

## 实验二 openEuler 基础命令及文本编辑器的使用

### 实验目的

- (1) 掌握 bash 命令的基本操作;
- (2) 掌握文件管理命令的常见操作。

### 实验内容

openEuler 操作系统的 bash 命令和文件管理命令的使用。

### 实验指导

#### 1. bash 命令基本操作

步骤 1 打开 VirtualBox，启动 openEuler 虚拟机，并使用 root 用户身份登录虚拟机。

步骤 2 练习使用基本的 bash 命令。

使用 reboot 命令重启 openEuler 操作系统。

```
[root@localhost ~]# reboot
```

重启之后使用 root 账户重新登录到 openEuler 操作系统。

使用 logout，或 exit 退出登录。

```
[root@localhost ~]# logout
```

#再次使用 root 用户重新登录到 openEuler 操作系统

```
[root@localhost ~]# su - openeuler      #切换用户
```

```
[openeuler@localhost ~]# exit           #退出当前用户回退到 root 用户
```

```
[root@localhost ~]#
```

exit 命令也可以操作退出登录，但是如果经常切换用户，建议每次切换后都使用 exit 退出当前用户。

#### 2. 目录及文件基本操作

##### 2.1 目录操作

步骤 3 使用 pwd 命令查看当前所在目录位置。

```
[root@localhost ~]# pwd
```

```
/root
```

```
[root@localhost ~]#
```

#回显表示当前是在/root 根目录下

步骤 4 ls 查看命令。

使用 ls 查看当前目录下的文件及文件夹。

```
[root@localhost ~]# ls
```

```
anaconda-ks.cfg
```

```
[root@localhost ~]# ls .
```

```
anaconda-ks.cfg
```

#回显表示当前目录有一个 anaconda-ks.cfg 文件

显示上一级目录的文件及文件夹。

```
[root@localhost ~]# ls ..
bin  dev  home  lib64      media  opt   root  sbin  sys  usr
boot  etc  lib   lost+found  mnt    proc  run   srv   tmp  var
```

查看/tmp 目录下的文件及文件夹。

```
[root@localhost ~]# ls /tmp
systemd-private-92622a8f3c5b45d6b45c4cc9012916e6-chronyd.service-6X7mn1
systemd-private-92622a8f3c5b45d6b45c4cc9012916e6-systemd-logind.service-GKj4CO
```

显示当前目录的所有文件及文件夹。

```
[root@localhost ~]# ls -a
.  anaconda-ks.cfg  .bash_logout  .bashrc  .tcshrc
.. .bash_history   .bash_profile .cshrc
# 回显表示当前目录存在隐藏文件及目录。
```

显示当前目录非隐藏的文件及文件夹详细信息。

```
[root@localhost ~]# ls -l
total 4
-rw-----. 1 root root 1986 Jul  8 11:07 anaconda-ks.cfg
```

显示当前目录所有文件及文件夹详细信息。

```
[root@localhost ~]# ls -al
total 36
dr-xr-x---.  2 root root 4096 Jul  8 11:38 .
dr-xr-xr-x. 18 root root 4096 Jul  8 11:00 ..
-rw-----.  1 root root 1986 Jul  8 11:07 anaconda-ks.cfg
-rw-----.  1 root root  236 Jul  8 11:45 .bash_history
-rw-r--r--.  1 root root   18 Oct 29  2019 .bash_logout
-rw-r--r--.  1 root root  176 Oct 29  2019 .bash_profile
-rw-r--r--.  1 root root  176 Oct 29  2019 .bashrc
-rw-r--r--.  1 root root  100 Oct 29  2019 .cshrc
-rw-r--r--.  1 root root  129 Oct 29  2019 .tcshrc
```

步骤 5 cd 切换目录。

切换到系统根目录。

```
[root@localhost ~]# cd /
[root@localhost /]#
# 注意观察, “~” 变成了 “/” 。
```

切换到 “/etc/” 目录。

```
[root@localhost /]# cd /etc
[root@localhost etc]#
```

使用相对路径方法, 切换到 “/etc/sysconfig/” 目录。

```
[root@localhost etc]# cd sysconfig
[root@localhost sysconfig]#
```

使用绝对路径方法, 切换到 “/etc/sysconfig/” 目录。

```
[root@localhost etc]# cd /etc/sysconfig
[root@localhost sysconfig]#
```

使用 “cd ..” 命令切换到上一级目录。

```
[root@localhost sysconfig]# cd ..
```

```
[root@localhost etc]#
```

使用 “cd” 切换到用户家目录。

```
[root@localhost sysconfig]#cd
```

```
[root@localhost ~]#
```

使用 “cd -” 返回进入此目录之前所在的目录。

```
[root@localhost sysconfig]# cd -
```

```
/etc
```

```
[root@localhost etc]#
```

使用 “cd ~” 切换到用户家目录。

```
[root@localhost etc]#cd /etc/sysconfig
```

```
[root@localhost sysconfig]#cd ~
```

```
[root@localhost ~]#
```

步骤 6 mkdir 命令创建目录。

在当前文件夹快速创建 test1 目录。

```
[root@localhost ~]#mkdir /root/test1
```

```
[root@localhost ~]#ls
```

```
anaconda-ks.cfg test1
```

使用相对路径创建目录。

```
[root@localhost ~]#mkdir ./test2
```

```
[root@localhost ~]#ls
```

```
anaconda-ks.cfg test1 test1
```

使用绝对路径创建目录。

```
[root@localhost ~]#mkdir test3
```

```
[root@localhost ~]#ls
```

```
anaconda-ks.cfg test1 test2 test3
```

步骤 7 touch 命令创建文件。

创建 huawei.txt 文件。

```
[root@localhost ~]#cd test1
```

```
[root@localhost test1]#touch /root/huawei.txt
```

```
[root@localhost test1]#touch huawei1.txt
```

```
[root@localhost test1]#ls
```

```
huawei.txt huawei1.txt
```

步骤 8 cp 复制命令。

复制 huawei.txt 到 /root/test2 目录，并命名为 huawei.txt.bak。

```
[root@localhost test1]#cp huawei.txt /root/test2/huawei.txt.bak
```

```
[root@localhost test1]#ls /root/test2
```

```
huawei.txt.bak
```

复制 test1 目录到 /root/test2 目录。

```
[root@localhost test1]#cp -r /root/test1 /root/test2/
```

```
[root@localhost test1]#ls /root/test2/
```

```
huawei.txt.bak test1
```

步骤 9 rm 删除命令。

删除 /root/test1 目录下的 huawei.txt 文件。

```
[root@localhost test1]#rm huawei.txt
```

```
rm: 是否删除普通空文件 'huawei.txt'? y    //这里输入 y, 同意删除
[root@localhost test1]#ls
[root@localhost test1]#
[root@localhost ~]# touch /root/huawei1.txt
[root@localhost ~]# rm -f /root/huawei1.txt
```

删除/root 目录下的 test1 文件夹。

```
[root@localhost test1]#cd
[root@localhost ~]#ls
anaconda-ks.cfg  test1  test2  test3
[root@localhost ~]#rmdir /root/test1
[root@localhost ~]#ls
anaconda-ks.cfg  test2  test3
[root@localhost ~]#mkdir /root/test1
[root@localhost ~]# rm -r /root/test1
rm: remove directory '/root/test1'? y
```

步骤 10 mv 命令。

剪切/root/test2 目录下的 huawei.txt.bak 文件到/root 目录下, 并重命名为 huawei.txt 文件。

```
[root@localhost ~]#mv /root/test2/huawei.txt.bak ~/huawei.txt
[root@localhost ~]#ls
anaconda-ks.cfg  huawei.txt  test2  test3
```

步骤 11 ln 链接命令。

创建 huawei.txt 的硬链接到/test3, 并命名为 huawei1.txt。

```
[root@localhost ~]#ln huawei.txt /root/test3/huawei1.txt
```

创建 huawei.txt 的软链接到/test3, 并命名为 huawei2.txt。

```
[root@localhost ~]#ln -s huawei.txt /root/test3/huawei2.txt
```

查看文件的 inode 节点信息。huawei.txt 文件的节点信息和 huawei1.txt 的节点信息是一致的。huawei.txt 文件的节点信息和 huawei2.txt 的节点信息是一致的。

```
[root@localhost ~]#ls -li
798457 -rw-----. 1 root root 1631  6月  9 16:40 anaconda-ks.cfg
798572 -rw-----. 2 root root    0  6月 10 10:20 huawei.txt
[root@localhost ~]#cd test3/
[root@localhost test3]#ls -li
798572 -rw-----. 2 root root    0  6月 10 10:20 huawei1.txt
798551 lrwxrwxrwx. 1 root root 10   6月 10 11:37 huawei2.txt -> huawei.txt
```

删除 huawei.txt 文件, 再次查看文件内容。

```
[root@localhost test3]# rm /root/huawei.txt    #删除源文件 huawei.txt
rm: 是否删除普通空文件 '/root/huawei.txt'? y
[root@localhost test3]# ls
huawei1.txt  huawei2.txt
[root@localhost test3]# cat huawei1.txt        #打开硬链接文件 huawei1.txt 正常
[root@localhost test3]# cat huawei2.txt        #打开软链接文件 huawei2.txt 失败
cat: huawei2.txt: 没有那个文件或目录
```

## 2.2 文件查看

步骤 12 拷贝/etc/passwd 文件到/root 目录。

```
[root@localhost test3]# cd  
[root@localhost ~]# cp /etc/passwd ~
```

步骤 13 cat 查看命令。

cat 查看 passwd 文件的内容。

```
[root@localhost ~]# cat passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin  
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/sbin/nologin  
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin  
systemd-timesync:x:998:995:systemd Time Synchronization:/sbin/nologin  
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin  
tss:x:59:59:Account used by the trousers package to sandbox the tcsd  
daemon:/dev/null:/sbin/nologin  
polkitd:x:996:993:User for polkitd:/sbin/nologin  
sasauth:x:995:76:Saslauthd user:/run/saslauthd:/sbin/nologin  
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin  
libstoragemgmt:x:994:991:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin  
pcp:x:993:990:PCP:/var/lib/pcp:/sbin/nologin  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin  
dnsmasq:x:988:988:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin  
radvd:x:75:75:radvd user:/sbin/nologin  
sanlock:x:179:179:sanlock:/var/run/sanlock:/sbin/nologin  
qemu:x:107:107:qemu user:/sbin/nologin  
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin  
dhcpcd:x:177:177:DHCP server:/sbin/nologin  
named:x:25:25:Named:/var/named:/bin/false  
gluster:x:987:985:GlusterFS daemons:/run/gluster:/sbin/nologin  
setroubleshoot:x:986:984:/var/lib/setroubleshoot:/sbin/nologin  
geoclue:x:985:983:User for geoclue:/var/lib/geoclue:/sbin/nologin  
cockpit-ws:x:984:982:User for cockpit-ws:/sbin/nologin  
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM services:/var/lib/Pegasus:/sbin/nologin  
chrony:x:983:981:/var/lib/chrony:/sbin/nologin  
pcpqa:x:982:980:PCP Quality Assurance:/var/lib/pcp/testsuite:/bin/bash  
pesign:x:981:979:Group for the pesign signing daemon:/var/run/pesign:/sbin/nologin
```

```
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
dbus:x:978:978:System Message Bus:/usr/sbin/nologin
openeuler:x:1000:1000:openEuler:/home/openeuler:/bin/bash
```

步骤 14 head 查看命令。

head 查看文件前 10 行内容。

```
[root@localhost ~]# head passwd # head 命令不加任何参数默认查看文件前 10 行内容
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

head 查看文件前 5 行内容。

```
[root@localhost ~]# head -n 5 passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

head 查看文件除最后 40 行以外的全部内容。

```
[root@localhost ~]# head -n -40 passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```

head 查看文件前 10 个字节内容。

```
[root@localhost ~]# head -c 10 passwd
root:x:0:0[root@localhost ~]#
```

思考：如何查看文件除了最后 100 个字节以外的全部内容？

步骤 15 tail 查看命令。

tail 查看文件最后 10 行内容。

```
[root@localhost ~]# tail passwd #同 head 一样，默认显示最后 10 行内容。
```

```
cockpit-ws:x:984:982:User for cockpit-ws:/:/sbin/nologin
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM services:/var/lib/Pegasus:/sbin/nologin
chrony:x:983:981::/var/lib/chrony:/sbin/nologin
pcpqa:x:982:980:PCP Quality Assurance:/var/lib/pcp/testsuite:/bin/bash
pesign:x:981:979:Group for the pesign signing daemon:/var/run/pesign:/sbin/nologin
```

```
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
radiusd:x:95:95:radiusd user:/var/lib/radiusd:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
dbus:x:978:978:System Message Bus:/usr/sbin/nologin
openeuler:x:1000:1000:openEuler:/home/openeuler:/bin/bash
```

tail 查看文件后 5 行内容。

```
[root@localhost ~]# tail -n 5 passwd
```

查看文件除了前面 20 行以外剩下的所有内容。

```
[root@localhost ~]# tail -n -20 passwd
```

步骤 16 Less 查看命令。

less 查看文件，按上下键翻行，按空格键向下翻页，按 Q 键退出。

```
[root@localhost ~]# less passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
...
```

步骤 17 More 查看命令。

more 查看文件，按空格键向下翻页，直至退出。也可以按 Q 键退出。

```
[root@localhost ~]# more passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
...
```

## 2.3 查找命令

步骤 18 Find 命令的使用。

查找/etc 目录下以 passwd 命名的文件。

```
[root@localhost ~]# find /etc -name passwd
/etc/raddb/mods-enabled/passwd
/etc/raddb/mods-available/passwd
/etc/passwd
/etc/pam.d/passwd
```

```
[root@localhost ~]# find /root -mtime -2
/root
/root/test2
/root/test2/test1
/root/test2/test1/huawei.txt
/root/.bash_history
/root/passwd
/root/test3
```

```
/root/test3/huawei1.txt
/root/test3/huawei2.txt
/root/anaconda-ks.cfg
```

查找/root/目录下属于 root 用户的文件。

```
[root@localhost ~]# find /root -user root
/root
/root/.bashrc
/root/.bash_profile
/root/test2
/root/test2/test1
/root/test2/test1/huawei.txt
/root/.bash_logout
/root/.bash_history
/root/.tcshrc
/root/passwd
/root/.cshrc
/root/test3
/root/test3/huawei1.txt
/root/test3/huawei2.txt
/root/anaconda-ks.cfg
```

查找/etc/目录下大于 512K 的文件。

```
[root@localhost ~]# find /etc -size +512k
/etc/services
/etc/ssh/moduli
/etc/brlty/Contraction/zh-tw.ctb
/etc/selinux/targeted/policy/policy.31
/etc/udev/hwdb.bin
```

#### 步骤 19 which 的使用。

查看 pwd 命令的绝对路径。which 是根据使用者所配置的 PATH 变量内的目录去搜寻可运行档的，所以，不同的 PATH 配置内容所找到的命令是不一样的。

```
[root@localhost ~]# which pwd
/usr/bin/pwd
```

#### 步骤 20 whereis 命令的使用。

该指令只能用于查找二进制文件、源代码文件和 man 手册页。

使用指令“whereis”查看指令“bash”的位置。

```
[root@localhost ~]# whereis bash
bash: /usr/bin/bash
```

## 2.4 打包和压缩命令

#### 步骤 21 zip 命令的使用。

使用 zip 命令制作 .zip 格式的压缩包。

第一行命令中，-r 参数表示递归打包包含子目录的全部内容，-q 参数表示为安静模式，即不向屏幕输出信息，-o，表示输出文件，需在其后紧跟打包输出文件名。要被打包的参数可以是文件也可以是目录。

```
[root@localhost ~]# zip -r -q -o passwd.zip passwd
```



```
[root@localhost ~]# ls
anaconda-ks.cfg  passwd  passwd.zip
```

按照不同级别压缩文件。

压缩级别为 9 和 1（9 最大，1 最小），重新打包

```
[root@localhost ~]# zip -r -9 -q -o passwd1.zip passwd
[root@localhost ~]# zip -r -1 -q -o passwd2.zip passwd
[root@localhost ~]# ls -lh
-rw-----. 1 root root 1.6K  6月  9 16:40 anaconda-ks.cfg
-rw-----. 1 root root 2.5K  6月 10 16:35 passwd
-rw-----. 1 root root 1.2K  6月 10 16:35 passwd1.zip
-rw-----. 1 root root 1.3K  6月 10 16:35 passwd2.zip
-rw-----. 1 root root 1.2K  6月 10 16:35 passwd.zip
```

步骤 22 unzip 命令的使用。

unzip 解压到当前目录下。

```
[root@localhost ~]# unzip passwd.zip
Archive:  passwd.zip
replace passwd? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: passwd
[root@localhost ~]# ls
anaconda-ks.cfg  passwd  passwd1.zip  passwd2.zip  passwd.zip  test2  test3
[root@localhost ~]#
```

unzip 解压到指定文件夹下

将压缩文件 password1.zip 在指定目录/root/test3 下解压缩，如果已有相同的文件存在，要求 unzip 命令不覆盖原先的文件。

```
[root@localhost ~]# unzip -n passwd1.zip -d /root/test3
Archive:  passwd1.zip
  inflating: /root/test3/passwd
[root@localhost ~]# ls /root/test3
huawei1.txt  huawei2.txt  passwd
[root@localhost ~]#
```

将压缩文件 passwd.zip 在指定目录/root/test3 下解压缩，如果已有相同的文件存在，要求 unzip 命令覆盖原先的文件。

```
[root@localhost ~]# unzip -o passwd2.zip -d /root/test3
Archive:  passwd2.zip
  inflating: /root/test3/passwd
[root@localhost ~]# ls /root/test3
huawei1.txt  huawei2.txt  passwd
[root@localhost ~]#
```

步骤 23 tar 命令的使用。

将/root/test3 目录里面的所有文件打包。

Tar 压缩命令中 -c 表示创建一个 tar 包文件，-f 用于指定创建的文件名，注意文件名必须紧跟在 -f 参数之后。你还可以加上 -v 参数以可视的方式输出打包的文件。上面会自动去掉表示绝对路径的 /，你也可以使用 -P 保留绝对路径符。

```
[root@localhost ~]# cd test3
[root@localhost test3]# ls
```

```

huawei1.txt huawei2.txt passwd
[root@localhost test3]# tar -cf tartest.tar *
[root@localhost test3]# ls
huawei1.txt huawei2.txt passwd tartest.tar

```

解压文件。

解包一个文件（-x 参数）到指定路径的已存在目录（-C 参数）：

```

[root@localhost test3]# tar -xvf /root/test3/tartest.tar -C /root/test2
huawei1.txt
huawei2.txt
passwd
[root@localhost test3]# cd /root/test2
[root@localhost test2]# ls
huawei1.txt huawei2.txt passwd test1

```

## 步骤 24 gzip 命令的使用。

使用 gzip 工具创建 \*.tar.gz 压缩文件文件。

```

[root@localhost ~]# tar -czvf gziptest.tar.gz /root/test2/
tar: 从成员名中删除开头的 "/"
/root/test2/
/root/test2/passwd
/root/test2/huawei1.txt
/root/test2/test1/
/root/test2/test1/huawei.txt
/root/test2/huawei2.txt
[root@localhost ~]# ls
anaconda-ks.cfg passwd passwd2.zip test2
gziptest.tar.gz passwd1.zip passwd.zip test3

```

解压 \*.tar.gz 文件。

```

[root@localhost ~]# ls
anaconda-ks.cfg passwd passwd2.zip test2
gziptest.tar.gz passwd1.zip passwd.zip test3
[root@localhost ~]# ls /root/test2
huawei1.txt huawei2.txt passwd test1
[root@localhost ~]# rm -rf /root/test2/*
[root@localhost ~]# ls /root/test2
[root@localhost ~]# tar -zxvf gziptest.tar.gz -C ~/test2/
root/test2/
root/test2/passwd
root/test2/huawei1.txt
root/test2/test1/
root/test2/test1/huawei.txt
root/test2/huawei2.txt
[root@localhost ~]# ls /root/test2
root

```

其它格式的压缩包解压的命令对应关系如下：

- 1、\*.tar 用 tar -xvf 解压
- 2、\*.gz 用 gzip -d 或者 gunzip 解压
- 3、.tar.gz 和 .tgz 用 tar -xzf 解压
- 4、\*.bz2 用 bzip2 -d 或者用 bunzip2 解压
- 5、\*.tar.bz2 用 tar -xjf 解压

- 6、\*.Z 用 uncompress 解压
- 7、\*.tar.Z 用 tar -xZf 解压
- 8、\*.rar 用 unrar e 解压
- 9、\*.zip 用 unzip 解压

## 2.5 帮助命令

步骤 25 help 命令的使用。

```
[root@localhost ~]# help pwd
pwd: pwd [-LP]

Print the name of the current working directory.

Options:
  -L          print the value of $PWD if it names the current working
              directory
  -P          print the physical directory, without any symbolic links

By default, `pwd' behaves as if `-L' were specified.

Exit Status:
Returns 0 unless an invalid option is given or the current directory
cannot be read.
[root@localhost ~]# help -d pwd
pwd - Print the name of the current working directory.
[root@localhost ~]# help -s pwd
pwd: pwd [-LP]
```

## 2.6 其它常见命令

步骤 26 last 命令显示用户最近登录信息。

```
[root@localhost ~]# last
root      pts/0          172.19.130.137    Wed Jul  8 14:06    still logged in
root      pts/0          172.19.130.137    Wed Jul  8 11:45 - 13:43  (01:58)
root      tty1                    Wed Jul  8 11:23    still logged in
reboot    system boot    4.19.90-2003.4.0 Wed Jul  8 11:19    still running
```

步骤 27 history 命令查看历史命令。

```
[root@localhost ~]# history
1  ls -l
```

步骤 28 tab 自动补齐命令。

当在输入命令时可以使用 tab 键自动补齐命令，文件路径等。例如，输入 wh 键入 tab 键之后，就会给出以下提示。

```
[root@localhost ~]# wh
whatis  whereis  which   while    whiptail  who      whoami
```

步骤 29 uptime 命令查看系统负载。

```
[root@localhost ~]# uptime
14:21:42 up 3:02, 2 users, load average: 0.00, 0.00, 0.00
```

步骤 30 date 可以用来显示或设定系统的日期与时间。

```
[root@localhost ~]# date
Wed Jul  8 14:23:34 CST 2020
[root@localhost ~]# date '+%c'
Wed 08 Jul 2020 02:24:02 PM CST
[root@localhost ~]# date '+%D'
07/08/20
[root@localhost ~]# date '+%x'
07/08/2020
[root@localhost ~]#
```

步骤 31 wget 命令用来从指定的 URL 下载文件。

注意使用 wget 命令的主机需要能够访问 Internet 网络。

```
[root@localhost ~]# wget https://wordpress.org/latest.zip
```

## 思考

- 1、创建一个/iamthebest 目录；
- 2、在/iamthebest 目录下创建/cat 和/dog 两个目录；
- 3、将/etc/passwd 文件复制到/iamthebest 目录；并查看被复制文件的操作权限；
- 4、尝试执行 `cp -i /etc/passwd .` 你会发现什么？为什么会出现这个状况？
- 5、将 passwd 重命名为 fun；
- 6、将 fun 文件移动到 cat 目录，然后再从 cat 目录移动到 dog 目录，最后在移动到/iamthebest 目录。
- 7、将 fun 文件硬链接到/cat 目录；
- 8、将 fun 文件软链接到/dog 目录；
- 9、删除/iamthebest 文件中的 fun 文件；
- 10、查看所有 fun 文件的节点信息；
- 11、找到 fun 文件，并将 fun 文件打包到/iamthebest 目录，并打包成\*.tar.gz 格式，命名为 iamstillfun.tar.gz；
- 12、将 iamstillfun.tar.gz 压缩包解压到 cat 目录下，并命名为 fun；
- 13、查找并显示出所有 fun 文件的位置。

