

电子科技大学信息与软件工程学院

实 验 报 告

学 号 2018091605010

姓 名 梁嘉城

(实验) 课程名称 程序设计与算法基础

理论教师 傅翀

实验教师 王瑞锦

电 子 科 技 大 学

实 验 报 告

学生姓名：梁嘉城 学号：2018091605010 指导教师：王瑞锦

实验地点：537 实验时间：2019.01

一、 实验名称：超市商品管理系统链表实现

二、 实验学时：4 学时

三、 实验目的

掌握单链表的定义和使用方法

掌握单链表的建立方法

掌握单链表中节点的查找与删除

掌握输出单链表节点的方法

掌握链表节点排序的一种方法

掌握 C 语言创建菜单的方法

掌握结构体的定义和使用方法

四、 实验原理

1. **关于商品信息的结构体**：包含商品的 ID、商品名称、商品价格、商品折扣、商品总数量、商品剩余数量 的不同变量的结构体。

2. **结构变量的声明**：

在商品信息的结构体中：商品 ID、商品名字、商品折扣 为字符变量；商品价格、商品总数量、商品剩余数量为整型变量

```
typedef struct {
    char    good_id[MAX_ID_LEN];
    char    good_name[MAX_NAME_LEN];
    int     good_price;
    char    good_discount[MAX_DISCOUNT_LEN];
    int     good_amount;
    int     good_remain;
}GoodInfo;
```

3. 每个链表的节点都有一个商品的结构。

```
typedef struct node
{
    GoodInfo data;
    struct node *next;
}GoodList;
```

4. 系统界面控制以及商品信息的初始化:

建立一个 txt 文件（名称为 goodinfo.txt）包含超市商品信息，每种商品包含下述信息：商品 ID、商品名称、商品价格、商品折扣、商品总数以及商品剩余数目的信息，每种信息之间用\t 分隔开。

5. 对链表的操作:

定义两个指针，一个是 prev，另一个是 cur，prev 指向的结点一直在 cur 指向的节点之前，通过这两个指针从前往后逐次移动读取链表里每个节点的信息，通过 strcmp 函数进行比较判断再决定是否对内容进行操作。这两个一前一后的指针基本可以完成所有函数的功能，主要用于插入、修改、查找、排序。

```
GoodList *prev, *cur;
prev = (*L);
cur = prev->next;
GoodList *new_node, *cur, *prev;
```

6. 结构类型的定义:

使用 typedef 定义结构

```
typedef struct {
    char    good_id[MAX_ID_LEN];
    char    good_name[MAX_NAME_LEN];
    int     good_price;
    char    good_discount[MAX_DISCOUNT_LEN];
    int     good_amount;
    int     good_remain;
}GoodInfo;
```

7. **链表存储结构**：在这个链表中，每一个结点只指向了一个结构，还是如上所述，我使用了两个指针 prev 与 cur （prev 一直指向 cur 的前面）。通过指针的移动来扫描整个链表的结点。

8. **链表结点的类型**：链表结点的类型我使用 data 表示，最后一个和第一个的指针都是 NULL。

```
typedef struct node
{
    GoodInfo data;
    struct node *next;
}GoodList;
```

9. **单链表的头结点**：我把头结点命名为 L 。L 的数据域为空，头结点的指针域指向第一个元素的结点。

```
GoodList *L;
L = (GoodList*)malloc(sizeof(GoodList));
L->next = NULL;
```

10. **链表的基本操作函数**：

- a) void info_change()完成商品信息的修改功能。
- b) info_dele()完成商品信息的删除功能
- c) void info_search()完成商品信息的查找功能。
- d) void info_insert()完成商品信息的插入功能。
- e) 函数 void bubble_sort () , 用冒泡法实现商品信息按照价格从低到高进行排序展示。

f) 函数 void info_flush()该函数把所有商品信息（链表中的所有商品）写入到已有的商品信息文件中。

11. **系统选择界面：**使用 for (;;)语句执行无限循环。当用户输入 6 时，调用 info_flush()函数，循环结束，商品存盘并退出系统。

然后对输入的数字进行讨论，分别调用不同的函数来完成用户的目的。

```
for (;;) {
    printf(
        "\n"
        "***梁嘉城 2018091605010 github.com/jackpurcell***\n"
        "***欢迎访问超市管理系统***\n"
        "\033[33m*****\n\033[0m"
        "\033[32m1.显示所有商品的信息:\n\033[0m"
        "\033[32m2.修改某个商品的信息:\n\033[0m"
        "\033[32m3.插入某个商品的信息:\n\033[0m"
        "\033[32m4.删除某个商品的信息:\n\033[0m"
        "\033[32m5.查找某个商品的信息:\n\033[0m"
        "\033[32m6.商品存盘并退出系统:\n\033[0m"
        "\033[32m7.对商品价格进行排序:\n\033[0m"
        "\033[32m8.(慎用)删除所有内容:\n\033[0m"
        "\033[32m其他.不存盘并退出系统:\n\033[0m"
        "\033[33m*****\n\033[0m"
        "输入你的选择:");
    scanf(" %c", &code);
    while (getchar() != '\n')
        ;
    switch (code) {
        case '1': OutputAll(L); //完成商品信息的输出功能
            break;
        case '2': info_change(&L); //完成商品信息的修改功能
            break;
        case '3': info_insert(&L); //完成商品信息的插入功能。
            break;
        case '4': info_dele(&L); //完成商品信息的删除功能
            break;
        case '5': info_search(L); //完成商品信息的搜索功能
            break;
        case '6': info_flush(&L); //该函数把所有商品信息（链表中的所有商品）写入到已有的商品信息文件中
            break;
        case '7': bubble_sort(&L); //用冒泡法实现商品信息按照价格从低到高进行排序展示
            break;
        case '8': DelAll(&L); //删除系统所有内容
            break;
        default: exit(0);
    }
}
```

12. **文件操作示例：**

对于文件的操作：

关于文件的打开：`FILE *fp = fopen("goodinfo.txt", "r");` 定义一个指针

`FILE *fp` 指向该文件。

使用 `fscanf` 读取并且写入文件。

```
fscanf(fp, "%s", &new_node->data.good_id);
fscanf(fp, "\t%s", &new_node->data.good_name);
fscanf(fp, "\t%d", &new_node->data.good_price);
fscanf(fp, "\t%s", &new_node->data.good_discount);
fscanf(fp, "\t%d", &new_node->data.good_amount);
fscanf(fp, "\t%d\n", &new_node->data.good_remain);
```

关于文件的关闭：`fclose(fp);`

五、 实验内容

用 C 语言+链表数据结构实现一个小型的超市商品管理系统，该系统需要具备商品信息录入、商品信息修改、商品信息删除、商品信息查找、商品信息的插入等这几个功能。具体实现步骤如下（注：图示为建议显示内容和格式，可自行增加显示内容）：

- 1) 软件界面控制:实现一个数字选项式的启动界面，其中至少包含显示所有商品信息、商品信息插入、商品信息修改、商品信息删除、商品信息查找、商品价格排序、退出系统并保存 7 个选项。并且这些功能可以循环调用。
- 2) 商品信息的初始化: 定义链表并初始化。实现从已有的商品信息文件中读入商品信息，并且分配内存保存至链表中。如 1) 里的图示所示，从文件中读取了 7 个商品记录。
- 3) 商品信息的增加: 实现一个函数完成单个商品信息的增加，接受用户的输入的各项信息，然后保存至链表结点。同时实现可以根据用户的输入，将该结点插入到列表的头部，尾部，或者中间 *i* 号位置。
- 4) 商品信息的修改: 实现一个函数完成商品信息的修改功能，实现可以根据商品的名称或者 ID 修改商品信息。其中用字符串比较的方式来查找待修改商品。

- 5) 商品信息的删除: 实现一个函数, 实现根据商品的名称来删除对应的商品信息的功能, 商品查找通过字符串比较的方式, 查找到后释放对应指针指向的内存区域, 完成删除。
- 6) 商品信息的查找: 实现一个函数, 函数的功能是根据输入的商品名称来查找对应的商品信息, 商品名称的判断用字符串比较的方式来实现, 然后调用格式化显示查找到的商品信息。
- 7) 所有商品信息的显示: 实现一个函数, 该函数的功能是将链表中所有的商品信息以格式化的方式打印输出到屏幕上。如下图所示的例子
- 8) 对商品按价格排序: 实现一个函数, 可以根据链表中的商品的价格, 对商品进行排序, 排序算法采用冒泡排序实现, 最后将排序后的链表打印至屏幕。
- 9) 退出系统, 并保存: 实现一个文件写入函数将所有信息的改动写入到商品信息文件, 0 然后清理系统运行过程中已分配的内存。

六、 实验器材 (设备、元器件)

笔记本电脑一台 (Microsoft Visual Studio), 显示屏一个

七、 实验步骤

- 1) 分别完成模板中注释为//TO DO YOUR WORK 的函数内容
- 2) 定义用于表示某种商品的所有信息的结构体, 并且定义链表结构用来组织所有的商品信息库中的商品信息,
- 3) 定义并实现一个函数: void info_init(GoodsList **L), **L 是链表的头节点指针的地址。该函数初始化一个链表, 然后完成从一个 txt 文件读入商品链表库并使用相应的商品信息来初始化商品库 (即初始化链表), 在读入的时候每读到一条商品信息就实时的动态分配内存来把信息放到分配得到的链表结点指针指向的内存单元中, 然后把链表节点加入到商品链表中。

- 4) 定义并实现函数: `void info_flush(GoodsList **L)`, `**L` 是链表的头节点指针的地址。该函数完成将系统运行期间改动过的商品信息库写回到存放商品信息的 txt 文件中, 然后销毁商品链表。
- 5) 定义并实现函数: `void OutputAll(GoodsList *L)`, `*L` 是链表头节点指针。以格式化的方式, 完成将商品链表中的每项信息打印到标准输出(即屏幕上)。
- 6) 定义并实现函数: `void info_output(GoodsList *p)`, 格式化输出某一项商品信息, 其中 `p` 是该商品结点指针。
- 7) 定义并实现函数: `void info_change(GoodsList **L)`, `**L` 是链表的头节点指针地址。该函数完成商品信息的修改功能, 其中要求用户输入需要修改的某项商品的名称或者 ID, 然后对名称或者 ID 进行查找, 找到则继续输入该商品的各项信息, 并提示修改成功; 如果没有找到则提示对应商品未找到在两种选择后都返回到初始的选择菜单
- 8) 定义并实现函数: `void info_dele(GoodsList **L)`, `**L` 是链表的头节点指针地址。该函数完成删除某条商品库中信息的功能, 通过输入的某项商品的名称或者 ID 删除对应的信息, 如果在商品库中找到对应的商品便删除该商品信息(即释放指针所指向的内存, 并把该指针赋值为 `NULL`, 同时完成链表的前后重新衔接), 并提示删除成功; 如果没有找到该商品要提示没有找到该商品信息, 在两种选择下都回到上一步选择界面
- 9) 定义并实现函数: `void info_search(GoodsList *L)`, `*L` 是链表头节点指针。该函数完成商品信息的查找功能, 然后通过输入某种商品信息的名称来检索商品信息库, 查找到则显示该商品的详细信息, 没有查找到则提示没有该商品

- 10) 定义并实现函数: `void info_insert(GoodsList **L)`, `**L` 是链表的头节点指针地址。该函数完成商品信息的插入, 在插入之前动态的分配内存用来存储插入的商品信息, 然后把指向该内存的指针加入到链表中, 加入链表时需要选择在头部插入或者在尾部插入, 或者在链表范围内的 `i` 号位置插入; 在插入之前必须考虑整个信息库的限定容量, 如果超过上限要给用户以提示
- 11) 定义并实现函数: `bubble_sort(GoodsList **L)`, `**L` 是链表的头节点指针地址。该函数实现将链表中的所有商品按照商品价格从低到高排序。
- 12) 实现程序的入口函数即 `main` 函数, 然后通过一个条件为 1 的 `while` 循环完成以上功能的循环调用, 直至选择正常退出
- 13) 编译、调试程序直至达到实验要求

八、 实验结果与分析 (含重要数据结果分析或核心代码流程分析)

(部分内容已放置在实验原理部分)

- 1) 软件界面控制: 实现一个数字选项式的启动界面, 其中至少包含显示所有商品信息、商品信息插入、商品信息修改、商品信息删除、商品信息查找、商品价格排序、退出系统并保存 7 个选项。并且这些功能可以循环调用。

```

商品的链表文件已建立，有3个商品记录

***梁嘉城 2018091605010 github.com/jackpurcell***
***欢迎访问超市管理系统***
*****
1. 显示所有商品的信息:
2. 修改某个商品的信息:
3. 插入某个商品的信息:
4. 删除某个商品的信息:
5. 查找某个商品的信息:
6. 商品存盘并退出系统:
7. 对商品价格进行排序:
8. (慎用)删除所有内容:
其他. 不存盘并退出系统:
*****
输入你的选择:

```

- 2) 商品信息的初始化: 定义链表并初始化。实现从已有的商品信息文件中读入商品信息，并且分配内存保存至链表中。如 1) 里的图示所示，从文件中读取了 3 个商品记录。
- 3) 商品信息的增加: 完成单个商品信息的增加，接受用户的输入的各项信息，然后保存至链表结点。同时实现可以根据用户的输入，将该结点插入到列表的头部，尾部，或者中间 i 号位置。该处添加特色功能，当输入的 i 值超过商品数量，无法插入到该位置时，程序会自动插入到尾部。

```

输入数字表明你要插入的商品位置: 0. 商品列表尾部 1. 商品列表头部 i. 商品列表中第i号位置100
无法在该位置插入!自动插入到尾部!插入成功!

```

```

输入你的选择:3
输入新的商品信息: 商品ID: 001
商品名称: NIKE
商品价格: 1399
商品折扣: 0.9
商品数量: 100
商品剩余: 100
输入数字表明你要插入的商品位置: 0. 商品列表尾部 1. 商品列表头部 i. 商品列表中第i号位置1
插入成功!

```

- 4) 商品信息的修改:完成商品信息的修改功能，实现可以根据商品的名称或者 ID 修改商品信息。其中用字符串比较的方式来查找待修改商品。该处添加了特色功能，输入 -1 会自动退出修改，返回主程序。

```

输入你的选择:2
输入需要修改的商品ID(-1退出修改): -1
已退出!

```

```

输入你的选择:2
输入需要修改的商品ID(-1退出修改): 002

+++++
ID:002  名称:MACBOOKPRO  价格:13999      折扣:0.9      数量:100      剩余:100
+++++
输入新的商品信息: 商品ID: 008
商品名称: MACBOOKPRO13
商品价格: 15999
商品折扣: 0.9
商品数量: 99
商品剩余: 99
修改成功!

```

- 5) 商品信息的删除:根据商品的名称来删除对应的商品信息的功能，商品查找通

过字符串比较的方式，查找到后释放对应指针指向的内存区域，完成删除。

该处添加了特色功能，输入-1 会自动退出删除，返回主程序。

```

输入你的选择:4
输入需要删除的商品ID(-1退出删除): 003
是否删除该商品(Y/N)Y

删除成功ID为003的商品成功
当前剩余商品1个

```

- 6) 商品信息的查找:根据输入的商品名称来查找对应的商品信息，商品名称的判

断用字符串比较的方式来实现，然后调用格式化显示查找到的商品信息。该

处添加了特色功能，输入-1 会自动退出查找，返回主程序。

```

输入你的选择:5
输入需要查找的商品名称(-1退出查找): NIKE

+++++
ID:001  名称:NIKE      价格:1399      折扣:0.9      数量:100      剩余:100
+++++

```

- 7) 所有商品信息的显示：将链表中所有的商品信息以格式化的方式打印输出到

屏幕上。

```

输入你的选择:1

+++++
ID:001  名称:NIKE      价格:1399      折扣:0.9      数量:100      剩余:100
+++++
+++++
ID:002  名称:MACBOOKPRO  价格:13999      折扣:0.9      数量:100      剩余:100
+++++
+++++
ID:003  名称:ADIDAS      价格:1399      折扣:0.9      数量:100      剩余:100
+++++

```

- 8) 对商品按价格排序：实现一个函数，可以根据链表中的商品的价格，对商品

进行排序，排序算法采用冒泡排序实现，最后将排序后的链表打印至屏幕。

```

输入你的选择:1
+++++
ID:003  名称:ADIDAS    价格:1399    折扣:0.9    数量:100    剩余:100
+++++
ID:001  名称:NIKE     价格:1399    折扣:0.9    数量:100    剩余:100
+++++
ID:002  名称:MACBOOKPRO 价格:13999    折扣:0.9    数量:100    剩余:100
+++++

```

9) 退出系统, 并保存:将所有信息的改动写入到商品信息文件, 然后清理系统运

行过程中已分配的内存。

```

输入你的选择:6
3个商品信息存入Goodinfo.txt文件

```

```

goodinfo.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
001      NIKE      1399      0.9      100      100
002      MACBOOKPRO 13999    0.9      100      100
003      ADIDAS    1399      0.9      100      100

```

10) 删除所有内容: 删除链表及 goodinfo.txt 文件, 释放内存, 并建立新的链表头

```

输入你的选择:8
删除成功

```

九、 总结及心得体会:

刚接触到这个实验时, 发现实验内容比较多, 实在是难以下手。我决定先完成最简单的 main 函数的编辑, 根据 ppt 中对函数的要求, 完成了对 main 函数的编译, 接着完成 read_line 和 Goodprint 等多次调用的函数的编辑。

随后, 就是 main 函数中引用的函数的编译, 这才是这个实验的难点, 书本的链表章节内容比较简短, 很多内容没有说透, 通过阅读《数据结构》的相关书籍, 对单链表有了更深的了解后, 才继续编写。在编译的过程中, 因为对新知识的不熟悉, 导致有时候信息没有正确地连接到链表上, 通过使用 Microsoft Visual Studio 的断点和局部变

量监视窗口功能，一步步修复 bug，再考虑到用户的各种操作，加入 if 判断语句，避免野指针等 bug 的出现。初期的坎坷，到后面几个函数的编辑时，我都事先用纸打好草稿，考虑到所有的情况，锻炼自己的编程逻辑思维，让自己的编程能力得到真正的提升。

这个实验的主要内容在于把商品信息储存在链表当中，链表中的每一个结点指向一个关于商品信息的结构。然后我主要调用两个指针，一个是 prev，另一个是 cur，prev 指针指向的位置总是在 cur 指向位置的前一位。通过两个指针的位置同时移动可以对链表进行扫描。扫描的时候使用 strcmp 函数与输入内容进行比较查看当前扫描结点是否为所需。另一个核心就在于建立一个临时的储存单位 new_code，把我们所需要插入的商品信息放入其中，再把 new_code 里面的信息根据用户需求放到相应的链表结点上。

最后，我完成了这个实验，并在用户主界面添加了颜色代码，让程序的主体部分更加显眼，方便用户识别。调试中莫名的错误，一定要耐心地修复，不可急躁，的确不行就推倒重来。这次实验我不仅编程能力得到了很大的提升，还懂得了成功的秘诀——就是坚持。最后把代码放在 GitHub 上开源分享，成就感满满。

十、 对本实验过程及方法、手段的改进建议：

该实验程序已经多次进行调试，确保不会出现错误。并添加多处特色功能，如 (-1) 退出查找、删除、修改返回主函数；用户界面添加颜色代码等，尽可能把用户的需求想到。

报告评分：

指导教师签字：