

链表的C语言实现之单链表的实现 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/166/2021_2022__E9_93_BE_E8_A1_A8_E7_9A_84C_c97_166363.htm 一、单链表的建立

有了动态内存分配的基础，要实现链表就不难了。所谓链表，就是用一组任意的存储单元存储线性表元素的一种数据结构。链表又分为单链表、双向链表和循环链表等。我们先讲讲单链表。所谓单链表，是指数据接点是单向排列的。一个单链表结点，其结构类型分为两部分：1、数据域：用来存储本身数据 2、链域或称为指针域：用来存储下一个结点地址或者说指向其直接后继的指针。例：typedef struct node{ char name[20]. struct node *link;}stud. 这样就定义了一个单链表的结构，其中char name[20]是一个用来存储姓名的字符型数组，指针*link是一个用来存储其直接后继的指针。定义好了链表的结构之后，只要在程序运行的时候在数据域中存储适当的数据，如有后继结点，则把链域指向其直接后继，若没有，则置为NULL。下面就来看一个建立带表头（若未说明，以下所指链表均带表头）的单链表的完整程序。

```
#include <stdio.h> #include <malloc.h> /*包含动态内存分配函数的头文件*/ #define N 10 /*N为人数*/ typedef struct node{ char name[20]. struct node *link;}stud. stud * creat(int n) /*建立单链表的函数，形参n为人数*/ { stud *p,*h,*s. /* *h保存表头结点的指针，*p指向当前结点的前一个结点，*s指向当前结点*/ int i. /*计数器*/ if((h=(stud *)malloc(sizeof(stud)))==NULL) /*分配空间并检测*/ { printf("不能分配内存空间!"). exit(0). } h->name[0]='\0. /*把表头结点的数据域置空*/ h->link=NULL.
```

```
/*把表头结点的链域置空*/ p=h. /*p指向表头结点*/ for(i=0.i  
< n.i ) { if((s= (stud *) malloc(sizeof(stud)))==NULL) /*分配新存  
储空间并检测*/ { printf("不能分配内存空间!"). exit(0). } p-  
> link=s. /*把s的地址赋给p所指向的结点的链域，这样就把p  
和s所指向的结点连接起来了*/ printf("请输入第%d个人的姓  
名",i 1). scanf("%s",s- > name). /*在当前结点s的数据域中存储姓  
名*/ s- > link=NULL. p=s. } return(h).} main(){ int number. /*保存  
人数的变量*/ stud *head. /*head是保存单链表的表头结点地址  
的指针*/ number=N. head=creat(number). /*把所新建的单链表  
表头地址赋给head*/} 这样就写好了一个可以建立包含N个人  
姓名的单链表了。写动态内存分配的程序应注意，请尽量对  
分配是否成功进行检测。 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
```