

链表的C语言实现之单链表的查找运算 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E9_93_BE_E8_A1_A8_E7_9A_84C_c97_134514.htm 建立了一个单链表之后，

如果要进行一些如插入、删除等操作该怎么办？所以还须掌握一些单链表的基本算法，来实现这些操作。单链表的基本运算包括：查找、插入和删除。下面我们就一一介绍这三种基本运算的算法，并结合我们建立单链表的例子写出相应的程序。

1、查找 对单链表进行查找的思路为：对单链表的结点依次扫描，检测其数据域是否是我们所要查好的值，若是返回该结点的指针，否则返回NULL。因为在单链表的链域中包含了后继结点的存储地址，所以当我们实现的时候，只要知道该单链表的头指针，即可依次对每个结点的数据域进行检测。 以下是应用查找算法的一个例子：

```
#include <stdio.h> #include <malloc.h> #include <string.h> /*包含一些字符串处理函数的头文件*/ #define N 10 typedef struct node { char name[20]. struct node *link.}stud. stud * creat(int n) /*建立链表的函数*/ { stud *p,*h,*s. int i. if((h=(stud *)malloc(sizeof(stud)))==NULL) { printf("不能分配内存空间!"). exit(0). } h->name[0]=\0. h->link=NULL. p=h. for(i=0.i < n.i) { if((s=(stud *) malloc(sizeof(stud)))==NULL) { printf("不能分配内存空间!"). exit(0). } p->link=s. printf("请输入第%d个人的姓名",i 1). scanf("%s",s->name). s->link=NULL. p=s. } return(h).} stud * search(stud *h,char *x) /*查找链表的函数，其中h指针是链表的表头指针，x指针是要查找的人的姓名*/ { stud *p. /*当前指针，指向要与所查找的姓名比较的结点*/
```

```
char *y. /*保存结点数据域内姓名的指针*/ p=h->link.  
while(p!=NULL) { y=p->name. if(strcmp(y,x)==0) /*把数据域里的  
姓名与所要查找的姓名比较，若相同则返回0，即条件成  
立*/ return(p). /*返回与所要查找结点的地址*/ else p=p->link.  
} if(p==NULL) printf("没有查找到该数据!").}main(){ int  
number. char fullname[20]. stud *head,*searchpoint. /*head是表头  
指针，searchpoint是保存符合条件的结点地址的指针*/  
number=N. head=creat(number). printf("请输入你要查找的人的  
姓名:"). scanf("%s",fullname). searchpoint=search(head,fullname).  
/*调用查找函数，并把结果赋给searchpoint指针*/} 100Test 下  
载频道开通，各类考试题目直接下载。详细请访问  
www.100test.com
```