知识点总结

· 在C语言的文件操作中，rewind和fseek函数用于对文件位置指示器进行定位，它们均被定义在stdio库里。借助这两个函数，程序能够灵活地在文件的不同位置进行读写操作。

· rewind函数：用于将文件位置指示器重新定位到文件开头。

· 调用rewind函数的语法格式：rewind(文件指针);

其中，“文件指针”指向关联着已打开文件的FILE结构体。

· 函数返回值：rewind函数没有返回值。

例：借助rewind函数将已读文件的位置指示器重新定位到文件开头，以便在起始处写入新数据。

```c
#include <stdio.h>

int main() {
    FILE* fp = fopen("example.txt", "r+");
    if (fp == NULL) {
        printf("无法打开文件\n");
        return 1;
    }

    // 从文件中读取一些数据
    char ch;
    while ((ch = fgetc(fp))!= EOF) {
        printf("%c", ch);
    }

    // 将文件位置指示器重新定位到文件开头
    rewind(fp);

    // 向文件中写入新的数据
    int result = fprintf(fp, "This is new data.");
    if (result < 0) {
        printf("写入失败\n");
        fclose(fp);
        return 1;
    }

    if (fclose(fp) != 0) {
        printf("关闭文件失败，数据可能未写入磁盘\n");
        return 1;
    }
    printf("写入成功\n");
    return 0;
}
```

· fseek函数：用于将文件指针移动到文件中的指定位置，从而实现对文件的随机访问。

· 调用fseek函数的语法格式：fseek(文件指针, 偏移量, 起始位置);

其中：

“文件指针”指向关联着已打开文件的FILE结构体。

“偏移量”是一个long类型整数，以字节为单位，用于指定文件位置指示器从起始位置开始要移动的距离。这个值可以是正数（表示向文件末尾方向移动）、负数（表示向文件开头方向移动）或零（表示不移动）。

“起始位置”：只可以取以下三个宏定义（均被定义在stdio库里）的值：

SEEK_SET：其对应的值为0，表示起始位置为文件开头。

SEEK_CUR：表示起始位置为文件位置指示器的当前位置。

SEEK_END：表示起始位置为文件末尾。

· 函数返回值：

如果文件位置指示器成功移动到指定位置，返回零。

如果操作失败（例如：偏移量超出文件范围、文件不支持随机访问等），返回非零值。

例：借助fseek函数读取从文件末尾开始前5个字节的内容。

```c
#include <stdio.h>
int main() {
    char str[6];
    FILE* fp = fopen("example.txt", "r");
    if (fp == NULL) {
        printf("无法打开文件\n");
        return 1;
    }

    // 将文件位置指示器移动到文件末尾前5个字节的位置
    if (fseek(fp, -5, SEEK_END) == 0) {
        size_t result = fread(str, 1, 5, fp);
        if (result != 5) {
            printf("到达文件末尾或读取错误\n");
            fclose(fp);
            return 1;
        }
        // 在字符数组末尾手动添加字符串结束符 '\0'，以便将其作为字符串处理
        str[5] = '\0';
        printf("从文件末尾前5个字节读取的内容: %s\n", str);
    }
    else {
        printf("文件位置指示器移动失败\n");
    }

    fclose(fp);
    return 0;
}
```

· 注意：

确保“偏移量”的值在合理范围内，避免移动到文件的非法位置。

当从文件开头（SEEK_SET）开始移动时，要满足：“偏移量”≥0。

当从文件位置指示器的当前位置（SEEK_CUR）开始移动时，要满足：

当前位置 +“偏移量”≥0。

例如：当前文件位置指示器指向第10个字节处时，偏移量不能小于-10。

当从文件末尾（SEEK_END）开始移动时，要满足：文件大小 +“偏移量”≥0。

例如：文件大小为100字节时，SEEK_END的偏移量不能小于-100。

· 补充：

关于r+/rb+、w+/wb+和a+/ab+的区别，如下表所示：

	文件存在时	文件不存在时	文件位置指示器初始位置	读取能力	写入特性	修改已有内容
r+/rb+（读写模式）	保留原内容	打开失败	文件开头	任意位置	任意位置覆盖/修改	可以
w+/wb+（写读模式）	清空原内容	创建新文件	文件开头	任意位置	任意位置覆盖/修改	可以（但原内容已清空）
a+/ab+（追加读写模式）	保留原内容	创建新文件	文件末尾	任意位置（需先定位）	强制末尾追加	不可以

rewind函数用来把文件指针重新定位到文件的开头

题目1、函数rewind(fp)的作用是（A）

A．使文件读写指针指向文件开始位置

B．使文件位置指针指向文件的末尾

C．使文件位置指针移至前一个字符的位置

D．使文件位置指针移至下一个字符的位置

题目2、有以下程序:

```
#include <stdio.h>
main()
{
    FILE *fp;
    char str[10];
    fp = fopen("myfile.dat", "w");
    fputs("abc", fp);
    fclose(fp);
    fp=fopen("myfile.dat", "a+");
    fprintf(fp, "%d", 28);
    rewind(fp);
    fscanf(fp, "%s", str);
    puts(str);
    fclose(fp);
}
```

程序运行后的输出结果是（C）

A. abc

B. 28c

C. abc28

D. 因类型不一致而出错

题目3、有以下程序:

```
#include <stdio.h>
main()
{
    FILE *fp;
    int i, a[6] = {1, 2, 3, 4, 5, 6},
k;
    fp=fopen("data.dat", "w+b");
    fprintf(fp, "%d\n", a[0]);
    for(i = 1; i < 6; i++)
    {
        fseek(fp, 0L, 0);
        fscanf(fp, "%d", &k);
        fseek(fp, 0L, 0);
        fprintf(fp, "%d\n", a[i] += k);
    }
    rewind(fp);
    fscanf(fp, "%d", &k);
    fclose(fp);
    printf("%d\n", k);
}
```

程序的运行结果是（A）

A. 21

B. 6

C. 123456

D. 11

题目4、有以下程序:

```
#include <stdio.h>
main()
{
    FILE *pf;
    char *s1 = "China", *s2 = "Beijing";
    pf = fopen("abc.dat", "wb+");
    fwrite(s2, 7, 1, pf);
    rewind(pf); /*文件位置指针回到文件开头*/
    fwrite(s1, 5, 1, pf);
    fclose(pf);
}
```

以上程序执行后abc.dat文件的内容是（B）

A. China

B. Chinang

C. ChinaBeijing

D. BeijingChina