GNU/Linux
备份与压缩

文件 / 目录打包 (归档)-tar

命令 :tar

功能 : 将文件 / 目录打包

语法格式 :tar < 选项 > < 打包文件名 > < 目标 >

文件 / 目录打包 (归档)-tar

选项 :

-f: 这个选项是一定要用的 , 表示使用归档文件

-c: 新建一个打包文件

-x : 解包参数

-C: 这个参数是指定解包的目的目录

-t : 显示 tar 打包文件里的内容

文件 / 目录打包 (归档)-tar

选项 :

-A　将其它打包中的文件追加到一个打包文件中

-r　追加一个或多个文件到已有的 tar 包中

-u　如果文件比打包文件中的新则替换打包中的文件

-k: 在解开打包文件时保持原有的文件

-v: 打包 / 解包时显示详细动作

文件 / 目录打包 (归档)-tar

选项 :

-M: 将一个包分成多个包，但必须配合 -L 参数一起使用

-L: 指定分包的每个包大小 (单位为 1024byte)

--remove-files: 打包完以后删除文件

--delete: 删除打包中的一个文件

--get: 提取某个文件

文件 / 目录打包 (归档)-tar

示例 :

1. 现在有 file1 file2 file3 这 3 个文件打包 . 打包文件为 niliu.tar

```
#tar  cvf  niliu.tar  file1  file2  file3
```

2. 解开 niliu.tar 包

```
#tar  xvf  niliu.tar
```

3. 把 niliu.tar 包解开到 /home 目录下

```
#tar  -xf niliu.tar  -C  /home
```

文件 / 目录打包 (归档)-tar

示例 :

4. 显示 niliu.tar 文件里的内容

```
#tar -tf niliu.tar
```

5. 将 snow.tar 包里的文件追加到 niliu.tar 中

```
#tar  -Af niliu.tar   snow.tar
```

6. 将 testc.txt,testd.txt 两个文件追加至 niliu.tar

```
#tar -rf  niliu.tar  testc.txt testd.txt
```

7. 判断 testc.txt,testd.txt 两个文件如比 niliu.tar 包中的新，则替换

```
#tar -uf  niliu.tar  testc.txt testd.txt
```

文件 / 目录打包 (归档)-tar

示例 :

8. 将 test1.txt,test2.txt 文件打入 lisa.tar 包中 , 完成打包后，将 2 个文件从本地删除。

```
#tar  cf  lisa.tar   test1.txt  test2.txt  --remove-files
```

9. 删除 niliu.tar 包中的 test1.txt 文件

```
#tar  f  niliu.tar  --delete  test1.txt
```

10. 提取 niliu.tar 包中的 test2.txt,testc.txt

```
#tar  f  niliu.tar  --get  test2.txt  testc.txt
```

文件 / 目录打包 (归档)-tar

示例 :

11. 将 etc 打包并分割为 1000K 大小的包

#tar cMf etc.tar /etc -L 1000

Prepare volume #2 for 'etc.tar' and hit return:n etc2.tar

n 为对 #2 起一个新文件名 , 文件名为 etc2.tar

Prepare volume #3 for 'etc.tar' and hit return:n etc3.tar

.....

文件 / 目录打包 (归档)-cpio

命令 :cpio

功能 : 将文件 / 目录归档 (打包), 其有关信息包括：文件名，属主，时标 (timestamp)，和访问权限。 归档包可以是磁盘上的 其他文件，也可以是磁带或管道。

语法格式 :cpio [选项] < 归档文件 >

文件 / 目录打包 (归档)-cpio

cpio 有三种操作模式 :

1. copy-out 模式 :

cpio 把文件复制到归档包中。它从标准输入获得文件名列表 (一行一个), 把归档包写到标准输出。

生成文件名列表的典型方法是使用 find 命令 ; 需要要在 find 后面用上 -depth 选项 , 减少因为进入没有访问权限的目录而引起的麻烦。

文件 / 目录打包 (归档)-cpio

cpio 有三种操作模式 :

2.copy-in 模式

1. cpio 从归档包里读取文件，或者列出归档包里的内容。

2. 它从标准输入读入归档包。任何不是选项的命令行参数被视为 shell 的通配符模式串 (globbing pattern)

3. 在归档包中，只有文件名匹配这些模式串的文件才能复制出来。

4. 与 shell 中不同，文件名起始处的 '.' 可以匹配模式串起始处的通配符，文件名中的 '/' 也可以匹配通配符。 如果没有给出模式串，那么将读出所有文件。

文件 / 目录打包 (归档)-cpio

cpio 有三种操作模式 :

3. copy-pass 模式

1. cpio 可以把文件从一棵目录树复制到另一棵 ,

2. 它结合了 copy-in 和 copy-out 的操作 , 但不使用归档包。

3. cpio 从标准输入读取欲复制的文件名列表 ; 目标目录作为非选项的命令行参数给出。

文件 / 目录打包 (归档)-cpio

cpio 支持的归档格式 :

1. binary
2. old ASCII,
3. new ASCII
4. crc
5. HPUX binary
6. HPUX old ASCII
7. old tar
8. POSIX.1 tar

文件 / 目录打包 (归档)-cpio

1. "binary" 格式是过时格式，因为它保存文件信息的方法无法应用在不同体系的机器间移植。

2. "old ASCII" 格式可以跨平台使用，但是不能用于超过 65536 个 i 节点的文件系统中。

3. "new ASCII" 格式可以跨平台使用，也适用于任意大小的文件系统，但不是所有版本的 cpio 都支持；目前只有 GNU 和 System VR4 的 cpio 支持。

4. "crc" 格式 类似于 "new ASCII" 格式，同时对每个文件计算校验和。 cpio 在创建归档包时算出校验和，解开文件时进行校验。

文件 / 目录打包 (归档)-cpio

5. "HPUX" 格式用于兼容 HP UNIX 的 cpio，它用了独特的方法来保存设备文件。

6. "tar" 格式用以兼容 tar 程序。它不能归档文件名超过 100 个字符的文件，也不能归档特殊文件 (块设备或字符设备)。

7. "POSIX.1 tar" 格式不能归档文件名超过 255 个字符的文件 (小于，除非文件名的最右边有一个 "/")。

文件 / 目录打包 (归档)-cpio

缺省情况下：

cpio 为了兼容老式的 cpio 程序，创建 "binary" 格式的归档包。当展开归档包时，cpio 能够自动识别归档包的格式，而且可以读取在其他字节顺序的机器上创建的归档包。

cpio 的某些选项只能用在对应的操作模式上。

文件 / 目录打包 (归档)-cpio

选项 :

-o: 将数据 copy 到文件或设备上

-i: 将数据从文件或设备上还原到系统中

-t : 查看 cpio 建立的文件或设备内容

-c : 一种比较新的 portable format 方式存储

-v : 在屏幕上显示备份过程中的文件名

文件 / 目录打包 (归档)-cpio

选项 :

-B: 让预设的 blocks 可以增加到 5120bytes ， 默认是 512bytes ， 这样可以使备份速度加快。

-d : 自动建立目录，这样还原时才不会出现找不到路径的问题。

-u ：更新，用较新的文件覆盖旧的文件

-m，创建文件时，保留以前的文件修改时间。

文件 / 目录打包 (归档)-cpio

示例 :

1. 将本目录所有子目录及文件归档之 niliu.cpio

   #find . | cpio -ocvB > /tmp/niliu.cpio

2. 查看归档文件 niliu.cpio 信息

   #cpio -tv < /tmp/niliu.cpio

3. 解开镜像文件

   1) 拷贝镜像文件

   #cp /boot/initram*.x86_64.img ~/img

文件 / 目录打包 (归档)-cpio

示例 :

3. 解开镜像文件

2) 确定 img 文件类型为 CPIO 归档文件

#file initram*.x86_64.img

3) 解开镜像文件 , 自动建立目录，保留权限 ,

#cpio -ivdm < initram*.x86_64.img

压缩指令 -gzip

命令 :gzip

功能 : 将文件 / 打包文件压缩

语法格式 :gzip < 选项 > < 目标 >

选项 :
-1 压缩速度最快，压缩比最小
-9 压缩速度最慢，压缩比最大
-r: 递归压缩目录中的文件
-l: 显示压缩整体情况
-t: 测试压缩完整性

压缩指令 -gzip

特点：

1. 不能压缩目录
2. 后缀名为 gz
3. 默认压缩比为 6

解压指令 -gunzip

命令 :gunzip

功能 : 将文件 / 打包文件解缩

语法格式 :gunzip < 选项 >   < 压缩文件 >

解压指令 -zcat

命令 :zcat

功能 : 查看 gzip 的文件

语法格式 :zcat < 选项 > < 压缩文件 >

示例 :
1 将 niliu.cpio.gz 文件解压
  #zat niliu.cpio.gz | cpio -idvm

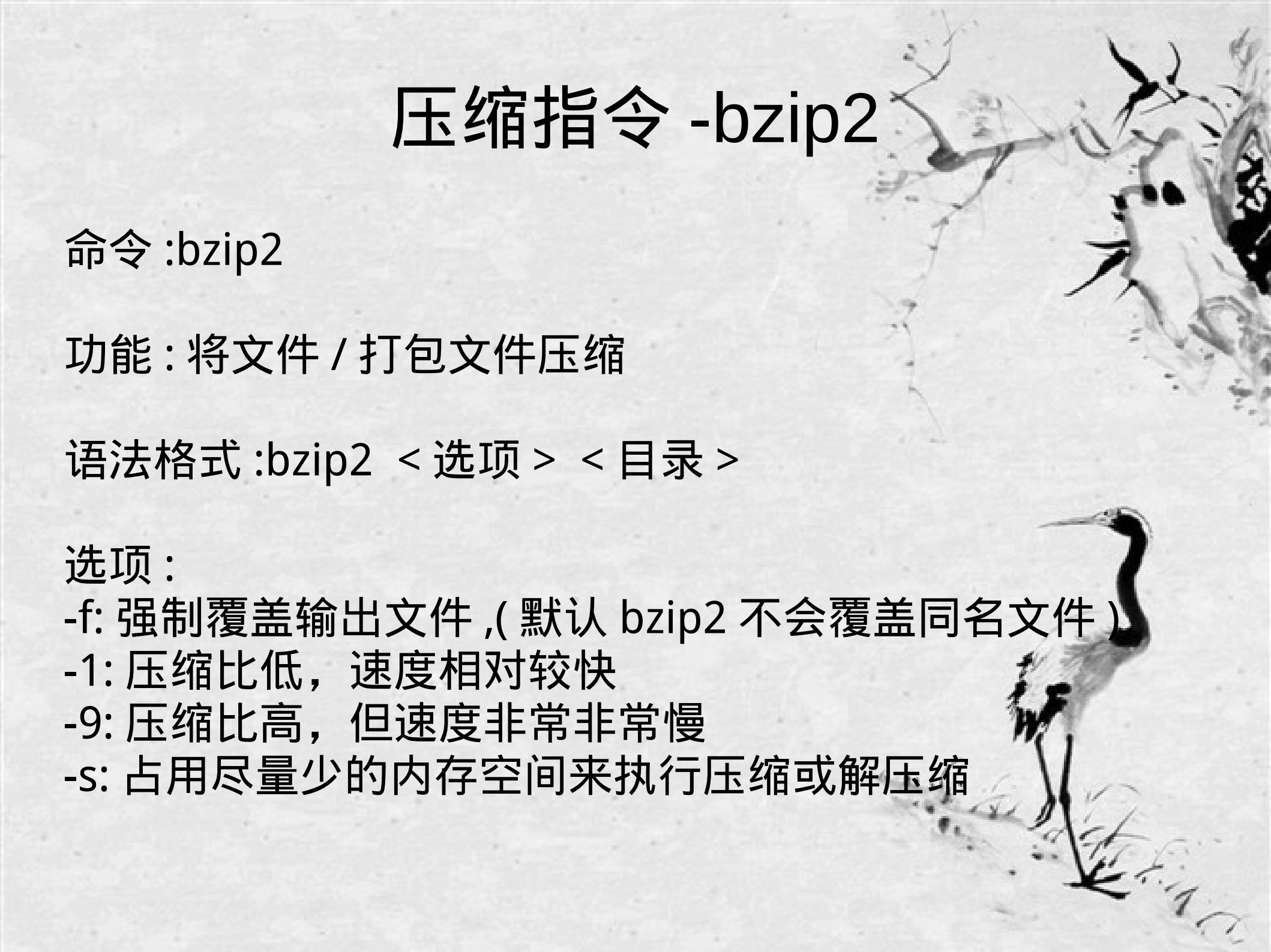
压缩指令 -bzip2

命令 :bzip2

功能 : 将文件 / 打包文件压缩

语法格式 :bzip2 < 选项 > < 目录 >

选项 :
-f: 强制覆盖输出文件 ,(默认 bzip2 不会覆盖同名文件)
-1: 压缩比低，速度相对较快
-9: 压缩比高，但速度非常非常慢
-s: 占用尽量少的内存空间来执行压缩或解压缩



解压指令 -bunzip2

命令 :bunzip2

功能 : 将文件 / 打包文件解缩

语法格式 :bunzip2 < 选项 >   < 压缩文件 >

压缩指令 -xz

命令 :xz

功能 : 将文件 / 打包文件压缩

语法格式 :xz < 选项 > < 压缩文件 >

解压指令 -unxz

命令 :unxz

功能 : 将文件 / 打包文件解压缩

语法格式 :unxz < 选项 > < 解压缩文件 >

打包并压缩 -tar

tar 可以实现直接调用相关压缩程序来完成打包及压缩的步骤

1. 调用 gzip/gunzip
   #tar cvfz etc.tar.gz /etc
   #tar xvfz etc.tar.gz

2. 调用 bzip2
   #tar cvfj etc.tar.bz2 /etc
   #tar xvfj etc.tar.bz2

打包并压缩 -tar

tar 可以实现直接调用相关压缩程序来完成打包及压缩的步骤

3. 调用 xz/unxz

#tar cvfJ etc.tar.xz /etc

#tar xvfJ etc.tar.xz

当压缩与打包一步完成后，后缀的标识可以为：

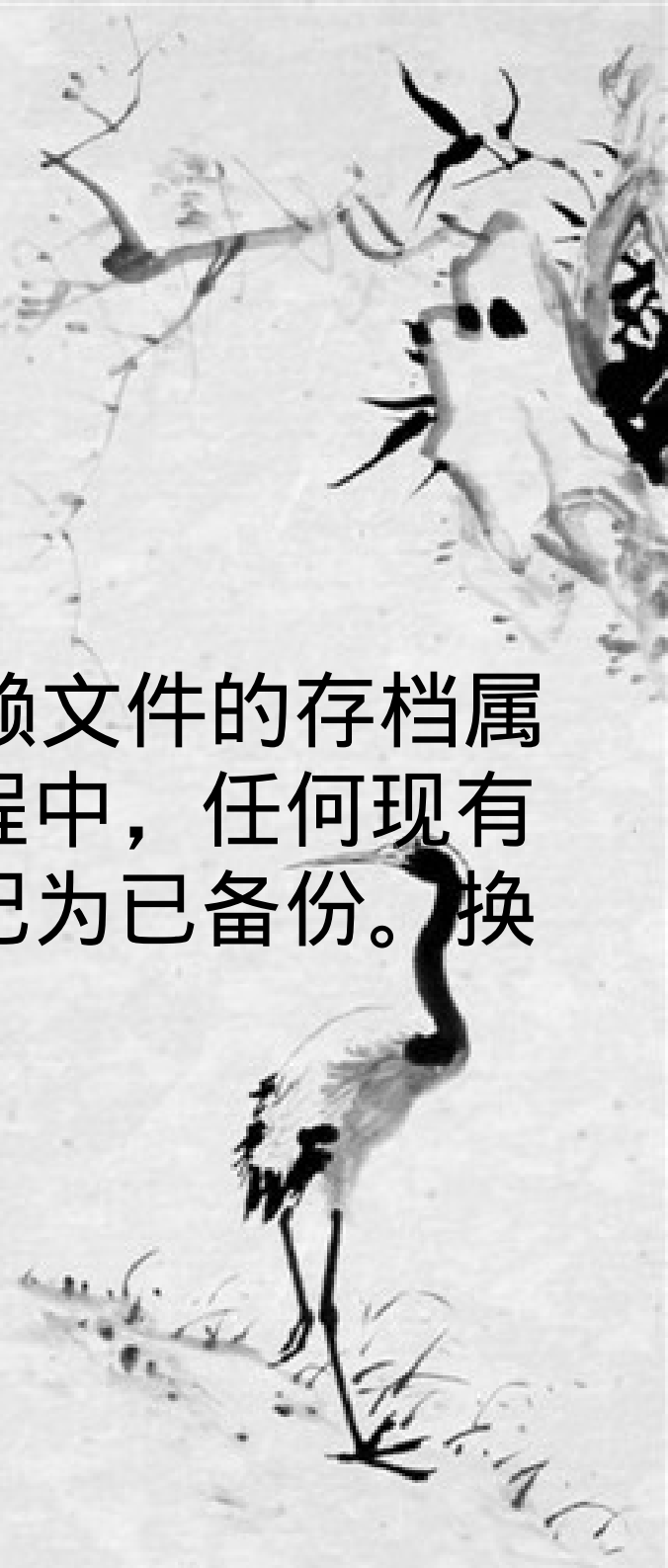
tar.gz=tgz　tar.bz2=tbz　tar.xz=txz

系统备份

备份的概念

完整备份（ Full Backup ）：

　备份全部选中的文件夹，并不依赖文件的存档属性来确定备份那些文件。在备份过程中，任何现有的标记都被清除，每个文件都被标记为已备份。换言之，清除存档属性。

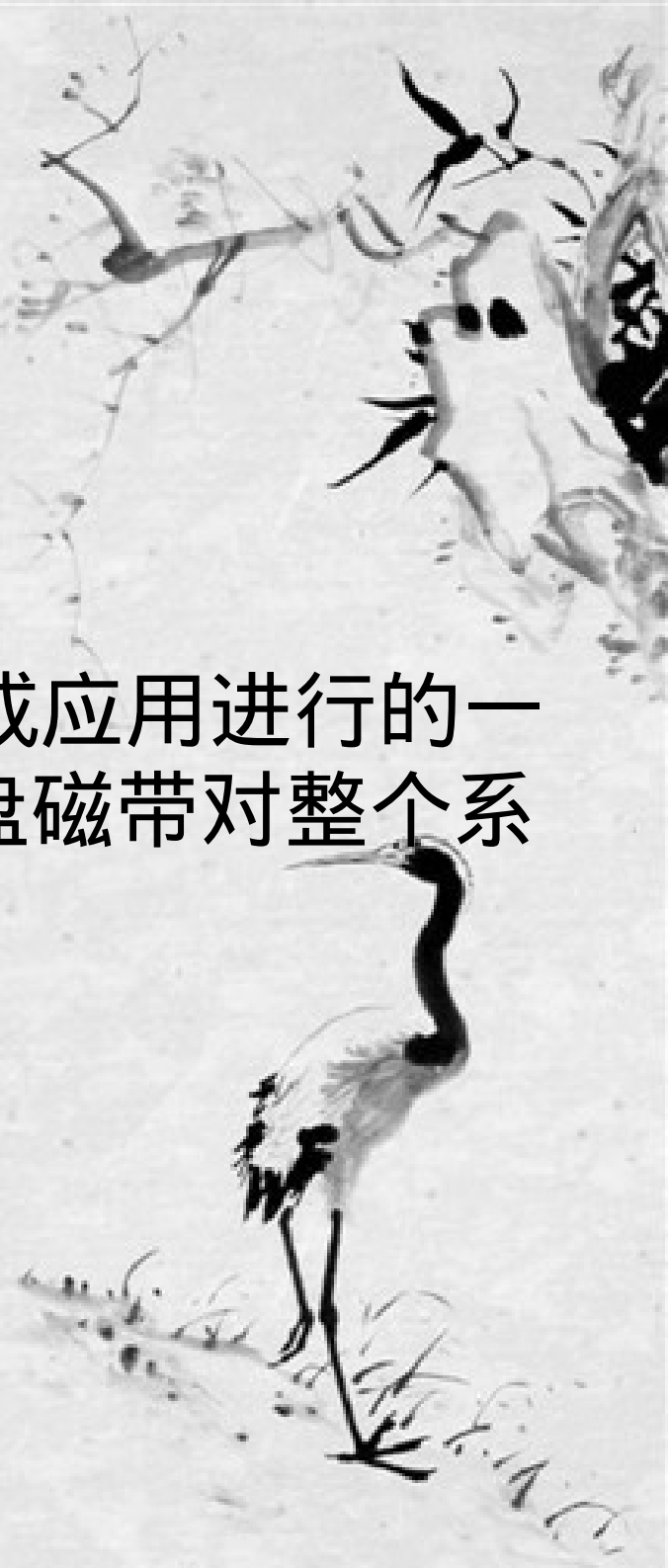


系统备份

备份的概念

完整备份（ Full Backup ）：

　指对某一个时间点上的所有数据或应用进行的一个完全拷贝。实际应用中就是用一盘磁带对整个系统进行完全备份。



系统备份

备份的概念

完整备份（ Full Backup ）：

优点：

就是只要用一盘磁带，就可以恢复丢失的数据。因此大大加快了系统或数据的恢复时间。

缺点：

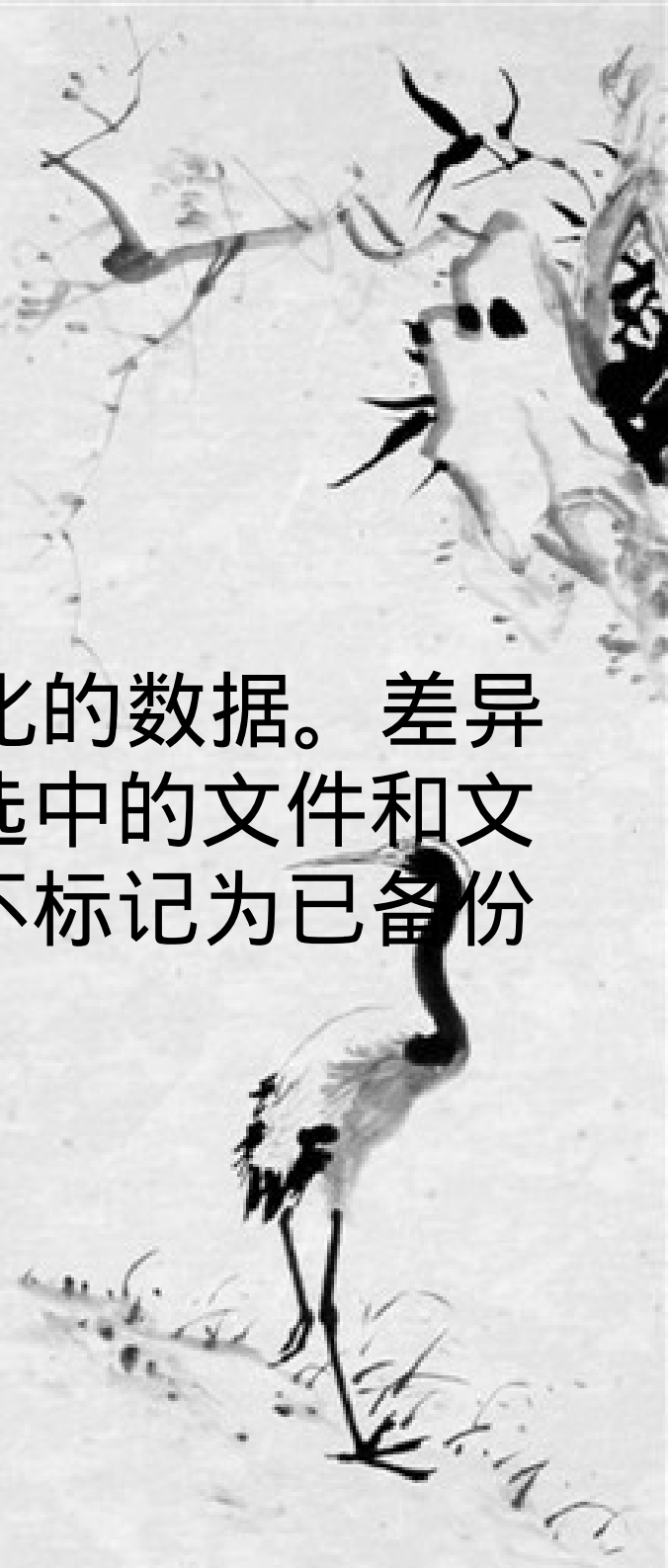
各个全备份磁带中的备份数据存在大量的重复信息；另外，由于每次需要备份的数据量相当大，因此备份所需时间较长。

系统备份

备份的概念

差异备份 (Differential Backup):

　备份自上一次完全备份之后有变化的数据。差异备份过程中，只备份有标记的那些选中的文件和文件夹。它不清除标记，也即备份后不标记为已备份文件。换言之，不清除存档属性。



系统备份

备份的概念

差异备份 (Differential Backup):

优点：

差异备份在避免了另外两种备份策略缺陷的同时，又具备了它们各自的优点。

1. 它具有了增量备份需要时间短、节省磁盘空间的优势；

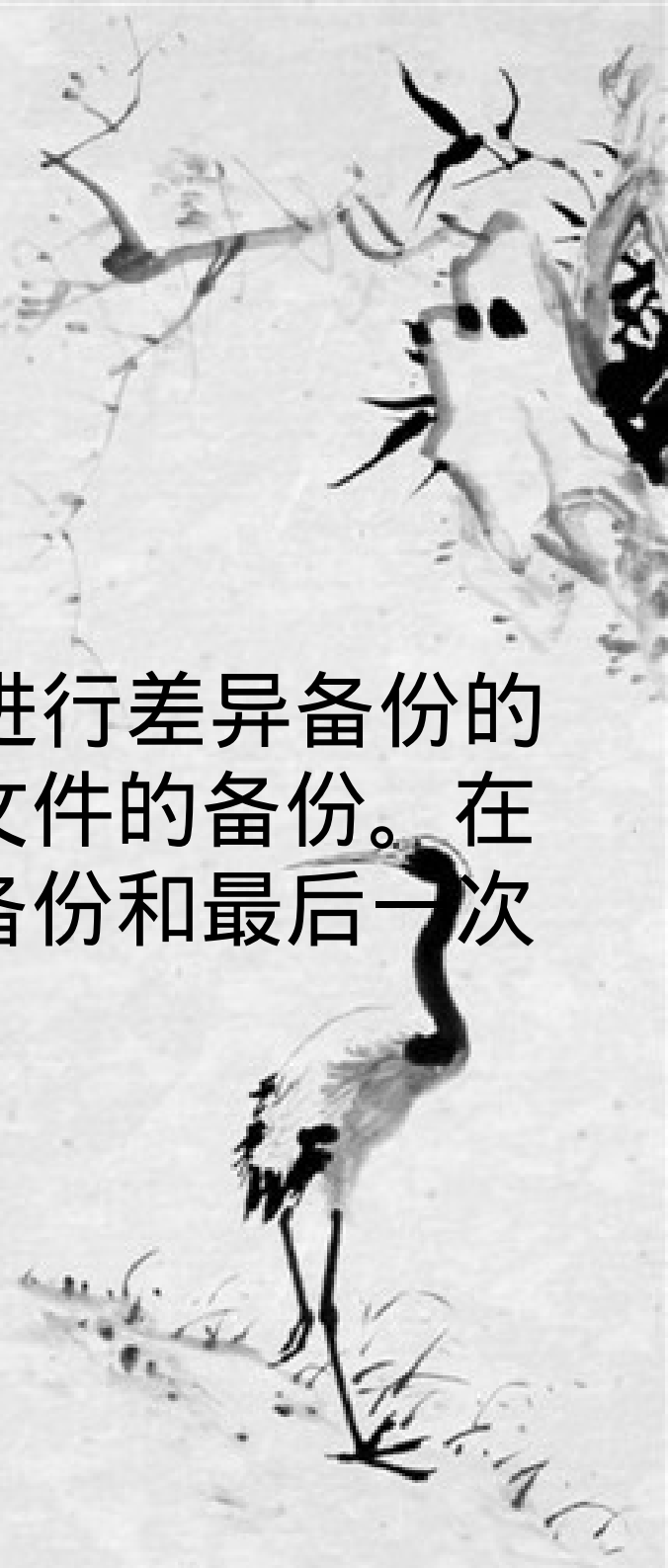
2. 它又具有了全备份恢复所需磁带少、恢复时间短的特点。系统管理员只需要两盘磁带，即全备份磁带与灾难发生前一天的差异备份磁带，就可以将系统恢复。

系统备份

备份的概念

差异备份 (Differential Backup):

差异备份是指在一次全备份后到进行差异备份的这段时间内，对那些增加或者修改文件的备份。在进行恢复时，我们只需对第一次全备份和最后一次差异备份进行恢复。

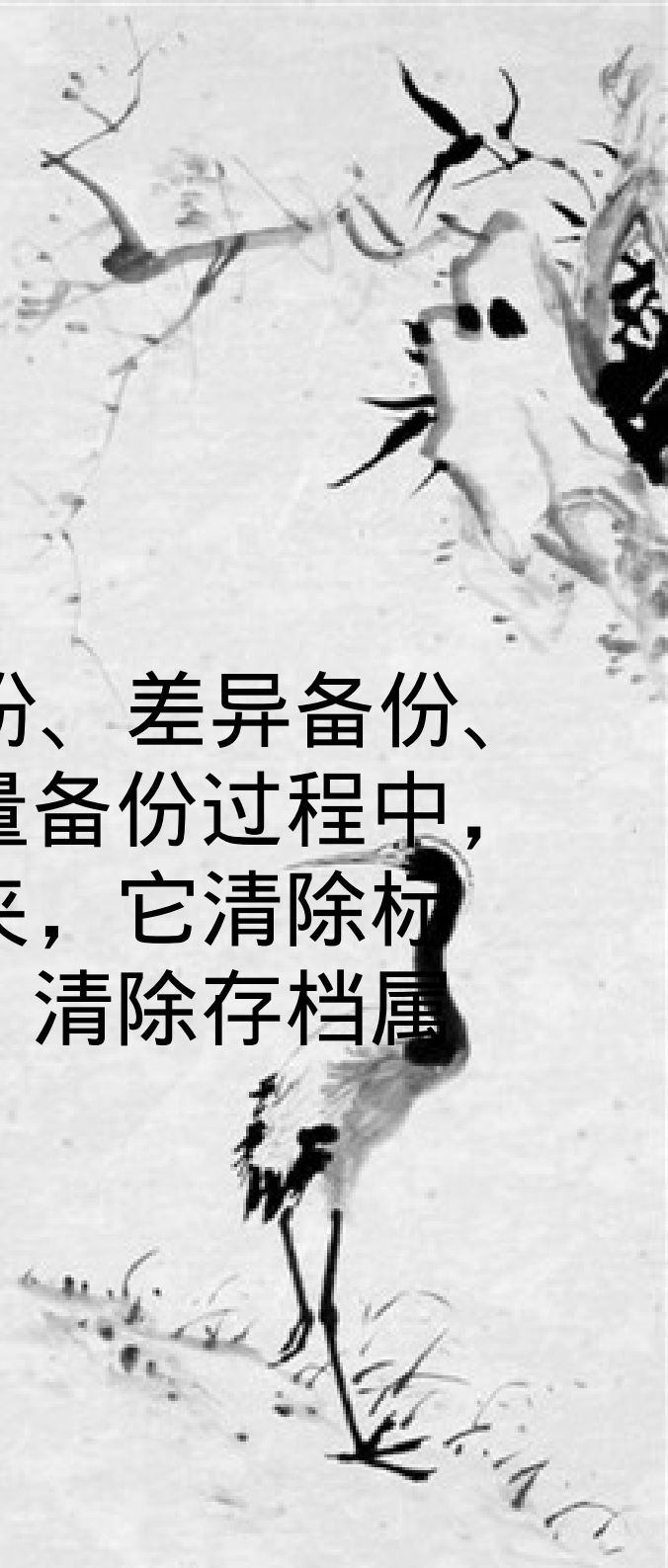


系统备份

备份的概念

增量备份 (Incremental Backup):

　备份自上一次备份（包含完全备份、差异备份、增量备份）之后有变化的数据。增量备份过程中，只备份有标记的选中的文件和文件夹，它清除标记，既：备份后标记文件，换言之，清除存档属性。

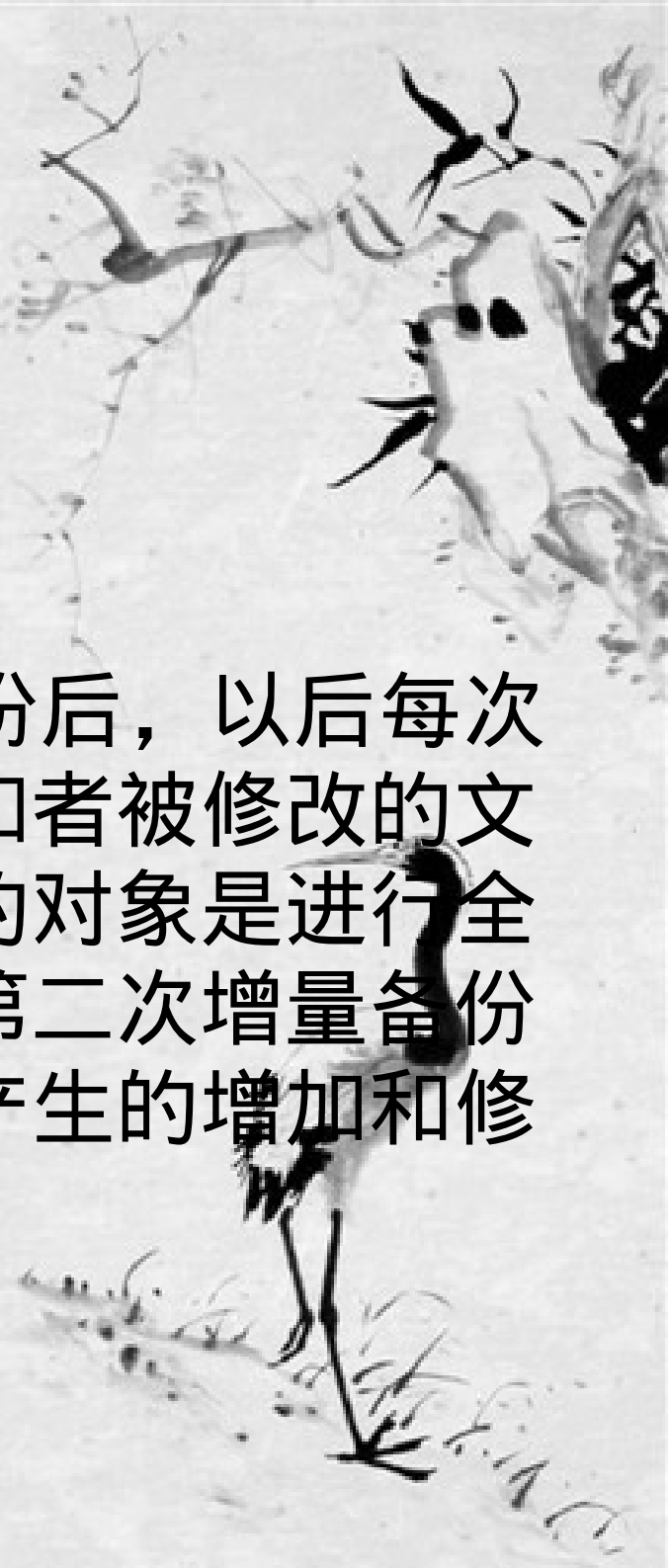


系统备份

备份的概念

增量备份 (Incremental Backup):

　指在一次全备份或上一次增量备份后，以后每次的备份只需备份与前一次相比增加和者被修改的文件。这就意味着，第一次增量备份的对象是进行全备后所产生的增加和修改的文件；第二次增量备份的对象是进行第一次增量备份后所产生的增加和修改的文件，如此类推。



系统备份

备份的概念

增量备份 (Incremental Backup):
优点 :

1. 没有重复的备份数据，因此备份的数据量不大。
2. 数据量不大意味着备份所需的时间很短。

系统备份

备份的概念

增量备份 (Incremental Backup):
缺点：

1. 增量备份的数据恢复是比较麻烦的。必须具有上一次全备份和所有增量备份磁带（一旦丢失或损坏其中的一盘磁带，就会造成恢复的失败）

2. 必须按照从全备份到依次增量备份的时间顺序逐个反推恢复，因此这就极大地延长了恢复时间。

系统备份 -dump

命令 :dump

功能 : 文件系统备份

语法格式 :

dump [选项] < 备份设备名称 > < 需备份的文件系统 >

系统备份 -dump

备份级别 :
0-9:

1)0 为基本级 (完整级),1 为 0 级改变的 ,2 为 1 级改变的 , 直至至 9 级别

2) 每次的基本将记录之 /etc/dumpupdates 文件

3) 从 9 后再次循环至 0 级开始重置 , 进行下次循环

系统备份 -dump

参数：

-0123456789： 备份级别

-b< 区块大小 >: 指定备份的区块大小，单位 KB

-B< 区块数目 >: 指定备份卷的区块数量

-c: 修改磁带备份预设的密度与容量

-d< 密度 >: 设定磁带备份的密度，单位 BPI

系统备份 -dump

参数：

-f< 备份设备 >: 指定备份的设备

-h< 层级 >: 当备份≥指定的备份级别时，将不备份用户标识为“ nodump” 文件

-n: 当备份工作需要管理员介入时，向所有“ operator” 群组发送通知

-s< 磁带长度 >: 备份的磁带长度，单位：英尺

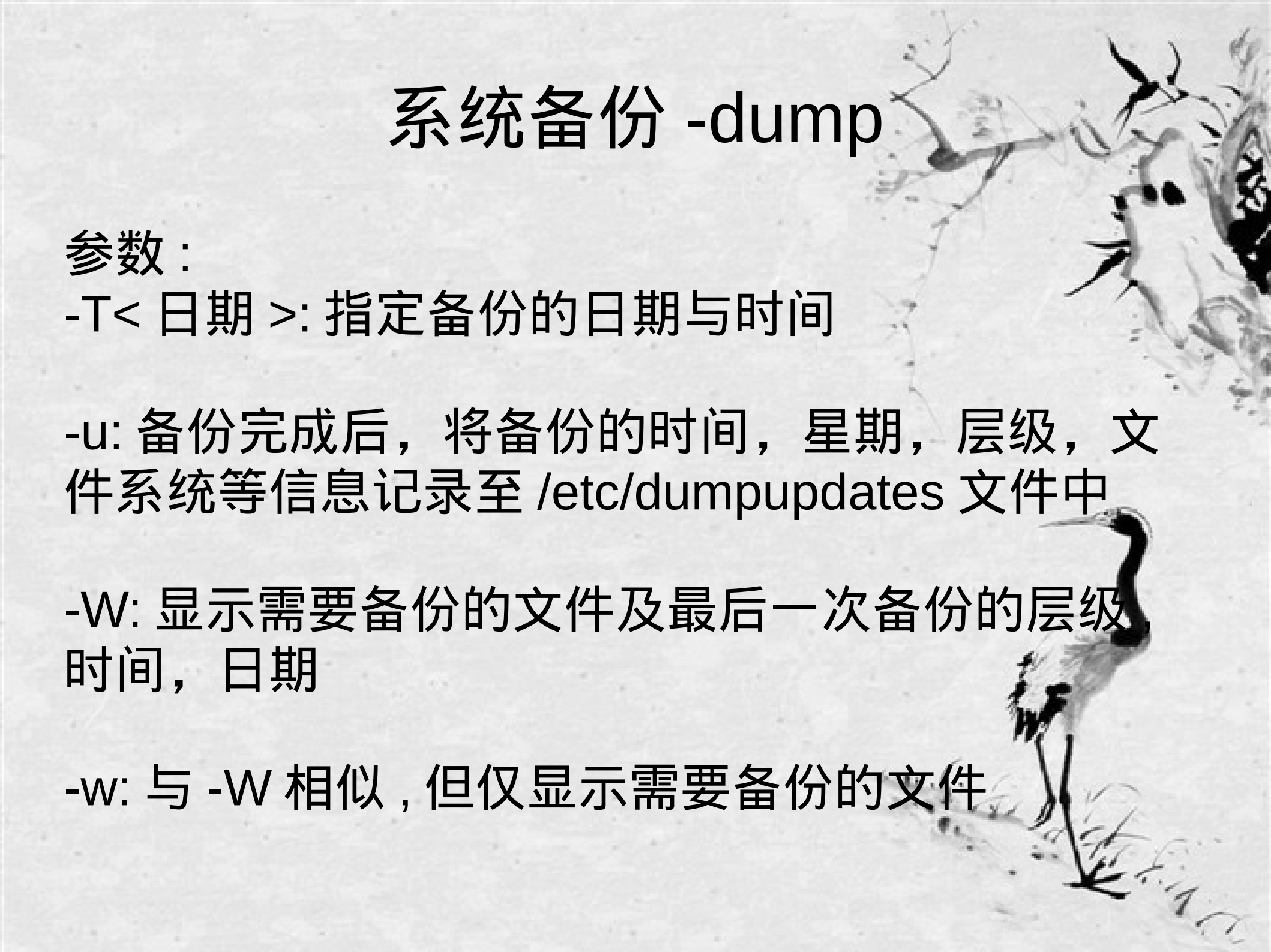
系统备份 -dump

参数：
-T< 日期 >: 指定备份的日期与时间

-u: 备份完成后，将备份的时间，星期，层级，文件系统等信息记录至 /etc/dumpupdates 文件中

-W: 显示需要备份的文件及最后一次备份的层级，时间，日期

-w: 与 -W 相似，但仅显示需要备份的文件



系统备份 -dump

示例

1. 将 /boot 目录备份到 /dev/sdb1 上

```
#dump  -0uf  /dev/sdb1  /boot
```

2. 将 /dev/sda1 文件系统备份到 /dev/sdb1 上，层级采用 1 级 .

```
#dump -1uf  /dev/sdb1  /dev/sda1
```

3. 查看最后一次备份的文件及详细信息

```
#dump  -W
```

系统还原 -restore

命令 :restore

功能 : 将 dump 备份内容进行还原

语法格式 :

restore [*选项*]... 备份介质或档案 [*文件或目录*]...

系统还原 -restore

选项 :
-f: 指定所需要还原的备份介质

-t: 指定备份介质中的文件

-r: 指定还原操作

-x: 提取备份介质中的指定文件

-i: 还原时采取交互模式

-v: 显示还原的详细动作

系统还原 -restore

示例 :

1. 显示 /dev/sdb1 下的完整备份中的文件及目录
   #restore tf /dev/sdb1

2. 将 /dev/sdb1 中的文件还原至当前目录
   #restore xf /dev/sdb1 ./grub/splash.xpm.gz
   如果提示 :
   specify next volume #.....(可按编号 , 如 1)
   set owner/mode for '.' ?[yn](键入 n ， 不做此操作)

系统还原 -restore

示例：

3. 恢复 /dev/sdb1 所备份的所有内容在当前目录

```
#restore  -rf  /dev/sdb1
```

系统备份 -dd

命令 :dd

功能 : 指定大小的块拷贝一个文件，并在拷贝的同时进行指定的转换。

语法格式 :dd [选项]

系统备份 -dd

选项：
if = 输入文件（或设备名称）。

of = 输出文件（或设备名称）。

ibs = bytes: 一次读取 bytes 字节，即读入缓冲区的字节数。

skip = blocks: 跳过读入缓冲区开头的 ibs*blocks 块。

obs = bytes: 一次写入 bytes 字节，即写入缓冲区的字节数。

系统备份 -dd

选项：
bs = bytes: 同时设置读 / 写缓冲区的字节数（等于设置 ibs 和 obs ）。

cbs = byte: 一次转换 bytes 字节。

count=blocks: 只拷贝输入的 blocks 块。

conv = ASCII: 把 EBCDIC 码转换为 ASCIl 码。

conv = ebcdic: 把 ASCIl 码转换为 EBCDIC 码。

conv = ibm: 把 ASCIl 码转换为 alternate EBCDIC 码。

系统备份 -dd

选项:

conv = block 把变动位转换成固定字符。

conv = ublock 把固定位转换成变动位。

conv = ucase 把字母由小写转换为大写。

conv = lcase 把字母由大写转换为小写。

conv = notrunc 不截短输出文件。

系统备份 -dd

选项:
conv = swab 交换每一对输入字节。

conv = noerror 出错时不停止处理。

conv = sync 把每个输入记录的大小都调到 ibs 的大小（用 NUL 填充）。

系统备份 -dd

示例：

1. 将光盘转换为 iso
   #dd if=/dev/cdrom of=./centos7.iso

2. 将 /dev/sda 转换到 /dev/sdb
   #dd if=/dev/sda of=/dev/sdb

3. 将 /dev/sda1 转换到 /dev/sdb1
   #dd if=/dev/sda1 of=/dev/sdb1

系统备份 -dd

示例：

4. 将 /dev/sda3 转换成一个文件

```
#dd  if=/dev/sda3  of=./sda3.img
```

5. 创建一个 100M 的虚拟硬盘，每次读写缓存为 1M，总计累计 100 次

```
#dd  if=/dev/zero  of=disk1.img  bs=1M  count=100
```

6. 创建一个 100M 的虚拟硬盘，每次读写缓存为 100M，总计累计 1 次

```
#dd  if=/dev/zero  of=disk1.img  bs=100M  count=1
```

系统备份 -dd

示例：

7. 建立一个 swap 文件，并使用。

```
#dd  if=/dev/zero  of=/tmp/swap  bs=100M count=1

#mkswap  /tmp/swap

#chmod 600 /tmp/swap

#swapon  /tmp/swap
```

系统备份 -dd

独特的设备文件

/dev/zero: 该设备无穷尽地提供 0 ，可以使用任何你需要的数目——设备提供的要多的多。他可以用于向设备或文件写入字符串 0

/dev/null: 空设备，也称为位桶（ bit bucket ）。任何写入它的输出都会被抛弃。如果不想让消息以标准输出显示或写入文件，那么可以将消息重定向到位桶。

系统备份 -genisoimage(mkisofs)

命令 :genisoimage

功能 : 制作 iso9660/Joliet/HFS 文件系统

语法结构 :genisoimage [选项] -o 输出文件 [目标文件]

系统备份 -genisoimage

选项：

-A< 应用程序 ID>: 指定光盘的应用程序 ID 。

-biblio<ISBN 文件 > : 指定 ISBN 文件的文件名， ISBN 文件位于光盘根目录下，记录光盘的 ISBN 。

-c< 开机文件名称 > : 制作可开机光盘时， mkisofs 会将开机映像文件中的全 -eltorito-catalog< 开机文件名称 > 全部内容作成一个文件。

系统备份 -genisoimage

选项：

-C< 盘区编号，盘区编号 > : 将许多节区合成一个映像文件时，必须使用此参数。

-copyright< 版权信息文件 >: 指定版权信息文件的文件名。

-d: 省略文件后的句号。

系统备份 -genisoimage

选项：
-D:ISO 9660 最多只能处理 8 层的目录，超过 8 层的部分， RRIP 会自动将它们设置成 ISO 9660 兼容的格式。使用 -D 参数可关闭此功能。

-f: 忽略符号连接。

-h: 显示帮助。

-hide< 目录或文件名 >: 使指定的目录或文件在 ISO 9660 或 Rock RidgeExtensions 的系统中隐藏。

系统备份 -genisoimage

选项 :
-hide-joliet< 目录或文件名 > : 使指定的目录或文件在 Joliet 系统中隐藏。

-J: 使用 Joliet 格式的目录与文件名称。

-L: 允许文件名的第一个字符为句号。

-log-file< 记录文件 > : 在执行过程中若有错误信息，预设会显示在屏幕上。

系统备份 -genisoimage

选项 :

-m< 目录或文件名 >: 指定的目录或文件名将不会放入映像文件中。

-M< 映像文件 >: 与指定的映像文件合并

-N: 省略 ISO 9660 文件中的版本信息。

-o< 映像文件 >: 指定映像文件的名称。

-print-size　显示预估的文件系统大小。

系统备份 -genisoimage

选项 :
 -quiet: 执行时不显示任何信息。

-r: 使用 Rock Ridge Extensions ，并开放全部文件的读取权限。

-R: 使用 Rock Ridge Extensions 。

-sysid< 系统 ID>: 指定光盘的系统 ID 。

-T: 建立文件名的转换表，适用于不支持 Rock Ridge Extensions 的系统上。

系统备份 -genisoimage

选项 :
-v: 执行时显示详细的信息。

-V< 光盘 ID>: 指定光盘的卷册集 ID 。

-volset-size< 光盘总数 >: 指定卷册集所包含的光盘张数。

-volset-seqno< 卷册序号 >: 指定光盘片在卷册集中的编号。

-x< 目录 >: 指定的目录将不会放入映像文件中。

系统备份 -genisoimage

示例 :
1. 将 /etc 目录做成一个 etc.iso
    #genisoimage -J -L -r -o etc.iso /etc

说明 :
-L: 允许文件名的第一个字符为句号。(隐藏文件)

-J: 使用 Joliet 格式的目录与文件名称

-r: 使用 Rock Ridge Extensions，并开放全部文件的读取权限。

-o< 映像文件 >: 指定映像文件的名称。

系统备份 -genisoimage

示例 :
2. 将 /mnt/backup 目录做成一个 backup.iso
#genisoimage -J -v -o bakcup.iso /mnt/backup

说明 :
-v: 显示详细动作

-J: 使用 Joliet 格式的目录与文件名称

-o< 映像文件 >: 指定映像文件的名称。

系统备份 -genisoimage

示例 :

3. 将 /root 目录做成一个 root.iso, 但不含 /root/test 目录

```
#genisoimage  -J  -L  -x  /root/test  -o  root.iso  /root
```

说明 :

-L: 允许文件名的第一个字符为句号。

-J: 使用 Joliet 格式的目录与文件名称

-x: 不包含指定的目录

-o< 映像文件 >: 指定映像文件的名称。

系统备份 -wodim

命令 :wodim

功能 : 光盘刻录命令

语法格式 :wodim [选项] < 设备 > < 刻录文件 >

系统备份 -wodim

示例：

1. 查找本地刻录设备

   #wodim --devices

2. 擦除光盘信息 (DVD-RW 介质)

   #wodim -v dev=/dev/dvdrw blank=fast

3. 刻录光盘

   #wodim -v dev=/dev/dvdrw ./rhel.iso

系统备份 -wodim

示例：

4. 刻录光盘后直接弹出光驱

```
#wodim  -eject  -v  dev=/dev/dvdrw  ./rhel.iso
```

5. 设定刻录速度为 1 倍速

```
#wodim -eject -v speed=1 dev=/dev/dvdrm ./rhel.iso
```

系统备份 -mt

命令 :mt

功能 : 磁带控制

软件包 :mt-st

语法格式 :mt [选项] < 设备 >

系统备份 -mt

示例：

1. 将磁带倒带，使磁带卷至起始位置
   #mt -f /dev/st0 rewind

2. 擦除磁带中的内容
   #mt -f /dev/st0 erase

3. 出带 - 即将磁带卷至起始位置并弹出磁带
   #mt -f /dev/st0 offline

系统备份 -mt

示例：

4. 通过 tar 查看磁带上的文件或目录

   #tar tvf /dev/st0 [文件 / 目录]

5. 将 niliu.tar 文档备份到磁带上

   #tar cvf /dev/st0 ~/niliu.tar

6. 通过 tar 继续将数据写入磁带而不是覆盖磁带原有数据

   #tar rvf /dev/st0 ~/niliu1.tar

系统备份 -mt

示例：

7. 通过磁带恢复备份的 niliu.tar 文件

   1)

   #tar tvf /dev/st0

   2)

   #tar xvf /dev/st0 niliu.tar