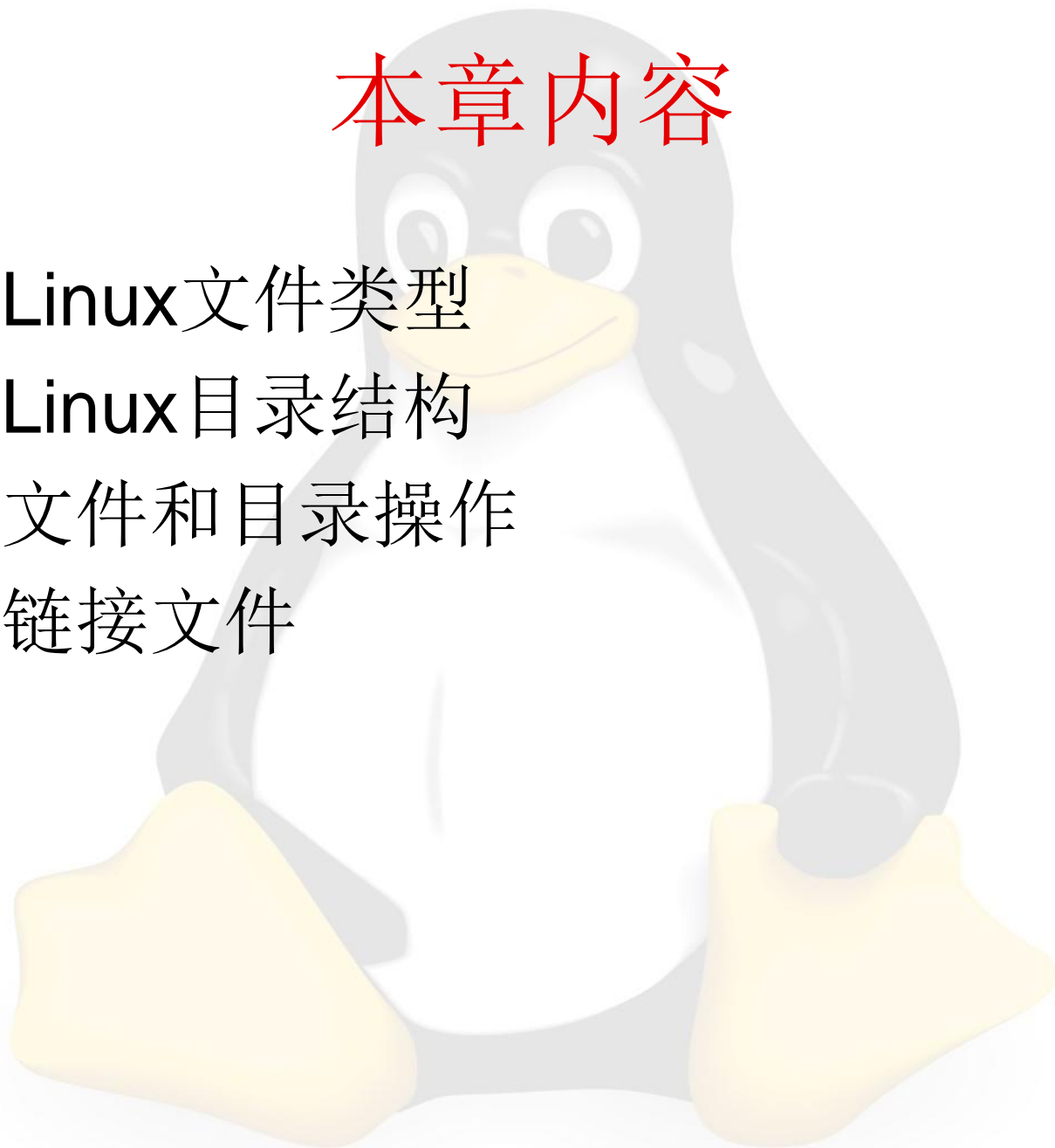


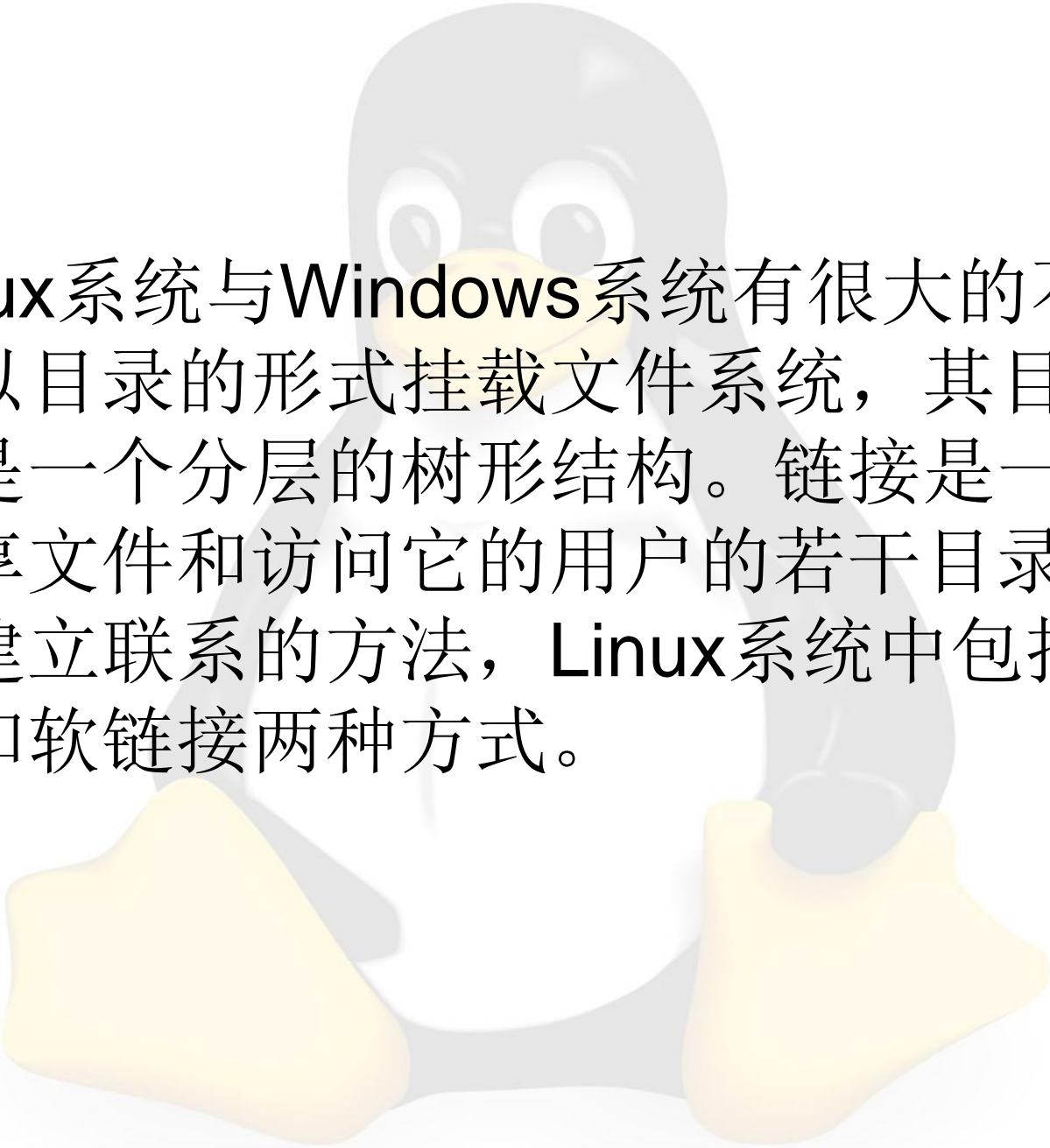
# 第4章 目录和文件管理



# 本章内容

- 4.1 Linux文件类型
- 4.2 Linux目录结构
- 4.3 文件和目录操作
- 4.4 链接文件

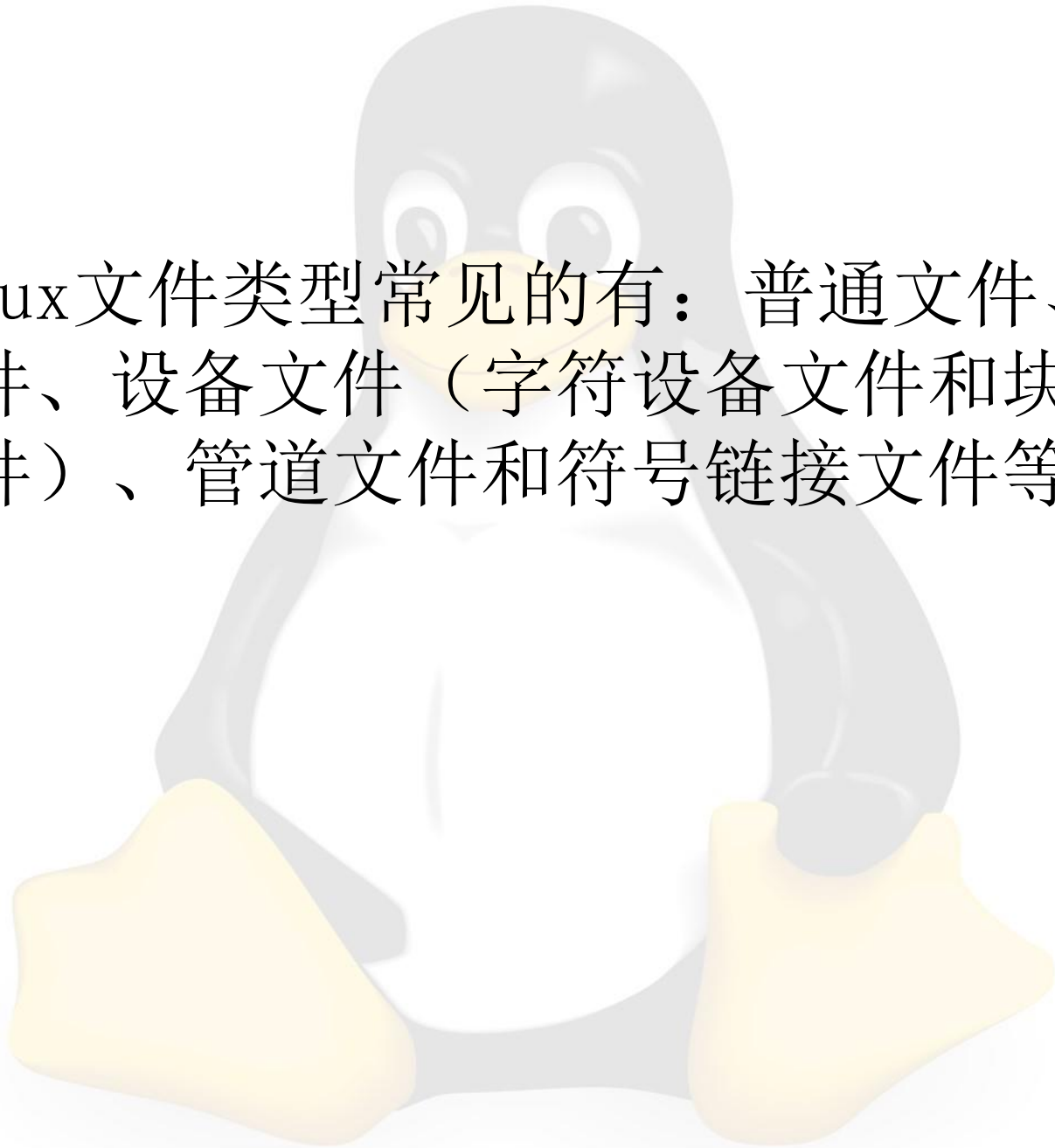


- 
- **Linux系统与Windows系统有很大的不同，它以目录的形式挂载文件系统，其目录结构是一个分层的树形结构。链接是一种在共享文件和访问它的用户的若干目录项之间建立联系的方法，Linux系统中包括硬链接和软链接两种方式。**

## 4.1 Linux文件类型

- 在Linux系统中除了一般文件之外，所有的目录和设备（如光驱、硬盘等）都是以文件的形式存在的。Linux文件类型和Linux文件的文件名所代表的意义是两个不同的概念。
- 通过一般应用程序创建的文件，比如file.txt、file.tar.gz，这些文件虽然要用不同的程序来打开，但放在Linux文件类型中衡量的话，大多称之为普通文件。

- Linux文件类型常见的有：普通文件、目录文件、设备文件（字符设备文件和块设备文件）、管道文件和符号链接文件等。



# 1. 普通文件

- 用“ls -lh”命令查看某个文件的属性，可以看到有类似“-rw-----”的属性符号，其属性第一个符号是“-”，这样的文件在Linux系统中就是普通文件。这些文件一般是用一些相关的应用程序创建，比如图像工具、文档工具或归档工具等。

## 2. 目录文件

- 当在某个目录下执行“ls -lh”命令，看到有类似“drwxr-xr-x”的属性符号，其属性第一个符号是“d”，这样的文件在Linux系统中就是目录文件。

# 3. 设备文件

- Linux系统中的/dev目录中有大量的设备文件，主要是块设备文件和字符设备文件。

## (1) 块设备文件

块设备的主要特点是可以随机读写，而最常见的块设备就是磁盘，如/dev/hda1、/dev/sda1等。用“ls -l”命令查看某个文件的属性，可以看到有类似“brw-rw---”的属性符号，其属性第一个符号是“b”，这样的文件在Linux系统中就是块设备文件。



## (2) 字符设备文件

最常见的字符设备文件是打印机和终端，可以接收字符流。`/dev/null`是一个非常实用的字符设备文件，送入这个设备的所有内容都被忽略。用“`ls -l`”命令查看某个文件的属性，可以看到有类似“`crw--w---`”的属性符号，其属性第一个符号是“`c`”，这样的文件在Linux系统中就是字符设备文件。

## 4. 管道文件

- 管道文件有时候也被叫做FIFO文件（FIFO是先进先出的意思），管道文件就是从一头流入，从另一头流出。用“ls -l”命令查看某个文件的属性，可以看到有类似“prw-----”的属性符号，其属性第一个符号是“p”，这样的文件在Linux系统中就是管道文件。

## 5. 链接文件

- 链接文件有两种类型：软链接文件和硬链接文件。

### (1) 软链接文件

- 软链接文件又叫符号链接文件，这个文件包含了另一个文件的路径名。其可以是任意文件或目录，可以链接不同文件系统的文件。在对软链接文件进行读写的时候，系统会自动地把该操作转换为对源文件的操作，但删除软链接文件时，系统仅仅删除软链接文件，而不删除源文件本身。
- 用“ls -l”命令查看某个文件的属性，可以看到有类似“lrwxrwxrwx”的属性符号，其属性第一个符号是“l”，这样的文件在Linux系统中就是软链接文件。

## (2) 硬链接文件

- 硬链接是已存在文件的另一个文件，对硬链接文件进行读写和删除操作时，结果和软链接相同。但如果删除硬链接文件的源文件，硬链接文件仍然存在，而且保留了原有的内容。这时，系统就“忘记”了它曾经是硬链接文件，而把它当成一个普通文件。
- 用“`ls -l`”命令查看某个文件的属性，可以看到第二列的文件硬链接数大于1，这样的文件在Linux系统中就是硬链接文件。

## 4.2 Linux目录结构

- Linux系统都有根文件系统，它包含系统引导和使其它文件系统得以挂载所必要的文件，根文件系统需要有单用户状态所必须的足够的内容，还应该包括修复损坏系统、恢复备份等工具。
- Linux系统的目录结构是分层的树形结构，都是挂载在根文件系统“/”下。

# Linux目录结构（1）

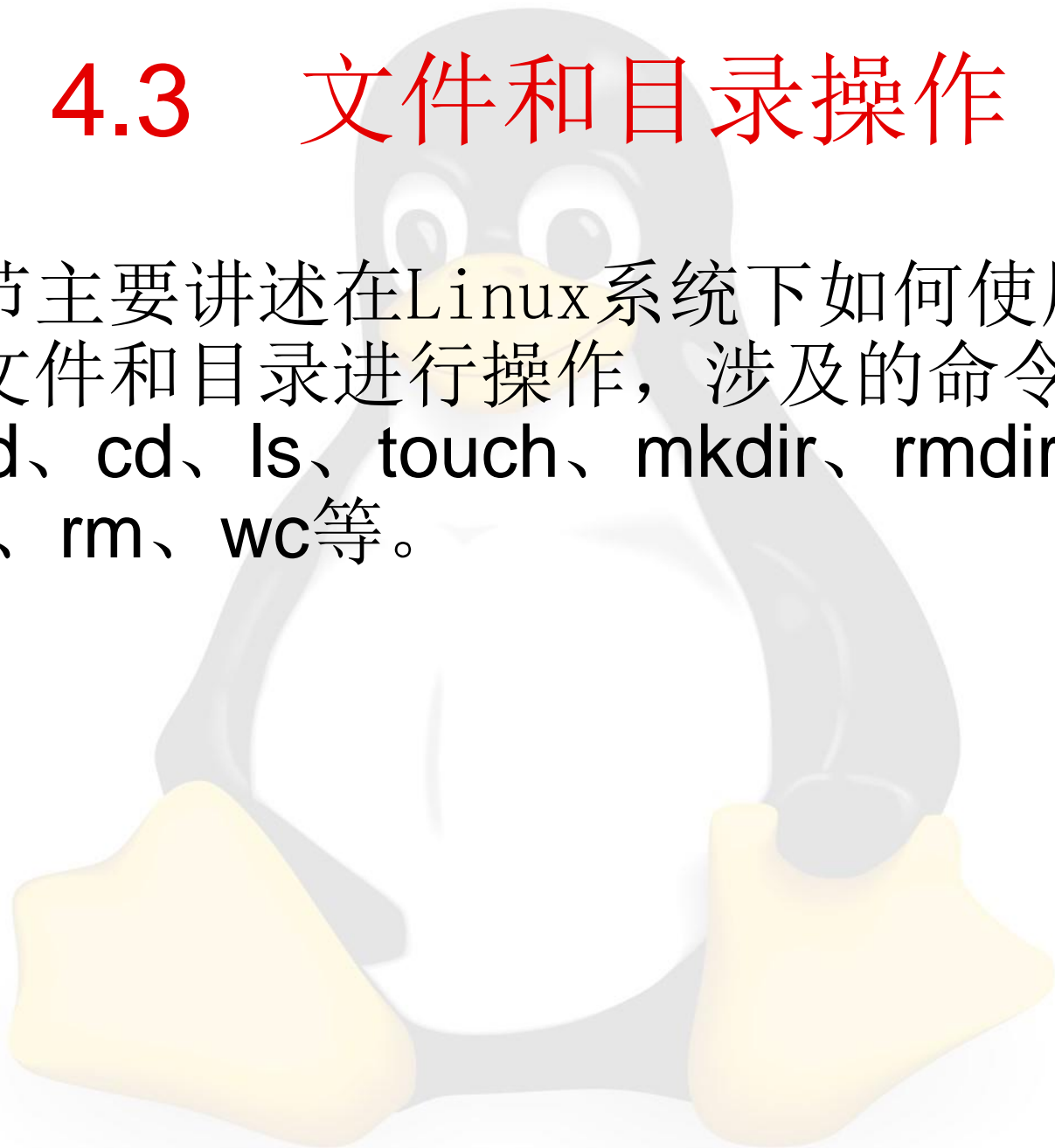
目录	描述
/home	包含Linux系统上各用户的主目录，子目录名称默认以该用户名命名
/root	是root用户的主目录
/bin	包含常用的命令文件，不能包含子目录
/sbin	包含系统管理员和root用户所使用的命令文件
/dev	包含大部分的设备文件，比如磁盘、光驱等
/lib	包含Linux系统的共享文件和内核模块文件 /lib/modules目录存放核心可加载模块
/lib64	包含64位版本Linux系统的共享文件和内核模块文件
/tmp	包含一些临时文件
/mnt	手动为某些设备（比如硬盘）挂载提供挂载目录

# Linux目录结构（2）

目录	描述
/boot	包含Linux系统的内核文件和引导装载程序（如GRUB）文件
/opt	包含某些第三方应用程序的安装文件
/media	由系统自动为某些设备（一般为光盘、U盘等设备）挂载提供挂载目录
/var	该目录存放不经常变化的数据，如系统日志、打印队列、DNS数据库文件等
/etc	包含Linux系统上大部分的配置文件，建议修改配置文件之前先备份
/usr	包含可以供所有用户使用的程序和数据
/srv	存储一些服务启动之后所需要取用的资料目录
/run	一个临时文件系统，一些程序或服务启动以后，会将他们的PID放置在该目录中
/sys	在Linux系统提供热插拔能力的同时，该目录包含所检测到的硬件设置，它们被转换成/dev目录中的设备文件
/proc	<p>是一个虚拟的文件系统，它不存在磁盘上，而是由内核在内存中产生，用于提供系统的相关信息。</p> <p>下面说明在/proc目录下的一些最重要的文件。</p> <p>/proc/cpuinfo：该文件保存计算机CPU信息。</p> <p>/proc/filesystems：该文件保存Linux文件系统信息。</p> <p>/proc/ioports：该文件保存计算机I/O端口号信息。</p> <p>/proc/version：该文件保存Linux系统版本信息。</p> <p>/proc/meminfo：该文件保存计算机内存信息。</p>

## 4.3 文件和目录操作

- 本节主要讲述在Linux系统下如何使用命令对文件和目录进行操作，涉及的命令有 `pwd`、`cd`、`ls`、`touch`、`mkdir`、`rmdir`、`cp`、`mv`、`rm`、`wc`等。





# pwd: 显示工作目录路径

- 显示当前用户所处的工作目录的绝对路径。

命令语法:

**pwd [选项]**

**【例4.1】** 显示用户当前工作目录路径。

```
[root@rhel sysconfig]# pwd
```

```
/etc/sysconfig
```

//用户当前工作目录路径是/etc/sysconfig

# cd: 更改工作目录路径

- 更改用户的工作目录路径。工作目录路径可以使用绝对路径名或相对路径名，绝对路径从/（根）开始，然后循序到所需的目录下，相对路径从当前目录开始。

命令语法：

**cd** [选项][目录]



【例4.2】 更改用户工作目录路径为/etc。

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

```
[root@rhel etc]# pwd
```

```
/etc
```

//查看当前目录，已经更改为/etc了

【例4.3】 更改目录位置至当前目录的父目录。

```
[root@rhel etc]# pwd
```

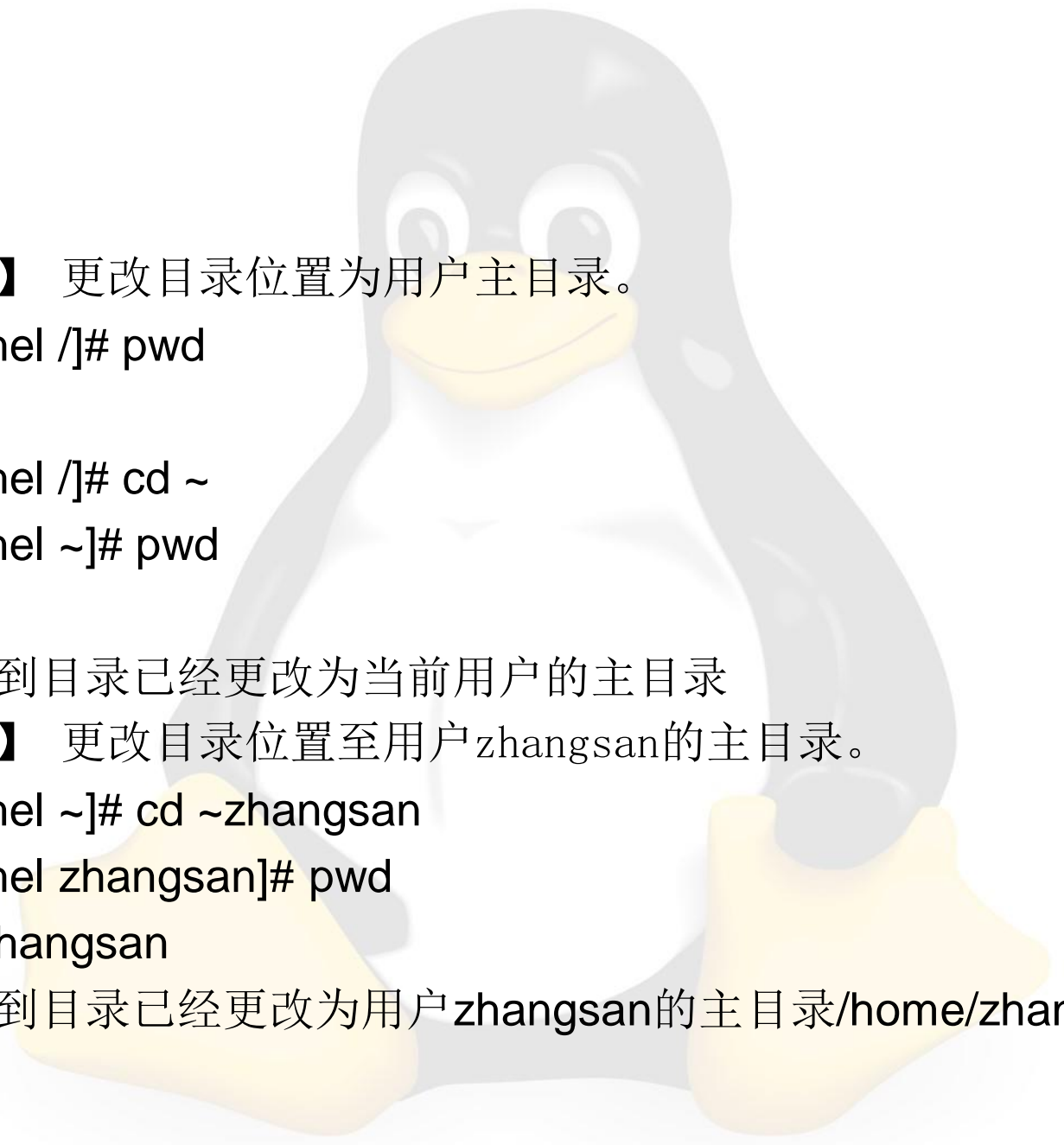
```
/etc
```

```
[root@rhel etc]# cd ..
```

```
[root@rhel /]# pwd
```

```
/
```

//可以看到目录已经更改为当前目录的父级目录“/”了



【例4.4】 更改目录位置为用户主目录。

```
[root@rhel /]# pwd
```

```
/
```

```
[root@rhel /]# cd ~
```

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

```
/root
```

//可以看到目录已经更改为当前用户的主目录

【例4.5】 更改目录位置至用户zhangsan的主目录。

```
[root@rhel ~]# cd ~zhangsan
```

```
[root@rhel zhangsan]# pwd
```

```
/home/zhangsan
```

//可以看到目录已经更改为用户zhangsan的主目录/home/zhangsan

# ls: 列出目录和文件信息

- 对于目录而言将列出其中的所有子目录与文件信息；对于文件而言将输出其文件名以及所要求的其它信息。

命令语法:

**ls** [选项] [目录|文件]

# ls命令显示的详细信息

列 数	描 述
第1列	第1个字符表示文件的类型 第2~4个字符表示文件的用户所有者对此文件的访问权限 第5~7个字符表示文件的组群所有者对此文件的访问权限 第8~10个字符表示其他用户对此文件的访问权限
第2列	文件的链接数
第3列	文件的用户所有者
第4列	文件的组群所有者
第5列	文件长度（也就是文件大小，不是文件的磁盘占用量）
第6~8列	文件的更改时间（mtime）， 或者是文件的最后访问时间（atime）
第9列	文件名称



**【例4.7】** 显示/root目录下所有文件目录的详细信息，包括隐藏文件。

```
[root@rhel ~]# ls -al /root
```

总用量 404

```
dr-xr-x---. 24 root root  4096 6月  3 06:00 .
dr-xr-xr-x. 18 root root  4096 6月  2 23:05 ..
drwxr-xr-x.  2 root root  4096 6月  3 05:29 .abrt
-rw-----.  1 root root 10670 6月  3 01:17 anaconda-ks.cfg
-rw-----.  1 root root   551 6月  3 05:55 .bash_history
-rw-r--r--.  1 root root    18 1月 15 05:25 .bash_logout
-rw-r--r--.  1 root root   176 1月 15 05:25 .bash_profile
-rw-r--r--.  1 root root   176 1月 15 05:25 .bashrc
```

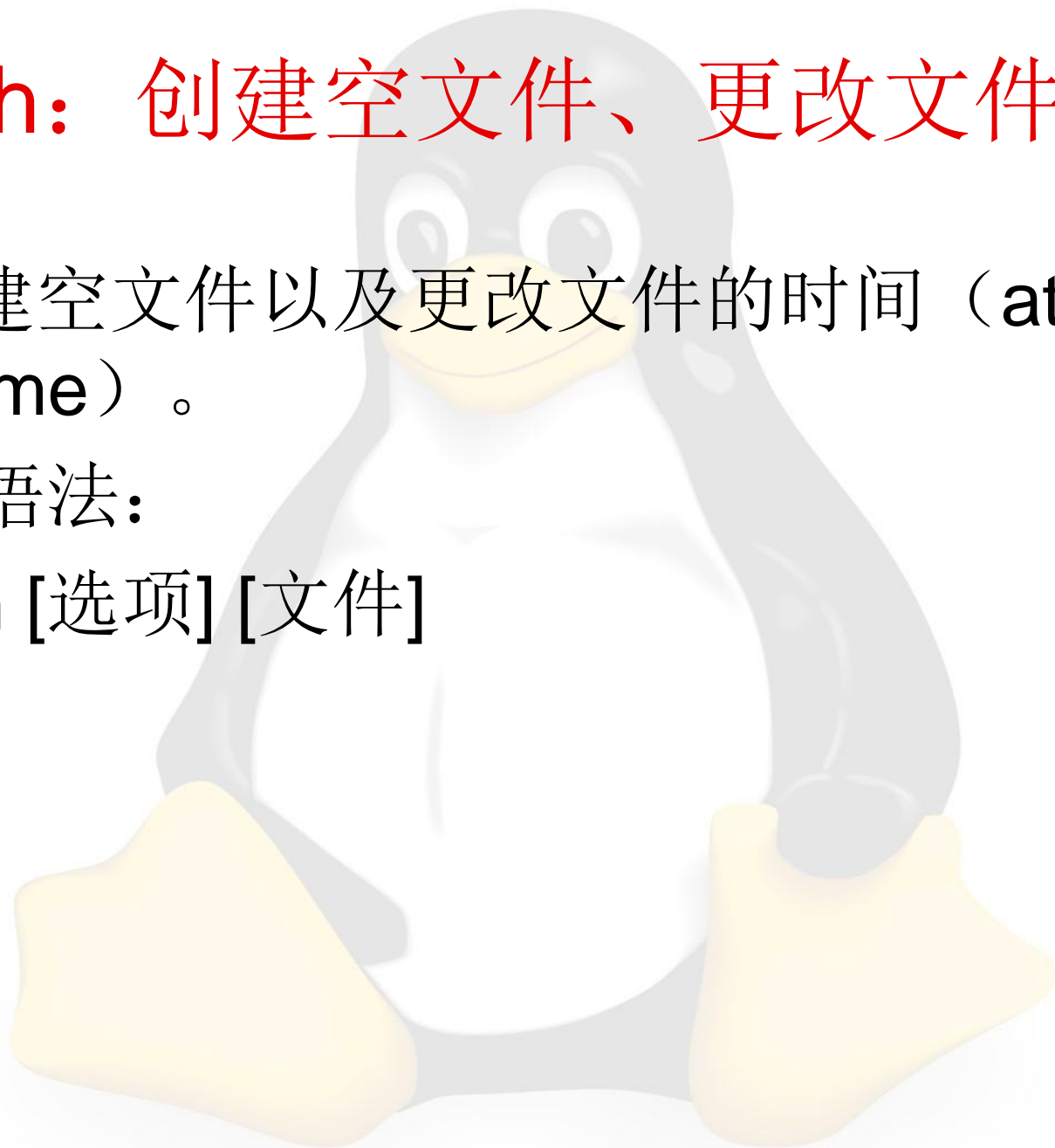
.....

# touch: 创建空文件、更改文件时间

- 创建空文件以及更改文件的时间（**atime**和**mtime**）。

命令语法:

**touch** [选项] [文件]





【例4.9】 创建空文件file，file1和file2。

```
[root@rhel ~]# touch file1
```

```
[root@rhel ~]# touch file2 file3
```

```
[root@rhel ~]# ls -l file1 file2 file3
```

```
-rw-r--r--. 1 root root 0 6月  3 05:45 file1
```

```
-rw-r--r--. 1 root root 0 6月  3 05:45 file2
```

```
-rw-r--r--. 1 root root 0 6月  3 05:45 file3
```

//file1，file2，file3这3个都是空文件，文件内没有任何数据



**【例4.10】** 将文件file1的时间记录改为6月7日19点30分，时间格式为MMDDHHmm。

```
[root@rhel ~]# ls -l file1
```

```
-rw-r--r--. 1 root root 0  6月  3 05:45 file1
```

//空文件file1其创建日期为6月3日5:45

```
[root@rhel ~]# touch -c -t 06071930 file1
```

```
[root@rhel ~]# ls -l file1
```

```
-rw-r--r--. 1 root root 0 6月  7 19:30 file1
```

//可以看到文件file1现在的时间已经改为6月7日19点30分

# mkdir: 创建目录

- 在Linux系统中创建目录。

命令语法:

`mkdir -p a/b/c`

**mkdir** [选项] [目录]

【例4.11】 创建目录newdir1，其默认权限为755。

```
[root@rhel ~]# mkdir newdir1
```

```
[root@rhel ~]# ls -ld
```

```
drwxr-xr-x. 2 root root 4096 6月  3 05:46 newdir1
```

//目录newdir1的权限为rwxr-xr-x（755）

# rmmdir: 删除空目录

- 在Linux系统中删除空目录。

命令语法:

**rmmdir [选项] [目录]**

【例4.13】 删除空目录newdir1。


```
[root@rhel ~]# rmmdir newdir1
```

# cp: 复制文件和目录

- 复制文件和目录到其它目录中。如果同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把它前面指定的所有文件或目录复制到该目录中。若同时指定多个文件或目录，而最后的目的地并非是一个已存在的目录，则会出现错误信息。

命令语法:

**cp** [选项] [源文件|目录] [目标文件|目录]



**【例4.15】** 将/etc/grub2.cfg文件复制到/root目录下，并改名为grub。

```
[root@rhel ~]# cp /etc/grub2.cfg /root/grub
```

**【例4.16】** 将文件/etc/grub2.cfg复制到/root目录下。

```
[root@rhel ~]# cp /etc/grub2.cfg /root
```

**【例4.17】** 将/boot目录中的所有文件及其子目录复制到目录/root中。

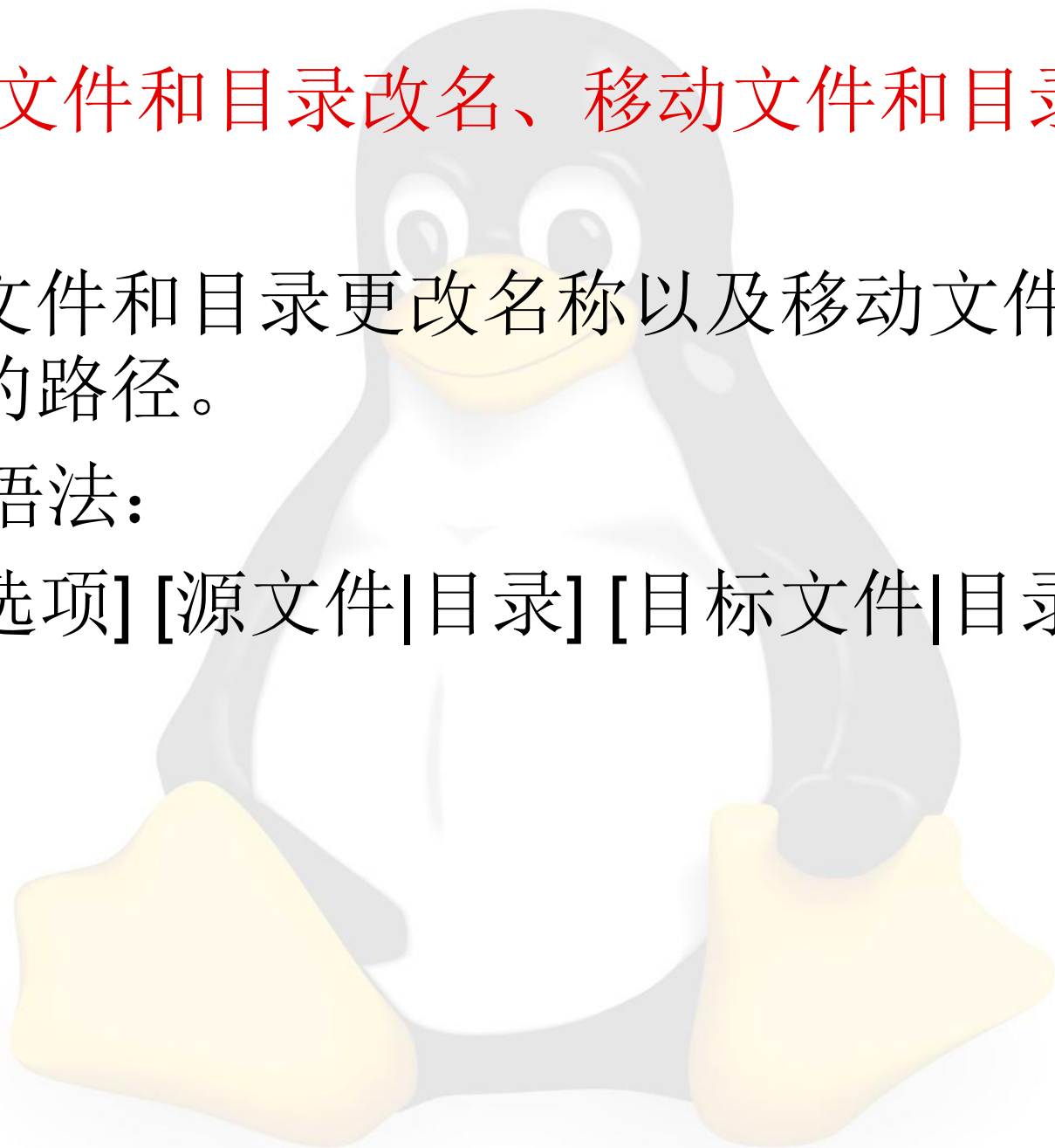
```
[root@rhel ~]# cp -r /boot /root
```


# mv: 文件和目录改名、移动文件和目录路径

- 对文件和目录更改名称以及移动文件和目录的路径。

命令语法:

**mv** [选项] [源文件|目录] [目标文件|目录]





**【例4.18】** 将/root/picture目录下所有的后缀名为“.png”的文件移到/usr/local/share/picture目录下。

```
[root@rhel ~]# mv -f /root/picture/*.png  
/usr/local/share/picture
```

**【例4.19】** 把/root/picture目录下的文件kdepic.png改名为life.png。

```
[root@rhel ~]# mv /root/picture/kdepic.png  
/root/picture/life.png
```

**【例4.20】** 把目录/root/pitcure名称更改为/root/mypicture。

```
[root@rhel ~]# mv /root/picture /root/mypicture
```

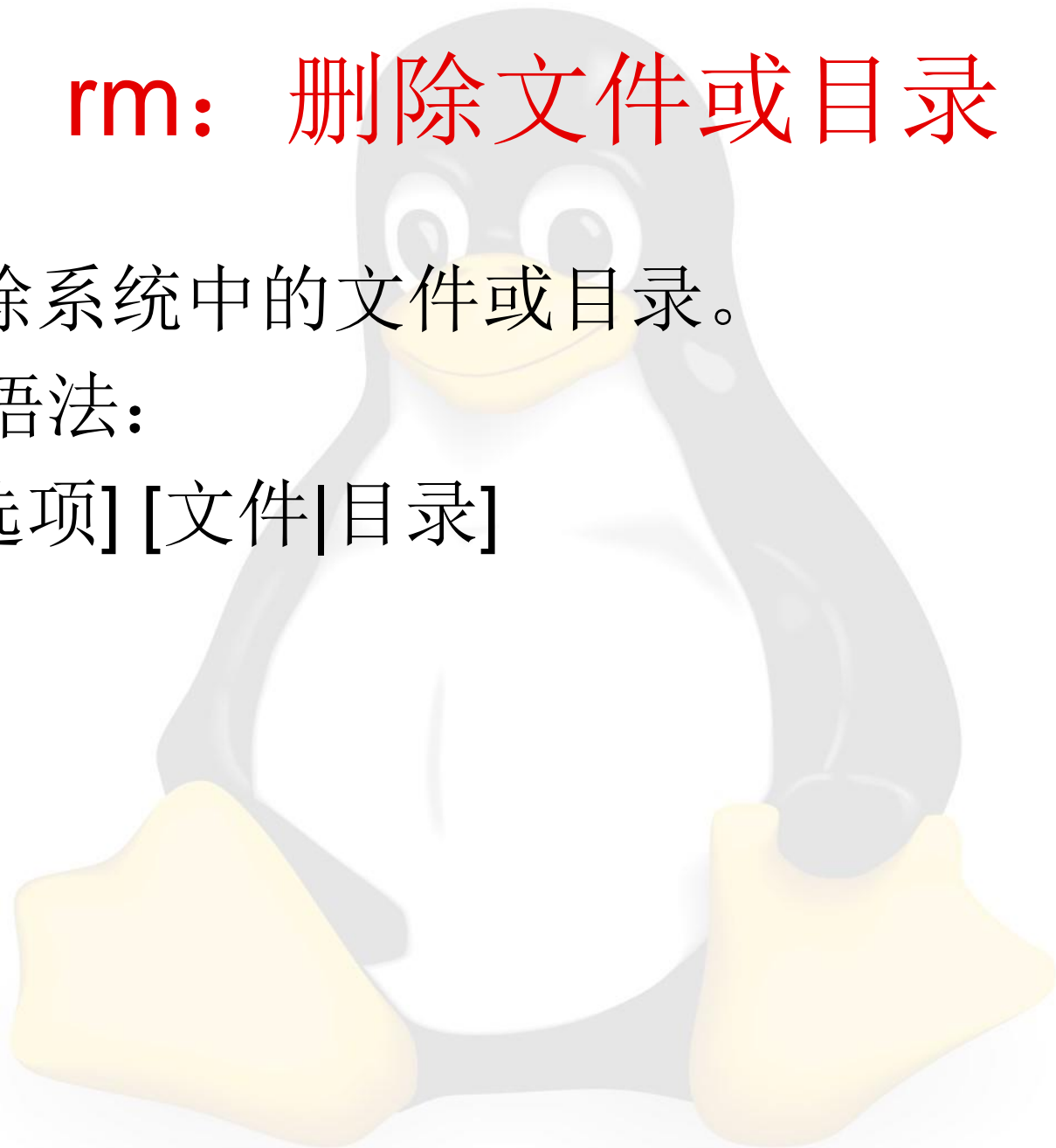


# rm: 删除文件或目录

- 删除系统中的文件或目录。

命令语法:

rm [选项] [文件|目录]





【例4. 21】 删除当前目录下的file4文件。

```
[root@rhel ~]#rm file4
```

rm: 是否删除普通文件 “file4” ? y//输入y确认删除该文件

【例4. 22】 连同文件/root/ab/a和目录/root/ab一起删除。

```
[root@rhel ~]# mkdir /root/ab
```

```
[root@rhel ~]# touch /root/ab/a
```

//创建目录/root/ab和文件/root/ab/a

```
[root@rhel ~]# rm -rf /root/ab
```

//连同文件/root/ab/a和目录/root/ab一起删除

## WC: 统计文件行数、单词数和字节数和字符数

- 统计指定文件的行数、单词数、字节数和字符数，并将统计结果显示输出到屏幕。如果没有给出文件名，则从标准输入读取。**wc**同时也给出所有指定文件的总统计数。单词是由空格字符区分开的最大字符串。输出列的顺序和数目不受选项的顺序和数目的影响。总是按行数、单词数、字节数、文件的顺序显示每项信息。

命令语法:

**wc** [选项] [文件]

【例4.23】 统计/root/aa文件的行数、单词数和字符数。

```
[root@rhel ~]#cat /root/aa
```

```
a b
```

```
c de f
```

```
中国 g h
```

//查看/root/aa文件内容

```
[root@rhel ~]#wc /root/aa
```

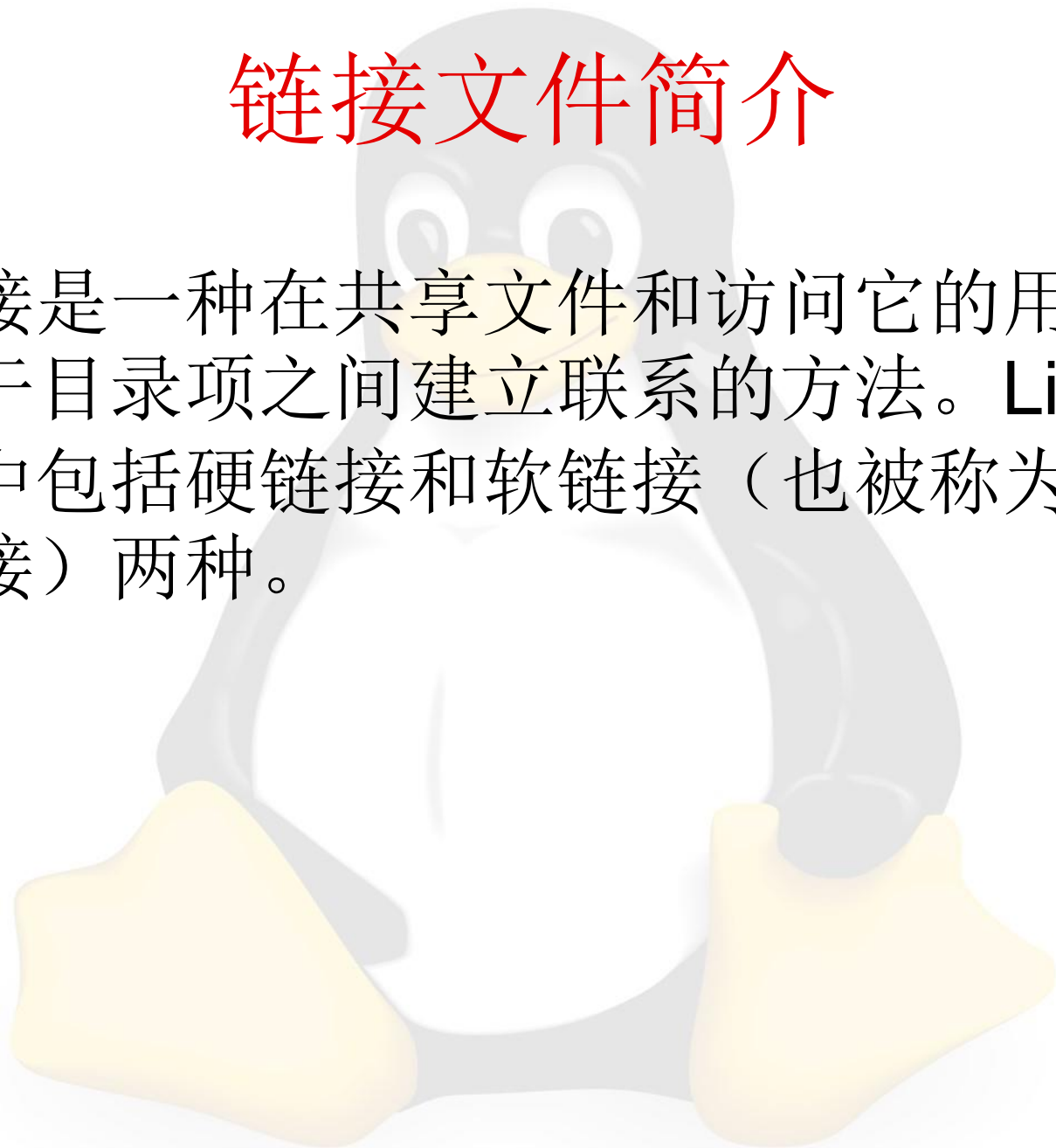
```
3 8 22 /root/aa
```

## 4.4 链接文件

- 在Linux系统中，内核为每一个新创建的文件分配一个inode（索引节点）号，文件属性保存在索引节点里，在访问文件时，索引节点被复制到内存里，从而实现文件的快速访问。

# 链接文件简介

- 链接是一种在共享文件和访问它的用户的若干目录项之间建立联系的方法。Linux系统中包括硬链接和软链接（也被称为符号链接）两种。

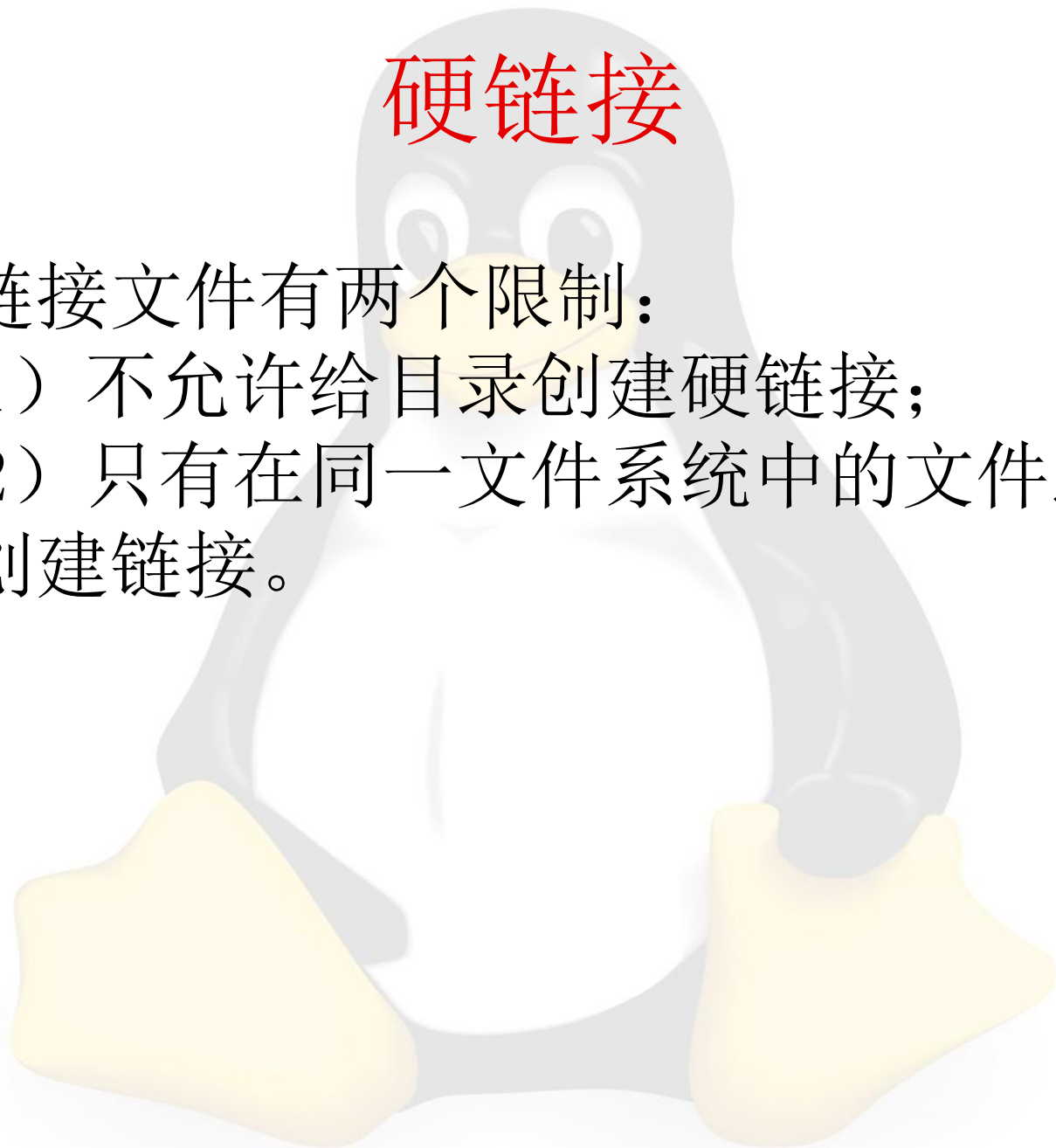


# 硬链接

- 硬链接是一个指针，指向文件inode，系统并不为它重新分配inode，两文件具有相同的inode。硬链接节省空间，也是Linux系统整合文件系统的传统方式。

# 硬链接

- 硬链接文件有两个限制：
  - (1) 不允许给目录创建硬链接；
  - (2) 只有在同一文件系统中的文件之间才能创建链接。





# 软链接

- 软链接也叫**符号链接**，这个文件包含了另一个文件的路径名。可以是任意文件或目录，可以链接不同文件系统的文件，和Windows下的快捷方式相似。链接文件甚至可以链接不存在的文件，这就产生一般称之为“断链”的问题，链接文件甚至可以循环链接自己。

# 硬链接和软链接的区别

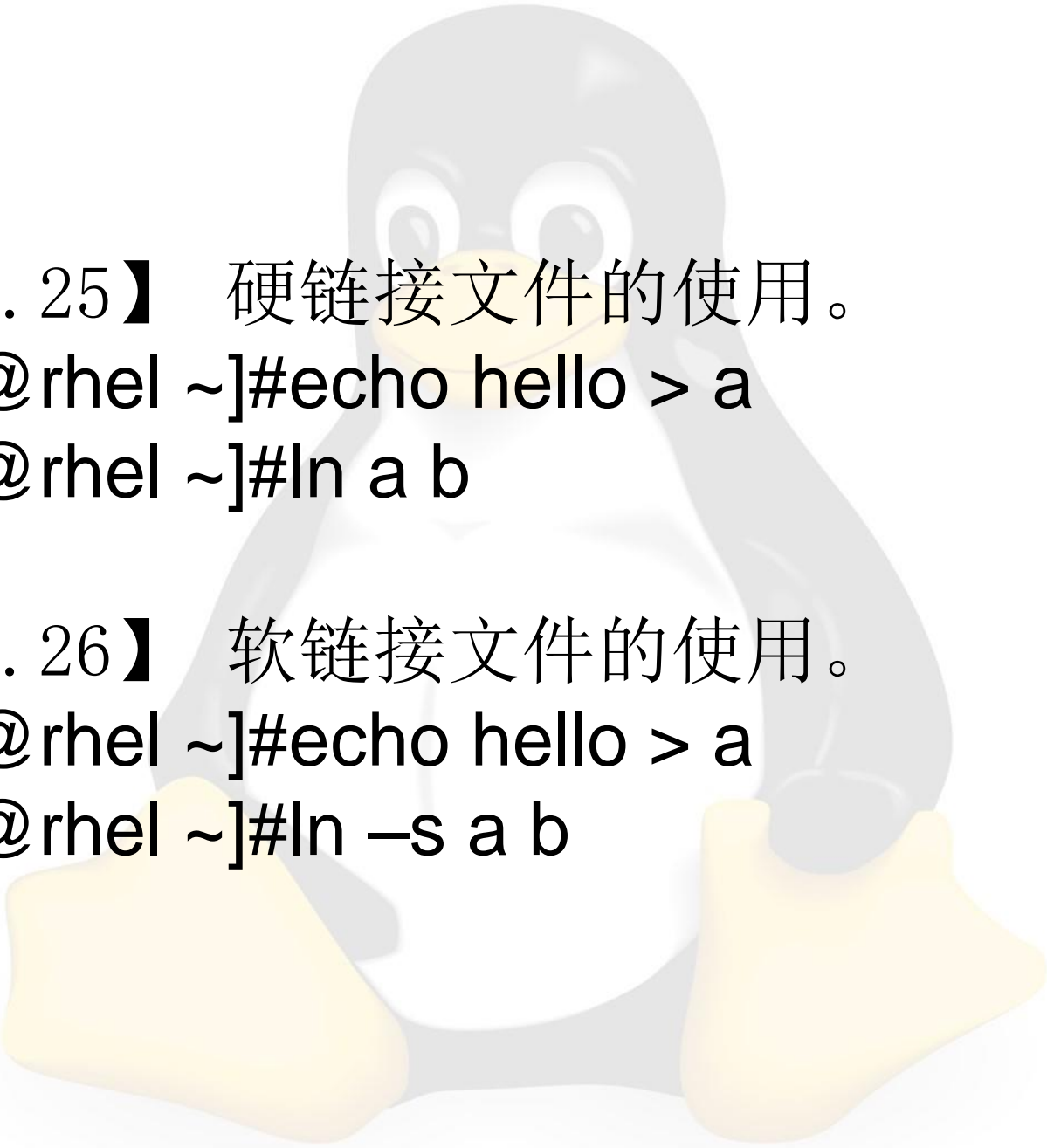
- 硬链接记录的是目标的**inode**，软链接记录的是目标的路径。软链接就像是快捷方式，而硬链接就像是备份。软链接可以做跨分区的链接，而硬链接由于**inode**的缘故，只能在本分区中做链接。所以软链接的使用频率要高得多。

# 创建和使用链接文件

- 使用ln命令可以创建链接文件（包括软链接文件和硬链接文件）。

命令语法：

**ln [选项] [源文件名] [链接文件名]**



**【例4. 25】** 硬链接文件的使用。  
[root@rhel ~]#echo hello > a  
[root@rhel ~]#ln a b

**【例4. 26】** 软链接文件的使用。  
[root@rhel ~]#echo hello > a  
[root@rhel ~]#ln -s a b