



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

规格严格 功夫到家



第11章 动态数据结构的C语言实现

动态数组



哈尔滨工业大学

赵玲玲 zhaoll@hit.edu.cn

动态数组

* 动态数组

特点	静态数组	动态数组
创建	定义	内存申请函数
释放	不需要	程序员
内存区域	不定	堆
生存期	不定	程序
数组长度	不变	可变

动态数组

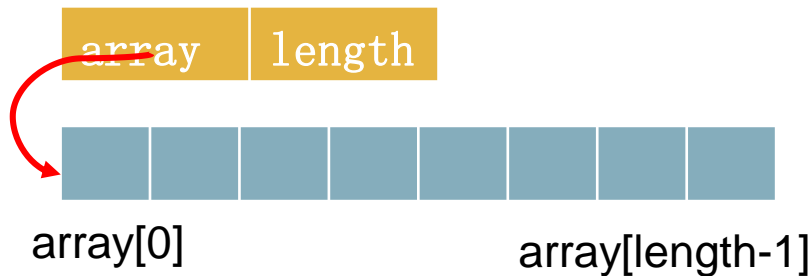
* 动态数组

- * 程序运行中动态申请的连续内存空间
- * 长度动态可变

```
typedef struct  
{  
    int *array;  
    int length;  
} DArray;
```

动态数组的地址

数组长度



动态数组

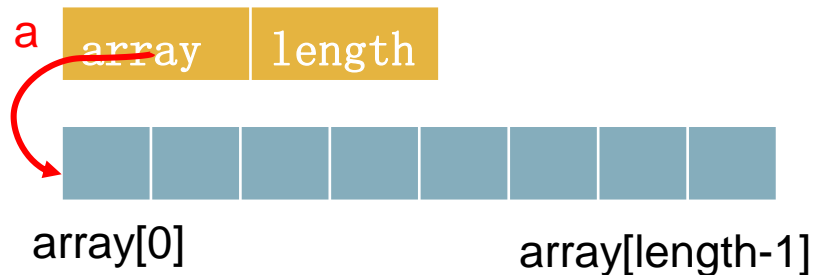
/Example: 创建一个长度为n的动态数组，读入数据后将其长度增长1倍，继续读入新的数据。输出长度增长前后的所有元素。

```
DArray create(int );  
void grow(DArray *, int);  
void print(DArray *);  
void release(DArray *);
```

```
int main()  
{  
    DArray a;  
    scanf("%d",&n);  
    a = create(n);  
    print(&a);  
    grow(&a, n*2);  
    print(&a);  
    release(&a);  
    return 0;  
}
```

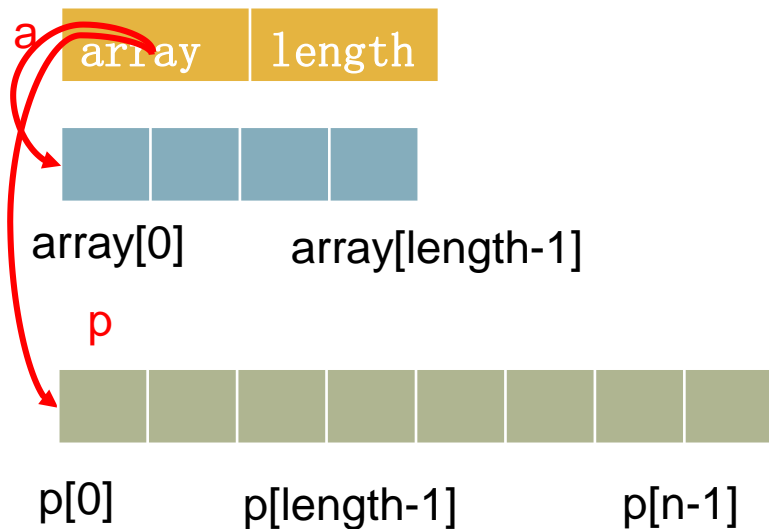
动态数组

```
typedef struct
{
    int *array;
    int length;
} DArray;
```



```
DArray create(int n)
{
    DArray a;
    int i;
    a.array = (int *)malloc(sizeof(int)*n);
    if(a.array == NULL)
    {
        printf("Allocation Error");
        exit(0);
    }
    else
    {
        a.length = n;
        for(i=0; i<a.length; i++)
            scanf("%d",&a.array[i]);
    }
    return a;
}
```

动态数组



```
void grow(DArray *aPtr, int n)
{
    int *p;
    int i;
    p = (int *)malloc(sizeof(int)*n);
    if(p==NULL)
        return;
    else
    {
        for(i=0; i<aPtr->length; i++)
            p[i] = aPtr->array[i];
        for(i=aPtr->length; i<n; i++)
            scanf("%d",&p[i]);
        free(aPtr->array);
        aPtr->array = p;
        aPtr->length = n;
    }
}
```

动态数组

```
void grow(Array *aPtr, int n)
{
    int *p;
    int i;
    p = (int *)realloc(aPtr, sizeof(int)*n);
    if(p==NULL)
        return ;
    else
    {
        for(i=0; i< aPtr ->length; i++)
            p[i] = aPtr ->array[i];
        for(i=aPtr->length; i<n; i++)
            scanf("%d",&p[i]);
        free(aPtr->array);
        aPtr ->array = p;
        aPtr ->length = n;
    }
}
```

动态数组

```
typedef struct
{
    int *array;
    int length;
} DArray;
```

```
void print(DArray *p)
{
    int i;
    for(i=0; i<p->length; i++)
        printf("%d ",p->array[i]);
    return;
}
```

p->array+i

*(p->array+i)

动态数组

```
typedef struct
{
    int *array;
    int length;
} DArray;
```

```
void release(DArray *p)
{
    free(p->array);
}
```