

链表的建立、插入和删除 (二) PDF转换可能丢失图片或格式, 建议阅读原文

[https://www.100test.com/kao\\_ti2020/136/2021\\_2022\\_\\_E9\\_93\\_BE\\_E8\\_A1\\_A8\\_E7\\_9A\\_84\\_E5\\_c97\\_136223.htm](https://www.100test.com/kao_ti2020/136/2021_2022__E9_93_BE_E8_A1_A8_E7_9A_84_E5_c97_136223.htm)

7.4.2 单链表的插入与删除在链表这种特殊的数据结构中, 链表的长短需要根据具体情况来设定, 当需要保存数据时向系统申请存储空间, 并将数据接入链表中。对链表而言, 表中的数据可以依此接到表尾或连结到表头, 也可以视情况插入表中; 对不再需要的数据, 将其从表中删除并释放其所占空间, 但不能破坏链表的结构。这就是下面将介绍的链表的插入与删除。

1. 链表的删除在链表中删除一个节点, 用图7-4描述如下: [例7-6]

创建一个学生学号及姓名的单链表, 即节点包括学生学号、姓名及指向下一个节点的指针, 链表按学生的学号排列。再从键盘输入某一学生姓名, 将其从链表中删除。首先定义链表的结构:

从图7-4中看到, 从链表中删除一个节点有三种情况, 即删除链表头节点、删除链表的中间节点、删除链表的尾节点。题目给出的是学生姓名, 则应在链表中从头到尾依此查找各节点, 并与各节点的学生姓名比较, 若相同, 则查找成功, 否则, 找不到节点。由于删除的节点可能在链表的头, 会对链表的头指针造成丢失, 所以定义删除节点的函数的返回值定义为返回结构体类型的指针。

```
struct node *delet(head,pstr)以/*head 为头指针, 删除pstr 所在节点*/struct node *head.char *pstr.{struct node *temp,*p,temp = head./* 链表的头指针*/if (head==NULL) /*链表为空*/printf("\nList is null!\n").else /*非空表*/{temp = head.while (strcmp(temp->str,pstr)!=0&temp->next!=NULL)/* 若节点的
```

```
字符串与输入字符串不同，并且未到链表尾* /{p = temp.  
temp = temp -> next. /* 跟踪链表的增长，即指针后移*  
/}if(strcmp(temp->str,pstr)==0) /* 找到字符串*  
/{if(temp==head) { /* 表头节点* /printf("0delete string  
:%s\n",temp->str).head = head -> next.free(temp). /* 释  
放被删节点* /}else{p->next=temp->next. /表*中节  
点*/printf("0delete string :%s\n",temp->str).free(temp).}}else  
printf("\nno find string!\n").没/找* 到要删除的字符串*/}return  
(head). /* 返回表头指针* /} 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
```