CSDN

博客 (http://feed.csdn.net/?ref=toolbar)

学院 (http://edu.csdn.net?ref=toolbar)

下载 (http://db 更多 ▼







登录 (https://passport.csdn.http://white.in/loghrended/bayshateled/likerity//passport.csdn.net/account/mobileregister?ref=toolbar&action= ref=toolutar)_source=csdnblog1)

结构体指针链表

2013年11月24日 00:02:25

标签: 结构体指针链表 (http://so.csdn.net/so/search/s.do?g=结构体指针链表&t=blog)

m 2353

用指针处理链表 (转)

辩表概述

链表是一种常见的重要的数据结构。它是动态地进行存储分配的一种结构。

链表有一个 头指针 变量,它存放一个地址,该地址指向一个元素,链表中每一个元素称为 结点,每个结点都应包括 两个部分,一为用户需要用的实际数据,二为下一个结点的地址。可以看出,头指针 head 指向第一个元素,第一个元素 又指向第二个元素,。。。。直到最后一个元素,该元素不再指向其他元素,它称为 表尾,它的地址部分放一个 NULL (表示 空地址) 链表到此结束。

可以看到链表中各元素在内存中可以不是连续存放的,要找某一元素,必须先找到上一个元素,根据它提供的下一元 素地址才能找到下一个元素。如果不提供 头指针 head 则整个链表无法访问。

可以看到。这种链表的数据结构,必须利用指针变量才能实现,即一个结点中应包含一个指针变量,用它存放下一结 点的地址。

前面介绍了结构体变量,用它作链表中的结点是最合适的,一个结构体变量包含若干成员,这些成员可以是数值类 型,字符类型,数组类型,也可以是指针类型,我们用这个指针类型成员来存放下一个结点的地址。例如可以设计这样一 个结构体类型:

struct student

{

int num

float score:

struct student *next;

}:

其中成员 num 和 score 用来存放结点中的有用数据(用户需要用到的数据),next 是指针类型成员,它指向 struct student 类型数据(这是 next 所在结构体类型)。一个指针类型的成员既可以指向其他类型的结构体数据,也可以指向自 己所在的结构体类型的数据。现在 next 是 struct student 类型中的一个成员,它又指向 struct student 类型的数据。用这种 方法就可以建立链表。

请注意: 只是定义一个 struct student 类型,并未实际分配存储空间,只有定义了变量才分配内存单元。

7.2 简单链表

下面通过一个例子来说明如何建立和输出一个简单链表







(http://blog.cs









原创 粉丝 喜欢 (https://gi 31 22 0 utm_sourc

他的最新文章

更多文章 (http://blog.csdn.net/u012890226)

ios开发中的一点小经验 (/haixialu/article/details/52399409)

[NSDate alloc] init]与[NSDate date] (/haixialu/article/details/50553306)

当设置个人热点, 状态栏变高造成布局 出错的处理

(/haixialu/article/details/50267729)

UIScrollView 使其平滑的滚动到指定位

(/haixialu/article/details/50267705)



上海单身公寓















G福高点型软件设计的面ourse/detail/594?]对象 m_source=blog9) **m**p:**从**更准.csdn.net/hui

yiCourse/detail/594?



olog9)

⚠ 内容举报

(H. 免费高纖)。d神経网络的ourse/detail/596? 原理及结构设计 utm_source=blog9) http://exam.net/hui

yiCourse/detail/596? utm_source=blog9)

TOP 返回顶部

热门文章

UIScrollView 使其平滑的滚动到指定位置 (/haixialu/article/details/50267705)

```
#include <stdio.h>
#include <stdlib.h>
#define NULL 0
struct student
{
    long num:
float score;
    struct student *next;
wid main()
struct student a. b. c. *head. *p:
    a.num = 99101; a.score = 89.5;
    b.num = 99103; b.score = 90;
    c.num = 99107; c.score = 85;//对结点的 num 和 score 成员赋值
    head = &a;//将结点 a 的起始地址赋给头指针 head
    a.next = &b;//将结点 b 的起始地址赋给 a 结点的 next 成员
    b.next = &c;
    c.next = NULL;// c 结点的 next 成员不存放其他结点地址
    p = head;//使 p 指针指向 a 结点
    do
       printf("%ld %5.1f\n", p->num, p->score);// 输出 p 指向的结点的数据
       p = p->next;//使 p 指向下一结点
    }while(p!= NULL);//输出完 c 结点后 p 的值为 NULL
    system("pause");
}
运行结果
99101 89.5
99103 90.0
99107 85.0
```

7.3 处理动态链表所需的函数

(1) malloc 函数

void *malloc(unsigned int size);

作用是在内存的动态存储区中分配一个长度为 size 的连接空间。些函数的值(即返回值)是一个指向分配空间起始地 址的指针(基类型为 void)。如果些函数未能成功地执行(例如内存空间不足)则返回空指针 NULL。

(2) calloc 函数

void *calloc(unsigned n, unsigned size);

其作用是在内存的动态区存储中分配 n 个长度为 size 的连续空间。函数返回一个指向分配空间起始地址的指针,如果 分配不成功,返回 NULL。

用 calloc 函数可以为一维数组开辟动态存储空间, n 为数组元素个数,每个元素长度为 size。

(3) free 函数

void free(void *p);

其作用是释放由 p 指向的内存区,使这部分内存区能被其它变量使用, p 是最后一次调用 calloc 或 malloc 函数时返 回的值。free 函数无返回值。

请注意:以前的C版本提供的 malloc 和 calloc 函数得到的是指向字符型数据的指针。ANSI C 提供的 malloc 和 calloc 函数规定为 void * 类型。

7.4 建立动态链表

所谓建立动态链表是指在程序执行过程中从无到有地建立起一个键表,即一个一个地开辟结点和输入各结点数据,并 建立起前后相链的关系。

#include <stdio.h> #include <stdlib.h> **4362**

Xcode(//网络请求 uld no#1b)e loade **3150**

结构体指针链表(s/16869703) **2349**

Objective-C常用 haixialu/article/c /人程序加盟



CI 2067

 \triangle 内容举报

TOP 返回顶部

```
#define NULL 0
#define LEN sizeof(struct student)
struct student
{
    long num;
float score;
    struct student *next;
struct student *create()
    struct student *p1, *p2, *head;
    int num;
    float score:
    int n = 0;
    head = NULL;
    p1 = p2 = (struct student *)malloc(LEN);
    printf("please input num and score.\n");
    scanf("%d,%f", &p1->num, &p1->score);
    while(p1->num != 0)
    {
         n ++;
         if(n == 1)
             head = p1;
         else
             p2->next = p1;
         p2 = p1;
         p1 = (struct student *)malloc(sizeof(struct student));
         printf("please input num and score.\n");
         scanf("%d,%f", &p1->num, &p1->score);
    }
    p2->next = NULL;
    return head;
}
void printlist(struct student *head)
    struct student *p;
    p = head;
    if(head != NULL)
    {
         do
         {
             printf("num=%d score=%f\n", p->num, p->score);
             p = p->next;
         }while(p != NULL);
}
void main()
```



⚠
内容举报

TÔP SE DITE TO

```
{
    struct student *head;
    head = create();
    printlist(head);
    system("pause");
}
ß
以下是对链表的各种操作
予印链表
void printlist(struct student *head)
struct student *p;
p = head;
if(head != NULL)
{
 do
 {
  printf("num=%d score=%5.2f\n", p->num, p->score);
  p = p->next;
 } while (p != NULL);
}
/* while(p -> next != NULL)
{
 printf("num=%d score=%f\n", p->num, p->score);
 p = p->next;
}*/
}
删除节点
struct student *delNode(struct student *head, int num)
{
printf("delNode.\n");
struct student *p1, *p2;
if(head == NULL)
 printf("The List is NULL.\n");
}
else
{
 p1 = head;
 while(p1->next != NULL && p1->num != num)
 {
  p2 = p1;
  p1 = p1->next;
 }
 if(p1->num == num)
 {
  if(p1 == head)
  head = p1->next;
  else
  p2->next = p1->next;
 }
```



⚠
内容举报

命 返回顶部

```
else
 printf("Can not find list num.\n");
}
return head;
}
产新节点
struct student *update(struct student *head, int index, int num, float score)
\square
printf("update.\n");
struct student *p;
if(head == NULL)
{
 printf("The List is NULL.\n");
}
else
{
 p = head;
 while(p->next != NULL && p->num != index)
 {
 p = p->next;
 }
 if(p->num == index)
  p->num = num;
  p->score = score;
 }
  printf("Can not find list index.\n");
}
return head;
}
增加节点
struct student *add(struct student *head, int index, int num, float score)
{
printf("add.\n");
struct student *p1, *p2, *p3;
if(head == NULL)
 printf("The List is NULL.\n");
}
else
{
 p1 = p2 = head;
 while(p1->next != NULL && p1->num != index)
  p1 = p1->next;
  p2 = p1;
 }
 if(p1->num == index)
  p3 = (struct student *)malloc(LEN);
  p3->num = num;
```



 \triangle 内容举报



```
p3->score = score:
  if(p2->next == NULL)
  p2->next = p3;
  p3->next = NULL;
1else
\bigcap_{p3->next} = p2->next;
   p2->next = p3;
 }
 else
  printf("Can not find list index.\n");
return head;
}
```



Д

相关文章推荐

为什么链表操作过程中对于结构体指针,都要用malloc来开辟内存空间 (/qqyuanhao163/artic...

int a=0和int a区别 前者是在内存中开辟了一块叫做a的内存空间,里面存放0;后者表示将要开辟内存空间了,但是还没有开 辟 简单的说、int a=0相当于建了一座仓库、然后再里面装了货物 ...

qqyuanhao163 (http://blog.csdn.net/qqyuanhao163) 2015-05-24 12:30 🕮2742

结构体指针怎么用-简单的链表 (/weixin_36748426/article/details/53301500)

目的: 用结构体指针做一个简单的链表 #include #include using namespace std; struct student { int ID; string...



weixin_36748426 (http://blog.csdn.net/weixin_36748426) 2016-11-23 10:04 Q203



大话成像之《数字成像系统》

CSDN学院10月课程热度排行榜首位曝光! 学过了都说好!

 $(http://www.baidu.com/cb.php?c=lgF_pyfqnHmsrHTYrjT0lZ0qnfK9ujYzP1nsrjDz0Aw-lgF_pyfqnHmsrHTYrjT0lZ0qnfK9ujYzP1nsrjDyyzP1nsyyzP1nsrjDyyzP1nsrjDyyzP1nsyyzP1nsyyzP1nsyyzP1nsyyzP1nsyyzP1nsyyzP1ns$ 5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dbnAwBrjPhPHKbuH0LnH0v0AwY5HDdnH0knj0drHc0lgF_5y9YIZ0lQzquZR8mLPbUB48ugfEpZNGXy-jULNzTvRETvNzpyN1gvw-IA7GUatvrjqdIAdxTvqdThP-5yF_UvTkn0KzujYzPsKBUHYs0ZKz5H00ly-b5HDdP1f1PWD0Uv-b5HDzrH63nHf0mvb5HTzPWb1n6KElv3qn0KsXHYznjm0mLFW5HckPjf)

链表创建,结构体指针 (/qq_29832469/article/details/67640737)

链表理解, 结构体指针, 结构体内存之我见

qq_29832469 (http://blog.csdn.net/qq_29832469) 2017-03-28 19:43 \(\mathbb{Q}\)89

 \triangle 内容举报



C++指向结构体变量的指针与链表结构的关系应用 (/yangbo_hbzjk/article/details/7438428)

指向结构体变量的指针与链表结构的关系应用 1、 一个结构体变量的指针就是该变量所占据的内存段的起始地址。可以设一 个指针变量,用来指向一个结构体变量,此时该指针变量的值是结构体变量的其实地址。指针...







基于结构体的循环链表 (http://download.csdn.net/detail/xiao20101028/4...

2012-04-24 18:45 776B



约瑟夫环一维数组、一维结构体数组、循环链表 (http://download.csdn.net...

2012-06-14 11·11 3KB

下载

用结构体变量和指向结构体变量的指针构成链表 (/jrz1900/article/details/47035587)

测试描述:建立一个简单的静态链表,它由3个学生数据的结点组成。输出各结点中的数据。 代码如下: #include using na mespace std: struct Student { ...



jrz1900 (http://blog.csdn.net/jrz1900) 2015-07-24 09:28



结构体数据类型与链表 (http://download.csdn.net/detail/cgilove5185/15...

(http://download

2009-07-21 00:11 712KB



C语言 XX管理系统 不用结构体 链表 (http://download.csdn.net/detail/blu...

2012-04-13 10:13 290KB 下载

3.结构体与链表 (/u012637501/article/details/45269401)

转载请标明出处: http://blog.csdn.net/u012637501 一、结构体 struct Student *p: p是一个struct Student *类型的指 针变...



🥁 u012637501 (http://blog.csdn.net/u012637501) 2015-04-25 13:05 🕮622



结构体数据类型与链表 (http://download.csdn.net/detail/xiaoheliu720/41...

(http://download

2012-03-29 12:48 710KB

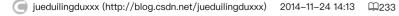


约瑟夫环(一位数组 结构体 循环链表)(http://download.csdn.net/detail...

2012-06-09 09:22 32KB

struct和typedef struct、enum与typedef enum、链表与结构体的区别 (/jueduilingduxxx/ar...

一、struct和typedef struct区别 1) 在C中定义一个结构体类型要用typedef: typedef struct Student { int a; }Stu; // 写...





c语言汽车销售查询系统 结构体 链表 (http://download.csdn.net/detail/cz...

2012-05-28 10:43 3.49MB 下载

结构体与函数、创建链表课堂举例 (http://download.csdn.net/detail/y574886995/4952379)



返回顶部

内容举报

TOP



2013-01-02 20:05 2KB

(http://download

【openjudge 计算概论(A)】[结构体与链表练习] (/reverie_mjp/article/details/52810496)

浩然气、赤子心,千里快哉风



reverie_mjp (http://blog.csdn.net/reverie_mjp) 2016-10-13 21:58 🕮 158





学生成绩管理系统 C语言 C++ 详细注释 可运行 结构体数组 链表 课程设计 (...

下载 2017-07-19 15:50 10KB



基于结构体的循环链表 (http://download.csdn.net/detail/huangshiruge/4...

(http://download

2012-05-05 17:52 776B

下载

C语言的大BOSS:结构体+单链表的学习体会 (/william441/article/details/51503203)

1.结构体: 本质上是一种在已有基本数据类型上复合而成的数据类型,它可以像int、float等类型一样使用(可以生成对象、指 针、数组等)。 注意结构体与结构体变量的区分: 如: ...



WILLIAM441 (http://blog.csdn.net/WILLIAM441) 2016–05–26 00:08
 □190





结构体与链表 ppt课件 (http://download.csdn.net/detail/zkgcs/3402649)

(http://download

2011-06-28 21:45 268KB

下载



 \triangle 内容举报

TOP 返回顶部