



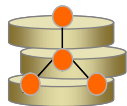
第十一单元（3）：文件的定位操作和随即读写

教学内容	文件的定位操作和随即读写
教学目标	
应知	<ul style="list-style-type: none">文件中位置指针的概念文件定位操作的意义文本文件的读写
应会	<ul style="list-style-type: none">能够将单个字符、字符串进行文件读写进行文件复制等程序的编写
难点	<ul style="list-style-type: none">文件的随机读写

1. 教学方法

项目拓展结合对比法

教学过程



1. 提问

- 文件指针的意义，引出文件随即读写
- 常用文件操作方式有哪几种，引出 w+, wb+, 从而引出文件定位
- 什么是文件
- C 语言的文件有什么特点
- feof()函数的作用是什么

2. 编程复习：将一个整型数组写入文件，再读取

3. 改进第 2 步，用 wb+的方式进行读写，不用两次打开、关闭文件，引入 rewind 函数

4. 拓展第三步，从文件中取出第 1、3、5、7、9 个数据，引入 fseek 函数

5. 改写第 4 步程序，从文件中读取第 2、4、6、8、10 个数据

6. 在第 4 步的基础上改变定位算法的实现即可



学生容易出错的地方

- ❏ 打开文件进行写之后，没有将文件位置指针指回文件
- ❏ 文件定位 `fseek` 函数的使用

问题与讨论

- ❏ 怎样实现随机文件读写



小结（可由问题与讨论方式给出）

- ❏ 文件定位 `rewind(fp)` 使文件位置指针重新指向文件头
- ❏ 文件定位函数 `fseek`(文件指针, 位移量, 起始点), 一般用于二进制文件, 因为文本文件要发生转换,

起始点: 0, 1, 2

`SEEK_SET`, `SEEK_CUR`, `SEEK_END`

文件头, 文件当前位置, 文件尾

位移量: 从起始点开始, 向前移动的字节数, ANSI C 规定在数字末尾加 L, 表示 long 型, 这样当文件长度大于 64K 时不至于出问题

- ❏ 在当前文件位置指针处进行读写



课后任务

- ❏ 考虑完成实训项目单 5 的后几项