

# 第12章 文件操作

---

# 内部存储器和外部存储器

- 我要买个电脑.....

我要内存  
**16G**



我要硬盘  
**2T**

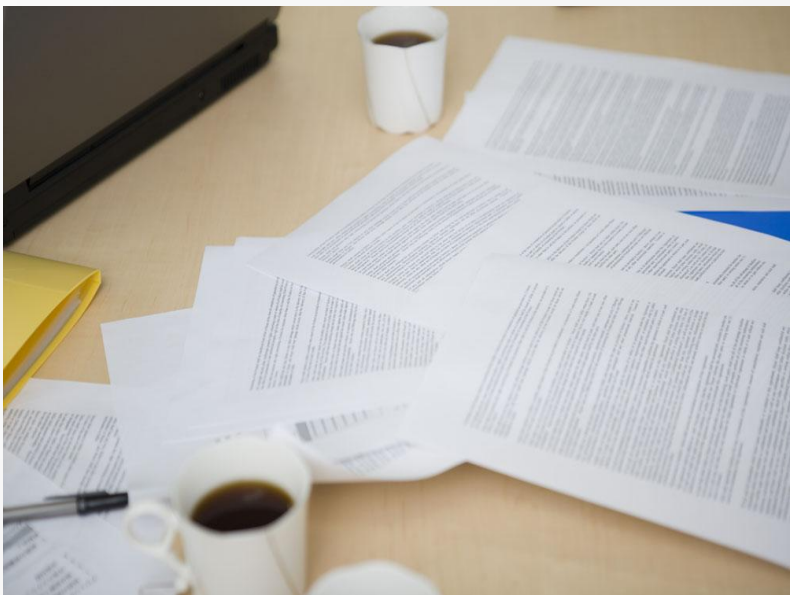
# 存储在外存的数据

## ■ 外部存储器

- \* 外存容量大、断电后数据不丢失，可重复使用，永久保存



# 什么是文件 (Files)



# 什么是文件（Files）

- 存储在外部介质上具有名字的一组相关数据的集合
  - \* 数据一般以文件的形式为用户及应用程序使用



程序与数据分离



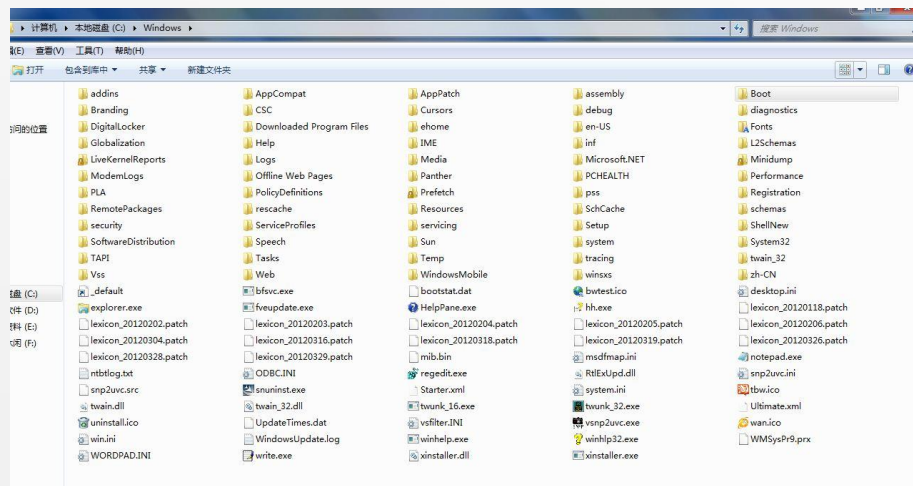
数据共享



长期保存数据



# 文件是如何组织的



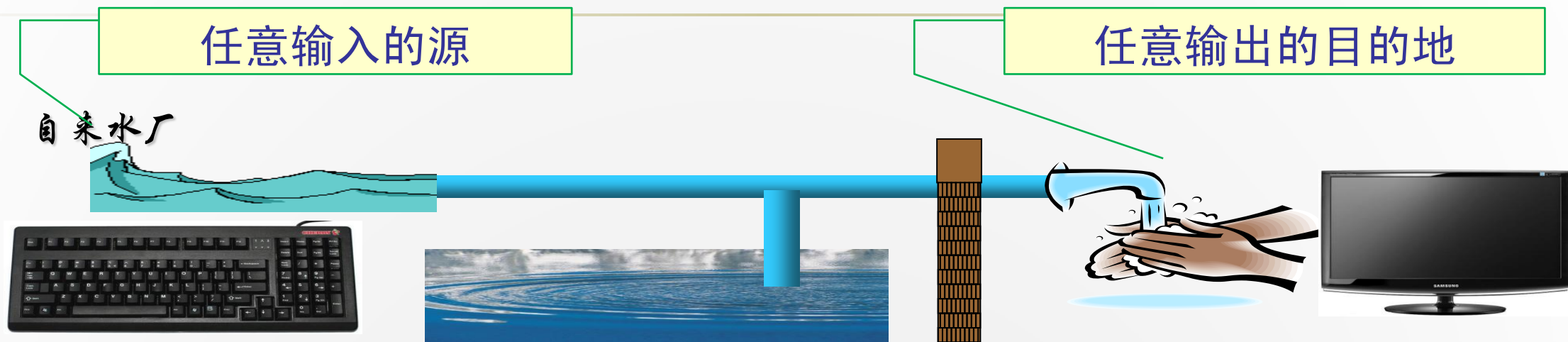
- 建立若干目录(文件夹, directory\folder)
  - \* 在目录里保存文件 (同一级目录里保存的文件不能同名)
- 对使用者而言, 只要知道文件的路径(path, 全目录)和文件名, 就能使用该文件

# 文件的分类

## ■ 按文件的逻辑结构

- \* 记录文件：由具有一定结构的记录组成（定长和不定长）
- \* 流式文件（**stream**）：由一个个字符（字节）数据顺序组成
  - \* 数据流
  - \* 字节流

# 文件的分类



- 通过一个流（通常和键盘相关）获得全部的输入
- 通过另一个流（通常和屏幕相关）写出全部的输出



# 文件的分类

## ■ C语言中的标准流

文件指针	标准流	默认的含义
stdin	标准输入	键盘
stdout	标准输出	终端显示器屏幕
stderr	标准错误输出	终端显示器屏幕

`scanf()` , `getchar()` , `gets()` 等通过`stdin`获得输入

`printf()` , `putchar()` , `puts()` 等用`stdout`进行输出

# 输入/输出重定向 ( **redirection** )

- 某些操作系统允许标准输入/输出重定向文件
  - \* DOS和UNIX允许程序从文件获得输入或向文件写数据
    - ∞ 这种重定向，程序本身是感觉不到的
- 输入重定向 ( **Input Redirection** )
  - \* `D:\>demo < infile.txt`
  - \* 从终端（键盘）输入数据改成从文件中读入数据
- 输出重定向 ( **Output Redirection** )
  - \* `D:\>demo > outfile.txt`
  - \* 从终端（显示器）输出数据重定向到文件

# 文件的分类

---

- 按数据的组织形式可以分为：

- \* 文本文件（Text file）

- \* C程序的源代码

- \* 二进制文件（Binary file）

- \* 可执行的C程序

# 文本文件和二进制文件

## ■ 文本文件

- \* 用字节表示字符的字符序列，存储每个字符的ASCII码

- \* 整数123在文本文件中占3个字节，分别存放这3个字符的ASCII码

字符：

十进制的 ASCII 值：

二进制的 ASCII 值：

'1'	'2'	'3'
49	50	51
00110001	00110010	00110011

## ■ 二进制文件

00000000	01111011
----------	----------

# 文本文件和二进制文件

- 二进制文件与文本文件的区别：
  - \* 按照数据在内存中的存储形式（二进制）存储到文件
  - \* 二进制文件存储的字节不一定表示字符，无需ASCII码表进行字符变换，读写速度快

TEST.BIN 内容	int 100		float 100.0				字符串 "100"				字符串 "END"			
	64	00	00	00	C8	42	31	30	30	00	45	4E	44	00
对应的 ASCII 码字符	d                                  lB100                  E  N  D													

- 必须按照数据存入的类型和格式读出才能恢复本来面貌
  - \* 先按int类型读，为0x0064，是整数100
  - \* 而按float读，将读出0x00000064，对应的float值为1.4012985e-43，近似为0，面目全非

# 文本文件和二进制文件

- 总之，二进制文件保存数据，不仅要按照存入时的类型，还要按存入时的格式读出，才能恢复其本来面貌。
  - \* 公开的标准文件格式
    - \* 如bmp、tif、gif、jpg和mp3等类型的文件，有大量软件生成和使用这些类型的文件
  - \* 不公开或加密的文件格式
    - \* 如Microsoft Word的doc格式就不公开



# 文本文件和二进制文件

## ■ 对于程序员来说，区分文本文件和二进制文件：

### \* 文本文件——按行划分

- \* 所以必须用特殊的字符标记行的结尾
- \* 某些OS还可能用特殊的字符标记文件的末尾
  - ∞ 例如，DOS将Ctrl+Z设定为文件的结束符

### \* 二进制文件——不是按行划分的

- \* 可合法地包含任何字符，故不可能留出文件结束符