

# 数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



## 静态链表

- 这一节课，我们试图通过静态链表的讲解来瞻仰古人的伟大！（似乎人总要挂了之后才能变得伟大~~）
- 神马是静态链表呢？又跟古人能有半毛钱关系？
- 地球人都知道C语言是个伟大的语言，他的魅力在于指针的灵活性，使得它可以非常容易地操作内存中的地址和数据，这比其他高级语言更加灵活方便。（面向对象使用对象引用机制间接地实现了指针的某些功能）



## 静态链表

- 但是古人还木有C语言丫，木有JAVA丫，只有原始的Basic，Fortran等早期的编程语言，这些语言没有类似于C的指针功能，但是他们又想描述单链表，就没法实现了，怎么办呢？
- 真是不得不佩服他们的智慧，有人想出了用数组代替指针来描述单链表。大家能猜一猜他们是如何做到的么？
- 在讲解原理之前，先让大家知道这种用数组描述的链表叫做静态链表，这种描述方法叫做游标实现法。



## 静态链表

游标	5	2	3	4	0	6	7	...	1
数据		A	C	D	E			...	
下标	0	1	2	3	4	5	6	...	999

- 线性表的静态链表存储结构

```
#define MAXSIZE 1000
```

```
typedef struct
```

```
{
```

```
    ElemType data; // 数据
```

```
    int cur;      // 游标 (Cursor)
```

```
} Component, StaticLinkList[MAXSIZE];
```



## 静态链表

- 对静态链表进行初始化相当于初始化数组:

```
Status InitList(StaticLinkList space)
```

```
{
```

```
    int i;
```

```
    for( i=0; i < MAXSIZE-1; i++)
```

```
        space[i].cur = i + 1;
```

```
    space[MAXSIZE-1].cur = 0;
```

```
    return OK;
```

```
}
```





## 静态链表

- 小甲鱼备忘录:

- 我们对数组的第一个和最后一个元素做特殊处理, 他们的data不存放数据。
- 我们通常把未使用的数组元素称为备用链表。
- 数组的第一个元素, 即下标为0的那个元素的cur就存放备用链表的第一个结点的下标。
- 数组的最后一个元素, 即下标为MAXSIZE-1的cur则存放第一个有数值的元素的下标, 相当于单链表中的头结点作用。



## 静态链表的插入操作

- 有些喜欢思考的鱼油会说，你这不是挂羊头买鱼肉吗？还是数组丫，貌似没看出太多单链表的端倪~~
- 那我们接着从行为学的角度来剖析下，静态链表如何模拟单链表进行插入和删除的操作呢？
- 现在我们来看看如何实现元素的插入。静态链表中要解决的是：如何用静态模拟动态链表结构的存储空间分配，也就是需要的时候申请，不需要的时候释放。



## 静态链表的插入操作

- 我们前面说过，在动态链表中，结点的申请和释放分别借用C语言的malloc()和free()两个函数来实现。
- 在静态链表中，操作的是数组，不存在像动态链表的结点申请和释放的问题，所以我们需要自己实现这两个函数，才可以做到插入和删除操作。
- 为了辨明数组中哪些分量未被使用，解决的办法是将所有未被使用过的及已被删除的用游标链成一个备用链表，应该如何操作呢？请大家课后头脑风暴一下！

