第7章 指针

动态内存分配

动态内存分配

无论是普通的变量，还是数组，都是在编译时就需要确定分配的内存大小。但是，很多时候在编程时，并不能知道程序运行时到底需要多少内存空间，此时需要在运行时，根据实际需要动态向操作系统申请内存空间，称为动态内存分配。

内存分配运算符:malloc、calloc
内存释放运算符:free

动态分配内存

```
void* malloc(unsigned int size);
```

```
void* calloc(unsigned int num,
             unsigned int size);
```

系统找到一块未占用的内存，将其标记为已占用，然后把地址返回，并标记此程序占用此块内存，其它程序不能再用它

```
#include <stdlib.h>
```

动态分配内存

```
void* malloc(unsigned int size);
```

向系统申请大小为size的内存块
把首地址返回
如果申请不成功，返回NULL

动态分配内存

```
void* calloc(unsigned int num,
             unsigned int size);
```

向系统申请num个size大小的内存块
把首地址返回
如果申请不成功，返回NULL

动态分配内存

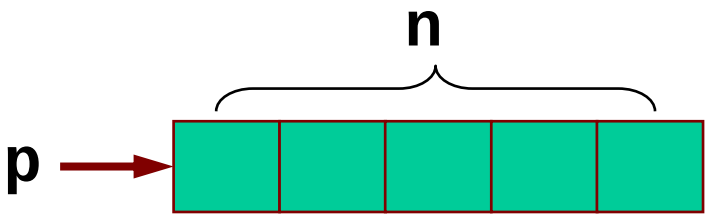
- 释放内存：

```
void* free(void* p);
```

释放由**malloc()**和**calloc()**申请的内存块
p是指向此块内存的指针
free时系统标记此块内存为未占用，可被重新分配

一维动态数组

```c
#include  <stdio.h>
#include  <stdlib.h>main()
{
    int  *p = NULL, n, i, sum;

    printf("Please enter array size:");
    scanf("%d", &n);
    p = (int *) malloc(n * sizeof (int));
    if (p == NULL)
    {
            printf("No enough memory!\n");
            exit(0);
    }

    printf("Please enter the score:");
    for (i=0; i<n; i++)
    {
            scanf("%d", p + i);
    }

    sum = 0;
    for (i=0; i<n; i++)
    {
            sum = sum + *(p + i);
    }

    printf("aver = %d\n", sum/n);
    free(p);
}
```



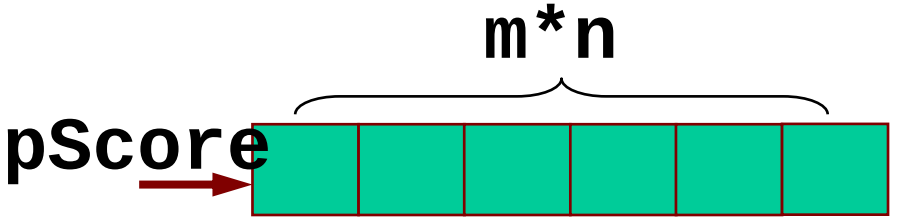
确保指针使用前是非空指针

p+i等价于&p[i]
***(p+i)等价于p[i]**
像使用一维数组一样使用

释放向系统申请的存储空间

二维动态数组

```
main()
{
    int  *pScore = NULL, i, j, m, n, maxScore, row, col;

    printf("Please enter array size m,n:");
    scanf("%d,%d", &m, &n);

    pScore = (int *) calloc(m*n, sizeof (int));
    if (pScore == NULL)
    {
        printf("No enough memory!\n");
        exit(0);
    }

    printf("Please enter the score:\n");
    for (i=0; i<m; i++)
    {
        for (j = 0; j<n; j++)
        {
                scanf("%d", &pScore [i*n+j]);
        }
    }

    maxScore = FindMax(pScore, m, n, &row, &col);

    printf("maxScore = %d, class = %d, number = %d\n",
                maxScore, row+1, col+1);

    free(pScore)
}
```



确保指针使用前是非空指针

&pScore[x]等价于pScore+x

pScore[x]等价于*(pScore+x)

仍然当作一维数组使用

释放向系统申请的存储空间