1. Linux命令基本格式
2. 命令提示符

[root@localhost ~]#

1. root

显示当前登录的用户。

1. @

分隔符号，无特殊含义。

1. localhost

当前系统的简写主机名。

1. ~

用户当前所在的目录，~表示在用户的家目录中。

1. #

超级用户是#，普通用户时$。

1. 命令基本格式

命令 [选项] [参数]

1. 选项

选项用于更改命令的功能，以ls命令举例：

1. -a :

显示所有的文件。

1. -d

显示目录信息，而不是目录下的文件。

1. -h

人性化显示，用KB，MB单位显示文件的大小。

1. -i

显示文件的i节点号，在分区时，每个文件都被分配了i节点号作为索引。

1. -l

长格式显示所有的文件和文件夹。如下所示：

-rw-------. 1 root root 1207 1月 14 18:18 aa.txt

第一列为文件或文件夹的权限；

第二列为引用计数，对于文件来说是文件的硬链接个数，对于文件夹来说是该目录下有多少个一级子目录；

第三列为该文件的建立用户；

第四列为该文件所属用户的所属组；

第五列为文件的大小，默认单位为字节，可以通过-h选项修改为人性化的单位；

第六列为文件修改时间；

第七列为文件名；

1. 目录操作命令
2. cd 命令

切换所在目录的命令（change directory），是Shell内置的命令，所有用户都可以执行。有简化的用法如下所示：

1. ~

代表用户的家目录。

1. –

代表上次所在的目录。

1. .

代表当前的目录。

1. ..

代表上级目录。

1. pwd命令

查询所在目录的命令（print name of current/working directory），命令路径在/bin/pwd中，所有用户都能够查询。

1. mkdir命令

创建目录的命令（make directories），命令路径在/bin/mkdir中，所有用户都能够创建空目录。注意，不能在同一个目录下创建同名的文件和目录！！！

-p 选项能够递归创建目录，如下所示：

mkdir -p 123/124/125

1. rm命令

删除文件或者删除文件夹，如下所示：

-r选项递归删除文件夹(recursive)

rm -r 123

-f 选项强制删除，不会询问(force)

rm -rf 123

1. 文件操作命令
2. touch 命令

创建空文件或者修改文件时间（change file timestamps），在/bin/touch中，所有用户都可以修改文件的时间戳。touch命令对于不存在的文件是创建，对于存在的文件就是修改该文件的状态时间了。

1. stat命令

查看文件详细信息（display file or file system status），在/usr/bin/stat中，所有用户都可以查询。

该命令可以看到文件的大小，占了几块block，每块block多大，是什么类型的文件，硬件设备号是什么，索引I节点号是什么，有几个硬链接，权限是什么，所属用户是哪位，所属组是哪个，最近一次访问该文件的时间是什么时候，最近一个修改该文件的内容是什么时候，最近一次改动文件的状态是什么时候。所谓的状态是指修改文件的权限等非内容参数。如下所示：

[root@localhost ~]# stat a

File: "a"

Size: 0 Blocks: 0 IO Block: 4096 普通空文件

Device: 802h/2050d Inode: 5744 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2020-12-06 04:32:07.925998031 +0800

Modify: 2020-12-06 04:32:07.925998031 +0800

Change: 2020-12-06 04:32:07.925998031 +0800

1. cat命令

查看文件内容（concatenate files and print on the standard output），在/bin/cat中，所有用户都可查看。有如下常用的选项：

1. -A

相当于-vET选项的整合，用于列出所有的隐藏符号。

1. -E

列出每行结尾的回车符$

1. -n

显示行号。

1. -T

把Tab键用^I显示出来。

1. -v

列出特殊字符。

1. more命令

cat命令对于内容较少的文件比较好用，但是当文件过大时，显示的行数也是有限制的，这里需要more命令分屏显示文件。

分屏显示文件（file perusal filter for crt viewin），在/bin/more中，所有用户都可使用。可以使用如下的命令进行交互：

1. 空格

向下翻页。

1. b

向上翻页。

1. 回车键

向下滚动一行。

1. /字符串

搜索字符串。

1. q

退出。

1. less命令

分行显示命令，在/usr/bin/less中。

1. head命令

显示文件开头的命令（output the first part of files），在/usr/bin/head中，所有用户可用。有如下的选项：

1. -n 行数

从文件头开始，显示指定的行数。

1. -v

显示文件名

1. tail命令

显示文件结尾（output the last part of files），在/usr/bin/tail中，所有用户可用。

1. ln命令
2. 分区原理

对于系统的分区，分为两块，一块是I节点，一块是block块。每个新建的文件都会有块和对应的I节点，I节点中保存着I节点号，时间，权限和block块的保存位置，而block块中保存着文件中的信息。可以看出，该文件的I节点和block块中并没有保存文件名，文件名保存在该文件所在目录的block块中。例如该文件保存在/root目录下，则/root目录的block块中保存的就是该目录下的所有文件以及子目录的所有文件的文件名和对应的I节点号。那么，我们要找到该文件的信息，就要首先找到该目录的文件名，这样就会一直走到根节点/中，那么最终就要先找到根节点/的I节点号，才能找到其他文件的信息。/节点的I节点号是2, 1在内核加载的过程中被占用了，所有I节点号都是从1开始的。

1. 硬链接

一个文件的信息保存在block块中，由对应的I节点提供索引，因此当我们创建一个文件时，仅仅是创建了一个硬链接保存了该文件block块所对应的I节点号而已，我们通过ln命令创建硬链接，将源文件和目标文件链接起来，所能达到的效果也就是在目标文件中存储物理文件所在block块的I节点号，用于索引而已。

因此，硬链接的各种特征就很好解释了：

1. 源文件和硬链接文件拥有相同的I节点和block块。
2. 修改任意一个文件，就是修改源文件的block块，所有硬链接的文件都会改变。
3. 删除任意一个文件，仅仅是删除了一个硬链接而已，源文件的block块不会被删除，因此其他的硬链接能够照常使用。

硬链接的缺陷：

1. 很难确定硬链接的文件位置。
2. 硬链接不能链接目录，因为目录的层级过多，链接的话过于复杂。
3. 硬链接不能够跨分区，因为硬链接是通过指向同一个分区的I节点号来链接的，不同分区的无法链接。
4. 软链接

软链接就相当于Windows中的快捷方式，该软链接的block块中仅仅保存源文件的I节点号，并没有实际保存数据。因此，软链接的特征如下：

1. 软链接和源文件用于不同的I节点和block块。
2. 修改软链接或者源文件，另一个都会被改变。
3. 删除源文件，软链接就无法找到源文件了，无法继续使用。
4. 软链接没有实际数据，仅保存源文件的I节点号，大小不会变化。
5. 软链接的权限是最大的，但是实际访问源文件还是要根据源文件权限决定。
6. 软链接可以链接目录，因为只需要记录目录的I节点号即可。
7. 软链接可以跨分区，但是注意使用软链接时要使用绝对路径，否则可能会找不到源文件。

使用方式：

使用ln 命令， 添加-s选项（soft）

ln -s /root/a /tmp/b