



www.enjoylinux.cn

第九课

无招胜有招 - 静态链表



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

继续讨论中。。。

单链表完美解决了顺序表的问题!

还有其它改进顺序表的方法吗?

学生A: 单链表很完美, 我觉得顺序表可以退休了。

学生B: 我也觉得, 老师为什么还要教我们顺序表呢? !

学生A: 那不是为了展现单链表的强大嘛!

学生B: 看来我们可以彻底抛弃顺序表了。

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



顺序表的思考



www.enjoylinux.cn

牛人小C出场。。。

顺序表有优势，单链表也同样有缺点！

学生C：单链表虽然综合起来胜过了顺序表，但是也有不足之处。

学生B：什么不足之处呢，Linux内核都用的单链表哦！

学生A：就是，一般的大型项目都能找到单链表的身影！

学生C：。。。我想你们会明白的。。。

小C转身离去。。。

嵌入式Linux技术咨询QQ号：550491596

嵌入式Linux学习交流QQ群：65212116



单链表的相对劣势

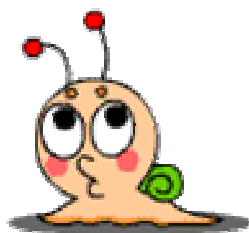


www.enjoylinux.cn

单链表的实现严重依赖指针!

数据元素中必须包含一个额外的指针域!

没有指针的程序设计语言无法实现!



嵌入式Linux技术咨询QQ号: 550491596

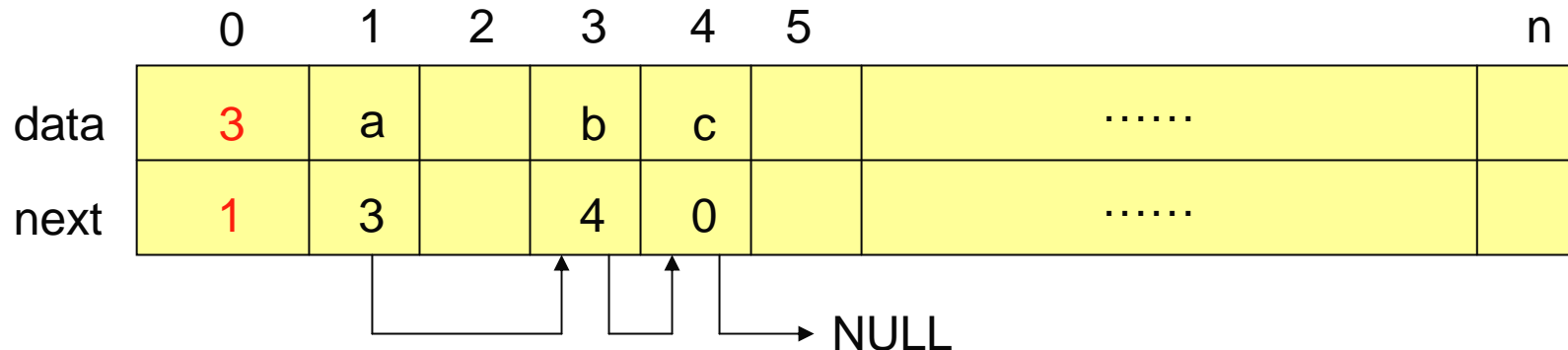
嵌入式Linux学习交流QQ群: 65212116



顺序表的改进

❖ 静态链表的定义

- 顺序表数组中的元素由两个数据域组成：data和next
- data域用于存储数据
- next域用于存储下一个元素在数组中的下标



header = 1 length = 3 capacity = n

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



静态链表的逻辑结构



www.enjoylinux.cn



静态链表是在顺序表的基础上利用数组实现的单链表！



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



静态链表相关定义



```
typedef struct _tag_StaticListNode
{
    unsigned int data;
    int next;
} TStaticListNode;
```

结点结构体定义

```
typedef struct _tag_StaticList
{
    int capacity;
    TStaticListNode header;
    TStaticListNode node[];
} TStaticList;
```

静态链表结构体定义

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



静态链表操作

❖ 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域移动pos次后，当前元素的next域即要获取元素在数组中的下标

```
sList->node[0] = sList->header;  
  
for(i=0; i<pos; i++)  
{  
    current = sList->node[current].next;  
}  
  
object = sList->node[current].next;
```

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



静态链表操作

❖ 插入元素到位置pos的算法

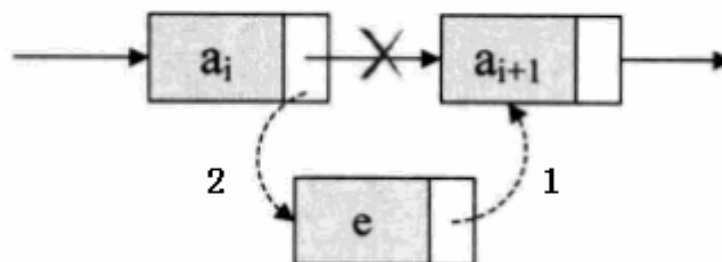
- 判断线性表是否合法
- 判断插入位置是否合法
- 在数组中查找空闲位置index
- 由表头开始通过next域移动pos次后，当前元素的next域为要插入的位置
- 将新元素插入
- 线性表长度加1

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



静态链表操作

❖ 插入元素操作



```
for(i=0; (i<pos) && (sList->node[current].next != 0); i++)  
{  
    current = sList->node[current].next;  
}
```

```
sList->node[index].next = sList->node[current].next;  
sList->node[current].next = index;
```

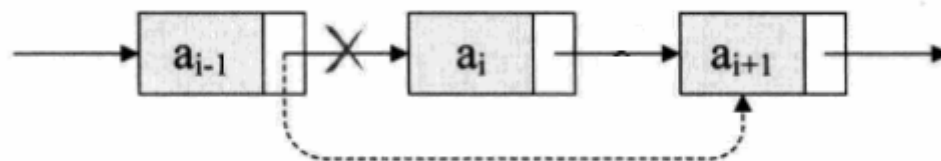
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



静态链表操作

❖ 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



```
object = sList->node[current].next;
```

```
sList->node[current].next = sList->node[object].next;
```

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



www.enjoylinux.cn



创建可复用静态链表

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116

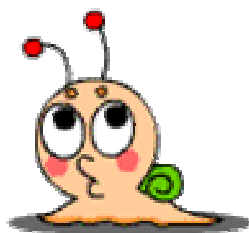


小结



www.enjoylinux.cn

- ❖ 静态链表其实是单链表的另一种实现方式
- ❖ 静态链表的实现“媒介”不是指针而是数组
- ❖ 静态链表主要用于不支持指针的程序设计语言中
- ❖ 静态链表的实现是一种内存管理的简易方法



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



思考



www.enjoylinux.cn

- ❖ 为什么静态链表结构体中要再定义个header成员，而不直接使用node[0]?

```
typedef struct _tag_StaticList
{
    int capacity;
    TStaticListNode header;
    TStaticListNode node[];
} TStaticList;
```



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

