0 250M 0% /dev/shm
```

在分区列表中，有一项是/dev/shm。该条目代表系统的虚拟内存文件系统。

du 命令显示被目录中的文件使用的估计空间数量。如果在 Shell 提示下键入了 du 命令，每个子目录的用量都会在列表中显示，当前目录和子目录的总和也会在列表的最后一行中被显示。如果我们不想查看每个子目录的用量，使用 du -hs 命令来使用人可读的格式只列出目录用量总和。使用 du -help 命令来查看更多选项。

要查看图形化的系统分区和磁盘空间用量，使用【系统监视器】->【file systems】标签，如图 7-8 所示。

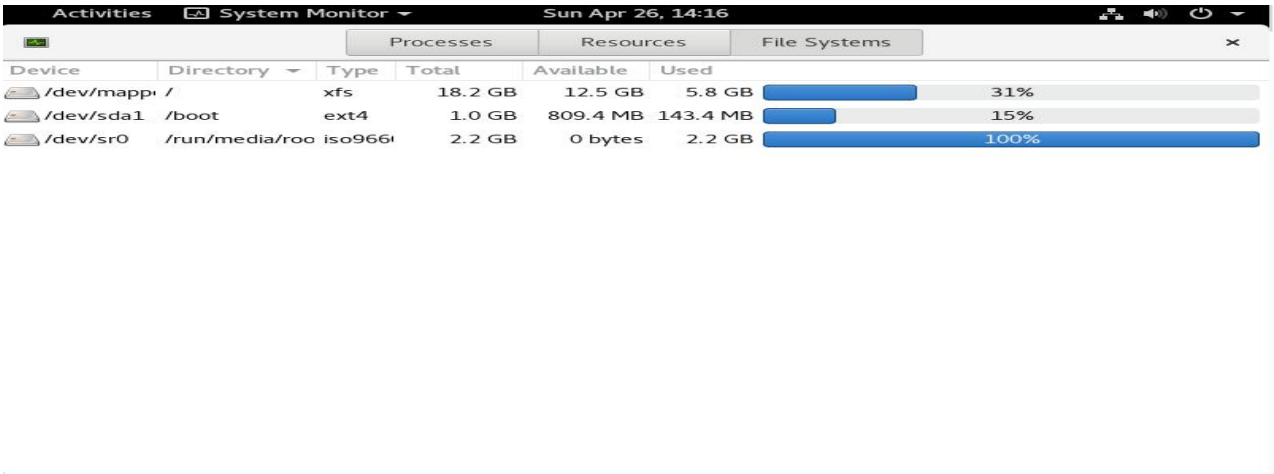


图 7-8 GNOME 系统监视器（3）

第 8 章

系统服务

CGSL 系统为标准服务器版本，为满足大多数用户的需求，系统提供了大量的通用服务和基本功能，本章主要介绍常用服务和系统基本功能的配置和使用。

8.1 NTP

NTP(Network Time Protocol)是用来使系统和一个精确的时间源保持时间同步的协议。通常用于局域网上的若干台主机通过互联网与其他的 NTP 主机同步时钟，接着再向局域网内其他客户端提供时间同步服务。

CGSLV6 默认 ntp 组件由 Chrony 代替。

Chrony 是网络事件协议的(ntp)的另一种实现，与网络事件协议后台程序（ntpd）不同，它可以更快地且更准确地同步系统时钟。请注意，ntpd 仍包含其中以提供需要运行 NTP 服务的客户使用(需要重新从安装光盘安装)。

注意：Chrony 不能使用本地时间作为时钟源（原 ntpd 可以使用本地时间作为时钟源）。

Chrony 的优势包括：

- 更快的同步只需要数分钟并非数小时时间，从而最大程度减少了时间和频率的误差，这对于并非全天 24 小时运行的台式计算机或系统而言非常有用。
- 能够更快地响应时钟频率的快速变化，这对于具备不稳定始终的虚拟机或导致始终频率发生变化的节能技术而言非常有用。
- 在初始同步后，它不会停止时钟，以防对需要系统时间保持单调的应用程序造成影响。
- 在应对临时非对称延迟时（例如，在大规模下载造成链接饱和时）提供了更好的稳定性。

- 无需对服务器进行定期轮询，因此具备间歇性网络连接的系统仍然可以快速同步时钟。

8.1.1 Chrony 配置文件

Chrony 服务主要通过/etc/chrony.conf 配置。本节主要介绍该配置文件中的主要影响配置。

8.1.1.1 上级时间服务器

首先用 server 这个参数设定上级时间服务器，语法为：

```
server IP 地址或域名 [prefer]
```

8.1.1.2 权限设置

权限的设定，主要的语法为：

```
allow IP 地址 mask 子网掩码
```

8.1.1.3 driftfile 参数

在与上级时间服务器联系时所花费的时间，记录在 driftfile 参数后面的文件内。语法为：

```
driftfile 包含路径的文件名
```

注意: driftfile 后面接的文件需要使用完整的路径文件名，不能是链接文件，并且文件的权限需要设定成 ntpd 守护进程可以写入。

8.1.2 NTP 配置实例

下面以本地配置一个服务器与客户端为例介绍 CGSL NTP 的配置过程。

8.1.2.1 服务器端配置

1、首先查询 chrony 软件版本

```
#rpm -q chrony
chrony-3.3-3.el8.x86_64
```

如果没有，可以从 CGS L 安装盘上查找，找到后用 rpm 命令安装，示例如下：

```
# rpm –ivh chrony-3.3-3.el8.x86_64.rpm
```

2、编辑配置文件

```
#vi /etc/ chrony.conf
```

修改配置文件/etc/chrony.conf，使用 192.168.17.234 为上层时钟源，并对下级客户端 192.168.17 网段开放权限

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 192.168.17.234 iburst 配置上层时钟源，不能为本机

# Ignore stratum in source selection.
stratumweight 0
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift  配置 drift 文件路径

# Enable kernel RTC synchronization.
rtcsync  自动同步系统时间到硬件时间

# In first three updates step the system clock instead of slew
# if the adjustment is larger than 10 seconds.
makestep 10 3

# Allow NTP client access from local network.
#allow 192.168/16
allow 192.168.17/24 配置下层客户端的访问权限

# Listen for commands only on localhost.
bindcmdaddress 127.0.0.1
bindcmdaddress ::1

# Serve time even if not synchronized to any NTP server.
#local stratum 10

keyfile /etc/chrony.keys
```

```
# Specify the key used as password for chronyc.
 commandkey 1
# Generate command key if missing.
generatecommandkey

# Disable logging of client accesses.
noclientlog

# Send a message to syslog if a clock adjustment is larger than 0.5 seconds.
logchange 0.5

logdir /var/log/chrony
#log measurements statistics tracking
```

3、重启 chrony 服务

```
# systemctl restart chronyd
```

4、查看 ntp 服务状态

如果同步正常，显示如下：

```
# chronyc sources -v
210 Number of sources = 1

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /    '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||                                                /    xxxx = adjusted offset,
||          Log2(Polling interval) -.             |     yyyy = measured offset,
||                                   \            |     zzzz = estimated error.
||                                    |           |
MS Name/IP address          Stratum Poll Reach LastRx Last sample
===============================================================
==============
^* 192.168.17.234                 6    7    377    128   -1712ns[-2000ns] +/-
11ms
```

8.1.2.2 客户端配置

1、首先查询 chrony 软件包

```
#rpm -q chrony
chrony-3.3-3.el8.x86_64
```

如果没有可以从 CGS Linux 安装盘上查找，找到后用下面的命令安装：

```
# rpm –ivh chrony-3.3-3.el8.x86_64.rpm
```

2、编辑/etc/ chrony.conf，假设上层时钟源为 192.168.17.215，则保留如下内容：

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 192.168.17.215 配置上层时钟源，不能为本机



# Ignore stratum in source selection.
stratumweight 0

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift 配置 drift 文件路径

# Enable kernel RTC synchronization.
rtcsync 自动同步系统时间到硬件时间

# In first three updates step the system clock instead of slew
# if the adjustment is larger than 10 seconds.
makestep 10 3

# Allow NTP client access from local network.
# Listen for commands only on localhost.
bindcmdaddress 127.0.0.1
bindcmdaddress ::1

# Serve time even if not synchronized to any NTP server.
#local stratum 10
```

```
keyfile /etc/chrony.keys

# Specify the key used as password for chronyc.
commandkey 1

# Generate command key if missing.
generatecommandkey

# Disable logging of client accesses.
noclientlog

# Send a message to syslog if a clock adjustment is larger than 0.5 seconds.
logchange 0.5

logdir /var/log/chrony
#log measurements statistics tracking
```

3、、重启 chrony 服务：

```
# systemctl restart chronyd
```

4、检查 ntp 服务状态

结果如下则同步正常：

```
# chronyc sources -v
210 Number of sources = 1

  .-- Source mode   '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /    '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||                                                /    xxxx = adjusted offset,
||            Log2(Polling interval) -.          |     yyyy = measured offset,
||                                     \         |     zzzz = estimated error.
||                                      |        |
MS Name/IP address          Stratum Poll Reach LastRx Last sample
```

```
==================================================================
==============
^* 192.168.17.215                  7    7    377     92     -22us[   -27us] +/-
10ms
```

如果输出和上面相似，则说明时间服务器成功，服务启动后，会自动同步时间。

8.2 vsftpd

vsftpd 的全称为"Very Secure FTP Daemon"。是 CGSL 中自带的默认 FTP 服务，使用 vsftpd 可以构建一个以安全为重的 FTP 服务器。

8.2.1 vsftpd 配置文件

vsftpd 是通过/etc/vsftpd/vsftpd.conf 文件来进行配置的。本节主要介绍该配置文件中的主要配置。 **主要参数说明**

```
anonymous_enable=YES   # 允许匿名登录
local_enable=YES   # 允许本地登录
write_enable=YES   # 允许用户上传数据
local_umask=022   # 设置默认的掩码为 022
dirmessage_enable=YES   # 当用户进入某个目录时，会显示该目录需要注意的内容
xferlog_enable=YES   # 默认上传或者下载的日志被记录在/var/log/vsftpd.log 中
connect_from_port_20=YES   # 用 20 端口作为数据传输端口
xferlog_std_format=YES   # 使用标准格式等级上传或者下载记录
listen=YES   # 当设为 yes 时，vsftpd 以 standalon 状态运行，默认开启
pam_service_name=vsftpd   # 列出与 vsftpd 相关的 PAM 文件
userlist_enable=YES
# 当该选项设为 yes 时，启用配置文件/etc/vsftpd/user_list
# 1：若此时没有 userlist_deny=NO，则/etc/vsftpd/user_list 中用户不能访问 ftp
# 2：若存在 userlist_deny=NO，则仅接受/etc/vsftpd/user_list 中存在用户登录 ftp
的请求（前提是这些用户不存在于/etc/vsftpd/ftpusers 中）
#    当为 NO 时，不启用/etc/vsftpd/user_list 配置文件
tcp_wrappers=YES   # 启用 TCP Wrapper 支持
sendport_off_enable=NO   # 不使用 sendport 模式
```

8.2.1.2 其它常用设置

- guest_enable=YES
- guest_username=ftp

guest 用户名，即所有非匿名用户将具有 guest 用户身份。

- local_root=/var/ftp
- anon_root=/var/ftp

设定本机用户和匿名用户的主目录

- pasv_enable=YES

#port_enable=YES

port 为主动模式，pasv 为被动模式，两个不能同时使用，必须注释掉一个

- pasv_min_port=9000
- pasv_max_prot=9200

使用被动模式时端口的范围，例为 9000-9200，只有在被动模式下有用

- use_localtime=YES

使用使用本地时间，如不使用，则使用格林威治时间，建议用 YES

- accept_timeout=60

被动模式下服务器等待客户端的延时时间 单位为秒

- max_clients=0

在 standalone 模式下最大客户连接数

- max_pre_ip=0

每个客户端的最大连接数

- local_max_rate=0

本地用户最大传输速率，单位为字节/秒；0 为不限

- anon_max_rate=0

匿名用户最大传输速率，单位为字节/秒；0 为不限

- chroot_local_user=YES

- chroot_list_enable=YES
- chroot_list_file=/etc/vsftpd/chroot_list

锁定用户主目录的设置 如果 chroot_list_enable=YES 时，用户主目录锁定，在 ftp 时切换到主目录是就切换到主目录下，不会访问主目录的上层目录

/etc/vsftpd/chroot_list 的格式为：用户名 锁定的目录

其他参数可使用命令 man vsftpd.conf 进行查阅。

8.2.2 vsftpd 配置实例

下面以一个实例介绍 vsftpd 服务器的配置。

vsftpd 配置要求如下：

- 不允许匿名登录；
- 能使用 root 或 test01 用户登录；
- test01 用户需锁定用户主目录为/home/test01

1. 首先查询 vsftpd 软件的版本

```
# rpm -q vsftpd
vsftpd-3.0.3-28.el8.x86_64
```

如果没有可以从 CGS Linux 安装盘上查找，找到后执行 rpm 命令安装，示例如下：

```
# rpm -ivh vsftpd-3.0.3-28.el8.x86_64.rpm
```

2. 编辑/etc/vsftpd/vsftpd.conf 文件：

1）将 anonymous_enable=YES 修改为 anonymous_enable=NO

2）在文件末添加：

```
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
```

3. 为了能使 root 用户登录，编辑/etc/vsftpd/ftpuser 和/etc/vsftpd/user_list 文件：

将 root 行用”#”注释掉

4. 编辑或新建/etc/vsftpd/chroot_list 文件，加入：

```
test01   /home/test01
```

5. 重启 vsftpd 服务：

```
# systemctl restart vsftpd
```

6. 使用过程中可能遇到由于 SELinux 会防止某些服务访问用户的主目录的问题，错误提示如：

```
500 OOPS: cannot change directory:/root
```

可执行如下命令关闭 SELinux 解决：

```
# setenforce 0
```

8.3 Samba

Samba 服务是为 CGSL 系统和 Windows 系统之间提供文件共享功能的一种服务。

8.3.1 Samba 配置文件

Samba 服务的主要配置文件为/etc/samba/smb.conf，如下介绍该配置文件中的基本配置：

```
[global]   #全局设定
 workgroup = MYGROUP
 server string = Samba Server Version %v
 log file = /var/log/samba/log.%m
 max log size = 50

 security = user
# 设置安全级别，即客户端访问 Samba 服务器的验证方式。
# 此部分中只能设置以下三种参数，参数设置：
# share  不需要提供用户名和密码
# user   只能被授权用户访问，由 Samba Server 负责检查账号和密码的有效性。账号和密码要在本 Samba Server 中建立
# server 依靠其他 Windows 或 Samba Server 来验证用户的账号和密码，是一种代理验证
 passdb backend = tdbsam
# 设定 Samba 用户密码的存放方式
# tdbsam：该方式使用一个数据库文件来建立用户数据库，数据库文件名为 passdb.tdb。可以使用#smbpasswd -a [用户名]来建立 Samba 用户。也可以使用
```

```
pdbedit 命令来建立用户。
# smbpasswd：该方式使用 Samba 提供的工具 smbpasswd 来给系统用户设置一个用于访问 Samba 服务的密码，客户端就用这个密码访问 Samba 共享资源。此方式还要使用一个 smb passwd file = /usr/local/samba/etc/smbpasswd（或 /etc/samba/smbpasswd）参数来指定保存用户名和密码的文件，该文件需要手动建立。
# ldapsam：该方式基于 LADP 的账户管理方式来验证用户，先要建立 LDAP 服务。
# mysql：该方式是将 Samba 服务器的用户名和密码存储到 MySQL 数据库中。
load printers = yes
 cups options = raw

[homes]                          # 用户个人主目录设置
 comment = Home Directories          # 主目录注释
 browseable = no                  # 是否允许其他用户浏览个人主目录
 writable = yes                   # 是否允许写主目录

[printers]
 comment = All Printers
 path = /var/spool/samba
 browseable = no
 guest ok = no
 writable = no
 printable = yes
```

更多参数设置请使用命令 man smb.conf 进行查阅。

8.3.2 Samba 配置实例

通过一个实例介绍 Samba 服务器的配置。

Samba 服务器的配置要求：

- 共享的目录为/data
- 只有 root 用户才能访问
- 可上传和下载

1. CGSL V6 默认安装 Samba 服务端，使用下面命令查询是否正确安装了 Samba：

```
# rpm -q samba
samba-4.10.4-101.el8_1.x86_64
```

2. 编辑/etc/samba/smb.conf 文件，在文件末尾添加：

```
[root]
 path = /data
 writable = yes
 valid users = root
```

3. 增加 Samba 服务的 root 账号：

```
# smbpasswd -a root
```

按提示输入密码。

4. 启动 Samba 服务：

```
# systemctl start smb
```

5. 如果共享目录为 SELinux 所保护的目录，需执行如下命令关闭 SELinux：

```
# setenforce 0
```

8.4 NFS

NFS (Network File System，网络文件系统)，可以通过网络让不同的机器、不同的操作系统彼此共享文件，最早由 Sun 公司开发。它是 CGSL 下常用的文件共享服务。

NFS 服务可以将网络上的 NFS 主机共享的目录挂载到本地，在本地端看来，被挂载的远程主机目录就好像本地目录一样，使用起来非常方便。

8.4.1 NFS 配置文件

NFS 的主要配置文件为/etc/exports，其中包括需要共享的目录（绝对路径），以及访问控制和共享参数。/etc/exports 文件中的一行表示一个共享目录，规范如下：

```
要共享的目录 可访问的主机(访问权限) 第二个可访问的主机(访问权限) ……
```

主机的设置：

具体的 IP 地址	如 172.16.100.138
一个网段	如 172.16.100.0/24 或 172.16.100.0/255.255.255.0

具体的主机名	如 host01，需要在 hosts 或 dns 中定义并能够解析到
*	表示匹配所有可能值

常用的权限参数的设置：

ro	只读
rw	可读可写
sync	数据同步写入内存和磁盘中（默认）
async	数据先暂存于内存中，而非直接写入磁盘
root_squash	将登入 NFS 主机的 root 身份映射为匿名用户（默认）
no_root_squash	登入 NFS 主机的 root 将获得 root 权限
all_squash	将登入 NFS 主机的所有账号都映射为匿名用户
anonuid=Num	将登入者的 UID 映射为 Num
anongid=Num	将登入者的 GID 映射为 Num

8.4.2 NFS 配置实例

以下以一个例子介绍 NFS 的配置。

NFS 配置要求：

- 一台主机(IP:172.16.100.78)要共享的目录为：/data
- 只有 172.16.100.0/24 和 192.168.1.0/24 这两个网段的主机才能访问
- 172.16.100.0/24 的用户具有可读可写，以及数据同步权限
- 172.16.100.0/24 的用户具有只读，以及数据同步权限

1. 首先检查下 NFS 软件的版本：

```
# rpm -q nfs-utils
nfs-utils-2.3.3-14.el8_0.2.x86_64
```

如果没有可以从 CGS Linux 安装盘上查找，找到后用下面的命令安装（为 32 位系统为例）：

```
# rpm -ivh nfs-utils-2.3.3-14.el8_0.2.x86_64.rpm
```

2. 编辑/etc/exports 文件，添加如下内容：

```
/data 172.16.100.0/24(rw,sync) 192.168.1.0/24(ro,sync)
```

3. 启动 NFS 服务：

```
# systemctl start nfs
```

4. 客户端也需启动 NFS 服务，之后才可通过以下命令访问 NFS 服务器：

```
#mount -t nfs172.16.100.78:/data /mnt
```

8.5 Telnet

Telnet 允许用户登录远程计算机，就像登录本地服务器一样。但是由于它是采用明文传输信息，安全性不好，所以默认情况下 Telnet 服务是关闭的。

8.5.1 Telnet 服务的启动

1.首先检查下 telnet 软件的版本：

```
#rpm -q telnet-server
telnet-server-0.17-73.el8.x86_64
```

如果没有可以从 CGS Linux 安装盘上查找，找到后用下面的命令安装：

```
#rpm -ivh telnet-server-0.17-73.el8.x86_64.rpm
```

1. 启动 telnet 服务：

```
# systemctl restart telnet.socket
```

2. Telnet 服务默认不允许 root 用户登录，要实现 root 登录，可执行如下命令：

```
#mv -v /etc/securetty /etc/securetty.bak
```

3. 启动 telnet 服务：

```
#lsof -i:23
COMMAND PID USER    FD    TYPE DEVICE SIZE/OFF NODE NAME
```

```
systemd    1 root    44u   IPv6   38164       0t0   TCP *:telnet (LISTEN)
```

4. 设置随机启动：

```
# systemctl enable telnet.socket
```

8.5.3 Telnet 客户端

首先检查 telnet 客户端软件的版本

```
# # rpm -q telnet
telnet-server-0.17-73.el8.x86_64
```

如果没有可以从 CGS Linux 安装盘上查找，找到后用下面的命令安装

```
# rpm -ivh telnet-server-0.17-73.el8.x86_64.rpm
```

8.6 OpenSSH

SSH(Secure Shell protocol)，可以将数据加密后再进行数据传递，因此数据比较安全。OpenSSH 是 SSH 协议的免费开源实现。它用安全、加密的网络连接工具代替了 telnet、ftp、rlogin、rsh 和 rcp 工具。该协议默认使用 RSA 钥匙。

8.6.1.1 sshd 服务配置

CGSL V6 默认已经安装并配置了 OpenSSH，不需要其他手动配置即可直接使用，SSH 相关功能由 sshd 服务提供。查看 sshd 服务状态，使用命令：

```
# systemctl status sshd
```

CGSL V6 默认的 sshd 服务配置文件为/etc/ssh/sshd_config，主要配置如下：

```
Protocol 2                          # 使用协议版本 2
SyslogFacility AUTHPRIV             # 日志产生设备
PasswordAuthentication yes          # 使用口令认证
ChallengeResponseAuthentication no# 不允许盘问应答方式
GSSAPIAuthentication yes                # 允许使用基于 GSSAPI 的用户认证
GSSAPICleanupCredentials yes
UsePAM yes                      # 使用 PAM 认证
AcceptEnv    LANG    LC_CTYPE    LC_NUMERIC    LC_TIME    LC_COLLATE
LC_MONETARY LC_MESSAGES
```

```
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE
LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS                # 登录后设置的环境变量
X11Forwarding yes                   # 允许 X11 转发
Subsystem sftp /usr/libexec/openssh/sftp-server # sftp 子系统
```

8.6.1.2 使用 ssh 命令

ssh 命令是 riogin、rsh 和 telnet 命令的安全替换。它允许用户登录远程机器并在其上执行命令。ssh 登录远程机器和使用 telnet 相似。

例如：登录到一个名为 example.test.com 的远程主机，在 Shell 提示下键入下面的命令：

```
#ssh example.test.com
```

第一次使用 ssh 登录远程机器时，会看到和下面相仿的消息：

```
The authenticity of host ' example.test.com ' can't be established.
RSA key fingerprint is 94:68:3a:3a:bc:f3:ga:gb:01:sd:b3:07:38:eZ:11:oc
Are you sure you want to continue connecting ( yes / no ) ?
```

键入 ***yes*** 继续，接下来会询问用户在远程主机上的口令，这时系统会将远程主机的密钥加入到用户主目录下的 .ssh/hostkeys 中。正确输入口令后，就会在远程主机的 Shell 提示符下了。

如果登录时没有指定用户名，本地客户机登录远程机器时用的用户名就会被传递给远程机器。如果想指定不同的用户名，使用下面的命令：

```
#ssh username @ example.test.com
```

或

```
#ssh -l username example.test.com
```

♣ 提示：使用 ssh 后需要用“exit”退出登陆。

---

ssh 命令还可以不经登录而在远程机器上执行命令。它的语法格式是：

```
#ssh hostname command
```

例如，如果想查看远程主机 example.test.com 上</usr/share/apps>目录下的内容，在 Shell 提示下键入命令：

```
#ssh example.test.com ls /usr/share/apps
```

正确的输入口令之后，远程机器</usr/share/apps>目录下的内容就会被显示，然后返回到本地 Shell 提示下。

8.6.1.3 使用 scp 命令

scp 命令可以通过安全、加密的连接在机器间传输文件。它的使用与 rcp 相似。

把本地文件传输给远程机器的一般语法是：

```
#scp localfile username@remotehostname:/remotefile
```

localfile 指定本地源文件，username @remotehostname:/remotefile 指定远程目标文件。

要把本地文件 filel 传送到用户在 example.test.com 上的主目录中，在 Shell 提示下键入：

```
#scp filel username@example.test.com:/home/username
```

把远程文件传输给本地系统的一般语法是：

```
#scp username@remotehostname:/remotefile/localfile
```

remotefile 指定远程源文件，localfile 指定本地目标文件。

源文件可以由多个文件组成。例如，要把目录/downloads 的内容传输到远程机器 example.test.com 上现存的 uploads 目录，键入下列命令：

```
#scp /downloads/* username@example.test.com:/uploads/
```

8.6.1.4 使用 sftp 命令

sftp 命令是 ftp 命令的安全替换，用来扫开一次安全的 FTP 交互会话。它的使用与 ftp 相似，只不过，它使用安全、加密的连接。

Sftp 一般语法是：

```
#sftp username@hostname
```

一旦通过验证，就可以使用一组和 FTP 相似的命令。

提示：请参阅 sftp 手册获取这些命令的列表，通过 ***man sftp*** 进入 sftp 手册页。

第 9 章

系统安全

随着现代通信技术的迅速发展，Internet 使用范围不断扩大、用户人数也在不断增加，而 Internet 上任何一台计算机都可能成为网络黑客试图攻击的对象。对于企业和关键应用领域的服务器系统来说，安全问题就显得更为重要。本章主要介绍 CGSL 的系统安全管理策略。

9.1 系统安全概要

网络服务器作为 Internet/Intranet 上的关键设备，往往储存了大量的重要信息，或是向大量用户提供重要服务，一旦遭到破坏，后果是很严重的。所以网络建设者和管理员应认真对待安全方面的问题，以保证服务器的正常运行。

9.1.1 安全管理

CGSL 系统安全包括 3 个要素：物理安全管理、普通用户安全管理和超级用户安全管理。

物理安全管理

1. 保证放置计算机的机房的安全，必要时需加报警系统，同时应提供软件备份方案，把备份好的软件放在安全的地点；
2. 保证所有的通信设施（包括有线通讯线、电话线、局域网、远程网等）都不会被非法人员监听；
3. 钥匙或信用卡识别设备、用户口令和钥匙分配、文件保护、备份或恢复方案等关键文档资料要保存在安全的位置。

普通用户安全管理

1. 系统管理员有责任发现并报告系统的安全问题，当普通用户登录时，其 Shell 在给出提示前先执行</etc/profile>文件，要确保该文件中的 PATH 环境变量指定最后搜索当前工作目录；
2. 系统管理员可定期抽取一个用户，将该用户安全检查结果（用户的登录情况简报、SUID/SGID 文件列表等）发送到其部门及相关人员；
3. 注意提高安全管理意识。系统管理员应强化安全规则，用户必须遵守个人安全标准，在权限允许的范围内进行操作，也可使用一些提高安全性的工具。

超级用户安全管理

1. 在日常使用中最好不要使用 root 账号，以普通用户进入系统可以防止对系统进行破坏性的操作，以 root 身份工作时应保证输入的每个命令的正确性；
2. 超级用户不要运行其他用户的程序，如有需要，就选用 su 命令进入普通用户账号；
3. 经常改变 root 的用户口令；
4. 设置用户口令的时效；
5. 不要把当前工作目录排在 PATH 路径表的前边，以免特洛伊木马的侵入，键入 /bin/su 来执行 su 命令；
6. 不要没注销帐户就离开终端，特别是作为 root 用户时更不能这样；
7. 可以将登录名 root 改成别的名称，使破坏者不能在 root 用户登录名下猜测各种可能的用户口令从而非法进入 root 帐户；
8. 查出不寻常的系统使用情况，如大量地占用磁盘、CPU 时间、进程，大量地使用 su 的企图，大量的无效登录与到某一系统的网络传输，以及可疑的 ucp 请求；
9. 保持系统文件安全的完整性，检查所有系统文件的存取许可，要特别注意设备文件的存取许可，任何具有 SUID 许可的程序都可能是黑客攻击的对象；
10. 将磁盘的备份存放在安全的地方；
11. 查出久未使用的登录帐户，并取消此帐户；
12. 确保没有无用户口令的登录帐户；
13. 启动系统记帐、加密、RSA 等安全机制；
14. 当安装来源不可靠的软件时，要检查源代码和 makefile 文件，查看特殊的子程序调用或命令；
15. 如认为系统已泄密，就设法查出责任人与事故原因，并及时进行补救。

9.1.2 常见安全问题及对策

系统安装时的考虑

在系统安装的分区步骤中，不要只图简单把所有的空间都留给根分区，应该把不同的部分放在不同的分区，如/home、/boot、/var 和/tmp。

关闭不必要的服务

通常情况下，系统默认自带的部分服务可能是用户不需要的，用户可以根据实际需要，关闭不必要的服务，以增强系统的安全性。

可执行如下命令停止指定服务：

```
# systemctl stop <服务名>.service
```

可执行如下命令禁止服务在系统启动时自动启动：

```
# systemctl disable <服务名>.service
```

其中，<服务名>为需要操作的服务名称，如 vsftpd。

帐户口令安全

CGSL 采用了将系统管理员和普通用户分开的策略，这种策略保证了系统的健壮性，同时也使 CGSL 下的病毒难以编写（用户编写的程序仅对自己的目录有写权限，而与操作系统的其它部分是隔离开的）。

脆弱的口令是系统不安全的最主要原因，建议用下面的规则选择有效的口令：

1. 至少要有 6 个字符，最好包含一个以上的数字或特殊字符；
2. 口令不能太简单，所谓的简单就是很容易猜出来，避免用自己的名字、电话号码、生日、职业或者其它个人信息作为口令；
3. 不要把口令写在日历上或计算机旁边等别人能看到的地方；
4. 应该设置口令的有效期，在一段时间之后就要更换口令；
5. 如果发现有人试图猜测您的口令，而且已经试过多次了，就必须重新设定口令。

对服务器进行设置

可以采用以下防范措施：

1. 设置机器的 BIOS 选项，将从硬盘以外的启动设为不允许状态，并在 BIOS 上加设口令，使无权用户无法改变其设定；

2. 为 GRUB 或 Lifo 设置口令，只有输入正确的口令时，才能启动机器；
3. 只允许超级用户有使用<Ctrl + Alt + Del>组合键的权利，或者禁止任何用户使用< Ctrl + Alt + Del >组合键的权利。这样可以有效地阻止用户在本地控制台登录后，使用< Ctrl + Alt + Del >来关闭或重启系统，导致服务的中断。

文件系统安全

系统中的文件权限通过设置十位的权限标志位实现。所以对一些关键系统文件的属性设置要十分小心，以免导致不可挽回的损失。

带有文件的附加权限位 ***SUID*** 与 ***SGID*** 的程序运行时会引起很大的安全漏洞，应该尽量减少使用机会。系统管理员应该经常使用 ***find*** 命令来浏览自己的文件系统以检查新的 ***SUID*** 程序。

此外，可以用 ***chattr*** 命令来改变文件的属性，这里主要注意的是两个属性。

a 只可添加属性；

i 不可改变属性；

对于系统的配置文件，最好设置为不可改变属性，而对于一些日志文件，可设置为只可添加属性。参见下面的示例：

```
#chattr + I /etc/xinetd.conf
#chattr + a /var/log/secure
```

如果要去掉这些属性，将上面命令中的“+ ”号改为“-”号即可。

9.2 系统备份

世界上没有百分之百的安全，为了防止不可顶料的网络攻击、系统硬件故障或用户的非法操作而发生数据丢失，系统管理员必须制定一个备份计划，并定期对系统进行安全备份，以便在系统万一崩溃时，可以及时将系统恢复到最佳状态。

9.2.1 备份前的准备

选择备份介质

有很多介质可以用来进行数据备份，目前比较常用的备份介质为硬盘。用户需要根据自己系统备份计划的实际情况，从可靠性、速度、可用性、易用性和费用成本几个方面考虑来进行备份介质的选择。

进行备份的时机

进行系统备份要定期执行，备份通常应该选择在系统比较空闲时进行，以免影响系统处理正常任务，如可以选择在 0:00 之后进行。

备份策略的选择

1. 完全备份

每隔一定时间对系统进行一次全面备份的方法，是最基本的备份方案。但这样做工作量很大，又需要过多的备份介质，因此不能频繁地进行全面备份，要隔一段较长时间，如一个月，进行一次完整备份。但这样一旦发生数据丢失，就只能恢复到上次备份的数据。

2. 增量备份

先进行一次完全备份，然后每隔一个较短时间进行一次备份，仅备份在这个期间更改的内容。当经过一个较长时间的积累后再进行一次完全备份。这样每次备份的工作量小，能够频繁操作，而且也比较经济。

3. 更新备份

与增量备份方式有些相似。首先每月进行一次完全备份，然后每天进行一次更新数据的备份。不同之处是：增量备份是备份该天更改的数据，而更新备份是备份从上次进行完全备份后更改的全部数据文件。一旦发生数据丢失，可以使用前一个完全备份恢复到前一个月的状态，再使用前一个更新备份恢复到前一天的情况。

增量备份和更新备份都能以较为经济的方式实现，在不同备份策略之间进行选择不但与系统数据更新的方式有关，也依赖于管理员的习惯。

4. 备份工具的选择

有许多工具可用于制作备份。CGSL 中提供了传统的 tar、bzipZ、gzip、cpio 等工具，当然也可以使用其它第三方的软件包。

9.2.2 常用备份命令

有时候，我们需要把一组文件贮存成一个文件以便备份或传输到另一个目录甚至另一台机算机上。我们还需要把一组文件压缩成一个文件，因而它占用少量的磁盘空间并能更快地通过网络上下载。

下面介绍 CGSL 下最常用的归档压缩工具 tar、bzipZ、gzip 和 zip。

9.2.2.1 tar 命令

利用 ***tar*** 可以将文件和目录归档，也可以在档案中改变文件，或者向档案中加入新的文件。*tar* 最初被用来在磁带上创建档案，现在则可以在任何设备上使用。*tar* 命令实现把一大堆文件和目录全部打成一个包的功能，这对于备份或将几个文件组合成一个文件以便网络传输是非常有用的。

tar 命令的语法格式为：

```
#tar <operation> [options]
```

使用时，主选项是必须的，辅助选项可以选用。主选项主要包括：

1. **c** ：创建新的档案文件。如果用户想备份一个目录或是一些文件，就选择此选项。
2. **r** ：把要存档的文件追加到档案文件末。如用户已完成备份文件，又发现还有一部分文件或目录忘记了，就可以使用此选项。
3. **t** ：列出档案文件的内容，查看已经备份了哪些文件。
4. **u** ：更新文件。即用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。
5. **x** ：从档案文件中释放文件。

辅助选项主要有：

6. **b** ：为磁带机而设定，其后跟一数字，用来说明区块的大小。
7. **f** ：使用档案文件或设备，此选项通常为必选项。
8. **k** ：保存已存在的文件，使用户在还原文件中，遇到相同的文件不会进行覆盖。
9. **m** ：在还原文件时，把所有文件的修改时间设定为现在。
10. **M** ：创建多卷的档案文件，以便在几个磁盘中存放。
11. **v** ：详细报告 tar 处理的文件信息。
12. **w** ：每一步都要求确认。
13. **z** ：用 gzip 来压缩/解压缩文件，加上此选项后可以将档案文件进行压缩，还原时一定要有该选项才能进行解压缩。

下面我们通过一些实例来熟悉对备份命令的使用。

例 1 ：把/home 目录包括其子目录全部做成备份文件 home.tar。

```
#tar cvf home.taf /home
```

例 2 ：把/home 目录包括其子目录全部备份并进行压缩，生成文件名为 home.tar.gz。

```
#tar czvf home.tar.gz /home
```

例 3 ：把 home.tar.gz 文件还原并解压缩。

```
#tar xzvf home.tar.gz
```

例 4 ：查看 home.tar 文件的内容，并以分屏方式显示在屏幕上。

```
#tar tvf home.taf lmofe
```

例 5 ：在软盘/dev/fdo 中创建一个备份文件，将/tmp 目录中所有的文件都拷贝进来。

```
#tar cf /dev/fd0 /tmp
```

要恢复设备磁盘中的文件，则可使用 xf 选项。

当需要备份的文件大小超过设备的可用存储空间时，可以创建一个多卷的 tar 文件，使用 M 选项向一个软盘存储过程中，系统在一张软盘已满时会提示放入新的软盘，以实现把 tar 档案存入多张磁盘中。如：

```
#tar cMf /dev/fd0 /home
```

9.2.2.2 bzip2 和 bunzip2

要使用 bzip2 来压缩文件，在 sehll 提示符下键入以下命令：

```
#bzip2 filename
```

该文件就会被压缩，并被保存为 filename.bz2。

要解开被压缩的文件，键入以下命令：

```
#bunzip2 filename.bz2
```

filename.bz2 文件会被删除，而代之以 filename 文件。

可以使用 bzip2 命令同时处理多个文件和目录，方法是将它们逐一列出，并用空格隔开。例如：

```
#bzip2 filename.bz2 filel file2 file3 /usr/local/rfinput
```

上面的命令把 filel、file2 、file3 以及/usr/local/rfinput 目录的内容压缩起来，存放到 filename.bz2 文件中。

9.2.2.3 gzip 和 gunzip 命令

gzip 是一个经常使用的文件压缩和解压缩命令。该命令的语法格式为：

```
#gzip [-acdfhlLnNrtvV19] [-S suffix] [name...]
```

常用的选项参数如下：

-c ：将输出写到标准输出上，并保留原有文件。

-d ：将压缩文件解压缩。

-l ：对每个压缩文件显示其大小、未压缩文件的大小、压缩比和名称等。

-r ：递归式地查找指定目录并压缩其中的所有文件或是解压缩。

-t ：测试、检查压缩文件是否完整。

-v ：对每一个压缩和解压缩文件，显示文件名和压缩比。

-num ：用指定的数字来调整压缩的速度。

现在假设在目录/home 下有文件 aatxt、bbtxt、cctxt，把它们压缩成 gz 文件的命令如下：

```
#gzip /home/*
#ls
aa.txt.gz bb.txt.gz cc.txt.gz
```

要将上例中的文件解压，并列出详细的信息，使用命令

```
#gzip -dv /home/*
```

要解开被压缩的文件，也可以使用以下命令：

```
#gunzip filename.gz
```

filename.gz 会被删除，而代之以 filename。

♣ 提示：要获得这两个命令的详细信息，可以在 Shell 提示下键入 man gzip 和 man gunzip 来阅读它们的帮组信息。

9.2.2.4 zip 和 unzip

要使用 zip 命令压缩文件，在 Shell 提示符下键入下面的命令：

```
#zip -r filename.zip filesdir
```

在上例中，filename.zip 表示要创建的压缩文件，filesdir 表示要压缩的文件目录。-r 选项表示递归地压缩所有包括在 filesdir 目录中的文件。

若要解压缩 filename.zip 文件，键入以下命令：

```
#unzip filename.zip
```

可以使用 zip 命令同时处理多个文件和目录，方法是将它们逐一列出，并用空格隔开：

```
#zip -r filename.zip filel file2 file3 /usr/local/rfinput
```

上面的命令把 filel 、file2 、file3 以及/usr/local/rfinput 目录的内容压缩起来，存放到 filename.zip 文件中。

♣ 提示：要获得这两个命令的详细信息，请参考 man zip 和 man unzip 的说明页。

9.3 加密措施

CGSL 在设计中充分考虑了安全因素，使用了多种有代表性的加密程序来保护系统与用户的安全。

9.3.1 SSH 和 RSA/DSA 认证

SSH (Secure Shell）是一个用来登录远程服务器并在远程服务器上执行命令的程序，在缺少安全防护的网络上，能为两台互相信任的主机间提供安全可靠的加密通信。SSH 缺省是打开的，可以直接使用。

OpenSSH 是 SSH 协议的免费开源实现。使用 OpenSSH 工具能够增强系统的安全性。OpenSSH 加密所有的通信(包括口令)，有效的防止了窃取和网络攻击。除此之外，OpenSSH 还提供了多种安全认证方法。而 telnet、riogin、ftp 等连接工具使用纯文本口令，并被明文发送，这些信息可能会被截取，未经授权的人员可能会使用截取的口令登录系统并造成危害。

OpenSSH 包括：ssh（替代了 rlogin 和 telnet）、scp（替代了 rcp）、sftp（替代了 ftp）和服务器端的 sshd。其他的基本工具包括 ssh-add、ssh-agent、ssh-keygen 等。

9.3.1.1 使用 RSA/DSA 认证

OpenSSH 不仅是安全的而且是加密的。OpenSSH 的一个更加吸引人的特性是其功能组件一一 RSA/DSA 密钥认证系统，它可以代替 OpenSSH 缺省使用的标准安全密码认证系统。

RSA 和 DSA 认证协议基于一对专门生成的密钥（公钥和私钥）的来认证用户。经过适当的配置，能够不必提供密码就同远程机器建立安全的连接。

RSA 和 DSA 认证需要一些初始配置。要设置 RSA 和 DSA 认证，首先需要生成一对密

钥，一把私钥和一把公钥。公钥用于对消息进行加密，只有拥有私钥的人才能对该消息进行解密。公钥只能用于加密，而私钥只能用于解密由匹配的公钥编码的消息。

钥匙必须单独为每个用户生成。要为某用户生成密匙，用将要连接到远程机器的用户身份来执行下面的步骤。如果以 root 身份执行下列步骤，就只有 root 用户才能使用这对密匙。

生成 RSA 密钥对

要生成 RSA 密匙对，先在 Shell 提示符下键入下列命令：

```
#ssh-keygen -t rsa
```

当要求输入存放密钥的位置时，按回车键接受<-/ssh/id_rsa>的默认位置。接下来输入一个与用户账号口令不同的口令，再输入一次以确认。

命令完成后，公钥被写入<-/ssh/id_rsapub>；私钥被写入<-/ssh/id_rsa>。注意，一定不要把私钥出示给任何人。

使用 ***chmod 755 -/ssh*** 命令改变用户主目录下 ssh 目录的许可权限。

把公钥 <-/ssh/id_rsapub> 的内容复制到想要连接的远程机器上的 <-/ssh/authorized_keys> 文件中。如果文件 <-/ssh/authorized_keys> 不存在，可以把 <-/ssh/id_rsapub>文件复制为远程机器的<-/ssh/authorized_keys>文件。

生成 DSA 密钥对

要生成 DSA 密匙对，先在 Shell 提示符下键入下面的命令：

```
#ssh-keygen -t dsa
```

当要求输入存放密钥的位置时，接受<-/ssh/id_dsa>的默认位置。接下来输入一个与用户账号口令不同的口令，再输入一次以确认。

命令完成后，公钥被写入<-/ssh/id_dsapub>；私钥被写入<-/ssh/id_dsa>。注意，一定不要把私钥出示给任何人。

使用 ***chmod 755 -/ssh*** 命令改变用户主目录下的 ssh 目录的许可权限。

把公钥 <-/ssh/id_dsapub> 的内容复制到想要连接的远程机器中的 <-/ssh/authorized_key2> 文件中。如果文件 <-/ssh/authorized_key2> 不存在，可以把 <-/ssh/id_dsapub>文件复制为远程机器上的<-/ssh/authorized_key2>文件。

配置 ssh-agent

ssh-agent 是一个用于保存私钥的授权代理。只要使用 ssh-add 命令把私钥添加到

ssh-agent 的高速缓存中，ssh 将从 ssh-agent 获取您的私钥，而不会提示要密码了。

在 Shell 提示符下，键入下面的命令：

```
#exec /usr/bin/ssh-agent $SHELL
```

然后，键入下面的命令：

```
#ssh-add
```

接着，输入我们的密钥口令。如果配置了不止一个密钥对，会被提示输入每个口令。

当用户注销后，口令就会被忘记。必须在每次登录到虚拟控制台或打开终端窗口时都执行这两条命令。

提示：默认情况下，系统禁止 root 用户通过 ssh 远程登陆，只能以普通用户身份登陆。

另外，我们可以到 OpenSSH 的官方站点 **http://www.openssh.com** 获得更多详细的信息。

9.3.2 PGP

PGP — Pretty Good Privacy，是一个基于 RSA 公匙加密体系的邮件加密软件。可以用它对您的邮件加密以防止非授权者阅读，它还能在邮件中加上数字签名从而使收信人可以确信邮件的来源。它让用户可以安全地和从未见过的人们通讯，事先并不需要任何保密的渠道用来传递密匙。它采用了审慎的密匙管理，这一种 RSA 和传统加密的杂合算法，用于数字签名的邮件文摘算法，加密前压缩等。

PGP 的创始人是美国的 Phil Zimmermann。它的创造性在于把 RSA 公匙体系的方便和传统加密体系的高速度结合起来，并且在数字签名和密匙认证管理机制上有巧妙的设计。

如果需要了解 PGP 的详细知识，请访问站点 **http://www.pgpi.org**。

9.3.3 OPENSSL

Openssl 是一个协议独立的加密方案，在网络信息包的应用层和传输层之间提供了安全的通道。

一些服务器软件，例如 IMAP、POP、Samba、FTP、Apache 等等，在提供服务时需要对用户进行认证，只有认证通过后服务才会被许可。然而对于 server/client 方式的服务，客户端和服务端之间通讯都是以明文方式进行的，Openssl 正是提供了对传输的数据的一种加密方式。

Openssl 可以安装在 CGSL 服务器上，它需要一些第三方提供的应用程序来为服务提供加密。简单说来，就是 HTML 或 CGI 经过幕后的服务器进行了加密处理，然而对 HTML

和 CGI 的作者来说是透明的。

openssl 软件包提供了 SSL（Secure Sockets Layer）及 TLS（Tansport Layer Security）协议的加密保护，而且提供了 apache 方式的许可证，从而强化了 HTTP 服务器的安全性。

我们可以到 Openssl 的官方站点 **http://www.openssl.org** 获得更多详细的信息。

9.4 账户安全

9.4.1 账户管理

见第 3 章 用户和组群管理

9.4.2 用户认证(PAM)

用户认证是操作系统对登录到系统中的用户的身份进行确认和授权的机制，是整个系统安全的重要关卡。PAM 框架是实现用户认证的基础，通过将应用程序与具体的认证机制分离，使得系统改变认证机制时，不再需要修改采用认证机制的应用程序，仅需由管理员配置应用程序的认证服务模块，极大的提高了认证机制的通用性和灵活性。

相关配置待续。

9.4.3 访问控制

9.4.3.1 文件权限控制

请参见 1.4 节内容。

9.4.3.2 用户密码强度配置

对用户密码强度的限制是在/etc/pam.d/system-auth 文件中设置的。在这个文件中缺省有下面的一行内容：

```
password requisite pam_pwquality.so try_first_pass local_users_only
```

可以在这一行后面附加和密码强度有关的配置选项：

try_first_pass：在提示用户输入密码之前，模块首先尝试先前的密码，以测试是否满足该模块的需求。

local_users_only：这个模块不会检查位于/etc/passwd 之外的用户，但是因为 password

stack 中随后的模块可以使用 use_authtok 选项，这时仍然会要求输入密码。

其配置文件位于/etc/security/pwquality.conf

difok：新密码中不同于旧密码的字符数，默认是 1。如果是 0 表示不作检查除非新密码和旧密码一样。

minlen：新密码的最小长度，默认是 8 位，不能低于 6 位。

dcredit：新密码中数字的最大个数，如果小于 0 就是最少个数。

ucredit：新密码中大写字母的最大个数，如果小于 0 就是最少个数。

lcredit：新密码中小写字母的最大个数，如果小于 0 就是最少个数。

ocredit：新密码中其它字符的最大个数，如果小于 0 就是最少个数。

minclass：新密码中数字，大写字母，小写字母，其它字符种类的最少个数。

maxrepeat：新密码中允许的最大连续字符数，如果值为 0 即为不启用，默认是 0。

maxsequence：新密码中单调字符序列的最大长度，比如 12345 这种序列。如果值为 0 即为不启用，默认是 0。

maxclassrepeat：新密码中同一类中允许的连续字符的最大数，如果值为 0 即为不启用，默认是 0。

gecoscheck：如果非 0，检查用户 passwd 条目的 GECOS 字段中长度超过 3 个字符的单词是否包含在新密码中，如果值为 0 即为不启用，默认是 0。

dictcheck：如果非 0，则对密码进行字典检查，当前字典检查是使用 cracklib 库做的，默认为 1。

usercheck=N：如果非 0，检查密码是否包含用户名，当用户名短于 3 个字符时不执行检查，默认为 1。

enforcing=N：如果非 0，如果密码未通过检查则拒绝密码，否则只打印警告信息。这项设置仅用于 pam_pwquality 模块，可能是其他应用程序基于它显式地改变了它们的行为，它不会影响 pwmake 和 pwscore，默认为 1。

badwords：密码中不能包含的空格隔开的单词列表。这些单词对于 cracklib 字典检查是额外的。该设置也能被应用用来对未创建的用户进行模拟的 gecos 检查。

dictpath：cracklib 字典的路径，默认就是使用 cracklib。

9.5 防火墙(Netfilter/Iptables/)

9.5.1 防火墙(Netfilter/Iptables)介绍

netfilter/iptables(简称 iptables)组成 Linux 平台下的包过滤防火墙,可以完成封包过滤,封包重定向和网络地址转换（NAT）等功能。用户可以根据自己特定的需求来配置防火墙,在防火墙解决方案上节省费用和对 IP 信息包过滤具有完全控制权。netfilter/iptables IP 信息包过滤系统可用来添加、编辑和除去规则，这些规则是在做信息包过滤决定时，防火墙所遵循和组成的规则。这些规则存储在专用的信息包过滤表中，而这些表集成在 CGSL 内核中。 在信息包过滤表中，规则被分组放在我们所谓的链（chain）中。下文将详细讨论这些规则以及如何建立这些规则，并将它们分组在链中。

9.5.2 建立规则和链

通过向防火墙提供有关对来自某个源、到某个目的地或具有特定协议类型的信息包要做些什么的指令，以设置相关规则来控制信息包的过滤。 通过使用 netfilter/iptables 系统提供的特殊命令 iptables，建立这些规则，并将其添加到内核空间的特定信息包过滤表内的链中。关于添加 / 除去 / 编辑规则的命令的一般语法如下：

```
$ iptables [-t table] command [match] [target]
```

表（table）

[-t table] 表是包含仅处理特定类型信息包的规则和链的信息包过滤表。有三种可用的表选项：filter、nat 和 mangle。该选项不是必需的，如果未指定，则 filter 用作缺省表。

filter 表用于一般的信息包过滤，它包含 INPUT、OUTPUT 和 FORWARD 链。nat 表用于要转发的信息包，它包含 PREROUTING、OUTPUT 和 POSTROUTING 链。如果信息包及其头内进行了任何更改，则使用 mangle 表。该表包含一些规则来标记用于高级路由的信息包，该表包含 PREROUTING 和 OUTPUT 链。

注：PREROUTING 链由指定信息包一到达防火墙就改变它们的规则所组成，而 POSTROUTING 链由指定正当信息包打算离开防火墙时改变它们的规则所组成。

命令（command）

上面这条命令中具有强制性的 command 部分是 iptables 命令的最重要部分。它告诉 iptables 命令要做什么，例如，插入规则、将规则添加到链的末尾或删除规则。以下是最常用的一些命令：

- -A 或--append：该命令将一条规则附加到链的末尾。

示例：

```
$ iptables -A INPUT -s 205.168.0.1 -j ACCEPT
```

该示例命令将一条规则附加到 INPUT 链的末尾，确定来自源地址 205.168.0.1 的信息包可以 ACCEPT。

- -D 或--delete：通过用-D 指定要匹配的规则或者指定规则在链中的位置编号，该命令从链中删除该规则。下面的示例显示了这两种方法。

示例：

```
$ iptables -D INPUT --dport 80 -j DROP
$ iptables -D OUTPUT 3
```

第一条命令从 INPUT 链删除规则，它指定 DROP 前往端口 80 的信息包。第二条命令只是从 OUTPUT 链删除编号为 3 的规则。

- -P 或--policy：该命令设置链的缺省目标，即策略。所有与链中任何规则都不匹配的信息包都将被强制使用此链的策略。

示例：

```
$ iptables -P INPUT DROP
```

该命令将 INPUT 链的缺省目标指定为 DROP。这意味着，将丢弃所有与 INPUT 链中任何规则都不匹配的信息包。

- -N 或--new-chain：用命令中所指定的名称创建一个新链。

示例：

```
$ iptables -N allowed-chain
```

- **-F 或--flush：**如果指定链名，该命令删除链中的所有规则，如果未指定链名，该命令删除所有链中的所有规则。此参数用于快速清除。

示例：

```
$ iptables -F FORWARD
$ iptables -F
```

- -L 或--list：列出指定链中的所有规则。

示例：

```
$ iptables -L allowed-chain
```

匹配（match）

iptables 命令的 match 部分(可选项)指定信息包与规则匹配所应具有的特征(如源和目的地地址、协议等)。匹配分为两大类：通用匹配和特定于协议的匹配。下面是一些重要且常用的通用匹配及其示例和说明：

- **-p 或--protocol：**该通用协议匹配用于检查某些特定协议。协议示例有 TCP、UDP、ICMP、用逗号分隔的任何这三种协议的组合列表以及 ALL（用于所有协议）。ALL 是缺省匹配。可以使用!符号，它表示不与该项匹配。示例：

```
$ iptables -A INPUT -p TCP, UDP
$ iptables -A INPUT -p ! ICMP
```

在上述示例中，这两条命令都执行同一任务—它们指定所有 TCP 和 UDP 信息包都将与该规则匹配。通过指定! ICMP，我们打算允许所有其它协议（在这种情况下是 TCP 和 UDP），而将 ICMP 排除在外。

- **-s 或--source：**该源匹配用于根据信息包的源 IP 地址来与它们匹配。该匹配还允许对某一范围内的 IP 地址进行匹配，可以使用!符号，表示不与该项匹配。缺省源匹配与所有 IP 地址匹配。示例：

```
$ iptables -A OUTPUT -s 192.168.1.1
$ iptables -A OUTPUT -s 192.168.0.0/24
$ iptables -A OUTPUT -s ! 203.16.1.89
```

第二条命令指定该规则与所有来自 192.168.0.0 到 192.168.0.24 的 IP 地址范围的信息包匹配。第三条命令指定该规则将与除来自源地址 203.16.1.89 外的任何信息包匹配。

- **-d 或--destination：**该目的地匹配用于根据信息包的目的地 IP 地址来与它们匹配。该匹配还允许对某一范围内 IP 地址进行匹配，可以使用!符号，表示不与该项匹配。示例：

```
$ iptables -A INPUT -d 192.168.1.1
$ iptables -A INPUT -d 192.168.0.0/24
$ iptables -A OUTPUT -d ! 203.16.1.89
```

目标（target）

目标是由规则指定的操作，对与那些规则匹配的信息包执行这些操作。除了允许用户定义的目标之外，还有许多可用的目标选项。下面是常用的一些目标及其示例和说明：

- **ACCEPT：**当信息包与具有 ACCEPT 目标的规则完全匹配时，会被接受（允许它前往目的地），并且它将停止遍历链（虽然该信息包可能遍历另一个表中的其它链，并且有可能在那里被丢弃）。该目标被指定为-j ACCEPT。

- **DROP**：当信息包与具有 DROP 目标的规则完全匹配时，会阻塞该信息包，并且不对它做进一步处理。该目标被指定为-j DROP。
- **REJECT**：该目标的工作方式与 DROP 目标相同，但它比 DROP 好。和 DROP 不同，REJECT 不会在服务器和客户机上留下死套接字。另外，REJECT 将错误消息发回给信息包的发送方。该目标被指定为-j REJECT。示例：

```
$ iptables -A FORWARD -p TCP --dport 22 -j REJECT
```

- **RETURN**：在规则中设置的 RETURN 目标让与该规则匹配的信息包停止遍历包含该规则的链。如果链是如 INPUT 之类的主链，则使用该链的缺省策略处理信息包。它被指定为-jump RETURN。示例：

```
$ iptables -A FORWARD -d 203.16.1.89 -jump RETURN
```

还有许多用于建立高级规则的其它目标，如 LOG、REDIRECT、MARK、MIRROR 和 MASQUERADE 等。

保存规则

用上述方法所建立的规则会被保存到内核中，当重新引导系统时，会丢失这些规则。所以，将没有错误的且有效的规则集添加到信息包过滤表，同时希望在重新引导之后再次使用这些规则，那么必须将该规则集保存在文件中。可以使用 iptables-save 命令来做到这一点：

```
$ iptables-save > iptables-script
```

现在，信息包过滤表中的所有规则都被保存在文件 iptables-script 中。无论何时再次引导系统，都可以使用 iptables-restore 命令将规则集从该脚本文件恢复到信息包过滤表，如下所示：

```
$ iptables-restore iptables-script
```

如果您愿意在每次引导系统时自动恢复该规则集，则可以将上面指定的这条命令放到任何一个初始化 shell 脚本中。

9.5.3 启动与关闭防火墙

1. 检查 iptables 的状态

```
# systemctl status iptables
```

2. 检查防火墙规则

```
#iptables -L -n
```

3. 检查防火墙 nat 规则

```
#iptables -t nat -L -n
```

4. 关闭防火墙

```
# systemctl stop iptables
```

5. 开启防火墙

```
# systemctl start iptables
```

9.6 防火墙(Netfilter/Firewalld)

9.6.1 防火墙(Netfilter/Firewalld)介绍

在 CGSL V6 里一个相互作用的 netfilter 被引入：firewalld。firewalld 是一个系统守护进程，可以配置和监控系统的防火墙规则。应用程序可以使用 DBus 消息系统请求 firewalld 打开端口，它可以禁用或锁定开放的端口。firewalld 涵盖了 IPv4、IPv6 和 ebtables 的设置。firewalld 守护进程来源于 firewalld 包。这个包在 Basice 安装方式会被安装，而在 minimal 安装方式不会被安装。系统默认使用 firewalld 为防火墙，而不是 iptables。

firewalld 是所有网络流量划分为区(zones)，简化防火墙管理。例如一个包传入的网络接口时，会根据源 IP 地址把流量转移到用于相应的区域(zone)的防火墙规则。每个区域(zone)都预设开放的或关闭的端口和服务列表。

9.6.2 区域(zones)概念

firewalld 附带了一些预定义的区域，以适合各种用途。默认区域设置为 public，网络接口将被分配到 public 区，但 lo 接口被分配到 trusted 区。下面的表详细介绍这些区域的结构，但系统管理员可能需要定制这些区域具有不同的设置。默认情况下，由系统发起的通信所有区域都允许任何入站流量与部分出站流量。

区域名字	默认设置
trusted	允许所有的流量传入。
home	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh**、**mdns**、**ipp-client**、**samba-client** 与 **dhcpv6-client** 服务。
internal	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh**、**mdns**、**ipp-client**、**samba-client** 与 **dhcpv6-client** 服务(与 **home**

	区相同)。
work	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh**、**ipp-client** 与 **dhcpv6-client** 服务。
public	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh** 与 **dhcpv6-client** 服务。新加的网络接口默认会被分配到此区。
external	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh** 服务。IPv4 传出流量通过该区域转发会被伪装成从 IPv4 网络接口上传出。
dmz	默认拒绝传入流量，除非是涉及到出站流量或预设的 **ssh** 服务。
block	默认拒绝所有的传入流量，除非是涉及到出站流量。
drop	默认丢弃所有的传入流量，除非是涉及到出站流量(甚至不使用 *ICMP 错误*回应)。

注：想知道所有的预设区域的列表和使用方法，请查阅 firewalld.zones（5）手册。

firewalld 还附带了一些预设的服务。这些服务可以用来容易地使特定网络服务流量穿过防火墙。下面的表格详细介绍防火墙中的区域的默认配置使用的预设服务的设置。

注：其他预设的服务可使用”firewall-cmd –get-services”命令查看，或者查看

服务名	默认设置
ssh	本地 ssh 服务。流量通过 22/tcp 端口。
dhcpv6-client	本地 DHCPv6 客户端。在 fe80::/64 IPv6 网络上流量通过 546/udp 端口。
ipp-client	本地 IPP 打印。流量通过 631/udp 端口。
samba-client	本地 windows 文件与打印共享客户端。流量通过 137/udp 端口与 138/udp 端口。
mdns	组播 DNS 本地连接名称解析。在组播地址 224.0.0.251(IPv4)或者 ff02::fb(IPv6)上流量通过 5353/udp 端口。

/usr/lib/firewalld/services 目录下的文件。

引导系统时自动恢复该规则集，则可以将上面指定的这条命令放到任何一个初始化 shell 脚本中。

9.6.3 防火墙(Netfilter/ firewalld)配置

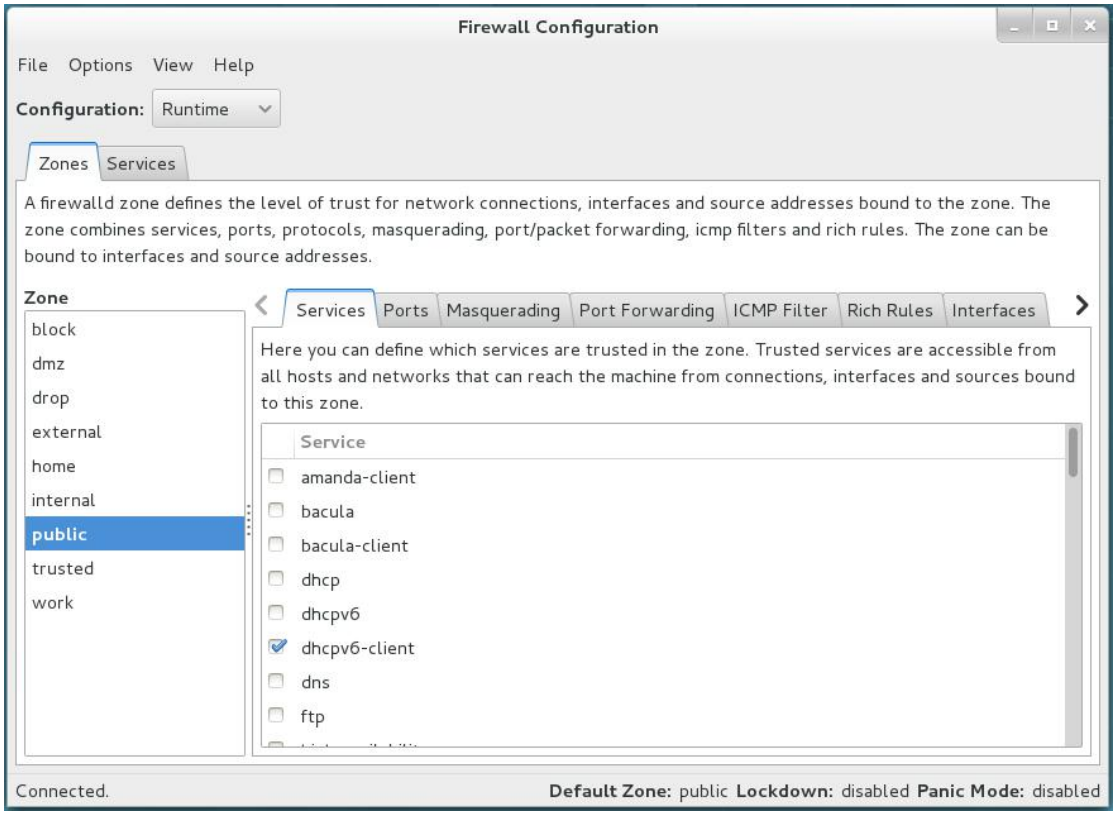
修改防火墙配置有三种主要方式：

- 通过修改/etc/firewalld 目录下的配置文件。
- 通过使用图形化工具 firewall-config；
- 通过使用命令 firewall-cmd；

注：firewalld 需要依赖 NetworkManager 区分网络接口属于哪个区域，使用 firewalld 配置防火墙前需要确定 NetworkManager 是正在运行。

配置防火墙(firewall-config)

图形化工具 firewall-config，可以检查或者修改防火墙正在运行的、和持续性的配置。通过安装 firewall-config 的 RPM 包里，firewall-config 工具可以从命令行上打开，或从应用程序菜单下 Applications > Sundry > Firewall 打开。如果是普通用户打开 firewall-config 工具需要 root 密码。



防火墙配置(firewall-config)主界面

在防火墙(firewall-config)配置主界面上，系统管理员可以选择修改当前的或是持续性(永久性)的配置。在大多数情况下，系统管理员将需要调整持久性（永久性）的配置，然后使用 Options > Reload Firewalld 菜单项激活防火墙的改变。

如果要修改网络接口和源 IP 地址/范围所属区域，选择左侧的菜单区的 Zones 选项卡。在右侧的 Interfaces 和 Sources 标签下分别填写网络接口和源 IP 地址/范围。

端口开放需要在 Services 标签下勾选，或通过在该区域的 Ports 标签下添加一个自定义的端口。

如果一个特定的端口组在多个区域中被开放时，系统管理员也可以定义为那些端口的服务。这可以在左侧的菜单区的 Services 选项卡来完成(配置模式需要选择 Permanent/永久配置)。

注：在 Permanent(永久性配置)所做的任何更改需要重新启动或重新加载 firewalld 服务才生效，在 Runtime(运行时配置)所做的任何更改不会保留在 firewalld 服务重新启动或重新加载后。

配置防火墙(firewall-cmd)

对于那些偏好工作在命令行上的管理员或没有图形环境的情况，可以使用 firewalld 的命令行接口 firewall-cmd。firewall-cmd 接口为主体 firewalld 包的一部分，firewall-config 上执行的操作也可以通过 firewall-cmd 实现。

下面的表列出了一些常用的 firewall-cmd 选项以及描述。注意，除非指定了--permanent 选项，几乎所有的命令都将是临时的配置。命令需要采取--zone=<ZONE>选项，以确定它们影响哪些区域。

firewall-cmd 选项	**描述**
--get-default-zone	查询现在的默认区域。
--set-default-zone=<ZONE>	设置默认区域。修改默认区域会影响临时的配置(runtime)和永久的配置(permanent)。
--get-zones	列出所有可用的区域。
--get-active-zones	列出当前所有正在使用的区域（具有依赖于它们的接口或源 IP 地址/范围），以及它们的接口或源 IP 地址/范围。
--add-source=<CIDR> [--zone=<ZONE>]	添加流量路由，所有从 IP 地址或网络/子网掩码<CIDR>的流量转到指定区域。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--remove-source=<CIDR> [--zone=<ZONE>]	删除流量路由，所有从 IP 地址或网络/子网掩码<CIDR>的流量转到指定区域的规则将被删除。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--add-interface=<INTERFACE> [--zone=<ZONE>]	添加流量路由，所有从接口<INTERFACE>的流量转到指定区域。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--change-interface=<INTERFACE> [--zone=<ZONE>]	将接口<INTERFACE>与区域<ZONE>关联，旧的关联将被移除。所有从接口<INTERFACE>的流量转到指定区域<ZONE>。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。

--list-all [--zone=<ZONE>]	列出区域<ZONE>内所有已关联的接口，源，服务和端口。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--list-all-zones	检索并列出所有的所有区域的信息。
--add-service-<SERVICE> [--zone=<ZONE>]	允许服务<SERVICE>通信。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--remove-service-<SERVICE> [--zone=<ZONE>]	在区域<ZONE>允许列表中移除服务<SERVICE>。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--add-port-<PORT/PROTOCOL> [--zone=<ZONE>]	开放端口<PORT/PROTOCOL>通信。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--remove-port-<PORT/PROTOCOL> [--zone=<ZONE>]	在区域<ZONE>允许列表中端口<PORT/PROTOCOL>。如果没有使用--zone=<ZONE>选项指定区域，将使用默认区域。
--reload	重新加载 firewall。临时的配置将被移除，加载永久的配置。

firewall-cmd 示例

下面的示例默认区域设置为 dmz，来自 192.168.0.0/24 网络的所有流量被分配到 internal 区域，开放 internal 区域的 MySQL 网络端口。

```
#firewall-cmd  --set-default-zone=dmz
#firewall-cmd  --permanent  --zone=internal  --add-source=192.168.0.0/24
#firewall-cmd  --permanent  --zone=internal  --add-service=mysql
#firewall-cmd  --reload
```

9.6.4 启动与关闭防火墙

6. 检查 firewalld 的状态

```
# systemctl status firewalld
```

7. 关闭防火墙

```
# systemctl stop firewalld
```

8. 开启防火墙

```
# systemctl start firewalld
```

9.7 防火墙(firewalld/nftables)

9.7.1 防火墙(firewalld/nftables)介绍

V6 中，iptables 已被 nftables 取代。firewalld/nftables(简称 nftables)组成 Linux 平台下的包过滤防火墙，可以完成封包过滤，封包重定向和网络地址转换（NAT）等功能。用户可以根据自己特定的需求来配置防火墙，在防火墙解决方案上节省费用和对 IP 信息包过滤具有完全控制权。firewalld/nftables IP 信息包过滤系统可用来添加、编辑和除去规则，这些规则是在做信息包过滤决定时，防火墙所遵循和组成的规则。这些规则存储在专用的信息包过滤表中，而这些表集成在 CGSL 内核中。 在信息包过滤表中，规则被分组放在我们所谓的链（chain）中。下文将详细讨论这些规则以及如何建立这些规则，并将它们分组在链中。注意：为了让服务之间不互相干扰，firewalld 与 nftables 只开其一。

9.7.2 编写执行 nftables 脚本

为了解决 iptables 到 nftables 的平滑过渡问题，系统提供了 iptables-translate 和 ip6tables-translate 命令将 iptables 规则转换成新的规则。但是一些扩展式仍缺乏支持，遇到这种情况，转换后的规则前会添加#符号，

例如：

```
iptables-translate -A INPUT -j CHECKSUM --checksum-fill
nft # -A INPUT -j CHECKSUM --checksum-fill
```

另外，用户还可以使用 iptables-restore-translate 和 ip6tables-restore-translate 工具转换一堆规则，可以先用 iptables-save 和 ip6tables-save 命令将现有的 iptables 规则导出。

例如：

```
iptables-save >/tmp/iptables.dump
iptables-restore-translate -f /tmp/iptables.dump
Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
add table ip nat
...
```

nftables 框架提供了一个本地脚本环境，这相对于使用 shell 脚本维护防火墙规则有一个主要的好处：脚本的执行是原子的。这意味着系统执行脚本的时候，一旦遇到错误就会停止执行。这保证防火墙一直处于连贯的状态。

另外 nftables 脚本环境还让管理员能够：

（1）添加注释

（2）定义变量

（3）包含其他的规则文件

当安装好 nftables 包之后，系统会在/etc/nftables/目录下生成*.nft 脚本，这些脚本包含了为不同目的创建表和空链的命令。也可以在这这些文件的基础上扩展成自己的脚本。

类似于其他的脚本，nftables 脚本也需要在第一行设置解释指令。

脚本总是要以如下开头：#!/usr/sbin/nft -f

可以写一个脚本，就像下面这样使用 nft list ruleset 命令同样的格式。

```
#!/usr/sbin/nft -f
# Flush the rule set
flush ruleset
table inet example_table {
chain example_chain {
# Chain for incoming packets that drops all packets that
# are not explicitly allowed by any rule in this chain
type filter hook input priority 0; policy drop;
# Accept connections to port 22 (ssh)
tcp dport ssh accept
}
```

```
}
```

可以使用 nft 的语法来写。

```
#!/usr/sbin/nft -f
# Flush the rule set
flush ruleset
# Create a table
add table inet example_table
# Create a chain for incoming packets that drops all packets
# that are not explicitly allowed by any rule in this chain
add chain inet example_table example_chain { type filter hook input priority 0 ;
policy drop ; }
# Add a rule that accepts connections to port 22 (ssh)
add rule inet example_table example_chain tcp dport ssh accept
```

要运行 nftables 脚本，首先它必须是可执行的，除非它被包含在另一个文件中。

流程：假设脚本是/etc/nftables/example_firewall.nft

（1）可选的，修改文件的属主

```
# chown root /etc/nftables/example_firewall.nft
```

（2）添加可执行权限

```
# chmod u+x /etc/nftables/example_firewall.nft
```

（3）运行脚本

```
# /etc/nftables/example_firewall.nft
```

没有输出即表示执行成功。

添加注释需要以#开头，写在命令后也是可以的。

例如：

```
# Flush the rule set
flush ruleset
add table inet example_table # Create a table
```

定义变量可以使用 define 关键字，对于更复杂的场景，可以使用集合或判定映射。

定义单个变量：

```
define INET_DEV = enp1s0
```

在变量前加$可以使用该变量

```
add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept
```

定义一个匿名集合：

```
define DNS_SERVERS = { 192.0.2.1, 192.0.2.2 }
```

nftables 脚本环境可以通过 include 包含其他的脚本文件，如果仅指定文件名而不指定路径，默认路径位于/etc/目录

包含单个文件：

```
include "example.nft"
```

包含某个目录下所有的 nftables 脚本文件：

```
include "/etc/nftables/rulesets/*.nft"
```

nftables 系统服务会自动加载被/etc/sysconfig/nftables.conf 包含的 nftables 脚本文件，如果希望随机启动使用，放这里就好了。

让 nftables 服务随机启动：

```
# systemctl enable nftables
```

启动 nftables 服务：

```
# systemctl start nftables
```

显示 nftables 的规则集：

流程：为了显示所有规则，请输入以下内容

```
# nft list ruleset
table inet example_table {
chain example_chain {
type filter hook input priority 0; policy accept;
Red Hat Enterprise Linux 8 Securing networks
70
tcp dport http accept
tcp dport ssh accept
}
}
```

1.创建 nftables 表

nftables 中的表是一个命名空间，是链，规则，集合和其他对象的集合。每个表必须定义一个地址族。表的地址族定义了表处理的地址类型。当创建一张表时，可以使用以下的地址族：

ip	仅匹配 ipv4 的包，如果不指定默认使用该地址族
ip6	进匹配 ipv6 的包
inet	同时匹配 ipv4 和 ipv6 的包
arp	匹配 IPv4 地址解析协议(ARP)包。
bridge	匹配通过桥接设备的数据包。
netdev	匹配来自入口的数据包。

使用 nft add table 命令创建新的表：

```
# nft add table inet example_table
```

2.创建 nftables 链

链是装载规则的容器，存在以下两种链类型：

基链：您可以使用基链作为来自网络堆栈的数据包的入口点。

常规链：您可以使用常规链作为跳转目标并更好地组织规则。

以下流程描述了如何向一个表里添加基链:

（1）使用 nft add chain 命令创建一个新的链。例如在 example_table 中创建一个 example_chain 链

```
# nft add chain inet example_table example_chain { type filter hook input
priority 0 \; policy accept \; }
```

这个链过滤输入的包，priority 参数指定 nftables 处理具有相同钩子值的链的顺序。低优先级值优先于高优先级值。策略参数设置此链中的规则的默认操作。请注意，如果您远程登录到服务器，并将默认策略设置为 drop，那么如果没有其他规则允许远程访问，则立即断开连接。

（2）可选的，显示所有的链

```
# nft list chains
table inet example_table {
```

```
chain example_chain {
type filter hook input priority 0; policy accept;
}
}
```

向 nftables 链添加规则

nftables add rule 命令可以将规则添加到链的末端。

例如在 example_table 表中的 example_chain 链中添加一个规则：

```
# nft add rule inet example_table example_chain tcp dport 22 accept
```

向 nftables 链插入规则

nftables insert rule 命令可以将规则添加到链的首部。

例如在 example_table 表中的 example_chain 链中插入一个规则：

```
# nft insert rule inet example_table example_chain tcp dport 22 accept
```

3.使用 nftables 配置 NAT

可以配置如下的 NAT 类型：Masquerading（伪装），SNAT（源地址转换），DNAT（目标地址转换）。

使用伪装或者源地址转换都可以改变包的源 IP 地址，比如 Internet 不会路由转发保留 ip，像 10.0.0.0/8 这种，如果局域网内的机器希望被公网所访问，需要将内网中包的源地址映射到公网 IP。地址伪装与源地址转换很类似，但是还是有区别的：

地址伪装自动使用出口接口的 IP 地址，如果的出口接口使用动态 IP 则使用地址伪装。

SNAT 将包的源 IP 地址转换成特定的 IP 地址，因此 SNAT 会更快，如果的出口接口使用静态 IP 则使用 SNAT。

DNAT 就是将进来的包转发到不同的主机。比如 web 服务器使用局域网保留地址，未直接连接 Internet，就可以在路由器上设置 DNAT 将对应的包转发给这台 web 服务器。

使用 nftables 配置地址伪装

以下流程描述了如何替换包的源 IP 地址，当它通过 ens3 接口的 IP 地址离开主机：

（1）创建一个表

```
# nft add table nat
```

（2）往表中添加 prerouting 和 postrouting 链

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
```

```
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

（3）往 postrouting 链中添加规则以匹配从 ens3 接口出去的包

```
# nft add rule nat postrouting oifname "ens3" masquerade
```

4.使用 nftables 配置 SNAT

以下的流程描述了如何替换包的源 IP 地址，当它通过 ens3 接口离开路由器前往 192.0.2.1：

（1）创建表

```
# nft add table nat
```

（2）往表中添加 prerouting 和 postrouting 链

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

（3）向 postrouting 链添加一条规则，用 192.0.2.1 替换通过 ens3 发出的数据包的源 IP

```
# nft add rule nat postrouting oifname "ens3" snat to 192.0.2.1
```

5.使用 nftables 配置 DNAT

下面的流程描述如何将发送到路由器 80 和 443 端口的传入流量重定向到具有 192.0.2.1 IP 地址的主机：

（1）创建表

```
# nft add table nat
```

（2）往表中添加 prerouting 和 postrouting 链

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

（3）向 prerouting 链添加一条规则，该规则将 ens3 接口上发送到 80 端口的传入流量重定向到 IP 是 192.0.2.1 的主机上。

```
# nft add rule nat prerouting iifname ens3 tcp dport { 80, 443 } dnat to 192.0.2.1
```

（4）根据环境，决定是使用地址伪装还是 SNAT

如果 ens3 接口使用动态 IP，则使用地址伪装：

```
# nft add rule nat postrouting oifname "ens3" masquerade
```

如果 ens3 接口使用静态 IP，则使用 SNAT：

```
#nft add rule nat postrouting oifname "ens3" snat to 198.51.100.1
```

在 nftables 命令中使用集合

nftables 框架原生就支持集合。如果一个规则应该匹配多个 IP 地址、端口号、接口或任何其他匹配条件，就可以使用集合。

在 nftables 中使用匿名集合

匿名集合包含在花括号中的以逗号分隔的值，比如{ 22, 80, 443 },可以直接在规则中使用。但是，匿名集的缺点是，如果要更改集，必须替换规则。

举例如下：

（1）例如，要在 example_table 表中向 example_chain 链添加一个规则，该规则允许传入流量到 22、80 和 443 端口:

```
# nft add rule inet example_table example_chain tcp dport { 22, 80, 443 } accept
```

（2）显示 example_table 表中的所有链中的规则

```
# nft list table inet example_table
table inet example_table {
chain example_chain {
type filter hook input priority 0; policy accept;
tcp dport { ssh, http, https } accept
}
}
```

6. 在 nftables 中使用命名集合

nftables 框架支持可变的命名集合，命名集合是一个列表或元素范围，您可以在一个表中的多个规则中使用它。与匿名集相比，另一个好处是您可以更新一个命名集，而不需要替换使用该集的规则。当创建一个命名集合时，必须指定集合包含的元素的类型，可以设置以下类型：

ipv4_addr	包含 ipv4 地址或范围的集合，比如 192.0.2.1 或 192.0.2.0/24.
ipv6_addr	包含 ipv6 地址或范围的集合，比如 2001:db8::1 或 2001:db8::1/24
ether_addr	包含 MAC 地址的集合，比如 52:54:00:6b:66:42
inet_proto	包含 internet 协议类型的集合，比如 tcp
inet_service	包含 internet 服务的集合，比如 ssh

mark	包含包标记列表的集合，包标记可以是任何正的 32 位整数值

以下为流程举例：

（1）创建一个空集合。以下的例子创建一个 ipv4 地址的集合。

创建一个能储存多个独立 ipv4 地址的集合

```
# nft add set inet example_table example_set { type ipv4_addr \; }
```

创建一个能储存 ipv4 地址范围的集合

```
# nft add set inet example_table example_set { type ipv4_addr \; flags interval
\; }
```

（2）使用集合创建规则，下面的命令向 example_table 中的 example_chain 添加一条规则，该规则将删除 example_set 中来自 IPv4 地址的所有数据包。

```
# nft add rule inet example_table example_chain ip saddr @example_set drop
```

因为当前 example_set 是空的，所以还未起作用。

（3）往 example_set 里添加规则

可以创建集合储存独立的 ipv4 地址

```
# nft add element inet example_table example_set { 192.0.2.1, 192.0.2.2 }
```

也可以创建集合储存 ipv4 地址范围

```
# nft add element inet example_table example_set { 192.0.2.0-192.0.2.255 }
```

当您指定一个 IP 地址范围时，您可以选择使用无类域间路由(CIDR)表示法，例如上面示例中的 192.0.2.0/24。

7.在 nftables 中使用文字映射

文字映射是在规则中直接使用的{match_criteria: action}语句，语句可以包含多个逗号分隔的映射。文字映射的缺点是，如果要更改映射，必须替换规则。

该示例描述了如何使用文本映射将 IPv4 和 IPv6 协议的 TCP 和 UDP 数据包路由到不同的链，分别计算传入的 TCP 和 UDP 数据包。

流程如下：

（1）创建 example_table

```
# nft add table inet example_table
```

（2）在 example_table 中创建 tcp_packets 链

```
# nft add chain inet example_table tcp_packets
```

（3）在 tcp_packets 链中创建规则

```
# nft add rule inet example_table tcp_packets counter
```

（4）在 example_table 中创建 udp_packets 链

```
# nft add chain inet example_table udp_packets
```

（5）在 udp_packets 链中创建规则

```
# nft add rule inet example_table udp_packets counter
```

（6）为传入的流量创建一个链。例如，在 example_table 中创建一个名为 incoming_traffic 的链来过滤传入的流量

```
# nft add chain inet example_table incoming_traffic { type filter hook input priority 0 \; }
```

（7）将带有文字映射的规则添加到 incoming_traffic

```
# nft add rule inet example_table incoming_traffic ip protocol vmap { tcp : jump tcp_packets,udp : jump udp_packets }
```

（8）要列出流量计数器，请显示 example_table

```
# nft list table inet example_table
table inet example_table {
chain tcp_packets {
counter packets 36379 bytes 2103816
}
chain udp_packets {
counter packets 10 bytes 1559
}
chain incoming_traffic {
type filter hook input priority 0; policy accept;
ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
}
}
```

tcp_packages 和 udp_packages 链中的计数器同时显示接收到的包的数量和字节数

8.在 nftable 中使用可变结果映射

nftables 框架支持可变的结果映射。您可以在一个表中的多个规则中使用这些映射。

文字映射的另一个好处是，您可以更新一个可变的映射，而不需要替换使用它的规则。

当您创建一个可变的结果映射时，您必须指定元素的类型：

ipv4_addr	包含 ipv4 地址或范围的集合，比如 192.0.2.1 或 192.0.2.0/24.
ipv6_addr	包含 ipv6 地址或范围的集合，比如 2001:db8::1 或 2001:db8::1/24
ether_addr	包含 MAC 地址的集合，比如 52:54:00:6b:66:42
inet_proto	包含 internet 协议类型的集合，比如 tcp
inet_service	包含 internet 服务的集合，比如 ssh
mark	包含包标记列表的集合，包标记可以是任何正的 32 位整数值
counter	计数器值，计数器值可以是任何正的 64 位整数值。
quota	配额值，配额值可以是任何正的 64 位整数值。

该示例描述了如何根据源 IP 地址允许或删除传入数据包。使用可变结果映射，您只需要一个规则来配置此场景，而 IP 地址和操作则动态地存储在映射中。该过程还描述了如何从映射中添加和删除条目。

流程如下：

（1）创建一个表。例如，要创建一个名为 example_table 的表来处理 IPv4 包:

```
# nft add table ip example_table
```

（2）创建一个链。例如，在 example_table 中创建一个名为 example_chain 的链

```
# nft add chain ip example_table example_chain { type filter hook input priority 0 \; }
```

（3）创建一个空映射。例如，要为 IPv4 地址创建映射:

```
# nft add map ip example_table example_map { type ipv4_addr : verdict \; }
```

（4）创建使用映射的规则。例如，下面的命令向 example_table 中的 example_chain 添加了一个规则，该规则将操作应用于 example_map 中定义的 IPv4 地址:

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

（5）将 IPv4 地址和相应的操作添加到 example_map:

```
# nft add element ip example_table example_map { 192.0.2.1 : accept, 192.0.2.2 :
```

```
drop }
```

这个示例定义了 IPv4 地址到操作的映射。与上面创建的规则相结合，防火墙接受来自 192.0.2.1 的包，并删除来自 192.0.2.2 的包。

（6）可选地，通过添加另一个 IP 地址和动作语句来增强映射:

```
# nft add element ip example_table example_map { 192.0.2.3 : accept }
```

（7）可选地，从映射中删除一个条目:

```
# nft delete element ip example_table example_map { 192.0.2.1 }
```

（8）可选的，显示规则集：

```
# nft list ruleset
CHAPTER 6. GETTING STARTED WITH NFTABLES
81
table ip example_table {
map example_map {
type ipv4_addr : verdict
elements = { 192.0.2.2 : drop, 192.0.2.3 : accept }
}
chain example_chain {
type filter hook input priority 0; policy accept;
ip saddr vmap @example_map
}
}
```

9. 在 nftable 中配置端口转发

将传入的包转发到不同的本地端口

本节描述如何将端口 8022 上传入的 IPv4 包转发到本地系统上的端口 22 的示例流程如下：

（1）用 ip 地址族创建一个名为 nat 的表

```
# nft add table ip nat
```

（2）将 prerouting 和 postrouting 链添加到表中

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
```

（3）在 prerouting 链中添加一条规则，将端口 8022 上的入站数据包重定向到本地端

口 22:

```
# nft add rule ip nat prerouting tcp dport 8022 redirect to :22
```

10. 将特定本地端口上的传入包转发到不同的主机

您可以使用目标网络地址转换(DNAT)规则将本地端口上的传入数据包转发到远程主机。这使 internet 上的用户能够访问在具有私有 IP 地址的主机上运行的服务。

该过程描述如何将本地端口 443 上传入的 IPv4 包转发到远程 IP 地址为 192.0.2.1 的相同端口号。

流程如下：

（1）用 ip 地址族创建一个名为 nat 的表

```
# nft add table ip nat
```

（2）将 prerouting 和 postrouting 链添加到表中

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
```

（3）在 prerouting 链中添加一条规则，将端口 443 上的入站数据包重定向到 192.0.2.1 上的同一端口:

```
# nft add rule ip nat prerouting tcp dport 443 dnat to 192.0.2.1
```

（4）在出活动链中添加一条规则，以伪装外出流量:

```
# nft add rule ip daddr 192.0.2.1 masquerade
```

（5）使包转发:

```
# echo "sysctl net.ipv4.ip_forward=1" > etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

11. 使用 nftables 限制连接的数量

nft 实用程序的 ct 计数参数使管理员能够限制连接的数量。该过程描述了如何限制传入连接的基本示例。

流程如下：

（1）添加一个规则，该规则只允许从 IPv4 地址到 SSH 端口(22)的两个并发连接，并拒绝来自同一 IP 的所有进一步连接

```
# nft add rule ip example_table example_chain tcp dport ssh meter example_meter { ip saddr ct count over 2 } counter reject
```

（2）可以选择显示在前一步中创建的仪表

```
# nft list meter ip example_table example_meter
table ip example_table {
meter example_meter {
type ipv4_addr
size 65535
elements = { 192.0.2.1 : ct count over 2 , 192.0.2.2 : ct count over 2 }
}
}
```

元素项显示当前与规则匹配的地址。在本例中，元素列出了与 SSH 端口具有活动连接的 IP 地址。注意，输出不显示活动连接的数量，或者如果连接被拒绝。

12. 阻塞试图在一分钟内超过 10 个新传入 TCP 连接的 IP 地址

nftables 框架允许管理员动态更新集合。本节解释如何使用此功能临时阻塞在一分钟内建立了 10 个以上 IPv4 TCP 连接的主机。五分钟后，nftables 自动将 IP 地址从黑名单中删除。

流程如下：

（1）使用 ip 地址族创建筛选表

```
# nft add table ip filter
```

（2）将输入链添加到筛选表

```
# nft add chain ip filter input { type filter hook input priority 0 \; }
```

（3）在过滤表中添加一个名为 blacklist 的集合:

```
# nft add set ip filter blacklist { type ipv4_addr \; flags dynamic, timeout \;
timeout 5m \; }
```

该命令为 IPv4 地址创建一个动态集。timeout 5m 参数定义 nftables 在 5 分钟后自动从集合中删除条目。

（4）添加一个规则，该规则自动将试图在一分钟内建立 10 个以上新 TCP 连接的主机的源 IP 地址添加到黑名单集

```
# nft add rule ip filter input ip protocol tcp ct state new, untracked limit rate
over 10/minute add @blacklist { ip saddr }
```

（5）添加一个规则，删除所有连接从 IP 地址在黑名单集:

```
# nft add rule ip filter input ip saddr @blacklist drop
```

13. 使用计数器创建规则

要确定某个规则是否匹配，可以使用计数器。本节描述如何使用计数器创建新规则。

流程如下：

（1）向链中添加带有计数器参数的新规则。下面的示例添加了一个带有计数器的规则，该规则允许端口 22 上的 TCP 流量，并计数与此规则匹配的数据包和流量:

```
# nft add rule inet example_table example_chain tcp dport 22 counter accept
```

（2）来显示计数器的值

```
# nft list ruleset
table inet example_table {
chain example_chain {
type filter hook input priority 0; policy accept;
tcp dport ssh counter packets 6872 bytes 105448565 accept
}
}
```

14. 向现有规则添加计数器

要确定某个规则是否匹配，可以使用计数器。本节描述如何向现有规则添加计数器。

流程如下：

（1）显示链中的规则，包括它们的句柄

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
chain example_chain { # handle 1
type filter hook input priority 0; policy accept;
tcp dport ssh accept # handle 4
}
}
```

（2）通过使用 counter 参数替换规则来添加计数器。下面的例子替换了前一步中显示的规则，并添加了一个计数器:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22
counter accept
```

（3）显示计数器的值

```
# nft list ruleset
table inet example_table {
chain example_chain {
type filter hook input priority 0; policy accept;
tcp dport ssh counter packets 6872 bytes 105448565 accept
}
}
```

15. 监视与现有规则匹配的包

nftables 中的跟踪功能与 nft 监视器命令结合使用，使管理员能够显示与规则匹配的包。该过程描述如何启用对规则的跟踪以及监视与此规则匹配的包。

流程如下：

（1）显示链中的规则，包括它们的句柄

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
chain example_chain { # handle 1
type filter hook input priority 0; policy accept;
tcp dport ssh accept # handle 4
}
}
```

（2）通过使用 meta nftrace set 1 参数替换规则来添加跟踪功能。下面的例子代替了前一步中显示的规则，并支持跟踪:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 meta
nftrace set 1 accept
```

（3）使用 nft monitor 命令显示跟踪。下面的示例过滤命令的输出，只显示包含 inet example_table example_chain 的条目

```
# nft monitor | grep "inet example_table example_chain"
trace id 3c5eb15e inet example_table example_chain packet: iif "enp1s0" ether
saddr
52:54:00:17:ff:e4 ether daddr 52:54:00:72:2f:6e ip saddr 192.0.2.1 ip daddr
192.0.2.2 ip dscp
cs0 ip ecn not-ect ip ttl 64 ip id 49710 ip protocol tcp ip length 60 tcp sport
```

```
56728 tcp dport
ssh tcp flags == syn tcp window 64240
trace id 3c5eb15e inet example_table example_chain rule tcp dport ssh nftrace
set 1 accept
(verdict accept)
...
```

16. 备份和恢复 NFTABLES 规则集

备份 nftables 规则集到文件

流程如下：

（1）以 nft list ruleset 格式保存

```
# nft list ruleset > file.nft
```

（2）以 json 格式保存

```
# nft -j list ruleset > file.json
```

从文件中恢复 nftables 规则集

流程如下：

（1）以 nft list ruleset 格式恢复

```
# nft -f file.nft
```

（2）以 json 格式恢复

```
# nft -j -f file.json
```

9.8 安全审计(Audit)

CGSL 内核有用日志记录事件的能力，比如记录系统调用和文件访问。然后，管理员可以评审这些日志，确定可能存在的安全裂口，比如失败的登录尝试，或者用户对系统文件不成功的访问，这种功能称为审计(Audit)，CGSL 审计功能由 auditd 服务提供。要使用审计系统，可采用下面的步骤：

(1) 配置审计守护进程(auditd)。

(2) 添加审计规则和观察器来收集所需的数据。

(3) 启动守护进程，它启用了内核中的审计系统并开始进行日志记录。

(4) 通过生成审计报表和搜索日志来周期性地分析数据。

9.8.1 配置审计守护进程(auditd)

审计守护进程(auditd)的默认配置文件为/etc/audit/auditd.conf，用户可以修改该文件定制产生的审计日志。配置文件示例如下：

```
log_file = /var/log/audit/audit.log
log_format = RAW
log_group = root
priority_boost = 4
flush = INCREMENTAL
freq = 20
num_logs = 4
disp_qos = lossy
dispatcher = /sbin/audispd
name_format = NONE
##name = mydomain
max_log_file = 5
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
##tcp_listen_port =
tcp_listen_queue = 5
tcp_max_per_addr = 1
##tcp_client_ports = 1024-65535
tcp_client_max_idle = 0
enable_krb5 = no
krb5_principal = auditd
##krb5_key_file = /etc/audit/audit.key
```

其中，

log_file

审计日志文件的完整路径。

log_format

写日志时要使用的格式。

priority_boost

审计应采用多少优先级推进守护进程。必须是非负数。0 表示没有变化。

flush

多长时间向日志文件中写一次数据。值可以是 NONE、INCREMENTAL、DATA 和 SYNC 之一。如果设置为 NONE，则不需要做特殊努力来将数据刷新到日志文件中。如果设置为 INCREMENTAL，则用 freq 选项的值确定多长时间发生一次向磁盘的刷新。如果设置为 DATA，则审计数据和日志文件一直是同步的。如果设置为 SYNC，则每次写到日志文件时，数据和元数据是同步的。

freq

如果 flush 设置为 INCREMETNAL，审计守护进程在写到日志文件中前从内核中接收的记录数。

num_logs

max_log_file_action 设置为 ROTATE 时要保存的日志文件数目。必须是 0~99 之间的数。如果设置为小于 2，则不会循环日志。默认为 0，意味着从来不循环日志文件。

dispatcher

当启动这个守护进程时，由审计守护进程自动启动程序。所有守护进程都传递给这个程序。可以用它来进一步定制报表或者以与您的自定义分析程序兼容的不同格式产生它们。非必须选项。

disp_qos

控制调度程序与审计守护进程之间的通信类型。有效值为 lossy 和 lossless。如果设置为 lossy，若审计守护进程与调度程序之间的缓冲区已满(缓冲区为 128 千字节)，则发送给调度程序的引入事件会被丢弃。然而，只要 log_format 没有设置为 nolog，事件就仍然会写到磁盘中。如果设置为 lossless，则在向调度程序发送事件之前和将日志写到磁盘之前，调度程序会等待缓冲区有足够的空间。

max_log_file

以兆字节表示的最大日志文件容量。当达到这个容量时，会执行 max_log_file_action 指定的动作。

max_log_file_action

当达到 max_log_file 的日志文件大小时采取的动作。值必须是 IGNORE、SYSLOG、SUSPEND、ROTATE 和 KEEP_LOGS 之一。如果设置为 IGNORE，则在日志文件达到 max_log_file 后不采取动作。如果设置为 SYSLOG，则当达到文件容量时会向系统日志/var/log/messages 中写入一条警告。如果设置为 SUSPEND，则当达到文件容量后不会向日志文件写入审计消息。如果设置为 ROTATE，则当达到指定文件容量后会循环日志文件，但是只会保存一定数目的老文件，这个数目由 num_logs 参数指定。老文件的文件名将为 audit.log.N，其中 N 是一个数字。这个数字越大，则文件越老。如果设置为 KEEP_LOGS，则会循环日志文件，但是会忽略 num_logs 参数，因此不会删除日志文件。

space_left

以兆字节表示的磁盘空间数量。当达到这个水平时，会采取 space_left_action 参数中。

space_left_action

当磁盘空间量达到 space_left 中的值时，采取这个动作。有效值为 IGNORE、SYSLOG、EMAIL、SUSPEND、SINGLE 和 HALT。如果设置为 IGNORE，则不采取动作。如果设置为 SYSLOG，则向系统日志/var/log/messages 写一条警告消息。如果设置为 EMAIL，则从 action_mail_acct 向这个地址发送一封电子邮件，并向/var/log/messages 中写一条警告消息。如果设置为 SUSPEND，则不再向审计日志文件中写警告消息。如果设置为 SINGLE，则系统将在单用户模式下。如果设置为 SALT，则系统会关闭。

action_mail_acct

负责维护审计守护进程和日志的管理员的电子邮件地址。如果地址没有主机名，则假定主机名为本地地址，比如 root。必须安装 sendmail 并配置为向指定电子邮件地址发送电子邮件。

admin_space_left

以兆字节表示的磁盘空间数量。用这个选项设置比 space_left_action 更多的主动性动作，以防万一 space_left_action 没有让管理员释放任何磁盘空间。这个值应小于 space_left_action。如果达到这个水平，则会采取 admin_space_left_action 所指定的动作。

admin_space_left_action

当自由磁盘空间量达到 admin_space_left 指定的值时，则采取动作。有效值为 IGNORE、SYSLOG、EMAIL、SUSPEND、SINGLE 和 HALT。与这些值关联的动作与 space_left_action 中的相同。

disk_full_action

如果含有这个审计文件的分区已满，则采取这个动作。可能值为 IGNORE、SYSLOG、SUSPEND、SINGLE 和 HALT。与这些值关联的动作与 space_left _action 中的相同。

如果不循环审计日志文件，则含有/var/log/audit/的分区可能变满并引起系统错误。因此，建议让/var/log/audit/位于一个单独的专用分区。

disk_error_action

如果在写审计日志或循环日志文件时，检测到错误时采取的动作。其值必须是 IGNORE、SYSLOG、SUSPEND、SINGLE 和 HALT 之一。这些值的含义与 space_left_action 中的相同。

9.8.2 编写审计规则

要添加审计规则，可在/etc/audit/audit.rules 文件中使用下面的语法：

```
-a <list>,<action> <options>
```

➢ 列表名(list)必须是下列名称之一

task

每个任务的列表。只有当创建任务时才使用。只有在创建时就已知的字段(比如 UID)才可以用在这个列表中。

entry

系统调用条目列表。当进入系统调用确定是否应创建审计时使用。

exit

系统调用退出列表。当退出系统调用以确定是否应创建审计时使用。

user

用户消息过滤器列表。内核在将用户空间事件传递给审计守护进程之前使用这个列表过滤用户空间事件。有效的字段只有 uid、auid、gid 和 pid。

exclude

事件类型排除过滤器列表。用于过滤管理员不想看到的事件。用 msgtype 字段指定您不想记录到日志中的消息。

➢ 动作（action）必须下面的参数之一：

never

不生成审计记录。

always

分配审计上下文，总是把它填充在系统调用条目中，总是在系统调用退出时写一个审

计记录。

➢ <options>可以包括下面几个选项中的一个或多个。

-S <syscall>

根据名称或数字指定一个系统调用，要指定所有系统调用，可使用 all 作为系统调用名称。如果程序使用了这个系统调用，则开始一个审计记录。可以为相同的规则指定多个系统调用，每个系统调用必须用-S 启动。在相同的规则中指定多个系统调用。

- F <name[=,!=,<,>,<=]value>

指定一个规则字段。如果为一个规则指定了多个字段，则只有所有字段都为真才能启动一个审计记录。每个规则都必须用-F 启动，最多可以指定 64 个规则。如果用用户名和组名作为字段，而不是用 UID 和 GID，则会将它们解析为 UID 和 GID 以进行匹配。下面是有效的字段名：

pid 进程 ID。

ppid 父进程的进程 ID。

uid 用户 ID。

euid 有效用户 ID。

suid 设置用户 ID。

fsuid 文件系统用户 ID。

gid 组 ID。

egid 有效组 ID。

sgid 设置组 ID。

fsgid 文件系统组 ID。

auid 审计 ID，或者用户登录时使用的原始 ID。

msgtype 消息类型号。只应用在排除过滤器列表上。

pers OS Personality Number。

Arch 系统调用的处理器体系结构。指定精确的体系结构，比如 i686(可以通过 uname -m 命令检索)或者指定 b32 来使用 32 位系统调用表，或指定 b64 来使用 64 位系统调用表。

inode Inode Number。

exit 从系统调用中退出值。

success 系统调用的成功值。1 表是真/是，0 表示假/否。

a0，a1，a2，a3 分别表示系统调用的前 4 个参数。只能用数字值。

Key 设置用来标记事件的审计日志事件消息的过滤键。当添加观察器时，类似于使用-k 选项。

obj_user 资源的 SELinux 用户。

obj_role 资源的 SELinux 角色。

obj_type 资源的 SELinux 类型。

obj_lev_low 资源的 SELinux 低级别。

obj_lev_high 资源的 SELinux 高级别。

subj_role 程序的 SELinux 角色。

subj_type 程序的 SELinux 类型。

subj_sen 程序的 SELinux 敏感性。

subj_clr 程序的 SELinux 安全级别(clearance)。

-a 选项向列表末尾添加规则。要向列表开头添加规则，可用-A 替换-a。删除语法相同的规则，用-d 替换-a。要删除所有规则，可指定-D 选项。审计规则示例：

```
#Record all file opens from user 501
#Use with caution since this can quickly
#produce a large quantity of records
-a exit,always -S open -F uid=501 -F key=501open
#Record file permission changes
 -a entry,always -S chmod
```

审计规则日志消息例子：

```
type=SYSCALL msg=audit(1168206647.422:5227): arch=c000003e
syscall=success=no exit=-2 a0=7fff37fc5a40 a1=0 a2=2aaaaaaab000 a3=0
items=1 ppid=26640 pid=2716 auid=501 uid=501 gid=501 euid=501 suid=501
fsuid=501 egid=501 sgid=501 fsgid=501 tty=pts5 comm="vim"
exe="/usr/bin/vim" key="501open"。
```

9.8.3 使用审计监控文件

CGSL 审计系统也允许管理员监控文件和目录。通过设置观察器在一个文件或目录上，

可以监控文件和目录上发生的指定动作，如：打开、读写和执行。

示例如下：

/etc/security/	-w /etc/security -p wa -k ETC_SECURITY
/etc/selinux/	-w /etc/selinux -p wa -k ETC_SELINUX
/etc/bashrc	-w /etc/bashrc -p wa -k ETC_BASHRC
/etc/profile	-w /etc/profile -p wa -k ETC_PROFILE

其中，-w 表示观察(watch)的对象，即需要观察的文件或目录；-p 用于指定观察的动作，wa 表示观察“write”和“append”动作，即当被观察对象发生“write”和“append”动作时，记录相关的审计日志信息；-k 用于指定本条规则的 key，利用该关键字可以过滤指定规则相关的审计日志信息。

设置好审计规则后，auditd服务将根据设定的规则，对指定文件和目录进行监控，并将相关的审计日志信息默认记录于/var/log/audit/audit.log文件中，用户可以通过该日志文件查看相关的审计信息。由于记录的审计信息量比较大，也可以使用ausearch工具通过审计规则中设定的key来对审计信息进行过滤，获取用户需要的信息，如：如果需要查看/etc/selinux/目录相关的审计信息，可以使用如下命令来获取：

```
#ausearch -k ETC_SELINUX
```

审计观察器的示例日志如下：

```
time->Mon Dec 20 10:16:05 2010
type=CONFIG_CHANGE    msg=audit(1292811365.978:24):    auid=4294967295
op=add rule key="ETC_SELINUX" list=4 res=1
```

提示：如果在运行守护进程时通过修改配置文件/etc/audit/audit.rules 修改审计规则，则须要以根用户身份用 systemctl auditd restart 命令启用修改。

9.9 日志系统

系统日志是操作系统安全事件的重要记录机制。CGSL 操作系统提供了完善的系统日志体系，日志记录内容涵盖常规日志、用户登录日志、用户操作日志、系统性能日志、系统审计日志等，为操作系统安全事件的记录、分析、追踪提供了有力的支撑。

9.9.1 定位日志文件

大多数日志文件都位于目录/var/log/目录下。一些应用程序，如 httpd，samba 等，这些应用程序会就在目录/var/log 下生成一个存放对应应用程序日志文件的目录。

在存放日志文件的目录中，我们会注意到每个日志文件后面都有一个数字编号，这些编号是在每个轮换周期结束的时候，有一个脚本或者工具程序更改每个文件的名字，然后把较早的数据向文件链的结尾推。例如，假设某个日志文件的名字叫做 logfile，则它的备份文件可能叫做 logfile.1、logfile.2，依此类推。如果每周轮换一次，并且保存 8 周的数据，那么就会有一个 logfile.8 文件但没有 logfile.9 文件。每周随着 logfile.7 文件覆盖 logfile.8 文件，logfile.8 中原来的数据就没了。由于日志文件在不停得轮转使用，所以每个日志文件都不会很大。日志文件轮转里有一个 cron 守护线程，它能依照/etc/logrotate.conf 配置文件和/etc/logrotate.d/目录下的配置文件在各日志文件之间进行轮转。默认情况下，轮转周期为一周，日志保存周期为 4 周。

大多数日志文件都是文本文件的格式，我们可以直接使用文本阅读器进行查看，例如 Vi、Emaces 工具等。某些日志文件可供系统中的所有用户查看，管理员权限的 root 帐户可以查看绝大多数的日志文件。

9.9.2 重要日志说明

/var/log/messages 日志是核心系统日志文件。它包含了系统启动时的引导消息，以及系统运行时的其他状态消息。IO 错误、网络错误和其他系统错误都会记录到这个文件中。其他信息，比如某个人的身份切换为 root，也在这里列出。如果服务正在运行，比如 DHCP 服务器，可以在 messages 文件中观察它的活动。通常，/var/log/messages 是在做故障诊断时首先要查看的文件。

/var/log/secure 日志是用户登录信息日志文件。它包含用户登录登出信息。

/var/log/maillog 日志是系统的邮件日志。它包含邮件服务器的发送和接收邮件信息。

/var/log/dmesg 日志是系统硬件的日志。

/var/log/mcelog 日志是记录系统运行过程中发现的硬件错误信息，包含内存错误，io 错误等日志。

9.9.3 rsyslog

rsyslog 是 CGSL V6 版本默认的日志管理软件，rsyslog 是一个 syslogd 的多线程增强版，它提供了 MySQL 和完全可配置的输出格式（包括大时间戳）的支持。

9.9.3.1 rsyslog 配置

rsyslog 的配置文件是</etc/rsyslog.conf>，如下介绍其基本配置。

/etc/rsyslog.conf 根据如下的格式定义规则：

```
facility.level action
```

设备.优先级 动作

facility.level 字段也被称为 seletor（选择条件），选择条件和动作之间用空格或 tab 分割开。#号开头的是注释，空白行会自动跳过。

■ **facility（设备）**

facility 定义日志消息的范围，其可使用的 key 有：

auth －由 pam_pwdb 报告的认证活动

authpriv －包括特权信息如用户名在内的认证活动

cron －与 cron 和 at 有关的计划任务信息

daemon －与 inetd 守护进程有关的后台进程信息

kern －内核信息，首先通过 klogd 传递

lpr －与打印服务有关的信息

mail －与电子邮件有关的信息

mark － syslog 内部功能用于生成时间戳

news －来自新闻服务器的信息

syslog －由 syslog 生成的信息

user －由用户程序生成的信息

uucp －由 uucp 生成的信息

local0-local7 －与自定义程序使用

* 通配符代表除了 mark 以外的所有功能

■ **level 级别（优先级）**

level 定义消息的紧急程度。按严重程度由高到低顺序排列为：

emerg －该系统不可用，等同 panic

alert －需要立即被修改的条件

crit －阻止某些工具或子系统功能实现的错误条件

err －阻止工具或某些子系统部分功能实现的错误条件，等同 error

warning －预警信息，等同 warn

notice －具有重要性的普通条件

info －提供信息的消息

debug －不包含函数条件或问题的其他信息

none －没有重要级，通常用于排错

* 所有级别，除了 none

■ **selector 选择条件**

通过小数点符号“.”把 facility 和 level 连接在一起则成为 selector（选择条件）。

可以使用分号“;”同时定义多个选择条件。也支持三个修饰符：

* － 所有日志信息

= － 等于，即仅包含本优先级的日志信息

! － 不等于，本优先级日志信息除外

■ **action（动作）**

由前面选择条件定义的日志信息，可执行下面的动作：

file－指定日志文件的绝对路径

terminal 或 print －发送到串行或并行设备标志符，例如/dev/ttyS2

@host －远程的日志服务器

username －发送信息本机的指定用户信息窗口中，但该用户必须已经登陆到系统中

named pipe －发送到预先使用 mkfifo 命令来创建的 FIFO 文件的绝对路径

例如：

```
*.info;mail.none;news.none;authpriv.none;cron.none /var/log/messages
#把除邮件、新闻组、授权信息、计划任务等外的所有通知性消息都写入 messages 文件中。
mail,news.=info /var/adm/info
#把邮件、新闻组中仅通知性消息写入 info 文件，其他信息不写入。
mail.*;mail.!=info /var/adm/mail
#把邮件的除通知性消息外都写入 mail 文件中。
mail.=info /dev/tty12
#仅把邮件的通知性消息发送到 tty12 终端设备
*.* @finlandia
#把所有信息都导向到 finlandia 主机（通过/etc/hosts 或 dns 解析其 IP 地址）
```

♣ 提示：默认的配置文件在多数情况下应该可以胜任。如果要进行更细致的定制，请阅读 rsyslog 的手册页。

9.9.3.2 设置 rsyslog 接收远程日志

默认情况下，rsyslog 进程是不能接受其他日志服务器发过来的消息的。而通过修改其启动参数，可实现远程日志接收功能。

修改/etc/sysconfig/rsyslog 文件中的 SYSLOGD_OPTIONS 配置，其中：

-r ：打开接受外来日志消息的功能；

-x ：关闭自动解析对方日志服务器的 FQDN 信息，这能避免 DNS 不完整所带来的麻烦；

-m ：修改 syslog 的内部 mark 消息写入间隔时间（0 为关闭），例如 240 为每隔 240 分钟写入一次“--MARK--”信息；

-c ：打开兼容模式

CGSL 默认情况下，/etc/sysconfig/rsyslog 实际配置文件参数为：

```
SYSLOGD_OPTIONS="-c2 -r -x"
```

修改配置并保存后，重启服务即可使其生效：

```
#systemctl restart rsyslog
```

客户机只要通过修改 rsyslog.conf，定义动作为@主机或 IP，即可发送日志信息到本服务器中。

9.9.4 Logrotate

logrotate 是 CGSL 系统日志的管理工具。它可以轮换，压缩，邮件系统日志文件。默认的 logrotate 被加入 cron 的/etc/cron.daily 中作为每日任务执行。

/etc/logrotate.conf 为其默认配置文件指定每个日志文件的默认规则。/etc/logrotate.d/* 为/etc/logrotate.conf 默认包含目录其中文件也会被 logrotate 读取。指明每个日志文件的特定规则。var/lib/logrotate.status 中默认记录 logrotate 上次轮换日志文件的时间。

logrotate 的配置文件是 /etc/logrotate.conf。主要参数如下表：

参数	功能描述
compress	通过 gzip 压缩转储以后的日志
nocompress	不需要压缩时，用这个参数
copytruncate	用于还在打开中的日志文件，把当前日志备份并截断
nocopytruncate	备份日志文件但是不截断
create mode owner group	转储文件，使用指定的文件模式创建新的日志文件
nocreate	不建立新的日志文件
delaycompress	与 compress 一起使用时，转储的日志文件到下一次转储时才压缩
nodelaycompress	覆盖 delaycompress 选项，转储同时压缩。
errors address	专储时的错误信息发送到指定的 Email 地址
ifempty	即使是空文件也转储，这个是 logrotate 的缺省选项。
notifempty	如果是空文件的话，不转储
mail address	把转储的日志文件发送到指定的 E-mail 地址
missingok	如果日志不存在则忽略该警告信息
nomail	转储时不发送日志文件
olddir directory	转储后的日志文件放入指定的目录，必须和当前日志文件在同一个文件系统
noolddir	转储后的日志文件和当前日志文件放在同一个目录下
prerotate/endscript	在转储以前需要执行的命令可以放入这个对，这两个关键字必须单独成行
postrotate/endscript	在转储以后需要执行的命令可以放入这个对，这两个关键字必须单独成行

daily	指定转储周期为每天
weekly	指定转储周期为每周
monthly	指定转储周期为每月
rotate count	指定日志文件删除之前转储的次数，0 指没有备份，5 指保留 5 个备份
tabootext [+] list	不转储指定扩展名的文件，缺省的扩展名是：.rpm-orig .rpmsave size size 当日志文件到达指定的大小时才转储，可以指定 bytes(缺省)以及 K 或者 M
dateext	使用日期作为命名格式
dateformat	配合 dateext 使用，紧跟在下一行出现，定义文件切割后的文件名，必须配合 dateext 使用，只支持 %Y %m %d %s 这四个参数

CGSL 中 logrotate 缺省的配置如下

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
dateext
# uncomment this if you want your log files compressed
#compress
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
```

```
        minsize 1M
    rotate 1
}
/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}
# system-specific logs may be also be configured here.
```

■ **默认参数解析：**

weekly：指定所有的日志文件每周转储一次

rotate 4：指定转储文件的保留 4 份

create：指定 logrotate 自动建立新的日志文件，新的日志文件具有和原来的文件一样的权限。

dateext：使用日期作为命名格式

include /etc/logrotate.d：选项允许系统管理员把分散到/etc/logrotate.d 目录下几个文件的转储信息，集中到一个主要的配置文件。、

■ **为指定的文件配置转储参数：**

经常需要为指定文件配置参数，一个常见的例子就是每月转储/var/log/wtmp。为特定文件而使用的参数格式是：

```
/full/path/to/file
{
option(s)
}
```

下面的例子就是每月转储/var/log/wtmp 一次：

```
#Use logrotate to rotate wtmp
/var/log/wtmp
{
monthly
rotate 1
```

```
}
```

Logrotate 默认是按天执行，任务计划存放在/etc/cron.daily/logrotate 文件里，如果需要修改 logrotate 立即生效，可执行以下命令：

```
#logrotate /etc/logrotate.conf
```

9.10 服务安全

CGSL 系统为满足用户需求默认启动了相关服务，而带来一定的安全风险，可以通过相关系统命令对系统中的服务进行控制。包括：服务的启动、停止和状态查询；查看系统服务在各运行级别下的默认启动情况；在系统启动时默认启动指定服务；去除在系统启动时默认启动的服务。

9.11 传输通道安全(ipsec vpn)

待续。

9.12 SELinux

9.12.1 简介

SELinux 是在 linux 内核的一种强制访问控制机制的实现。SELinux 构架提供了多种增强的访问控制策略，包含基于类型加强，基于角色的访问控制，和多级别/多安全。

9.12.2 SELinux 的工作流程

SELinux 是采用 LSM 方式集成到 Linux 内核的安全构架，为 linux 系统提供了 MAC，它是一种柔性的强制访问控制方式。传统的 Linux 使用的是一种随意的访问控制方式----DAC。在这种方式下，某一用户运行的应用或进程拥有该用户的所有权限，可以访问这个用户能访问这个用户能访问的文件、套接口等对象。而采用 MAC 的内核可以保护系统免受一些错误的或恶意的应用程序对系统的破坏。

SELinux 为系统中的每一个用户、应用、进程和文件定义访问权力，然后把这些实体之间的交互定义成安全策略，再用安全策略来控制各种操作是否允许。

如图 9-1，当进程等访问者向系统提出对文件等访问对象的访问请求时，位于内核的策略增强服务器收到了这个请求，就到访问向量缓存 AVC 中查找是否有有关该请求的策略。如果有，就按照该策略来决定是否允许访问；如果没有，就继续要求安全增强服务器

到访问策略矩阵中查找是否有有关该请求的策略。如果有允许访问策略，就允许访问，否则，将禁止访问，并把“ave denied”类型的日志写到/var/log/messages 文件中。

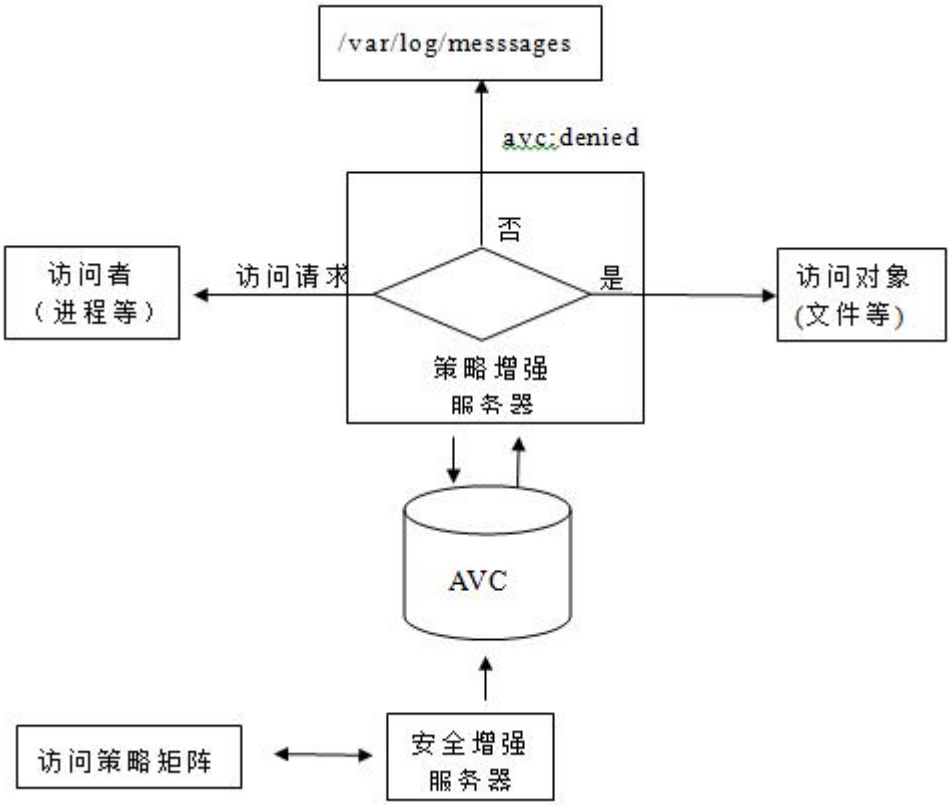


图 9-1 SELinux 的工作原理

9.12.3 SELinux 中的安全上下文

SELinux 系统中的进程和文件都标记了 SELinux 的上下文，这个上下文包含了许多动有用的信息：SELinux 用户、角色、类型、级别等。这此安全上下文都用来辅助地访问控制决策。

当我们要查看查文件或目录的安全上下文的属性时，可以在 ls 命令加“-Z”选项，如

```
# clear
# ls    -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
```

同时，我们也可以通过“-Z”选项，来显示某进程的安全上下文

```
[root@localhost www]# ps -Z |grep bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 17605 pts/4 00:00:00
bash
```

SELinux 上下文的组成为：

SELinux **user:role:type:level**

user----表示每一个 Linux 系统用户都通过 SELinux 机制被映射为一个 SELinux 用户，这也使得 Linux 用户可以继承 SELinux 用户的访问权限。我们可以通过 semanage login –l 命令，来查看 SELinux 用户和 Linux 用户的映射关系，如下所示：

```
# semanage login -l

Login Name          SELinux User        MLS/MCS Range        Service

__default__         unconfined_u        s0-s0:c0.c1023       *
root                unconfined_u        s0-s0:c0.c1023       *
system_u            system_u            s0-s0:c0.c1023       *
```

role---表示 SELinux 采用了基于角色的访问控制机制。而 role 是 RBAC 机制中的一个属性。SELinux 用户被授权为对应的可访问域。因此，角色是域和 SELinux 用户之间联系的媒介。通过角色可以决定 SELinux 用户可以进入哪些域。而最终决定了 SELinux 用户可以访问哪些对象类型。通过这种机制可以降低权限提升的风险。

Type---表示类型是类型强制机制的一个属性。这个类型定义了进程类型和文件类型。SELinux 机制规则明确定义了类型间互相访问、域访问类型或者是域间相互访问的规则和许可。只有某条 SELinux 机制规则允许的情况下，才允许上述的访问发生。

Level---级别是 MLS 和 MCS 机制的另一个重要属性。

9.12.4 SElinux 的配置

CGSL V6 中， SELinux 可以使用/etc/selinux 目录下的文件来进行配置。其中，/etc/selinux/config 文件是 SELinux 的主配置文件,其可以控制系统下一次启动过程中载入哪个策略，以及运行在哪个模式下。其初始内容如下：

```
[root@localhost sysconfig]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are pro
tected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted


[root@localhost sysconfig]#
```

图 9-2 SELinux 主配置文件

/etc/selinux/config 配置文件实际包含两项设置，一项是 SELINUX 选项，可以是 enforcing、permissive 和 disabled 3 个值，分别表示强制、随意和禁用 SELinux 3 种选择。

Enforcing---策略被完整执行，这是 SELinux 的主要模式，应该在所有要求增强的 Linux 安全性的操作系统中使用。

Permissive---SELinux 策略规则不被强制执行，相反，只是审核遭受拒绝的消息，除些之外，SELinux 不会影响系统安全性，这个模式在调试和测试一个策略时非常有用。

Disabled---SELinux 内核完全关闭的，只有系统启动时策略载入前系统才会处于 disabled 模式，这个模式和 permissive 模式有所不同 permissive 模式有 SELinux 内核特征操作，但不会拒绝任何访问，只是进行审核，在 disabled 模式下，SELinux 将不会有任何动作，只有在极端环境下才使用这个模式，例如，当策略错误阻止用户登录系统时，即使在 permissive 模式下也有可能发生这种事性，或用户不想使用 SELinux。

另一个选项是 SELINUXTYPE,它可以是 targetd、minimum 和 mls 三个值。Targeted 表示对大多数用户进程没有限制，只对指定服务放在被策略限制的特定安全域的中。Minimum 表示对 target 策略进行修改，只对选定的进程进行保护。Mls 表示所有进程被分到被策略限制细微粒的安全域中。

此外，还可以使用 setenforce 命令实时的修改 SELinux 的运行模式：

/usr/sbin/setenforce 1 设置 SELinux 成为 enforcing 模式

/usr/sbin/setenforce 0 设置 SELinux 成为 permissive 模式

9.12.5 启动与关闭 SELinux

1、检查 selinux 的运行状态

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28 检查防火墙 nat 规则
#iptables -t nat -L -n
```

2、selinux 运行在 permissive 模式，有两种方法：

SELinux 在 Enforcing 的运行状态时

```
# setenforce 0
```

这种方法会在系统重启后，恢复原来的状态。

通过修改/etc/selinux/config 配置文件，把 SELINUX 改为 permissive，并重启系统

```
]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are
protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

这种方法可以使系统重启后，始终保持这个状态。

3、selinux 运行在 Enforcing 模式，有两种方法：

SELinux 在 permissive 的运行状态时

```
# setenforce 1
```

这种方法会在系统重启后，恢复原来的状态。

通过修改/etc/selinux/config 配置文件，把 SELINUX 改为 enforcing，并重启系统

```
]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX= enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#        minimum - Modification of targeted policy. Only selected processes are
protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

这种方法可以使系统重启后，始终保持这个状态。

4、selinux 运行在 disabled 模式，有两种方法：

通过修改/etc/selinux/config 配置文件，把 SELINUX 改为 disabled，并重启系统

```
]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
```

```
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX= disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are
protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

这种方法可以使系统重启后，始终保持这个状态。

9.12.6 与 SELinux 有关的日志文件

SELinux 的决策都被缓存，如允许或不允许访问。这种缓存被称为访问向量缓存（AVC）。当 SELinux 拒绝访问时，这拒绝消息会被记录。这种消息也被称为”AVC 否认”，并且依赖一些守护进程，把这些消息记录到不同的位置。

守护进程	记录的位置
auditd 开启	/var/log/audit/audit.log
auditd 关闭,rsyslogd 开启	/var/log/messages
Setroubleshootd,rsyslogd,和 auditd 开启	把/var/log/audit/audit.log 中易于阅读的否定消息发送到/var/log/messages 日志中

当以 X Window 的方式运行 CGSL V6 时，CGSL V6 提供的 SELinux Troubleshooter 工具可以详细的分析 SELinux 为什么拒绝访问，并尽可能提供允许访问的决解方法。

使用 SELinux Troubleshooter 工具查看遭到 SELinux 拒绝访问的详细消息。要从桌面上启动这个程序，可点击【应用程序】->【杂项】->【SELinux 故障排除工具】

第 10 章

网络

CGSL 系统应用的绝大部分场景都是需要使用网络的，CGSL 系统提供了完善的网络支持，为用户提供所需的网络服务。本章主要介绍 CGSL 系统中网络相关配置。

相比 CGSL V5，在 CGSL V6 里面已经没有 network 服务，是通过 NetworkManager 服务控制网络服务。

10.1 网络配置

CGSL V6 开始，默认不再使用 eth0、eth1 等名字命令网卡设备，网卡名称遵循以下规律：

1、基于固件名称的板载网卡的命名，比如 eno1

2、基于 PCI 扩展插槽的热插拔网卡设备的命名，比如：ens1

3、基于总线号的命名，比如 enp2s0

4、基于 MAC 地址的命名，比如 enx78e7d1ea46da

5、其他设备，采用传统方式命名，比如 eth0

10.1.1 使用 NetworkManager 服务管理网络

启动 NetworkManager：

```
#systemctl start NetworkManager
```

使能 NetworkManager：

```
#systemctl enable NetworkManager
```

10.1.1.1 NetworkManager 服务特点

1、一个设备对应一个网络接口；

2、一整套关于某块设备的网络配置参数的集合称为一个连接件，一个设备可对应存在多个连接件；

3、连接件被激活后，对应的网络配置才生效；对于同一块设备，不能同时激活多个连接件；

4、每一个连接件必须拥有一个名称和一个唯一的 ID；

5、连接件对应的永久生效的配置文件保存在 /etc/sysconfig/network-scripts/ifcfg-NAME，NAME 对应着相应的连接件名称，该文件在需要时可手动修改；

6、在命令提示窗口中，可使用 nmcli 或 nmtui 工具创建和编辑连接件。

10.1.1.2 使用 nmcli 工具管理配置网络

Nmcli 工具配置网络会使得命令非常长，所以 Nmcli 子命令和配置项名称均支持 tab 键自动补全。使用 Nmcli dev status 查看网卡设备信息，如图 10-1：

```
[root@localhost ~]# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
ens33   ethernet  connected  ens33
ens37   ethernet  connected  Wired connection 1
ens38   ethernet  connected  Wired connection 2
lo      loopback  unmanaged  --
[root@localhost ~]#
```

图 10-1 nmcli 命令查看网卡接口状态。

图中显示服务器一共有 3 块网卡，网卡类型均是 ethernet，连接状态。Ens33 这块网卡当前生效的连接件名称是 ens33，对应的配置文件名在 /etc/sysconfig/network-scripts/ifcfg-ens33。当/etc/sysconfig/network-scripts/目录下没有找到对应的连接件名称时，系统对网卡分配的默认连接件，例如：“Wired Connection 1”和“Wired Connection 2”，表示使用 DHCP 获取 IP。

添加网络连接件：nmcli connection add 命令可用于添加一个连接件，指定的配置内容将写入到对应的 ifcfg 文件中。添加连接件时，不要使用已经存在的名称。以下命令将添加一个新的连接件，并且默认使用 DHCP 方式获取 IPv4 地址，新的配置文件将保存到

/etc/sysconfig/network-scripts/ifcfg-ens_net5 中。

```
# nmcli connection add con-name ens33_net5 type Ethernet ifname ens33
```

或者在添加连接件时，使用以下命令配置指定 IP：

```
# nmcli connection add con-name ens33_net5 type Ethernet ifname ens33 \
> ip4 192.168.5.110/24 gw4 192.168.5.1
```

注：使用以上命令之后，需要使用 nmcli connection up nes33_net5 将其激活才能生效。

修改网络配置件内容：除了使用 vi 工具直接编辑配置对应 ifcfg 文件外，还可使用 nmcli 工具进行编辑和保存。执行 Nmcli connection show con-Name 命令，可以查看对应 connection 的详细参数，如下图 10-2，左边是配置参数项，右边是参数值。

```
[root@localhost ~]# nmcli connection show ens33_net5
connection.id:                          ens33_net5
connection.uuid:                        6ca52fd7-df3d-4420-9cf5-fb35e69f399e
connection.interface-name:              ens33
connection.type:                        802-3-ethernet
connection.autoconnect:                 yes
connection.timestamp:                   0
connection.read-only:                   no
connection.permissions:
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.secondaries:
connection.gateway-ping-timeout:        0
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          yes
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.mac-address-blacklist:
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:
ipv4.method:                            manual
ipv4.dns:
ipv4.dns-search:
ipv4.addresses:                         { ip = 192.168.5.110/24, gw = 192.168.5.1 }
ipv4.routes:
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
ipv4.never-default:                     no
ipv4.may-fail:                          yes
ipv6.method:                            auto
ipv6.dns:
ipv6.dns-search:
ipv6.addresses:
ipv6.routes:
ipv6.ignore-auto-routes:                no
ipv6.ignore-auto-dns:                   no
ipv6.never-default:                     no
ipv6.may-fail:                          yes
ipv6.ip6-privacy:                       -1 (unknown)
ipv6.dhcp-hostname:                     --
[root@localhost ~]#
```

图 10-2

找到要修改的参数项，比如修改 ens33_net5 这个 connection 的 IP 地址，参数项是 ipv4.addresses，执行以下命令修改：

```
nmcli con mod ens33_net5   ipv4 . addresses "192.168.5.120/24 192.168.5.1"
```

注：在指定 IP 时，必须将 ipv4.method 设置为 manual。

如果需要配置多个值，使用”+”号添加，比如配置备用 DNS，配置参数项是 ipv4.dns

```
nmcli con mod ens33_net5 +ipv4.dns “8.8.4.4”
```

同理，如果要删除某个项，使用“-”号。

下表列出了 nmcli 和修改 ifcfg-*两种配置方式的对比：

nmcli con mod	ifcfg-* file	Effect
ipv4.method manual	BOOTPROTO=none	IPv4 addresses configured statically.
ipv4.method auto	BOOTPROTO=dhcp	Will look for configuration settings from a DHCPv4 server. If static addresses are also set, will not bring those up until we have information from DHCPv4.
ipv4.addresses "192.0.2.1/24 192.0.2.254"	IPADDR0=192.0.2.1 PREFIX0=24 GATEWAY0=192.0.2.254	Sets static IPv4 address, network prefix, and default gateway. If more than one is set for the connection, then instead of **0**, the **ifcfg-*** directives end with 1, 2, 3 and so on.
ipv4.dns 8.8.8.8	DNS0=8.8.8.8	Modify **/etc/resolv.conf** to use this **nameserver**.
ipv4.dns-search example.com	DOMAIN=example.com	Modify **/etc/resolv.conf** to use this domain in the **search** directive.
ipv4.ignore-auto-dns true	PEERDNS=no	Ignore DNS server information from the DHCP server.
connection.autoconnect yes	ONBOOT=yes	Automatically activate this connection at boot.
connection.id eth0	NAME=eth0	The name of this connection.
connection.interface-name eth0	DEVICE=eth0	The connection is bound to the network interface with this name.

Nmcli networking off|on 是总开关，控制 NetworkManager 的对网络的管理状态。

Nmcli con down CNAME 是使某一个处于激活状态连接件 CNAME 失效，由于大部分

有线连接的系统中都会默认自动重新连接，所以这个命令执行后，可能又会自动 up 了。使用 Nmcli dev dis IName 可以有效地断开指定的网卡，以防止自动连接。

10.1.1.3使用 nmcli 创建网卡绑定

创建 bond 设备

```
# nmcli connection add type bond con-name mybond0 ifname mybond0 mode active-backup
Connection 'mybond0' (4db6cde4-2757-40b8-b866-7ec931b46e70) successfully added.
```

添加子网卡

```
# nmcli connection add type bond-slave ifname ens39 master mybond0
Connection 'bond-slave-ens39' (b75ce31e-dd12-4b4d-8249-e53e48a3aa6b) successfully added.
# nmcli connection add type bond-slave ifname ens40 master mybond0
Connection 'bond-slave-ens40' (ad9e9479-d20b-4a39-af17-5659e9256f40) successfully added.
```

启动 bond 和对应的子网卡

```
# nmcli connection up bond-slave-ens39
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)
# nmcli connection up bond-slave-ens40
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
# nmcli connection up mybond0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
```

10.1.1.4 使用 nmtui 创建网卡绑定

在命令提示窗口运行 nmtui 命令打开配置界面，选中“Edit a connection”配置连接件，然后选择”Add”添加一个连接件，再选中”Bond”进行网卡绑定的连接件配置，如下图 10-3：

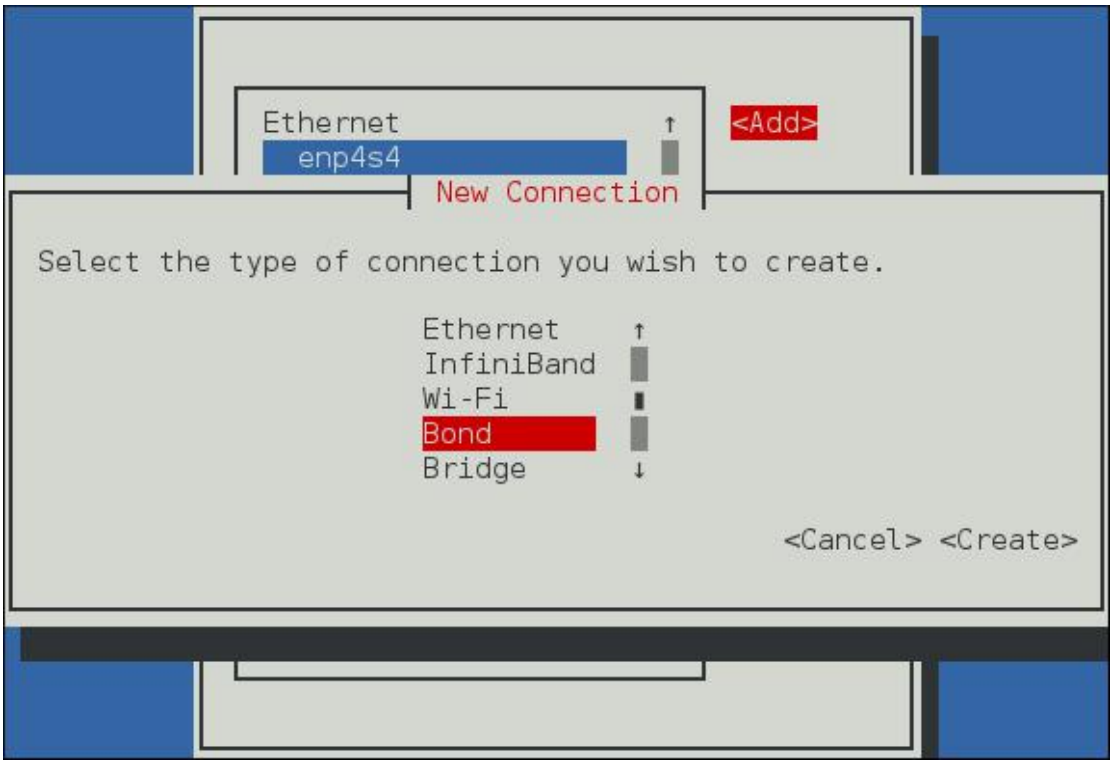


图 10-3

在打开的绑定配置界面中，选择需要绑定的子网卡、相关的网络参数配置以及绑定模式的选择，如下图 10-4：

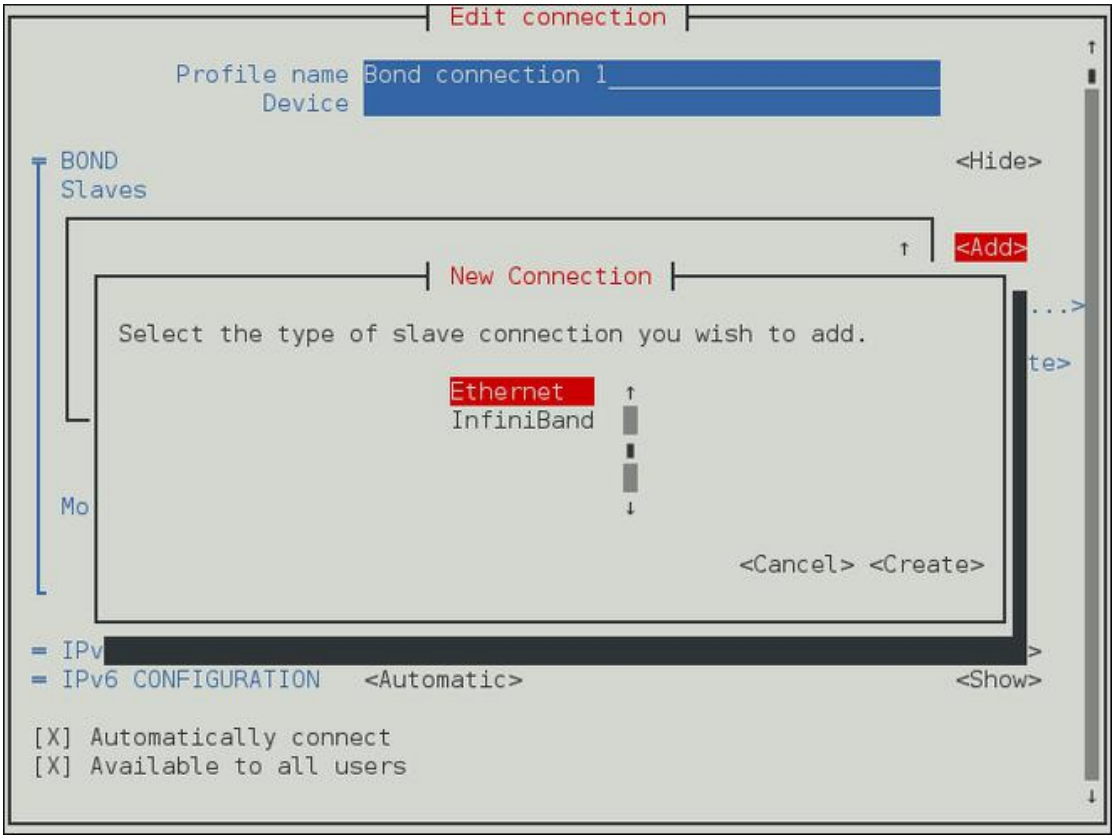


图 10-5

10.1.1.5 解除网卡绑定

1、断开 bond 设备

```
nmcli dev dis bond0
```

2、删除子网卡及 bond 的连接件

```
nmcli con del bond-slave-ens33
```

3、删除 bond 设备名称

```
echo -bond0 > /sys/class/net/bonding_masters
```

10.1.1.6 DNS 配置文件

DNS 配置文件/etc/resolv.conf，示例如下：

```
generated by /sbin/dhclient-script
search gdlc.org
```

```
nameserver 172.16.100.3          #nameserve 表示域名服务器，ip 地址就是 DNS 服务器
nameserver 172.16.100.92
```

10.2 网络常用命令

请参见 1.7 节“基本网络命令”。

10.3 网卡绑定

本节以将一台机器上的 eth3 和 eth4 两块网卡绑定成 trunk0，将 eth5 和 eth6 两块网卡绑定为 trunk1 为例说明网卡绑定操作方法。创建 trunk1 的步骤请参考本文中的注释完成。

1、备份系统原有的网络配置脚本。

稳妥起见，可先将/etc/sysconfig/network-scripts/目录下所有以 ifcfg- 开头的文件备份到一个另外的目录，如：/root/eth-bak 目录。

2、创建绑定。

1）创建绑定配置文件：/etc/sysconfig/network-scripts/ifcfg-trunk0（若创建 trunk1 绑定，则文件名为 ifcfg-trunk1）内容如下：

```
DEVICE=trunk0                #如果创建 trunk1 绑定时，为 trunk1
IPADDR=10.215.32.67          #如果创建 trunk1 绑定时，为 trunk1 绑定的 IP
NETMASK=255.255.240.0        #如果创建 trunk1 绑定时，为 trunk1 绑定的掩码
GATEWAY=10.215.32.65         #如果创建 trunk1 绑定时，为 trunk1 绑定的网关
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
```

2）修改/etc/sysconfig/network-scripts/ifcfg-eth3 为如下内容（如果创建 trunk1 绑定时，则修改的是 ifcfg-eth5 文件）：

```
DEVICE=eth3                  #如果创建 trunk1 绑定时，为 eth5
ONBOOT=yes
HWADDR=xx:xx:xx:xx:xx        #该物理设备的实际 MAC 地址
MASTER=trunk0                #如果创建 trunk1 绑定时，为 trunk1
SLAVE=yes
BOOTPROTO=none
```

```
USERCTL=no
```

3）修改/etc/sysconfig/network-scripts/ifcfg-eth4 为如下内容（如果创建 trunk1 绑定时，则修改的是 ifcfg-eth6）：

```
DEVICE=eth4                    #如果创建 trunk1 绑定时，为 eth6
ONBOOT=yes
HWADDR=xx:xx:xx:xx:xx          #该物理设备的实际 MAC 地址
MASTER=trunk0                  #如果创建 trunk1 绑定时，为 trunk1
SLAVE=yes
BOOTPROTO=none
USERCTL=no
```

4）在/etc/modprobe.d/local.conf 里面添加如下语句：

```
alias trunk0 bonding                #如果创建 trunk1 绑定时，为 trunk1
options trunk0 miimon=100 mode=1 #如果创建 trunk1 绑定时，为 trunk1，mode 的值可以为 1 或者 0，分别代表 active-backup 与 balance-rr 的方式
```

5）用下面的命令重启网络服务，绑定即可生效。

```
# systemctl restart network
```

3、解除绑定的方法

1）删除绑定网卡对应的配置文件

```
#rm -f /etc/sysconfig/network-scripts/ifcfg-trunk0   #如果已经创建 trunk1 时，则删除的是 ifcfg-trunk1
```

2）还原原来的配置文件

```
#cp -f   /root/eth-bak/ifcfg-eth3 /etc/sysconfig/network-scripts/   #如果已经创建 trunk1 时，则拷贝的是 ifcfg-eth5
#cp -f   /root/eth-bak/ifcfg-eth4 /etc/sysconfig/network-scripts/   #如果已经创建 trunk1 时，则拷贝的是 ifcfg-eth6
```

3）删除/etc/modprobe.d/local.conf 里面如下语句：

```
alias trunk0 bonding                #如果已创建 trunk1 绑定时，为 trunk1
options trunk0 miimon=100 mode=1   #如果已创建 trunk1 绑定时，为 trunk1
```

4）按顺序执行下面的命令：

```
#ifconfig eth3 down   #如果已创建 trunk1 绑定时，为 eth5
#echo -eth3 > /sys/class/net/trunk0/bonding/slaves   #如果已创建 trunk1 绑定时，
```

```
为 eth5 和 trunk1
#ifconfig eth4 down  #如果已创建 trunk1 绑定时，为 eth6
#echo -eth4 > /sys/class/net/trunk0/bonding/slaves  #如果已创建 trunk1 绑定时，为 eth6 和 trunk1
#ifconfig trunk0 down  #如果已创建 trunk1 绑定时，为 trunk1
#echo -trunk0 > /sys/class/net/bonding_masters #如果已创建 trunk1 绑定时，为 trunk1
```

5）重启网络服务，绑定即可解除成功。

```
# systemctl restart network
```

10.4 Network teaming 服务

Network teaming 服务将多个物理链路聚合成一个逻辑链路，以提供更高的吞量和冗余。以前的“网卡绑定”、”通道聚合“、“端口绑定”、“网卡均衡负载”和“链路聚合”等概念将被 teamd 服务统一实行管理。使用 teamd 服务，不会对已经存在的网卡绑定造成影响，是 CGSLV6 之后做网卡绑定的一种可选方式。该服务依赖于 NetworkManager。

10.4.1 使用 nmcli 工具创建一个网卡绑定

10.4.1.1 创建 team 设备名称

命令格式：

```
nmcli con add type team con-name CNAME ifname INAME [config JSON]
```

CNAME 是 connection 的名字，INAME 是网卡名称。JSON 部分按这样格式书写：

{" runner" :{"name" : "METHOD"}}

METHOD 可以是：**broadcast, roundrobin, activebackup, loadbalance**, or **lacp**.

例如添加一个名字为 team0 的 bond 网口：

```
# nmcli con add type team con-name team0 ifname team0 config ' {" runner" : {"name" : " activebackup "}} '
Connection 'team0' (c3609f3f-3746-4c47-a52a-97c2009d55aa) successfully added.
```

10.4.1.2 分配 IP 地址

命令格式：

```
nmcli con mod team0 ipv4.addresses 172.168.17.120/24
```

例如：

```
# nmcli con mod team0 ipv4.addresses 172.168.17.120/24
```

10.4.1.3 分配接口

命令格式：

```
nmcli con add type team-slave con-name CNAME ifname INAME master TEAM
```

说明：CNAME 是 connection 配置名，INAME 是设备名，TEAM 是 bond 网口的名字。

例 1：

```
# nmcli con add type team-slave ifname ens37 master team0
Connection 'team-slave-ens37' (5c8cc7de-6334-4ba1-89f4-a84bc3d35938) successfully added.
```

例 2：

```
# nmcli con add type team-slave con-name team0-slave2-ens38 ifname ens38 master team0
Connection 'team0-slave2-ens38' (5c6944cc-0d50-4ccf-9826-46042d2b58e9) successfully added.
```

以上两条命令将会创建两个新的 connection，以前对应 ens37 和 ens38 的配置(connection)默认会保留。（建议删除，否则下次故障恢复可能导致原来的配置文件生效，而不是 bond 配置生效）

```
[root@localhost ~]# nmcli connection show
```

10.4.1.4 启动

先断开所有子网卡以及 bond 网口，再启动所有子网口。

nmcli dev dis INAME　　断开子网卡设备

nmcli con up CNAME　　启动两个子网卡对应 team 设备的连接件

跟启动 bond 设备类似，启动 team 设备时，也有以下特点

1、启动 team 网口不会自动启动对应的子网口

2、启动其中的一个子网口，会启动 team 网口

3、停用 team 网口，会自动停止子网口

4、没有配置子网口的 team 网口，可以配置静态 IP 后启动

5、如果没有配置子网口的 team 网口，如果使用了 DHCP 方式获取 IP，启动后将一直等待子网口完成 IP 获取。

6、使用 DHCP 获取 IP 的 team 网口，当子网口插上网线后，会一直等待它完成 IP 获取。

7、即便子网口没有接上网线，使用 DHCP 方式获取 IP 的 team 网口将持续等待子网口完成 IP 获取。

10.4.2 使用 nmtui 工具创建网卡绑定

在命令提示窗口运行 nmtui 命令打开配置界面，选中“Edit a connection”配置连接件，然后选择”Add”添加一个连接件，再选中”Team”进行网卡绑定的连接件配置，如图 10-4：

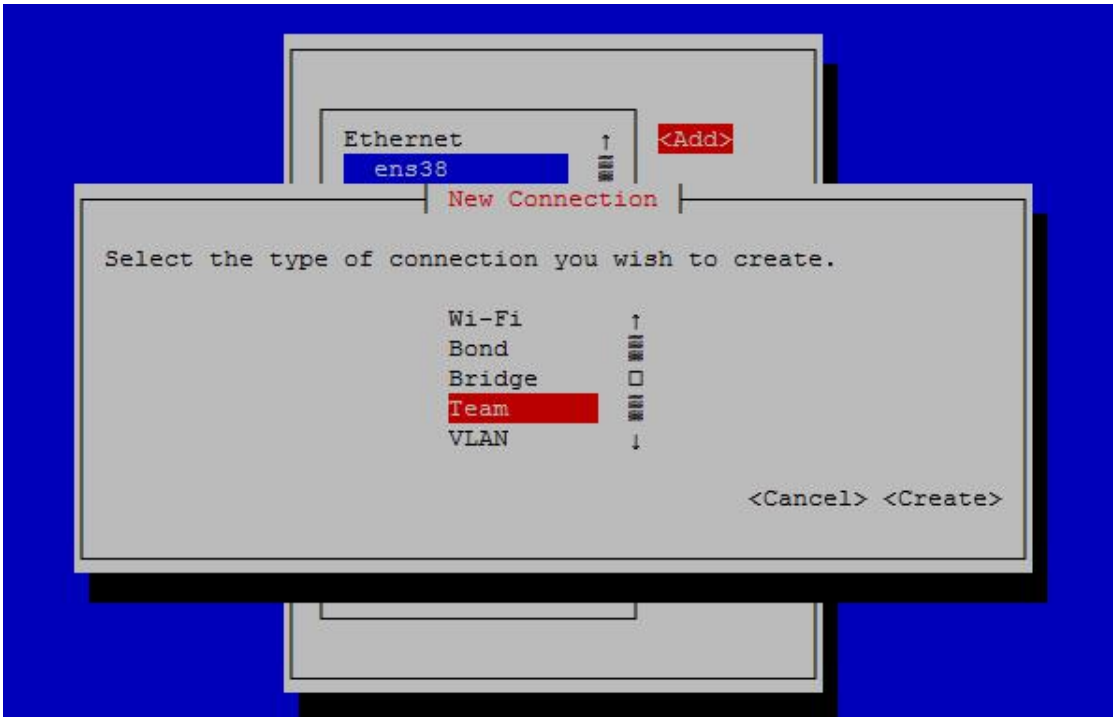


图 10-4

在打开的绑定配置界面中，选择需要绑定的子网卡、相关的网络参数配置以及对应的

JSON 格式的配置内容，如下图 10-5：

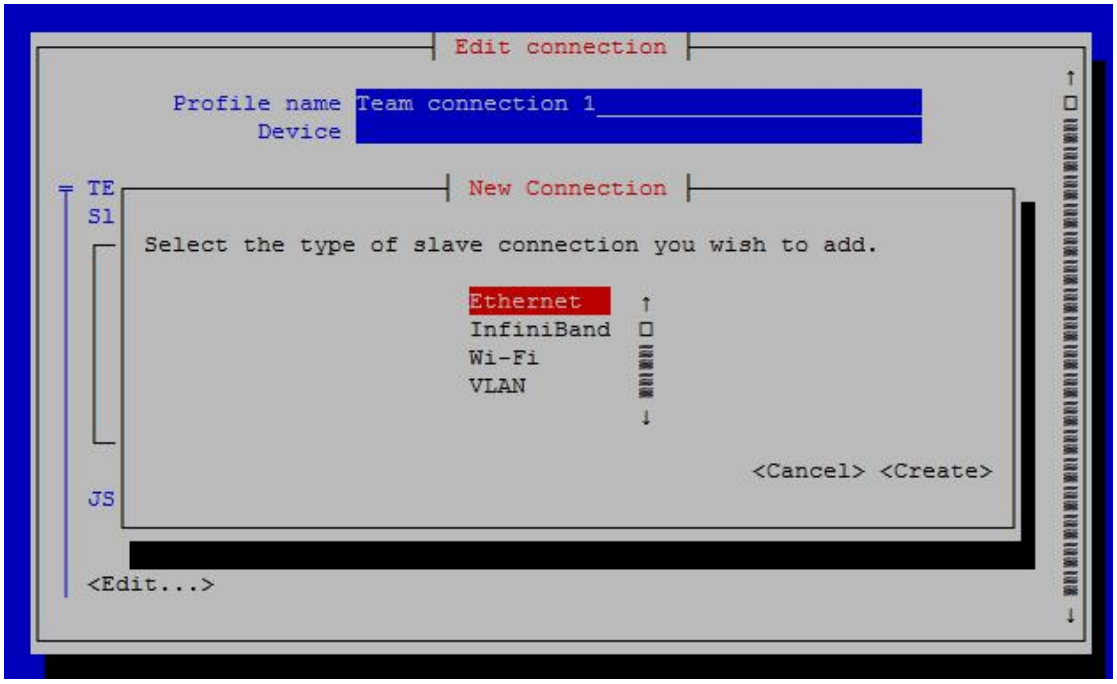


图 10-5

10.4.2.1 管理 team 设备

查看 team0 设备的状态

```
# teamdctl team0 state view
```

查看更多信息需加-v 参数

```
# teamdctl team0 state view –v
```

以 JSON 格式查看状态

```
# teamdctl team0 state dump
# teamdctl team0 config dump
```

添加子网 em1 口到 team0

```
# teamdctl team0 port add em1
```

从 team0 中删除子网口 em1

```
# teamdctl team0 port remove em1
```

以 JSON 格式修改子网口 em1 的配置

```
# teamdctl team0 port config update em1 JSON-config-string
```

10.4.2.2解绑 team 设备

断开 team0 设备

```
nmcli device disconnect team0
```

删除子网卡的连接件

```
nmcli connection delete team-slave-ens37
nmcli connection delete team-slave-ens38
```

刷新连接件

```
nmcli connection reload
```

第 11 章

图形环境

CGSL V6 系统默认使用了 Wayland 系统，为用户提供图形化环境。本章主要介绍图形环境管理相关的基本配置和操作。

11.1 VNC

VNC 是一种常用的远程桌面控制工具，本节介绍了 VNC 服务端的安装和配置。

11.1.1 VNC 安装

11.1.1.1 安装 xorg-x11-fonts-misc

执行如下命令确认系统是否已经安装了 xorg-x11-fonts-misc 软件包。

```
#rpm -qa|grep xorg-x11-fonts-misc
```

如果没有安装，则从 CGSL 安装光盘中找到以下的 rpm 包，使用下面的命令进行安装。

```
#rpm -ivh xorg-x11-fonts-misc-7.5-19.el8.noarch
```

11.1.1.2 安装 vnc 服务端

执行如下命令确认系统是否已经安装了 tigervnc-server 软件包。

```
#rpm -qa|grep tigervnc-server
```

如果没有安装，则从 CGSL 安装光盘中找到以下的 rpm 包，使用下面的命令进行安装（以 32 位为例）：

```
#rpm -ivh tigervnc-server-1.9.0-12.el8_1.x86_64
```

11.1.2 VNC 配置

复制配置文件模板：

```
#cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:桌面号.service
```

桌面号为自定义，例如 vncserver@:1.service,下面使用 1 为桌面号。

编辑/etc/system/system/vncserver@:1.service 里面的参数，配置后如下所示：

如果使用 root 用户连接，则需要设置[Service]段中的<USER>为 root。正确配置后如下所示：

```
ExecStart=/sbin/runuser -l root -c "/usr/bin/vncserver %i"
PIDFile=/root/.vnc/%H%i.pid
```

刷新 systemctl 配置。

```
#systemctl daemon-reload
```

配置 vnc 登录密码。

```
#vncpasswd
```

运行 VNC 服务与添加开机启动(需要使用 root 用户)。

```
#systemctl start vncserver@:1.service
# systemctl enable vncserver@:1.service
```

如果系统启用了防火墙 firewalld，则需要添加开放 vncserver 网络端口。

```
# firewall-cmd –permanent –add-service=vnc-server
# firewall-cmd –reload
```

11.2 XManager

XManager 是一种常用的远程桌面控制工具，本节主要介绍使用 XManager 工具需要的相关基本配置。

11.2.1 XManager 服务端配置

CGSL V6 版本默认不能使用 XManager 工具远控，通过修改/etc/gdm/custom.conf 来启用：

1. [xdmcp]段的 Enable 选项，需要设置 Enable 为 1 或 true。配置后如下所示：

```
[xdmcp]
Enable=1
```

2. 如果需要允许 root 用户连接，则需要设置[security]段中的 AllowRomoteRoot 为 1 或 true。正确配置后如下所示。

```
[security]
AllowRemoteRoot=1
```

3. 修改配置后需要执行下面的命令重启图形环境，请注意当前有无运行在图形界面下的程序，以免受到图形环境重启的影响。

```
# systemctl restart gdm.service
```

11.2.2 注意事项

GNOME 3 使用了 OpenGL，需要直接访问图形硬件。而 Xorg 使用 DR（Direct Rendering Interface）作为默认渲染技术，将无法运行 Xserver 上面。所以 Window 上面使用的 XManager 客户端不能用于连接 XDMCP。

11.2.3 在 xinetd 上配置 VNC 与 XDCMP

1. 安装 thgervnc-server，xorg-x11-fonts-Type1，xinetd 包：

```
#rpm –ivh thgervnc-server xorg-x11-fonts-Type1 xinetd
```

2. 创建 xinetd 服务

使修改/etc/xinetd.d/vncserver 文件（若不存在此文件就创建），添加如下内容：

```
service vncserver
{
disable = no
socket_type = stream
protocol = tcp
group = tty
wait = no
user = nobody
server = /usr/bin/Xvnc
```

```
server_args = -inetd -query localhost -geometry 1024×768 -depth 16 -once -fp
/usr/share/X11/fonts/misc -securitytypes=none
}
```

3.添加 vnc 服务：

在/etc/services 文件中最后加入

```
vncserver        5900/tcp                    # VNC and GDM
```

4. 重启 xinetd 服务：

```
#systemctl restart xinetd.service
```

5. 在 Windows 上通过 vnc 客户端连接：

连接地址是“IP：5900”，连接后画面下

第 12 章

COPYRIGHT NOTICE AND WARRANTY DISCLAIMER

ZTE NewStart CORPORATION CGSL series products are released under the GNU General Public License (Version2) that comes together with this product and can also be found under http://www.gnu.org/licenses/gpl.html.

COPYRIGHT NOTICE

The majority of programs in CGSL falls under the GNU General Public License (GPL). The license agreement for software component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (subject to certain obligations in some cases) .The distribution and use of software (even if it is free software) must honor certain license conditions. Not all programs in CGSL are free software. Some of them are shareware, restricted to noncommercial use, or may have other restrictive conditions.

The package information mentions the respective license and authors. We cannot, however, ensure the correctness of this information. In cases of doubt, refer to the original copyright information of the respective programs.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The Free Software Foundation has exempted Bash from the requirement ofParagraph 2c of the General Public License. This is to say, there is no requirement for Bash to print a notice when it is started interactively in the usual way. We made this exception because users and standards expect shells not to print such messages. This exception applies to any program that serves as a shell and that is based primarily on Bash as opposed to other GNU software.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent

must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

(Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program,and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the

operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of

protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.

SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

WRITTEN OFFER

If you would like a copy of the GPL source code contained in this product shipped to you on CD, for a charge which is no more than the cost of preparing and mailing a CD to you, please contact os@gd-linux.com.