



第 3 章

openEuler 文件系统

对任意一个通用的操作系统而言，文件目录管理是其必不可少的一项功能，openEuler 操作系统提供了十分强大的文件与目录管理功能，本章主要介绍文件系统的概念、文件管理常用命令及配置文件权限相关知识。

3.1 openEuler 文件概念及特点



视频讲解

3.1.1 openEuler 文件简介

在 openEuler 中，文件是存储在文件系统中的数据集合，它以一种有序的方式组织和存储数据。文件可以包含各种类型的内容，如文本、二进制代码、图像、音频、视频等。文件是用户和程序访问数据的基本单位。openEuler 支持多种类型的文件，每种文件类型都有其特定的用途和访问方式。

1. 普通文件

普通文件 (Regular Files) 是文件系统中最常见的文件类型，用于存储数据或程序代码。普通文件可以进一步细分为如下 3 类。

(1) 文本文件 (Text Files): 存储可读的文本内容，例如文档、脚本、配置文件等。文本文件的内容可以用文本编辑器 (如 Vi、nano) 查看和编辑。

(2) 二进制文件 (Binary Files): 存储可执行程序或机器代码，通常由编译器生成。用户可以直接运行二进制文件来执行程序。

(3) 数据文件 (Data Files): 存储应用程序生成的数据，例如数据库文件、日志文件等。

2. 目录文件

目录文件 (Directory Files) 是一种特殊的文件，用于组织其他文件和目录。目录以树形结构组织文件系统，从根目录 (/) 开始，向下分支。目录文件存储了子目录和文件的名称、权限、位置等元数据。

3. 设备文件

设备文件 (Device Files) 用于表示系统中的硬件设备，它们允许用户和程序访问硬件

资源。设备文件分为如下两类。

（1）字符设备文件（Character Device Files）：以字符流的方式访问设备，例如键盘、鼠标、串行端口等。字符设备文件的 I/O 操作是逐个字符进行的。

（2）块设备文件（Block Device Files）：以块的方式访问设备，如硬盘、固态硬盘等。块设备文件的 I/O 操作是以固定大小的块进行的，通常用于存储设备。

4. 符号链接

符号链接（Symbolic Links）类似于 Windows 中的快捷方式，是一个指向其他文件或目录的引用。符号链接可以跨越不同的文件系统，并且可以指向文件或目录。符号链接本身是一个独立的文件，存储了目标文件或目录的路径。

5. 硬链接

硬链接（Hard Links）是文件系统中对同一个文件的多个名称的引用。硬链接直接指向文件的 inode（索引节点），而不是文件路径。硬链接不能跨越文件系统，也不能指向目录。

6. 命名管道

命名管道（Named Pipes）是一种进程间通信机制，允许不相关的进程通过管道进行数据交换。

3.1.2 openEuler 文件系统的目录结构

在 openEuler 系统中，目录结构遵循 Linux 文件系统层次化标准（Filesystem Hierarchy Standard, FHS），这是一种广泛应用于 Linux 发行版的标准，定义了系统中各个目录的用途和内容。与 Windows 系统的文件结构不同，openEuler 路径中不使用驱动器盘符，而是将整个文件系统表示为树状结构，文件结构的开始处只有一个单独的顶级目录结构，该目录叫作根目录。该操作系统中一切都从“根”开始。以下是构成 openEuler 目录的主要内容。

- /：根目录，整个文件系统的最顶层目录，所有其他目录和文件都从这里开始。
- /bin：存放系统中最常用的命令和可执行文件，如 ls、cp、mv 等。
- /boot：存放启动 openEuler 时使用的核心文件，如内核文件、引导程序文件等。
- /dev：存放设备文件，包括字符设备文件和块设备文件，用于访问硬件设备。
- /etc：存放系统的配置文件和子目录，如网络配置文件、用户账户信息文件等。
- /lib：存放系统最基本的动态库文件。
- /media：自动挂载外部存储设备（如 U 盘、光驱等）的目录。
- /mnt：用于临时挂载文件系统的目录。
- /opt：存放可选的软件包和应用程序。
- /proc：虚拟文件系统，存放有关系统进程和内核信息的文件。
- /root：超级用户（root 用户）的主目录。
- /sbin：存放系统管理员使用的系统命令，如 fdisk、mount、umount 等。
- /srv：存放服务数据，如 Web 服务器的数据文件。
- /sys：虚拟文件系统，用于访问和设置内核对象。
- /tmp：存放临时文件，在运行过程中可以将临时数据存储在在这个目录下。
- /usr：存放用户程序、库文件、文档等共享的只读数据。

- `/var`: 存放系统运行时产生的可变数据文件, 如日志文件、邮件文件等。
- `/home`: 普通用户的主目录, 每个用户在登录系统后, 都会被分配一个专属的主目录, 用于存放个人文件和数据。
- `/run`: 存放系统运行时产生的临时数据, 类似于 `/tmp`, 但重启后数据不会清空。
- `/system`: 在某些嵌入式或特定配置中, 用于存放系统特定的文件和服务, 该目录并非 openEuler 官方标准目录, 仅在特定定制版出现。

3.1.3 绝对路径和相对路径

在 openEuler 操作系统中, 路径是用于定位文件和目录的一种方式。路径可以分为绝对路径 (Absolute Path) 和相对路径 (Relative Path), 它们在表示文件或目录位置时有不同的特点和用途。

1. 绝对路径

绝对路径是从文件系统的根目录开始的完整路径, 它明确地指定了文件或目录在整个文件系统中的位置。绝对路径是唯一的, 不会因当前工作目录的变化而改变。

绝对路径具有如下特点。

- (1) 唯一性: 绝对路径唯一地标识了文件或目录的位置。
- (2) 独立性: 不依赖于当前工作目录, 无论当前工作目录在哪里, 绝对路径始终指向同一个位置。

2. 相对路径

相对路径是从当前工作目录开始的路径, 描述了从当前目录到目标文件或目录的路径。相对路径依赖于当前工作目录, 因此相同的相对路径在不同的工作目录下可能指向不同的文件或目录。

相对路径具有如下特点。

- (1) 依赖性: 依赖于当前工作目录, 路径的解析从当前工作目录开始。
- (2) 灵活性: 在某些情况下, 相对路径更简洁, 尤其适用于处理当前目录或子目录中的文件。

3. 特殊符号

在路径中, 还有一些特殊符号用于表示特定的目录。

- (1) `“.”`: 表示当前工作目录。
- (2) `“..”`: 表示当前目录的上一级目录 (父目录)。
- (3) `“~”` (用户主目录): 表示当前用户的主目录, 例如 `/home/userexample`。

3.2 openEuler 文件管理常用命令

3.2.1 文件查看命令

1. 查看文件目录

1) `cd`

`cd` (Change Directory, 变换目录) 命令用于切换到指定目录。



视频讲解

该命令的语法如下。

```
cd [目录]
```

说明：“cd -”命令会切换到上一次的工作目录，并且打印当前切换到的目录路径，cd 目录可以与特殊符号一起使用，如“cd..”表示切换回上一级目录，cd 命令不带参数时，表示切换到用户主目录。

2) pwd

pwd（Print Working Directory，打印工作目录）命令用于打印当前工作目录的完整路径。

说明：pwd 命令的输出结果是一个绝对路径，从根目录开始。一般情况下，pwd 命令执行时不带任何参数，如果当前目录是通过符号链接进入的，可以使用 pwd -P 显示实际的目录路径。

3) ls

ls（List，列出目录）命令用于显示指定目录中的文件和子目录的名称、权限、所有者、大小、修改时间等信息。如果不指定目录，默认显示当前工作目录的内容。

该命令的语法如下。

```
ls [选项] [路径]
```

ls 命令常用选项如表 3-1 所示。

表 3-1 ls 命令常用选项

选 项	说 明
-a	显示所有文件，包括隐藏文件（以“.”开头的文件）
-l	显示文件信息，包括权限、所有者、大小、修改时间等
-h	显示文件大小（常与“-l”一起使用）
-R	递归显示目录下的所有子目录和文件
-t	按修改时间排序，最近修改的文件或目录显示在前面
-S	按文件大小排序，最大的文件或目录显示在前面

多个选项可以联合使用，以便更加直观地查看文件，如图 3-1 所示。

2. 查看文件内容

1) less

less 命令是一个功能强大的文本查看工具，常用于在终端查看文本文件的内容，支持用户在查看文件时向前或向后滚动、搜索特定内容等。

该命令的语法如下。

```
less [选项] 文件名
```

less 命令常用选项如表 3-2 所示。

```
[root@openEuler ~]# ls -al
总计 80
dr-xr-xr-x. 20 root root 4096 1月25日 11:41 .
dr-xr-xr-x. 20 root root 4096 1月25日 11:41 ..
dr-xr-xr-x. 2 root root 4096 11月19日 22:13 afs
lrwxrwxrwx. 1 root root 7 11月19日 22:13 bin -> usr/bin
dr-xr-xr-x. 7 root root 4096 1月16日 21:28 boot
drwxr-xr-x. 19 root root 3420 2月 6日 14:27 dev
drwxr-xr-x. 135 root root 12288 2月 6日 14:27 etc
drwxr-xr-x. 11 root root 4096 2月 1日 13:37 home
lrwxrwxrwx. 1 root root 7 11月19日 22:13 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 11月19日 22:13 lib64 -> usr/lib64
drwx----- 2 root root 16384 1月16日 20:29 lost+found
drwxr-xr-x. 13 root root 4096 2月 1日 13:37 media
drwxr-xr-x. 2 root root 4096 11月19日 22:13 mnt
drwxr-xr-x. 4 root root 4096 1月25日 11:41 openEuler
drwxr-xr-x. 3 root root 4096 1月16日 20:35 opt
dr-xr-xr-x. 307 root root 0 2月 6日 14:27 proc
dr-xr-xr-x. 17 root root 4096 2月 6日 14:27 root
drwxr-xr-x. 49 root root 1360 2月 6日 14:27 run
lrwxrwxrwx. 1 root root 8 11月19日 22:13 sbin -> usr/sbin
drwxr-xr-x. 2 root root 4096 11月19日 22:13 srv
dr-xr-xr-x. 13 root root 0 2月 6日 14:27 sys
drwxrwxrwt. 13 root root 540 2月 6日 14:28 tmp
drwxr-xr-x. 12 root root 4096 1月16日 20:30 usr
drwxr-xr-x. 22 root root 4096 1月16日 20:47 var
[root@openEuler ~]#
```

图 3-1 查看文件

表 3-2 less命令常用选项

选 项	说 明
-g	只高亮显示最后一个搜索匹配项，而不是所有匹配项
-i	在搜索时忽略大小写
-N	显示每行的行号
-S	不折叠长行，即长行会自动换行显示
-w	在搜索时高亮显示匹配项

less 命令运行时，可以通过表 3-3 所示的快捷键进行操作。

表 3-3 less命令快捷键

快 捷 键	说 明
空格	向下滚动一页
b	向上滚动一页
Enter	向下滚动一行
k	向上滚动一行
/	向下搜索指定字符串
?	向上搜索指定字符串
n	查找下一个匹配项（在搜索后使用）
N	查找上一个匹配项（在搜索后使用）
G	跳转到文件末尾
g	跳转到文件开头
:	输入命令（如q退出、n跳转到指定行号等）
q	退出

2) more

more 命令同样是一个简单的文本查看工具，用于在终端中逐页显示文本文件的内容。它比 less 命令出现得更早，功能相对简单，主要用于逐页显示文件内容，但不支持像 less 那样灵活的交互操作。

该命令的语法如下。

```
more [选项] [文件名]
```

more 命令常用选项如表 3-4 所示。

表 3-4 more命令常用选项

选 项	说 明
-d	在显示提示信息时，提供更详细的说明
-l	忽略文件中的换页符，直接连续显示文件内容
-n	指定每页显示的行数，默认值是终端窗口的行数
-p	在显示下一页时，不重复显示上一页的最后一行，避免内容重复
-s	将连续的多个空行压缩为一行显示
-u	禁用文件中的下画线显示

more 命令运行时，同样可以通过快捷键进行操作，如表 3-5 所示。

表 3-5 more命令快捷键

快 捷 键	说 明
空格	向下滚动一页
Enter	向下滚动一行
q	退出more
/	向下搜索指定的字符串
n	查找下一个匹配项（在搜索后使用）

3) cat

cat（Concatenate）命令是最基本且常用的文本处理工具之一，主要用于查看、合并和创建文本文件。

该命令的语法如下。

```
cat [选项] [文件名]
```

用 cat 命令查看文件内容，执行结果如图 3-2 所示。

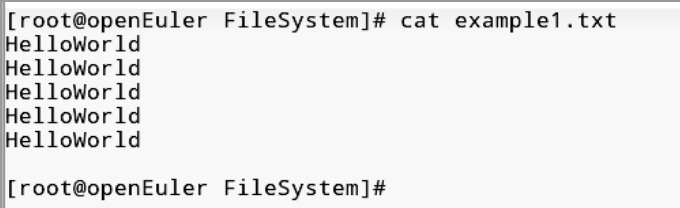


图 3-2 用 cat 命令查看文件内容

用 cat 命令合并文件内容，执行结果如图 3-3 所示。

```
[root@openEuler FileSystem]# cat example1.txt
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld

[root@openEuler FileSystem]# cat example2.txt
openEuler
openEuler
openEuler
openEuler
openEuler
openEuler
[root@openEuler FileSystem]# cat example1.txt example2.txt
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld

openEuler
openEuler
openEuler
openEuler
openEuler
openEuler
[root@openEuler FileSystem]# █
```

图 3-3 用 cat 命令合并文件内容

用 cat 命令创建文件，执行结果如图 3-4 所示。

```
[root@openEuler FileSystem]# cat > new.txt
This is a new file.
[root@openEuler FileSystem]# ls
example1.txt example2.txt new.txt
[root@openEuler FileSystem]# cat new.txt
This is a new file.
[root@openEuler FileSystem]#
```

图 3-4 用 cat 命令创建文件

输入内容后，按 Ctrl + C 组合键结束输入，将输入的内容保存在 new.txt 文件中。

说明：重定向知识将在后续章节中介绍。

cat 命令常用选项如表 3-6 所示。

表 3-6 cat命令常用选项

选 项	说 明
-n	在输出的每一行前面显示行号
-b	只对非空行编号（与“-n”结合使用）
-s	压缩连续的多个空行，只显示一个空行

4) tac

tac 命令用于逆序显示文件内容，其名称源于“cat”的反向拼写。它与 cat 命令功能相

反，cat 是从文件的第一行开始输出，而 tac 则是从文件的最后一行开始逐行向上输出。
tac 命令的语法如下。

```
tac [选项] [文件名]
```

5) head

head 命令用于查看文件开头的部分内容。用户可以通过参数选项自定义显示的行数，在不指定行数的情况下默认会显示文件的前 10 行，当文件内容较多时，它可以帮助用户快速了解文件的结构和内容。

该命令的语法如下。

```
head [选项] [文件名]
```

head 命令常用选项如表 3-7 所示。

表 3-7 head命令常用选项

选 项	说 明
-n	指定显示的行数，默认显示10行
-c	指定显示的字节数
-q	在处理多个文件时，不显示文件名
-v	在处理多个文件时，始终显示文件名

示例：显示文件前 3 行，执行结果如图 3-5 所示。

```
[root@openEuler FileSystem]# cat example1.txt
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld

[root@openEuler FileSystem]# head -n 3 example1.txt
HelloWorld
HelloWorld
HelloWorld
[root@openEuler FileSystem]#
```

图 3-5 显示文件前 3 行

显示前 10 个字符，执行结果如图 3-6 所示。

```
[root@openEuler FileSystem]# head -c 20 example1.txt
HelloWorld
HelloWorld
HelloWorld
[root@openEuler FileSystem]#
```

图 3-6 显示前 10 个字符

6) tail

与 head 命令相反，tail 命令用于查看文件的尾部内容，默认情况下，tail 命令会显示文件的最后 10 行内容。

tail 命令的语法如下。

```
tail [选项] [文件名]
```


tail 命令常用选项如表 3-8 所示。

表 3-8 tail命令常用选项

选 项	说 明
-n	指定显示的行数
-c	指定显示的字节数
-f	循环读取文件的新增内容
-q	在处理多个文件时，不显示文件名
-v	在处理多个文件时，始终显示文件名

示例：显示文件最后 3 行，执行结果如图 3-7 所示。

```
[root@openEuler FileSystem]# tail -n 3 example1.txt
HelloWorld

This is a example file.
[root@openEuler FileSystem]#
```

图 3-7 显示文件最后 3 行

说明：如果文件为空，tail 命令不会输出任何内容；实时跟踪文件时，可以通过按 Ctrl+C 组合键中断跟踪。

3. 查看文件类型

file 命令用于识别文件的真实类型，它通过分析文件的内容和特征来判断，而不是仅仅依赖文件的扩展名进行判断。所有用户均有权限使用该命令。

该命令的语法如下。

```
file [文件名]
```

file 命令常用选项如表 3-9 所示。

表 3-9 file命令常用选项

选 项	说 明
-b或--brief	以简洁的形式显示输出，不显示文件名
-L或--dereference	对符号链接进行解引用，显示链接指向的文件类型
-z或--uncompress	尝试解压缩文件并分析其内容
-s或--special-files	对特殊文件（如设备文件）也进行内容检查

查看文件类型，执行结果如图 3-8 所示。

```
[root@openEuler FileSystem]# file example1.txt
example1.txt: ASCII text
[root@openEuler FileSystem]# file /etc/passwd
/etc/passwd: ASCII text
[root@openEuler FileSystem]#
```

图 3-8 file 命令查看文件类型



视频讲解

3.2.2 目录操作命令

1. touch

touch 命令用于创建空文件或更新文件时间戳。

该命令的语法如下。

```
touch [-d 日期] [选项] [文件名]
```

说明：如果文件不存在，则创建一个空文件；如果文件已存在，则更新其时间戳。

2. mkdir

mkdir（Make Directory）命令用于创建指定名称的目录（文件夹），要求创建目录的用户对当前文件夹有写权限。

该命令的语法如下。

```
mkdir [选项] [文件名]
```

mkdir 命令常用选项如表 3-10 所示。

表 3-10 mkdir命令常用选项

选 项	说 明
-p	递归创建多级目录。如果父目录不存在，会自动创建所有必要的父目录
-m	设置目录的权限
-v	在创建目录时显示详细信息

3. rmdir

rmdir（Remove Directory）命令用于删除目录，但需要注意的是，它只能删除空目录，如果目录中包含文件或子目录，则无法直接删除。

该命令的语法如下。

```
rmdir [选项] [文件名]
```

4. mv

mv（Move）命令用于移动文件或目录，同时也可用于重命名文件或目录。

该命令的语法如下。

```
mv [选项] [源文件或目录] [目标文件或目录]
```

说明：如果目标文件或目录与源文件或目录相同，则进行的操作是重命名；进行操作时需要确保用户对文件有足够的权限。

5. rm

rm（Remove）命令用于删除文件和目录，该命令不仅可以删除指定的文件或目录，也可以递归删除某个目录及其中所有子目录和文件。

该命令的语法如下。

rm [选项] [文件或目录]

rm 命令常用选项如表 3-11 所示。

表 3-11 rm命令常用选项

选 项	说 明
-f	强制删除文件或目录，不提示用户确认，即使文件不存在也不会报错
-i	在删除文件或目录之前提示用户确认
-r	递归删除目录及其内容，包括所有子目录和文件

6. cp

cp（Copy）命令用于将指定文件或目录复制到指定路径。

该命令的语法如下。

cp [选项] [源文件或目录] [目标文件或目录]

3.2.3 文件搜索命令

1. locate

locate 命令用于查找文件或目录。

该命令的语法如下。

locate [文件或目录名]

说明：locate 命令并不是直接在文件系统中逐个查找文件，而是依赖于一个预先构建的文件名数据库（/var/lib/mlocate/mlocate.db）。这个数据库包含系统中所有文件的路径信息。当运行 locate 命令时，它会在这个数据库中快速搜索匹配的文件路径，这个数据库通常由 updatedb 命令生成。updatedb 会扫描文件系统中的所有文件和目录，并将它们的路径信息存储到数据库中。该数据库由系统自动创建并且每天自动更新一次，如果需要立即更新数据库（例如在添加或删除大量文件后），可以手动运行 sudo updatedb 命令。

2. find

find 命令是一个非常强大的文件查找命令，与 locate 命令不同，find 命令是直接在文件系统中搜索文件，因此不依赖任何数据库。find 命令可以基于多种条件（如文件名、文件类型、文件大小、修改时间等）来查找文件。

该命令的语法如下。

find [路径] [选项] [表达式]

find 命令常用选项如表 3-12 所示。

表 3-12 find命令常用选项

选 项	说 明
-name	按文件名查找，支持通配符（*、?、[]）
-size	按文件大小查找
-user	按文件所有者查找



视频讲解

续表

选 项	说 明
-group	按文件所属组查找
-type	按文件类型查找
-iname	按文件名查找，不区分大小写

示例：在当前目录下查找后缀名为“.txt”的文件，执行结果如图 3-9 所示。

```
[root@openEuler FileSystem]# ls
example1.txt example2.txt new.txt
[root@openEuler FileSystem]# find . -name "*.txt"
./new.txt
./example2.txt
./example1.txt
[root@openEuler FileSystem]#
```

图 3-9 在当前目录下查找后缀名为“.txt”的文件

示例：查找/home 目录中所有属于用户 euler 的文件，如图 3-10 所示。

```
[root@openEuler FileSystem]# find /home/ -user euler | more -n 10
/home/euler
/home/euler/.themes
/home/euler/.bash_history
/home/euler/.bash_profile
/home/euler/.bash_logout
/home/euler/.bashrc
/home/euler/.config
/home/euler/.config/deepin
/home/euler/.config/deepin/deepin-terminal
/home/euler/.config/deepin/deepin-terminal/wininfo-config.conf
--更多--
```

图 3-10 查找/home 目录中所有属于用户 euler 的文件

3. type

type 命令用于显示命令的类型和来源。可以帮助用户了解某个命令是内置命令、可执行文件、别名还是函数等。
该命令的语法如下。

```
type [选项] 命令名
```

type 命令常用选项如表 3-13 所示。

表 3-13 type命令常用选项

选 项	说 明
-a	显示所有匹配的命令，包括别名、函数和可执行文件
-p	只显示可执行文件的路径
-t	只显示命令的类型
-f	强制显示命令的类型

示例：显示所有匹配的 ls 命令，执行结果如图 3-11 所示。

```
[root@openEuler FileSystem]# type -a ls
ls 是 "ls --color=auto" 的别名
ls 是 /usr/bin/ls
ls 是 /bin/ls
[root@openEuler FileSystem]#
```

图 3-11 显示所有匹配的 ls 命令

4. which

which 命令用于查找可执行文件或脚本的绝对路径。它通过搜索在系统环境变量 PATH 中定义的目录，来定位指定命令的完整路径。which 命令通常用于确认某个命令是否可用，以及它的具体位置。

该命令的语法如下。

which 命令名

示例：查找所有匹配的 ls 命令路径，执行结果如图 3-12 所示。

```
[root@openEuler FileSystem]# which -a ls
/usr/bin/ls
/bin/ls
[root@openEuler FileSystem]#
```

图 3-12 查找所有匹配的 ls 命令路径

5. whereis

whereis 命令同样用于查找文件的路径。与 which 命令不同，whereis 不仅可以查找可执行文件，还可以查找相关的帮助文档、源代码文件等。

该命令的语法如下。

whereis [选项] 文件名

3.2.4 文件操作命令

1. grep

grep（Global Regular Expression Print）命令用于在文件或输入流中搜索符合特定模式的文本行。

该命令的语法如下。

grep [选项] [模式] [文件名]

说明如下。

模式：指定要搜索的文本或正则表达式。

文件名：指定要搜索的文件。如果不指定文件名，默认从标准输入读取。

正则表达式相关知识将在后续章节介绍。

grep 命令常用选项如表 3-14 所示。

表 3-14 grep命令常用选项

选 项	说 明
-i	忽略大小写进行搜索
-n	显示匹配行的行号
-r	在目录中递归搜索
-c	只输出匹配的行数
-l	只显示包含匹配内容的文件名
-w	只匹配整个单词
-v	取反，输出不匹配的行



视频讲解

示例：在 `example1.txt` 中搜索包含 `Hello` 的行，执行结果如图 3-13 所示。

```
[root@openEuler FileSystem]# grep 'Hello' example1.txt
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld
[root@openEuler FileSystem]#
```

图 3-13 在 `example1.txt` 中搜索包含 `Hello` 的行

示例：列出当前目录下所有包含 `Hello` 的文件，如图 3-14 所示。

```
[root@openEuler FileSystem]# grep -r 'Hello' .
./Hello.txt:Hello
./Hello.txt:Hello
./Hello.txt:Hello
./example1.txt:HelloWorld
./example1.txt:HelloWorld
./example1.txt:HelloWorld
./example1.txt:HelloWorld
./example1.txt:HelloWorld
./example1.txt:HelloWorld
[root@openEuler FileSystem]#
```

图 3-14 列出当前目录下所有包含 `Hello` 的文件

2. sed

`sed`（Stream Editor）命令会以流的形式处理文本，可实现文本替换、删除、插入等编辑操作，适用于简单文本处理场景。通过灵活使用编辑命令和选项，`sed` 可以完成各种复杂的文本处理任务。

该命令的语法如下。

```
sed [选项] [编辑命令] [文件名]
```

`sed` 命令常用选项如表 3-15 所示。

表 3-15 sed命令常用选项

选 项	说 明
-e	以指定的脚本来处理输入的文本文件
-f[脚本文件]	以指定的脚本文件来处理输入的文本文件
-n、--quiet或--silent	仅显示处理后的结果



视频讲解

3.2.5 文件和文件权限

在 `openEuler` 操作系统中，“万物皆文件”，每个文件都归属于对应的所有者和所有组，并且规定了文件的所有者、所有组及其他用户对文件的读（`r`）、写（`w`）、执行（`x`）等权限。可以使用“`ls -l`”命令查看文件的权限信息，如图 3-15 所示。

```
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r-- 1 root root 0 2月 6日 15:15 echo
-rw-r--r-- 1 root root 80 2月 6日 14:48 example1.txt
-rw-r--r-- 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r-- 1 root root 19 2月 6日 15:16 grep
-rw-r--r-- 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r-- 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]#
```

图 3-15 查看文件的权限信息

说明：输出结果的开头 10 个字符表示文件权限信息。

(1) 第 1 位：标识文件类型（可选类型包括 d、-、l、c、b）。

- d 表示目录。
- - 表示普通文件。
- l：表示软链接文件。
- c 表示字符设备文件，如鼠标、键盘等。
- b 表示块设备，如硬盘。

(2) 第 2~4 位：确定所有者拥有的权限（r、w、x）。

(3) 第 5~7 位：确定所属组的权限。

(4) 第 8~10 位：确定其他用户对该文件拥有的权限。

说明：r、w、x 这 3 种权限，对于文件和目录的作用存在差异。

- 对于文件：r 代表可读（read），即可以读取、查看文件内容；w 代表可写（write），即可以修改，但不代表可以删除该文件，删除文件的前提是对该文件所在的目录有写权限；x 代表可执行（execute）。
- 对于目录：r 代表可以使用 ls 命令查看目录内容；w 代表可写，即可以在目录内创建、删除和重命名子目录；x 代表可以进入该目录。

执行结果如图 3-16 所示。

```
-rw-r--r--. 1 root root 19  2月  6日 15:06 new.txt
```

图 3-16 查看文件权限

图 3-16 中的语句代表 new.txt 是一个普通文件，其所有者拥有读和写的权限但无执行权限，同组成员和其他用户仅对其拥有读权限，该文件的创建者是 root 用户，所属组是 root 组。

3.2.6 配置 openEuler 文件权限

文件在创建时会自动为该文件的所有者、同组成员和其他用户分配权限，如果想重新设定文件的访问权限，可以使用 chmod 命令。

该命令的语法如下。

```
chmod [选项] 模式 文件或目录
```

权限可以用符号模式或八进制模式表示。

1. 符号模式

在符号模式下，u（所有者）、g（所属组）、o（其他用户）和 a（所有用户）这 4 个字母代表不同用户范围，使用“+”（添加权限）、“-”（移除权限）和“=”（设置权限）这 3 个符号设定文件的访问权限。

示例：给文件 example1.txt 添加同组成员的读写权限，如图 3-17 所示。



视频讲解

```
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-r--r--. 1 root root 80 2月 6日 14:48 example1.txt
-rw-r--r--. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]# chmod g+w example1.txt
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-rw-r--. 1 root root 80 2月 6日 14:48 example1.txt
-rw-r--r--. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]#
```

图 3-17 给文件 example1.txt 添加同组成员的读、写权限

2. 八进制模式

八进制模式使用数字来表示权限，每个数字对应一种权限：4 代表读权限（r）；2 代表写权限（w）；1 代表执行权限（x）。每个文件或目录的权限由 3 个数字组成，分别表示所有者、组和其他用户的权限。

示例：设置文件 example2.txt 的权限为所有者读写、组读、其他用户无权限，如图 3-18 所示。

```
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-rw-r--. 1 root root 80 2月 6日 14:48 example1.txt
-rw-r--r--. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]# chmod 640 example2.txt
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-rw-r--. 1 root root 80 2月 6日 14:48 example1.txt
-rw-r-----. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]#
```

图 3-18 设置文件权限

如果要修改某个文件或目录的所有者或所属组，可以使用 chown 命令。

```
chown [选项] 用户（组）文件或目录
```

chown 命令常用选项如表 3-16 所示。

表 3-16 chown命令常用选项

选 项	说 明
-c	显示成功修改的所有者/所属组信息
-f	静默模式，不显示错误信息
-h	仅修改符号链接本身的所有者，而不是链接指向的目标文件的所有者
-R	递归修改目录及其所有子目录和文件的所有者

示例：修改文件 example1.txt 的属主和属组，执行结果如图 3-19 所示。


```
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-rw-r--. 1 root root 80 2月 6日 14:48 example1.txt
-rw-r-----. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]# chown euler:group1 example1.txt
[root@openEuler FileSystem]# ls -l
总计 20
-rw-r--r--. 1 root root 0 2月 6日 15:15 echo
-rw-rw-r--. 1 euler group1 80 2月 6日 14:48 example1.txt
-rw-r-----. 1 root root 50 2月 6日 14:33 example2.txt
-rw-r--r--. 1 root root 19 2月 6日 15:16 grep
-rw-r--r--. 1 root root 27 2月 6日 15:20 Hello.txt
-rw-r--r--. 1 root root 19 2月 6日 15:06 new.txt
[root@openEuler FileSystem]#
```

图 3-19 修改文件 example1.txt 的属主和属组

说明：任何用户都可以修改文件的属组，但前提是该用户必须同时是原属组和新属组的成员。只有 root 用户能修改文件的属主。

3.3 openEuler 文件系统管理

3.3.1 修改文件与目录的默认权限与隐藏属性

在 openEuler 操作系统中创建一个文件或目录时，系统会自动为该文件或目录赋予默认权限。默认权限是根据 `umask`（User File-Creation Mode Mask）命令确定的。

`umask` 命令的语法如下。

```
umask [值]
```

`umask` 用于指定“当前用户在创建文件或目录时的权限默认掩码”。默认情况下 `umask` 值为 022，用户也可以根据需要进行修改。默认权限是通过从完全权限中减去 `umask` 值来计算的。计算方法如下。

- （1）文件的默认权限 = 666 - `umask` 值。
- （2）目录的默认权限 = 777 - `umask` 值。

需要注意的是：出于安全考虑，新创建的文件默认没有执行权限，需要用户使用 `chmod` 命令手动赋予。

除了默认属性，为了进一步增强数据安全性，系统还设置了隐藏属性。这些属性无法通过标准的“`ls -l`”命令查看，需要使用 `lsattr`（List Attributes）命令查看文件或目录的隐藏属性，如图 3-20 所示。

```
[root@openEuler FileSystem]# lsattr
-----e----- ./new.txt
-----e----- ./echo
-----e----- ./Hello.txt
-----e----- ./example2.txt
-----e----- ./grep
-----e----- ./example1.txt
[root@openEuler FileSystem]#
```

图 3-20 查看文件或目录的隐藏属性



视频讲解

修改文件或目录的隐藏属性需要使用 `chattr`（Change Attributes）命令。
该命令的语法如下。

`chattr [+==] [属性] 文件或目录`

`chattr` 命令常用属性如表 3-17 所示。

表 3-17 chattr命令常用属性

属 性	说 明
i	文件或目录不可修改，无法删除、重命名、修改内容或创建链接
a	对于文件，只能追加内容，不能删除或修改现有内容；对于目录，只能创建和修改文件，不能删除文件
u	删除文件时，其内容会被保留，便于后续恢复
s	删除文件时，其内容会被彻底覆盖并从硬盘中删除，不可恢复
S	文件内容变更后会立即同步到硬盘，确保数据的完整性

示例：设置 `example1.txt` 文件为不可修改，如图 3-21 所示。

```
[root@openEuler FileSystem]# lsattr
-----e----- ./new.txt
-----e----- ./echo
-----e----- ./Hello.txt
-----e----- ./example2.txt
-----e----- ./grep
-----e----- ./example1.txt
[root@openEuler FileSystem]# chattr +i example1.txt
[root@openEuler FileSystem]# lsattr
-----e----- ./new.txt
-----e----- ./echo
-----e----- ./Hello.txt
-----e----- ./example2.txt
-----e----- ./grep
-----i----- ./example1.txt
[root@openEuler FileSystem]# rm example1.txt
rm: 是否删除普通文件 'example1.txt'? y
rm: 无法删除 'example1.txt': 不允许的操作
[root@openEuler FileSystem]#
```

图 3-21 设置 `example1.txt` 文件为不可修改

说明：查看和修改文件的隐藏属性需要 `root` 权限。



视频讲解

3.3.2 文件访问控制列表

仅使用基本权限管理文件存在局限性，即只能将文件或目录的权限分配给单个组或单个用户。在复杂的生产环境中，对于文件和目录，不同的组需要不同的权限，基本权限无法满足需求。这就需要用到一种更精细的权限控制工具——访问控制列表（Access Control List, ACL）。

使用 `getfacl` 命令查看文件或目录的 ACL，如图 3-22 所示。

```
[root@openEuler FileSystem]# getfacl new.txt
# file: new.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
[root@openEuler FileSystem]#
```

图 3-22 查看文件或目录的 ACL

使用 `setfacl` 命令设置文件或目录的 ACL，语法如下。

```
setfacl [选项] 权限 文件或目录
```

`setfacl` 命令常用选项如表 3-18 所示。

表 3-18 setfacl命令常用选项

选 项	说 明
-m	修改或添加ACL条目
-x	删除指定的ACL条目
-b	移除所有扩展的ACL条目，恢复为默认权限
-k	移除默认的ACL条目
-R	递归设置ACL，适用于目录及其所有子文件和子目录
-d	设置默认ACL，仅对目录有效，新创建的文件或目录会继承这些权限

ACL 规范定义形式如下。

```
[user|group]:[name]:[permissions]
```

示例：为用户 `euler` 设置对文件 `file1.txt` 的读写权限，如图 3-23 所示。

```
[root@openEuler FileSystem]# ls -l file1.txt
-rw-r--r--. 1 root root 144  2月  6日 16:43 file1.txt
[root@openEuler FileSystem]# getfacl file1.txt
# file: file1.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--

[root@openEuler FileSystem]# setfacl -m u:euler:rw file1.txt
[root@openEuler FileSystem]# ls -l file1.txt
-rw-rw-r--+. 1 root root 144  2月  6日 16:43 file1.txt
[root@openEuler FileSystem]# getfacl file1.txt
# file: file1.txt
# owner: root
# group: root
user::rw-
user:euler:rw-
group::r--
mask::rw-
other::r--
```

图 3-23 为用户 euler 设置对文件 file1.txt 的读写权限

3.4 实验项目：文件权限管理

1. 实验背景

某单位有 10 名工作人员，分属 3 个部门。同一部门人员在一个组中，每个用户都有各自的工作目录。需要对每个部门和每个用户在服务器上的工作目录进行权限限制。

2. 实验要求

(1) 用户在自己的工作目录下创建文件并设定权限，检验其他用户和组对该文件的实际使用权限是否和设想的一致。

(2) 以超级用户身份修改文件或目录的所有者，再次验证权限的实际效果。

3. 实验步骤

1) 设置文件权限

(1) 创建组 part1, 创建用户 USER1 并设置密码, 指定用户 USER1 的所属组为 part1, 如图 3-24 和图 3-25 所示。

```
[root@openEuler home]# groupadd part1
[root@openEuler home]# useradd USER1 -g part1
[root@openEuler home]# ls -l
总计 60
drwx-----, 11 alice      alice      4096  3月14日 21:24  alice
drwx-----, 11 bob        bob        4096  3月14日 21:24  bob
drwx-----, 15 euler      euler      4096  2025年  2月  1日  euler
dr-x-----, 11 ftpuser    ftpuser    4096  3月25日 20:08  ftpuser
drwxr-xr-x,  2 root        root        4096  2025年  2月  1日  test
drwx-----, 11 tom        tom        4096  2025年  2月  1日  tom
drwx-----, 11 user01     user01     4096  8月13日 16:30  user01
drwx-----, 11 user02     group01    4096  8月13日 16:42  user02
drwx-----, 11 user1      user1      4096  2025年  2月  1日  user1
drwx-----, 11 user2      group1     4096  2025年  2月  1日  User_1
drwx-----, 11 USER1      part1      4096  8月13日 17:09  USER1
drwx-----, 11 user2      user2      4096  2025年  2月  1日  user2
drwx-----, 11 user3      user3      4096  2025年  2月  1日  user3
drwx-----, 11 user4      user4      4096  2025年  2月  1日  user4
drwx-----, 11 userexample userexample 4096  2025年  2月  1日  userexample
[root@openEuler home]#
```

图 3-24 创建组 part1 和用户 USER1

```
[root@openEuler USER1]# passwd USER1
修改用户USER1的密码。
新的密码：
输入新密码：
passwd: 所有的身份验证令牌已经成功更新。
```

图 3-25 为用户 USER1 设置密码

(2) 在用户 USER1 的主目录下创建目录 test, 进入 test 目录创建空文件 file1, 执行结果如图 3-26 所示。

```
[USER1@openEuler ~]$ cd /home/USER1/
[USER1@openEuler ~]$ mkdir test
[USER1@openEuler ~]$ ls -l
总计 4
drwxr-xr-x, 2 USER1 part1 4096  8月13日 17:32  test
[USER1@openEuler ~]$ cd ./test/
[USER1@openEuler test]$ touch file1
[USER1@openEuler test]$
```

图 3-26 创建 test 目录和空文件 file1

(3) 对文件 file1 设置权限, 使其他用户可以对此文件进行写操作, 并查看设置结果。注意对比权限更改前后的不同, 执行结果如图 3-27 所示。

```
[USER1@openEuler test]$ ls -l
总计 0
-rw-r--r--, 1 USER1 part1 0  8月13日 17:32  file1
[USER1@openEuler test]$ chmod o+w file1
[USER1@openEuler test]$ ls -l
总计 0
-rw-r--rw-, 1 USER1 part1 0  8月13日 17:32  file1
```

图 3-27 修改文件 file1 的权限

(4) 取消同组用户对此文件的读权限。设置完成后查看结果, 如图 3-28 所示。

```
[USER1@openEuler test]$ chmod g-r file1
[USER1@openEuler test]$ ls -l
总计 0
-rw----rw-. 1 USER1 part1 0  8月 13日  17:32 file1
```

图 3-28 取消同组用户对文件 file1 的读权限

(5) 用数字形式为文件 file1 设置权限, 使所有者可读、可写、可执行, 其他用户和所属组用户只有读和执行的权限。设置完成后查看结果, 如图 3-29 所示。

```
[USER1@openEuler test]$ chmod 755 file1
[USER1@openEuler test]$ ls -l
总计 0
-rwxr-xr-x. 1 USER1 part1 0  8月 13日  17:32 file1
[USER1@openEuler test]$
```

图 3-29 使用数字形式设置文件 file1 的权限

(6) 用数字形式修改文件 file1 的权限, 使得只有所有者可对该文件执行完全操作, 其他任何用户都没有权限。设置完成后查看结果, 如图 3-30 所示。

```
[USER1@openEuler test]$ chmod 700 file1
[USER1@openEuler test]$ ls -l
总计 0
-rwx-----. 1 USER1 part1 0  8月 13日  17:32 file1
```

图 3-30 再次修改文件 file1 的权限

(7) 为同组用户添加写权限。设置完成后查看结果, 如图 3-31 所示。

```
[USER1@openEuler test]$ chmod g+w file1
[USER1@openEuler test]$ ls -l
总计 0
-rwx-w----. 1 USER1 part1 0  8月 13日  17:32 file1
[USER1@openEuler test]$
```

图 3-31 对文件 file1 添加同组用户的写权限

(8) 回到上层目录, 查看文件 test 的权限, 如图 3-32 所示。

```
[USER1@openEuler test]$ cd ../
[USER1@openEuler ~]$ ls -l
总计 4
drwxr-xr-x. 2 USER1 part1 4096  8月 13日  17:32 test
```

图 3-32 查看文件 test 的权限

(9) 为其他用户添加对文件 test 的写权限, 这一步需要切换至 root 用户执行操作, 执行结果如图 3-33 所示。

```
[USER1@openEuler ~]$ chmod o+r test/
[USER1@openEuler ~]$ ls -l
总计 4
drwxr-xr-x. 2 USER1 part1 4096  8月 13日  17:32 test
```

图 3-33 为其他用户添加对文件 test 的写权限

2) 改变文件所有者

(1) 查看目录 `test` 及其子文件的所属用户和组，执行结果如图 3-34 所示。

```
[USER1@openEuler ~]$ ls -l
总计 4
drwxr-xr-x. 2 USER1 part1 4096  8月 13日  17:32 test
[USER1@openEuler ~]$ cd test/
[USER1@openEuler test]$ ls -l
总计 0
-rwx-w----. 1 USER1 part1 0  8月 13日  17:32 file1
[USER1@openEuler test]$
```

图 3-34 查看目录 `test` 及其子文件的所属用户和所属组

(2) 将目录 `test` 及其中所有文件的所有者都改为 `euler`，设置完成后查看结果，如图 3-35 所示。

```
[root@openEuler USER1]# chown euler -R test/
[root@openEuler USER1]# ls -l
总计 4
drwxr-xr-x. 2 euler part1 4096  8月 13日  17:32 test
[root@openEuler USER1]# cd test/
[root@openEuler test]# ls -l
总计 0
-rwx-w----. 1 euler part1 0  8月 13日  17:32 file1
[root@openEuler test]#
```

图 3-35 将目录 `test` 及其中所有文件的所有者改为 `euler`

(3) 删除目录 `test` 及其中的文件，如图 3-36 所示。

```
[root@openEuler USER1]# rm -rf test/
[root@openEuler USER1]# ls
[root@openEuler USER1]#
```

图 3-36 删除目录 `test` 及其中的文件

3.5 法治教育：文件保密与国家机密保护

在信息技术高度发达的今天，文件保密与国家机密保护显得尤为重要。作为 IT 从业者或使用者，不仅要掌握文件操作的基本技能，更要深刻认识文件保密的重要性，尤其是涉及国家机密的文件。

首先，要明确什么是国家机密。国家机密是指关系国家安全和利益，依照法定程序确定，在一定时间内只限一定范围的人员知悉的事项。这些机密一旦泄露，可能会对国家安全和利益造成重大损害。

在日常工作中，系统管理人员可能会接触到各种文件，其中一些可能涉及敏感信息或国家机密。因此，必须严格遵守相关法律法规和规章制度，确保这些文件的安全。具体来说，应该采取以下措施。

一是加强文件权限管理。通过合理使用 `chmod`、`chown` 等命令，可以严格控制文件的访问权限，确保只有授权的人员才能访问和修改这些文件。同时，还可以使用 ACL 等更精细的权限控制方法，以满足不同用户对文件和目录的不同权限需求。

二是使用加密技术保护文件安全。对于特别敏感的文件，可以使用加密技术进行保护，

确保即使文件被非法获取,也无法被轻易解密和读取。

三是定期备份和审计文件。通过定期备份文件,可以防止意外情况导致文件丢失或损坏。同时,通过审计文件访问记录,能够及时发现并处理任何可疑行为。

四是加强员工培训和意识教育。作为 IT 从业者或使用者,应该不断增强自身安全意识,了解并遵守相关法律法规和规章制度。同时,企业也应该加强员工培训,提高员工对文件保密和国家机密保护的认识和重视程度。

文件保密与国家机密保护是每位 IT 从业者或使用者的责任和义务。从业人员应该时刻保持警惕,采取有效措施确保文件的安全,为维护国家安全和利益贡献自己的力量!

本章小结

本章详细介绍了 openEuler 操作系统的文件与目录管理功能,涵盖了文件系统的概念、文件管理常用命令及文件权限的配置知识。通过对 openEuler 文件系统的深入剖析,读者可以较为全面地掌握文件管理的核心技能,为高效使用和管理 openEuler 系统奠定坚实基础。

练习题

一、选择题

1. openEuler 文件系统中,普通文件不包括()。
A. 文本文件
B. 二进制文件
C. 数据文件
D. 符号链接
2. 以下目录()用于存放系统配置文件。
A. /bin
B. /etc
C. /dev
D. /home
3. 绝对路径的特点不包括()。
A. 唯一性
B. 依赖于当前工作目录
C. 从根目录开始
D. 不会因当前工作目录变化而改变
4. 命令()用于查看文件内容,且支持向前或向后滚动、搜索特定内容。
A. cat
B. more
C. less
D. head
5. 在 openEuler 中,文件权限的 rwx 对于文件和目录的作用不同,以下描述正确的是()。
A. 对于文件, x 表示可以进入该文件
B. 对于目录, r 表示可以读取目录内的文件内容
C. 对于文件, w 表示可以修改文件内容
D. 对于目录, w 表示可以查看目录内容
6. 在 openEuler 中,目录()用于存放系统运行时产生的临时数据,且重启后不会清空。
A. /tmp
B. /run
C. /mnt
D. /media
7. 命令()用于在文件系统中递归查找文件。
A. locate
B. find
C. grep
D. whereis

D. mkdir -r

D. setfacl -b

D. `setfacl -m u:euler:rw, g:developers:rwx file1.txt`

3. 简述绝对路径和相对路径的区别。