

实验指导手册

Linux 操作系统相关命令

Linux 中的命令格式为：command [options] [arguments] 中括号表示可选的，即有些命令不需要选项也不需要参数，但有的命令在运行时需要多个选项或参数。

- options(选项)：选项是调整命令执行行为的开关，选项的不同决定了命令的显示结果不同。
- arguments(参数)：参数是指命令的作用对象。

vi/vim

文本编辑器，若文件存在则是编辑，若不存在则是创建并编辑文本。

命令语法：

```
vim [参数]
```

参数说明：可编辑的文件名。

命令示例：

- 编辑名为 clusterconfig 的 xml 文本：

```
vim clusterconfig.xml
```

注：

vim 编辑器有以下三种模式：

- 正常模式：其它模式下按 Esc 或 Ctrl+[进入，左下角显示文件名或为空。
- 插入模式：正常模式下按 i 键进入，左下角显示--INSERT--。
- 可视模式：正常模式下按 v 键进入，左下角显示--VISUAL--。

退出命令（正常模式下）：

- :wq 保存并退出。
- :q! 强制退出并忽略所有更改。
- :e! 放弃所有修改，并打开原有文件。

cd

显示当前目录的名称，或切换当前的目录（打开指定目录）。

命令语法：

```
cd [参数]
```

参数说明：

- 无参数：切换用户当前目录。
- `.`：表示当前目录；
- `..`：表示上一级目录；
- `~`：表示 home 目录；
- `/`：表示根目录。

命令示例：

- 切换到 `usr` 目录下的 `bin` 目录中：

```
cd /usr/bin
```

- 切换到用户 home 目录：

```
cd
```

- 切换到当前目录(cd 后面接一个.)：

```
cd .
```

- 切换到当前目录上一级目录(cd 后面接两个.)：

```
cd ..
```

- 切换到用户 home 目录：

```
cd ~
```

- 切换到根目录下：

```
cd /
```

注：切换目录需要理解绝对路径和相对路径这两个概念。

- 绝对路径：在 Linux 中，绝对路径是从 `/`（即根目录）开始的，例如 `/opt/software`、`/etc/profile`，如果目录以 `/` 就是绝对目录。
- 相对路径：是以 `.` 或 `..` 开始的目录。`.` 表示用户当前操作所在的位置，而 `..` 表示上级目录。例如 `./gs_om` 表示当前目录下的文件或者目录。

mv

文件或目录改名(move (rename) files)或将文件或目录移入其它位置,经常用来备份文件或者目录。

命令语法：

```
mv [选项] 参数 1 参数 2
```

常用选项：

- -b：若需覆盖文件，则覆盖前先行备份。

参数说明：

- 参数 1：源文件或目录。
- 参数 2：目标文件或目录。

命令示例：

- 将文件 python 重命名为 python.bak：

```
mv python python.bak
```

- 将/physical/backup 目录下的所有文件和目录移到/data/dbn1 目录下：

```
mv /physical/backup/* /data/dbn1
```

curl

在 Linux 中 curl 是一个利用 URL 规则在命令行下工作的文件传输工具。支持文件的上传和下载，是综合传输工具。

命令语法：

```
curl [选项] [URL]
```

常用选项：

- -A/--user-agent <string>：设置用户代理发送给服务器；
- -C/--continue-at <offset>：断点续转；
- -D/--dump-header <file>：把 header 信息写入到该文件中；
- -e/--referer：来源网址；
- -o/--output：把输出写到该文件中；
- -O/--remote-name：把输出写到该文件中，保留远程文件的文件名；
- -s/--silent：静默模式。不输出任何东西；
- -T/--upload-file <file>：上传文件；
- -u/--user <user[:password]>：设置服务器的用户和密码；
- -x/--proxy <host[:port]>：在给定的端口上使用 HTTP 代理；
- -#/--progress-bar：进度条显示当前的传送状态。

参数说明：

- URL：指定的文件传输 URL 地址。

命令示例：

- 将 url(<https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo>)的内容保存到/etc/yum.repos.d/CentOS-Base.repo 文件中。

```
curl -o /etc/yum.repos.d/CentOS-Base.repo https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo
```

- 如果在传输过程中掉线，可以使用-C 的方式进行续传。

```
curl -C -O https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo
```

yum

Shell 前端软件包管理器。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，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载和安装。

命令语法：

```
yum [options] [command] [package ...]
```

常用选项：

- -h：查看帮助；
- -y：当安装过程提示选择全部为 "yes"；
- -q：不显示安装的过程。

参数说明：

- command：要进行的操作。
- package：安装的包名。

命令示例：

- 列出所有可更新的软件清单命令：

```
yum check-update
```

- 更新所有软件命令：

```
yum update
```

- 列出所有可安装的软件清单命令：

```
yum list
```

- 安装指定的软件：

```
yum install -y libaio-devel flex bison ncurses-devel glibc-devel patch lsb_release wget python3
```

wget

wget 是 Linux 下下载文件的最常用命令。wget 支持 HTTP,HTTPS 和 FTP 协议,支持自动下载,即可以在用户退出系统后在后台执行,直到下载结束。

命令语法:

```
wget [选项] [URL]
```

常用选项:

- -c : 接着下载没下载完的文件 ;
- -b : 启动后转入后台执行 ;
- -P : 指定下载目录 ;
- -O : 变更下载文件名 ;
- --ftp-user --ftp-password : 使用 FTP 用户认证下载。

参数说明:

- 指定的文件下载 URL 地址。

命令示例 :

- 下载 openGauss 数据库安装文件到当前文件夹 :

```
wget https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz
```

- 使用 wget 断点续传 :

```
wget -c https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz
```

ln

为某一个文件在另外一个位置建立一个同步的链接 (软硬链接 , 不带选项为硬链接) 。

当需要在不同的目录 , 用到相同的文件时 , 就不需要在每一个需要要的目录下都放一个必须相同的文件 , 我们只要在某个固定的目录 , 放上该文件 , 然后在 其它的目录下用 ln 命令链接 (link) 它就可以 , 不必重复的占用磁盘空间。

命令语法 :

```
ln [选项] 参数 1 参数 2
```

常用选项 :

- -b --删除 , 覆盖以前建立的链接 ;
- -d --允许超级用户制作目录的硬链接 ;
- -s --软链接(符号链接)。

参数说明 :

- 参数 1 : 源文件或目录。
- 参数 2 : 被链接的文件或目录。

命令示例 :

- 为 python3 文件创建软链接/usr/bin/python , 如果 python3 丢失 , /usr/bin/python 将失效 :

```
ln -s python3 /usr/bin/python
```

- 为 python3 创建硬链接/usr/bin/python , python3 与/usr/bin/python 的各项属性相同：

```
ln python3 /usr/bin/python
```

mkdir

创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

命令语法：

```
mkdir [选项] [参数]
```

常用选项：

- -p --可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录（递归）；
- -v --每次创建新目录都显示信息；
- -m --设定权限<模式> (类似 chmod)，而不是 rwxrwxrwx 减 umask。

参数说明：

- 需要创建的目录。

命令示例：

- 创建一个空目录：

```
mkdir test
```

- 递归创建多个目录：

```
mkdir -p /opt/software/openGauss
```

- 创建权限为 777 的目录(目录的权限为 rwxrwxrwx)：

```
mkdir -m 777 test
```

chmod 命令

更改文件权限。

命令语法：

```
chmod [选项] <mode> <file...>
```

常用选项：

- -R, --以递归的方式对目前目录下的所有文件与子目录进行相同的权限变更。

参数说明：

- mode：权限设定字串，详细格式如下：

```
[ugoa...][[+=[rwxX]...][...],
```

其中，[ugoa...]：u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示所有（包含上面三者）；[+=[rwxX]：+ 表示增加权限，- 表示取消权限，= 表示唯一设定权限；[rwxX]：r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该档案是个子目录或者该档案已经被设定过为可执行。

- file：文件列表（单个或者多个文件、文件夹）。

命令示例：

- 设置所有用户可读取文件 cluterconfig.xml：

```
chmod ugo+r cluterconfig.xml
或
chmod a+r cluterconfig.xml
```

- 设置当前目录下的所有档案与子目录皆设为任何人可读写：

```
chmod -R a+rw *
```

数字权限使用格式：

- 这种使用方式中，规定数字 4、2 和 1 表示读、写、执行权限，即 r=4,w=2,x=1。
- 例：rwx = 7 (4+2+1)；rw = 6 (4+2)；r-x = 5 (4+0+1)；r-- = 4 (4+0+0)；--x = 1 (0+0+1)；

每个文件都可以针对三个粒度，设置不同的 rwx(读写执行)权限。即我们可以用三个 8 进制数字分别表示 拥有者、群组、其它组(u、g、o)的权限详情，并用 chmod 直接加三个 8 进制数字的方式直接改变文件权限。语法格式为：

```
chmod <abc> file...
```

其中，a,b,c 各为一个数字，分别代表 User、Group、及 Other 的权限，相当于简化版的 chmod u=权限,g=权限,o=权限 file...，而此处的权限将用 8 进制的数字来表示 User、Group、及 Other 的读、写、执行权限。

命令示例：

- 赋予 cluterconfig.xml 文件可读可写可执行权限（所有权限）：

```
chmod 777 cluterconfig.xml
```

- 赋予/opt/software/openGauss 目录下所有文件及其子目录 用户所有权 群组可读可执行权限，其他用户可读可执行权限：

```
chmod R 755 /opt/software/openGauss
```

chown

利用 chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户 ID；组可以是组名或者组 ID；文件是以空格分开的要改变权限的文件列表，支持通配符。只有系统管理者(root)才有这样的权限。使用权限：**root**。

命令语法：

```
chown [选项] user[:group] file...
```

常用选项:

- -c：显示更改的部分的信息；
- -f：忽略错误信息；
- -R：处理指定目录以及其子目录下的所有文件。

参数说明

- user：新的文件拥有者的使用者 ID。
- group：新的文件拥有者的使用者组(group)。
- file：文件。

命令示例：

- 将文件 file1.txt 的拥有者设为 omm，群体的使用者 dbgrp:

```
chown omm:dbgrp /opt/software/openGauss/clusterconfig.xml
```

- 将目前目录下的所有文件与子目录的拥有者皆设为 omm，群体的使用者 dbgrp:

```
chown -R omm:dbgrp *
```

ls

列出文件和目录的内容。

命令语法：

```
ls [选项] [参数]
```

常用选项：

- -l --以长格式显示，列出文件的详细信息，如创建者，创建时间，文件的读写权限列表等等；
- -a --列出文件下所有的文件，包括以"."和".."开头的隐藏文件（Linux 下文件隐藏文件是以 .开头的，如果存在 .. 代表存在着父目录）；
- -d --列出目录本身而非目录内的文件，通常要与-l 一起使用；

- -R --同时列出所有子目录层，与-l 相似，只是不显示出文件的所有者，相当于编程中的“递归”实现；
- -t --按照时间进行文件的排序，Time (时间)；
- -s --在每个文件的后面打印出文件的大小，size (大小)；
- -S --以文件的大小进行排序。

参数说明：

- 目录或文件。

命令示例：

- 以长格式列出当前目录中的文件及目录：

```
ls -l
```

cp

复制文件或者目录。

命令语法：

```
cp [选项] 参数 1 参数 2
```

常用选项：

- -f --如果目标文件无法打开则将其移除并重试(当 -n 选项存在时则不需再选此项)；
- -n --不要覆盖已存在的文件(使前面的 -i 选项失效)；
- -I --覆盖前询问(使前面的 -n 选项失效)；
- -p --保持指定的属性(默认：模式,所有权,时间戳)，如果可能保持附加属性：环境、链接、xattr 等；
- -R,-r --复制目录及目录内的所有项目。

参数说明：

- 参数 1：源文件。
- 参数 2：目标文件。

命令示例：

- 将 home 目录中的 abc 文件复制到 opt 目录下：

```
cp /home/abc /opt
```

注：目标文件存在时，会询问是否覆盖。这是因为 cp 是 cp -i 的别名。目标文件存在时，即使加了-f 标志，也还会询问是否覆盖。

rm

删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

rm 是一个危险的命令，使用的时候要特别当心，否则整个系统就会毁在这个命令（比如在 /（根目录）下执行 `rm * rf`）。所以，我们在执行 rm 之前最好先确认一下在哪个目录，到底要删除什么东西，操作时保持高度清醒的头脑。

命令语法：

```
rm [选项] 文件
```

常用选项：

- `-f` --忽略不存在的文件，从不给出提示；
- `-r` --指示 rm 将参数中列出的全部目录和子目录均递归地删除。

参数说明：

- 需要删除的文件或目录。

命令示例：

- 删除文件：

```
rm qwe
```

注：输入 `rm qwe` 命令后，系统会询问是否删除，输入 `y` 后就会删除文件，不想删除文件则输入 `n`。

- 强制删除某个文件：

```
rm -rf clusterconfig.log
```

cat

连接文件并在标准输出上输出。这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

命令语法：

```
cat [选项] [参数]
```

常用选项：

- `-E` --在每行结束显示\$；
- `-n` --由 1 开始对给所有输出行编号；
- `-b` 或 `--number-nonblank`：和 `-n` 相似，只不过对于空白行不编号；
- `-v` --使用 `^` 和 `M-` 符号，除了 `LFD` 和 `TAB` 之外。

参数说明：

- 可操作的文件名。

命令示例:

- 显示 testfile 文件的内容：

```
cat testfile
```

- 把 testfile1 和 testfile2 的文档内容加上行号（空白行不加）之后将内容追加到 testfile3 文档里：

```
cat -b testfile1 testfile2 >> testfile3
```

- 向/etc/profile 中追加内容（输入 EOF 表示结束追加）：

```
cat >>/etc/profile<<EOF
>export LD_LIBRARY_PATH=$packagePath/script/gspylib/clib:$LD_LIBRARY_PATH
>EOF
```

注：

- EOF 是 end of file 的缩写，表示"文字流"（stream）的结尾。"文字流"可以是文件（file），也可以是标准输入（stdin）。在 Linux 系统之中，EOF 是当系统读取到文件结尾，所返回的一个信号值（也就是-1）。