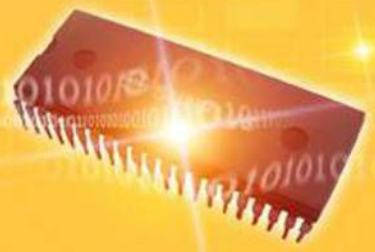
凌阳教育
www.sunplusedu.com
值得信赖的教育品牌

嵌入式系统工程师



Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

数据链表

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

大纲

- 链表的基本概念
- 单向链表的操作
- 其它链表的概念及应用

大纲

- 链表的基本概念
- 单向链表的操作
- 其它链表的概念及应用

链表的基本概念

- 问题？
- 假如：做一个班级信息管理系统，统计班级学生的信息

  而我们事先不知道班级人数，或者知道人数，但是中间人员可能发生变化：比如有新同学加入，有同学请假，又或者我们需要统计班级的平均成绩等等
- 假如：要做一个类似QQ、飞秋类似的通信软件，其中有一个功能，类似用户上下线检测：有新的用户上线、下线实时更新显示，可以实时查询在线状态、按姓名排序等
- **以上问题如何使用学过的C语言知识处理呢？**

链表的基本概念

- 使用数组远远不能达到我们的要求

  因为数组必须实现确定大小，不能实现动态申请、释放

- 使用malloc动态内存分配也无法实现

  malloc申请的空间，不能实现局部申请、释放

- 这里我们学习一种很强大也很重要的数据结构——链表

链表的基本概念

- **定义：**

  链表是一种**物理存储**上**非连续**，数据元素的**逻辑顺序**通过链表中的**指针**链接次序，实现的一种**线性**存储结构。

- 特点：

  链表由一系列节点（链表中每一个元素称为节点）组成，节点在运行时**动态生成（malloc），**每个节点包括两个部分：

  一个是存储数据元素的**数据域**

  另一个是存储下一个节点地址的**指针域**。

链表的基本概念

我们假设把一个链表比喻成一串灯笼

1. 每个**灯笼**相当于连表中的一个**节点**
2. 灯笼上的**字**相当于链表的**数据域**
3. 灯笼下面的**钩子**，相当于链表的**指针域**
4. 灯笼通过钩子一个个**串链**起来，链表通过**指针域**链起来（指针存放下一个节点的地址）

- 我们可以根据需要**增加、减少、排列**灯笼，类似的根据需要**调整链表节点**的位置

大纲

- 链表的基本概念
- 单向链表的操作
  - 链表的结构
  - 链表的构建
  - 链表的遍历、查找、释放
  - 链表的删除、有序插入
- 其它链表的概念及应用

链表的基本概念

链表的构成:

- 链表由一个个节点构成，每个节点一般采用结构体的形式组织，例如:

```
typedef struct student
{
    int num;
    float score;
    struct student *next;
}STU;
```

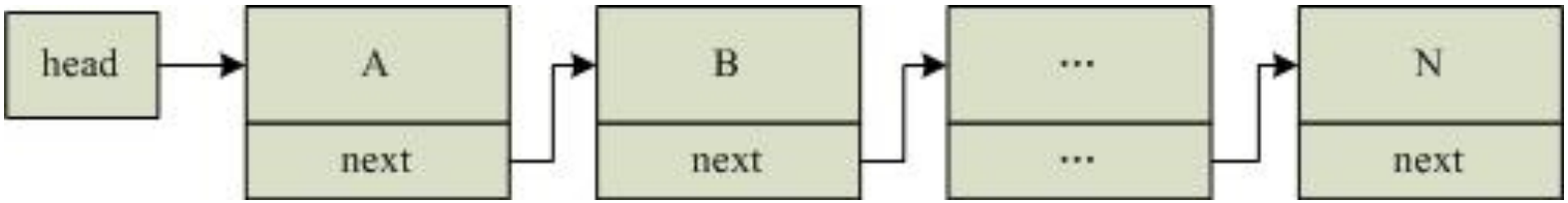
- 链表节点分为两个域

数据域: 存放各种实际的数据，如: num、score等

指针域: 存放下一节点的首地址，如: next等.

链表的结构

- 链表通过在每个**结构体变量（节点）**中设置指向**下一个变量**的指针域，并存入下一个变量的地址，实现将一组结构体变量链成一条链表
- 链表中的每个节点的单元一般都没有名字，它们是通过动态分配函数生成的，并由上一个节点来记录他的位置
- 链表的一般结构如下：



链表的基本概念

- 特点

① 链表作为一个线性存储结构，链表的使用比数组更加灵活（前面所讲的数组都是在静态存储区中定义的数组）.

② 构造一个链表时不一定在程序中指定链表的长度（节点的个数），可以在程序的运行过程中动态的生成一个链表.

③ 链表使用完后可以通过调用free释放节点的方式完成对整个链表空间的释放.

大纲

- 链表的基本概念
- 单向链表的操作
  - 链表的结构
  - 链表的构建
  - 链表的遍历、查找、释放
  - 链表的删除、有序插入
- 其它链表的概念及应用

链表的构建

- 我们以一个宿舍的信息管理系统为例，学习一个链表的创建过程及其它操作
  - 按照初始人数依次输入每个学生的信息，并建立连接关系（创建链表）
  - 对整个链表的信息进行必要的操作

    比如查找信息（查找链表）、修改信息（替换链表）、加入新的学生信息（插入链表）、删除旧信息（删除链表）、排列顺序（排列链表）等等.

链表的构建

- 链表的构建
  - 构建算法：

    首先申请一个链表节点，然后为该节点成员赋值，最后将链表节点添加到链表中
  - 链表节点的添加有多种形式，我们简单介绍两种

1. **添加到链表尾部：**

   顺序创建，新加入的节点放在**链表尾部**

2. **添加到链表头部：逆序创建**

   逆序创建，新加入的节点放在**链表头部**

链表的构建

1. 添加到链表尾部—分两种情况:

- 当链表**为空**时，将链表头直接指向待添加节点（该节点成为了链表的第一个节点）；
- 链表**不为空**时，首先遍历链表找到**链表尾节点**，然后将待添加节点**挂接到尾节点上**.

creat_link_order.c

链表的构建

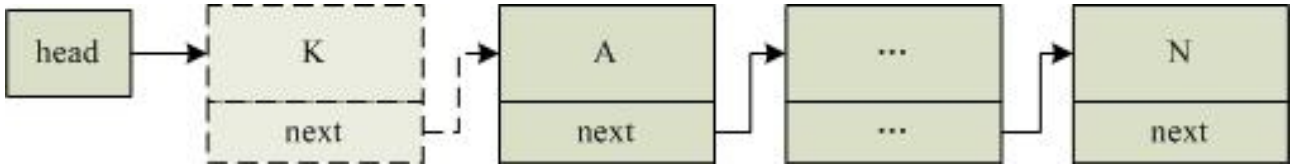
2. 添加到链表头部—分两种情况:

- 情况一同上
- 链表不为空时

首先将之前的第一个链表节点**挂接到新插入**的节点上

然后将链表头指向新插入的节点,

creat_link_reversed.c

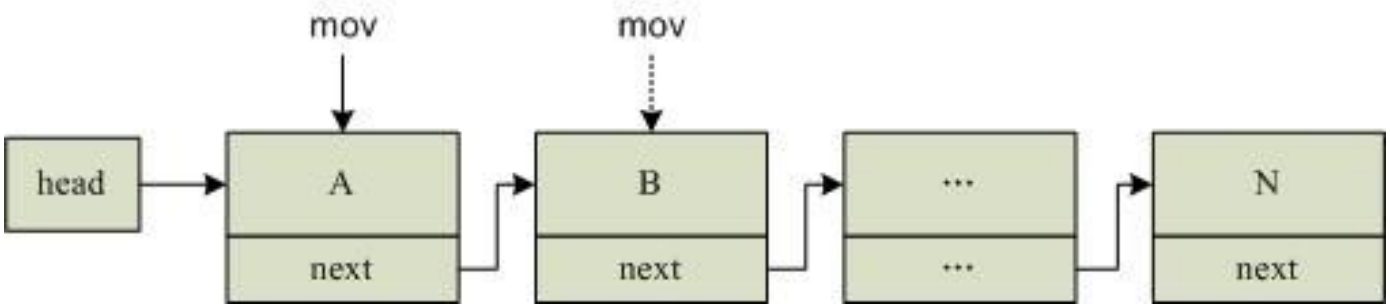


大纲

- 链表的基本概念
- 单向链表的操作
  - 链表的结构
  - 链表的构建
  - 链表的遍历、查找、释放
  - 链表的删除、有序插入
- 其它链表的概念及应用

链表的遍历、查找、释放

- **遍历链表：遍历输出链表所有节点**
  - a）得到链表第一个节点的地址，即head的值
  - b）设一个临时指针变量p_mov，指向第一个节点head，即可获取p_mov所指节点的信息
  - c）使p_mov后移一个节点，即可访问下一节点，直到链表的尾节点（注意结尾判断条件）



链表的遍历、查找、释放

- **查找节点：按照指定关键字查找所需节点**

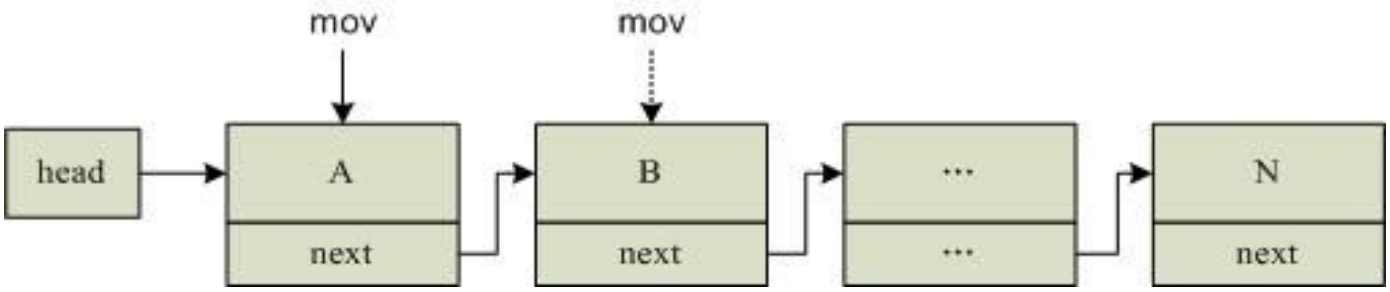
a）得到链表第一个节点的地址，即head的值

b）设一个临时指针变量p_mov，指向第一个节点head，即可获取p_mov所指节点的信息

c）比较是否是要查找的节点

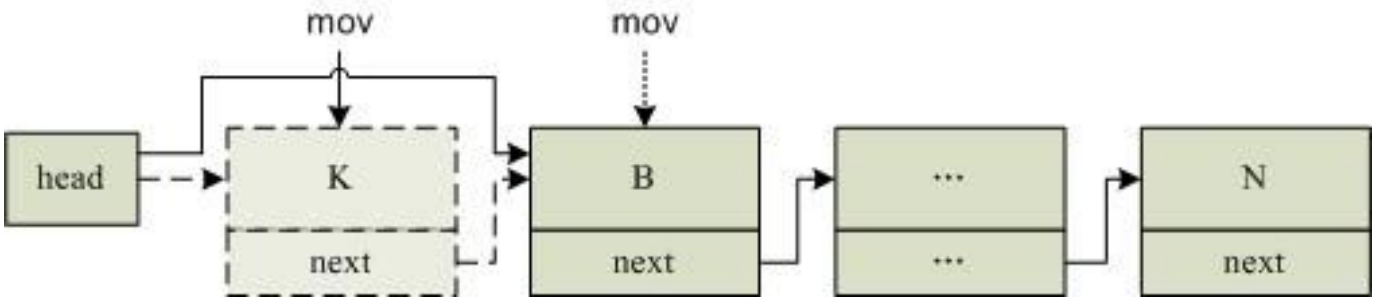
是，则返回相应节点地址，停止查找

不是，使p_mov后移一个节点，即可访问下一节点，直到链表的尾节点(注意结尾判断条件)，最后找不到返回NULL



链表的释放

- **释放链表**
  - 同遍历链表类似，区别在于p_mov每指向某个节点后都将该节点释放
  - 释放前要先保存下一个节点，释放后**备份恢复给p_mov**，否则释放了当前节点，下一个节点的地址就将失去
  - 依次将所有节点释放后，最后返回NULL（标示释放完毕）



大纲

- 链表的基本概念
- 单向链表的操作
  - 链表的结构
  - 链表的构建
  - 链表的遍历、查找、释放
  - 链表的删除、有序插入
- 其它链表的概念及应用

链表的删除

- **删除链表节点**
  - 删除是将某一节点从链中摘除

    将待删节点与其前一节点解除联系（中间或尾部）或本阶段删除（头节点），并释放相应空间(free)

  - 删除的第一步是找到要删除的节点

    同查找算法，如果找不到或链表为空，提示未找到

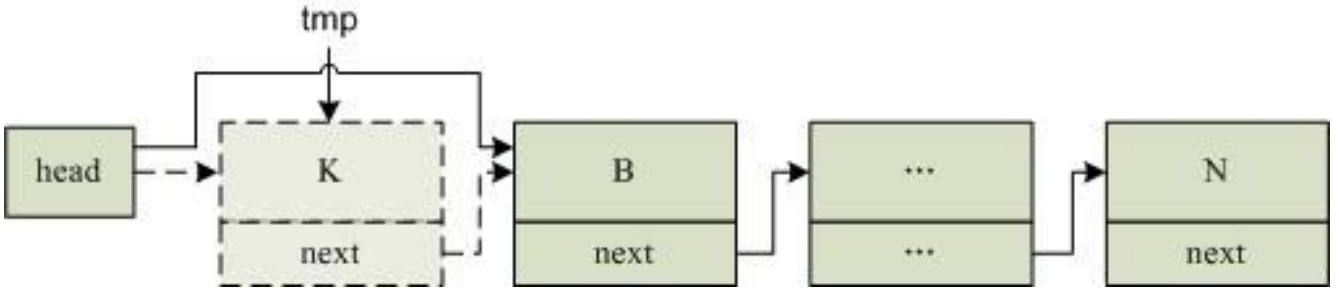
  - 找到后根据情况删除此节点

    分为两种情况：第一个节点，后面节点

链表的删除

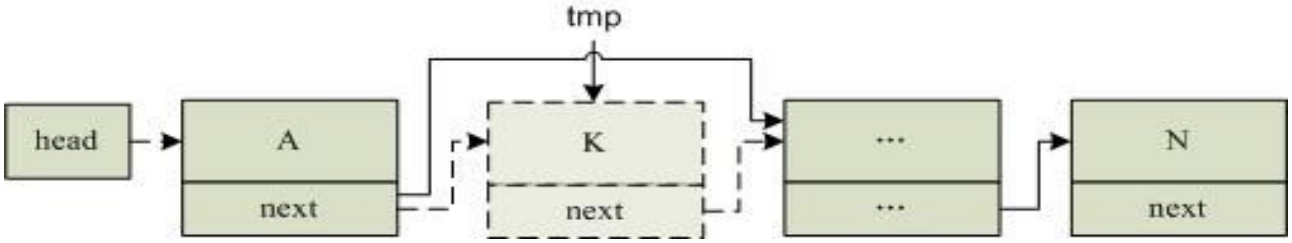
- 被删除节点是第一个节点：

  只需使head指向第二个节点即可



- 被删节点不是第一个节点：

  使被删节点的前一节点指向被删节点的后一节点即可



链表的插入

- **插入链表（有序插入）**
  - 在一个链表的指定位置插入节点，要求链表本身必须是已按某种规律排好序的，例如:

    例如：在学生数据链表中，按学号顺序插入一个节点.
  - **链表的插入分为四种情况:**

    ①原链表为空：只需使head指向被插节点即可.

    ②在第一个节点之前插入：使head指向被插节点，被插节点的next指向原来的第一节点.
    - 在中间位置插入：使插入位置的前一节点的next指向被插节点，使被插节点的next指向插入位置的后一节点.
    - 在表末插入，使链表尾节点next指向被插节点，被插节点next置为NULL .

大纲

- 链表的基本概念
- 单向链表的操作
  - 链表的结构
  - 链表的构建
  - 链表的遍历、查找、释放
  - 链表的删除、有序插入
- 其它链表的概念及应用

链表排序、逆序

- **排序**
  - 当链表本身是无序的时候
  - 我们需要对链表的所有数据进行排序
  - 算法同：冒泡法、选择法

- **逆序**
  - 将链表的所有节点逆序存放，原来的头节点变为结尾，原来的尾节点，变为头节点

链表的插入

- 单向循环
- 双向链表
- 双向循环
- 共享链表
- ...

凌阳教育
www.sunplusedu.com

值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com