|  |
| --- |
| 北京交通大学 |
| 实验报告(2) |
| 文件、链表 |

|  |
| --- |
| 殷盛宗  2018-3-24 |

目录

[1. typedef的作用 2](#_Toc509692521)

[2. malloc的具体功能 3](#_Toc509692522)

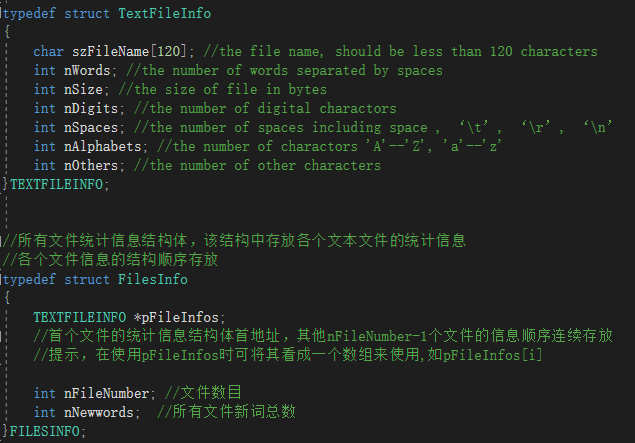
[3. CountWordsOfEuropeanTxtFile函数分析 4](#_Toc509692523)

[4. fprintf的功能 5](#_Toc509692524)

[5. 实验二函数调用关系表 6](#_Toc509692525)

[6. 心得体会 6](#_Toc509692526)

# typedef的作用

在本实验中，typedef主要起定义抽象数据类型的作用：

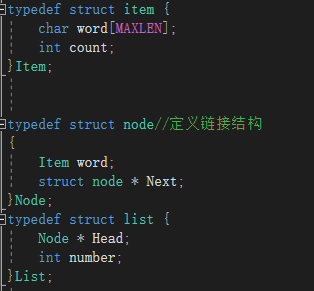
typedef的本质是给类型起别名。例如，

Figure ‑2

Figure ‑1

typedef double Real;

就定义了“实数型”，用双精度浮点数实现。

但是，双精度浮点数并不能表示所有的实数，将double称作real其实是“虚伪的“，至少是不正确的。

所以，虽然typedef在程序中的作用仅仅是给某个类型起了别名，但这两个名字在概念上指代的不是同一个事物。前者是抽象数据类型的实现的名字，后者是抽象数据类型本身的名字——double是实型在程序中的实现，而real是不依赖于程序的数学中的抽象类型。这二者之间的关系有些类似于“法院“和”公平正义“、”学校“和”教育“、”医院”和“医疗”。

同一个抽象类型可以有多种不同的实现，实现的选择需要考虑要解决的问题的要求，同时受到计算机硬件的限制。在计算机硬件改变时，可能需要调整抽象类型的实现。使用typedef可以减少这种调整的工作量，有时甚至只需更改头文件中的定义。

# malloc的具体功能

malloc，即memory allocate（内存分配）。它的功能就是动态内存分配，即，在运行过程中为变量分配内存。

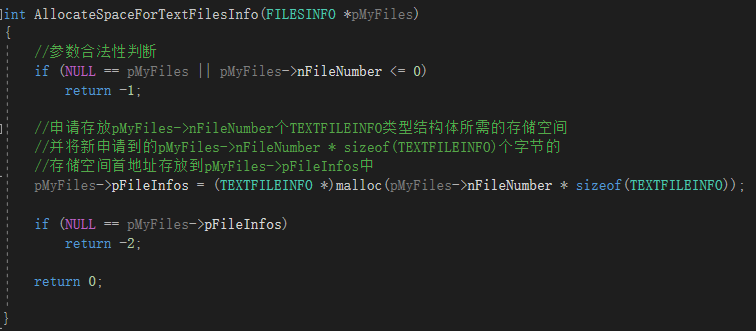
如图，该函数给指向存放单个文件信息的结构体TEXTFILEINFO的指针pFileInfos分配了等于文件数目的连续内存,形成了一个结构体数组.这个数组的元素个数不是在编译时确定的,而是在运行过程中由用户输入的.可以看出,malloc的使用提高了程序的灵活性,减少了内存空间的浪费。

Figure ‑

另外，动态内存分配在构建链表时必不可少。

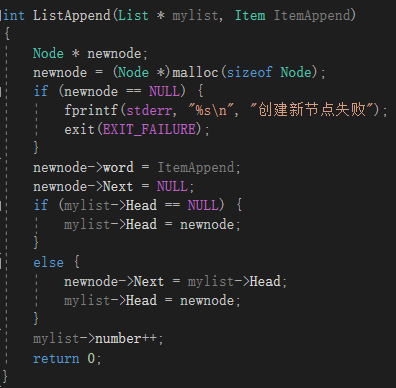
当需要向链表中添加项时，动态分配一段内存并将其于链表链接。

Figure ‑

为防止内存泄漏，malloc后必须free。

# CountWordsOfEuropeanTxtFile函数分析

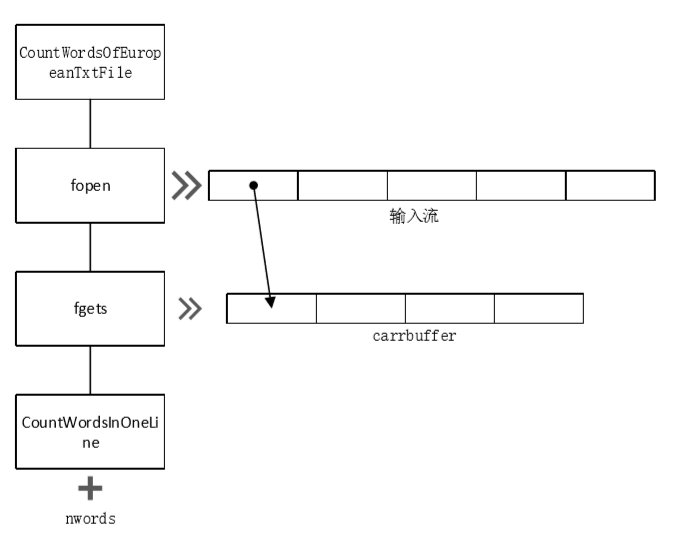
该函数的函数流程图如下：

Figure ‑

如图，fopen创建一个流，并把一个文件指针与之连接，使fgets从中得到数据。fclose的作用是删除文件指针指向的流，从而断开文件指针和文件的联系。

# fprintf的功能

函数ExportFilesInfoToWebPage大量调用了fprintf。

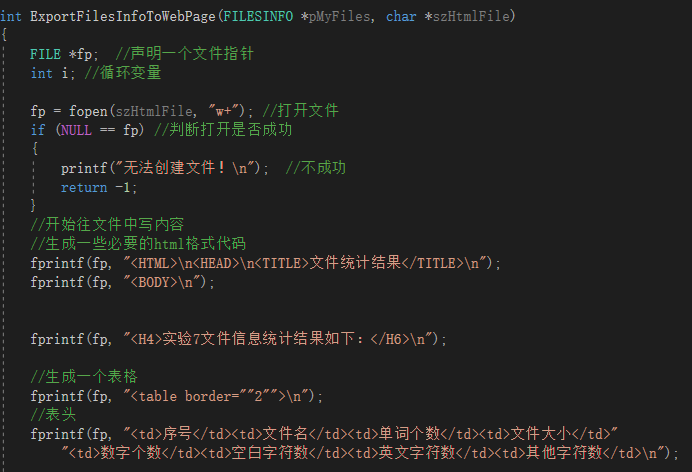
fprintf的功能是将指定格式的内容输出到文件。这里，向该文件中输出html指令，从而生成一个网页文件。可以用浏览器打开该文件。

Figure ‑

# 实验二函数调用关系表

Figure ‑

# 心得体会

我通过对样例代码的补充，完全实现了约定的功能。

这次实验，无论从规模还是实现技术上都有一定难度。如果不能理解工程的概念、理清函数之间的调用关系，实验过程将举步维艰。

运用链表做词频统计无疑是这次实验中的难点。这项任务主要难在两个方面，一个是链表及相关操作本身的实现，一个是在工程中应用自己编写的链表。无论是链表的实现出错还是接口设计不当，都会使目标功能无法实现。

文件操作是本次实验的重点，但不是难点。倒是老师给出的输出网页的函数令人眼前一亮，对“什么是网页？”的理解深入了一层。

大规模代码的debug是一项对智力和耐力的双重考验。我在后期的调试中发现，自己一口气写出的几个链表操作函数几乎个个都有问题。但这时我已经在多个函数中调用了这些问题函数，修改它们令人身心俱疲。由此得知，没有事先良好的设计、没有为每个子函数编写的驱动程序，大型程序的编写过程绝不会一帆风顺。

遗憾地是，我最终也没能实现调用库函数qsort对链表排序。qsort函数运行迅速，接口设计也极具启发性，不过只对连续储存的变量有效。这意味着不能直接用qsort对链表排序。我的对策是将链表的每个节点的指针保存在数组中，用qsort对该指针数组排序，最后根据该数组重构链表。由于未知的错误，这个倾注了我大量时间精力的方案至今未能实现。上传的程序使用了冒泡排序的替代方案。