



www.enjoylinux.cn

第八课

线性表的链式存储结构



版权声明：本课件及其印刷物、视频的版权归成都国嵌信息技术有限公司所有，并保留所有权力：任何单位或个人未经成都国嵌信息技术有限公司书面授权，不得使用该课件及其印刷物、视频从事商业、教学活动。已经取得书面授权的，应在授权范围内使用，并注明“来源：国嵌”。违反上述声明者，我们将追究其法律责任。

顺序表的思考



www.enjoylinux.cn

讨论中。。。

顺序表的最大问题是插入和删除需要移动大量的元素！

如何解决？

学生A：在线性表数据元素之间空出位置，为以后插入使用。

学生B：这样不行！中间无论空多少都有可能用完！

学生A：那不是无解了嘛！

学生B：我觉得让每个元素都知道他的下个元素就行了，哪有空插哪。

嵌入式Linux技术咨询QQ号：550491596

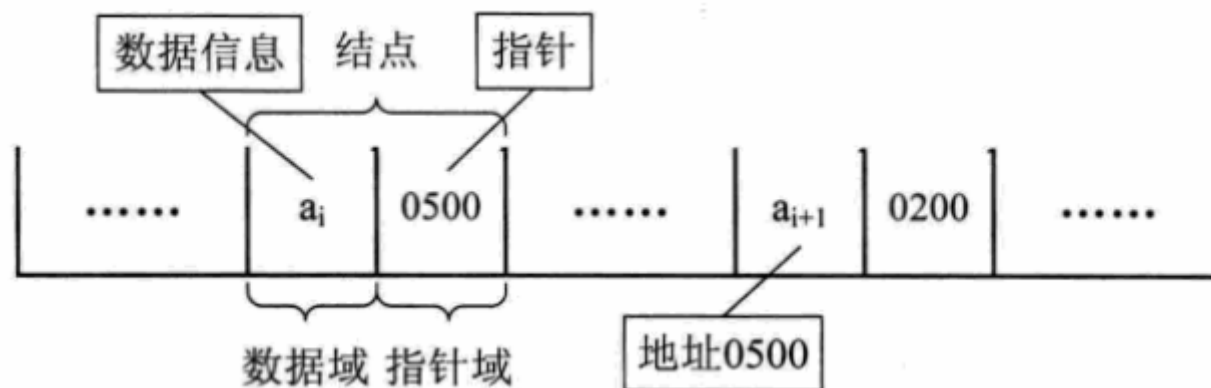
嵌入式Linux学习交流QQ群：65212116



链式存储结构

❖ 链式存储定义

为了表示每个数据元素与其直接后继元素之间的逻辑关系，每个元素除了存储本身的信息外，还需要存储指示其直接后继的信息。



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116





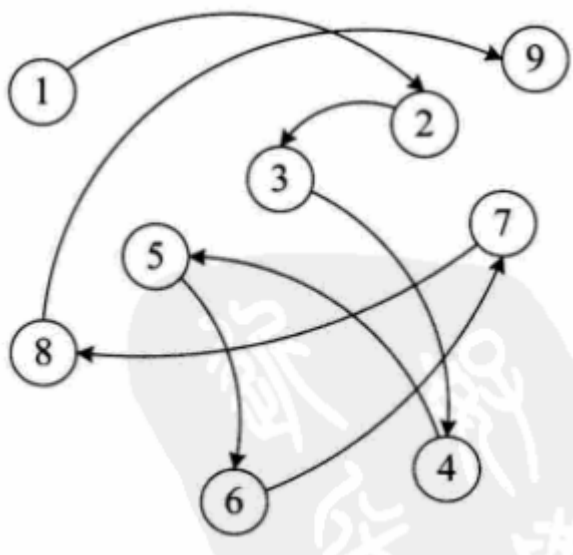
www.enjoylinux.cn

链式存储结构

❖ 链式存储逻辑结构

n 个结点链接成一个链式线性表的结构叫做链表，

当每个结点中只包含一个指针域时，叫做单链表。



嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



链表的基本概念



❖ 表头结点

- 链表中的第一个结点，包含指向第一个数据元素的指针以及链表自身的一些信息

❖ 数据结点

- 链表中代表数据元素的结点，包含指向下一个数据元素的指针和数据元素的信息

❖ 尾结点

- 链表中的最后一个数据结点，其下一元素指针为空，表示无后继

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

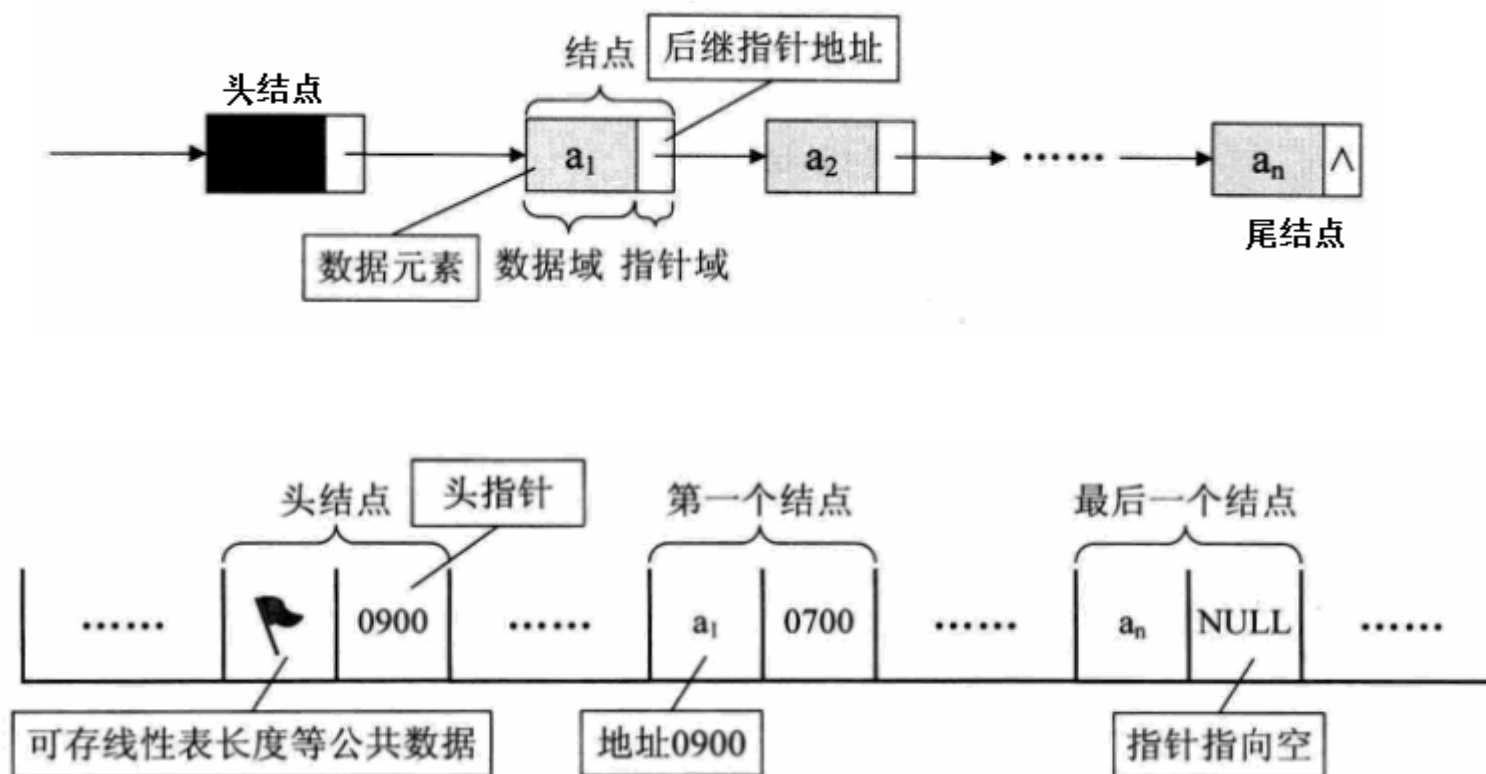




www.enjoylinux.cn

链表的基本概念

❖ 单链表示例



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



链式存储结构



www.enjoylinux.cn

- ❖ 在C语言中可以用结构体来定义链表中的指针域
- ❖ 链表中的表头结点也可以用结构体实现

```
typedef struct _tag_LinkListNode LinkListNode;  
struct _tag_LinkListNode  
{  
    LinkListNode* next;  
};
```

结点指针域定义

```
typedef struct _tag_LinkList  
{  
    LinkListNode header;  
    int length;  
} TLinkList;
```

头结点定义

```
struct Value  
{  
    LinkListNode header;  
    int v;  
};
```

数据元素定义示例

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



链式存储结构

❖ 获取第pos个元素操作

- 判断线性表是否合法
- 判断位置是否合法
- 由表头开始通过next指针移动pos次后，当前元素的next指针即指向要获取的元素

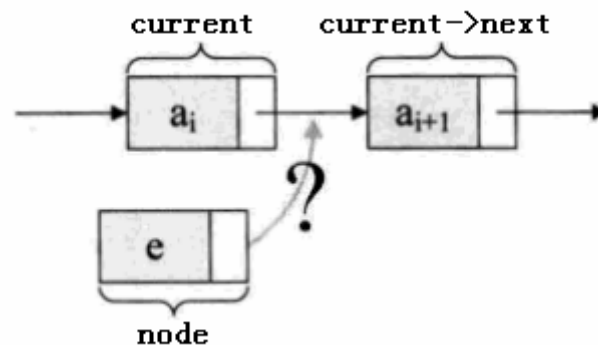
```
LinkListNode* current = (LinkListNode*) list;  
  
for(i=0; i<pos; i++)  
{  
    current = current->next;  
}  
  
ret = current->next;
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

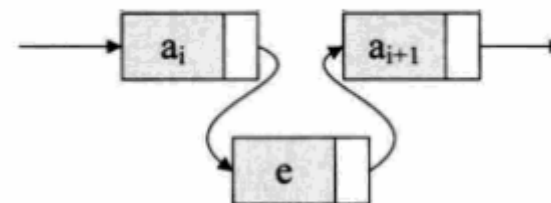
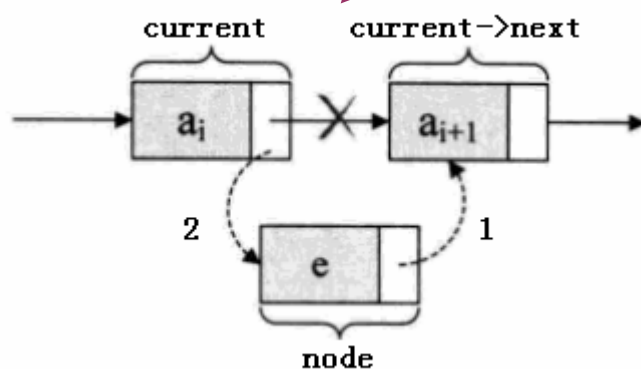


链式存储结构

❖ 插入元素操作



```
node->next = current->next;  
current->next = node;
```



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116





www.enjoylinux.cn

链式存储结构

❖ 插入元素到位置pos的算法

- 判断线性表是否合法
- 判断插入位置是否合法
- 由表头开始通过next指针移动pos次后，当前元素的next指针即指向要插入的位置
- 将新元素插入
- 线性表长度加1



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



链式存储结构

❖ 插入元素到位置pos的算法

```
LinkListNode* current = (LinkListNode*) list;

for(i=0; (i < pos) && (current->next != NULL); i++)
{
    current = current->next;
}

node->next = current->next;
current->next = node;

sList->length++;
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

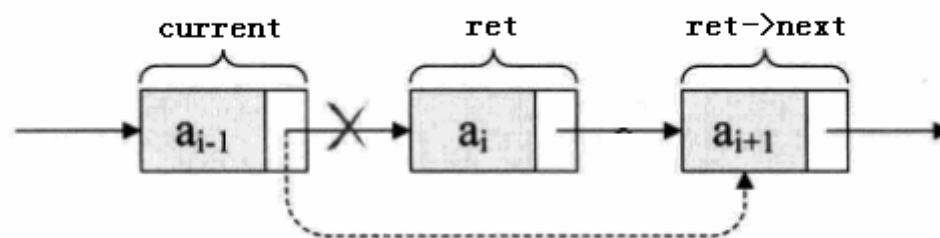




www.enjoylinux.cn

链式存储结构

❖ 删除元素操作



```
current->next = ret->next;
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116





www.enjoylinux.cn

链式存储结构

❖ 删除第pos个元素的算法

- 判断线性表是否合法
- 判断插入位置是否合法
- 获取第pos个元素
- 将第pos个元素从链表中删除
- 线性表长度减1



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



链式存储结构

❖ 删除第pos个元素的算法

```
TLinkList* sList = (TLinkList*) list;
LinkListNode* ret = NULL;
int i = 0;

if( (sList != NULL) && (0 <= pos) && (pos < sList->length) )
{
    LinkListNode* current = (LinkListNode*) list;

    for(i=0; i<pos; i++)
    {
        current = current->next;
    }

    ret = current->next;
    current->next = ret->next;

    sList->length--;
}
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



手把手教你写代码



创建可复用单链表

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



小结



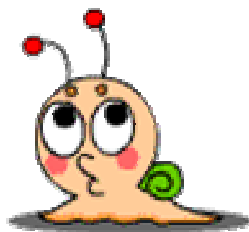
www.enjoylinux.cn

❖ 优点:

- 无需一次性定制链表的容量
- 插入和删除操作无需移动数据元素

❖ 缺点:

- 数据元素必须保存后继元素的位置信息
- 获取指定数据的元素操作需要顺序访问之前的元素



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



课后练习



www.enjoylinux.cn

- ❖ 顺序表的实现中为什么保存的是具体数据元素的地址？而保存的地址存放在unsigned int中，而不是具体的指针类型中？为什么不是void*呢？



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



课后练习



www.enjoylinux.cn

- ❖ 以下关于链式存储结构的叙述中正确的是
- A. 链式存储结构不是顺序存取结构
 - B. 逻辑上相邻的结点物理上必相邻
 - C. 可以通过计算直接确定第*i*个元素
 - D. 插入和删除操作方便，不必移动其它结点



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

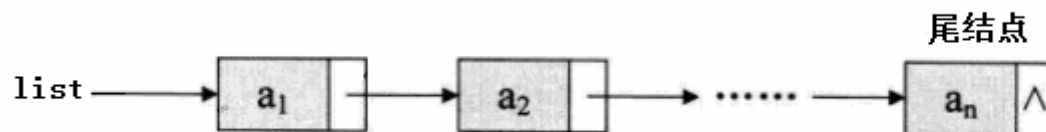


课后练习



www.enjoylinux.cn

- ❖ 在单链表的实现和应用中，还有一种无表头结点的单链表。根据我们列出的操作，编程实现这种无表头结点的单链表，并对比表头实现方式和无表头实现方式哪种更好。



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



课后练习



www.enjoylinux.cn

- ❖ 为我们实现的顺序表和单链表添加一个反转操作。这个操作使得链表中的次序反转，即第一个元素变为最后一个元素，第二个元素变为倒数第二个元素。。。如：abcd反转的结果为dcba。
- ❖ 要求时间复杂度达到 $O(n)$ 。

`SeqList_Reverse(SeqList* list);`

`LinkList_Reverse(LinkList* list);`

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116

