文件属性、树形结构

文件控制块及文件目录

下面我们来介绍文件控制块和文件目录的概念

文件属性

- 文件控制块（File Control Block）
  为管理文件而设置的数据结构，保存管理文件所需的所有有关信息
  （文件属性或元数据）

- 常用属性
  文件名，文件号，文件大小，文件地址，创建时间，最后修改时间，最后访问时间，保护，口令，创建者，当前拥有者，文件类型，共享计数，各种标志(只读、隐藏、系统、归档、ASCII/二进制、顺序/随机访问、临时文件、锁)

基本文件操作

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek
- Get Attributes
- Set Attributes
- Rename
- ... ...

文件的读写指针，定位在文件的某一个位置
还有我们要获取文件的属性，我们要设置文件的属性

```
/* File copy program. Error checking and reporting is minimal. */

#include <sys/types.h>                  /* include necessary header files */
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]);       /* ANSI prototype */

#define BUF_SIZE 4096                   /* use a buffer size of 4096 bytes */
#define OUTPUT_MODE 0700                /* protection bits for output file */

int main(int argc, char *argv[])
{
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if (argc != 3) exit(1);             /* syntax error if argc is not 3 */

    /* Open the input file and create the output file */
    in_fd = open(argv[1], O_RDONLY);    /* open the source file */
    if (in_fd < 0) exit(2);             /* if it cannot be opened, exit */
    out_fd = creat(argv[2], OUTPUT_MODE); /* create the destination file */
    if (out_fd < 0) exit(3);            /* if it cannot be created, exit */
```

用基本文件操作构造其他操作示例

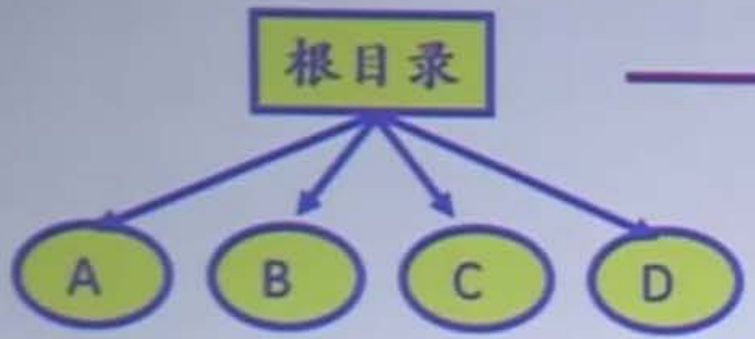
```
/* Copy loop */
while (TRUE) {
        rd_count = read(in_fd, buffer, BUF_SIZE); /* read a block of data */
if (rd_count <= 0) break;                         /* if end of file or error, exit loop */
        wt_count = write(out_fd, buffer, rd_count); /* write data */
        if (wt_count <= 0) exit(4);               /* wt_count <= 0 is an error */
}

/* Close the files */
close(in_fd);
close(out_fd);
if (rd_count == 0)                                /* no error on last read */
        exit(0);
else
        exit(5);                                  /* error on last read */
}
```

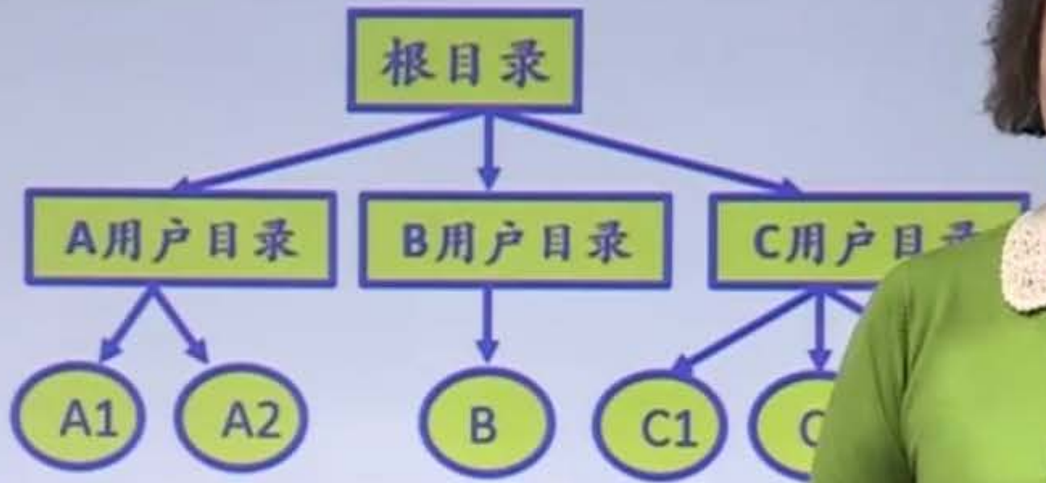
文件目录、目录项与目录文件

- 文件目录
  - 统一管理每个文件的元数据，以支持文件名到文件物理地址的转换
  - 将所有文件的管理信息组织在一起，即构成文件目录
- 目录文件
  - 将文件目录以文件的形式存放在磁盘上
- 目录项
  - 构成文件目录的基本单元
  - 目录项*可以*是FCB，目录是文件控制块的有序集合

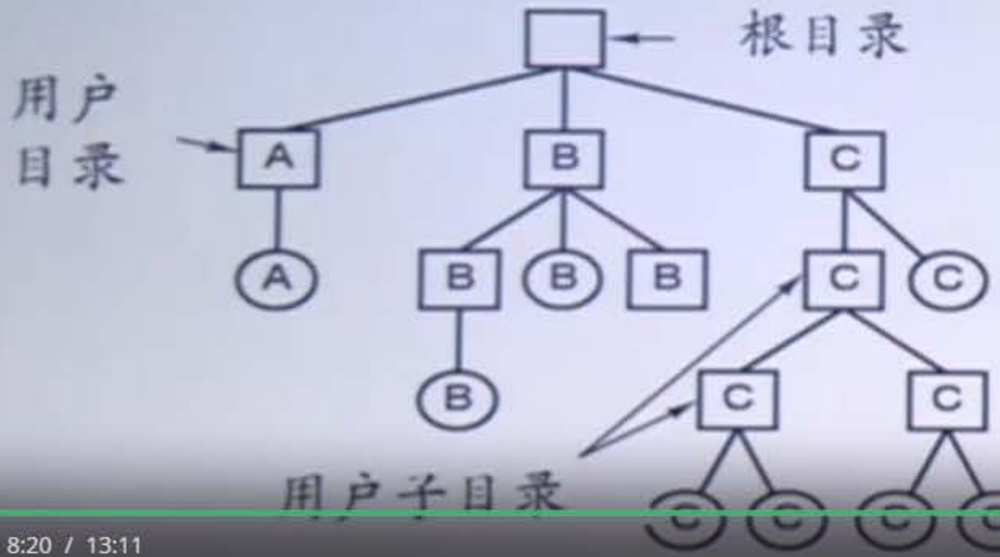
文件目录结构的演化



一级目录结构



二级目录结构



树形目录结构

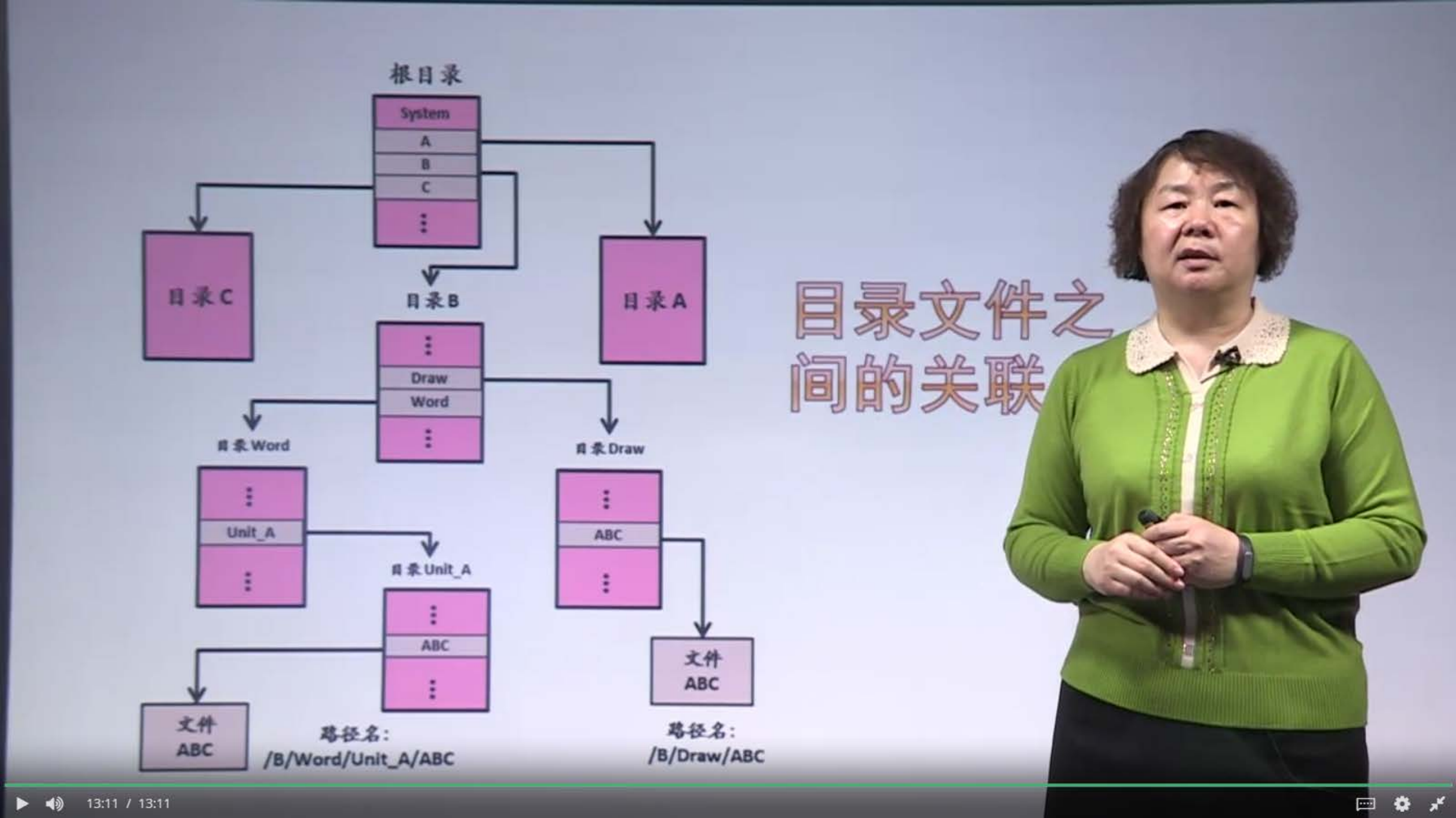
与目录相关的概念

- 路径名（文件名）
  - 绝对路径名：从根目录开始
  - 相对路径名：从当前目录开始
- 当前目录/工作目录
- 目录操作
  - 创建目录、删除目录
  - 读目录、写目录、改名、复制

当然了，在目录上 也可以提供各种各样的操作，创建目录，删除目录等等等等

目录文件之
间的关联

下面我们来介绍一下，目录文件之间的关联


根目录
System
A
B
C
目录C
目录B
目录A
Draw
Word
目录Word
目录Draw
Unit_A
目录Unit_A
ABC
ABC
文件
ABC
文件
ABC
路径名:
/B/Word/Unit_A/ABC
路径名:
/B/Draw/ABC
目录文件之
间的关联

13:11 / 13:11

0:00 下面我们来介绍文件控制块和文件目录的概念 操作系统为了管理文件 会把文件的各种属性都记录下来 因此，这些属性是操作系统 管理文件所需要的信息 按照我们对进程的这样一个设计 我们就把文件属性所存放的这些信息呢，把它称之为文件控制块 也就是操作系统为了管理文件，设置了一个专门的数据结构 在这个数据结构里头，保存了你要管理这个文件所需要的各种各样的信息 那么文件控制块呢，就是 简称就是 FCB ，有的时候呢，我们也称之为文件的属性 或者是叫做元数据，那么 元数据呢 就是指的是数据的数据 那么我们为了管理文件，我们又有一些相应的这个 信息，而这些信息呢 又以文件的形式保存在磁盘上，所以呢 就是元数据 元数据。 文件控制块 FCB 应该包含哪些信息呢？ 我们可以看一下，无外乎就是这些，比如说文件名 文件号呢，就有点相当于说 统一唯一标识一个文件的一个信息 文件的大小 文件的地址，这个地址呢 就指的是这个文件的 内容在磁盘上怎么能找到，通过这个文件地址可以找到 还有文件的创建的时间，修改的时间 和有最后一次访问的时间等等 我们还可以看到呢 这样一些各种各样的标志，比如说这个文件是只读啊，这个文件是隐藏的等等等等 所以我们在使用一般的操作系统过程中，我们会 看到这些一些设置属性的一些这个窗口 好，这是文件的属性 那么对于文件的操作有哪些呢？我们这里列举出了一些基本的文件操作 包括创建文件，删除文件 打开文件，关闭文件，读文件，写文件 当然还有一些比如说像 Seek 就是将 文件的读写指针，定位在文件的某一个位置 还有我们要获取文件的属性，我们要设置文件的属性 另外呢，我们可以给文件改名，这里头没有全，列全，比如说它可以拷贝文件 那么下面我们就以拷贝文件为例 说明一下如何用基本的文件操作，来构造新的操作 我们是用基本的文件操作，就是刚才我们列举出来的这些操作 来构造其他的操作，我们举一个例子，这个例子呢就是拷贝，拷贝文件 那么代码的话呢 大家自己回去慢慢看，当然我们主要去介绍一下它的主体的流程 那么要拷贝一个文件，就是首先要有一个 被拷贝的文件，我们要把这个文件呢 已经存在了，我们要打开它 你把一个文件拷贝到一个新的文件里头，那我们要创建一个文件，所以我们利用了 打开文件的基本操作，创建文件基本操作，我们来建立新的文件，把 原来这个要拷贝的文件呢打开，然后呢我们要做的事情就是 读这个原来有的文件，把这个内容读出来之后呢，写在了 新的文件里头，那么经过了若干个循环，啊，我们就把文件的内容都拷贝到新的文件里头了 最后呢 ，我们就要把这个文件呢关闭，这就是我们 利用前面的基本的文件操作，就可以构造出 Copy 这样一个新的操作，大家也可以想一下，如果我用 Readname 也用基

本文件操作搭建呢，那么这里头还要增加一个重要的过程 那就是要删除原来的这个文件 就是这样就相当于 Readname，当然我们是说 这是用已有的这个 文件的基本操作来搭建一个新的这个操作 当然我们也可以直接实现新的操作 那么下面我们来看一下非常非常重要的概念就是文件目录 目录项和目录文件，什么是文件目录呢？ 文件目录实际上是操作系统当中，统一管理每个文件的元数据 以支持文件名到 文件物理地址转换的这么一个重要的这个概念 所谓文件目录实际上就是将所有 文件的管理信息，把它组织在一起，就构成了文件目录 当然了，怎么组织，待会儿我们会看到有不同的组织方式 那么这些管理的信息，我们把它有机的组织在一起 然后我们要把它将文件的形式 按照文件的形式，存放在磁盘上 那么这个文件它的内容，实际上是文件目录，因此呢这个文件我们就称之为目录文件 那么目录文件呢，前面我们讲过它的 逻辑结构呢，应该是记录式结构，也就是 构成文件的 或者说构成目录文件的基本单位就是目录项 那么目录项究竟是什么东西呢？ 目前为止，我们可以认为目录项就是 FCB 也可以说目录就是 FCB 的一个有序的集合，啊，有序的集合 好，那么我们这一页的概念是非常重要的 特别是大家要搞清楚，目录文件它的逻辑结构 是由目录项，若干目录项组成的 那么刚才我们说了，这个文件的目录结构是有一个演化过程的 最早我们是一级目录结构 也就是说所有的文件都不分 子目录，子文件夹，都是在根底下建立文件 当然这样很不方便，我们起了一个文件名， 级和其他的用户起的文件名重名 后来呢就演化到了二级目录结构 那么也就是每个用户有自己的一个文件夹，或者是目录 那么你的文件名就不容易重了，因为你是自己起文件名 这是二级，当然二级呢 只是我们现在呢 就已经演化到了多级树形目录结构 树形目录结构，其实我们最熟悉的，在根底下，可以创建子目录 然后再创建子目录，或者是叫文件夹 然后呢还可以创建一些文件，所以现在我们常用的文件系统呢，都是树形结构 下面我们探讨一下与这个目录相关的几个概念 因为我们采用的是一个树形结构，所以我们的文件名呢 实际上就扩展为叫做路径名 因为我们是一棵树，所以这个路径名呢可以从根目录开始 来产生这个路径名，那么这样的话呢，就称之为绝对路径名 也可以从一个当前目录开始，也就是从这棵树的某一个 子根的结点开始往下查找这个文件，因此呢它是当前目录 那么这就叫做相对路径名，那么什么叫当前目录呢？ 当前目录呢 有的时候也称之为工作目录 实际上是这个用户，或者这个进程 当前正在使用的目录，我们平常在用 word 的时候 在用 powerpoint 做

一些这个 PPT 的时候 当我们要保存这个文件的时候，我们可以说另存为什么什么，这时候我们会发现它会 缺省的把你这个文件存放在某个目录 那么这个缺省的目录，其实就是这个用户或者这个进程的当前目录，也叫工作目录 当然了，在目录上 也可以提供各种各样的操作，创建目录，删除目录等等等等 下面我们来介绍一下，目录文件之间的关联 我们可以看到这是一个根目录文件 那么目录文件是由若干目录项组成的，所以我们可以看到这里头有 A B C 三个目录项，通过这个目录项，我可以找到这个目录项 所对应的那个文件，比如说我们可以看一下 A 目录项 我们就可以通过它找到 A 目录文件 B 目录项呢，我们就可以找到 B 目录文件，然后我们可以找到 C 目录文件 那我们再看看目录 B ，也就是 B 目录文件 那么 B 目录文件呢，也有两个 目录项，其他的我们没看到，我们画出了两个目录项 每一个目录项就会找到对应的目录文件 比如说 Draw 这个目录项，就能找到 目录 Draw 对应的那个目录文件 在这个 目录 Draw 底下呢，我们又可以啊，创建 一些其他的文件，比如说我们创建了一个文件 A B C 那么对 A B C 而言，在目录 Draw 里头，有一个 A B C 这个文件的目录项 然后通过这个目录项，我可以找到它对应的文件 那么 Word ，那么 Word 里头呢，这个 Word 这个目录文件呢，它的目录项是存放在 目录 B 的这个文件里头，通过 目录 B 当中的 Word 目录项，我们就找到了 Word 这个目录文件 而这个目录文件当中呢，我们又创建了一个啊 子目录，或者是文件夹，叫做 Unit_A 好，那么我们继续画下来就出现了一个 Unit_A 的这样一个 一个目录文件，而 Unit_A 的目录文件的目录项 就在我们说的 Word 这个目录文件里存放 在 Unit_A 这个目录底下，或者叫 文件夹里头，我们又创建了一个啊， A B C 文件 当然这也是允许的，因为它是在不同的目录底下创建的同名 文件，啊，名字相同，但是在不同的目录底下，因此它的路径名 是不一样的，那么我们就要把这层关系搞清楚 每一个方框，在我这里的方框，啊，实际上就是一个目录文件 目录文件呢，是由目录项组成的，所以我们可以看到目录文件里头有 若干个目录项，那么通过这个目录项，我们 可以找到这个目录项所对应的目录文件 这就是我们这一章教片希望大家能够掌握的